Einführung in Computational Engineering

Grundlagen der Modellierung und Simulation

TECHNISCHE UNIVERSITÄT DARMSTADT

10. Vorlesung: Zeitkontinuierliche Modellierung und Simulation

16. Dezember 2013

Prof. Dr. Jan Peters

produziert vom HRZ Hochschulrechenzentrum

Überblick der Vorlesungsinhalte

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Einführung
2. Diskrete Modellierung und Simulation
3. Zeitkontinuierliche Modellierung und Simulation
4. Teilschritte einer Simulationsstudie
5. Interpretation und Validierung
6. Modulare und objektorientierte Modellierung und Simulation
7. Regression

Grundlagen der Modellierung und Simulation

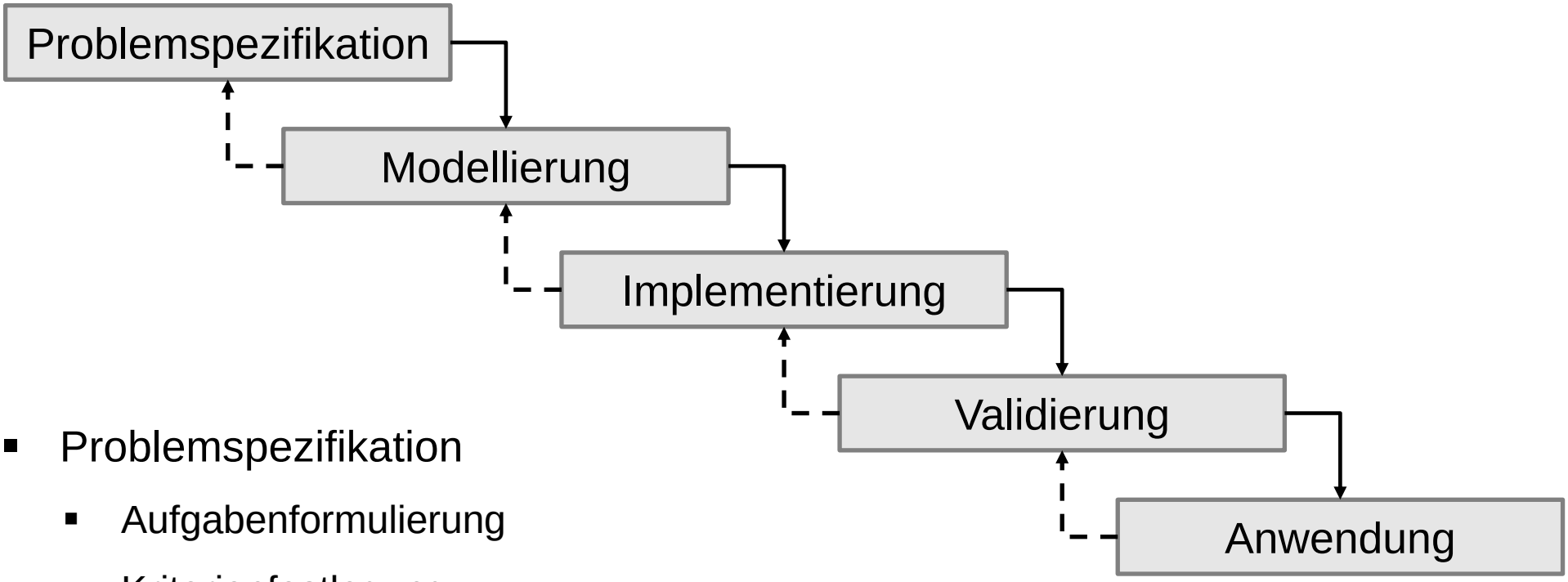
4. TEILSCHRITTE EINER SIMULATIONSSTUDIE

Heutige Lernziele: Kernfragen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wie funktioniert eine Simulationstudie im Detail?
- Was bedeuten Modellbildung, Spezifikation, Implementierung, Validierung, Anwendung im Detail? Sind sie in der Lage, jeden Schritt selber zu tun?
- Wie kann man Simulationswerkzeuge klassifizieren?
- Welche „Levels“ gibt es und was unterscheidet diese?
- Wie funktionieren MATLAB und SIMULINK?
- Wie kann man ODE45 nutzen?
- Wie funktioniert die blockorientiert Darstellung?

4.1 Problemspezifikation



- Problemspezifikation
  - Aufgabenformulierung
  - Kriterienfestlegung
  - Datenerhebung

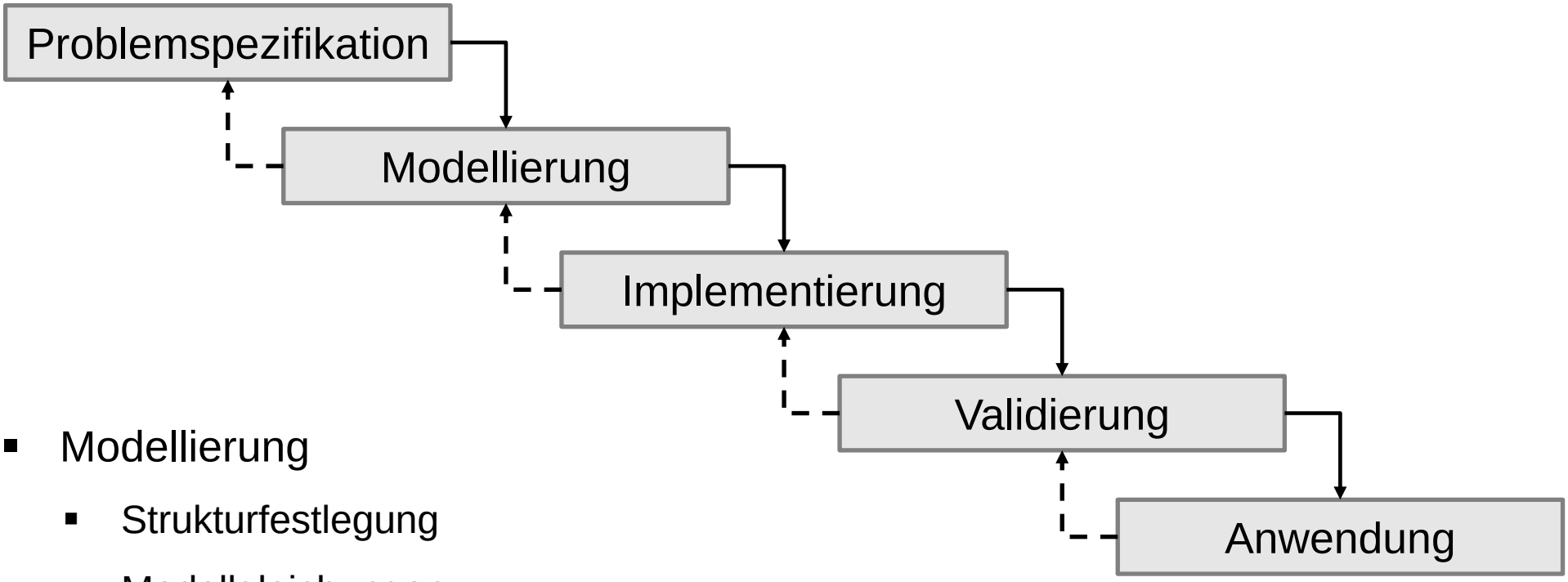
4.1 Problemspezifikation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mögliche Zielsetzungen der Untersuchung
  - Realisierbarkeit
  - Wartezeiten und Durchsatz von Kunden
  - Benötigte Zeiten und auftretende Kräfte bei Unterschiedlichen Schaukelstrategien

(Folien zur Schiffschaukel beruhen auf W. Wiechert, Uni Siegen)

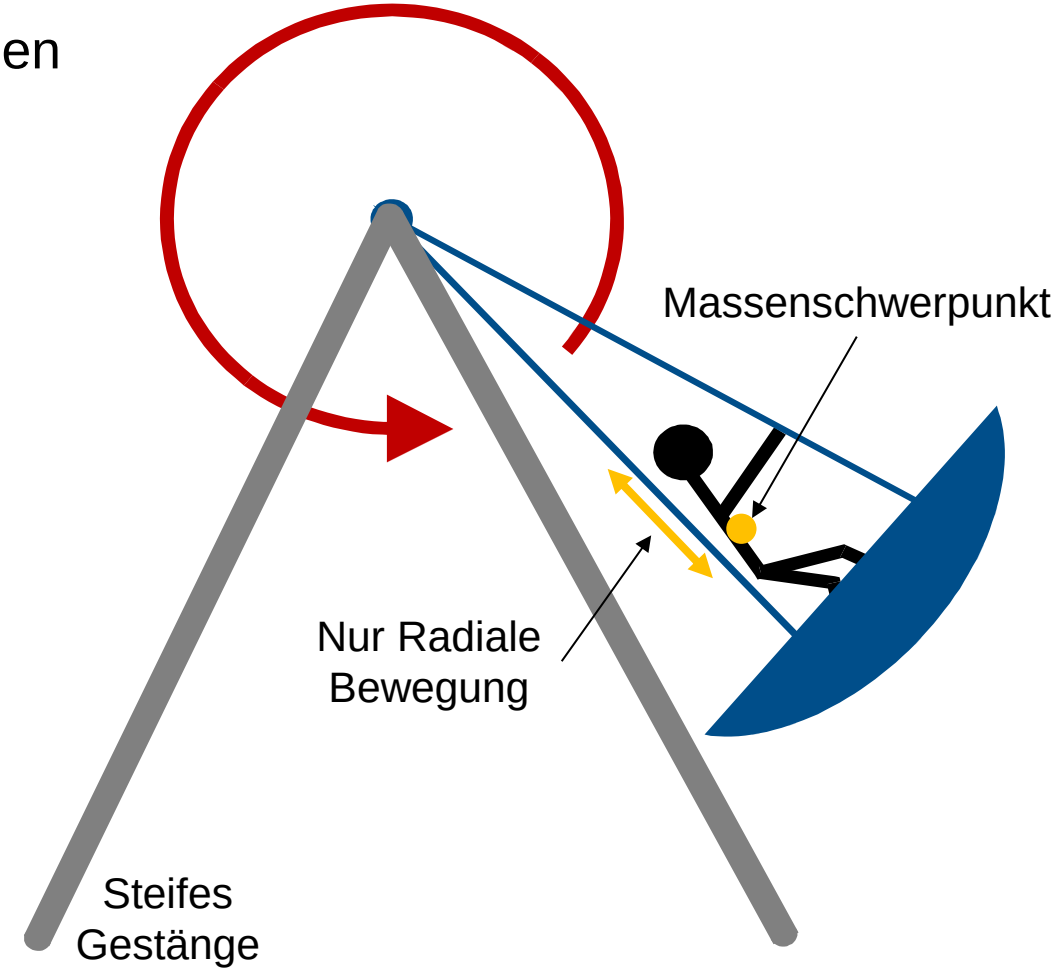
4.2 Modellierung



- Modellierung
  - Strukturfestlegung
  - Modellgleichungen
  - Modellvereinfachung

4.2 Mathematische Modellierung: Annahmen

- Möglicher Grund der Untersuchungen
  - Benötigte Zeiten und auftretende Kräfte bei unterschiedlichen Schaukelstrategien
  - Man nehme an es gebe keinen Luftwiderstand



4.2 Modellierung: Veranschaulichung

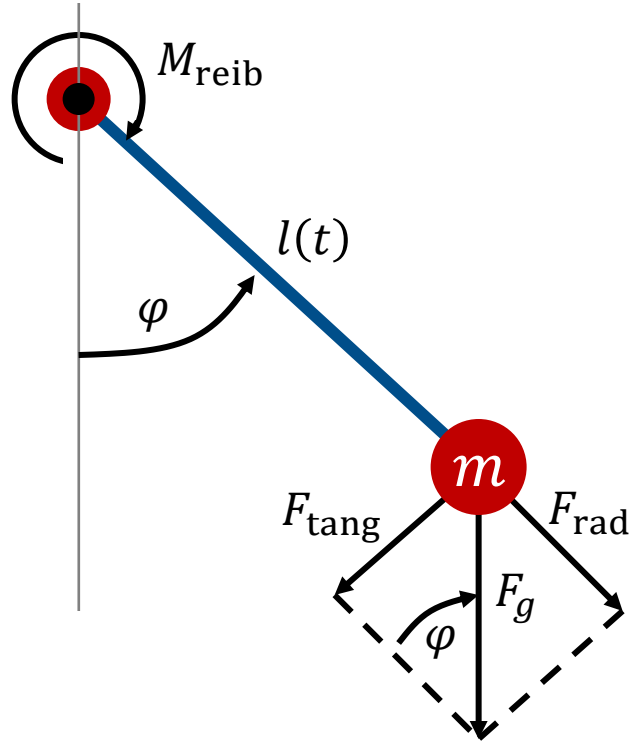
Das Modell kann reale Bewegungen nur
mit gewissen Einschränkungen wiedergeben

4.2 Modellbildung: Physikalisches Ersatzmodell

TECHNISCHE UNIVERSITÄT DARMSTADT

- Drehbewegung in einer Ebene mit fester Drehachse

- Drehimpulssatz
  - „Zeitliche Änderung des Drehimpulses = Summe der äußeren Drehmomente“

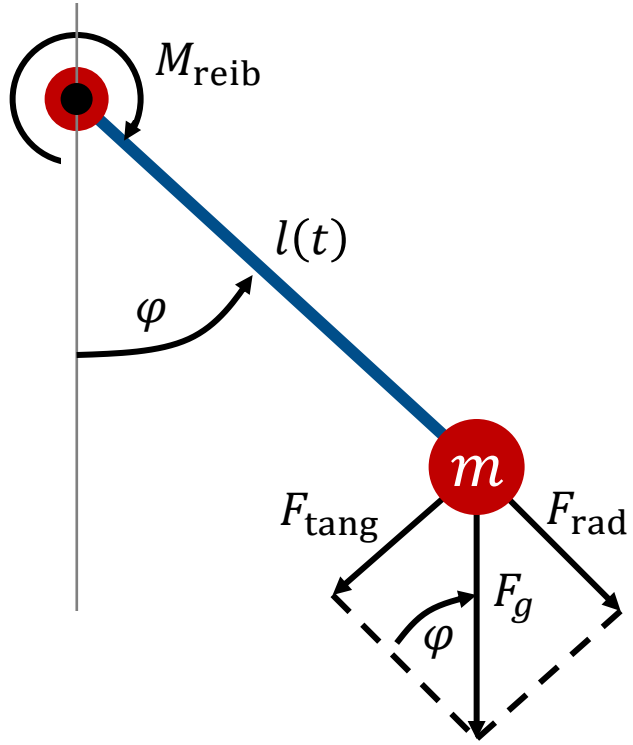


4.2 Modellbildung: Physikalisches Ersatzmodell

- Drehimpuls L bei fester Drehachse
  - $L =$ „Trägheitsmoment $\times$ Winkelgeschwindigkeit“
  - $L(t) = I(t) \cdot \dot{\varphi}(t)$ mit $I(t) = m \cdot l^2(t)$

  - Damit lautet die zeitliche Änderung des Drehimpuls

$$\dot{L}(t) = \dot{I}(t) \cdot \dot{\varphi}(t) + I(t) \cdot \ddot{\varphi}(t)$$
$$= 2m \cdot l(t)\dot{l}(t)\dot{\varphi}(t) + ml^2(t)\ddot{\varphi}(t)$$



4.2 Modellbildung: Physikalisches Ersatzmodell

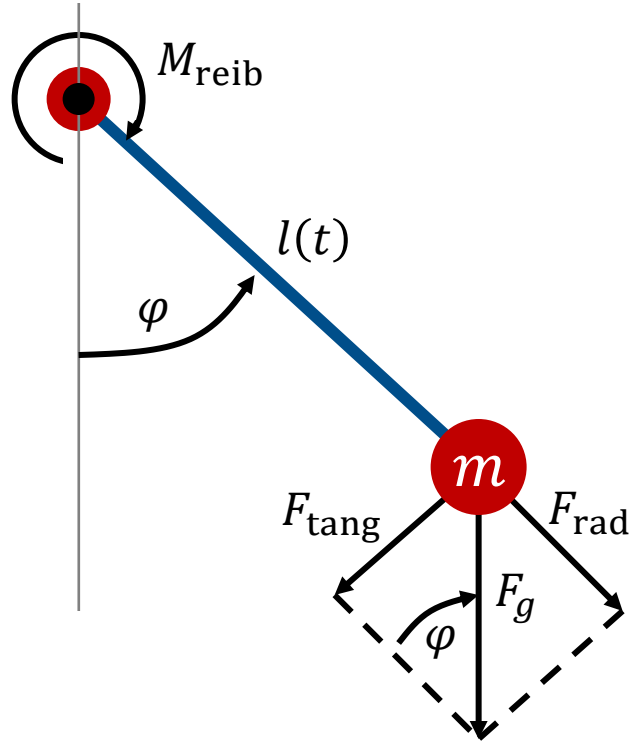
TECHNISCHE UNIVERSITÄT DARMSTADT

- Bestimmung des Tangentialmomentes M_{tang}

$$\sin\varphi = \frac{F_{\text{tang}}}{F_g} \quad \Rightarrow \quad F_{\text{tang}} = mg \cdot \sin\varphi$$

mit $F_g = mg$ und damit folgt für M_{tang}

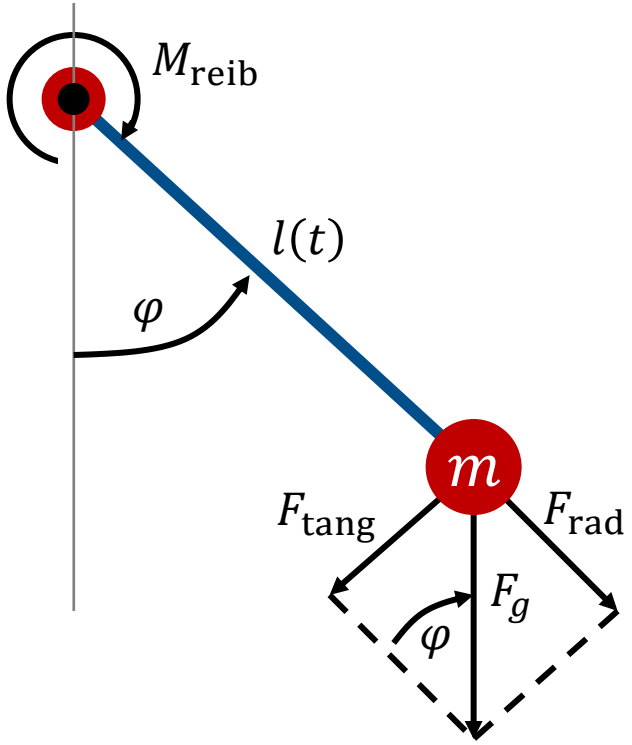
$$M_{\text{tang}} = l \cdot \left(-F_{\text{tang}}\right) = -mgl \cdot \sin\varphi$$



4.2 Modellbildung: Physikalisches Ersatzmodell

TECHNISCHE UNIVERSITÄT DARMSTADT

- Modellierung des Reibungsmomentes M_{reib}
  - $M_{\text{reib}} = -d \cdot \dot{\varphi}$ mit einer Konstanten d



- Einsetzen in den Drehimpulssatz ergibt

$$\dot{L}(t) = M_{\text{tang}} + M_{\text{reib}}$$

$$2m \cdot l\dot{l}\dot{\varphi} + ml^2\ddot{\varphi} = -mgl \sin\varphi - d\dot{\varphi}$$

- Umformung ergibt

$$2\frac{\dot{l}}{l}\dot{\varphi} + \ddot{\varphi} = -\frac{g}{l}\sin\varphi - \frac{d}{ml^2}\dot{\varphi}$$

4.2 Modellbildung: Physikalisches Ersatzmodell

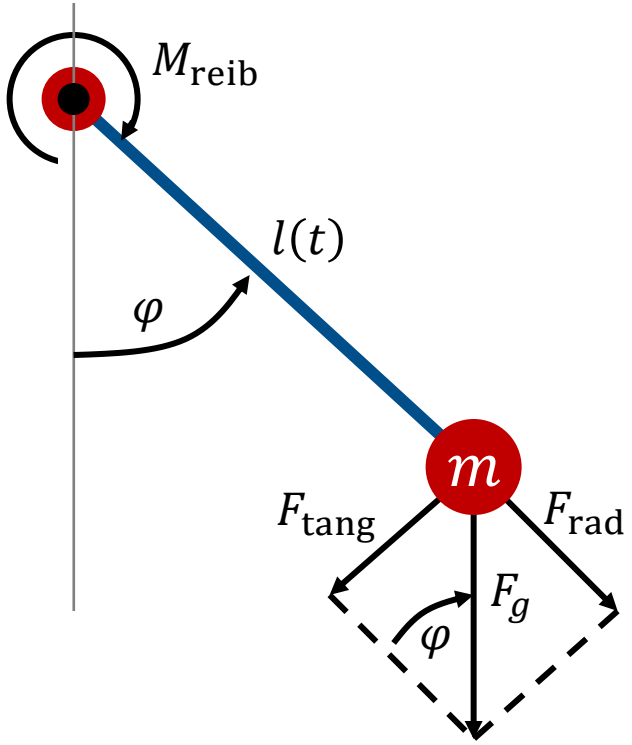
TECHNISCHE UNIVERSITÄT DARMSTADT

- Umformung ergab

$$2\frac{\dot{l}}{l}\dot{\varphi} + \ddot{\varphi} = -\frac{g}{l}\sin\varphi - \frac{d}{ml^2}\dot{\varphi}$$

- Daraus folgt die Bewegungs-DGL

$$\ddot{\varphi} = -\left(2\frac{\dot{l}}{l} + \frac{d}{ml^2}\right)\dot{\varphi} - \frac{g}{l}\sin\varphi$$

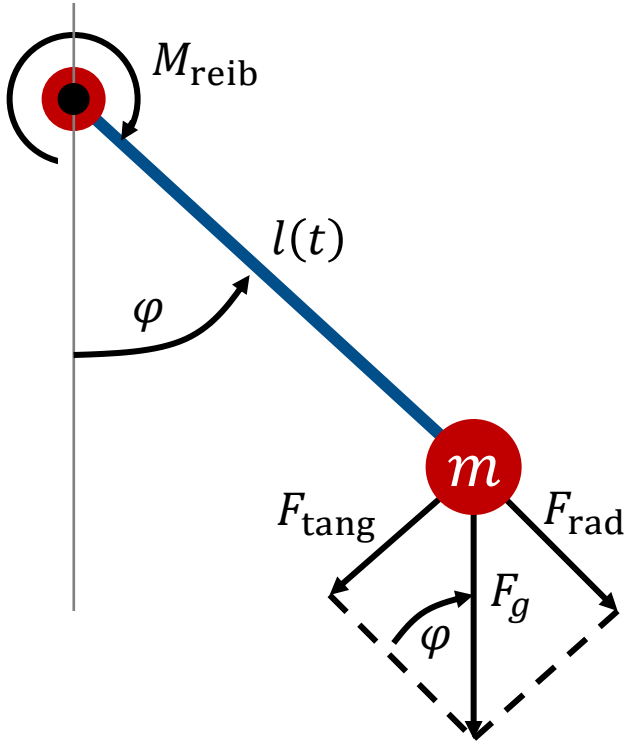


4.2 Modellbildung: Physikalisches Ersatzmodell

- Man transformiert die DGL zu DGL 1.Ordnung
  - Einführung einer Winkelgeschwindigkeit ω

$$\dot{\varphi} = \omega$$

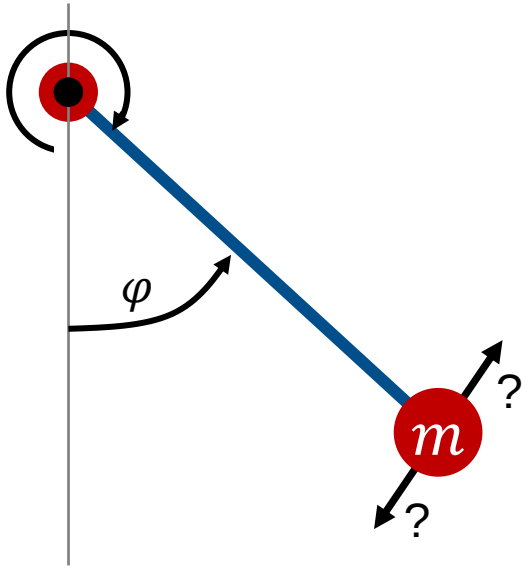
$$\dot{\omega} = -\left(2\frac{\dot{l}}{l} + \frac{d}{ml^2}\right)\omega - \frac{g}{l}\sin\varphi$$



4.2.1 Modell: Zustandsvariablen

- Zustandsvariablen
  - Sind zeitabhängige Größen
  - Legen die aktuelle Konfiguration eines Systems exakt fest
  - Legen den zukünftigen Verlauf genau fest
  - Sind nicht redundant

- Beispiel Schiffsschaukel
  - φ und ω sind Zustandsvariablen
  - φ alleine reicht nicht aus
  - φ, ω und $\dot{\omega}$ sind redundant



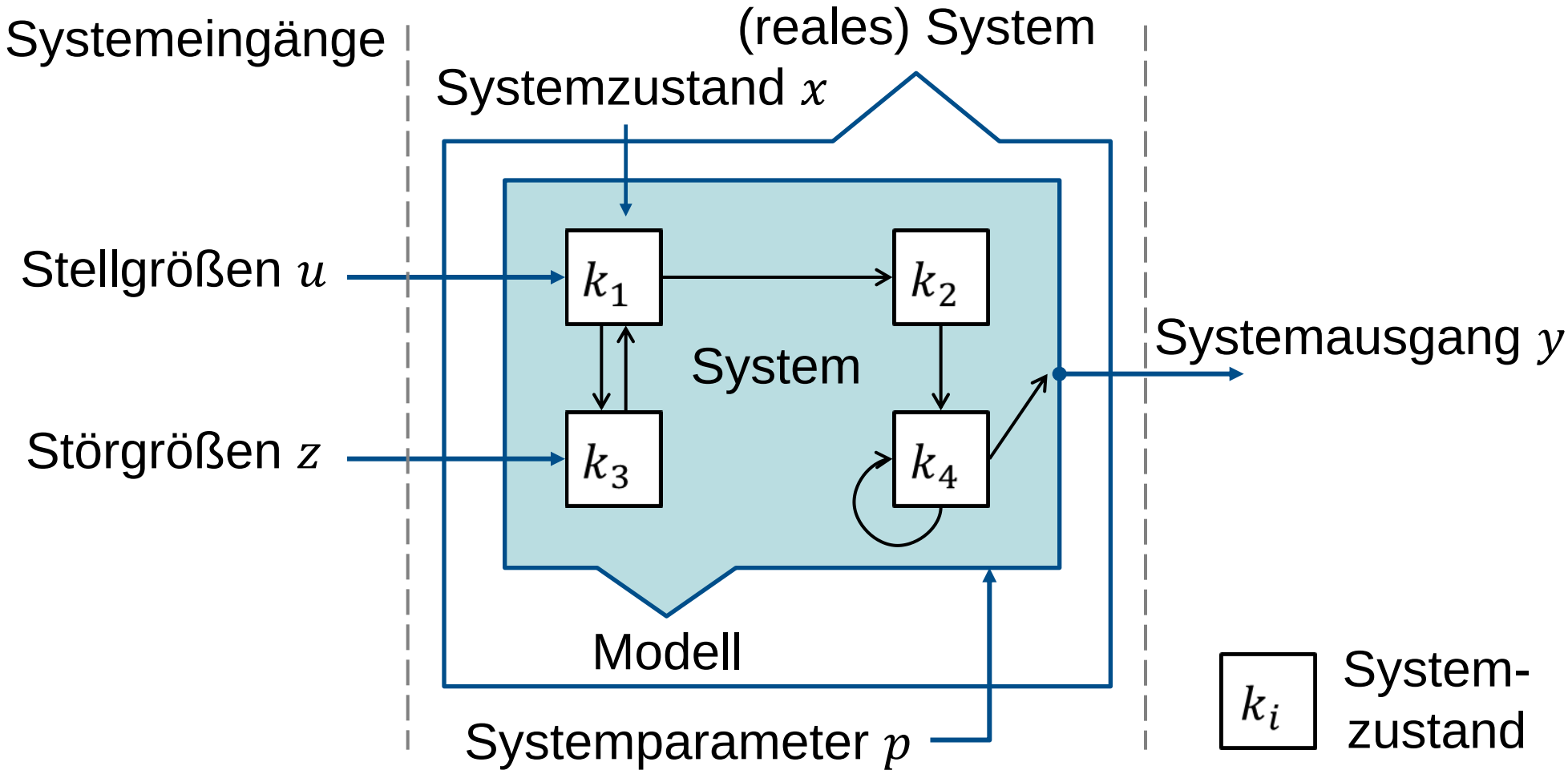
4.2.1 Modell: Zustandsvariablen

TECHNISCHE UNIVERSITÄT DARMSTADT

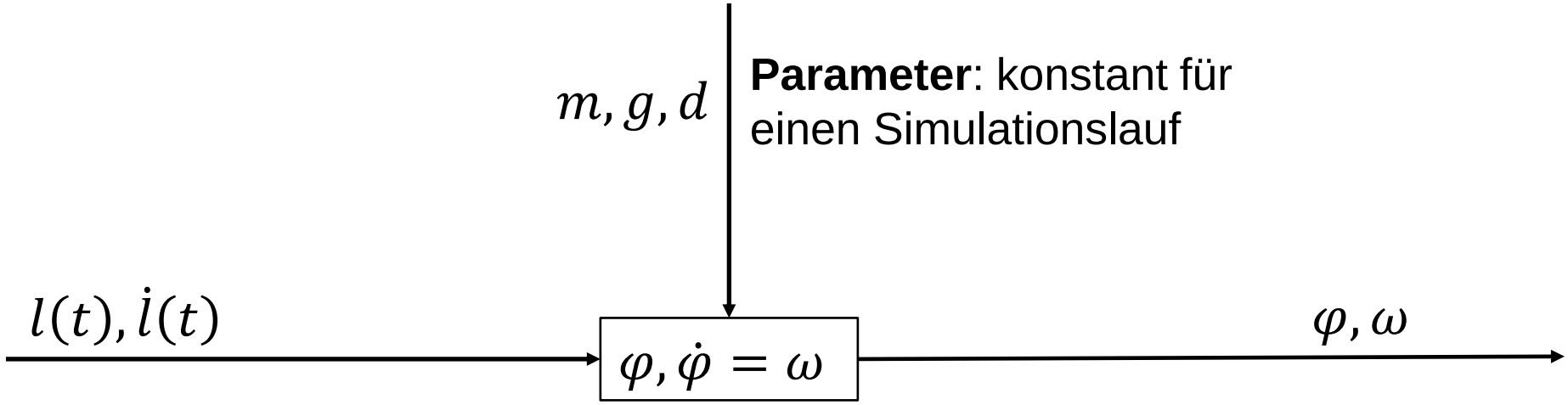
- Grundprinzip:

> Die Festlegung der Zustandsvariablen
> ist der Ausgangspunkt jeder Modellbildung

4.2.2 Modell: Systemstruktur (Wiederholung 3.2)



4.2.2 Modell: Systemstruktur



Eingangsgröße: Während der Simulation veränderbar (Steuervariable)

Systemzustand: charakterisiert das System (Zustandsvariable)

Ausgangsgröße: beobachtbare / messbare / berechenbare aussagekräftige Größen für das Problem

4.2.2 Modell: Systemstruktur

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zustandsvariablen

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \varphi \\ \omega \\ l \end{pmatrix}$$

- Steuervariablen

$$u = (u_1) = (l)$$

- Systemparameter

$$p = \begin{pmatrix} m \\ d \\ g \end{pmatrix}$$

- Zustands-DGLn

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\left(2\frac{u_1}{x_3} + \frac{p_2}{p_1 x_3}\right) x_2 - \frac{p_3}{x_3} \sin x_1$$

$$\dot{x}_3 = u_1$$

4.2.2 Modell: Systemstruktur

- Zustandsvariablen

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \varphi \\ \omega \\ l \end{pmatrix}$$

- Steuervariablen

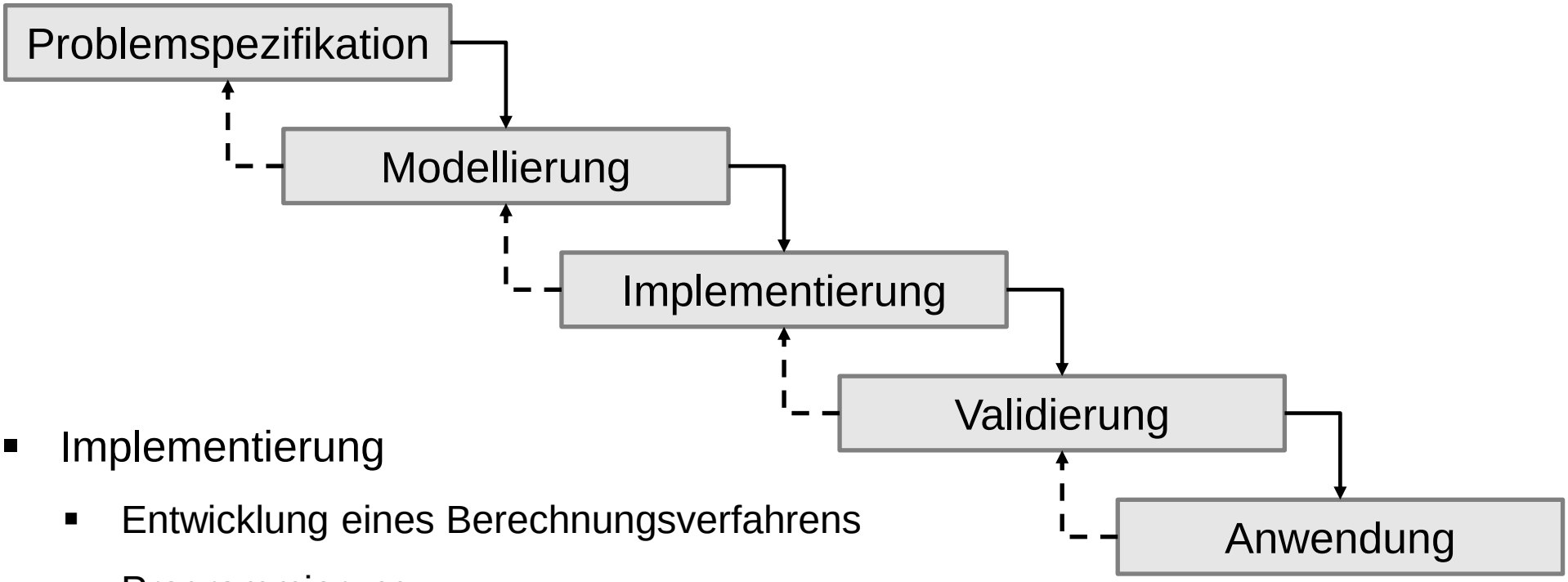
$$u = (u_1) = \left(\dot{l}\right)$$

- **Bemerkung:** Wenn $l(t)$ als Steuervariable vorgegeben ist, dann ist $\dot{l}(t)$ festgelegt.

  Hier wird $\dot{l}(t)$ vorgegeben, das zusammen mit einem Anfangswert $l(0)$ dann $l(t)$ festlegt, vgl. die letzte DGL.

  $l(t)$ wird somit zur Zustandsvariablen.

4.3 Implementierung



- Implementierung
  - Entwicklung eines Berechnungsverfahrens
  - Programmierung
  - Visualisierung

4.3 Implementierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Auswahl oder Entwicklung eines Berechnungsverfahrens (für das jeweilige Modell)
- Programmierung von Modell und Berechnungsverfahren
- Visualisierung von Berechnungsergebnissen
- Laufzeitoptimierung

4.3.1 Implementierung: Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zustandsvariablen sind Funktionen der Zeit

$$\varphi = \varphi(t)$$

$$\omega = \omega(t)$$

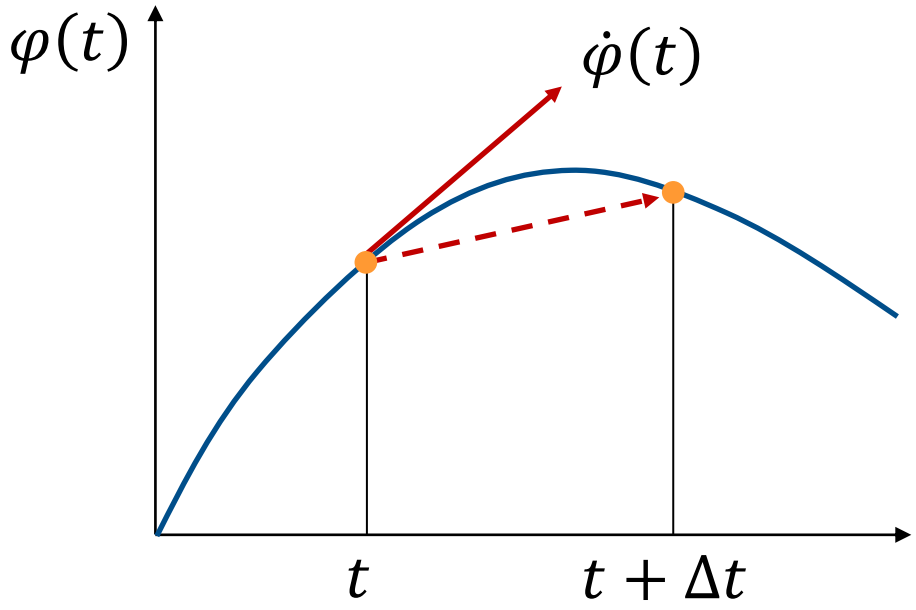
- Näherung für die zeitliche Ableitung durch den Differenzenquotienten

$$\dot{\varphi}(t) \approx \frac{\varphi(t + \Delta t) - \varphi(t)}{\Delta t}$$

$$\dot{\omega}(t) \approx \frac{\omega(t + \Delta t) - \omega(t)}{\Delta t}$$

4.3.1 Implementierung: Motivation

- Die Verwendung des Differenzenquotienten anstelle der Ableitung bedeutet anschaulich die Verwendung der Sekante anstelle der Tangente von φ in t



4.3.2 Implementierung: Euler-Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gleichung (ohne Längenveränderung, d.h. für $\dot{l}(t) = 0$)

$$\frac{\varphi(t + \Delta t) - \varphi(t)}{\Delta t} \approx \dot{\varphi}(t) = \omega(t)$$

$$\frac{\omega(t + \Delta t) - \omega(t)}{\Delta t} \approx \dot{\omega}(t) = -\frac{d}{ml^2}\omega(t) - \frac{g}{l}\sin\varphi(t)$$

- Gesucht ist die numerische Näherungslösung bei $t + \Delta t$ mit gegebenem $\varphi(t), \omega(t)$ und t

4.3.2 Implementierung: Euler-Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

- Numerische Lösung mit dem Euler

$$\varphi(t + \Delta t) \approx \varphi(t) + \Delta t \cdot \dot{\varphi}(t)$$
$$\omega(t + \Delta t) \approx \omega(t) + \Delta t \cdot \dot{\omega}(t)$$

- Damit folgt

$$\varphi(t + \Delta t) \approx \varphi(t) + \Delta t \cdot [\omega(t)]$$

$$\omega(t + \Delta t) \approx \omega(t) + \Delta t \cdot \left[-\frac{d}{ml^2}\omega(t) \; -\frac{g}{l}\sin\varphi(t) \right]$$

4.3.3 Implementierung: Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

$m \coloneqq 100;\ l \coloneqq 2.5;\ d \coloneqq 100;\ g \coloneqq 9.81;$ Modellparameter

$\Delta t \coloneqq 0.01;\ t_{\max} \coloneqq 10;$ Simulationsparameter

$t \coloneqq 0;\ \varphi \coloneqq 1;\ \omega \coloneqq 0;$ Startwerte

$while\ t < t_{\max}\ \ begin$ Zeitschleife

$t, \varphi, \omega\ \ ausgeben;$

$\varphi_{\text{neu}} \coloneqq \varphi + \Delta t \cdot \omega;$ Euler-Schritt

$\omega_{\text{neu}} \coloneqq \omega + \Delta \cdot \left(-\frac{d}{ml^2} \cdot \omega - \frac{g}{l} \sin \varphi\right);$ Euler-Schritt

$t \coloneqq t + \Delta t;\ \ \varphi = \varphi_{\text{neu}};\ \ \omega \coloneqq \omega_{\text{neu}};$

$end;$

4.3.4 Auswertung und Interpretation

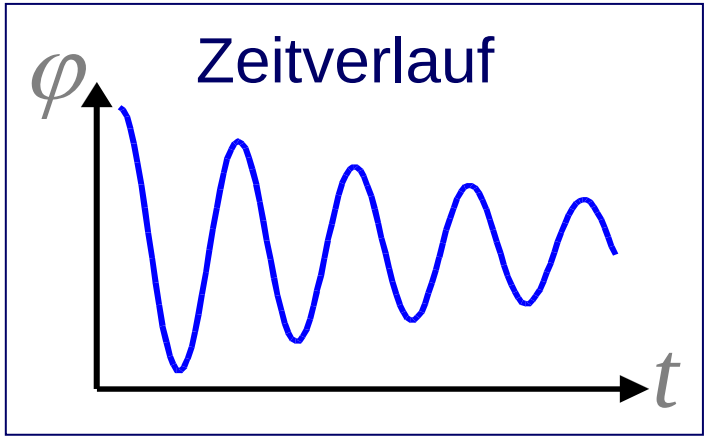
TECHNISCHE UNIVERSITÄT DARMSTADT

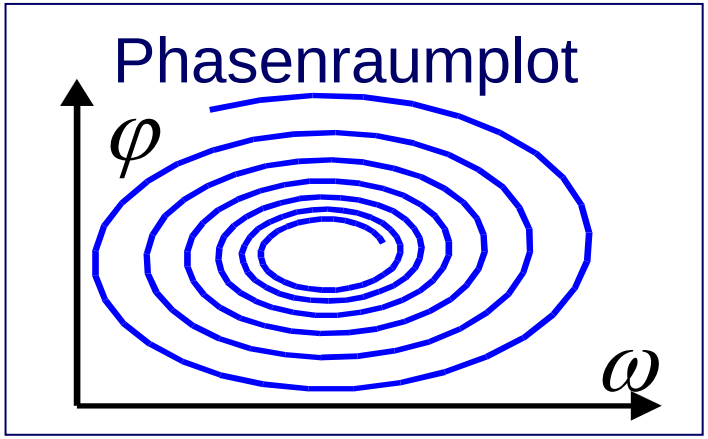
- Die Durchführung der Simulation liefert eine sehr große Menge an Daten
  - Beim Beispiel der Schiffsschaukel eine große Tabelle mit den Einträgen von t, φ und ω

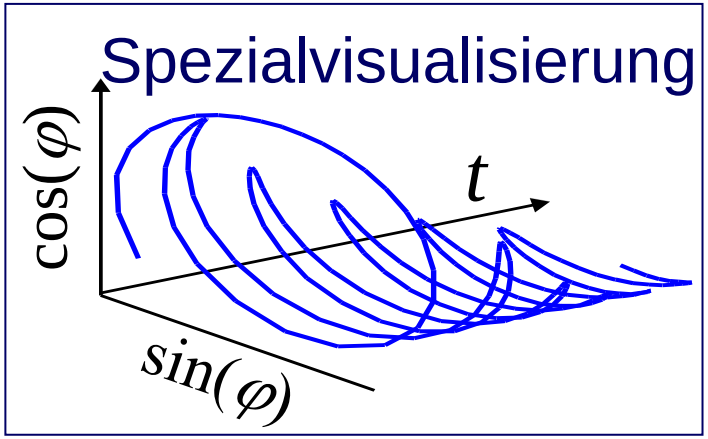
- Wie geht man mit den Ergebnissen um?

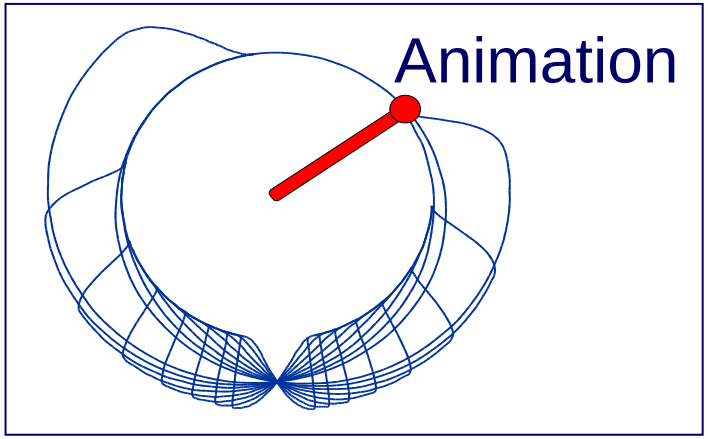
- Visualisierung und Animation

4.3.4 Visualisierungsmöglichkeiten









4.4 Begriffe

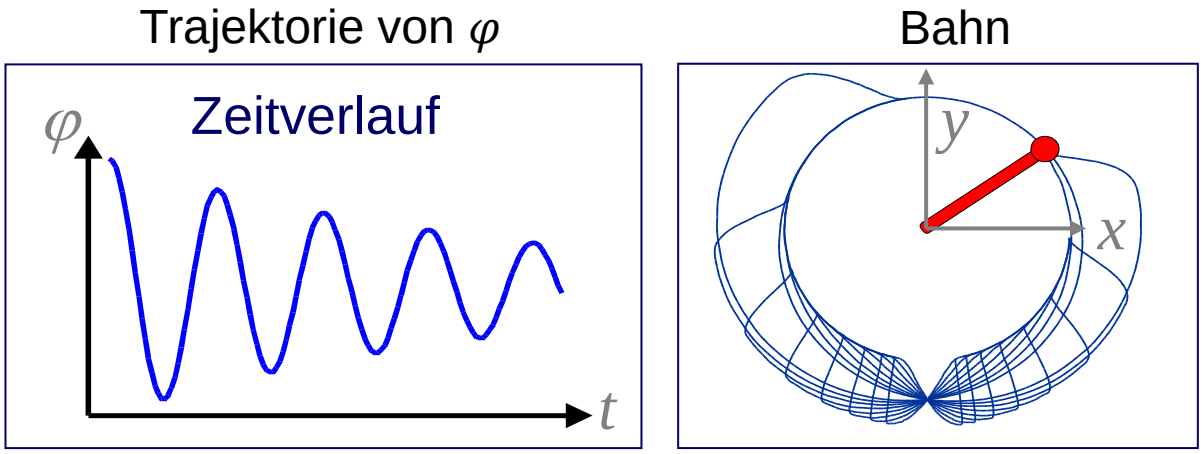
TECHNISCHE UNIVERSITÄT DARMSTADT

- Bahn (Pfad, path) = kontinuierliche, räumliche Punktfolge

- Trajektorie (trajectory) = Bahn mit „zeitlichen Beschränkungen“
  - Z.B. außer Position auch Geschwindigkeit (und Beschleunigung) an jedem Punkt der Bahn gegeben

4.4 Begriffe: Beispiel

- Beispiel der Schiffsschaukel



- Graph der Bahn enthält keine Zeitinformationen
- Beispiel: Allgemein Zustands-DGL $\dot{x}(t) = f\big(x(t)\big), \quad \text{mit } x(0) = x_0 \in \mathbb{R}^n$
- Gesucht: Trajektorie des Zustands $x(t), \quad 0 \leq t \leq t_f$

4.5 Klassifikation zeitkontinuierlicher Simulationswerkzeuge

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eine vereinfachte Möglichkeit der Implementierung bieten Simulatoren
  - Klassifikation z.B. nach „Komfort“ für den Anwender

- Bei der Implementierung der Simulation unterscheidet man zwischen vier Level

4.5 Klassifikation zeitkontinuierlicher Simulationswerkzeuge

TECHNISCHE UNIVERSITÄT DARMSTADT

Level 3
Multidisziplinäre Modellgenerierung (Modelica, VHDL-A)

Level 2
Graphische Modellierung (SIMULINK, WorkingModel, Aspen, STELLA,...)
Spezialsimulatoren (ADAMS, SIMPACK, KSIM,...)

Level 1
Simulationssprachen (ACSL, VHDL, Dare-P, Desire,...)
Simulationsframeworks (SIMULINK, C++-Klassenbibliothek)

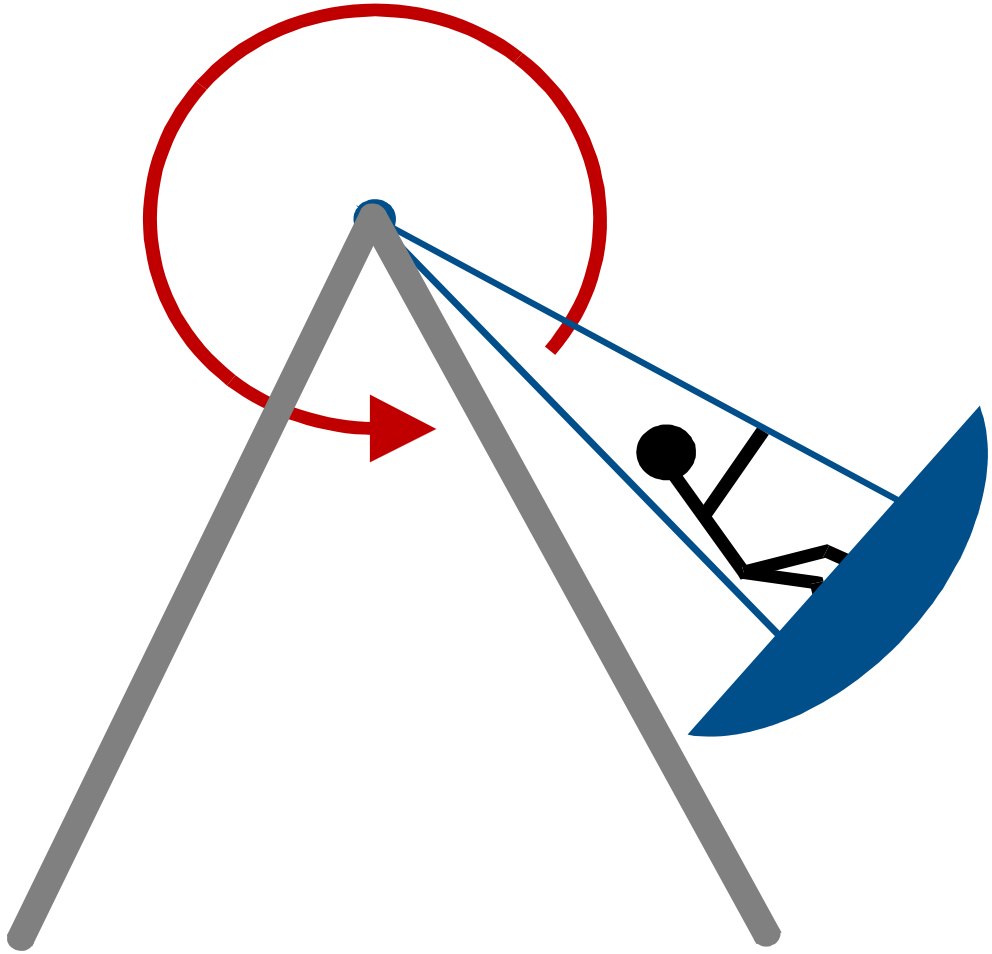
Level 0
Direkte Programmierung (FORTRAN, C, Pascal, MATLAB,...)

4.5.1 Beispiel: Schiffsschaukel

- Man betrachte wieder das Beispiel der Schiffsschaukel

$$\dot{\varphi} = \omega$$

$$\dot{\omega} = -\frac{d}{ml^2} \cdot \omega - \frac{g}{l} \sin \varphi$$

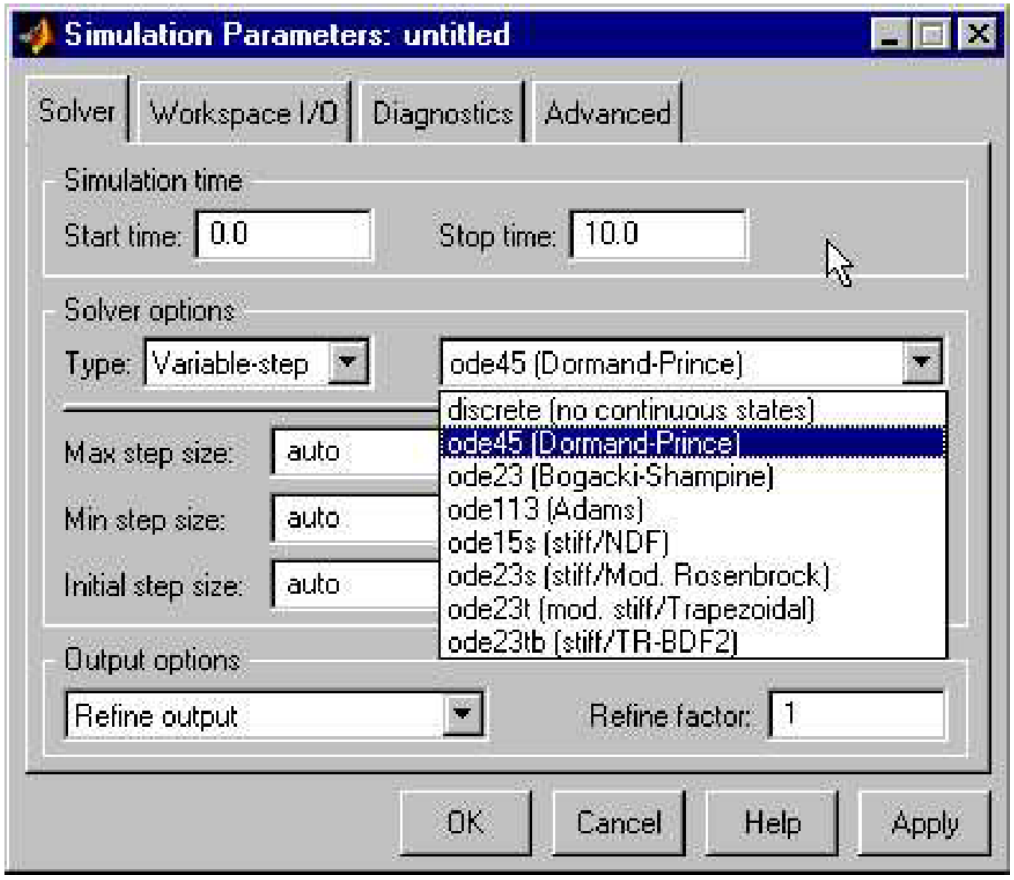


4.5.1 Level 0: Programmierung mit MATLAB

TECHNISCHE UNIVERSITÄT DARMSTADT

```
d = 100.0;                          % Reibungskonstante
m = 100.0;                          % Schaukelmasse inkl. Mensch
l = 2.5;                            % Abstand Schwerpunkt zu Drehachse
g = 10.0;                           % Erdbeschleunigung

deltat = 0.1;                       % Schrittweite
tEnd = 20.0;                        % Endzeit

t = 0.0;                            % Startzeit
phi = 1.0;                          % Startwinkel
omega = 0.0;                        % Startwinkel

while t <= tEnd
        disp(sprintf('%8.4f  %8.4f  %8.4f', t, phi, omega));   % Ausgabe der Werte
        dphi_dt = omega;                                       % Berechnung der rechten ODE-Seite
        domega_ dt = -d/(m * l^2) * omega - g/l * sin(phi);

        t = t + deltat;                                        % Neue Zeiten und Werte
        phi = phi + deltat * dphi_dt;
        omega = omega + deltat * domega_dt;
end;
```

4.5.2 Integrationsverfahren in MATLAB



- **ode1** Euler
- **ode2** Heun
- **ode3** Bogacki-Shampine
- **ode4** RK4

  ⋮
- **ode45** basierend auf einem expliziten RK (4,5)

  ⋮

4.5.2 Level 1: Differentialgleichungslöser in MATLAB

TECHNISCHE UNIVERSITÄT DARMSTADT

```
function dxdt = Schaukel (t,x);
```

```
d = 100.0;
m = 100.0;
l = 2.5;
g = 10.0;
```

Modellparameter

```
phi = x(1);
omega = x(2);
```

Variablen

```
dphi_dt = omega;
domega_dt = -d/(m * l^2) * omega - g/l * sin(phi);
```

Rechte Seite

```
dxdt = [dphi_dt;  domega_dt];
```

Ergebnis

4.5.2 Level 1: Differentialgleichungslöser in MATLAB

TECHNISCHE UNIVERSITÄT DARMSTADT

- Aufruf eines Differentialgleichungslösers (Integrators)

```
phi0 = 1.0;
omega = 0.0;
tEnd = 20.0;
```

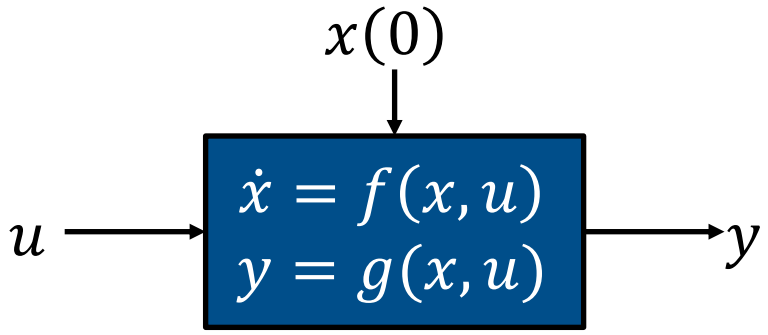
Simulationsparameter

```
[T, X] = ode45(@Schaukel, [0, tEnd], [phi0; omega0]);
```

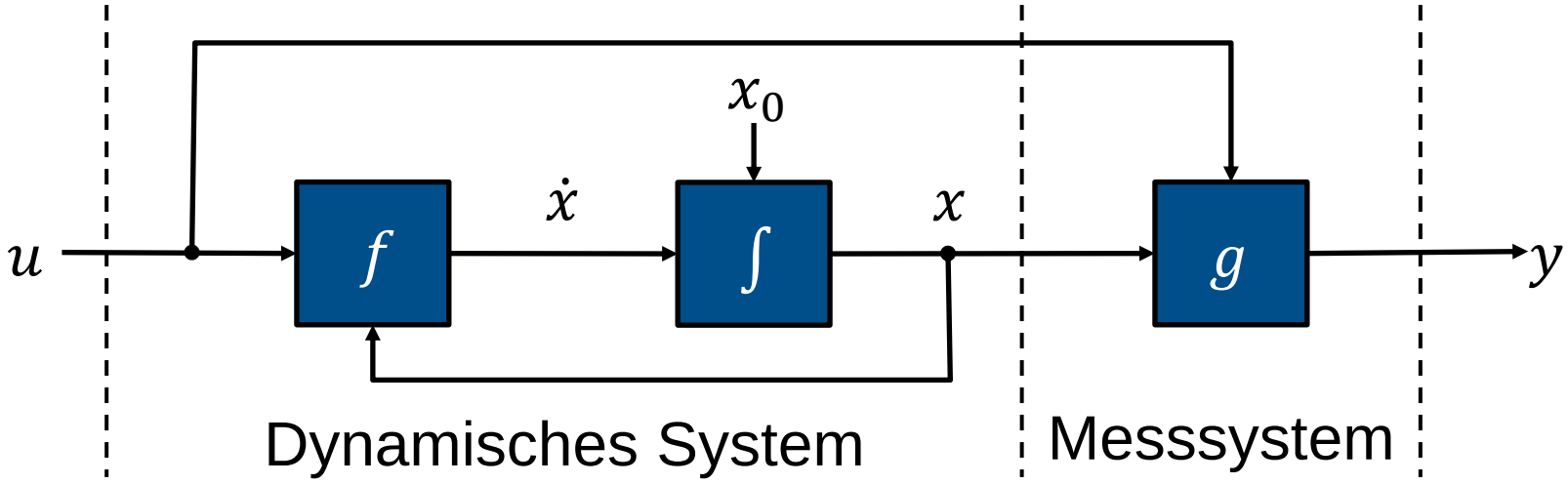
Integrationsaufruf

4.5.3 Level 2: Blockorientierte Darstellung

- Das DGL-System mit Anfangswert
  - $\dot{x}(t) = f\big(x(t), u(t)\big), \quad x(0) = x_0 \in \mathbb{R}^n$



- Äquivalente Form
  - $x(t) = x_0 + \int_0^t f\big(x(s), u(s)\big)\, ds$



4.5.3 Level 2: Blockorientierte Darstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

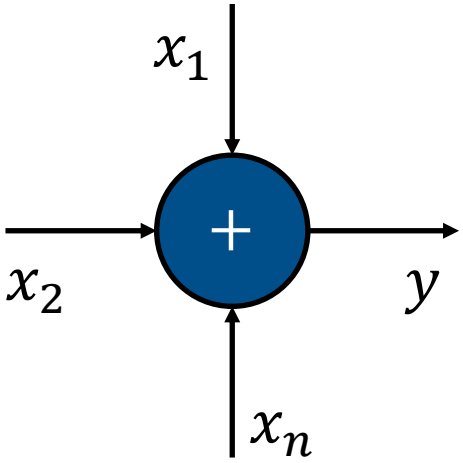
- Zustandsgrößen: 

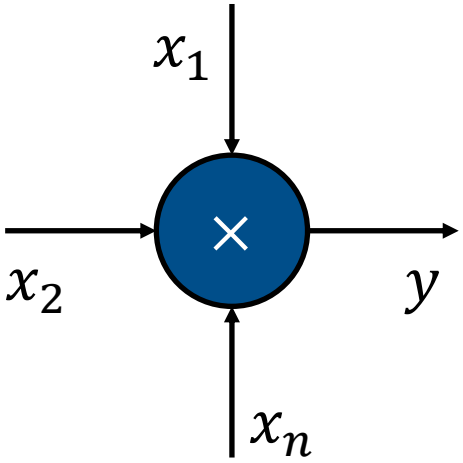
- Eingaben und Parameter: 

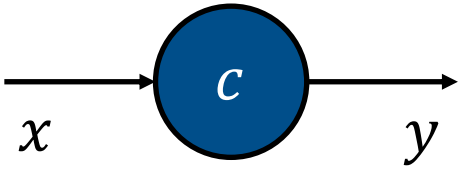
- Wirkfaktoren: $x \xrightarrow{-a} -ax$

4.5.3 Level 2: Blockorientierte Darstellung

- Funktionsblock: Ausgabe ist Funktion der Eingaben







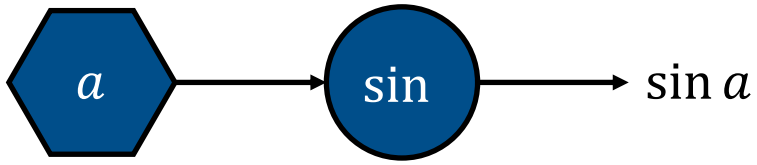
$$y = \sum x_i$$

$$y = \prod x_i$$

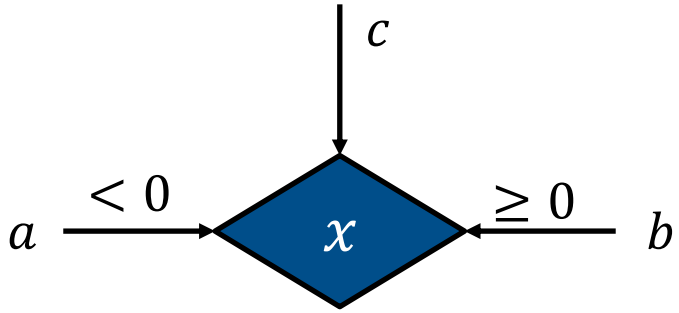
$$y = c \cdot x$$

4.5.3 Level 2: Blockorientierte Darstellung

- Funktionen



- Schaltfunktionen

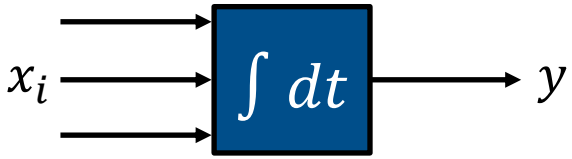


$$\text{if } c < 0 \text{ then } x = a$$
$$\text{if } c \geq 0 \text{ then } x = b$$

4.5.3 Level 2: Blockorientierte Darstellung

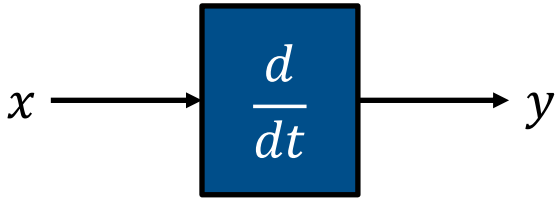
TECHNISCHE UNIVERSITÄT DARMSTADT

- Integration



$$\frac{dy}{dt} = \sum x_i$$

- Differentiation



$$y = \frac{dx}{dt}$$

4.5.3 Level 2: Blockorientierte Darstellung

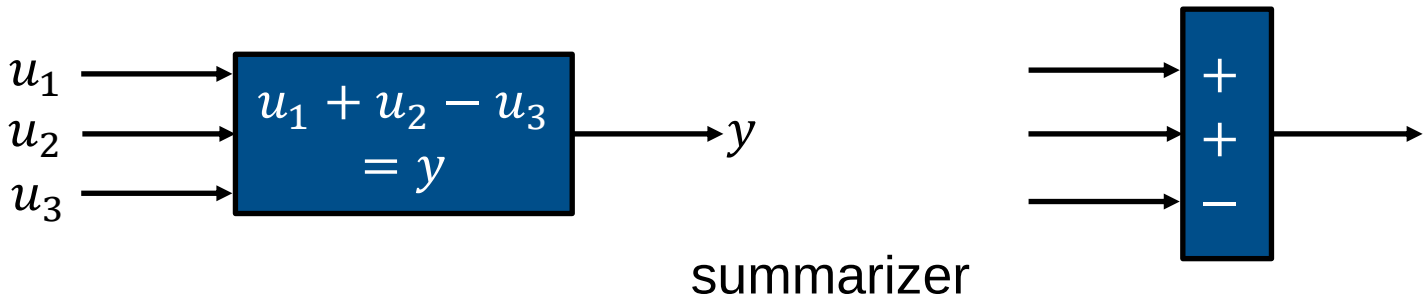
- Der allgemeine Funktionsblock sieht wie folgt aus

- Eine wichtige Funktion und die Matlab / Simulink Darstellung

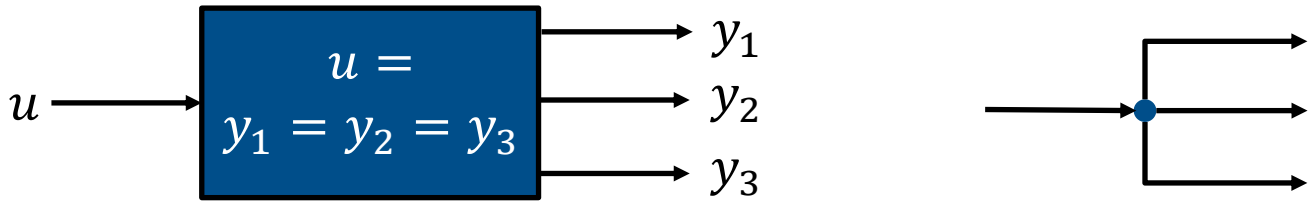
integrator

4.5.3 Level 2: Blockorientierte Darstellung

- Weitere wichtige Funktionen und die Matlab / Simulink Darstellung



summarizer



branch

4.5.3 Level 2: Blockorientierte Darstellung

- Eine weitere wichtige Funktion und die Matlab / Simulink Darstellung

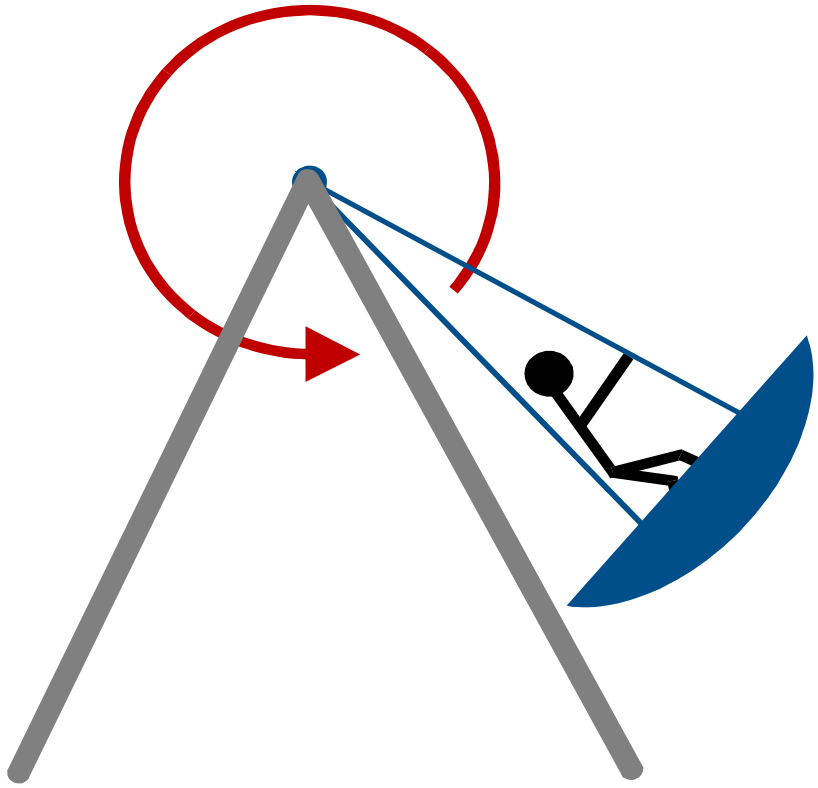
4.5.3 Level 2: Blockorientierte Simulation mit SIMULINK

TECHNISCHE UNIVERSITÄT DARMSTADT

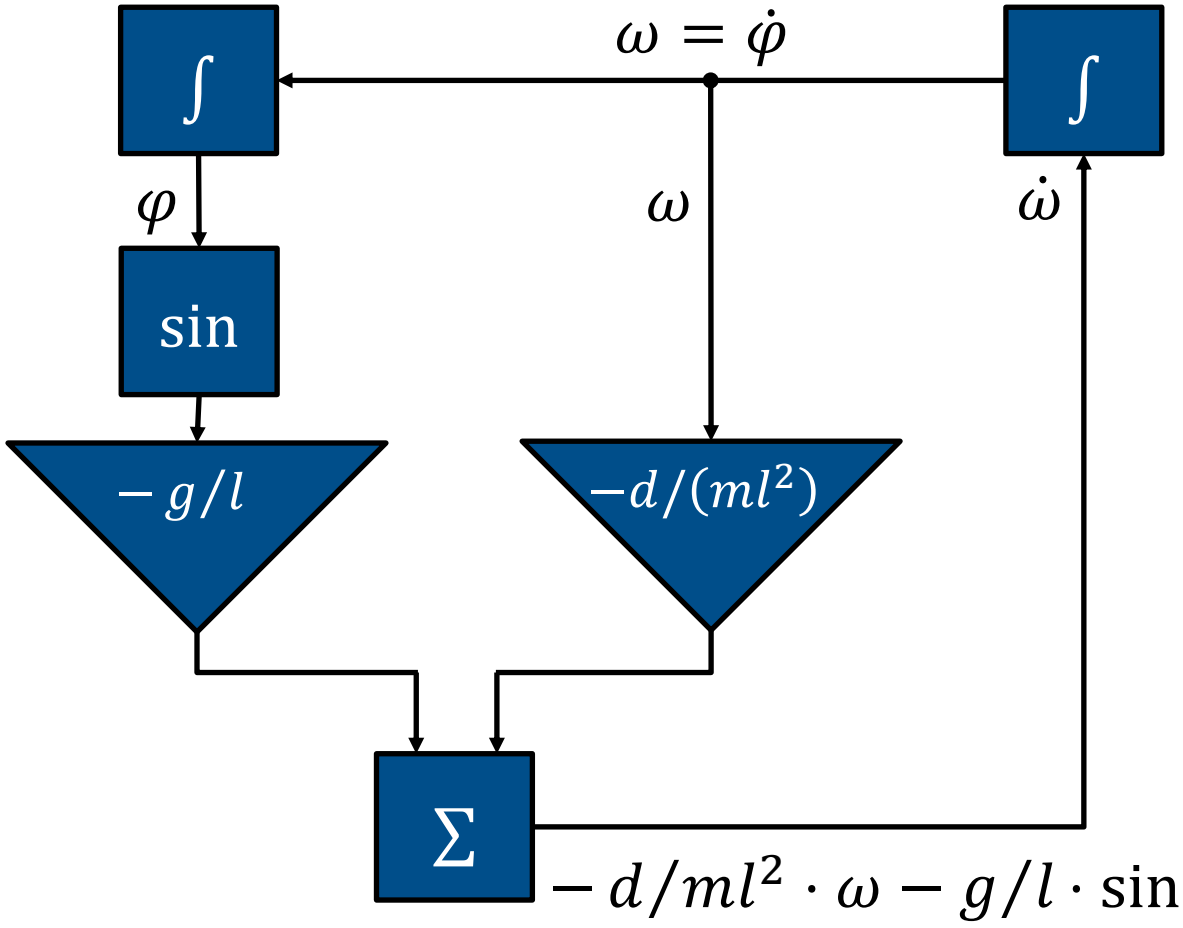
- Man betrachte erneut das Beispiel der Schiffsschaukel

$$\dot{\varphi} = \omega$$

$$\dot{\omega} = -\frac{d}{ml^2} \cdot \omega - \frac{g}{l} \sin \varphi$$

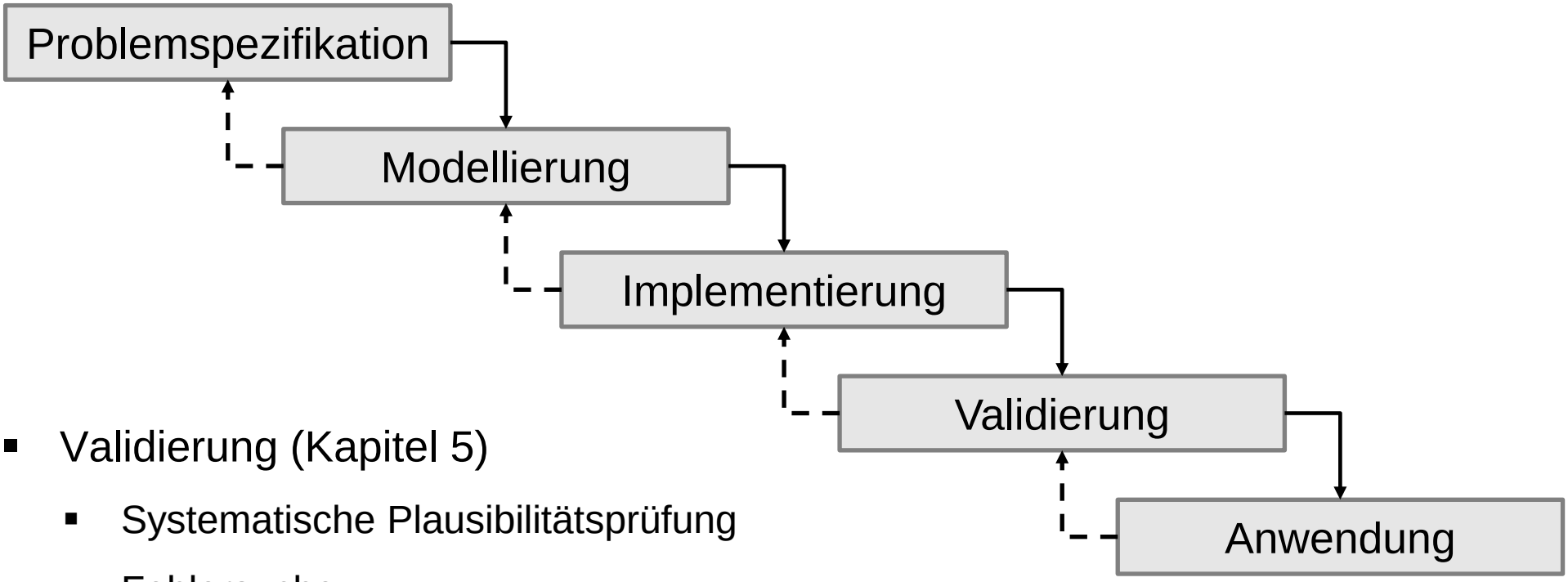


4.5.3 Level 2: Blockorientierte Simulation mit SIMULINK



$$\dot{\varphi} = \omega$$
$$\dot{\omega} = -\frac{d}{ml^2} \cdot \omega - \frac{g}{l} \sin\varphi$$

4.6 Validierung



- Validierung (Kapitel 5)
  - Systematische Plausibilitätsprüfung
  - Fehlersuche
  - Konsistenzprüfung
  - Daten- und Parameterabgleich

4.6 Validierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Modellierungsfehler
  - Vereinfachende Modellannahmen (z.B. starrer statt elastischer Körper)
  - Ungenauigkeiten in Modellparametern

- Approximationsfehler des iterativen Berechnungsverfahrens
  - Z.B. beim Euler-Verfahren proportional zur Schrittweite

4.6 Validierung

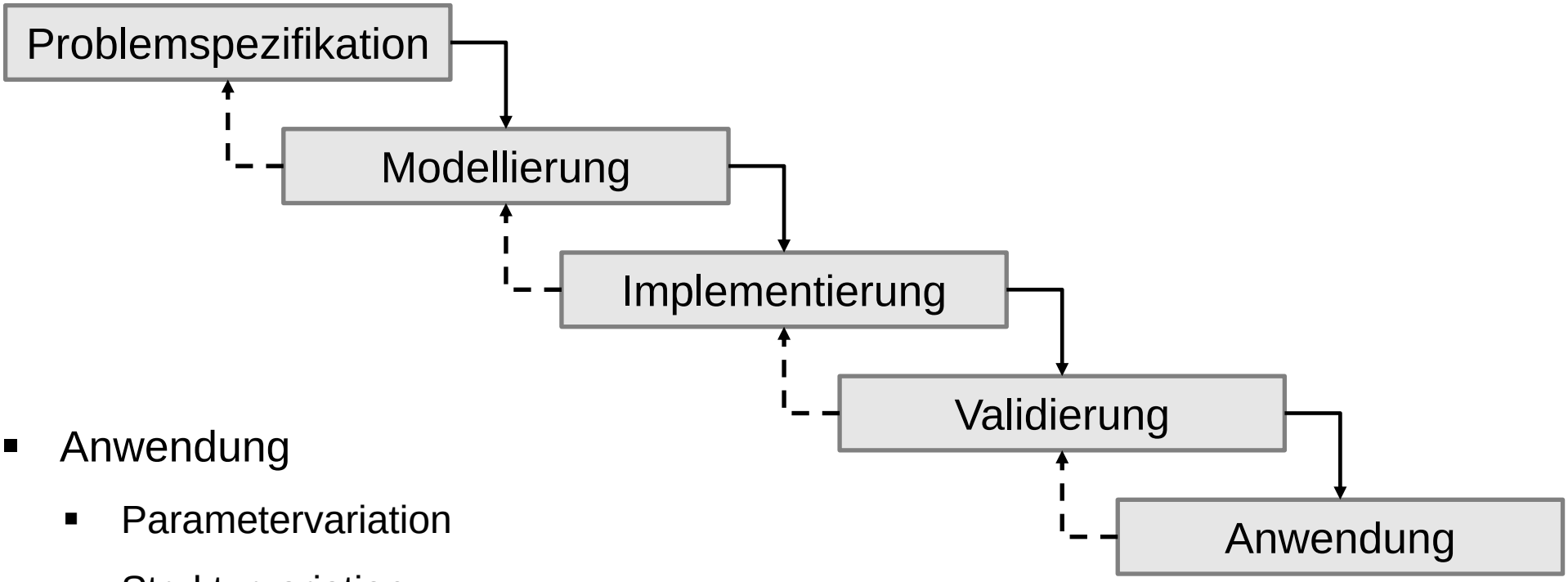
TECHNISCHE UNIVERSITÄT DARMSTADT

- Rundungsfehler
  - Ausführung des Berechnungsverfahrens auf Computern mit endlicher Zahlendarstellung

- Programmier- und Implementierungsfehler

- Notwendigkeit der Validierung!
  - Die Visualisierung und Animation spielt bei der Validierung eine wichtige Rolle, z.B. bei der Plausibilitätsüberprüfung der visualisierten Lösung von Testsimulationsläufen bei bekanntem Verhalten des realen Systems.

4.7 Anwendung



- Anwendung
  - Parametervariation
  - Strukturvariation
  - Vorhersage und Optimierung

4.7 Anwendung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nach den Schritten der Modellbildung, Implementierung und Validierung nehmen die eigentlichen Simulationsläufe in der Anwendung nur relativ wenig (Entwickler-) Zeit in Anspruch

Heutige Lernziele: Kernfragen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wie funktioniert eine Simulationstudie im Detail?
- Was bedeuten Modellbildung, Spezifikation, Implementierung, Validierung, Anwendung im Detail? Sind sie in der Lage, jeden Schritt selber zu tun?
- Wie kann man Simulationswerkzeuge klassifizieren?
- Welche „Levels“ gibt es und was unterscheidet diese?
- Wie funktionieren MATLAB und SIMULINK?
- Wie kann man ODE45 nutzen?
- Wie funktioniert die blockorientiert Darstellung?
- **Selbsttest: Können Sie diese Fragen beantworten? Wenn nicht, schnell nochmal das Video anschauen!**