

Konstantin Wrede

TU Dresden, Institut für Regelungs- und Steuerungstheorie

# Verteidigung der Studienarbeit: Modellbildung und Reglerentwurf für ein Brückenkransystem

Dresden, XX. November 2022

# Gliederung

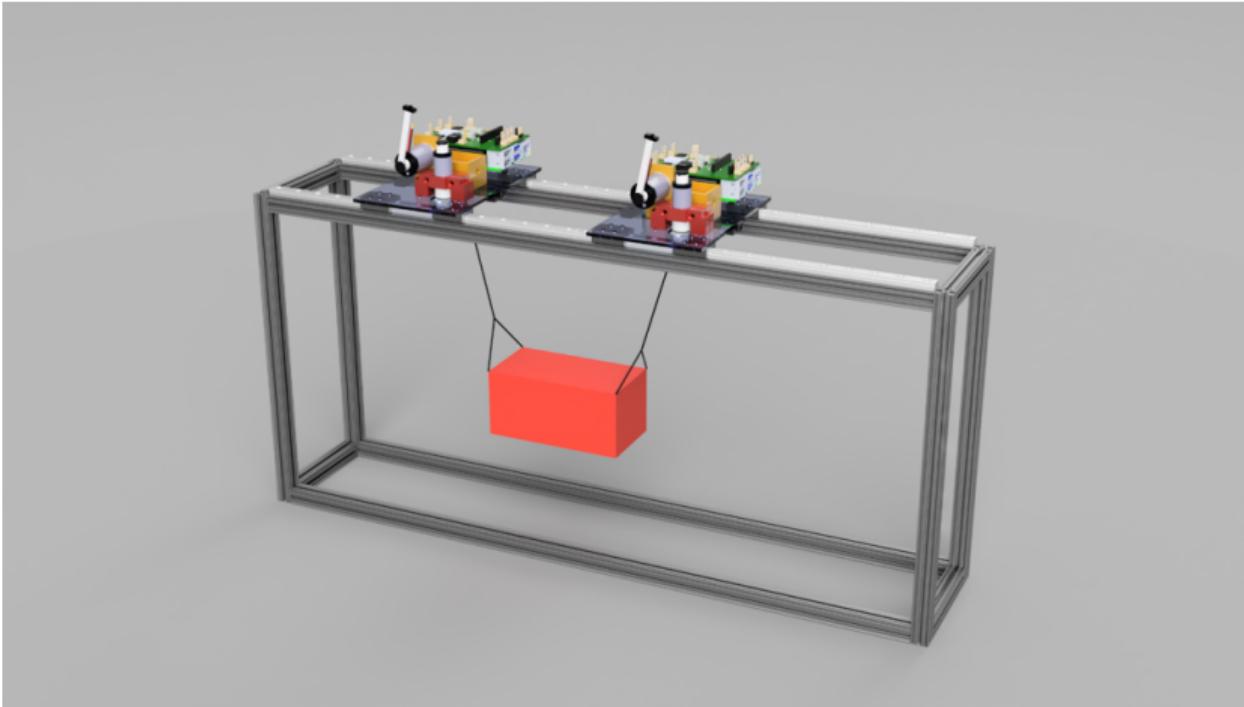
- System- und Problembeschreibung
- Analytische Modellbildung
- Flachheitsanalyse
- Steuerungs- und Regelungsentwurf

# Gliederung

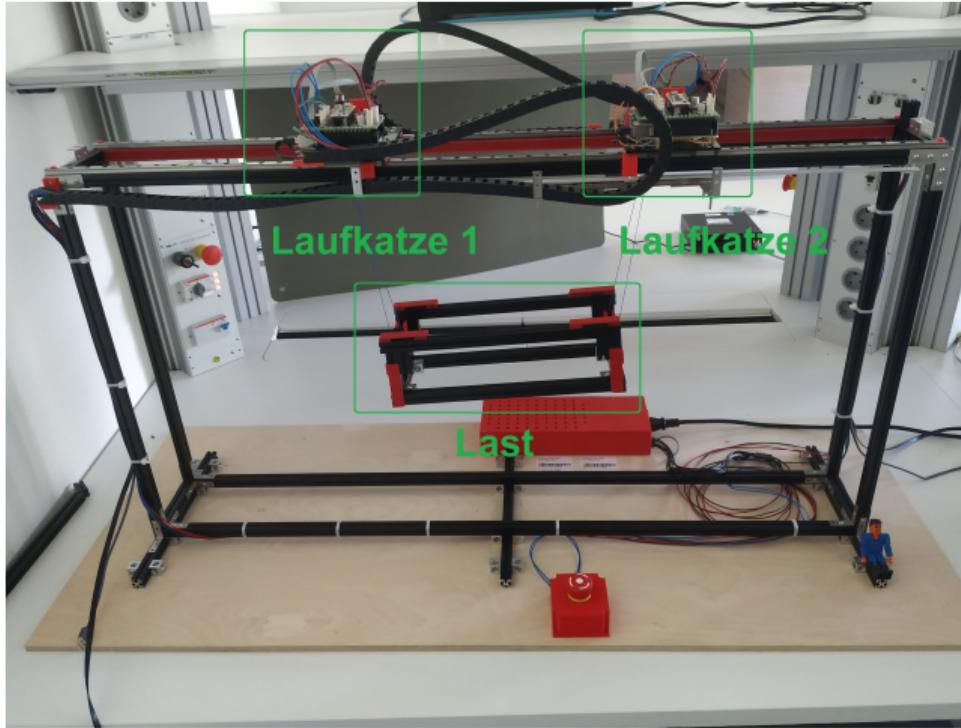
## → System- und Problembeschreibung

- Analytische Modellbildung
- Flachheitsanalyse
- Steuerungs- und Regelungsentwurf

# System- und Problembeschreibung



# System- und Problembeschreibung



# System- und Problembeschreibung

## Zielsetzung:

- Ruckarme Überführung der Last zwischen Ruhelagen in der vertikalen Aufhangungsebene

# System- und Problembeschreibung

## Zielsetzung:

- Ruckarme Überführung der Last zwischen Ruhelagen in der vertikalen Aufhängungsebene
- Zentrale Trajektorienplanung und Referenzregelstrategie unter Vorgabe von Sollposen der Last

# System- und Problembeschreibung

## Zielsetzung:

- Ruckarme Überführung der Last zwischen Ruhelagen in der vertikalen Aufhängungsebene
- Zentrale Trajektorienplanung und Referenzregelstrategie unter Vorgabe von Sollposen der Last
- Regelungsentwurf auf Basis von Modellierung des Versuchsstands als Mehrkörpersystem

# System- und Problembeschreibung

## Zielsetzung:

- Ruckarme Überführung der Last zwischen Ruhelagen in der vertikalen Aufhängungsebene
- Zentrale Trajektorienplanung und Referenzregelstrategie unter Vorgabe von Sollposen der Last
- Regelungsentwurf auf Basis von Modellierung des Versuchsstands als Mehrkörpersystem
- Perspektivisch Vergleich der zentralen Regelung mit verteilten Regelungsansätzen und Grundlage für maschinelles Lernen

# Gliederung

System- und Problembeschreibung

→ **Analytische Modellbildung**

Flachheitsanalyse

Steuerungs- und Regelungsentwurf

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last
- Last trotz Aussparungen mit homogener Masseverteilung modelliert

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last
- Last trotz Aussparungen mit homogener Masseverteilung modelliert
- Vernachlässigung dissipativer Kräfte

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last
- Last trotz Aussparungen mit homogener Masseverteilung modelliert
- Vernachlässigung dissipativer Kräfte

## Vorgehen bei der Modellierung:

- Modellierung Einzelkran mit Lagrange-Gleichungen zweiter Art (LG2)

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last
- Last trotz Aussparungen mit homogener Masseverteilung modelliert
- Vernachlässigung dissipativer Kräfte

## Vorgehen bei der Modellierung:

- Modellierung Einzelkran mit Lagrange-Gleichungen zweiter Art (LG2)
- Modellierung Doppelkran mit LG2

# Analytische Modellbildung

## Allgemeine Modellannahmen:

- Bewegung des Systems auf vertikale Ebene beschränkt
- Seile mit vernachlässigbarer Masse gegenüber Laufkatzen, Last
- Last trotz Aussparungen mit homogener Masseverteilung modelliert
- Vernachlässigung dissipativer Kräfte

## Vorgehen bei der Modellierung:

- Modellierung Einzelkran mit Lagrange-Gleichungen zweiter Art (LG2)
- Modellierung Doppelkran mit LG2
- (Modellierung Doppelkran mit Lagrange-Gleichungen erster Art)

# Lagrange-Formalismus

## Symbol:

- Konfigurationskoordinaten  $\theta = (\mathbf{q}, \mathbf{p})^T$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$
- äußere Stellkraft  $f$ , interne Reibungskraft  $D$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$
- äußere Stellkraft  $f$ , interne Reibungskraft  $D$

## Lagrange-Gleichungen zweiter Art:

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$
- äußere Stellkraft  $f$ , interne Reibungskraft  $D$

## Lagrange-Gleichungen zweiter Art:

- $\theta$  sind unabhängig (ohne Zwangsbedingung verkoppelt)

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$
- äußere Stellkraft  $f$ , interne Reibungskraft  $D$

## Lagrange-Gleichungen zweiter Art:

- $\theta$  sind unabhängig (ohne Zwangsbedingung verkoppelt)
- Bewegungsgleichungen:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} \right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = Q_i, \quad i = 1, \dots, n$$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- direkt aktivierte Koordinaten  $q$ , nicht direkt aktivierte Koordinaten  $p$
- kinetische Energie  $T(\theta, \dot{\theta})$ , potentielle Energie  $V(\theta)$
- Lagrange-Funktion  $\mathcal{L}(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$
- verallgemeinerte Kraft  $Q = f - D$
- äußere Stellkraft  $f$ , interne Reibungskraft  $D$

## Lagrange-Gleichungen zweiter Art:

- $\theta$  sind unabhängig (ohne Zwangsbedingung verkoppelt)
- Bewegungsgleichungen:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} \right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = Q_i, \quad i = 1, \dots, n$$

- Woher  $Q_i$ ?

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- verallgemeinerte Kraft  $Q = f - D$

# Lagrange-Formalismus

## Symbole:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- verallgemeinerte Kraft  $Q = f - D$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $r_k$ , Stellkraft  $F_k$  entlang  $r_k$

# Lagrange-Formalismus

## Symbole:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- verallgemeinerte Kraft  $Q = f - D$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $r_k$ , Stellkraft  $F_k$  entlang  $r_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta r_k$  und Koordinate  $\theta_i$

# Lagrange-Formalismus

## Symbole:

- Konfigurationskoordinaten  $\theta = (q, p)^T$
- verallgemeinerte Kraft  $Q = f - D$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $r_k$ , Stellkraft  $F_k$  entlang  $r_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta r_k$  und Koordinate  $\theta_i$

## Prinzip der virtuellen Arbeit zur Bestimmung der $Q_i$ :

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (\mathbf{q}, \mathbf{p})^T$
- verallgemeinerte Kraft  $\mathbf{Q} = \mathbf{f} - \mathbf{D}$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $\mathbf{r}_k$ , Stellkraft  $\mathbf{F}_k$  entlang  $\mathbf{r}_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta \mathbf{r}_k$  und Koordinate  $\theta_i$

## Prinzip der virtuellen Arbeit zur Bestimmung der $Q_i$ :

- $\delta W = \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_1} \delta \theta_1 + \dots + \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_n} \delta \theta_n$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (\mathbf{q}, \mathbf{p})^T$
- verallgemeinerte Kraft  $\mathbf{Q} = \mathbf{f} - \mathbf{D}$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $\mathbf{r}_k$ , Stellkraft  $\mathbf{F}_k$  entlang  $\mathbf{r}_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta \mathbf{r}_k$  und Koordinate  $\theta_i$

## Prinzip der virtuellen Arbeit zur Bestimmung der $Q_i$ :

- $\delta W = \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_1} \delta \theta_1 + \dots + \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_n} \delta \theta_n$
- $\delta \mathbf{r}_k = \sum_{i=1}^n \frac{\partial \mathbf{r}_k}{\partial \theta_i} \delta \theta_i$

# Lagrange-Formalismus

## Symbolen:

- Konfigurationskoordinaten  $\theta = (\mathbf{q}, \mathbf{p})^T$
- verallgemeinerte Kraft  $\mathbf{Q} = \mathbf{f} - \mathbf{D}$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $\mathbf{r}_k$ , Stellkraft  $\mathbf{F}_k$  entlang  $\mathbf{r}_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta \mathbf{r}_k$  und Koordinate  $\theta_i$

## Prinzip der virtuellen Arbeit zur Bestimmung der $Q_i$ :

- $\delta W = \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_1} \delta \theta_1 + \dots + \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_n} \delta \theta_n$
- $\delta \mathbf{r}_k = \sum_{i=1}^n \frac{\partial \mathbf{r}_k}{\partial \theta_i} \delta \theta_i$
- $\delta W = \sum_{k=1}^l \delta \mathbf{r}_k^T \mathbf{F}_k = Q_1 \delta \theta_1 + \dots + Q_n \delta \theta_n$

# Lagrange-Formalismus

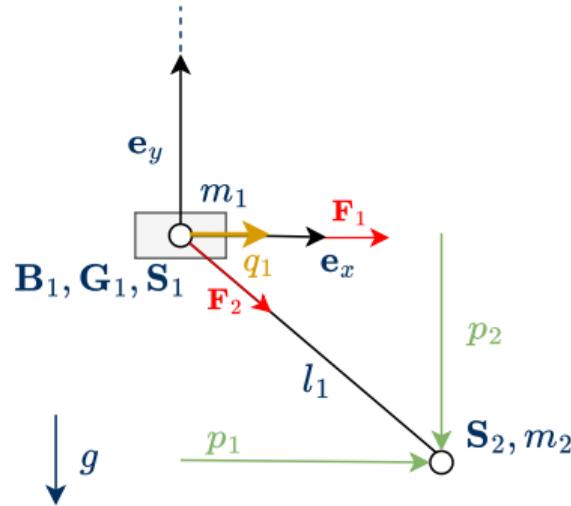
## Symbolen:

- Konfigurationskoordinaten  $\theta = (\mathbf{q}, \mathbf{p})^T$
- verallgemeinerte Kraft  $\mathbf{Q} = \mathbf{f} - \mathbf{D}$
- Richtungsvektor zu  $k$ -tem massebehaftetem Partikel  $\mathbf{r}_k$ , Stellkraft  $\mathbf{F}_k$  entlang  $\mathbf{r}_k$
- virtuelle Arbeit  $\delta W$ , virtuelle Verschiebung von Partikel  $\delta \mathbf{r}_k$  und Koordinate  $\theta_i$

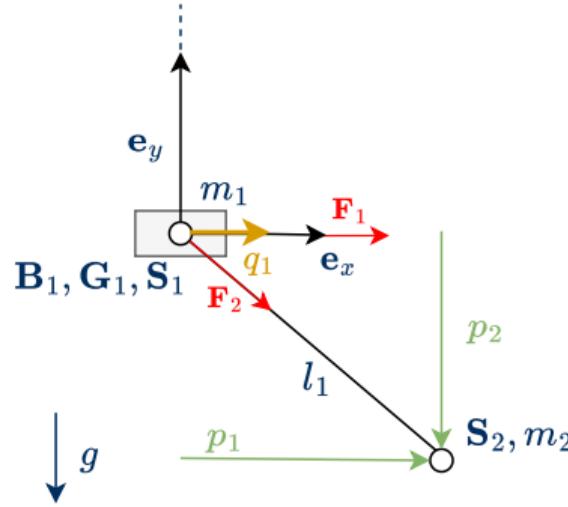
## Prinzip der virtuellen Arbeit zur Bestimmung der $Q_i$ :

- $\delta W = \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_1} \delta \theta_1 + \dots + \sum_{k=1}^l \mathbf{F}_k \cdot \frac{\partial \mathbf{r}_k}{\partial \theta_n} \delta \theta_n$
  - $\delta \mathbf{r}_k = \sum_{i=1}^n \frac{\partial \mathbf{r}_k}{\partial \theta_i} \delta \theta_i$
  - $\delta W = \sum_{k=1}^l \delta \mathbf{r}_k^T \mathbf{F}_k = Q_1 \delta \theta_1 + \dots + Q_n \delta \theta_n$
- $$\rightarrow Q_i = \sum_{k=1}^l \left( \frac{\partial \mathbf{r}_k}{\partial \theta_i} \right)^T \mathbf{F}_k = \frac{\partial \delta W}{\partial \delta \theta_i}, \quad i = 1, \dots, n$$

# Analytisches Modell Einzelkran

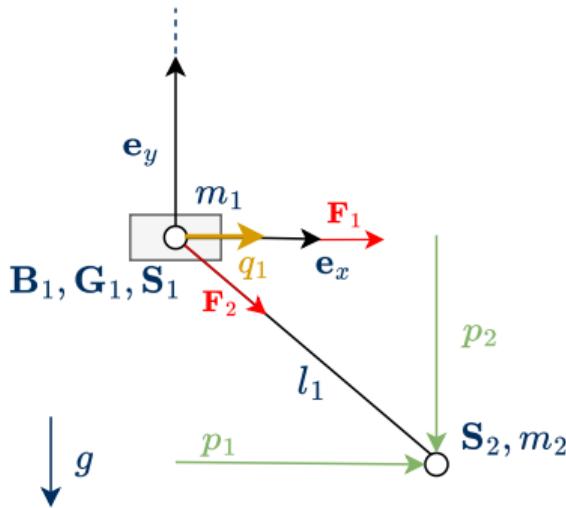


# Analytisches Modell Einzelkran



- Massen bei  $\mathbf{S}_1 = (q_1, 0)^T, \mathbf{S}_2 = (p_1, p_2)^T$
- variable Seillänge  $l_1 = \sqrt{(p_1 - q_1)^2 + p_2^2}$

# Analytisches Modell Einzelkran

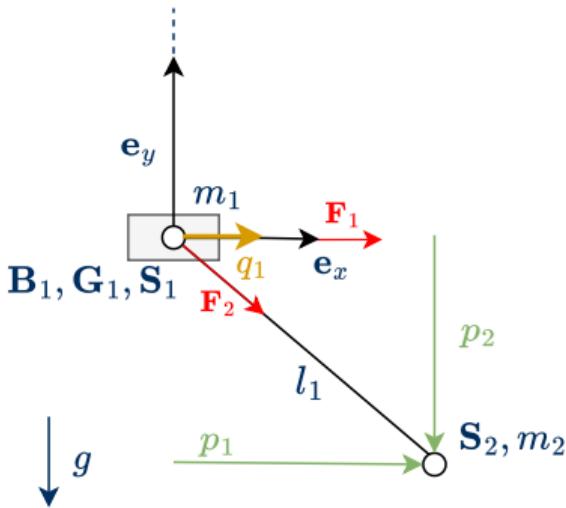


## Energien:

- $T = \frac{m_1}{2} \dot{\mathbf{S}}_1^T \dot{\mathbf{S}}_1 + \frac{m_2}{2} \dot{\mathbf{S}}_2^T \dot{\mathbf{S}}_2 = \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2 \dot{p}_1^2}{2} + \frac{m_2 \dot{p}_2^2}{2}$
- $V = m_2 g \mathbf{S}_2^T \mathbf{e}_y = m_2 g p_2$

- Massen bei  $\mathbf{S}_1 = (q_1, 0)^T, \mathbf{S}_2 = (p_1, p_2)^T$
- variable Seillänge  $l_1 = \sqrt{(p_1 - q_1)^2 + p_2^2}$

# Analytisches Modell Einzelkran



## Energien:

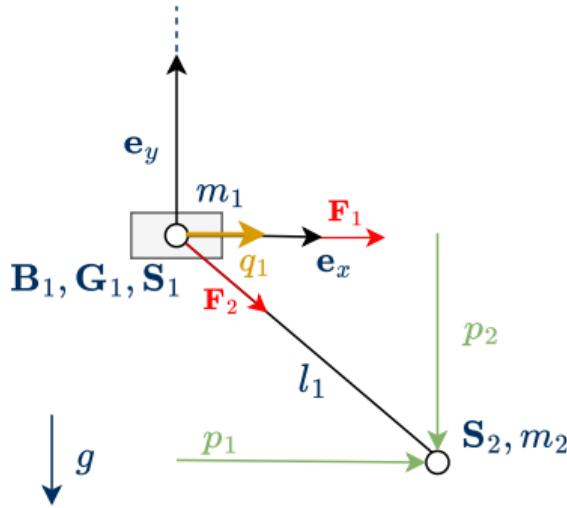
- $T = \frac{m_1}{2} \dot{\mathbf{S}}_1^T \dot{\mathbf{S}}_1 + \frac{m_2}{2} \dot{\mathbf{S}}_2^T \dot{\mathbf{S}}_2 = \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2 \dot{p}_1^2}{2} + \frac{m_2 \dot{p}_2^2}{2}$
- $V = m_2 g \mathbf{S}_2^T \mathbf{e}_y = m_2 g p_2$

## Verallgemeinerte Kraft aus virtueller Arbeit:

- $\mathbf{F}_1 = (\tau_1, 0)^T, \mathbf{F}_2 = \left( \frac{\tau_2(p_1 - q_1)}{l_1}, \frac{p_2 \tau_2}{l_1} \right)^T$   
 $\rightarrow \mathbf{Q} = \left( \frac{\tau_2(p_1 - q_1)}{l_1}, \frac{p_2 \tau_2}{l_1}, \tau_1 - \frac{\tau_2(p_1 - q_1)}{l_1} \right)^T$

- Massen bei  $\mathbf{S}_1 = (q_1, 0)^T, \mathbf{S}_2 = (p_1, p_2)^T$
- variable Seillänge  $l_1 = \sqrt{(p_1 - q_1)^2 + p_2^2}$

# Analytisches Modell Einzelkran



Systemgleichungen aus LG2:

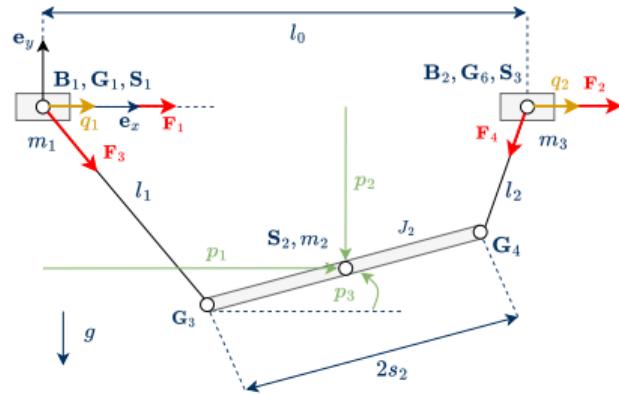
$$m_2 \ddot{p}_1 - \frac{\tau_2 (p_1 - q_1)}{l_1} = 0$$

$$gm_2 + m_2 \ddot{p}_2 - \frac{p_2 \tau_2}{l_1} = 0$$

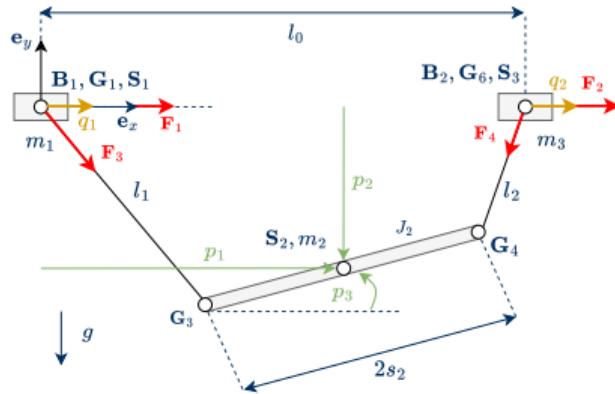
$$m_1 \ddot{q}_1 - \tau_1 + \frac{\tau_2 (p_1 - q_1)}{l_1} = 0$$

- Massen bei  $\mathbf{S}_1 = (q_1, 0)^T$ ,  $\mathbf{S}_2 = (p_1, p_2)^T$
- variable Seillänge  $l_1 = \sqrt{(p_1 - q_1)^2 + p_2^2}$

# Analytisches Modell Doppelkran



# Analytisches Modell Doppelkran

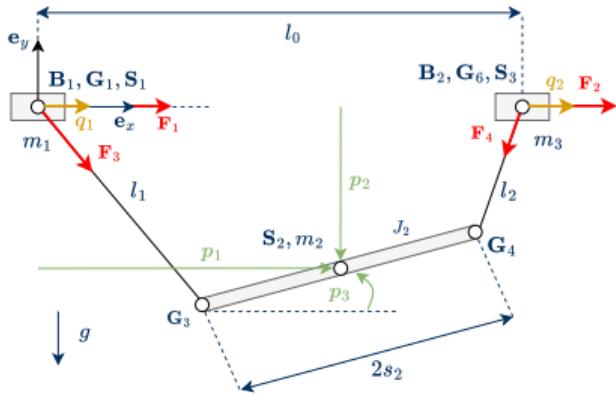


**variable Seillängen:**

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

# Analytisches Modell Doppelkran



## Energien:

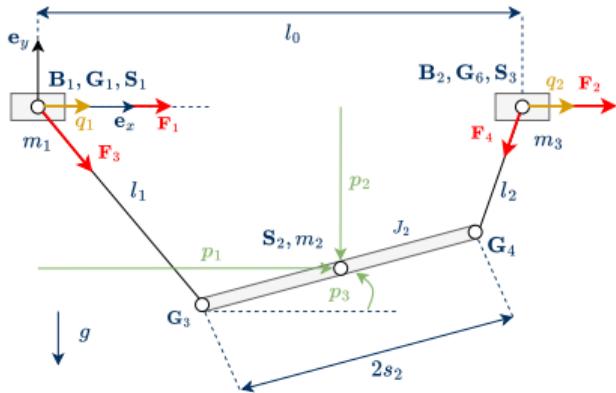
- $T = \frac{J_2 \dot{p}_3^2}{2} + \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2 \dot{p}_1^2}{2} + \frac{m_2 \dot{p}_2^2}{2} + \frac{m_3 \dot{q}_2^2}{2}$
- $V = m_2 g p_2$

## variable Seillängen:

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

# Analytisches Modell Doppelkran



**variable Seillängen:**

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

**Energien:**

- $T = \frac{J_2 \dot{p}_3^2}{2} + \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2 \dot{p}_1^2}{2} + \frac{m_2 \dot{p}_2^2}{2} + \frac{m_3 \dot{q}_2^2}{2}$
- $V = m_2 g p_2$

**Stellkräfte entlang Massepartikel:**

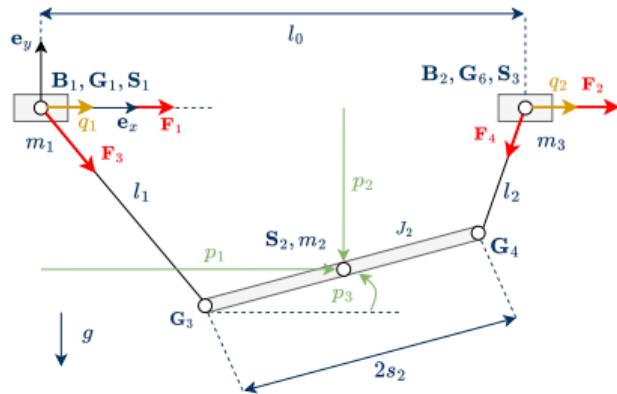
$$\mathbf{F}_1 = (\tau_1, 0)^T,$$

$$\mathbf{F}_2 = (\tau_2, 0)^T,$$

$$\mathbf{F}_3 = \begin{pmatrix} \frac{\tau_3(p_1 - q_1 - s_2 \cos(p_3))}{l_1} \\ \frac{\tau_3(p_2 - s_2 \sin(p_3))}{l_1} \end{pmatrix},$$

$$\mathbf{F}_4 = \begin{pmatrix} \frac{\tau_4(-l_0 + p_1 - q_2 + s_2 \cos(p_3))}{l_2} \\ \frac{\tau_4(p_2 + s_2 \sin(p_3))}{l_2} \end{pmatrix}$$

# Analytisches Modell Doppelkran



Verallgemeinerte Kraft aus virtueller Arbeit:

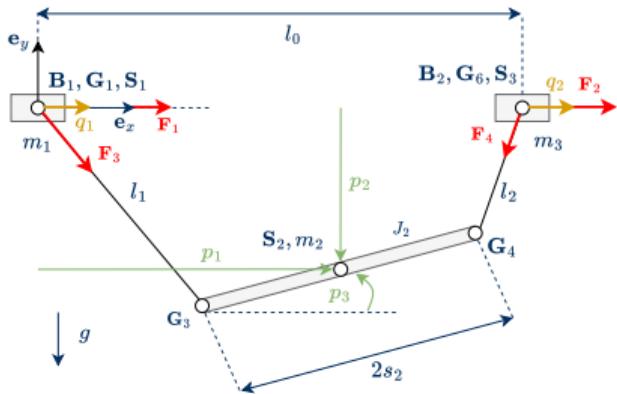
$$Q = \dots$$

variable Seillängen:

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

# Analytisches Modell Doppelkran



**variable Seillängen:**

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

**Verallgemeinerte Kraft aus virtueller Arbeit:**

$$Q = \dots$$

**Systemgleichungen aus LG2:**

$$m_2 \ddot{p}_1 - \frac{\tau_4 (-l_0 + p_1 - q_2 + s_2 \cos p_3)}{l_2} - \frac{\tau_3 (p_1 - q_1 - s_2 \cos p_3)}{l_1} = 0$$

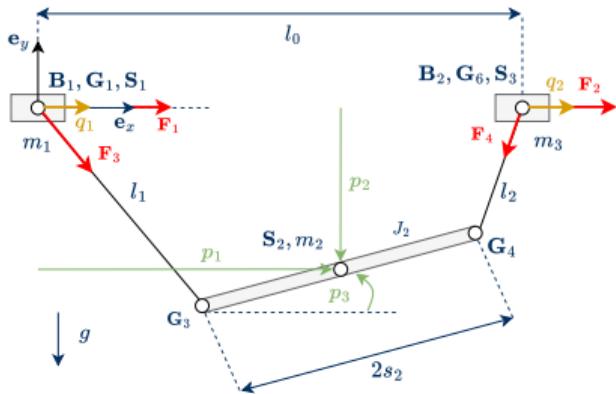
$$g m_2 + m_2 \ddot{p}_2 - \frac{\tau_4 (p_2 + s_2 \sin p_3)}{l_2} - \frac{\tau_3 (p_2 s_2 \sin p_3)}{l_1} = 0$$

$$\begin{aligned} J_2 \ddot{p}_3 - \frac{s_2 \tau_4 (p_2 + s_2 \sin p_3) \cos p_3 + s_2 \tau_4 (p_1 - q_2 + s_2 \cos p_3 - l_0)}{l_2} \\ + \frac{s_2 \tau_3 (p_2 - s_2 \sin p_3) \cos p_3}{l_1} - \frac{s_2 \tau_3 (p_1 - q_1 - s_2 \cos p_3) \sin p_3}{l_1} = 0 \end{aligned}$$

$$m_1 \ddot{q}_1 - \tau_1 + \frac{\tau_3 (p_1 - q_1 - s_2 \cos p_3)}{l_1} = 0$$

$$m_3 \ddot{q}_2 - \tau_2 + \frac{\tau_4 (-l_0 + p_1 - q_2 + s_2 \cos p_3)}{l_2} = 0$$

# Analytisches Modell Doppelkran



**variable Seillängen:**

$$l_1 = \sqrt{(p_2 - s_2 \sin(p_3))^2 + (p_1 - q_1 - s_2 \cos(p_3))^2}$$

$$l_2 = \sqrt{(p_2 + s_2 \sin(p_3))^2 + (-l_0 + p_1 - q_2 + s_2 \cos(p_3))^2}$$

**Eingangsaffines Zustandsraummodell:**

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\tau \text{ mit}$$

$$\mathbf{x} = (p_1, p_2, p_3, q_1, q_2, \dot{p}_1, \dot{p}_2, \dot{p}_3, \dot{q}_1, \dot{q}_2)^T,$$

$$\mathbf{f}(\mathbf{x}) = (\dot{p}_1, \dot{p}_2, \dot{p}_3, \dot{q}_1, \dot{q}_2, 0, -g, 0, 0, 0)^T,$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \\ * & 0 & * & 0 \\ 0 & * & 0 & * \end{pmatrix} \quad \text{wobei } * \neq 0$$

# Gliederung

- System- und Problembeschreibung
- Analytische Modellbildung
- **Flachheitsanalyse**
- Steuerungs- und Regelungsentwurf

# Flachheitsanalyse

## Differenzielle Flachheit

Ein System der Form  $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u})$  mit  $\mathbf{F}, \mathbf{x} \in \mathbb{R}^n$  und  $\mathbf{u} \in \mathbb{R}^m$  heißt (differenziell) flach, falls ein  $m$ -Tupel  $y := (y_1, \dots, y_m)^T$  sowie glatte Funktionen  $\Psi, \theta$  existieren, so dass

$$\begin{aligned}\mathbf{x} &= \Psi(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(n_x)}) \text{ mit } n_x < \infty \text{ und} \\ \mathbf{u} &= \theta(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(n_u)}) \text{ mit } n_u < \infty \text{ gilt.}\end{aligned}$$

# Flachheitsanalyse

## Differenzielle Flachheit

Ein System der Form  $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u})$  mit  $\mathbf{F}, \mathbf{x} \in \mathbb{R}^n$  und  $\mathbf{u} \in \mathbb{R}^m$  heißt (differenziell) flach, falls ein  $m$ -Tupel  $\mathbf{y} := (y_1, \dots, y_m)^T$  sowie glatte Funktionen  $\Psi, \theta$  existieren, so dass

$$\begin{aligned}\mathbf{x} &= \Psi(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(n_x)}) \text{ mit } n_x < \infty \text{ und} \\ \mathbf{u} &= \theta(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(n_u)}) \text{ mit } n_u < \infty \text{ gilt.}\end{aligned}$$

## Erläuterungen:

- Systemzustand  $\mathbf{x}$ , Systemeingang  $\mathbf{u}$
- **flacher Ausgang  $\mathbf{y}$**

→ Parametrisierung aller Systemgrößen durch  $\mathbf{y}$  und endlich viele Ableitungen ohne Lösung von DGL/Integration möglich

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad \text{mit} \quad \mathbf{x}, \mathbf{f} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{g} \in \mathbb{R}^{n \times m}$$

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System  
 $\dot{x} = f(x) + g(x)u \quad \text{mit} \quad x, f \in \mathbb{R}^n, u \in \mathbb{R}^m, g \in \mathbb{R}^{n \times m}$
- Elimination einer Komponente von  $u$  durch jeweils eine Systemgleichung

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System  
 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$  mit  $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{g} \in \mathbb{R}^{n \times m}$
- Elimination einer Komponente von  $\mathbf{u}$  durch jeweils eine Systemgleichung  
→ autonomes System aus  $p := n - m$  Gleichungen

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System  
 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$  mit  $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{g} \in \mathbb{R}^{n \times m}$
- Elimination einer Komponente von  $\mathbf{u}$  durch jeweils eine Systemgleichung  
→ autonomes System aus  $p := n - m$  Gleichungen  
→ Elimination von  $p$  Zustandskomponenten und aller übrigen Systemgleichungen

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System  
 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$  mit  $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{g} \in \mathbb{R}^{n \times m}$
- Elimination einer Komponente von  $\mathbf{u}$  durch jeweils eine Systemgleichung
  - autonomes System aus  $p := n - m$  Gleichungen
  - Elimination von  $p$  Zustandskomponenten und aller übrigen Systemgleichungen
  - übrig bleiben  $n - p = m$  Zustandskomponenten als **flacher Ausgang**  $\mathbf{y}$

# Flachheitsanalyse von MIMO-Systemen

## Prinzipielles Vorgehen:

- Eingangsaffines nichtlineares MIMO-System  
 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$  mit  $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{g} \in \mathbb{R}^{n \times m}$
- Elimination einer Komponente von  $\mathbf{u}$  durch jeweils eine Systemgleichung
  - autonomes System aus  $p := n - m$  Gleichungen
  - Elimination von  $p$  Zustandskomponenten und aller übrigen Systemgleichungen
  - übrig bleiben  $n - p = m$  Zustandskomponenten als **flacher Ausgang**  $\mathbf{y}$
- Wonach Systemgrößen und -gleichungen für Elimination auswählen?

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Jacobi-Matrix  $J_i$  der  $i$  bisher nicht eliminierte Systemgleichungen bezüglich nicht eliminierte Eingangskomponenten  $u_{m-(n-i)}$  (später analog für  $x_{i+m}$ ):

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Jacobi-Matrix  $\mathbf{J}_i$  der  $i$  bisher nicht eliminierte Systemgleichungen bezüglich nicht eliminierte Eingangskomponenten  $\mathbf{u}_{m-(n-i)}$  (später analog für  $\mathbf{x}_{i+m}$ ):

$$\mathbf{J}_i = \begin{pmatrix} * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \\ * & \cdots & * & \varepsilon & * & \cdots & * \\ * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \end{pmatrix}$$

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Jacobi-Matrix  $\mathbf{J}_i$  der  $i$  bisher nicht eliminierte Systemgleichungen bezüglich nicht eliminierte Eingangskomponenten  $\mathbf{u}_{m-(n-i)}$  (später analog für  $\mathbf{x}_{i+m}$ ):

$$\mathbf{J}_i = \begin{pmatrix} * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \\ * & \cdots & * & \varepsilon & * & \cdots & * \\ * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \end{pmatrix}$$

→ Identifikation einer Spalte mit nur einem nicht-Null-Eintrag

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Jacobi-Matrix  $\mathbf{J}_i$  der  $i$  bisher nicht eliminierte Systemgleichungen bezüglich nicht eliminierte Eingangskomponenten  $\mathbf{u}_{m-(n-i)}$  (später analog für  $\mathbf{x}_{i+m}$ ):

$$\mathbf{J}_i = \begin{pmatrix} * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \\ * & \cdots & * & \varepsilon & * & \cdots & * \\ * & \cdots & * & 0 & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & 0 & * & \cdots & * \end{pmatrix}$$

- Identifikation einer Spalte mit nur einem nicht-Null-Eintrag
- Elimination der mit Spalte korrespondierenden Eingangskomponente und Zeile korrespondierender Gleichung

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Wie vorgehen, wenn keine solche Zeile auffindbar?

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Wie vorgehen, wenn keine solche Zeile auffindbar?  
→ Transformation der Systemgleichungen mit Matrix  $\mathbf{T}_i$ :

$$\mathbf{T}_i \mathbf{J}_i = \begin{pmatrix} \mathbf{I}_{m-(n-i)} \\ \mathbf{0}_{(n-m) \times (m-(n-i))} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ \hline 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

# Flachheitsanalyse von MIMO-Systemen

## Elimination von Systemgrößen- und Gleichungen:

- Wie vorgehen, wenn keine solche Zeile auffindbar?  
→ Transformation der Systemgleichungen mit Matrix  $\mathbf{T}_i$ :

$$\mathbf{T}_i \mathbf{J}_i = \begin{pmatrix} \mathbf{I}_{m-(n-i)} \\ \mathbf{0}_{(n-m) \times (m-(n-i))} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ \hline 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

→ Konstruktion von  $\mathbf{T}_i$  aus linkem Orthokomplement  $\mathbf{J}_i^{\text{L}\perp}$  und Pseudoinverser  $\mathbf{J}_i^{\text{L}+}$ :

$$\mathbf{J}_i^{\text{L}+} \mathbf{J}_i = \mathbf{I}_{m-(n-i)}, \quad \mathbf{J}_i^{\text{L}\perp} \mathbf{J}_i = \mathbf{0}_{(n-m) \times (m-(n-i))} \quad \Rightarrow \mathbf{T}_i = \begin{pmatrix} \mathbf{J}_i^{\text{L}+} \\ \mathbf{J}_i^{\text{L}\perp} \end{pmatrix}$$

# Flachheitsanalyse von MIMO-Systemen

## Parametrisierung der Systemgrößen durch den flachen Ausgang:

- Systemgleichungen in Umgekehrter Reihenfolge zur Elimination nach Systemgrößen auflösen

# Flachheitsanalyse von MIMO-Systemen

## Parametrisierung der Systemgrößen durch den flachen Ausgang:

- Systemgleichungen in Umgekehrter Reihenfolge zur Elimination nach Systemgrößen auflösen
- Parametrisierungen vorheriger Größen ggf. in folgende Parametrisierungen einsetzen

# Flachheitsanalyse von MIMO-Systemen

## Parametrisierung der Systemgrößen durch den flachen Ausgang:

- Systemgleichungen in Umgekehrter Reihenfolge zur Elimination nach Systemgrößen auflösen
- Parametrisierungen vorheriger Größen ggf. in folgende Parametrisierungen einsetzen
- ggf. Zeitableitungen vorangegangener Parametrisierungen nötig

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 3$ ) Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_2 = (\tau_1, \tau_2)^T$ :

$$\begin{aligned} m_2 \ddot{p}_1 - \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \\ gm_2 + m_2 \ddot{p}_2 - \frac{p_2 \tau_2}{l_1} &= 0 \quad \Rightarrow \quad \mathbf{J}_3 = \begin{pmatrix} 0 & -\frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ 0 & -\frac{p_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ -1 & \frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \end{pmatrix} \\ m_1 \ddot{q}_1 - \tau_1 + \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \end{aligned}$$

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 3$ ) Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_2 = (\tau_1, \tau_2)^T$ :

$$\begin{aligned} m_2 \ddot{p}_1 - \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \\ gm_2 + m_2 \ddot{p}_2 - \frac{p_2 \tau_2}{l_1} &= 0 \quad \Rightarrow \quad \mathbf{J}_3 = \begin{pmatrix} 0 & -\frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ 0 & -\frac{p_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ -1 & \frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \end{pmatrix} \\ m_1 \ddot{q}_1 - \tau_1 + \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \end{aligned}$$

→ Elimination von  $\tau_1$  und letzter Gleichung

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\begin{aligned} m_2 \ddot{p}_1 - \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \\ gm_2 + m_2 \ddot{p}_2 - \frac{p_2 \tau_2}{l_1} &= 0 \end{aligned} \quad \Rightarrow \quad \mathbf{J}_2 = \begin{pmatrix} -\frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ -\frac{p_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \end{pmatrix}$$

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\begin{aligned} m_2 \ddot{p}_1 - \frac{\tau_2 (p_1 - q_1)}{l_1} &= 0 \\ gm_2 + m_2 \ddot{p}_2 - \frac{p_2 \tau_2}{l_1} &= 0 \end{aligned} \quad \Rightarrow \quad \mathbf{J}_2 = \begin{pmatrix} -\frac{p_1 - q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \\ -\frac{p_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \end{pmatrix}$$

→ keine Spalte mit nur einem nicht-Null-Eintrag, also Transformation  $\mathbf{T}_2$  nötig:

$$\begin{aligned} \mathbf{J}_2^{\text{L}+} &= \left( (J_{2,(1,1)})^{-1} \mathbf{0}_{1 \times 1} \right), \mathbf{J}_2^{\text{L}\perp} = (-J_{2,(2,1)} \quad J_{2,(1,1)}) \\ \Rightarrow \mathbf{T}_2 &= \begin{pmatrix} \frac{\sqrt{p_1^2 - 2p_1q_1 + p_2^2 + q_1^2}}{-p_1 + q_1} & 0 \\ \frac{p_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}} & \frac{-p_1 + q_1}{\sqrt{p_2^2 + (p_1 - q_1)^2}} \end{pmatrix} \end{aligned}$$

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) transformierte Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\frac{\left( -m_2 \ddot{p}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + \tau_2 (p_1 - q_1) \right)}{(p_1 - q_1)} = 0 \quad \Rightarrow \quad \mathbf{J}'_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$g \frac{m_2 (-gp_1 + gq_1 - p_1 \ddot{p}_2 + p_2 \ddot{p}_1 + \ddot{p}_2 q_1)}{\sqrt{p_1^2 - 2p_1 q_1 + p_2^2 + q_1^2}} = 0$$

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) transformierte Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\frac{\left( -m_2 \ddot{p}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + \tau_2 (p_1 - q_1) \right)}{(p_1 - q_1)} = 0 \quad \Rightarrow \quad \mathbf{J}'_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$g \frac{m_2 (-gp_1 + gq_1 - p_1 \ddot{p}_2 + p_2 \ddot{p}_1 + \ddot{p}_2 q_1)}{\sqrt{p_1^2 - 2p_1 q_1 + p_2^2 + q_1^2}} = 0$$

→ Elimination von  $\tau_2$  und erster transformierter Gleichung

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) transformierte Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\frac{\left( -m_2 \ddot{p}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + \tau_2 (p_1 - q_1) \right)}{(p_1 - q_1)} = 0 \quad \Rightarrow \quad \mathbf{J}'_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$g \frac{m_2 (-gp_1 + gq_1 - p_1 \ddot{p}_2 + p_2 \ddot{p}_1 + \ddot{p}_2 q_1)}{\sqrt{p_1^2 - 2p_1 q_1 + p_2^2 + q_1^2}} = 0$$

→ Elimination von  $\tau_2$  und erster transformierter Gleichung

→ letzte übrige Gleichung enthält folgende Menge an Systemgrößen:

$$\mathcal{M} = \{p_1, \ddot{p}_1, p_2, \ddot{p}_2, q_1\}$$

# Flachheitsanalyse am Einzelkran

## Elimination von Systemgrößen- und Gleichungen:

- ( $i = 2$ ) transformierte Systemgleichungen und Jacobi-Matrix bezüglich  $\mathbf{u}_1 = \tau_2$ :

$$\frac{\left( -m_2 \ddot{p}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + \tau_2 (p_1 - q_1) \right)}{(p_1 - q_1)} = 0 \quad \Rightarrow \quad \mathbf{J}'_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$g \frac{m_2 (-gp_1 + gq_1 - p_1 \ddot{p}_2 + p_2 \ddot{p}_1 + \ddot{p}_2 q_1)}{\sqrt{p_1^2 - 2p_1 q_1 + p_2^2 + q_1^2}} = 0$$

→ Elimination von  $\tau_2$  und erster transformierter Gleichung

→ letzte übrige Gleichung enthält folgende Menge an Systemgrößen:

$$\mathcal{M} = \{p_1, \ddot{p}_1, p_2, \ddot{p}_2, q_1\}$$

- rein algebraisches Auftreten von  $q_1$  führt zu Wahl von  $\mathbf{y} = (p_1, p_2)^T$

# Flachheitsanalyse am Einzelkran

Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :

# Flachheitsanalyse am Einzelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- aus letzter transformierter Systemgleichung:

$$q_1 = \frac{gp_1 + p_1\ddot{p}_2 - p_2\ddot{p}_1}{g + \ddot{p}_2}$$

# Flachheitsanalyse am Einzelkran

Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :

- aus letzter transformierter Systemgleichung:

$$q_1 = \frac{gp_1 + p_1\ddot{p}_2 - p_2\ddot{p}_1}{g + \ddot{p}_2}$$

- aus erster transformierter Systemgleichung und Einsetzen von  $q_1(y, \dot{y})$ :

$$\tau_2 = \frac{m_2\ddot{p}_1 \sqrt{p_2^2 + \left(p_1 - \frac{gp_1 + p_1\ddot{p}_2 - p_2\ddot{p}_1}{g + \ddot{p}_2}\right)^2}}{p_1 - \frac{gp_1 + p_1\ddot{p}_2 - p_2\ddot{p}_1}{g + \ddot{p}_2}}$$

# Flachheitsanalyse am Einzelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- aus zuerst eliminiertem originaler Systemgleichung:

$$\tau_1 = \frac{m_1 \ddot{q}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + p_1 \tau_2 - q_1 \tau_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}}$$

# Flachheitsanalyse am Einzelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- aus zuerst eliminiertem originaler Systemgleichung:

$$\tau_1 = \frac{m_1 \ddot{q}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + p_1 \tau_2 - q_1 \tau_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}}$$

→ auch Größen  $q_1, \ddot{q}_1, \tau_2 \notin y$  in dieser Parametrisierung von  $\tau_1$  enthalten

# Flachheitsanalyse am Einzelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- aus zuerst eliminiertem originaler Systemgleichung:

$$\tau_1 = \frac{m_1 \ddot{q}_1 \sqrt{p_2^2 + (p_1 - q_1)^2} + p_1 \tau_2 - q_1 \tau_2}{\sqrt{p_2^2 + (p_1 - q_1)^2}}$$

→ auch Größen  $q_1, \ddot{q}_1, \tau_2 \notin y$  in dieser Parametrisierung von  $\tau_1$  enthalten  
→ Einsetzen der zweiten Ableitung von  $q_1(y, \ddot{y})$  und  $\tau_2(y, \ddot{y})$ :

$$\begin{aligned}\tau_1 = & \frac{1}{g^3 + 3g^2\ddot{p}_2 + 3g\ddot{p}_2^2 + \ddot{p}_2^3} (g^3 m_1 \ddot{p}_1 + g^3 m_2 \ddot{p}_1 - g^2 m_1 p_2 \ddot{\ddot{p}}_1 - 2g^2 m_1 \ddot{p}_1 \dot{p}_2 \\ & + 2g^2 m_1 \ddot{p}_1 \ddot{p}_2 + 3g^2 m_2 \ddot{p}_1 \ddot{p}_2 - 2g m_1 p_2 \ddot{\ddot{p}}_1 \ddot{p}_2 + g m_1 p_2 \ddot{\ddot{p}}_2 \ddot{p}_1 + 2g m_1 p_2 \ddot{\ddot{p}}_1 \ddot{p}_2 \\ & - 4g m_1 \ddot{\ddot{p}}_1 \ddot{p}_2 \dot{p}_2 + 2g m_1 \ddot{\ddot{p}}_2 \ddot{p}_1 \dot{p}_2 + g m_1 \ddot{p}_1 \ddot{p}_2^2 + 3g m_2 \ddot{p}_1 \ddot{p}_2^2 - m_1 p_2 \ddot{\ddot{p}}_1 \ddot{p}_2^2 + m_1 p_2 \ddot{\ddot{p}}_2 \ddot{p}_1 \ddot{p}_2 \\ & + 2m_1 p_2 \ddot{\ddot{p}}_1 \ddot{p}_2 \ddot{p}_2 - 2m_1 p_2 \ddot{p}_2^2 \ddot{p}_1 - 2m_1 \ddot{p}_1 \ddot{p}_2^2 \dot{p}_2 + 2m_1 \ddot{p}_2 \ddot{p}_1 \ddot{p}_2 \dot{p}_2 + m_2 \ddot{p}_1 \ddot{p}_2^3)\end{aligned}$$

# Flachheitsanalyse am Einzelkran

**Parametisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- Zusammenfassung der Parametrisierungen:

$$\tau_1 = \theta_1 \left( p_2, \dot{p}_2, \ddot{p}_1, \ddot{p}_2, p_1^{(3)}, p_2^{(3)}, p_1^{(4)}, p_2^{(4)} \right)$$

$$\tau_2 = \theta_2(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

$$q_1 = \Psi_1(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

# Flachheitsanalyse am Einzelkran

**Parametisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- Zusammenfassung der Parametrisierungen:

$$\tau_1 = \theta_1 \left( p_2, \dot{p}_2, \ddot{p}_1, \ddot{p}_2, p_1^{(3)}, p_2^{(3)}, p_1^{(4)}, p_2^{(4)} \right)$$

$$\tau_2 = \theta_2(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

$$q_1 = \Psi_1(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

→ alle Systemgrößen durch flachen Ausgang  $y = (p_1, p_2)^T$  parametrisiert

# Flachheitsanalyse am Einzelkran

**Parametisierung der Systemgrößen durch  $y = (p_1, p_2)^T$ :**

- Zusammenfassung der Parametrisierungen:

$$\tau_1 = \theta_1 \left( p_2, \dot{p}_2, \ddot{p}_1, \ddot{p}_2, p_1^{(3)}, p_2^{(3)}, p_1^{(4)}, p_2^{(4)} \right)$$

$$\tau_2 = \theta_2(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

$$q_1 = \Psi_1(p_1, p_2, \ddot{p}_1, \ddot{p}_2)$$

→ alle Systemgrößen durch flachen Ausgang  $y = (p_1, p_2)^T$  parametrisiert

→ konstruktiver Flachheitsnachweis erbracht     $\square$

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran
- unpraktikable Darstellung umfangreicher Terme, über CAS *Sympy* durchgeführt

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran  
→ unpraktikable Darstellung umfangreicher Terme, über CAS SymPy durchgeführt
- $n = 10$  Zustandskomponenten und -gleichungen,  $m = 4$  Eingangskomponenten

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran
- unpraktikable Darstellung umfangreicher Terme, über CAS SymPy durchgeführt
- $n = 10$  Zustandskomponenten und -gleichungen,  $m = 4$  Eingangskomponenten
- Elimination des Eingangs, autonomes System aus  $p = n - m = 6$  Gleichungen

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran
- unpraktikable Darstellung umfangreicher Terme, über CAS SymPy durchgeführt
- $n = 10$  Zustandskomponenten und -gleichungen,  $m = 4$  Eingangskomponenten
- Elimination des Eingangs, autonomes System aus  $p = n - m = 6$  Gleichungen
- Elimination von  $p$  Zuständen zu flachem Ausgang aus  $m = 4$  Komponenten

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran
- unpraktikable Darstellung umfangreicher Terme, über CAS SymPy durchgeführt
- $n = 10$  Zustandskomponenten und -gleichungen,  $m = 4$  Eingangskomponenten  
→ Elimination des Eingangs, autonomes System aus  $p = n - m = 6$  Gleichungen  
→ Elimination von  $p$  Zuständen zu flachem Ausgang aus  $m = 4$  Komponenten
- letzte übrige Gleichung enthält folgende Menge an Systemgrößen:

$$\mathcal{M} = \{p_1, p_2, p_3, \ddot{p}_1, \ddot{p}_2, \ddot{p}_3, q_1, q_2\}$$

# Flachheitsanalyse am Doppelkran

## Elimination von Systemgrößen- und Gleichungen:

- Vorgehen analog zum Einzelkran
- unpraktikable Darstellung umfangreicher Terme, über CAS SymPy durchgeführt
- $n = 10$  Zustandskomponenten und -gleichungen,  $m = 4$  Eingangskomponenten  
→ Elimination des Eingangs, autonomes System aus  $p = n - m = 6$  Gleichungen  
→ Elimination von  $p$  Zuständen zu flachem Ausgang aus  $m = 4$  Komponenten
- letzte übrige Gleichung enthält folgende Menge an Systemgrößen:

$$\mathcal{M} = \{p_1, p_2, p_3, \ddot{p}_1, \ddot{p}_2, \ddot{p}_3, q_1, q_2\}$$

→ algebraisches Auftreten von  $q_1, q_2$  führt o. B. d. A. zur Wahl von  $\mathbf{y} = (p_1, p_2, p_3, q_1)^T$

# Flachheitsanalyse am Doppelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2, p_3, q_1)^T$  :**

- funktionale Zusammenhänge der Parametrisierungen:

$$\tau_1 = \theta_1(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1, \ddot{q}_1)$$

$$\tau_2 = \theta_2\left(p_1, \dot{p}_1, \ddot{p}_1, p_1^{(3)}, p_1^{(4)}, p_2, \dot{p}_2, \ddot{p}_2, p_2^{(3)}, p_2^{(4)}, p_3, \dot{p}_3, \ddot{p}_3, p_3^{(3)}, p_3^{(4)}, q_1, \dot{q}_1, \ddot{q}_1\right)$$

$$\tau_3 = \theta_3(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

$$\tau_4 = \theta_4(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

$$q_2 = \Psi_1(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

# Flachheitsanalyse am Doppelkran

**Parametrisierung der Systemgrößen durch  $y = (p_1, p_2, p_3, q_1)^T$  :**

- funktionale Zusammenhänge der Parametrisierungen:

$$\tau_1 = \theta_1(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1, \ddot{q}_1)$$

$$\tau_2 = \theta_2\left(p_1, \dot{p}_1, \ddot{p}_1, p_1^{(3)}, p_1^{(4)}, p_2, \dot{p}_2, \ddot{p}_2, p_2^{(3)}, p_2^{(4)}, p_3, \dot{p}_3, \ddot{p}_3, p_3^{(3)}, p_3^{(4)}, q_1, \dot{q}_1, \ddot{q}_1\right)$$

$$\tau_3 = \theta_3(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

$$\tau_4 = \theta_4(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

$$q_2 = \Psi_1(p_1, \ddot{p}_1, p_2, \ddot{p}_2, p_3, \ddot{p}_3, q_1)$$

→ alle Systemgrößen durch flachen Ausgang  $y = (p_1, p_2, p_3, q_1)^T$  parametrisiert

# Gliederung

- System- und Problembeschreibung
  - Analytische Modellbildung
  - Flachheitsanalyse
- **Steuerungs- und Regelungsentwurf**

# Steuerungs- und Regelungsentwurf

## Allgemeines Vorgehen:

- Planung polynom basierter Referenztrajektorien für den flachen Ausgang  $y$

# Steuerungs- und Regelungsentwurf

## Allgemeines Vorgehen:

- Planung polynom basierter Referenztrajektorien für den flachen Ausgang  $y$
- Trajektorien der Systemeingänge  $\tau(t)$  folgen aus  $y(t)$

# Steuerungs- und Regelungsentwurf

## Allgemeines Vorgehen:

- Planung polynombasierter Referenztrajektorien für den flachen Ausgang  $y$
  - Trajektorien der Systemeingänge  $\tau(t)$  folgen aus  $y(t)$
- **Vorsteuerung** durch Parametrisierung aus Flachheitsnachweis möglich

# Steuerungs- und Regelungsentwurf

## Allgemeines Vorgehen:

- Planung polynombasierter Referenztrajektorien für den flachen Ausgang  $y$
  - Trajektorien der Systemeingänge  $\tau(t)$  folgen aus  $y(t)$
- **Vorsteuerung** durch Parametrisierung aus Flachheitsnachweis möglich
- Entwurf einer **Folgeregelung** um diese Trajektorien

# Trajektorienplanung

## Anforderungen:

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen
- bisher keine formal spezifizierten Grenzwerte für Beschleunigungen etc.

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen
- bisher keine formal spezifizierten Grenzwerte für Beschleunigungen etc.

## Vorgabe von Randbedingungen:

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen
- bisher keine formal spezifizierten Grenzwerte für Beschleunigungen etc.

## Vorgabe von Randbedingungen:

- Differenzierbarkeitsbedingungen an  $y(t)$  folgen aus  $\tau(y(t), \dot{y}(t), \dots)$

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen
- bisher keine formal spezifizierten Grenzwerte für Beschleunigungen etc.

## Vorgabe von Randbedingungen:

- Differenzierbarkeitsbedingungen an  $y(t)$  folgen aus  $\tau(y(t), \dot{y}(t), \dots)$
- höchste Ableitungen von  $y(t)$  in  $\tau_2 = \theta_2 \left( y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots \right)$

# Trajektorienplanung

## Anforderungen:

- Überführung des Doppelkransystems zwischen zwei Ruhelagen
- stetiger Verlauf von Eingangs- bzw. Stellgrößen
- bisher keine formal spezifizierten Grenzwerte für Beschleunigungen etc.

## Vorgabe von Randbedingungen:

- Differenzierbarkeitsbedingungen an  $y(t)$  folgen aus  $\tau(y(t), \dot{y}(t), \dots)$   
→ höchste Ableitungen von  $y(t)$  in  $\tau_2 = \theta_2 \left( y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots \right)$
- stetig differenzierbarer Verlauf von  $\tau(t)$  für Überführung von  $(t_0, y_{i,0})$  in  $(t_e, y_{i,e})$

# Trajektorienplanung

**Vorgabe von Randbedingungen an  $y(t)$  für  $\tau_2 = \theta_2 \left( y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots \right)$ :**

$$y_i(t_0) = y_{i,0} \quad \text{für } i = 1, 2, 3, 4$$

$$y_i(t_e) = y_{i,e} \quad \text{für } i = 1, 2, 3, 4$$

$$\dot{y}_i(t_0) = \ddot{y}_i(t_0) = y_i^{(3)}(t_0) = y_i^{(4)}(t_0) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_i(t_e) = \ddot{y}_i(t_e) = y_i^{(3)}(t_e) = y_i^{(4)}(t_e) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_4(t_0) = \ddot{y}_4(t_0) = \dot{y}_4(t_e) = \ddot{y}_4(t_e) = 0$$

# Trajektorienplanung

**Vorgabe von Randbedingungen an  $y(t)$  für  $\tau_2 = \theta_2(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots)$ :**

$$y_i(t_0) = y_{i,0} \quad \text{für } i = 1, 2, 3, 4$$

$$y_i(t_e) = y_{i,e} \quad \text{für } i = 1, 2, 3, 4$$

$$\dot{y}_i(t_0) = \ddot{y}_i(t_0) = y_i^{(3)}(t_0) = y_i^{(4)}(t_0) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_i(t_e) = \ddot{y}_i(t_e) = y_i^{(3)}(t_e) = y_i^{(4)}(t_e) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_4(t_0) = \ddot{y}_4(t_0) = \dot{y}_4(t_e) = \ddot{y}_4(t_e) = 0$$

→ Polynomansatz für  $y_i(t)$  mit Ordnung  $N_i - 1$  mit  $N_i$  Anzahl der Randbedingungen:

# Trajektorienplanung

**Vorgabe von Randbedingungen an  $y(t)$  für  $\tau_2 = \theta_2 \left( y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots \right)$ :**

$$y_i(t_0) = y_{i,0} \quad \text{für } i = 1, 2, 3, 4$$

$$y_i(t_e) = y_{i,e} \quad \text{für } i = 1, 2, 3, 4$$

$$\dot{y}_i(t_0) = \ddot{y}_i(t_0) = y_i^{(3)}(t_0) = y_i^{(4)}(t_0) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_i(t_e) = \ddot{y}_i(t_e) = y_i^{(3)}(t_e) = y_i^{(4)}(t_e) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_4(t_0) = \ddot{y}_4(t_0) = \dot{y}_4(t_e) = \ddot{y}_4(t_e) = 0$$

→ Polynomansatz für  $y_i(t)$  mit Ordnung  $N_i - 1$  mit  $N_i$  Anzahl der Randbedingungen:

$$y_i(t) = a_{i,9}t^9 + a_{i,8}t^8 + \dots + a_{i,0} \quad \text{für } i = 1, 2, 3; t_0 < t < t_e$$

$$y_4(t) = a_{4,5}t^5 + a_{4,4}t^4 + \dots + a_{4,0} \quad \text{für } t_0 < t < t_e.$$

# Trajektorienplanung

**Vorgabe von Randbedingungen an  $y(t)$  für  $\tau_2 = \theta_2(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \ddot{y}_4, \dots)$ :**

$$y_i(t_0) = y_{i,0} \quad \text{für } i = 1, 2, 3, 4$$

$$y_i(t_e) = y_{i,e} \quad \text{für } i = 1, 2, 3, 4$$

$$\dot{y}_i(t_0) = \ddot{y}_i(t_0) = y_i^{(3)}(t_0) = y_i^{(4)}(t_0) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_i(t_e) = \ddot{y}_i(t_e) = y_i^{(3)}(t_e) = y_i^{(4)}(t_e) = 0 \quad \text{für } i = 1, 2, 3$$

$$\dot{y}_4(t_0) = \ddot{y}_4(t_0) = \dot{y}_4(t_e) = \ddot{y}_4(t_e) = 0$$

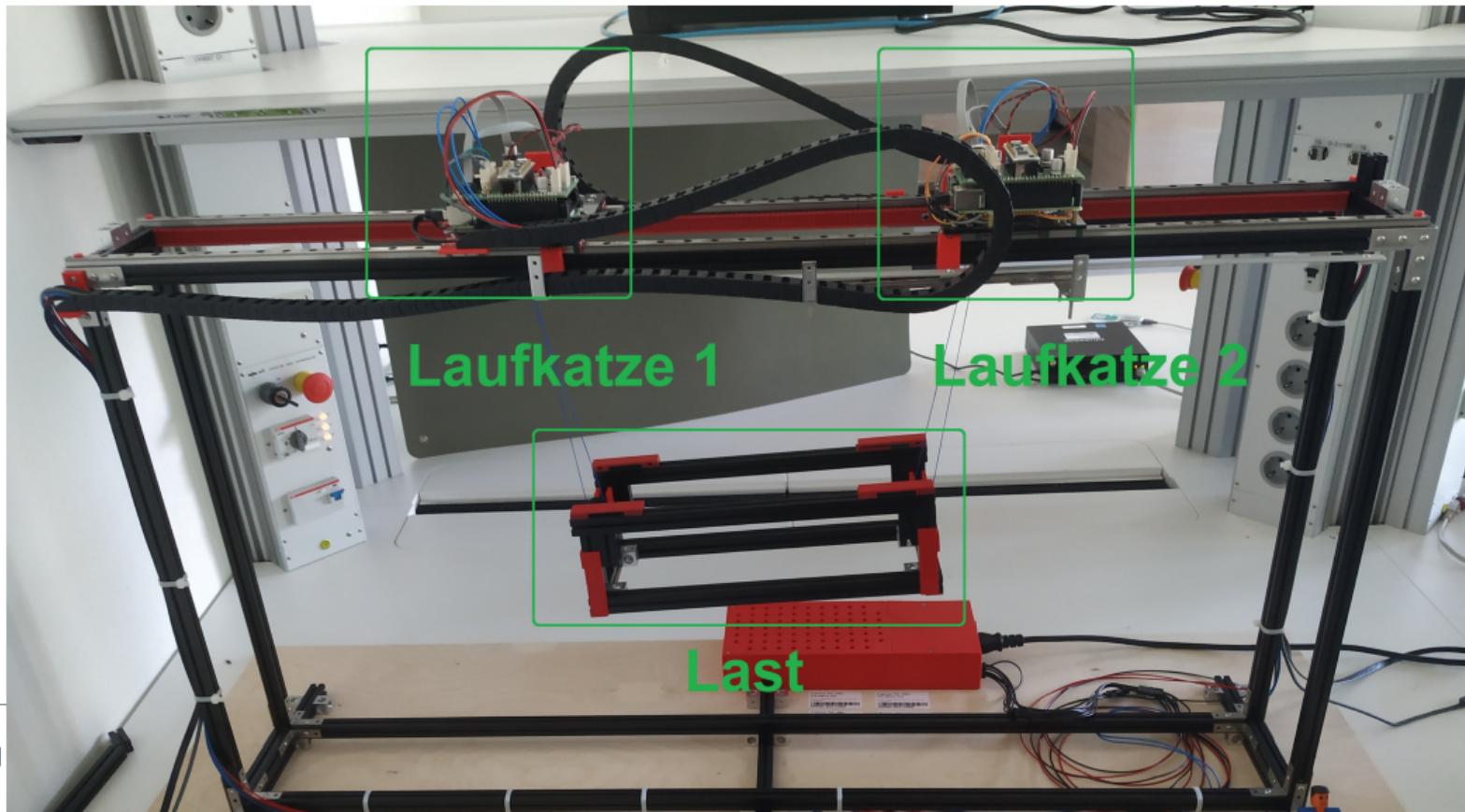
→ Polynomansatz für  $y_i(t)$  mit Ordnung  $N_i - 1$  mit  $N_i$  Anzahl der Randbedingungen:

$$y_i(t) = a_{i,9}t^9 + a_{i,8}t^8 + \dots + a_{i,0} \quad \text{für } i = 1, 2, 3; t_0 < t < t_e$$

$$y_4(t) = a_{4,5}t^5 + a_{4,4}t^4 + \dots + a_{4,0} \quad \text{für } t_0 < t < t_e.$$

→ Bestimmung der  $a_{i,j}$  aus linearem Gleichungssystem der Randbedingungen

# Trajektorienplanung



# Trajektorienfolgeregelung

## Vektorieller relativer Grad

Ein Mehrgrößensystem mit  $m$  Eingangskomponenten  $u_1, \dots, u_m$  und  $m$  Ausgangskomponenten  $y_1, \dots, y_m$  der Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x})$$

# Trajektorienfolgeregelung

## Vektorieller relativer Grad

Ein Mehrgrößensystem mit  $m$  Eingangskomponenten  $u_1, \dots, u_m$  und  $m$  Ausgangskomponenten  $y_1, \dots, y_m$  der Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x})$$

mit  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{f} : \mathcal{M} \rightarrow \mathbb{R}^n$ ,  $\mathbf{h} : \mathcal{M} \rightarrow \mathbb{R}^m$ ,  $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_m) : \mathcal{M} \rightarrow \mathbb{R}^{n \times m}$  wobei  $\mathcal{M} \subseteq \mathbb{R}^n$  hat an der Stelle  $\gamma \in \mathcal{M}$  den vektoriellen relativen Grad  $\mathbf{r} = (r_1, \dots, r_m)^T$ , falls:

# Trajektorienfolgeregelung

## Vektorieller relativer Grad

Ein Mehrgrößensystem mit  $m$  Eingangskomponenten  $u_1, \dots, u_m$  und  $m$  Ausgangskomponenten  $y_1, \dots, y_m$  der Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x})$$

mit  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{f} : \mathcal{M} \rightarrow \mathbb{R}^n$ ,  $\mathbf{h} : \mathcal{M} \rightarrow \mathbb{R}^m$ ,  $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_m) : \mathcal{M} \rightarrow \mathbb{R}^{n \times m}$  wobei  $\mathcal{M} \subseteq \mathbb{R}^n$  hat an der Stelle  $\gamma \in \mathcal{M}$  den vektoriellen relativen Grad  $\mathbf{r} = (r_1, \dots, r_m)^T$ , falls:

1. Die Lie-Ableitungen  $L_{\mathbf{g}_j} L_{\mathbf{f}}^k h_i(\mathbf{x}) = 0$  für alle  $\mathbf{x}$  aus einer Umgebung von  $\gamma$  sowie für alle  $i, j \in \{1, \dots, m\}$  und  $k \in \{0, \dots, r - 2\}$  und

# Trajektorienfolgeregelung

## Vektorieller relativer Grad

Ein Mehrgrößensystem mit  $m$  Eingangskomponenten  $u_1, \dots, u_m$  und  $m$  Ausgangskomponenten  $y_1, \dots, y_m$  der Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x})$$

mit  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{f} : \mathcal{M} \rightarrow \mathbb{R}^n$ ,  $\mathbf{h} : \mathcal{M} \rightarrow \mathbb{R}^m$ ,  $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_m) : \mathcal{M} \rightarrow \mathbb{R}^{n \times m}$  wobei  $\mathcal{M} \subseteq \mathbb{R}^n$  hat an der Stelle  $\gamma \in \mathcal{M}$  den vektoriellen relativen Grad  $\mathbf{r} = (r_1, \dots, r_m)^T$ , falls:

1. Die Lie-Ableitungen  $L_{\mathbf{g}_j} L_{\mathbf{f}}^{k-1} h_i(\mathbf{x}) = 0$  für alle  $\mathbf{x}$  aus einer Umgebung von  $\gamma$  sowie für alle  $i, j \in \{1, \dots, m\}$  und  $k \in \{0, \dots, r-2\}$  und
2. die Entkopplungsmatrix  $\Lambda$  im Punkt  $\mathbf{x} = \gamma$  regulär ist:

$$\Lambda = \begin{pmatrix} L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_1-1} h_1(\mathbf{x}) & \dots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{r_1-1} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_m-1} h_m(\mathbf{x}) & \dots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{r_m-1} h_m(\mathbf{x}) \end{pmatrix}.$$

# Motivation und Rückblick (1)

## Regelungstheorie hat heterogenes Methodenspektrum

- PT1-Glied, ..., nichtlineare PDEs, ...

# Motivation und Rückblick (1)

## Regelungstheorie hat heterogenes Methodenspektrum

- PT1-Glied, ..., nichtlineare PDEs, ... ⇒ z. T. hochspezialisierte Mathematik

# Motivation und Rückblick (1)

## Regelungstheorie hat heterogenes Methodenspektrum

- PT1-Glied, ..., nichtlineare PDEs, ... ⇒ z. T. hochspezialisierte Mathematik

## Regelungstheorie hat heterogenes Anwendungsspektrum

- Verfahrens-, Fahrzeug-, Gebäudetechnik, Robotik, ...

## Beständiger Wissenszuwachs

- überlineares Wachstum der Publikationen → Spezialisierung notwendig

# Motivation und Rückblick (1)

## Regelungstheorie hat heterogenes Methodenspektrum

- PT1-Glied, ..., nichtlineare PDEs, ... ⇒ z. T. hochspezialisierte Mathematik

## Regelungstheorie hat heterogenes Anwendungsspektrum

- Verfahrens-, Fahrzeug-, Gebäudetechnik, Robotik, ...

## Beständiger Wissenszuwachs

- überlineares Wachstum der Publikationen → Spezialisierung notwendig

## ⇒ Wissenstransfer: nichttrivial

- ... innerhalb der Regelungstechnik
- ... in Anwendungsdomänen

# Thesen

## These 1

Die gegenwärtig dominierenden Medien (Fließtext, Formeln, Grafiken) der Wissensrepräsentation sind in Bezug auf die angesprochenen Probleme suboptimal.

# Thesen

## These 1

Die gegenwärtig dominierenden Medien (Fließtext, Formeln, Grafiken) der Wissensrepräsentation sind in Bezug auf die angesprochenen Probleme suboptimal.

## These 2

Formale Wissensrepräsentationsmethoden stellen eine vielversprechende Ergänzung dar.

# Gliederung

Motivation und Rückblick

→ **Formale Wissensrepräsentation: Ist-Stand**

Ansatz: Emergent Representation of Knowledge (ERK)

Ontology of Control Systems Engineering (OCSE)

Zusammenfassung und Diskussion

# Formale Wissensrepräsentation (1): Begriffe

*Eine Ontologie [nach Studer et. al. 1998]*

Maschinenverarbeitbare Spezifikation der begrifflichen Abdeckung einer Wissensdomäne.

→ Welche Begriffe gibt es? In welcher Beziehung stehen sie zueinander?

# Formale Wissensrepräsentation (1): Begriffe

*Eine Ontologie [nach Studer et. al. 1998]*

Maschinenverarbeitbare Spezifikation der begrifflichen Abdeckung einer Wissensdomäne.

→ Welche Begriffe gibt es? In welcher Beziehung stehen sie zueinander?

Taxonomie

Hierarchisches Klassifikationssystem von ist-ein-Beziehungen

Beispiel: Hauskatze → Säugetier → Wirbeltier → Lebewesen

# Formale Wissensrepräsentation (2): Wissensgraphen und RDF

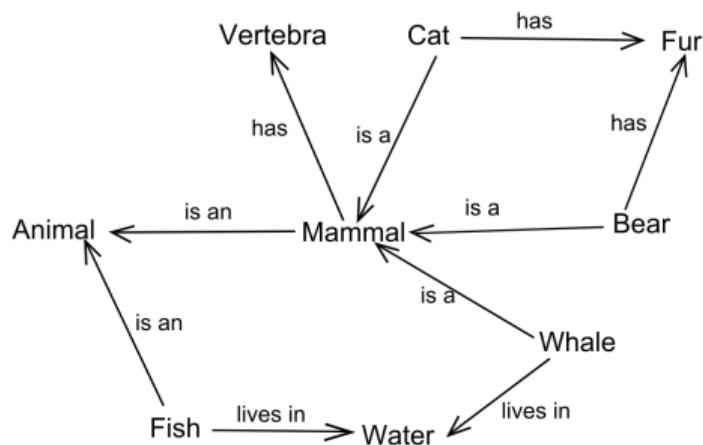
## „Knowledge Graph“:

- Knoten: Begriffe
- Kanten: Beziehungen

# Formale Wissensrepräsentation (2): Wissensgraphen und RDF

## „Knowledge Graph“:

- Knoten: Begriffe
- Kanten: Beziehungen

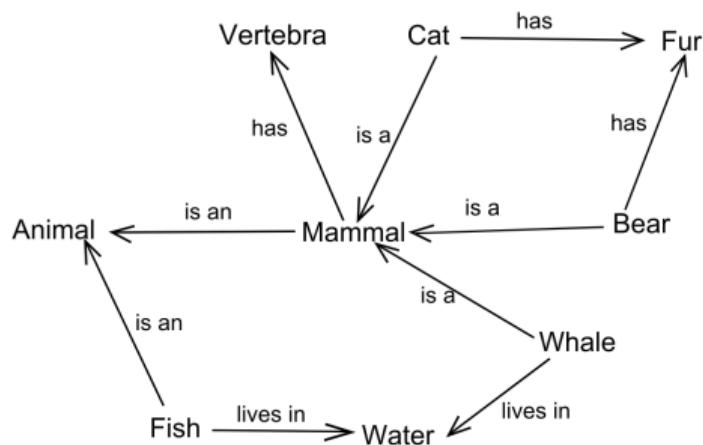


Quelle: [wikipedia.org/...](https://wikipedia.org/) (CC0)

# Formale Wissensrepräsentation (2): Wissensgraphen und RDF

## „Knowledge Graph“:

- Knoten: Begriffe
- Kanten: Beziehungen



Quelle: [wikipedia.org/...](https://wikipedia.org/) (CC0)

## Ressource Description Framework:

- Sprache zur Beschreibung von Subjekt-Prädikat-Objekt-Tripeln



- Zugehörige Abfragesprache:

**SPARQL**

(SPARQL Protocol And RDF Query Language)

# Formale Wissensrepräsentation (3): OWL und Inferenz

## Web Ontology Language

- OWL2: definierter Standard; basiert auf RDF
  - Theoretische Basis: Beschreibungslogik(en)
    - „Profile“ mit Unterschiedlicher Ausdrucksstärke
    - Entscheidbare Fragmente der Prädikatenlogik 1. Stufe
- ⇒ Einfluss auf Komplexität von Inferenz-Algorithmen

# Formale Wissensrepräsentation (3): OWL und Inferenz

## Web Ontology Language

- OWL2: definierter Standard; basiert auf RDF
  - Theoretische Basis: Beschreibungslogik(en)
    - „Profile“ mit Unterschiedlicher Ausdrucksstärke
    - Entscheidbare Fragmente der Prädikatenlogik 1. Stufe
- ⇒ Einfluss auf Komplexität von Inferenz-Algorithmen

## Inferenzsystem („Schließer“ bzw. Reasoner)

- Kann Schlussfolgerungen aus Behauptungen (Axiomen) ableiten
- Kann Inkonsistenzen aufdecken (widersprüchliche Axiome identifizieren)
- Kann implizit enthaltene Informationen explizit machen („Logikrätsel lösen“)

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:
  - [https://www.wikidata.org/wiki/Q252446 Anif bei Salzburg](https://www.wikidata.org/wiki/Q252446)
  - [https://www.wikidata.org/wiki/Q4917288 Control Engineering](https://www.wikidata.org/wiki/Q4917288)

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:
  - <https://www.wikidata.org/wiki/Q252446> *Anif bei Salzburg*
  - <https://www.wikidata.org/wiki/Q4917288> *Control Engineering*
- $\exists \text{ Properties}$ :
  - <https://www.wikidata.org/wiki/P31> *is instance of*
  - <https://www.wikidata.org/wiki/P2534> *has defining formula*
- $\exists \text{ Statements}$  (Kanten im Wissensgraph)

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:
  - [https://www.wikidata.org/wiki/Q252446 Anif bei Salzburg](https://www.wikidata.org/wiki/Q252446)
  - [https://www.wikidata.org/wiki/Q4917288 Control Engineering](https://www.wikidata.org/wiki/Q4917288)
- $\exists \text{ Properties}$ :
  - [https://www.wikidata.org/wiki/P31 is instance of](https://www.wikidata.org/wiki/P31)
  - [https://www.wikidata.org/wiki/P2534 has defining formula](https://www.wikidata.org/wiki/P2534)
- $\exists \text{ Statements}$  (Kanten im Wissensgraph)
- Abfrageschnittstelle über SPARQL



```
1 SELECT ?item ?itemLabel ?formula  
2 WHERE  
3 {  
4   ?item wdt:P31 wd:Q877802.    # P31 → instance of; Q877802 → integral transformation  
5   ?item wdt:P2534 ?formula.    # P2534 → defining formula  
6   ?item rdfs:label ?itemLabel.  
7 }
```



⚙️

?

9 results in 246 ms

Code

Download

Link

item	itemLabel	formula
wd:Q2867	wavelet transform	$w(f)(t, a, b) = \int_{-\infty}^{\infty} f\left(\frac{x-b}{a}\right)\psi(x, a, b) dx$
wd:Q199691	Laplace transform	$F(s) = \int_0^{\infty} f(t)e^{-st} dt$
wd:Q210857	convolution	$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau$

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:
  - <https://www.wikidata.org/wiki/Q252446> *Anif bei Salzburg*
  - <https://www.wikidata.org/wiki/Q4917288> *Control Engineering*
- $\exists \text{ Properties}$ :
  - <https://www.wikidata.org/wiki/P31> *is instance of*
  - <https://www.wikidata.org/wiki/P2534> *has defining formula*
- $\exists \text{ Statements}$  (Kanten im Wissensgraph)
- Abfrageschnittstelle über SPARQL

→ Repräsentation von mathematischen Inhalten in WD: umfangreich  
↳ Repräsentation von regelungstheoretischen Inhalten in WD: **dürftig**

# Thesen

## These 1

Die gegenwärtig dominierenden Medien (Fließtext, Formeln, Grafiken) der Wissensrepräsentation sind in Bezug auf die angesprochenen Probleme suboptimal.

## These 2

Formale Wissensrepräsentationsmethoden stellen eine vielversprechende Ergänzung dar.

## These 3

Bisher existierende technische Lösungen (OWL, Wikidata, ...) sind ungeeignet.

# Gliederung

- Motivation und Rückblick
- Formale Wissensrepräsentation: Ist-Stand

→ **Ansatz: Emergent Representation of Knowledge (ERK)**

- Ontology of Control Systems Engineering (OCSE)
- Zusammenfassung und Diskussion

# Experimenteller Ansatz: ERK

## Verklemmung:

- ① Kein geeigneter Repräsentationsformalismus → ② Keine Inhalte → ③ Keine Anwendungen → ④ Keine Aufmerksamkeit → ⑤ Keine Entwicklung → ①

# Experimenteller Ansatz: ERK

## Verklemmung:

- ① Kein geeigneter Repräsentationsformalismus → ② Keine Inhalte → ③ Keine Anwendungen → ④ Keine Aufmerksamkeit → ⑤ Keine Entwicklung → ①

## Pragmatischer Ansatz:

- ① Repräsentationsformalismus: **Emergent Representation of Knowledge**
- ② initiale Inhalte: OCSE

# Experimenteller Ansatz: ERK

## Verklemmung:

- ① Kein geeigneter Repräsentationsformalismus → ② Keine Inhalte → ③ Keine Anwendungen → ④ Keine Aufmerksamkeit → ⑤ Keine Entwicklung → ①

## Pragmatischer Ansatz:

- ① Repräsentationsformalismus: Emergent Representation of Knowledge  
② initiale Inhalte: OCSE

## Anforderungen

- Fokus auf Ausdrucksstärke
- Basiert auf menschenlesbarem Text
- Unterstützung für Automatisierung

# Experimenteller Ansatz: ERK

## Verklemmung:

- ① Kein geeigneter Repräsentationsformalismus → ② Keine Inhalte → ③ Keine Anwendungen → ④ Keine Aufmerksamkeit → ⑤ Keine Entwicklung → ①

## Pragmatischer Ansatz:

- ① Repräsentationsformalismus: **Emergent Representation of Knowledge**  
② initiale Inhalte: OCSE

## Anforderungen

- Fokus auf Ausdrucksstärke
  - Basiert auf menschenlesbarem Text
  - Unterstützung für Automatisierung
- } Erfüllt von Allzweckprogrammiersprache  
z. B. Python

# ERK: Imperative statt deklarative Wissensrepräsentation

## Deklarativ

- Formuliert in *Beschreibungssprache* (z.B. OWL (Turtle Syntax))

# ERK: Imperative statt deklarative Wissensrepräsentation

## Deklarativ

- Formuliert in *Beschreibungssprache* (z.B. OWL (Turtle Syntax))

```
I3749 rdfs:label "Cayley-Hamilton theorem".
```

```
I3749 rdfs:comment "every square matrix is a root of its own char. poly.". 
```

```
I3749 rdf:type :I15_implication_proposition.
```

# ERK: Imperative statt deklarative Wissensrepräsentation

## Deklarativ

- Formuliert in *Beschreibungssprache* (z.B. OWL (Turtle Syntax))

```
I3749 rdfs:label "Cayley-Hamilton theorem".  
I3749 rdfs:comment "every square matrix is a root of its own char. poly.".  
I3749 rdf:type :I15_implication_proposition.
```

## Imperativ

- Formuliert in *Programmiersprache* (z.B. Python)

```
I3749 = p.create_item(  
    R1__has_label="Cayley-Hamilton theorem",  
    R2__has_description="every square matrix is a root of its own char. poly.",  
    R4__is_instance_of=p.I15["implication proposition"],  
)
```

# ERK: Imperative statt deklarative Wissensrepräsentation (2)

## Vorteile

- Direkte Programm-interne Repräsentation (kein Parsen)
- Direkte Erweiterbarkeit (Plugins im Graphen)

# ERK: Imperative statt deklarative Wissensrepräsentation (2)

## Vorteile

- Direkte Programm-interne Repräsentation (kein Parsen)
- Direkte Erweiterbarkeit (Plugins im Graphen)

## Anwendung: Geltungsbereiche (scopes)

- Gliederung einer Aussage in *Kontext-Etablierung, Prämisse, Behauptung*
- Gesamtaussage abhängig davon in welchem *scope* eine Teilaussage steht
- Deklarativ: nur sehr aufwendig umsetzbar
- Imperativ: einfach umsetzbar (automatisches Erzeugen von Hilfsknoten/kanten)

# ERK: Imperative statt deklarative Wissensrepräsentation (2)

## Vorteile

- Direkte Programm-interne Repräsentation (kein Parsen)
- Direkte Erweiterbarkeit (Plugins im Graphen)

## Anwendung: Geltungsbereiche (scopes)

- Gliederung einer Aussage in *Kontext-Etablierung, Prämisse, Behauptung*
- Gesamtaussage abhängig davon in welchem scope eine Teilaussage steht
- Deklarativ: nur sehr aufwendig umsetzbar
- Imperativ: einfach umsetzbar (automatisches Erzeugen von Hilfsknoten/kanten)

```
I3749 = p.create_item(  
    R1__has_label="Cayley-Hamilton theorem",  
    R2__has_description="every square matrix is a root of its own char. poly.",  
    R4__is_instance_of=p.I15["implication proposition"],  
)
```

## ERK - Beispiel: *Inhalt* des Satzes von Cayley-Hamilton

```
with I3749["Cayley-Hamilton theorem"].scope("context") as cm:  
    cm.new_var(A=uq_instance_of(I9906["square matrix"]))  
    cm.new_var(n=uq_instance_of(p.I39["positive integer"]))
```

# ERK - Beispiel: *Inhalt* des Satzes von Cayley-Hamilton

```
with I3749["Cayley-Hamilton theorem"].scope("context") as cm:  
    cm.new_var(A=uq_instance_of(I9906["square matrix"]))  
    cm.new_var(n=uq_instance_of(p.I39["positive integer"]))  
  
    cm.new_var(P=p.instance_of(I4240["matrix polynomial"]))  
    cm.new_var(Z=p.instance_of(I9905["zero matrix"]))
```

## ERK - Beispiel: *Inhalt* des Satzes von Cayley-Hamilton

```
with I3749["Cayley-Hamilton theorem"].scope("context") as cm:  
    cm.new_var(A=uq_instance_of(I9906["square matrix"]))  
    cm.new_var(n=uq_instance_of(p.I39["positive integer"]))  
  
    cm.new_var(P=p.instance_of(I4240["matrix polynomial"]))  
    cm.new_var(Z=p.instance_of(I9905["zero matrix"]))  
  
    cm.new_rel(cm.A, R5938["has row number"], cm.n)  
    cm.new_rel(cm.A, R5940["has characteristic polynomial"], cm.P)  
    cm.new_rel(cm.Z, R5938["has row number"], cm.n)  
    cm.new_rel(cm.Z, R5939["has column number"], cm.n)
```

## ERK - Beispiel: *Inhalt* des Satzes von Cayley-Hamilton

```
with I3749["Cayley-Hamilton theorem"].scope("context") as cm:  
    cm.new_var(A=uq_instance_of(I9906["square matrix"]))  
    cm.new_var(n=uq_instance_of(p.I39["positive integer"]))  
  
    cm.new_var(P=p.instance_of(I4240["matrix polynomial"]))  
    cm.new_var(Z=p.instance_of(I9905["zero matrix"]))  
  
    cm.new_rel(cm.A, R5938["has row number"], cm.n)  
    cm.new_rel(cm.A, R5940["has characteristic polynomial"], cm.P)  
    cm.new_rel(cm.Z, R5938["has row number"], cm.n)  
    cm.new_rel(cm.Z, R5939["has column number"], cm.n)  
  
with I3749["Cayley-Hamilton theorem"].scope("assertions") as cm:  
    cm.new_equation(lhs=cm.P(cm.A), rhs=cm.Z)
```

$$P(\mathbf{A}) = \mathbf{Z} \quad \text{mit } \mathbf{Z} := \mathbf{0}$$

# ERK-Inferenzsystem

## Nachteile Imperativer Repräsentation

- ...
- Kein Einsatz existierender *Reasoner*

# ERK-Inferenzsystem

## Nachteile Imperativer Repräsentation

- ...
- Kein Einsatz existierender *Reasoner*

## → Regelbasierte Inferenz

- Beispiel: Klassifikation von  $\dot{x} = a \sin(x) + bx^2 + cx + u$
- *Eingangsaffinität*  $\supset$  *Polynomialität* ( $a=0$ )  $\supset$  *Linearität* ( $a,b=0$ )

# ERK-Inferenzsystem

## Nachteile Imperativer Repräsentation

- ...
- Kein Einsatz existierender *Reasoner*

## → Regelbasierte Inferenz

- Beispiel: Klassifikation von  $\dot{x} = a \sin(x) + bx^2 + cx + u$
- *Eingangsaffinität*  $\supset$  *Polynomialität* ( $a=0$ )  $\supset$  *Linearität* ( $a,b=0$ )
- Im Graph: I4761["linear"] R17["is subproperty of"] I5247["polynomial"]
- Wunsch: Schlussfolgerung von I4761["linear"] R17 I6091["control affine"]

# ERK-Inferenzsystem

## Nachteile Imperativer Repräsentation

- ...
- Kein Einsatz existierender *Reasoner*

## → Regelbasierte Inferenz

- Beispiel: Klassifikation von  $\dot{x} = a \sin(x) + bx^2 + cx + u$
- *Eingangsaffinität*  $\supset$  *Polynomialität* ( $a=0$ )  $\supset$  *Linearität* ( $a,b=0$ )
- Im Graph: I4761["linear"] R17["is subproperty of"] I5247["polynomial"]
- Wunsch: Schlussfolgerung von I4761["linear"] R17 I6091["control affine"]
- Abstrakt: Transitivität der Relation R17["is subproperty of"]
- Wunsch: Regel soll selbst Teil des Wissensgraphen sein

# ERK-Inferenzsystem: Regelspezifikation

```
I400 = p.create_item(  
    R1__has_label="transitivity of R17__is_subproperty_of",  
    R4__is_instance_of=p.I41["semantic rule"],  
)
```

# ERK-Inferenzsystem: Regelspezifikation

```
I400 = p.create_item(  
    R1__has_label="transitivity of R17__is_subproperty_of",  
    R4__is_instance_of=p.I41["semantic rule"],  
)  
  
with I400["subproperty rule 1"].scope("context") as cm:  
    cm.new_var(P1=p.instance_of(p.I11["mathematical property"]))  
    cm.new_var(P2=p.instance_of(p.I11["mathematical property"]))  
    cm.new_var(P3=p.instance_of(p.I11["mathematical property"]))
```

# ERK-Inferenzsystem: Regelspezifikation

```
I400 = p.create_item(
    R1__has_label="transitivity of R17__is_subproperty_of",
    R4__is_instance_of=p.I41["semantic rule"],
)

with I400["subproperty rule 1"].scope("context") as cm:
    cm.new_var(P1=p.instance_of(p.I11["mathematical property"]))
    cm.new_var(P2=p.instance_of(p.I11["mathematical property"]))
    cm.new_var(P3=p.instance_of(p.I11["mathematical property"]))

with I400.scope("premises") as cm:
    cm.new_rel(cm.P2, p.R17["is subproperty of"], cm.P1)
    cm.new_rel(cm.P3, p.R17["is subproperty of"], cm.P2)
```

# ERK-Inferenzsystem: Regelspezifikation

```
I400 = p.create_item(
    R1__has_label="transitivity of R17__is_subproperty_of",
    R4__is_instance_of=p.I41["semantic rule"],
)

with I400["subproperty rule 1"].scope("context") as cm:
    cm.new_var(P1=p.instance_of(p.I11["mathematical property"]))
    cm.new_var(P2=p.instance_of(p.I11["mathematical property"]))
    cm.new_var(P3=p.instance_of(p.I11["mathematical property"]))

with I400.scope("premises") as cm:
    cm.new_rel(cm.P2, p.R17["is subproperty of"], cm.P1)
    cm.new_rel(cm.P3, p.R17["is subproperty of"], cm.P2)

with I400.scope("assertions") as cm:
    cm.new_rel(cm.P3, p.R17["is subproperty of"], cm.P1)
```

# ERK-Inferenzsystem: Regelauswertung

- Für jede Regel: „Prototypgraphen“ konstruieren
  - lokale Variablen aus `scope("context")` → Knoten
  - Relationen aus `scope("premises")` → Kanten

# ERK-Inferenzsystem: Regelauswertung

- Für jede Regel: „Prototypgraphen“ konstruieren
  - lokale Variablen aus `scope("context")` → Knoten
  - Relationen aus `scope("premises")` → Kanten
- Passende Knoten aus dem Gesamtgraph suchen
  - Mathematisches Problem: ***Subgraphisomorphismen*** finden
  - $\exists$  VF2-Algorithmus (fertig implementiert)

# ERK-Inferenzsystem: Regelauswertung

- Für jede Regel: „Prototypgraphen“ konstruieren
  - lokale Variablen aus `scope("context")` → Knoten
  - Relationen aus `scope("premises")` → Kanten
- Passende Knoten aus dem Gesamtgraph suchen
  - Mathematisches Problem: ***Subgraphisomorphismen*** finden
  - $\exists$  VF2-Algorithmus (fertig implementiert)
- Beziehungen aus `scope("assertions")` abstrahieren und anwenden

# ERK – Bemerkungen

- Alle *Items* und *Relations* haben eindeutige URI (z. B. `erk:/builtins#I41`)
- Unterstützung für Mehrsprachigkeit (Label, Beschreibung, ...)
- Unterstützung für Qualifier (Kanten, die auf Kanten zeigen)
  - Semantische Information steckt in Knoten und Kanten
  - menschenlesbare Texte sind Hilfsattribute
  - Lesbarkeit für Mensch und Maschine: `I41["semantische Regel"@de] ≡ I41`
- Zur Begriffswahl „Emergent“ (ERK)
  - *Emergenz* wörtlich: „das Auftauchen“
  - Bedeutung: Phänomen, dass in komplexen Systemen Eigenschaften auftreten, die nicht aus den Eigenschaften der Elemente vorhergesagt werden können.  
→ : *Das Ganze ist mehr als die Summe seiner Teile.*
  - Ursprünglich: ERK – „Easy Knowledge Representation“
- $\exists$  **RDF-Export → SPARQL Suche möglich**

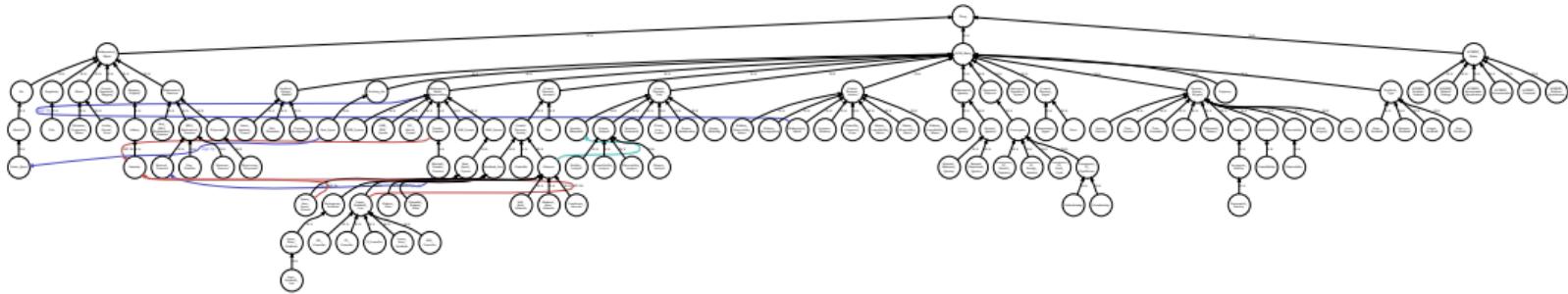
# Gliederung

- Motivation und Rückblick
- Formale Wissensrepräsentation: Ist-Stand
- Ansatz: Emergent Representation of Knowledge (ERK)
- **Ontology of Control Systems Engineering (OCSE)**
- Zusammenfassung und Diskussion

# Ontology of Control Systems Engineering

## 2021: OCSE 0.1

- in OWL implementiert
- Taxonomie einfach umsetzbar
- Weitere Beziehungen schwierig (mangelnde OWL-Ausdrucksstärke)



# Ontology of Control Systems Engineering

## 2021: OCSE 0.1

- in OWL implementiert
- Taxonomie einfach umsetzbar
- Weitere Beziehungen schwierig (mangelnde OWL-Ausdrucksstärke)

## 2022: OCSE 0.2

- Basierend auf ERK implementiert
- Größere Modellierungstiefe möglich

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? →

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? → Dynamische Systeme.

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? → Dynamische Systeme.

```
I5948 = p.create_item(  
    R1__has_label="dynamical system",  
    R4__is_instance_of=p.I2["Metaclass"] # <- Metaklassen-Instanzen sind Klassen  
)
```

Geht es wirklich um Systeme?

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? → Dynamische Systeme.

```
I5948 = p.create_item(  
    R1__has_label="dynamical system",  
    R4__is_instance_of=p.I2["Metaclass"] # <- Metaklassen-Instanzen sind Klassen  
)
```

Geht es wirklich um Systeme? → Es geht um *Modelle*.

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? → Dynamische Systeme.

```
I5948 = p.create_item(  
    R1__has_label="dynamical system",  
    R4__is_instance_of=p.I2["Metaclass"] # <- Metaklassen-Instanzen sind Klassen  
)
```

Geht es wirklich um Systeme? → Es geht um *Modelle*.

```
I7641 = p.create_item(  
    R1__has_label="general system model",  
    R4__is_instance_of=p.I2["Metaclass"],  
)
```

# OCSE – Wie anfangen?

Worum geht es in der Regelungstheorie? → Dynamische Systeme.

```
I5948 = p.create_item(  
    R1__has_label="dynamical system",  
    R4__is_instance_of=p.I2["Metaclass"] # <- Metaklassen-Instanzen sind Klassen  
)
```

Geht es wirklich um Systeme? → Es geht um *Modelle*.

```
I7641 = p.create_item(  
    R1__has_label="general system model",  
    R4__is_instance_of=p.I2["Metaclass"],  
)  
R7641 = p.create_relation(  
    R1__has_label="has approximation",  
    R8__has_domain_of_argument_1=I5948["dynamical system"],  
    R11__has_range_of_result=I7641["general system model"],  
)
```

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik
- Pragmatischer Ansatz: Irgendwo anfangen und schrittweise ergänzen

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik
- Pragmatischer Ansatz: Irgendwo anfangen und schrittweise ergänzen

**Beispiel:** R8133["relative degree"]

( $\hat{=}$   $\frac{d}{dt}$ -Ordnung des Systemausgangs, die erstmals explizit vom Eingang abhängt)

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik
- Pragmatischer Ansatz: Irgendwo anfangen und schrittweise ergänzen

**Beispiel:** R8133["relative degree"]

( $\hat{=}$   $\frac{d}{dt}$ -Ordnung des Systemausgangs, die erstmals explizit vom Eingang abhängt)  
benötigt:

- I1371["iterated Lie derivative of scalar field"]

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik
- Pragmatischer Ansatz: Irgendwo anfangen und schrittweise ergänzen

**Beispiel:** R8133["relative degree"]

( $\hat{=}$   $\frac{d}{dt}$ -Ordnung des Systemausgangs, die erstmals explizit vom Eingang abhängt)  
benötigt:

- I1371["iterated Lie derivative of scalar field"]
  - I1347["Lie derivative of scalar field"]

## Lie Ableitung

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

$$L_f h := \left( \frac{d}{dt} \varphi_t^f(x) \right) \Big|_{t=0}$$

# OCSE – Repräsentation von Mathematik

- Regelungstheoretische Aussagen benötigen unverzichtbar Mathematik
- Pragmatischer Ansatz: Irgendwo anfangen und schrittweise ergänzen

**Beispiel:** R8133["relative degree"]

( $\hat{=}$   $\frac{d}{dt}$ -Ordnung des Systemausgangs, die erstmals explizit vom Eingang abhängt)  
benötigt:

- I1371["iterated Lie derivative of scalar field"]
  - I1347["Lie derivative of scalar field"]
    - I2075["substitution"]
    - I3513["derivative w.r.t. scalar parameter"]
    - I2753["flow of a vector field"]
    - I9273["explicit first order ODE system"]
- ...

Lie Ableitung

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

$$L_f h := \left( \frac{d}{dt} \varphi_t^f(x) \right) \Big|_{t=0}$$

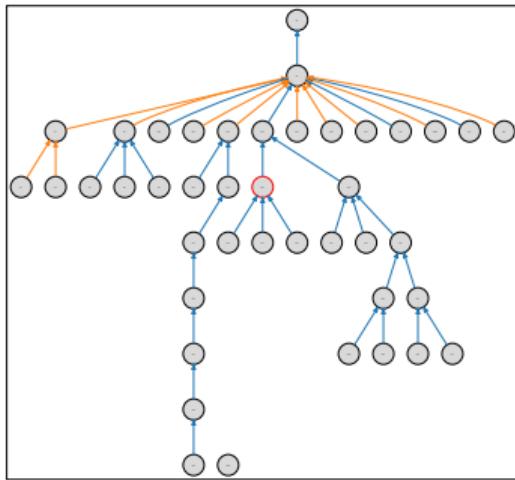
# OCSE – Status und Anwendungen

**Status:**  $\langle \text{erk:}/\text{builtins} \rangle \cup \langle \text{erk:}/\text{ocse}/0.2 \rangle: \approx 230$  Knoten, 400 Kanten

# OCSE – Status und Anwendungen

**Status:** `<erk:/builtins> ∪ <erk:/ocse/0.2>`:  $\approx 230$  Knoten, 400 Kanten

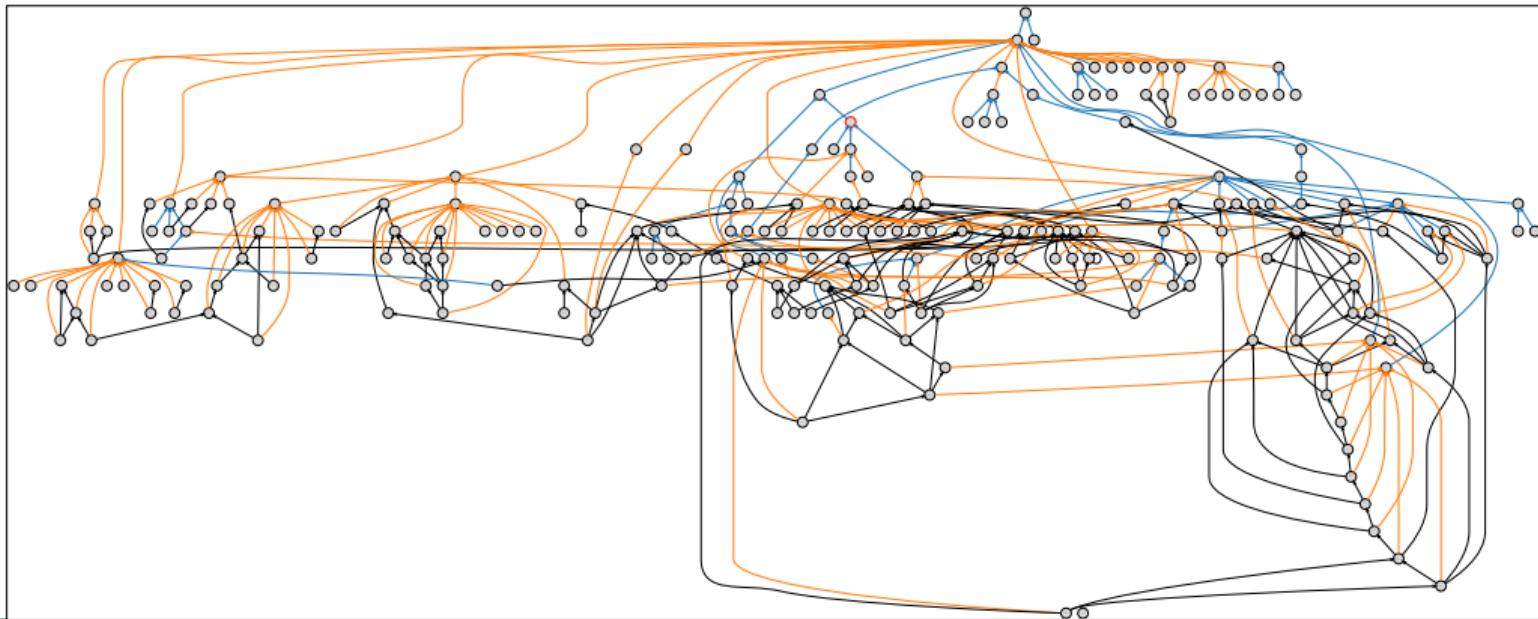
## Visualisierung:



# OCSE – Status und Anwendungen

**Status:** `<erk:/builtins> ∪ <erk:/ocse/0.2>`:  $\approx 230$  Knoten, 400 Kanten

## Visualisierung:

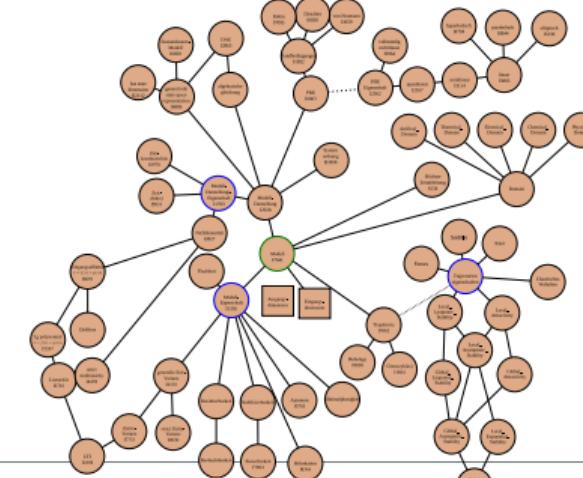


# OCSE – Status und Anwendungen

**Status:** `<erk:/builtins> ∪ <erk:/ocse/0.2>`:  $\approx 230$  Knoten, 400 Kanten

## Bisherige Anwendung:

Klassifikation von Entitäten im **Automatic Control Knowledge Repository (ACKREP)**  
(Systemmodelle, Problembeschreibungen, Lösungsmethoden)



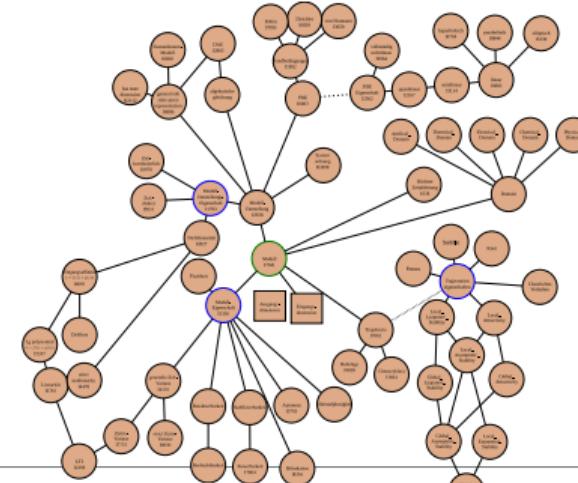
# OCSE – Status und Anwendungen

**Status:** `<erk:/builtins> ∪ <erk:/ocse/0.2>`:  $\approx 230$  Knoten, 400 Kanten

## Bisherige Anwendung:

Klassifikation von Entitäten im **Automatic Control Knowledge Repository (ACKREP)**  
(Systemmodelle, Problembeschreibungen, Lösungsmethoden)

→ Ermöglicht SPARQL-Suche  
(z. B. exakt E-Z-linearisierbare Modelle mit  $n > 3$ )



# OCSE – Status und Anwendungen

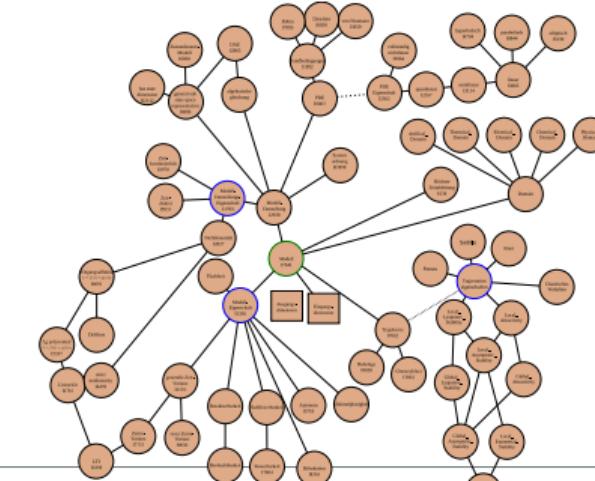
**Status:** `<erk:/builtins> ∪ <erk:/ocse/0.2>`:  $\approx 230$  Knoten, 400 Kanten

## Bisherige Anwendung:

Klassifikation von Entitäten im **Automatic Control Knowledge Repository (ACKREP)**  
(Systemmodelle, Problembeschreibungen, Lösungsmethoden)  
→ Ermöglicht SPARQL-Suche  
(z. B. exakt E-Z-linearisierbare Modelle mit  $n > 3$ )

## Mögliche zukünftige Anwendungen

- (Autom.) Klassifikation von Veröffentlichungen  
Denkbar: bis auf Satz- bzw. Gleichungsebene.
- Assistenzsoftware für Reglerentwurf  
→ Wissenstransfer



# Gliederung

- Motivation und Rückblick
- Formale Wissensrepräsentation: Ist-Stand
- Ansatz: Emergent Representation of Knowledge (ERK)
- Ontology of Control Systems Engineering (OCSE)
- **Zusammenfassung und Diskussion**

## Zusammenfassung und Diskussion

- Formale Wissensrepräsentation potenziell nützlich für Wissenstransfer

## Zusammenfassung und Diskussion

- Formale Wissensrepräsentation potenziell nützlich für Wissenstransfer
- Für Regelungstheorie existiert noch keine etablierte Technologie

## Zusammenfassung und Diskussion

- Formale Wissensrepräsentation potenziell nützlich für Wissenstransfer
- Für Regelungstheorie existiert noch keine etablierte Technologie
- Experimenteller Vorschlag: ERK + OCSE (Code und Daten sind Open Source)

→ <https://ackrep.org>

## Zusammenfassung und Diskussion

- Formale Wissensrepräsentation potenziell nützlich für Wissenstransfer
- Für Regelungstheorie existiert noch keine etablierte Technologie
- Experimenteller Vorschlag: ERK + OCSE (Code und Daten sind Open Source)

→ <https://ackrep.org>

## Diskussion (offene Fragen)

- Grundsätzliche Tauglichkeit?
- (Automatisierte) Qualitätssicherung?
- Wissensintegration als sozialer Prozess?

## Zusammenfassung und Diskussion

- Formale Wissensrepräsentation potenziell nützlich für Wissenstransfer
- Für Regelungstheorie existiert noch keine etablierte Technologie
- Experimenteller Vorschlag: ERK + OCSE (Code und Daten sind Open Source)

→ <https://ackrep.org>

## Diskussion (offene Fragen)

- Grundsätzliche Tauglichkeit?
- (Automatisierte) Qualitätssicherung?
- Wissensintegration als sozialer Prozess?



→ Herzliche Einladung zur Kollaboration

[Carsten.Knoll@tu-dresden.de](mailto:Carsten.Knoll@tu-dresden.de)

# Ergänzungsfolien

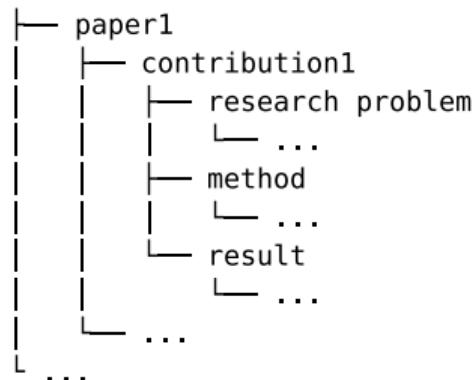
## Wissensrepräsentation (5): Open Research Knowledge Graph

- Projekt der Technischen Informationsbibliothek (TIB) Hannover
- Selbstbeschreibung: *Infrastruktur-Dienst zur Sammlung von akademischem Wissen in maschinenverarbeitbarer Form.*

## Wissensrepräsentation (5): Open Research Knowledge Graph

- Projekt der Technischen Informationsbibliothek (TIB) Hannover
- Selbstbeschreibung: *Infrastruktur-Dienst zur Sammlung von akademischem Wissen in maschinenverarbeitbarer Form.*

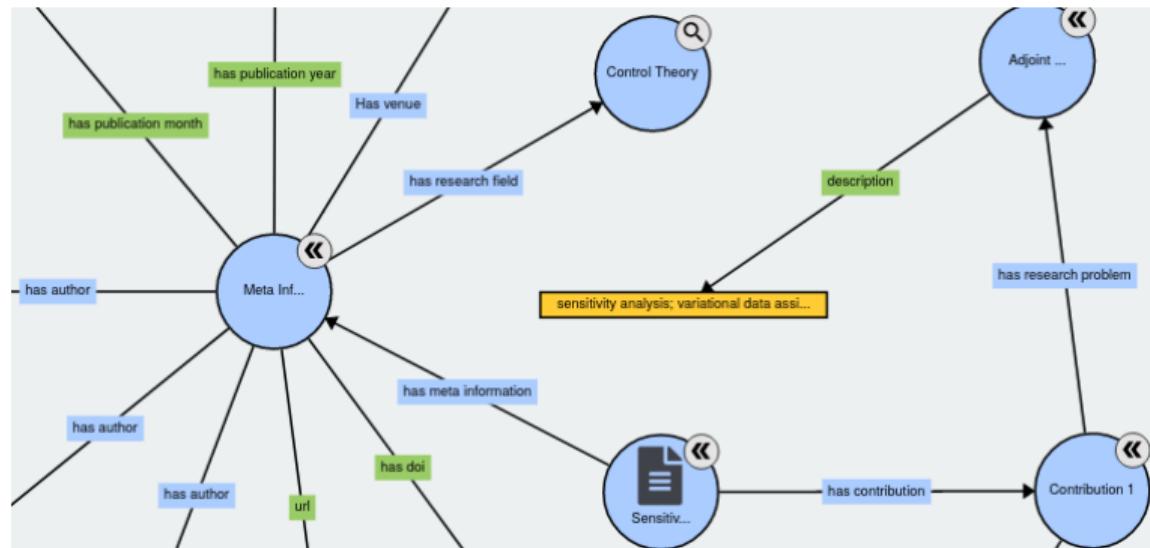
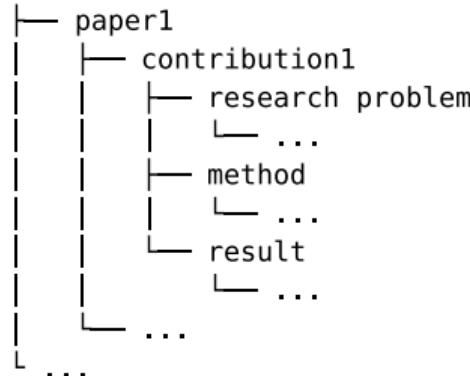
Grundlegendes Schema:



# Wissensrepräsentation (5): Open Research Knowledge Graph

- Projekt der Technischen Informationsbibliothek (TIB) Hannover
- Selbstbeschreibung: *Infrastruktur-Dienst zur Sammlung von akademischem Wissen in maschinenverarbeitbarer Form.*

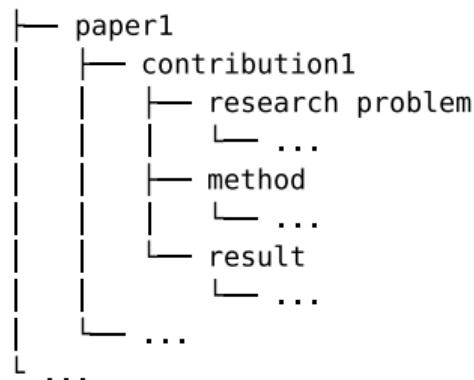
Grundlegendes Schema:



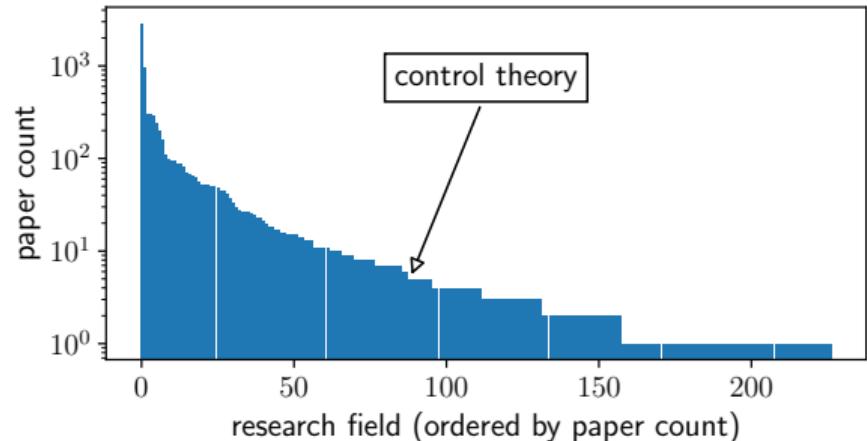
## Wissensrepräsentation (5): Open Research Knowledge Graph

- Projekt der Technischen Informationsbibliothek (TIB) Hannover
- Selbstbeschreibung: *Infrastruktur-Dienst zur Sammlung von akademischem Wissen in maschinenverarbeitbarer Form.*

Grundlegendes Schema:



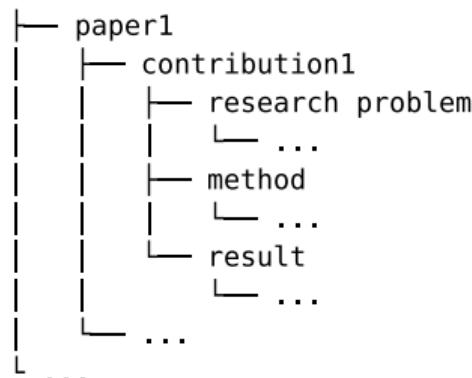
Repräsentanz der Regelungstheorie:



## Wissensrepräsentation (5): Open Research Knowledge Graph

- Projekt der Technischen Informationsbibliothek (TIB) Hannover
- Selbstbeschreibung: *Infrastruktur-Dienst zur Sammlung von akademischem Wissen in maschinenverarbeitbarer Form.*

Grundlegendes Schema:



→ Bisher nur „grobe“ Wissensrepräsentation möglich/üblich

## Formale Wissensrepräsentation (4): Wikidata und SPARQL

- Weltweit größter frei zugänglicher Wissensgraph
- Kollaborativ erstellt, von Wikimedia Foundation organisiert
- $\exists \text{ Items}$ : u. a. zu jedem Wikipedia-Eintrag:
  - <https://www.wikidata.org/wiki/Q252446> *Anif bei Salzburg*
  - <https://www.wikidata.org/wiki/Q4917288> *Control Engineering*
- $\exists \text{ Properties}$ :
  - <https://www.wikidata.org/wiki/P31> *is instance of*
  - <https://www.wikidata.org/wiki/P2534> *has defining formula*
- $\exists \text{ Statements}$  (Kanten im Wissensgraph)
- Abfrageschnittstelle über SPARQL

→ Repräsentation von mathematischen Inhalten in WD: umfangreich  
↳ Repräsentation von regelungstheoretischen Inhalten in WD: **dürftig**

Main page  
Community portal  
Project chat  
Create a new Item  
Recent changes  
Random Item  
Query Service  
Nearby  
Help  
Donate

Lexicographical data  
Create a new Lexeme  
Recent changes  
Random Lexeme

Tools  
What links here  
Related changes  
Special pages  
Permanent link  
Page information  
Concept URI  
Cite this page

# PID controller (Q716829)

control loop mechanism used in control engineering  
proportional-integral-derivative controller

[edit](#)

► In more languages

## Statements

subclass of	<a href="#">control engineering</a>	<a href="#">edit</a>
	▼ 0 references	<a href="#">Q4917288</a>

[+ add reference](#)

[+ add value](#)

described by source	<a href="#">Armenian Soviet Encyclopedia</a>	<a href="#">edit</a>
	volume 4	
	page(s) 313	

► 1 reference

[+ add value](#)

schematic		<a href="#">edit</a>
-----------	--	----------------------