

# EasyLab – Die Software für die Automatisierung

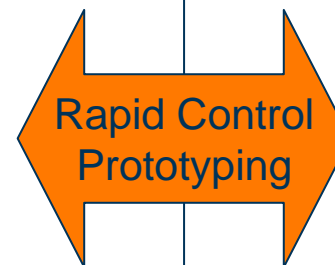
11.11.2009

Ch. Ament

Institut für Automatisierungs- und Systemtechnik  
TU Ilmenau

# Entwurf von Regelungen

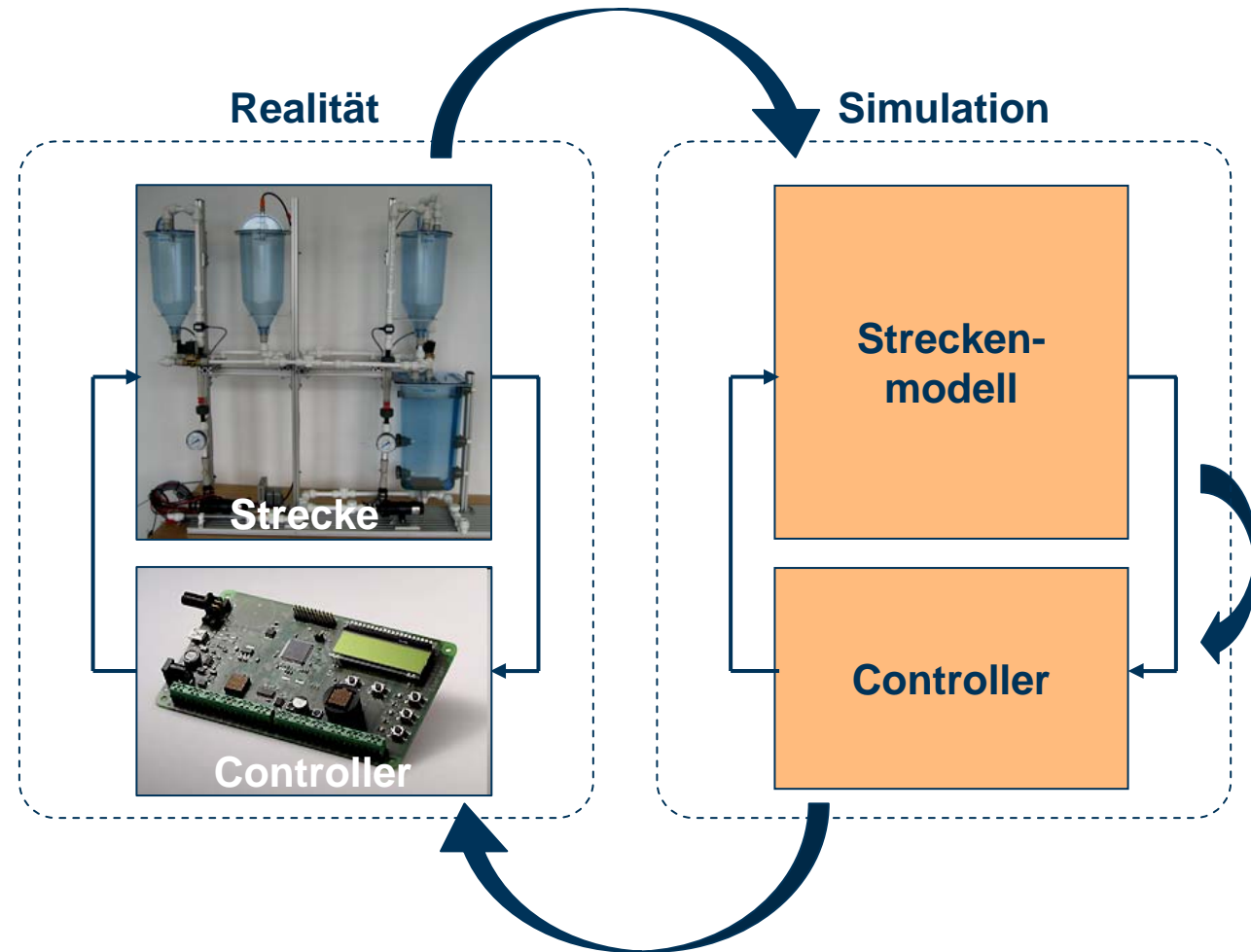
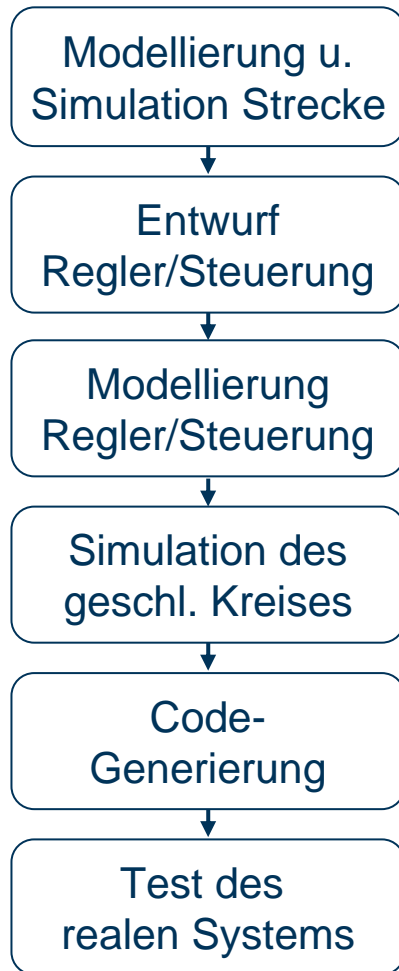
- Theoretischer Entwurf auf Basis eines Modells der Strecke



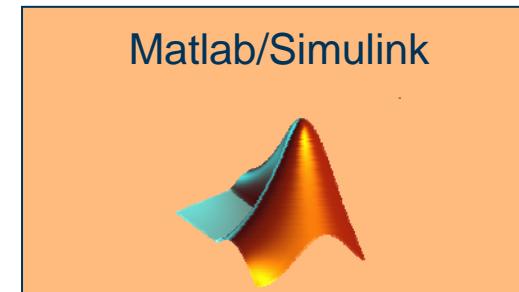
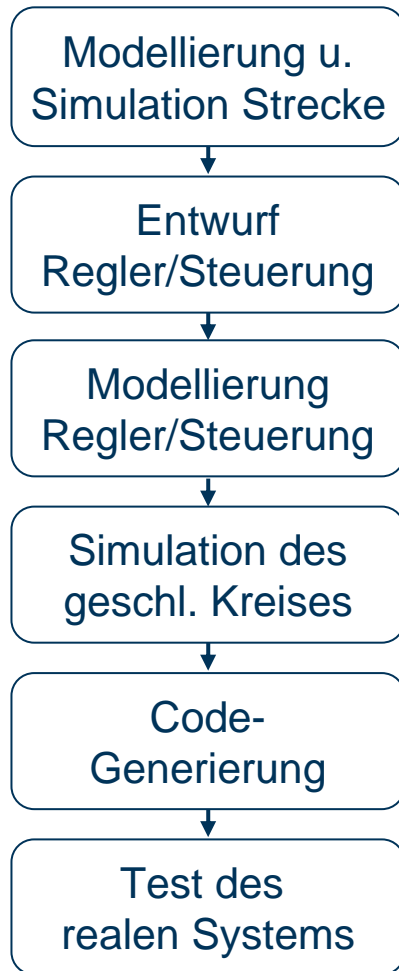
- Implementierung der Regelung auf der Ziel-Hardware



# Rapid Control Prototyping



# Rapid Control Prototyping Produkte



dSpace-System

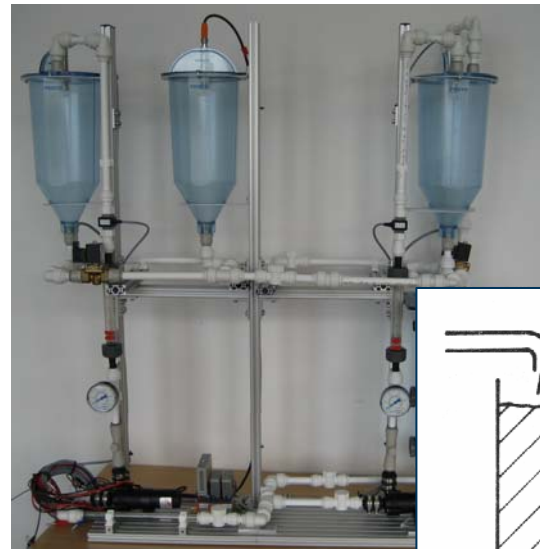
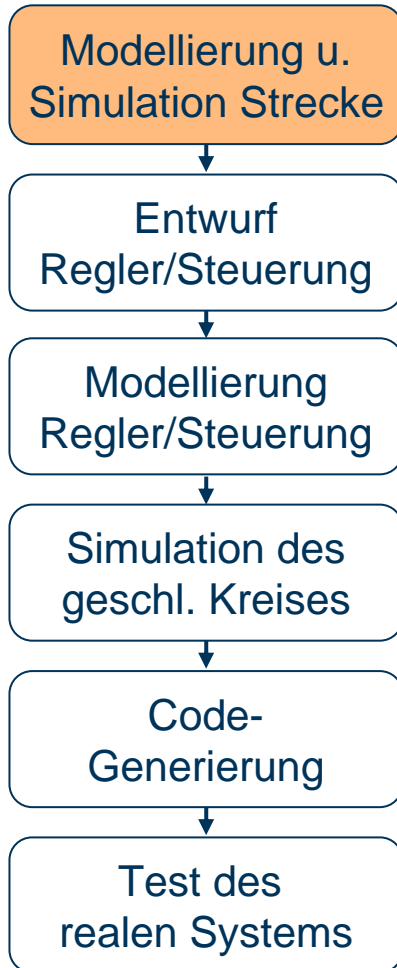


Industrie-PC

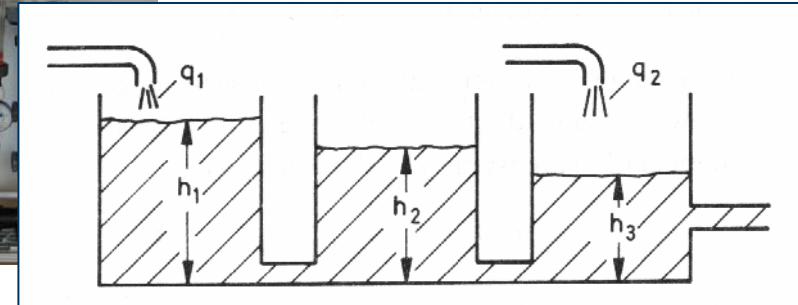


ECUs

# Bsp. Prozessanlage: Modell der Strecke



aus: Föllinger, Regelungstechnik

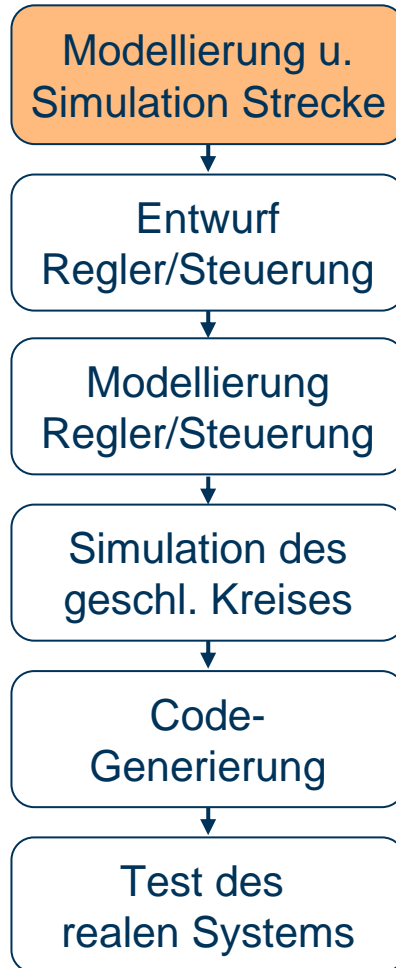


$$\underbrace{\frac{d}{dt} \begin{bmatrix} h_1(t) \\ h_2(t) \\ h_3(t) \end{bmatrix}}_{\underline{\dot{x}}(t)} = \underbrace{\begin{bmatrix} -1/T_1 & +1/T_1 & 0 \\ +1/T_1 & -1/T_1 - 1/T_2 & 1/T_2 \\ 0 & 1/T_2 & -1/T_2 - 1/T_3 \end{bmatrix}}_{\underline{A}} \cdot \underbrace{\begin{bmatrix} h_1(t) \\ h_2(t) \\ h_3(t) \end{bmatrix}}_{\underline{x}(t)} + \underbrace{\begin{bmatrix} q & 0 \\ 0 & 0 \\ 0 & q \end{bmatrix}}_{\underline{B}} \cdot \underbrace{\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}}_{\underline{u}(t)}$$

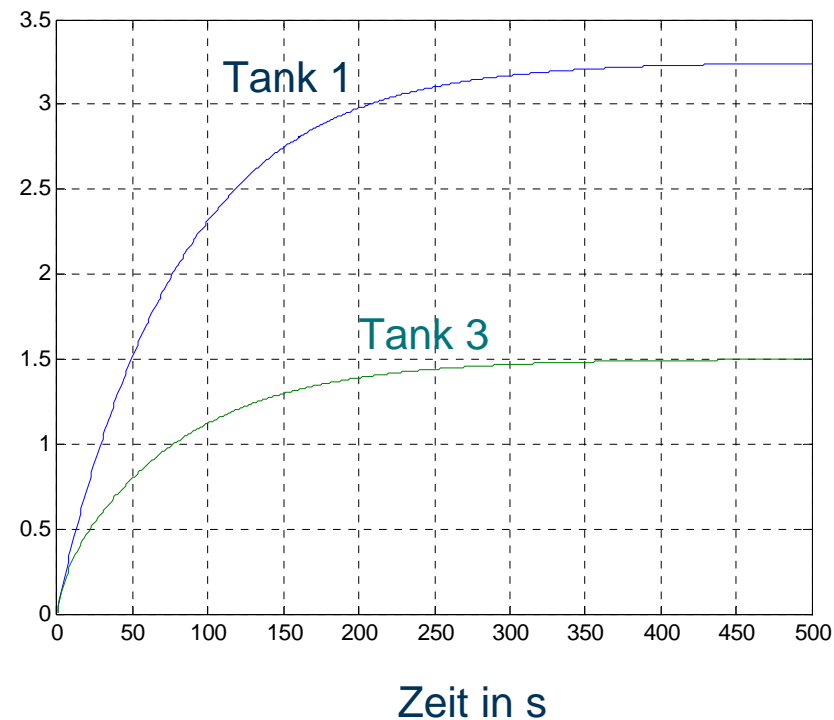
$$\underline{y}(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\underline{C}} \cdot \underline{x}(t)$$

Systembeschreibung im Zustandsraum

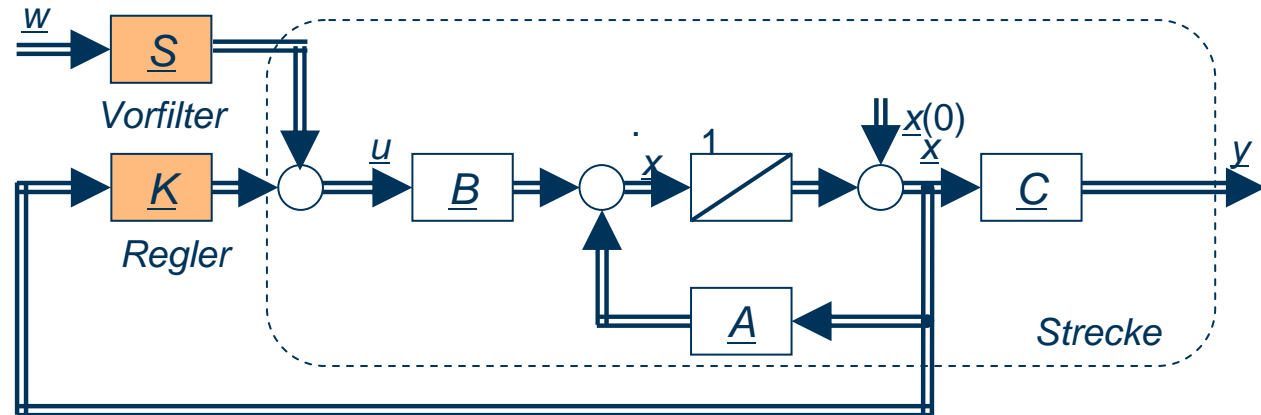
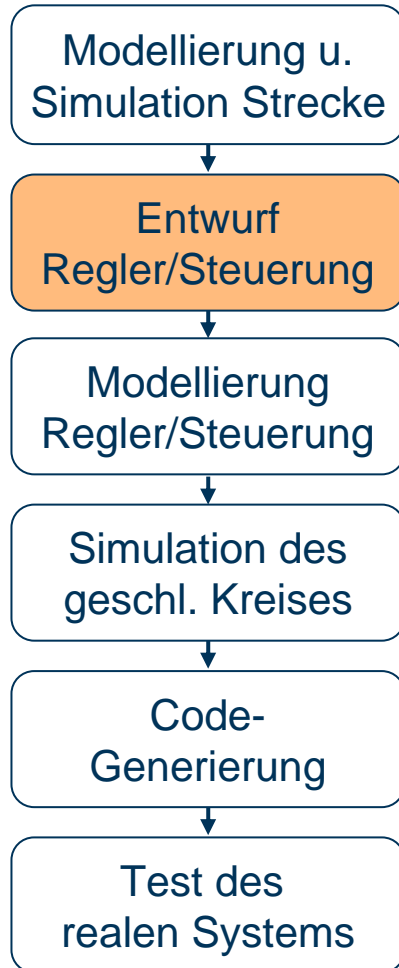
# Simulation der Strecke



Sprungantwort



# Reglerentwurf



z.B. mit Matlab:

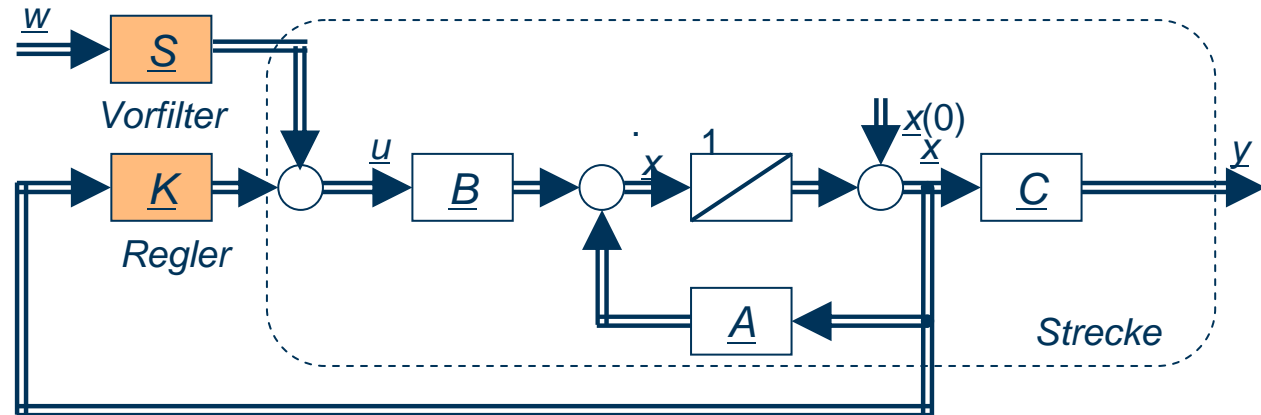
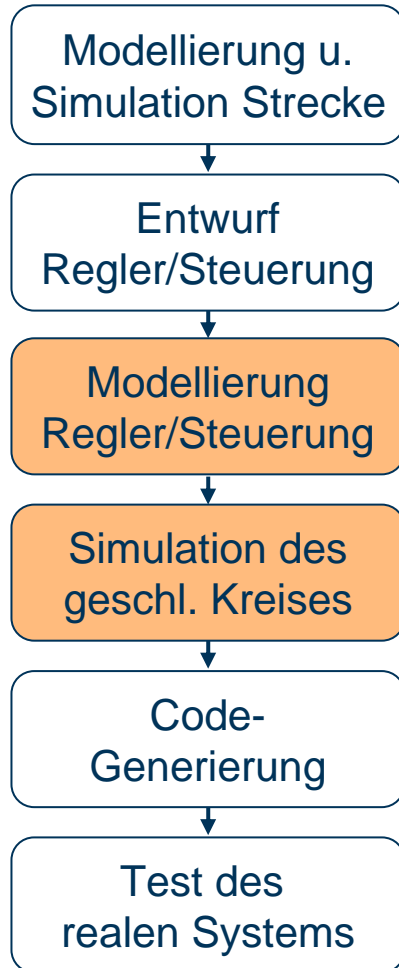
```
% Entwurf des Riccati-Reglers:
```

```
K = -lqr(sys, Q, R);
```

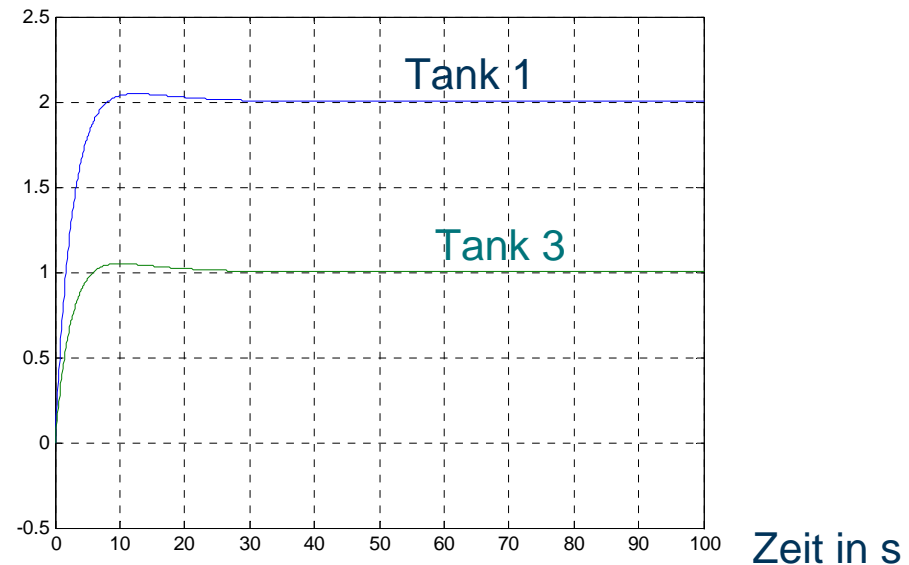
```
% Entwurf des Vorfilters:
```

```
S = -inv(C*inv(A+B*K)*B);
```

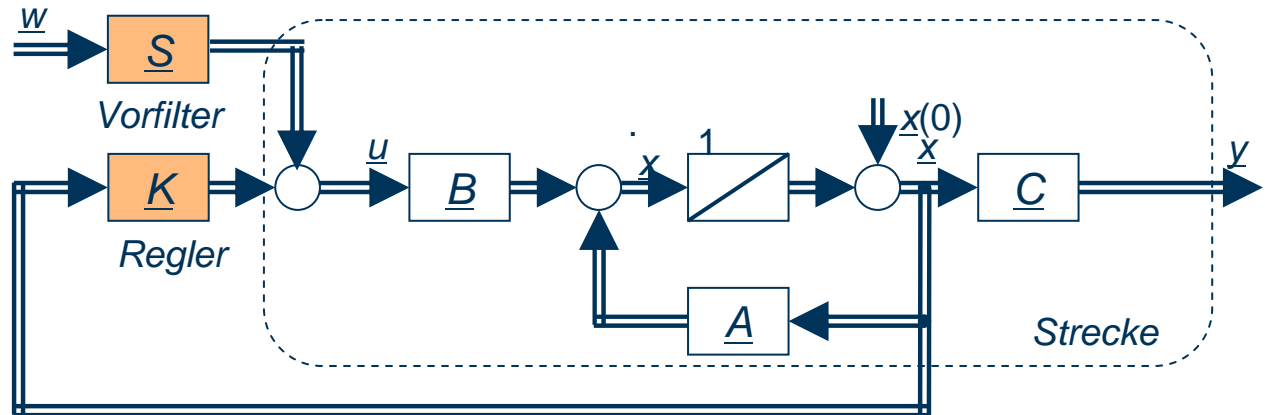
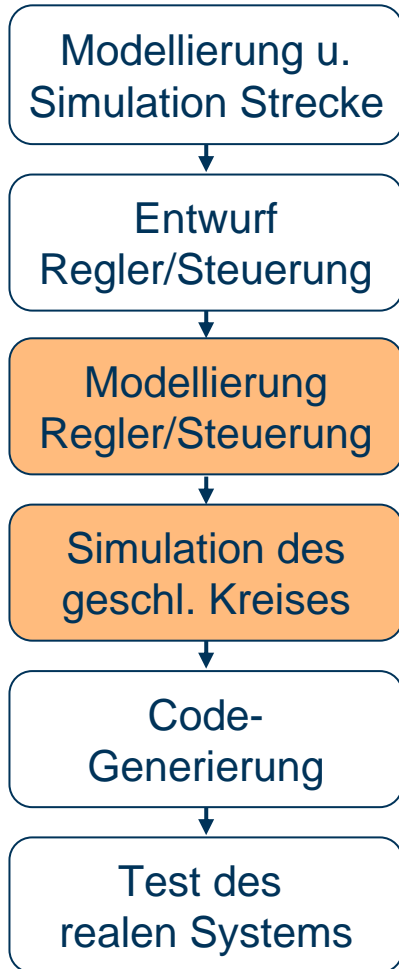
# Geschlossener Regelkreis



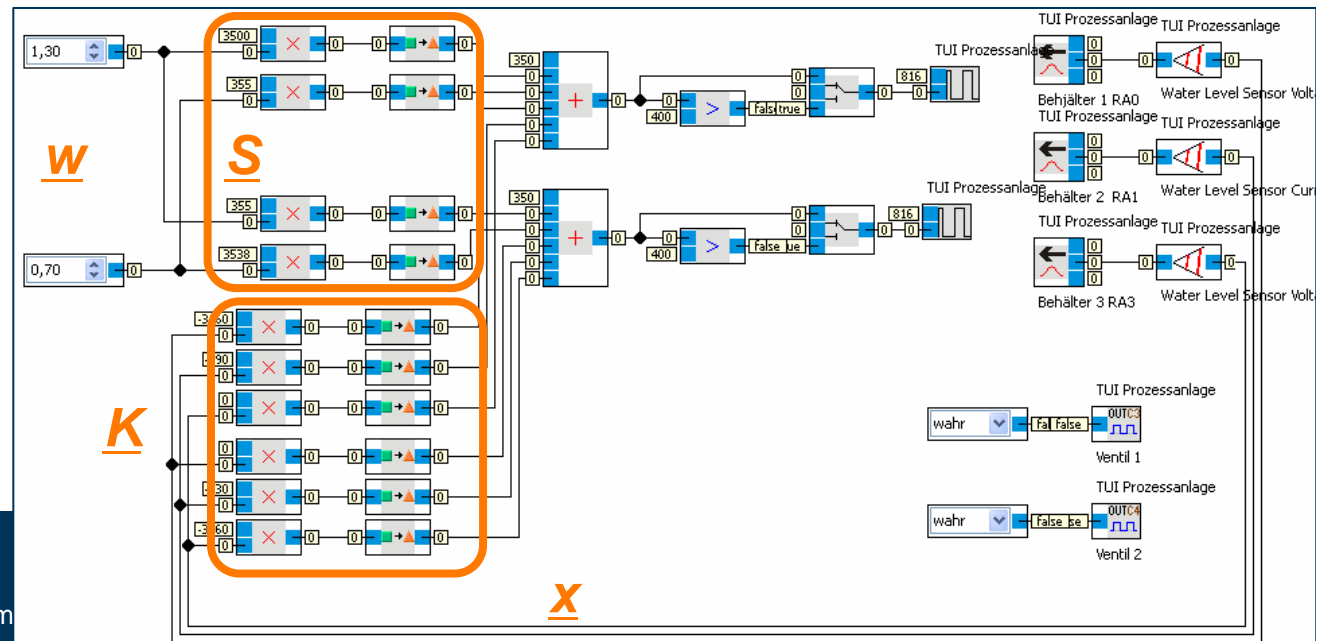
**Sprungantwort**



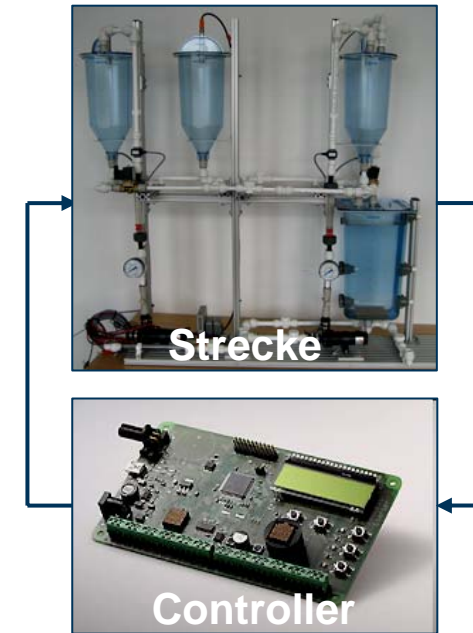
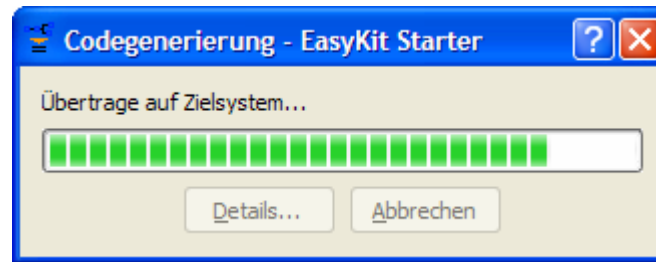
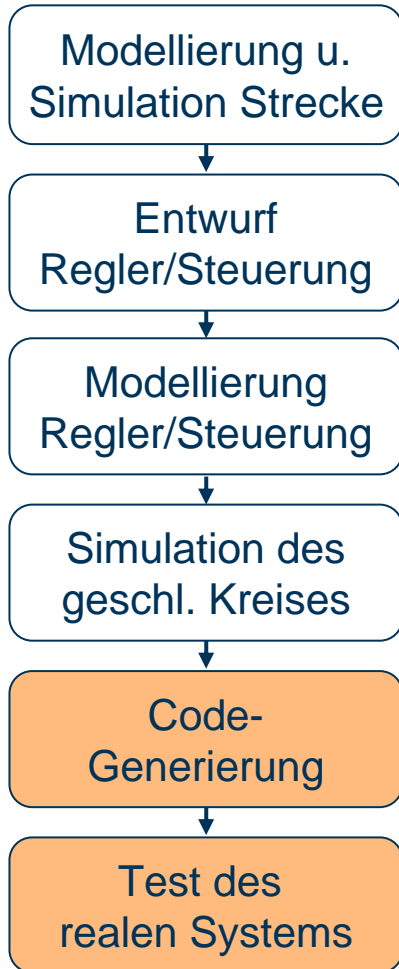
# Geschlossener Regelkreis



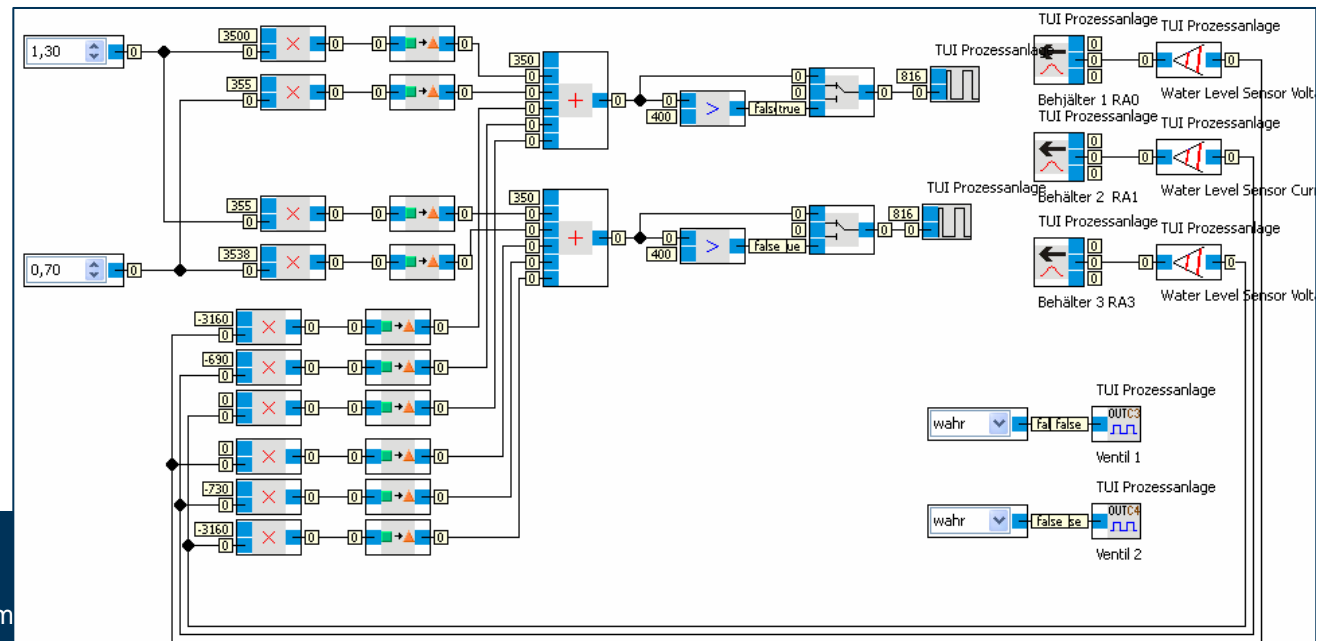
## In EasyLab:



# Code-Generierung und Test

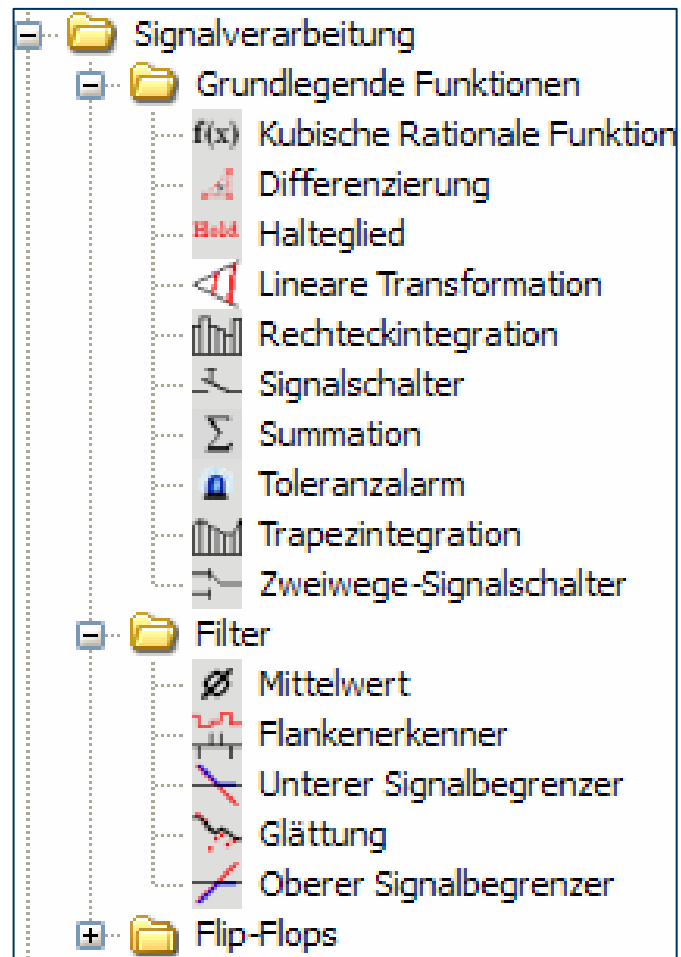


Live-Debugging

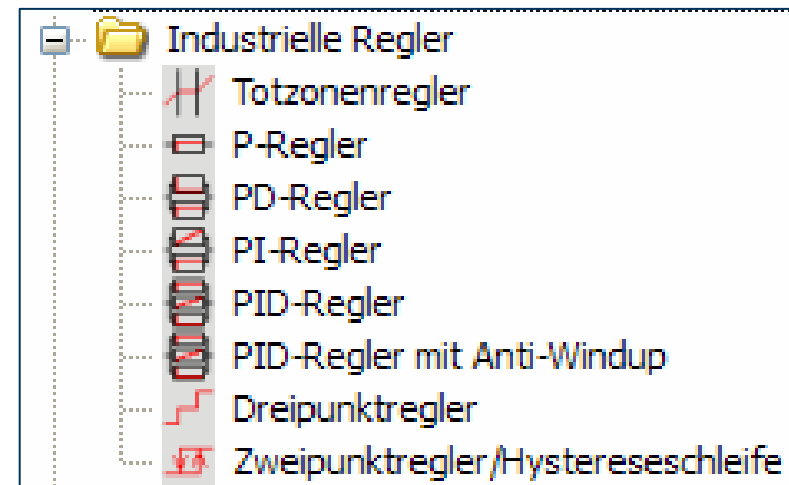


# Funktionsbausteine in EasyLab

## Signalverarbeitung:



## Regelung:



# Zusammenfassung

- Rapid Control Prototyping wird für kleine Mikrocontroller-Plattformen möglich
- Automatische Codegenerierung erhöht Zuverlässigkeit
- Regelentwurf ist plattformunabhängig

## Anwendungen in der universitären Ausbildung:

- Demonstration von Anwendungen „live“ in der Vorlesung möglich
- Für Praktika auch im PC-Pool einsetzbar
- Grundlage für Projektarbeiten (z.B. Bachelor-Arbeiten), in denen Einarbeitung (z.B. in C-Programmierung) vermieden werden kann