Kybernetik

Systemidentifikation

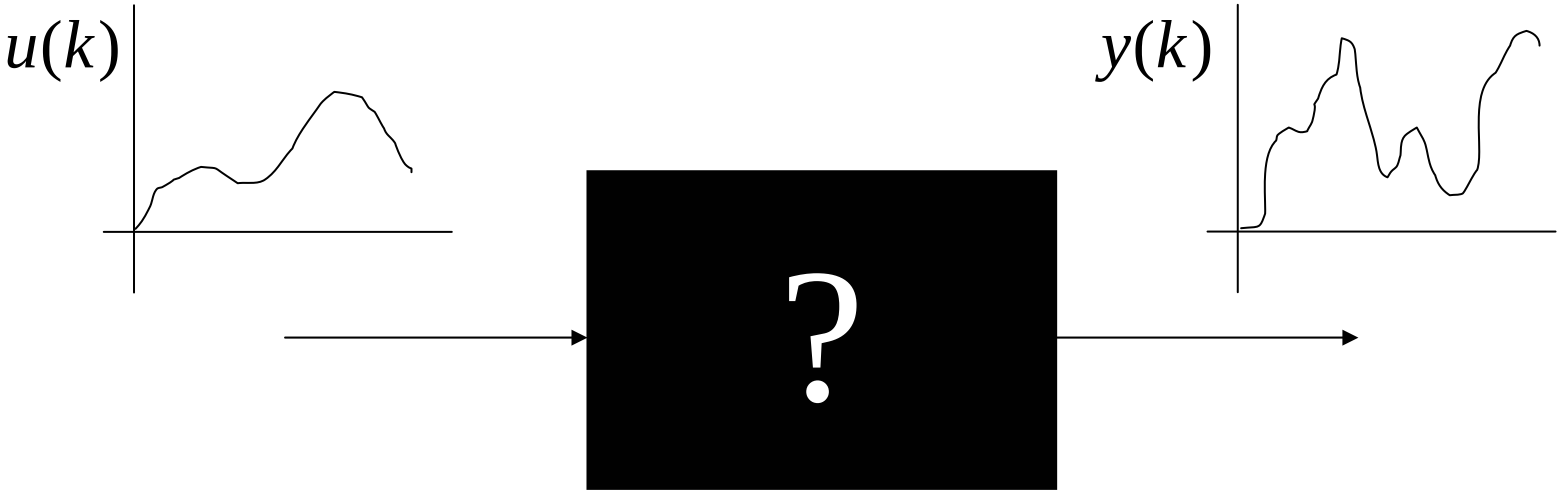
Mohamed Oubbati
Institut für Neuroinformatik

Tel.: (+49) 731 / 50 24153
mohamed.oubbati@uni-ulm.de

12. 06. 2012

Was ist Systemidentifikation?

Der Begriff Systemidentifikation beschreibt Methoden und Algorithmen, die aus gemessenen Ein-/Ausgangsdaten vom System, ein empirisches Modell des Systems erzeugen.



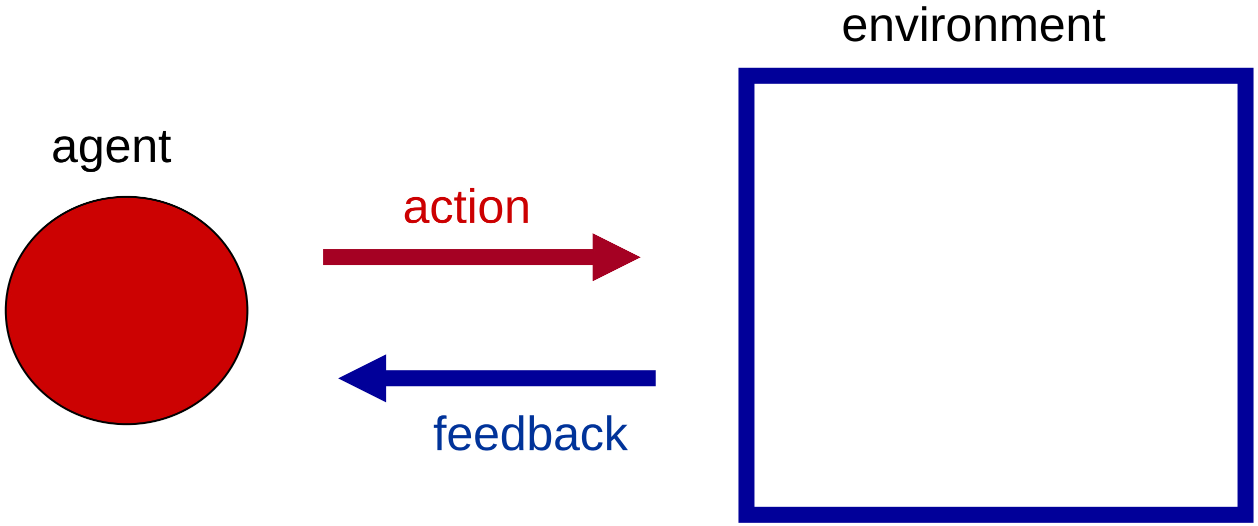
Input/output $u(n)$ /$y(n)$ data ----------→ Systemidentifikation ----------→ empirisches Modell

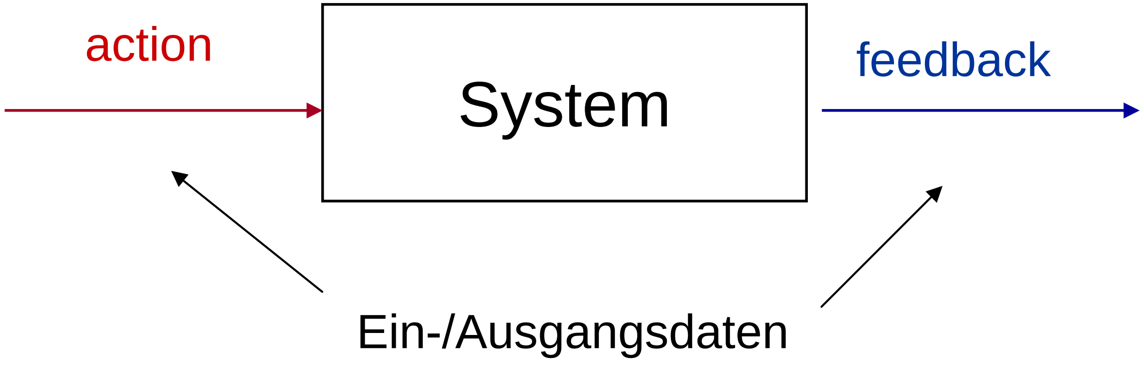
Etappen der Systemidentifikation

Etappen der Systemidentifikation

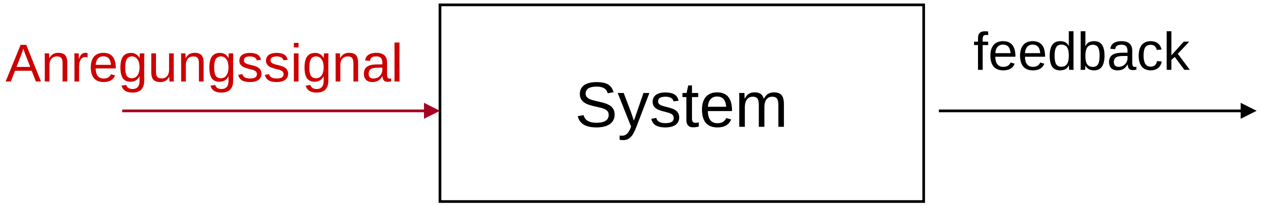
1. Ein-/Ausgangsdaten vom System messen.
2. Modellstruktur auswählen (AR, ARMA, …).
3. Parameter des Modells anpassen.
4. Das gefundene Modell validieren (prüfen).

1. Ein-/Ausgangsdaten vom System messen





1. Ein-/Ausgangsdaten vom System messen



Anregungssignale, mit denen das System (agent-environment) angeregt wird, spielen eine große Rolle bei dem Identifikationsverfahren. Es ist dabei wichtig ein möglichst breites Frequenzspektrum abzudecken.

1. Ein-/Ausgangsdaten vom System messen

Anregungssignale zur Systemidentifikation

Dies kann mithilfe verschiedener Eingangssignale produziert:

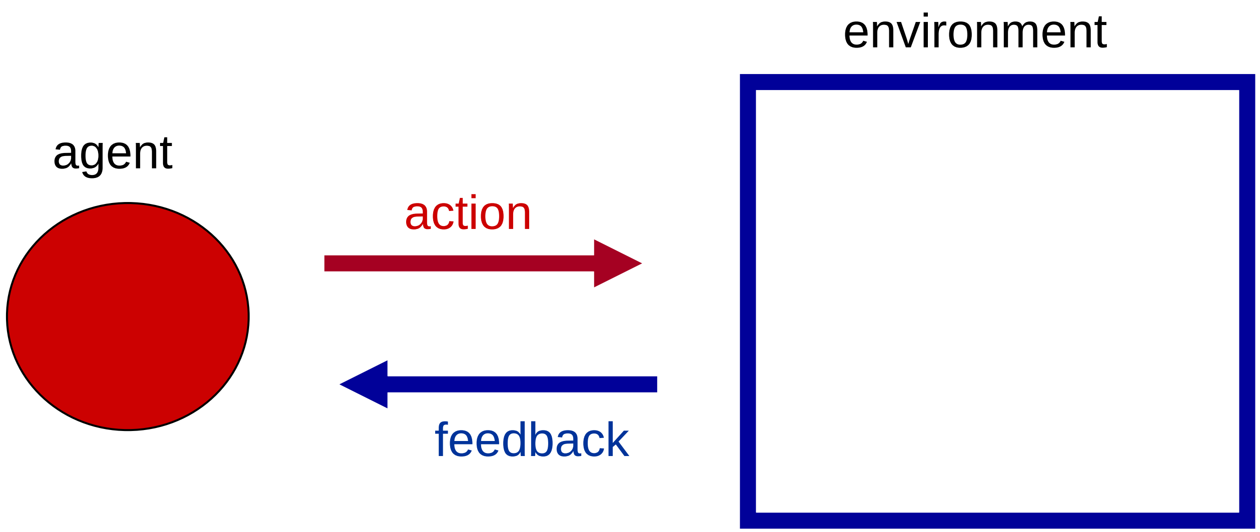
- Impuls: eher für die theoretische Analyse hilfreich.

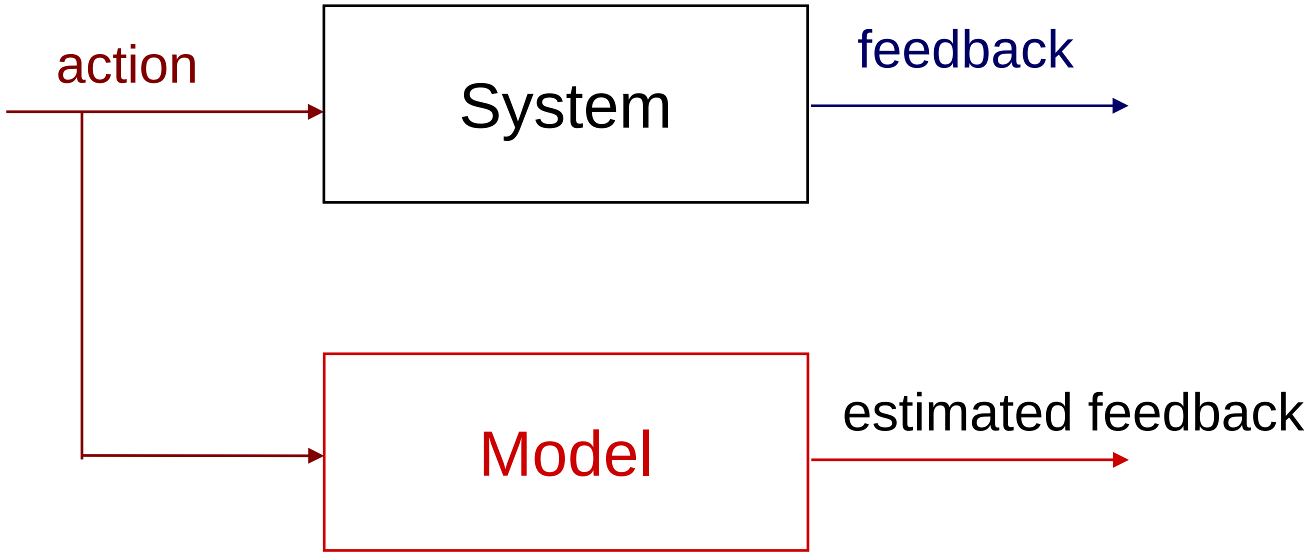
- Sinus-Sweep: Sinusschwingung mit ständig ansteigender Frequenz.

- Rauschen: können als regelmäßige Sprünge verschiedener Amplitude erstellt werden.

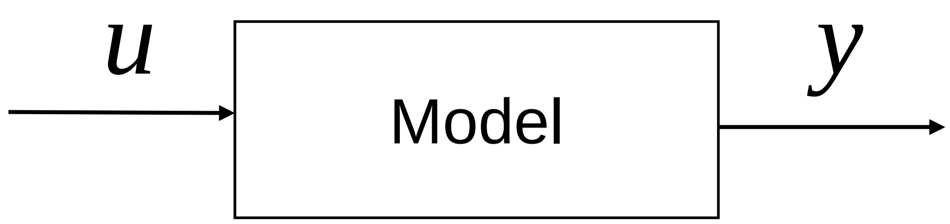
- PRBS (Pseudo Random Binary Signal): können als eine Folge von Rechteckimpulsen unterschiedlicher Länge erstellt werden.

2. Eine Modellstruktur auswählen





2. Eine Modellstruktur auswählen



• AR-Modell

Bei *AR-Modellen* benutzt man eine spezielle Regressionsgleichung die *Autoregressionsgleichung* (kurz: AR).

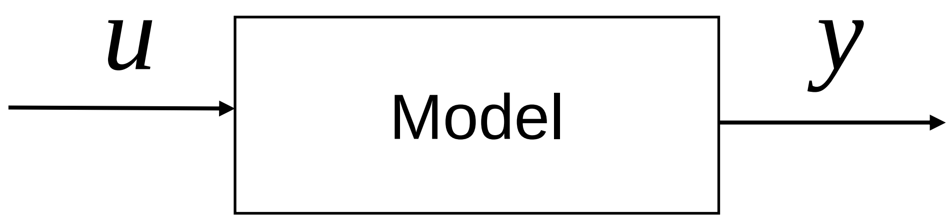
$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n)$$

• MA-Modell

Das *MA-Modell* “Moving-Average” (kurz: MA) beschreibt die Abhängigkeitsbeziehungen der Inputs wie folgt:

$$y(k) = u(k) + b_1 u(k-1) + \ldots b_m u(k-m)$$

2. Eine Modellstruktur auswählen



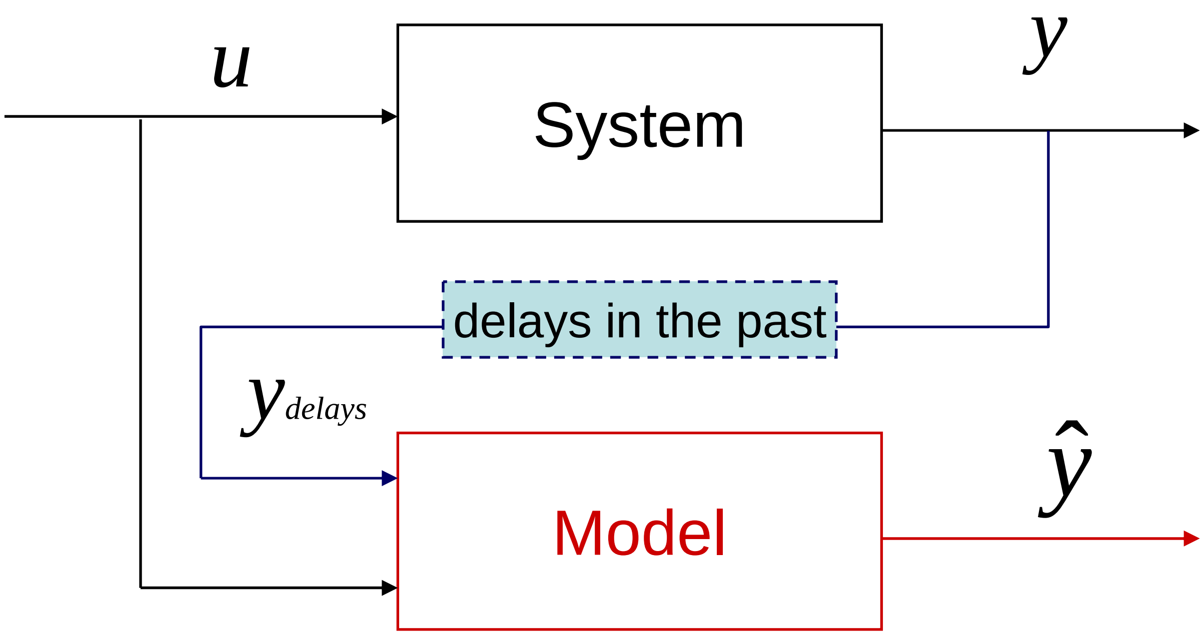
• ARMA-Modell

AR-Modelle und *MA-Modelle* verknüpft ergeben eine sehr nützliche Klasse von Zeitreihenmodellen, die *ARMA-Modelle*:

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n)$$
$$+ u(k) + b_1 u(k-1) + \ldots b_m u(k-m)$$

2. Eine Modellstruktur auswählen

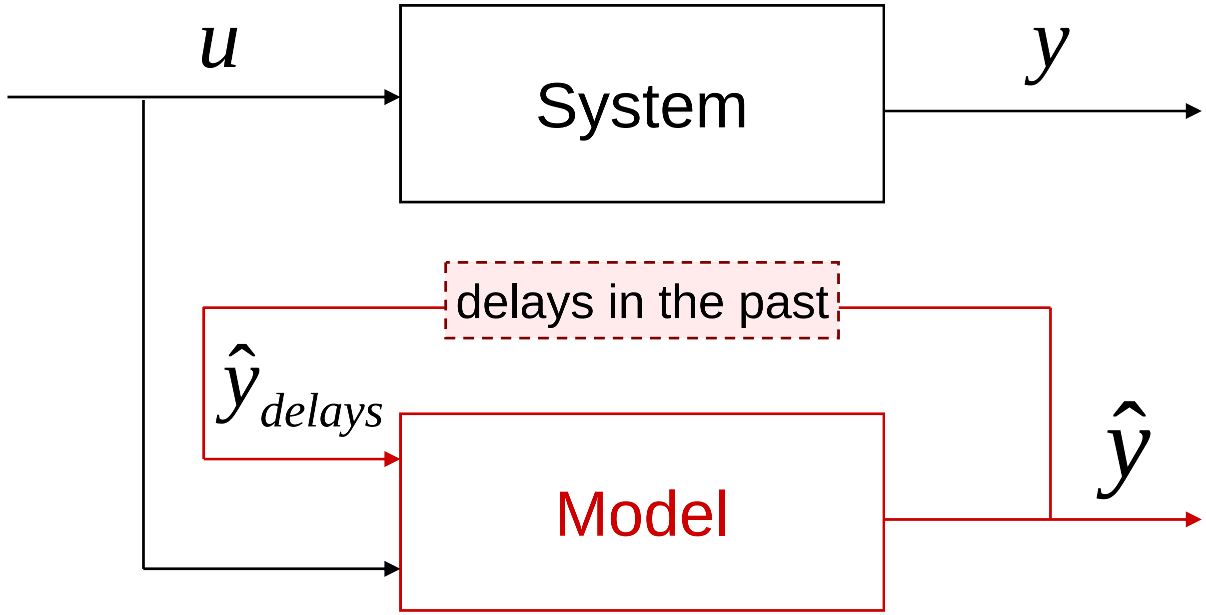
• Series-Parallel model: needs actual input and past output of the system as input to the model.



$$\hat{y}(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n) + u(k)$$

2. Eine Modellstruktur auswählen

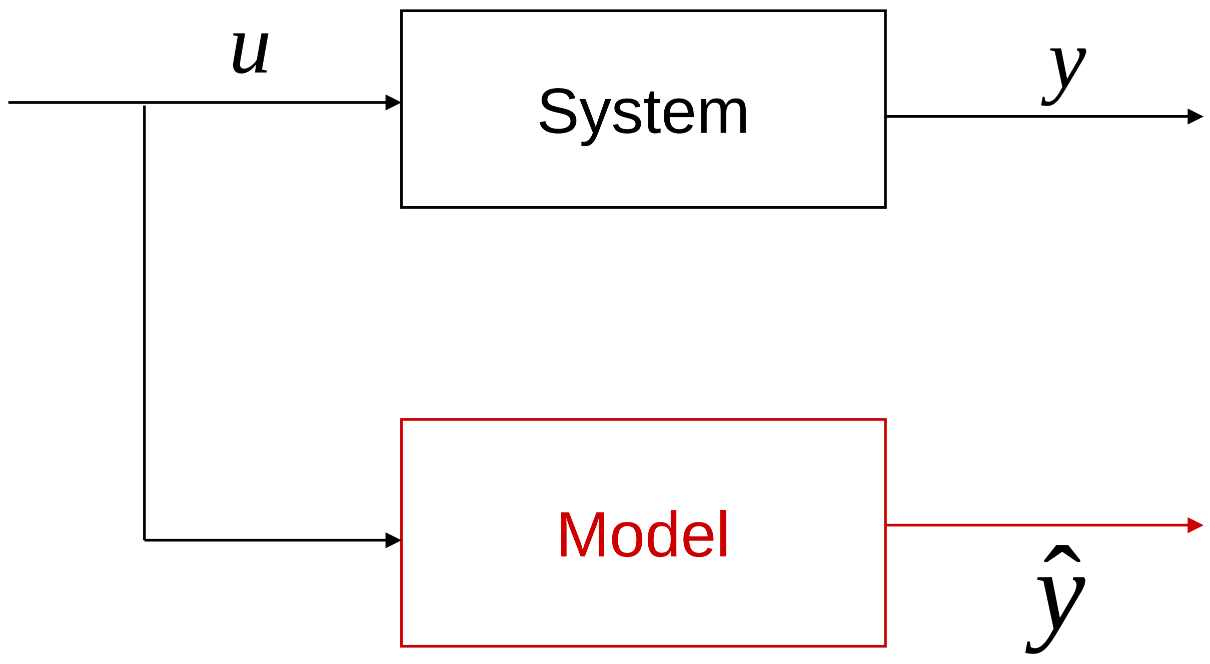
- Parallel-Parallel model: needs actual input and its own past outputs as the inputs.



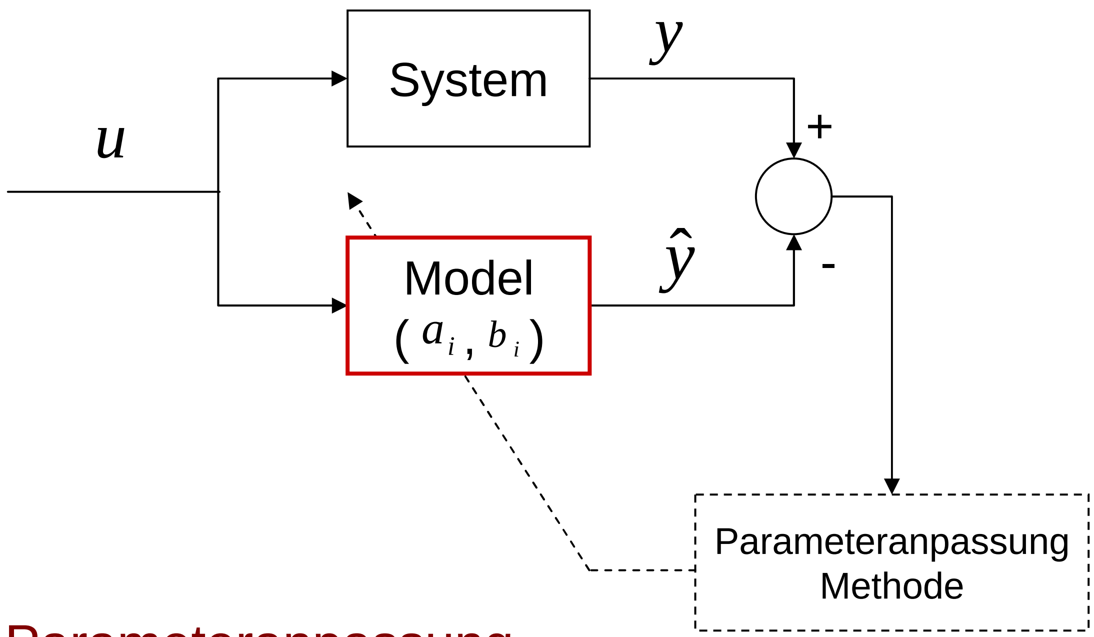
$$\hat{y}(k) = a_1\hat{y}(k-1) + a_2\hat{y}(k-2) + \ldots + a_n\hat{y}(k-n) + u(k)$$

3. Parameter des Modells anpassen

Ist eine Modellstruktur gefunden, so muss noch eine Parameteranpassung vorgenommen werden. Dies wird erreicht, sodass die Differenz zwischen dem realen System und dem ausgewählten Modell so klein wie möglich sein soll.



3. Parameter des Modells anpassen



a_i , b_i ?

sodass

$\hat{y} \approx y$

Methoden zur Parameteranpassung

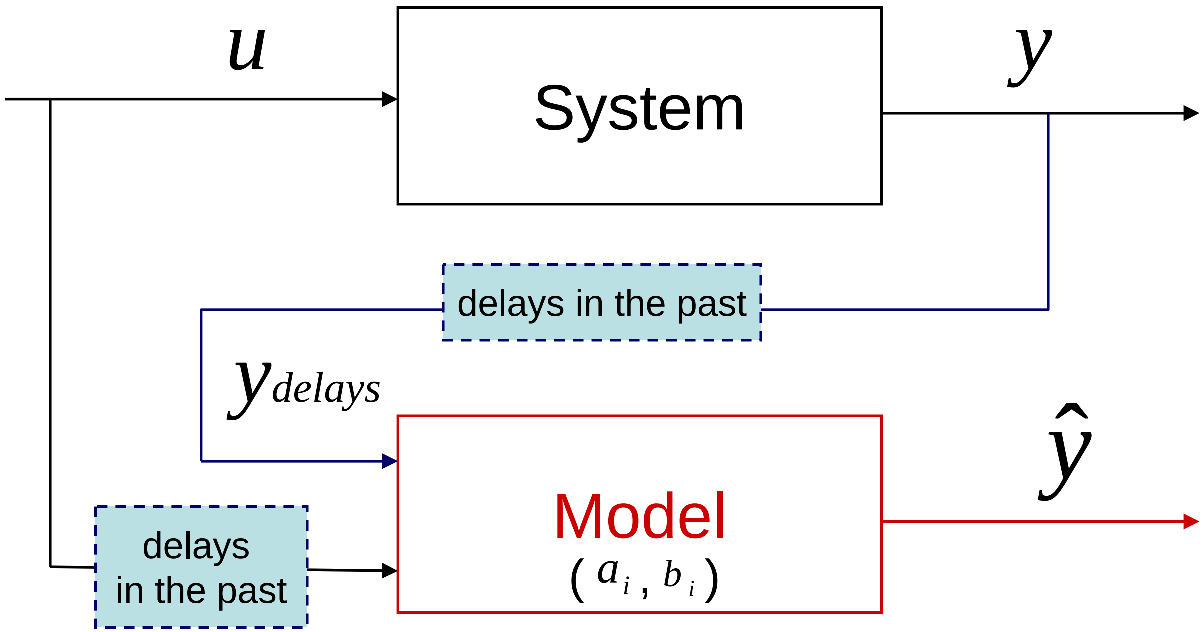
- Least Squares (LS)
- Recursive Least Squares (RLS)
- Prediction Error Method (PEM)
- Recursive Prediction Error Method (RPE)
- Artificial Neural Networks
- …

Methode der kleinsten Quadrate

(Least Squares Method)

Methode der kleinsten Quadrate

Die Modellstruktur des Systems wird als ARMA-Modell in Serie-Parallel Form ausgewählt.



$$\hat{y}(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n)$$
$$+ u(k) + b_1 u(k-1) + \ldots b_m u(k-m)$$

Methode der kleinsten Quadrate

- Messdaten

$$U^T = \left(u(k-1), u(k-2), \ldots, u(k-m)\right)$$

$$Y^T = \left(y(k-1), y(k-2), \ldots, y(k-n)\right)$$

$$k = 1, 2, \ldots, N$$

$$\varphi(k) = \begin{pmatrix} Y \\ U \end{pmatrix}$$

- Parameter

$$a^T = \left(a_1, a_2, \ldots, a_n\right)$$

$$b^T = \left(b_1, b_2, \ldots, b_m\right)$$

$$\rightarrow \qquad \theta = \begin{pmatrix} a \\ b \end{pmatrix}$$

Unbekannt!

- Die Vorhersage des Modells im Zeitpunkt k

$$\hat{y}(k) = \varphi^T(k)\,\theta \qquad k = 1, 2, \ldots, N$$

Methode der kleinsten Quadrate

• Die Differenz zwischen den gemessenen Ausgangswerten $y(k)$ und den Ausgangswerten des Modells $\hat{y}(k)$ ist

$$\xi(k,\theta) = y(k) - \hat{y}(k) = y(k) - \varphi^T(k)\theta$$

• Die Aufgabe der systemidentifikation ist es, der Vektor $\hat{\theta}$ (eine Approximation von θ) zu finden, sodass

$$\xi(k,\hat{\theta}) = \hat{y}(k) - \varphi^T(k)\hat{\theta} \approx 0$$

Methode der kleinsten Quadrate

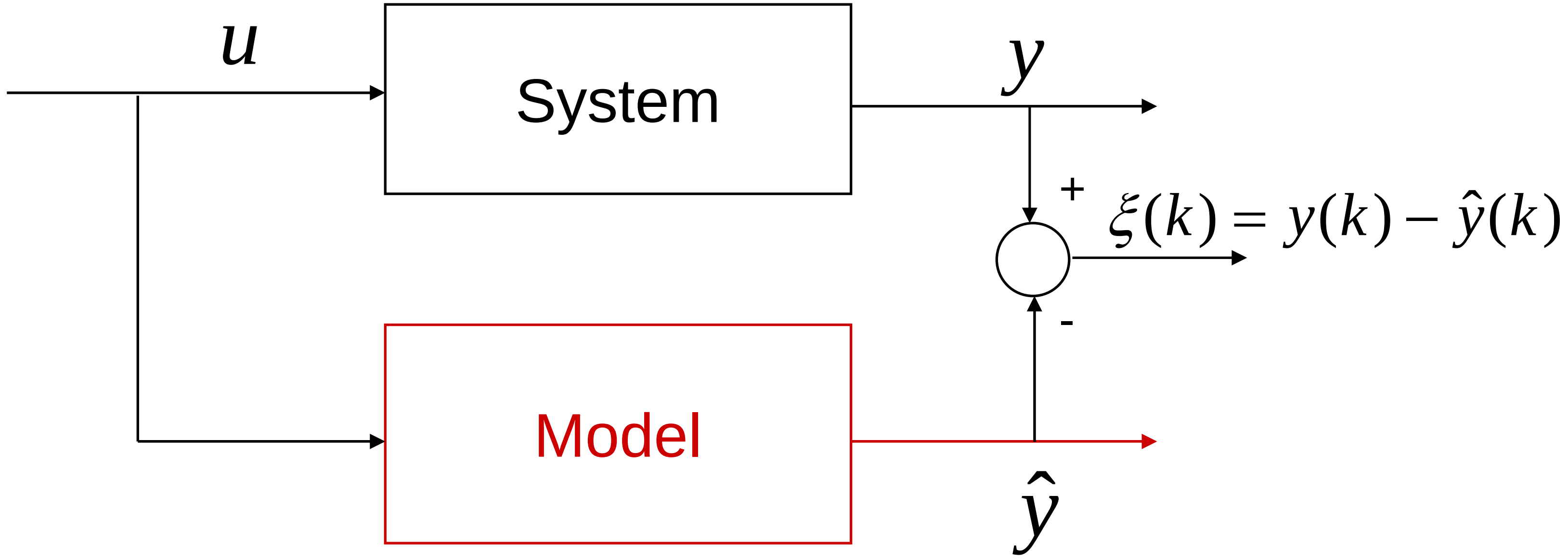
Das Least-Square Algorithm versucht eine Approximation $\hat{\theta}$ zu finden, durch die Minimierung der Funktion $V_N(\theta)$

$$V_N(\theta) = \frac{1}{2}\sum_{k=1}^{N} \xi^2(k,\theta)$$

$$\hat{\theta} = \arg\min_{\theta} V_N(\theta)$$

$$\rightarrow \qquad \hat{\theta} = \left[\sum_{k=1}^{N} \varphi(k)\varphi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \varphi(k)y(k)\right]$$

4. Das gefundene Modell validieren (prüfen).



Following statistics are often used for model validation

- mean error: $m_N = \frac{1}{N} \sum_{K=1}^{N} \xi(k)$
- variance: $Var_N = \frac{1}{N} \sum_{k=1}^{N} \left(\xi(k) - m_N\right)^2$
- mean square error: $MSE_N = \frac{1}{N} \sum_{K=1}^{N} \xi^2(k)$