

Systemidentifikation und Regelung in der Medizin

5. Vorlesung Prediction Error Method

Sommersemester 2020

2. Juni 2020

Thomas Schauer

Technische Universität Berlin
Fachgebiet Regelungssysteme

Literaturempfehlung: 1) E. Ikonen und K. Najim, Advanced Process Identification and Control, Marcel Dekker, Inc., 2002;
2) David T. Westwick, Robert E. Kearney, Identification of Nonlinear Physiological Systems, Wiley-IEEE Press, 2003 (Kapitel 8)

Summary on last page!

5. Parameterschätzung / Prediction Error Method (PEM)

- Bestimmung von Modellparametern aus Messdaten
- Least Squares nur anwendbar für lineare statische Abbildungen und einige dynamische Systeme (ARX) (Systemausgang ist linear abhängig von den Parametern sowie Parametervektor und Regressionsvektor sind unabhängig voneinander)
- Für nichtlineare Modelle: iterative Optimierungsverfahren (Iterative Least Squares Methods)
 - wiederholte Anwendung des gesamten Datensatzes zur Parameterschätzung
⇒ Batch-Adaptation

Gütefunktional basierend auf der Einschritt vorausprädiktion:

$$J(\Theta) = \sum_{k=1}^K \frac{1}{2} (y(k) - \hat{y}(k, \Theta | k-1))^2 \quad \text{identisch zu least square} \quad (1)$$

Matlab von 1

- Anfangswert: $\hat{\Theta}(1)$
- In jedem Updateschritt l (Achtung: l ist nicht der Abtastindex k) Bestimmung einer Richtung $\mathbf{u}(l)$ und Schrittweite $\eta(l)$
- Update der Parameter mittels

$$\hat{\Theta}(l+1) = \hat{\Theta}(l) + \eta(l)\mathbf{u}(l) \quad (2)$$

Wichtig: In $J \rightarrow \hat{y}(k, \hat{\Theta}(l))$ ist immer eine Funktion der aktuellen Parameterschätzung $\hat{\Theta}(l)$

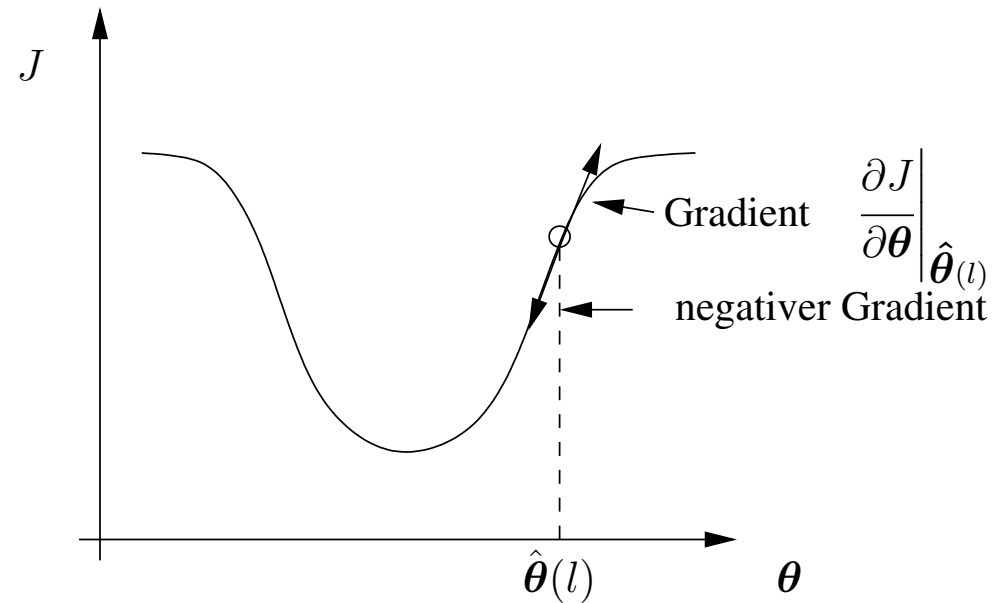
Ziel: $\mathbf{u}(l)$, $\eta(l)$ so wählen, dass

choose $\mathbf{u}(l)$ and $\eta(l)$ so that we minimize J through the parameter

$$\hat{\Theta}(l) \rightarrow \arg \min_{\Theta} J(\Theta) \text{ für } l \rightarrow \infty$$

- lokale Verfahren: suche nach lokalen Minimum

5.1 Verfahren 1. Ordnung



$\hat{\Theta}(l)$: aktuelle Schätzung des Parametervektors

Taylorentwicklung von $J(\Theta)$ bis zur 1. Ordnung:

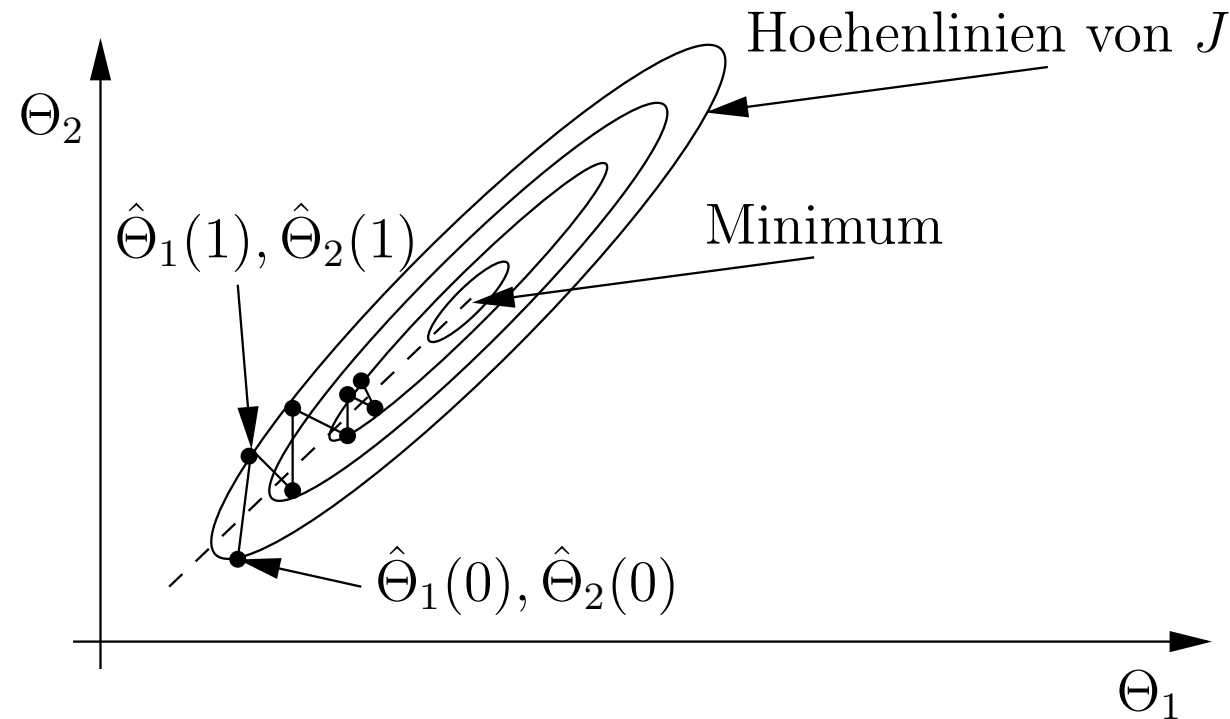
$$J(\Theta) \simeq J(\hat{\Theta}(l)) + \sum_{i=1}^I \left[\frac{\partial J}{\partial \Theta_i} \right] \bigg|_{\Theta = \hat{\Theta}(l)} \tilde{\Theta}_i, \quad \tilde{\Theta} = \Theta - \hat{\Theta}(l)$$

mit

$$J(\Theta) = \sum_{k=1}^K \frac{1}{2} (y(k) - \hat{y}(k, \Theta | k-1))^2 \quad \text{ableiten und zu null setzen}$$

$$\left[\frac{\partial J}{\partial \Theta_i} \right] \Big|_{\Theta = \hat{\Theta}(l)} = - \sum_{k=1}^K [y(k) - \hat{y}(k, \hat{\Theta}(l))] \frac{\partial \hat{y}(k)}{\partial \Theta_i} \Big|_{\Theta = \hat{\Theta}(l)}$$

$$\hat{\Theta}(l+1) = \hat{\Theta}(l) - \eta \left[\frac{\partial J}{\partial \Theta} \right] \Big|_{\Theta = \hat{\Theta}(l)}$$



Stellt das Gütefunktional ein langgestrecktes Tal dar, so kann es bei einem Optimierungsverfahren 1. Ordnung zu Oszillationen kommen, der denen die Lösung immer von einer Seite des Tales zur anderen Seite des Tales springt.

Verfahren 1. Ordnung - Eigenschaften:

- Verfahren 1. Ordnung (wie „steilster Abstieg“) sind **einfach** zu realisieren
- **schlechte (langsame) Konvergenz**
- Siehe Beispiel: langes Tal
 - Oszillationen
 - kleine Schrittweite damit man das Tal nicht verlässt
 - Richtung immer orthogonal zu den Höhenlinien

5.2 Verfahren 2. Ordnung

- Ausnutzung der lokalen Krümmung der Gütefunktionalen (Fläche)
- Beispiel langes Tal:
 - Krümmung am größten orthogonal zum Tal
 - Krümmung am kleinsten entlang des Tals
 - optimale Suchrichtung
- Annahme: lokales Aussehen des Gütefunktionalen entspricht einer quadratischen Funktion der Parameter

Taylorreihenentwicklung

$$\begin{aligned}
J(\Theta) &= J(\hat{\Theta}(l)) + \sum_{i=1}^I \frac{\partial J}{\partial \Theta_i} \bigg|_{\Theta=\hat{\Theta}(l)} \tilde{\Theta}_i + \sum_{i=1, i^*=1}^I \frac{\partial^2 J}{\partial \Theta_i \partial \Theta_{i^*}} \bigg|_{\Theta=\hat{\Theta}(l)} \tilde{\Theta}_i \tilde{\Theta}_{i^*} + \dots \\
&\simeq J(\hat{\Theta}(l)) - \mathbf{b}^T \tilde{\Theta} + \frac{1}{2} \tilde{\Theta}^T \mathbf{H} \tilde{\Theta}, \quad \tilde{\Theta} = \Theta - \hat{\Theta}(l)
\end{aligned}$$

mit $\mathbf{b} = - \frac{\partial J}{\partial \Theta} \bigg|_{\Theta=\hat{\Theta}(l)}$ (negativer Gradient)

\mathbf{H} - Hession Matrix (Hessische Matrix):

$$\frac{\partial^2 J}{\partial \Theta_i \partial \Theta_{i^*}} \bigg|_{\Theta=\hat{\Theta}(l)} = h_{i,i^*}$$

Hessian matrix: Describes the local curvature of a function of many variables
 - Diagonal element is 2x derivative of a variable

Nullsetzen der Ableitung von J :

$$\frac{\partial J}{\partial \tilde{\Theta}} = -\mathbf{b}^T + \tilde{\Theta}^T \mathbf{H}^T = 0 \quad \Rightarrow \quad \tilde{\Theta} = \mathbf{H}^{-1} \mathbf{b}$$

- Wahre Funktion ist nicht quadratisch; daher verwendet man $\mathbf{H}^{-1} \mathbf{b}$ als Richtung (meistens mit Schrittweite versehen) und macht dann iterativ weiter.

5.2.1 Newton Verfahren

$$\hat{\Theta}(l+1) = \hat{\Theta}(l) + \eta H^{-1}(\hat{\Theta}(l)) b(\hat{\Theta}(l))$$

η : Lernrate

- erfolgreicher Schritt: $J(\hat{\Theta}(l+1)) < J(\hat{\Theta}(l)) \rightarrow \eta$ vergrössern (\approx Faktor 10)
- nicht erfolgreicher Schritt: $J(\hat{\Theta}(l+1)) \geq J(\hat{\Theta}(l)) \rightarrow$ letzten Schritt verwerfen und η verkleinern (\approx Faktor 10)

Problem: Berechnen von $H^{-1}(\hat{\Theta}(l))$

H : y zweimal abgeleitet

5.2.2 Approximative Verfahren

K - Anzahl der Messungen, I - Anzahl der Parameter, $e(k)$ Prädiktionsfehler für Abtastzeitpunkt k

$$e(k, \hat{\Theta}(l)) = y(k) - \hat{y}(k, \hat{\Theta}(l)), \quad \mathbf{E}(\hat{\Theta}(l)) = \begin{bmatrix} e(1, \hat{\Theta}(l)) & \dots & e(K, \hat{\Theta}(l)) \end{bmatrix}^T$$

Error Matrix

Gütefunktional

$$J(\hat{\Theta}(l)) = \frac{1}{2} \mathbf{E}(\hat{\Theta}(l))^T \mathbf{E}(\hat{\Theta}(l))$$

$$\frac{\partial J(\hat{\Theta}(l))}{\partial \hat{\Theta}(l)} = \sum_{k=1}^K e(k, \hat{\Theta}(l)) \frac{\partial e(k, \hat{\Theta}(l))}{\partial \hat{\Theta}(l)} = \mathbf{G}(\hat{\Theta}(l))^T \mathbf{E}(\hat{\Theta}(l))$$

$\mathbf{G}(\hat{\Theta}(l)) \in \mathbb{R}^{K \times I}$: Jakobimatrix im Iterationsschritt l

$$g_{k,i} = \frac{\partial e(k, \hat{\Theta}(l))}{\partial \hat{y}(k, \hat{\Theta}(l))} \frac{\partial \hat{y}(k, \hat{\Theta}(l))}{\partial \hat{\Theta}_i(l)} = - \frac{\partial \hat{y}(k, \hat{\Theta}(l))}{\partial \hat{\Theta}_i(l)}$$

$$\mathbf{G}(\hat{\boldsymbol{\Theta}}(l)) = \begin{bmatrix} -\frac{\partial \hat{y}(1, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\Theta}_1(l)} & \dots & -\frac{\partial \hat{y}(1, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\Theta}_I(l)} \\ \vdots & \ddots & \vdots \\ -\frac{\partial \hat{y}(K, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\Theta}_1(l)} & \dots & -\frac{\partial \hat{y}(K, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\Theta}_I(l)} \end{bmatrix} \quad \text{Jacobi Matrix}$$

$$\begin{aligned} \frac{\partial^2 J((\hat{\boldsymbol{\Theta}}(l)))}{\partial (\hat{\boldsymbol{\Theta}}(l))^2} &= \sum_{k=1}^K \left[\frac{\partial e(k, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\boldsymbol{\Theta}}(l)} \left[\frac{\partial e(k, \hat{\boldsymbol{\Theta}}(l))}{\partial \hat{\boldsymbol{\Theta}}(l)} \right]^T + e(k, \hat{\boldsymbol{\Theta}}(l)) \frac{\partial^2 e(k, \hat{\boldsymbol{\Theta}}(l))}{\partial (\hat{\boldsymbol{\Theta}}(l))^2} \right] \\ &= \mathbf{G}(\hat{\boldsymbol{\Theta}}(l))^T \mathbf{G}(\hat{\boldsymbol{\Theta}}(l)) + \mathbf{S}(\hat{\boldsymbol{\Theta}}(l)) = \mathbf{H}(\hat{\boldsymbol{\Theta}}(l)) \end{aligned}$$

- $\mathbf{G}(\hat{\boldsymbol{\Theta}}(l))^T \mathbf{G}(\hat{\boldsymbol{\Theta}}(l))$: leicht zu bestimmen, da nur erste Ableitungen vorkommen
- $\mathbf{S}(\hat{\boldsymbol{\Theta}}(l))$: schwer zu berechnen, da zweite Ableitungen vorkommen

5.2.2.1 Gauß-Newton-Verfahren

$$\hat{\Theta}(l+1) = \hat{\Theta}(l) - \eta \underbrace{\left[G(\hat{\Theta}(l))^T G(\hat{\Theta}(l)) \right]^{-1}}_{\hat{H}^{-1}(\hat{\Theta}(l))} \underbrace{G(\hat{\Theta}(l))^T E(\hat{\Theta}(l))}_{-b(\hat{\Theta}(l))}$$

η : Schrittweite \rightarrow anpassen ähnlich zum Newton-Verfahren

5.2.2.2 Levenberg-Marquardt-Verfahren

$$\hat{\Theta}(l+1) = \hat{\Theta}(l) - \underbrace{\left[G(\hat{\Theta}(l))^T G(\hat{\Theta}(l)) + \eta I \right]^{-1}}_{\hat{H}^{-1}(\hat{\Theta}(l))} \underbrace{G(\hat{\Theta}(l))^T E(\hat{\Theta}(l))}_{-b(\hat{\Theta}(l))}$$

- $\eta(l)$ erhöhen für schlechten Schritt: $\eta(l+1) = \eta(l)\gamma$, $\gamma > 1$ und

Verwerfen des letzten Schritts! Wie Newton-Verfahren in Seite 11

- $\eta(l)$ erniedrigen für erfolgreichen Schritt: $\eta(l+1) = \eta(l)/\gamma$, $\gamma > 1$, $l = l+1$
- $\eta(l)$ sehr groß: $\hat{\Theta}(l+1) \approx \hat{\Theta}(l) - \frac{1}{\eta} G(\hat{\Theta}(l))^T E(\hat{\Theta}(l))$ (steilster Abstieg)
 η dominiert
 → Verfahren erster Ordnung
- $\eta(l)$ sehr klein: $\hat{\Theta}(l+1) \approx \hat{\Theta}(l) - \left[G(\hat{\Theta}(l))^T G(\hat{\Theta}(l)) \right]^{-1} G(\hat{\Theta}(l))^T E(\hat{\Theta}(l))$
 → Gauß-Newton-Verfahren

5.3 Gradientenberechnung für die PEM

$$J(\hat{\Theta}(l)) = \frac{1}{2} \sum_{k=1}^K \left(y(k) - \hat{y}(k, \hat{\Theta}(l) | k-1) \right)^2$$

$$\frac{\partial J(\hat{\Theta}(l))}{\partial \hat{\Theta}_i(l)} = - \sum_{k=1}^K \left([y(k) - \hat{y}(k, \hat{\Theta}(l) | k-1)] \frac{\partial \hat{y}(k, \hat{\Theta}(l))}{\partial \hat{\Theta}_i(l)} \right)$$

Gesucht: $\frac{\partial \hat{y}(k, \hat{\Theta}(l))}{\partial \hat{\Theta}_i(l)} \longrightarrow$ für lineare dynamische Systeme (OE, ARMAX, BJ)

5.3.1 Beispielhafte Gradientenberechnung für ein OE-Modell

Parametervektor:

$$\Theta = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_A} & b_0 & b_1 & b_2 & \cdots & b_{n_B} \end{bmatrix}^T \quad (3)$$

Prädiktor:

schon vorher hergeleitet

$$\hat{y}(k, \Theta(l)|k-1) = \hat{B}(q^{-1}, l)u(k-d) - \hat{A}_1(q^{-1}, l)\hat{y}(k-1|k-2)$$

Um die Gleichungen kompakter zu halten, wird $|\dots$ im folgenden weggelassen.

Berechnung der Gradienten:

Was ist A und B? Siehe VL 4 Seite 18

Prädiktor ableiten:

$$\frac{\partial \hat{y}(k|k-1)}{\partial \hat{a}_n(l)} = \underbrace{\frac{\partial}{\partial \hat{a}_n(l)} \left(\hat{B}(q^{-1}, l)u(k-d) \right)}_0 - \frac{\frac{\partial \hat{A}_1(q^{-1}, l)}{\partial \hat{a}_n(l)} \hat{y}(k-1)}{\hat{y}(k-1)} \quad (4)$$

0 B enthält kein a

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial \hat{a}_n(l)} &= -\hat{a}_1(l) \frac{\partial \hat{y}(k-1)}{\partial \hat{a}_n(l)} \cdots - \frac{\partial (\hat{a}_n(l) \hat{y}(k-n))}{\partial \hat{a}_n(l)} \cdots - \hat{a}_{n_A}(l) \frac{\partial \hat{y}(k-n_A)}{\partial \hat{a}_n(l)} \\ \frac{\partial \hat{a}_n(l) \hat{y}(k-n)}{\partial \hat{a}_n(l)} &= \hat{y}(k-n) + \hat{a}_n(l) \frac{\partial \hat{y}(k-n)}{\partial \hat{a}_n(l)} \end{aligned} \quad (5)$$

product rule = (fq)' = f'q + fq'

$$\frac{\partial \hat{y}(k)}{\partial \hat{a}_n(l)} = \overset{\text{über alle Koeffizienten}}{-\hat{y}(k-n) - \sum_{m=1}^{n_A} \hat{a}_m(l) \frac{\partial \hat{y}(k-m)}{\partial \hat{a}_n(l)}} \quad \text{rekursive Gleichung}$$

$$\frac{\partial \hat{y}(k)}{\partial \hat{b}_n(l)} = \underbrace{\frac{\partial(\hat{\mathbf{B}}(q^{-1}, l)u(k-d))}{\partial \hat{b}_n(l)}}_{*} - \underbrace{\frac{\partial(\hat{\mathbf{A}}_1(q^{-1}, l)\hat{y}(k-1))}{\partial \hat{b}_n(l)}}_{**}$$

$$* = u(k-d-n)$$

$$** = \sum_{m=1}^{n_A} \hat{a}_m(l) \frac{\partial \hat{y}(k-m)}{\partial \hat{b}_n(l)}$$

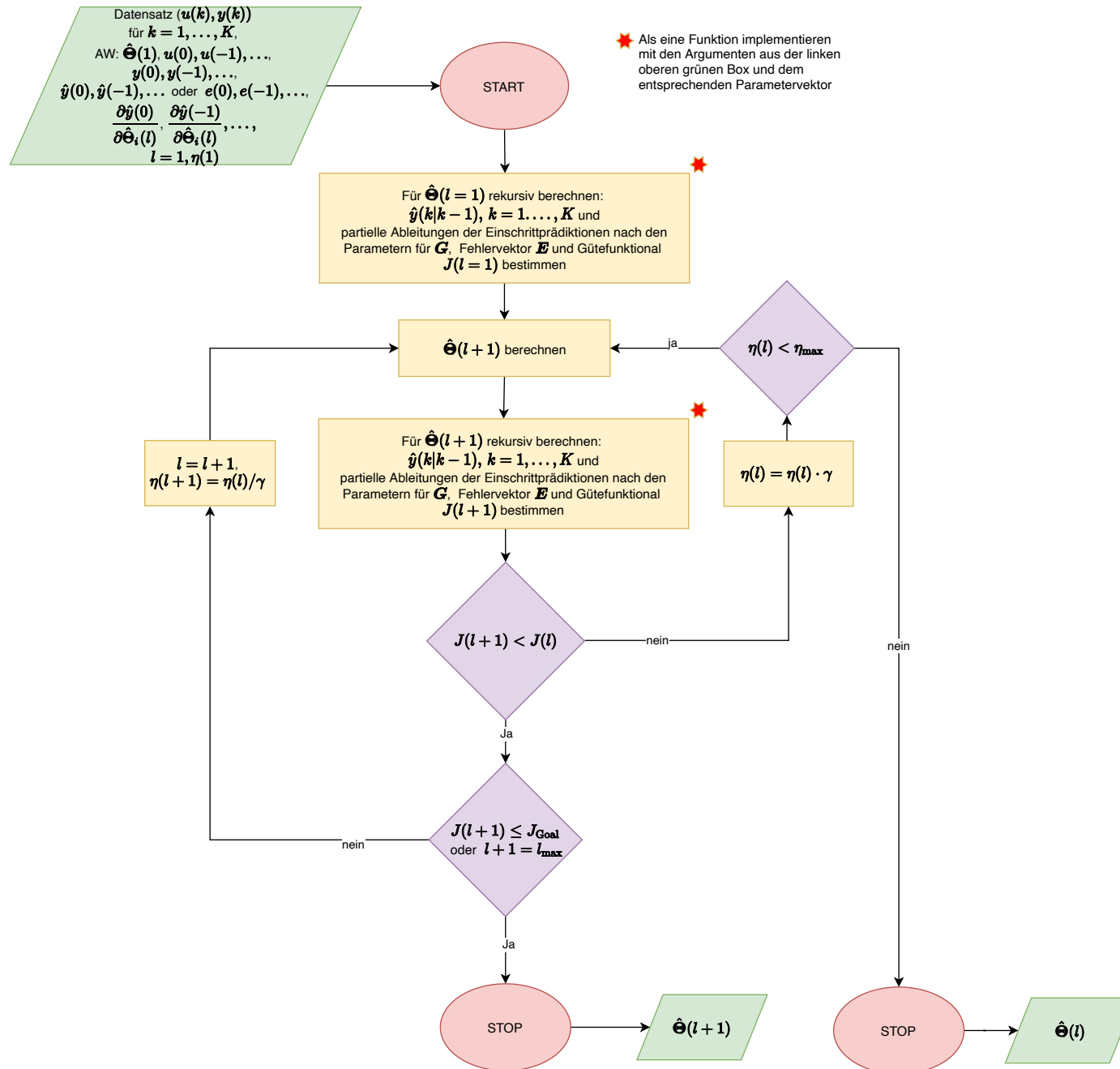
$$\frac{\partial \hat{y}(k)}{\partial \hat{b}_n(l)} = u(k-d-n) - \sum_{m=1}^{n_A} \hat{a}_m(l) \frac{\partial \hat{y}(k-m)}{\partial \hat{b}_n(l)}$$

Rekursiver Algorithmus:

$$\frac{\partial \hat{y}(k)}{\partial \hat{a}_n(l)} \Rightarrow \Psi_n^a(k), \quad \frac{\partial \hat{y}(k)}{\partial \hat{b}_n(l)} \Rightarrow \Psi_n^b(k)$$

$$\Psi_n^b(k) = u(k - d - n) - \sum_{m=1}^{n_A} a_m \Psi_n^b(k - m), \quad k = 1 \cdots K$$

$$\Psi_n^a(k) = -\hat{y}(k - n) - \sum_{m=1}^{n_A} a_m \Psi_n^a(k - m), \quad k = 1 \cdots K$$



5.4 Batch versus Sample Adaptation (Rekursive Prädiktionsfehlermethode)

Normalerweise wird ein ganzer Datensatz (Batch) mit vielen Messungen für die iterativen Parameter-Updates verwendet. Alle Daten fließen in das Gütefunktional und dessen Ableitungen ein. Die allgemeine Update-Gleichung lautet

$$\hat{\Theta}_{l+1} = f \left(\hat{\Theta}_j, J(\hat{\Theta}_j), \frac{\partial J(\hat{\Theta}_j)}{\partial \hat{\Theta}_j}, \dots \right) \quad \text{mit } j = l, l-1, \dots,$$

wobei in den meisten Fällen nur $j = l$ verwendet wird:

$$\hat{\Theta}_{l+1} = f \left(\hat{\Theta}_l, J(\hat{\Theta}_l), \frac{\partial J(\hat{\Theta}_l)}{\partial \hat{\Theta}_l}, \dots \right).$$

Einen Sonderfall erhält man, wenn man den Datensatz auf eine Messung reduziert und $l = k$ und $J = J(k) = \frac{1}{2}e(k)^2$ ansetzt. Man spricht dann von der sogenannten *Sample Adaptation* oder *Rekursiver PEM*.

$$\hat{\Theta}_{k+1} = f \left(\hat{\Theta}_k, \frac{1}{2}e(k, \hat{\Theta}_k)^2, \frac{\partial \frac{1}{2}e(k, \hat{\Theta}_k)^2}{\partial \hat{\Theta}_k}, \dots \right)$$

- Es liegt die Versuchung nahe, ein solches Verfahren für die Onlineschätzung von Parametern zu verwenden. Jedoch muss bedacht werden, dass es sich hier um ein iteratives Verfahren und nicht um ein rekursives Verfahren wie RLS handelt.
 - RLS liefert das gleiche Ergebnis wie LS
 - Sample Adaptation liefert nicht das gleiche Ergebnis wie ein Batch Adaptation-Schritt der gleichen Daten, da in jedem Schritt der Sample Adaptation ein anderes Gütefunktional optimiert wird.
 - Auch ist die Konvergenz bei der Sample Adaptation extrem langsam, da über alle Messdaten nur ein Iterationsschritt im Sinne des Batchverfahrens näherungsweise durchgeführt wird.
-

—

—