

# Zustands- und Parameterschätzung für nichtlineare Networked Control Systems

Peter Philipp

Lehrstuhl für Regelungstechnik, Technische Universität München

Boltzmannstr. 15, 85748 Garching

E-Mail: peter.philipp@tum.de

*In dieser Arbeit wird ein Beobachterentwurf zur Zustands- und Parameterschätzung für abgetastete nichtlineare Systeme vorgestellt, bei denen der Sensor eine unsynchronisierte Uhr besitzt. Die Messdaten werden dabei zusammen mit den relativen Zeitstempeln als Pakete über ein Kommunikationsnetzwerk an den Beobachter gesendet. Diese Pakete erfahren zufällige und variable Totzeiten, oder können gar gänzlich ausfallen. Unter Zuhilfenahme einer festen Anzahl an Paketen, sowie den Modellgleichungen, kann das Beobachtungsproblem als Optimierungsproblem formuliert und gelöst werden. Abschließend wird dieses Beobacherverfahren auf einen nichtlinearen Rührkesselreaktor angewandt.*

## 1 Einleitung

Fortschritte im Bereich der Kommunikationstechnik eröffnen die Möglichkeit, verteilte Networked Control Systems (NCS) zu entwerfen, bei denen die Kommunikation zwischen den einzelnen Elementen (z.B. zwischen Sensor und Regler oder Regler und Aktor) über ein gemeinsames funkbasiertes Datennetzwerk (z.B. WLAN) erfolgt. Dadurch können neue flexible Strukturen realisiert werden, die einerseits die Installations- und Wartungskosten durch den Wegfall der drahtgebundenen Kommunikationsleitungen senken und andererseits neue regelungstechnische Konzepte ermöglichen. Einer der größten Nachteile von NCS ist die Tatsache, dass die gesendeten Datenpakete variablen und zufälligen Totzeiten sowie Datenverlusten unterliegen. Es ist deswegen ratsam, beide Phänomene explizit im Regelungsentwurf zu berücksichtigen, z.B. durch Einsatz entsprechender Beobachter, die die angesprochenen Auswirkungen der Datenübertragung kompensieren und darüber hinaus die nicht messbaren Zustände und Parameter schätzen können.

Für den klassischen Fall, d.h. nichtlineare Zustandsschätzung ohne Netzwerkeinflüsse, existieren viele verschiedene Ansätze. Dabei gewinnen die sogenannten Moving Horizon Estimator (MHE) [RRM03] für das hier betrachtete Problem an Bedeutung, insbesondere die unconstrained Moving Horizon Observer (MHO), die Newtons Algorithmus oder entsprechende Modifikationen verwenden. In [Zim97, MM95] wurde je ein MHO für zeitkontinuierliche Messungen vorgeschlagen. Beide Konzepte basieren auf der Idee der Echtzeitminimierung einer Gütefunktion auf einem Beobachtungshorizont. Für den zeitdiskreten Fall wurde eine Methode in [MG95] vorgestellt, die das Beobachtungsproblem als Lösung eines Satzes algebraischer Gleichungen formuliert. Ein weiterer Newton-basierter MHO wurde in [Vor03] vorgestellt, der die Ideen aus [Zim97] aufgreift, und für den zeitdiskreten Fall erweitert.

Aufbauend auf diesen Ideen, besteht das Ziel dieses Beitrags darin, einen Beobachter für Zustands- und Parameterschätzung für abgetastete nichtlineare NCS zu entwerfen, der zufällige und variable Totzeiten der Pakete sowie Packetausfälle zwischen Sensor und Beobachter explizit berücksichtigt. Das Beobachtungsproblem wird zusätzlich durch die Tatsache erschwert, dass der Sensor über keine synchronisierte Uhr verfügt, und deshalb mit Hilfe der eigenen Uhr einen relativen Zeitstempel generiert.

Dabei ist der Rest des Beitrag wie folgt strukturiert: In Kapitel 2 erfolgt neben der formalen Problembeschreibung auch die Erläuterung der verwendeten Notation. Diese werden in Kapitel 3 benötigt, um das Beobachtungsproblem als mathematisches Optimierungsproblem formulieren zu können. Eine Übersicht über unrestringierte Optimierungsverfahren (Line-Search Methoden vs. Trust-Region Verfahren) in Kapitel 4 ermöglicht die Umformulierung und damit die Lösung des Optimierungsproblems mittels dem Gauß-Newton bzw. dem Levenberg-Marquardt Verfahren. Simulationsergebnisse an einem nichtlinearen Rührkesselreaktor werden in Kapitel 5 präsentiert, bevor der Beitrag in Kapitel 6 mit einer Zusammenfassung endet.

## 2 Problembeschreibung

Gegeben sei die NCS Struktur aus Abb. 1. Die Strecke genügt folgender nichtlinearer Systemgleichung

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{a}, \mathbf{u}(t)) \quad (1a)$$

$$\mathbf{x}(0) = \mathbf{x}_S \quad (1b)$$

$$\mathbf{y}(t) = h(\mathbf{x}(t)) \quad (1c)$$

mit dem Zustandsvektor  $\mathbf{x} \in \mathbb{R}^n$ , dem Parametervektor  $\mathbf{a} \in \mathbb{R}^q$ , dem Eingangsvektor  $\mathbf{u} \in \mathbb{R}^m$  und dem Ausgang  $y \in \mathbb{R}$ . Der Anfangswert des Zustandsvektors zur Zeit  $t = 0$  sei  $\mathbf{x}_S \in \mathbb{R}^n$ . Für die Funktionen  $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^m \mapsto \mathbb{R}^n$  und  $h : \mathbb{R}^n \mapsto \mathbb{R}$  gelte:  $\mathbf{f}, h \in \mathcal{C}^r$ , mit  $r \geq 1$ .

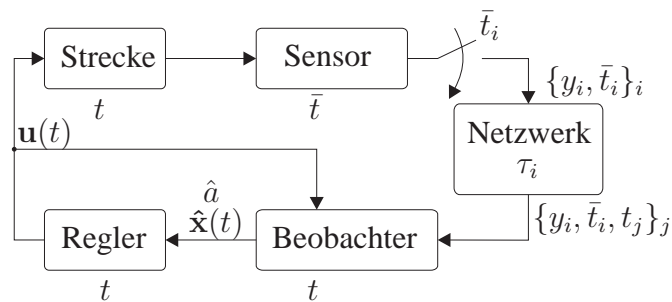


Abbildung 1: Struktur des Networked Control Systems (NCS)

Das Kommunikationsnetzwerk befindet sich zwischen Sensor und Beobachter. Es wird angenommen, dass das Netzwerk als Kanal aufgefasst werden kann, das Pakete, bestehend aus reellen Zahlen, übermittelt, und dass diese Pakete zufällig ausfallen können. Diese Annahme ist gerechtfertigt, wenn die Anzahl an Bits pro Paket ausreichend groß ist, so dass Quantisierungseffekte vernachlässigt werden können, nicht aber Paketverluste. Zudem erfährt jedes Paket eine zufällige und variable Totzeit  $\tau_i \geq 0$ , was zu Paketüberholvorgängen führen kann.

Die Strecke, der Regler und der Beobachter sind zeitsynchronisiert, d.h. sie besitzen eine globale Zeit  $t$ . Der Sensor hingegen besitzt keine drahtgebundene Verbindung zu diesen Elementen, und somit auch keine globale Zeit. Deshalb verwendet er seine eigene unsynchronisierte lokale Zeit  $\bar{t}$ . Die Beziehung zwischen den einzelnen Zeiten wird wie folgt angenommen

$$t = \bar{t} + t_o, \quad (2)$$

wobei  $t_o$  der unbekannte Zeitversatz ist. Zudem tastet der Sensor den Ausgang  $y(t)$  nichtäquidistant ab, indem er z.B. eine Kombination aus zeit- und ereignisbasierter Abtaststrategie benutzt. Die abgetasteten Messwerte  $y_i$  werden zusammen mit den relativen Zeitstempeln  $\bar{t}_i$ , die

den Abtastzeitpunkt indizieren, als einzelne Pakete der Form  $\{y_i, \bar{t}_i\}_i$  durch das Kommunikationsnetzwerk geschickt. Falls keine Paketverluste auftreten, kommen diese Pakete nach den unbekanntem und zufälligen Totzeiten  $\tau_i$  zu den Zeitpunkten  $t_j$  am Beobachter an. Aufgrund der Tatsache, dass diese Totzeiten  $\tau_i$  variabel sind, kann es zu Paketüberholvorgängen kommen, so dass die Ankunftssequenz anders als die Sendereihenfolge ist (daraus folgt der andere Index  $j$ , der eine chronologische Nummerierung anhand der jeweiligen Uhren impliziert). Das Ziel besteht nun darin, aus der Folge der 3-Tupel  $\{y_i, \bar{t}_i, t_j\}$ , zusammen mit dem Eingangsvektor  $\mathbf{u}(t)$ , den aktuellen Zustandsvektor  $\mathbf{x}(t)$ , den unbekanntem Parametervektor  $\mathbf{a}$  sowie den unbekanntem Zeitversatz  $t_o$  zu schätzen.

### 3 Formulierung des Beobachtungsproblems als Optimierungsproblem

Die unterschiedlichen Zeiten, sowie die Einflüsse der Netzwerkeffekte, sind in Abb. 2 visualisiert. Ein Kreuz auf der oberen Zeitachse  $\bar{t}$  des Sensors repräsentiert ein Paket mit dem abgetasteten Messwert  $y_i$  zur Zeit  $\bar{t}_i$ . Diese Situation kann mit Hilfe von (2) auch in der globalen Zeit  $t$  dargestellt, und auf der mittleren Zeitachse betrachtet werden. Zu den Zeiten  $t_j - \tau_j$ , die äquivalent zu den Zeiten  $\bar{t}_i + t_o$  sind, wird das entsprechende Paket über das Kommunikationsnetzwerk abgeschickt und erreicht, falls kein Paketverlust auftritt, zur Zeit  $t_j$  den Beobachter. Zwei sich schneidende Pfeile weisen auf einen Paketüberholvorgang hin.

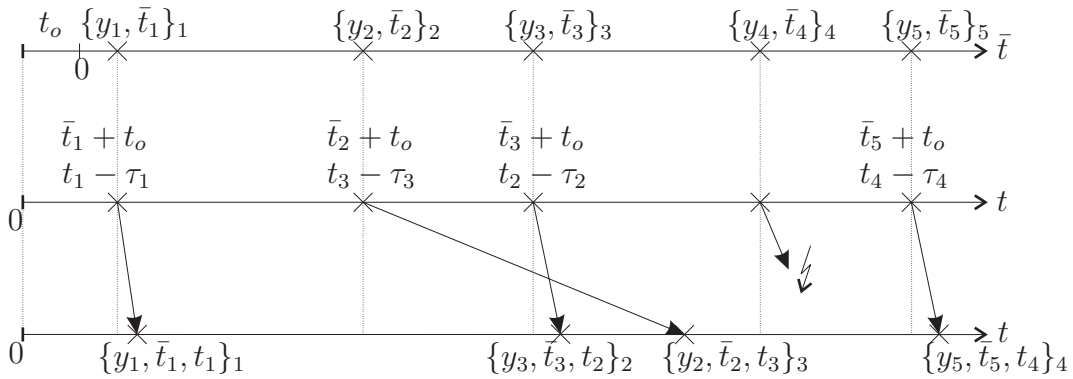


Abbildung 2: Illustration der verschiedenen Zeiten sowie der Netzwerkeinflüsse

Bei bekanntem Zeitversatz  $t_o$  könnten alle Totzeiten  $\tau_j$  berechnet werden

$$\tau_j = t_j - \bar{t}_i - t_o. \quad (3)$$

Die zeitdiskretisierten Messwerte  $y_i$  der Strecke (1) in der globalen Zeit  $t$  können als

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \int_{\bar{t}_i + t_o}^{\bar{t}_{i+1} + t_o} \mathbf{f}(\mathbf{x}(t), \mathbf{a}, \mathbf{u}(t)) dt \quad (4a)$$

$$\mathbf{x}_0 = \mathbf{x}_S \quad (4b)$$

$$y_i = h(\mathbf{x}_i) \quad (4c)$$

geschrieben werden. Damit ist die Folge der 3-Tupel  $\{y_i, \bar{t}_i, t_j\}$  definiert.

Der nächste Schritt um das Beobachtungsproblem als Optimierungsproblem formulieren zu können, besteht in der Einführung eines bewegten Horizonts der Länge  $N + 1$ , um die Pakete in

eine chronologische Reihenfolge zu bringen. Dazu werden die ersten  $N + 1$  Pakete, die am Beobachter ankommen, anhand der relativen Zeitstempel sortiert und nummeriert, hier beispielhaft die Pakete aus Abb. 2 mit  $N + 1 = 4$

$$\begin{pmatrix} \{y_1, \bar{t}_1, t_1\}_1 \\ \{y_3, \bar{t}_3, t_2\}_2 \\ \{y_2, \bar{t}_2, t_3\}_3 \\ \{y_5, \bar{t}_5, t_4\}_4 \end{pmatrix} \xrightarrow{\text{Sortierung}} \begin{pmatrix} \{y_1, \bar{t}_1, t_1\}_1 \\ \{y_2, \bar{t}_2, t_3\}_3 \\ \{y_3, \bar{t}_3, t_2\}_2 \\ \{y_5, \bar{t}_5, t_4\}_4 \end{pmatrix} \xrightarrow{\text{Nummerierung}} \begin{pmatrix} \{y_1, \bar{t}_1, t_1\}_1 \\ \{y_2, \bar{t}_2, t_3\}_2 \\ \{y_3, \bar{t}_3, t_2\}_3 \\ \{y_5, \bar{t}_5, t_4\}_4 \end{pmatrix} \quad (5)$$

Sobald ein weiteres Paket eintrifft, erfolgt die Sortierung und die fortlaufende Nummerierung analog. Anschließend wird das erste Paket im Horizont, also das Paket mit dem geringsten relativen Zeitstempel, gelöscht. Der Vorteil dieses bewegten Horizonts besteht zum einen darin, den Paketüberholvorgängen entgegen zu wirken, und zum anderen, die Länge des Horizonts, und damit den Speicheraufwand, konstant zu halten. Dies wird sich später vorteilhaft auf die Optimierung auswirken. Falls das neu angekommene Paket zugleich das erste Paket nach der Sortierung sein sollte, so wird dieses Paket als Paketverlust deklariert.

Im folgenden werden Notationen der Form  $\hat{\mathbf{x}}_{j|k}$  auftauchen. Das Symbol “ $\hat{\phantom{x}}$ ” über einer Variablen bedeutet, dass es sich um einen Schätzwert der selbigen zum Abtastzeitpunkt im  $j$ -ten Paket handelt, bezogen auf den sortierten  $(k - N)$ -ten Horizont. Am Beispiel von (5) bedeutet  $\hat{\mathbf{x}}_{4|4}$ , dass es sich um einen Schätzwert von  $\mathbf{x}$  zum Abtastzeitpunkt  $\bar{t}_5 + t_o$  bzw.  $t_4 - \tau_4$  handelt, basierend auf Informationen im  $4 - 3 = 1$ -ten Horizont.

Mit Hilfe dieser Vorüberlegungen lässt sich das Beobachtungsproblem wie folgt als Optimierungsproblem schreiben:

$$\min_{\substack{\hat{\mathbf{x}}_{k-N|k} \\ \hat{t}_o \\ \hat{\mathbf{a}}}} \Phi_k = \sum_{j=k-N}^k r_{j|k} R_j^{-1} r_{j|k} \quad (6)$$

mit den Nebenbedingungen

$$\hat{\mathbf{x}}_{j+1|k} = \hat{\mathbf{x}}_{j|k} + \int_{\hat{t}_i + \hat{t}_o}^{\bar{t}_{i+1} + \hat{t}_o} \mathbf{f}(\hat{\mathbf{x}}(t), \hat{\mathbf{a}}, \mathbf{u}(t)) dt, \quad j = k - N, \dots, k - 1 \quad (7a)$$

$$\hat{\mathbf{x}}_{k-N|k} = \hat{\mathbf{x}}_{k-N|k-1} \quad (7b)$$

$$r_{j|k} = y_j - h(\hat{\mathbf{x}}_{j|k}), \quad j = k - N, \dots, k \quad (7c)$$

und den Anfangsbedingungen

$$\hat{\mathbf{x}}_{0|N} = \hat{\mathbf{x}}_S, \quad \hat{t}_o = \hat{t}_S, \quad \hat{\mathbf{a}} = \hat{\mathbf{a}}_S \quad (8)$$

Die Idee ist dabei, zuerst die unbekannt Parameter  $\hat{\mathbf{x}}_{k-N|k}$ ,  $\hat{t}_o$  und  $\hat{\mathbf{a}}$  für den aktuellen Beobachtungshorizont zu finden, und anschließend mit den optimalen Werten das System bis zum aktuellen Zeitpunkt zu simulieren und damit den aktuellen Zustandsvektor zu schätzen.

Die zu minimierende Gütefunktion  $\Phi_k$  (6) besteht aus der gewichteten quadratischen Summe der Differenzen zwischen gemessenen und simulierten Messwerten des Systems, bezogen auf den aktuellen Horizont. Die Gewichtungsfaktoren  $R_i^{-1}$  für den Eingrößenfall, d.h. nur ein Sensor, erlaubt die Priorisierung der einzelnen Pakete. Die Nebenbedingungen (7) stellen zum einen die Bestimmungsgleichungen für die zeitdiskretisierten Messwerte  $y_i$  aus (4) dar, und zum anderen legen sie den Startwert für den Anfangswert  $\hat{\mathbf{x}}_{k-N|k}$  auf dem Horizont fest (7b). Die Anfangswerte  $\hat{\mathbf{x}}_S$  und  $\hat{\mathbf{a}}_S$  müssen vorgegeben werden, der Anfangswert  $\hat{t}_S$  hingegen nicht.

Dieser kann aus den Informationen in den ersten  $N + 1$  Paketen, und optional über die Vorgabe einer geschätzten mittleren Totzeit  $\tau_m$  des Netzwerks, wie folgt für die erste Optimierung generiert werden:

$$\hat{t}_S = \min \left\{ \frac{1}{N+1} \sum_{j=1}^{N+1} \tilde{t}_j - \tau_m, \tilde{t}_1, \dots, \tilde{t}_{N+1} \right\}, \quad \text{s.t. } \tilde{t}_j = t_j - \bar{t}_i. \quad (9)$$

Für jeden neuen Horizont, d.h. mit Ankunft eines neuen, nicht als Paketausfall deklarierten Pakets, wird das Optimierungsproblem erneut gelöst; allerdings mit verbesserten Startwerten gegenüber dem vorherigen Optimierungsdurchgang, da die neuen Startwerte auf den Ergebnissen der vorherigen Optimierung basieren.

## 4 Lösung des Optimierungsproblems

### 4.1 Unrestringierte Optimierungsverfahren

Ein unrestringiertes Optimierungsproblem ist definiert als

$$\min_{\mathbf{z}} \Phi(\mathbf{z}), \quad (10)$$

mit  $\mathbf{z} \in \mathbb{R}^n$ ,  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ . Um dieses Optimierungsproblem zu lösen, existieren viele verschiedene Verfahren, die ausgehend von einem Startwert  $\mathbf{z}_0$  eine Iterationsfolge  $\{\mathbf{z}_i\}_{i=0}^{\infty}$  generieren. Diese wird terminiert, wenn entweder kein weiterer Fortschritt erreicht werden kann, oder wenn eine Lösung mit ausreichender Genauigkeit approximiert wurde. Dabei lassen sich die Algorithmen in zwei unterschiedliche Ansätze unterteilen: die Line-Search Verfahren und die Trust-Region Verfahren. Sie unterscheiden sich in der Art und Weise, wie sie ausgehend von dem aktuellen Punkt  $\mathbf{z}_i$  zu dem nächsten Punkt  $\mathbf{z}_{i+1}$  fortschreiten. Beide Verfahren verwenden im  $i$ -ten Iterationsschritt ein Modell  $m_i(\cdot)$  von der ursprünglichen Funktion  $\Phi(\cdot)$ .

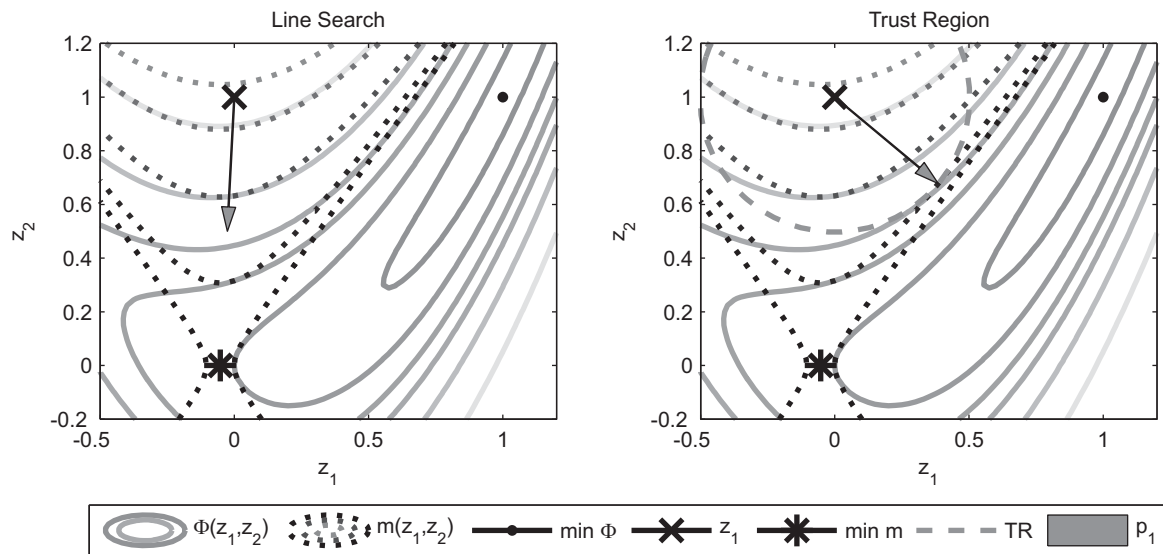


Abbildung 3: Line-Search Verfahren vs. Trust-Region Methoden

Line-Search Algorithmen (s. Abb. 3 links) bestimmen anhand des Modells  $m_i$  eine Suchrichtung  $\mathbf{p}_i$  durch Lösen des Minimierungsproblems

$$\min_{\mathbf{p}_i} m_i(\mathbf{z}_i + \mathbf{p}_i). \quad (11)$$

Anschließend wird ausgehend von dem aktuellen Punkt  $\mathbf{z}_i$  entlang der gefundenen Suchrichtung  $\mathbf{p}_i$  ein neuer Punkt  $\mathbf{z}_{i+1}$  mit einem geringeren Funktionswert  $\Phi(\cdot)$  gesucht. Die Schrittweite  $\lambda_i$ , die angibt, wie weit in diese Richtung gegangen werden soll, wird als näherungsweise Lösung des eindimensionalen Minimierungsproblems

$$\min_{\lambda_i > 0} \Phi(\mathbf{z}_i + \lambda_i \mathbf{p}_i) \quad (12)$$

berechnet. Der nächste Schritt ist damit durch

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda_i \mathbf{p}_i \quad (13)$$

bestimmt.

Trust-Region Verfahren (s. Abb. 3 rechts) tragen der Tatsache Rechnung, dass dem Modell  $m_i$  nur in einem gewissen Gebiet, der sogenannten Trust-Region, vertraut werden darf. Die Suchrichtung  $\mathbf{p}_i$  wird deshalb durch Lösen des folgenden Minimierungsproblems

$$\min_{\mathbf{p}_i} m_i(\mathbf{z}_i + \mathbf{p}_i), \quad \text{s.t. } \|\mathbf{D}_i \mathbf{p}_i\|_2 \leq \Delta_i. \quad (14)$$

bestimmt. Damit ist nicht nur die Suchrichtung, sondern auch die Schrittweite festgelegt, so dass sich der nächste Schritt direkt aus

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \mathbf{p}_i \quad (15)$$

ergibt.

## 4.2 Bestimmung des Modells $m_i$

Zunächst wird die Gütefunktion  $\Phi_k$  (6) im  $k$ -ten Beobachtungshorizont durch Einführung von

$$\mathbf{r}_k = \begin{pmatrix} r_{k-N|k} \\ r_{k-N+1|k} \\ \vdots \\ r_{k|k} \end{pmatrix}, \quad \mathbf{W}_k = \begin{bmatrix} R_{k-N}^{-1} & & \\ & \ddots & \\ & & R_k^{-1} \end{bmatrix}, \quad (16)$$

umformuliert zu

$$\Phi_k = \mathbf{r}_k^T \mathbf{W}_k \mathbf{r}_k. \quad (17)$$

Im folgenden wird die Optimierung auf einem festen Beobachtungshorizont betrachtet, bei der die Indizes  $j|k$  konstant bleiben und deshalb aus Übersichtsgründen weggelassen werden. Der Index  $i$  steht im weiteren für den  $i$ -ten Iterationsschritt innerhalb der Optimierung auf dem Beobachtungshorizont. Die Optimierungsvariablen sowie dessen Änderungen im  $i$ -ten Optimierungsschritt werden zu den Vektoren

$$\mathbf{z}_i = \begin{pmatrix} \hat{\mathbf{x}}_{k-N|k} \\ \hat{t}_o \\ \hat{\mathbf{a}} \end{pmatrix}, \quad \mathbf{p}_i = \begin{pmatrix} \Delta \hat{\mathbf{x}}_{k-N|k} \\ \Delta \hat{t}_o \\ \Delta \hat{\mathbf{a}} \end{pmatrix} \quad (18)$$

zusammengefasst.

Durch eine Taylorreihenentwicklung von  $\Phi$  (17) um den Optimierungsvektor  $\mathbf{z}_i$  mit Abbruch nach dem quadratischen Glied liefert das Modell  $m_i$

$$m_i(\mathbf{z}_i + \mathbf{p}_i) = \Phi_i + \mathbf{p}_i^T \nabla \Phi_i + \frac{1}{2} \mathbf{p}_i^T \mathbf{B}_i \mathbf{p}_i, \quad (19)$$

mit

$$\Phi_i = \mathbf{r}_i^T \mathbf{W} \mathbf{r}_i \quad (20)$$

$$\nabla \Phi_i = \mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{r}_i \quad (21)$$

$$\mathbf{B}_i = \mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{J}_i + \underbrace{\sum_{j=1}^{N+1} \mathbf{r}_i^T (\mathbf{W}^T + \mathbf{W})_j \mathbf{H}_j}_{\approx 0}, \quad (22)$$

wobei  $\mathbf{J}_i$  die Jakobimatrix von  $\mathbf{r}_i$ ,  $\mathbf{H}_j$  die Hessematrix der  $j$ -ten Zeile von  $\mathbf{r}_i$  und  $(\mathbf{W}^T + \mathbf{W})_j$  die  $j$ -te Spalte von  $(\mathbf{W}^T + \mathbf{W})$  ist. Der zweite Summand von  $\mathbf{B}_i$  kann vernachlässigt werden, wenn

$$\sum_{j=1}^{N+1} |\mathbf{r}_i^T (\mathbf{W}^T + \mathbf{W})_j| \|\mathbf{H}_j\|_2 \ll \max \text{eig}(\mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{J}_i), \quad (23)$$

d.h. wenn entweder die Residuen  $\mathbf{r}_i$  sehr klein sind, oder wenn sich diese im betrachteten Bereich wie lineare Funktionen verhalten. Somit muss für das Modell nur die Jakobimatrix  $\mathbf{J}_i$  berechnet werden.

### 4.3 Berechnung der Jakobimatrix $\mathbf{J}_i$

Die Jakobimatrix  $\mathbf{J}_i$  ist definiert als

$$\mathbf{J}_i = \frac{\partial \mathbf{r}_i}{\partial \mathbf{z}_i} = \begin{bmatrix} \frac{\partial \mathbf{r}_i}{\partial \hat{\mathbf{x}}_{k-N|k}} & \frac{\partial \mathbf{r}_i}{\partial \hat{t}_o} & \frac{\partial \mathbf{r}_i}{\partial \hat{\mathbf{a}}} \end{bmatrix}, \quad (24)$$

und besteht aus den drei partiellen Ableitungen von  $\mathbf{r}_i$  nach den Optimierungsvariablen. Für jede dieser partiellen Ableitungen gilt

$$\frac{\partial \mathbf{r}_i}{\partial \omega} = \begin{bmatrix} - \left( \frac{\partial h}{\partial \hat{\mathbf{x}}} \Big|_{\hat{\mathbf{x}}_{k-N|k}} \right)^T \frac{\partial \hat{\mathbf{x}}_{k-N|k}}{\partial \omega} \\ \vdots \\ - \left( \frac{\partial h}{\partial \hat{\mathbf{x}}} \Big|_{\hat{\mathbf{x}}_{k|k}} \right)^T \frac{\partial \hat{\mathbf{x}}_{k|k}}{\partial \omega} \end{bmatrix}, \quad (25)$$

wobei  $\omega$  ein Platzhalter für die Optimierungsvariablen  $\hat{\mathbf{x}}_{k-N|k}$ ,  $\hat{t}_o$  bzw.  $\hat{\mathbf{a}}$  darstellt. Der erste Faktor einer jeden Zeile besteht aus der Ableitung von  $h(\cdot)$ , ausgewertet an den entsprechenden Stellen, und ist somit für alle drei partiellen Ableitungen identisch. Für den zweiten Faktor einer jeden Zeile müssen jeweils unterschiedliche partielle Ableitungen berechnet werden.

Für die partiellen Ableitung nach  $\hat{\mathbf{x}}_{k-N|k}$  ist folgende Abkürzung hilfreich

$$\mathbf{X}(\bar{t}_j + \hat{t}_o) = \frac{\partial \hat{\mathbf{x}}_{j|k}}{\partial \hat{\mathbf{x}}_{k-N|k}}, \quad j = k - N, \dots, k. \quad (26)$$

Damit resultieren die partiellen Ableitungen in (25) aus der Lösung des matrixwertigen Anfangswertproblems

$$\dot{\mathbf{X}}(t) = \frac{\partial \mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{a}}, \mathbf{u})}{\partial \hat{\mathbf{x}}} \mathbf{X}(t), \quad (27)$$

mit  $t \in [\bar{t}_{k-N} + \hat{t}_o, \bar{t}_k + \hat{t}_o]$  und dem Anfangswert

$$\mathbf{X}(\bar{t}_{k-N} + \hat{t}_o) = \mathbf{I}. \quad (28)$$



Die partiellen Ableitungen nach  $\hat{t}_o$  in (25) ergeben sich aus

$$\frac{\partial \hat{\mathbf{x}}_{j|k}}{\partial \hat{t}_o} = \mathbf{f}(\hat{\mathbf{x}}(\bar{t}_j + \hat{t}_o), \hat{\mathbf{a}}, \mathbf{u}(\bar{t}_j + \hat{t}_o)), \quad (29)$$

mit  $j = k - N, \dots, k$ .

Die letzten noch fehlenden partiellen Ableitungen nach  $\hat{\mathbf{a}}$  in (25) können mit Hilfe der Abkürzung

$$\mathbf{Y}(\bar{t}_j + \hat{t}_o) = \frac{\partial \hat{\mathbf{x}}_{j|k}}{\partial \hat{\mathbf{a}}}, \quad j = k - N, \dots, k, \quad (30)$$

aus der Lösung eines weiteren matrixwertigen Anfangswertproblems

$$\dot{\mathbf{Y}}(t) = \frac{\partial \mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{a}}, \mathbf{u})}{\partial \hat{\mathbf{x}}} \mathbf{Y}(t) + \frac{\partial \mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{a}}, \mathbf{u})}{\partial \hat{\mathbf{a}}}, \quad (31)$$

mit  $t \in [\bar{t}_{k-N} + \hat{t}_o, \bar{t}_k + \hat{t}_o]$  und dem Anfangswert

$$\mathbf{Y}(\bar{t}_{k-N} + \hat{t}_o) = \mathbf{0}. \quad (32)$$

gewonnen werden. Die partiellen Ableitungen in  $\mathbf{J}_i$  nach den unterschiedlichen Optimierungsvariablen sind unabhängig voneinander und können deshalb parallel berechnet werden.

#### 4.4 Gauß-Newton Verfahren

Das Gauß-Newton Verfahren zählt zu den Line-Search Verfahren. Die Suchrichtung wird mit Hilfe des Modells (19) als Lösung des Optimierungsproblems (11) bestimmt. Die notwendige Bedingung für ein Minimum, und damit die Suchrichtung  $\mathbf{p}_i$  lautet

$$\frac{\partial \Phi_i}{\partial \mathbf{z}_i} = \mathbf{0} \Leftrightarrow \mathbf{p}_i = -\mathbf{B}_i^{-1} \nabla \Phi_i = -[\mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{J}_i]^{-1} \mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{r}_i. \quad (33)$$

Die ideale Schrittweite  $\lambda_i$  ist das globale Minimum von  $\Phi(\lambda_i) = \Phi(\mathbf{z}_i + \lambda_i \mathbf{p}_i)$ . Im allgemeinen ist die exakte Berechnung sehr aufwändig, so dass inexakte Line-Search Verfahren bevorzugt werden. Diese finden mit möglichst wenig Funktions- und Jakobimatrixaufrufen eine Schrittweite, die eine adäquate Reduktion von  $\Phi(\cdot)$  mit minimalem Kostenaufwand ermöglicht. Dabei wird eine Folge von möglichen Schrittweiten  $\lambda_i$  solange getestet, bis verschiedene Bedingungen erfüllt werden. Ein weit verbreiteter Vertreter sind die sogenannten starken Wolfe Bedingungen [NW99]

$$\underbrace{\Phi(\mathbf{z}_i + \lambda_i \mathbf{p}_i)}_{\Phi(\lambda_i)} \leq \underbrace{\Phi(\mathbf{z}_i) + c_1 \lambda_i \nabla \Phi_i^T \mathbf{p}_i}_{l(\lambda_i)} \quad (34a)$$

$$\underbrace{|\nabla \Phi(\mathbf{z}_i + \lambda_i \mathbf{p}_i)^T \mathbf{p}_i|}_{|\Phi'(\lambda_i)|} \leq \underbrace{c_2 |\nabla \Phi_i^T \mathbf{p}_i|}_{c_2 |\Phi'(0)|}, \quad (34b)$$

mit  $0 < c_1 < c_2 < 1$ . Die erste Bedingung (34a), die auch als Armijo Bedingung bekannt ist, sagt dabei aus, dass der Funktionswert  $\Phi(\cdot)$  an der Stelle  $\lambda_i$  unterhalb der Geraden  $l(\lambda_i)$  liegen soll, d.h. die Reduktion in  $\Phi(\cdot)$  soll proportional zu der Schrittweite  $\lambda_i$  sowie der Richtungsableitung  $\nabla \Phi_i^T \mathbf{p}_i$  sein (s. Abb. 4). Der Wert von  $c_1$  wird dabei meist sehr klein gewählt, z.B.  $c_1 = 10^{-4}$ .

Die erste Bedingung langt dabei alleine nicht aus, um eine signifikante Reduktion in  $\Phi(\cdot)$  zu erzielen, so dass eine zweite Bedingung (34b), die sogenannte Krümmungsbedingung, hinzu



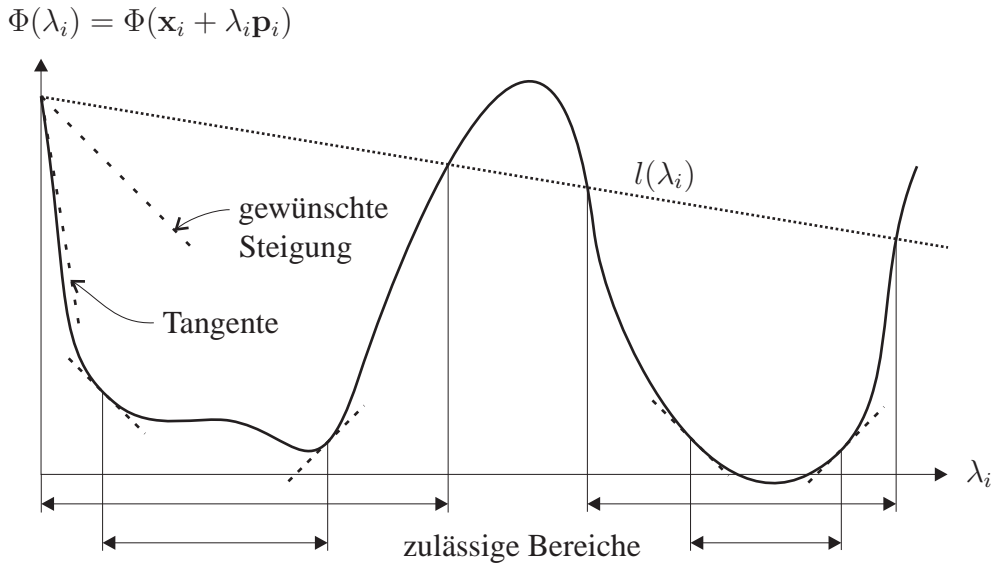


Abbildung 4: Schrittweitenbestimmung mittels den starken Wolfe Bedingungen

kommt. Diese Bedingung lässt nur Schrittweiten  $\lambda_i$  zu, bei denen der Betrag der Steigung  $|\Phi'(\lambda_i)|$  höchstens so groß ist wie  $c_2$  Mal der Betrag der Steigung  $|\Phi'(0)|$ . Damit wird der mögliche Bereich einer zulässigen Schrittweite  $\lambda_i$  auf eine Umgebung um einen stationären Punkt von  $\Phi(\cdot)$  weiter eingeschränkt (s. Abb. 4). Ein typischer Wert von  $c_2$  ist 0.9

Nachdem die Suchrichtung  $\mathbf{p}_i$  und die Schrittweite  $\lambda_i$  bestimmt sind, kann der nächste Schritt gemäß (13) bestimmt werden.

#### 4.5 Levenberg-Marquardt Verfahren

Das Levenberg-Marquardt Verfahren gehört zu der Klasse der Trust-Region Verfahren. Die Suchrichtung wird ebenfalls wie beim Gauß-Newton Verfahren mit Hilfe des Modells  $m_i$  (19) bestimmt, allerdings mit dem Unterschied, dass nur ein bestimmtes Gebiet  $\|\mathbf{D}_i \mathbf{p}_i\|_2 \leq \Delta_i$  für die Suchrichtung und damit für den neuen Schritt  $\mathbf{z}_{i+1}$  zugelassen ist. Die Suchrichtung ergibt sich formal aus der Lösung des Optimierungsproblems mit einer Ungleichungsnebenbedingung

$$\min_{\mathbf{p}_i} m_i(\mathbf{z}_i + \mathbf{p}_i), \quad \text{s.t.} \|\mathbf{D}_i \mathbf{p}_i\|_2 \leq \Delta_i. \quad (35)$$

Zunächst muss überprüft werden, ob die Ungleichungsnebenbedingung die linear independence constraint qualification (LICQ) erfüllt, d.h. ob die Gradienten der aktiven Nebenbedingungen am lokalen Optimum  $\mathbf{p}_i^*$  linear unabhängig sind. Dies ist für alle  $\mathbf{p}_i^* \in \mathbb{R}^{n+q+1}$  möglich, da  $-\mathbf{D}_i^T \mathbf{D}_i \mathbf{p}_i^* \neq \mathbf{0}$  ist. Somit dürfen die Karush-Kuhn-Tucker (KKT) Bedingungen angewandt werden, die notwendig für die Lösung des Optimierungsproblems sind, und für diesen Fall wie folgt lauten

$$\nabla \Phi_i + \mathbf{B}_i \mathbf{p}_i^* + \lambda_i \mathbf{D}_i \mathbf{D}_i \mathbf{p}_i^* = 0 \quad (36a)$$

$$\Delta_i^2 - \|\mathbf{D}_i \mathbf{p}_i^*\|_2^2 \geq 0 \quad (36b)$$

$$\lambda_i \geq 0 \quad (36c)$$

$$\lambda_i (\Delta_i^2 - \|\mathbf{D}_i \mathbf{p}_i^*\|_2^2) = 0. \quad (36d)$$

Gleichung (36a) kann nach der optimalen Suchrichtung aufgelöst werden

$$\mathbf{p}_i^* = - [\mathbf{B}_i + \lambda_i \mathbf{D}_i^T \mathbf{D}_i]^{-1} \nabla \Phi_i \quad (37a)$$

$$= - [\mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{J}_i + \lambda_i \mathbf{D}_i^T \mathbf{D}_i]^{-1} \mathbf{J}_i^T (\mathbf{W}^T + \mathbf{W}) \mathbf{r}_i. \quad (37b)$$

Die eigentliche Trust-Region Methode, die aus einem vorgegebenen Radius  $\Delta_i$  den Lagrange-Multiplikator  $\lambda_i$  und damit die Suchrichtung  $\mathbf{p}_i$  bestimmt, soll jetzt nicht weiter verfolgt werden. Stattdessen soll die klassische Methode der Dämpfungsstrategie verwendet werden. Diese benutzt eine  $\lambda_i$ -Steuerung (=Dämpfungsstrategie) und damit indirekt eine Radiussteuerung. Somit wird direkt aus (37a) zu einem gewählten  $\lambda_i$  (und damit für den zugehörigen Radius) die optimale Suchrichtung  $\mathbf{p}_i^*$  berechnet. Eine mögliche Dämpfungsstrategie [Lam97] schaut dabei wie folgt aus

$$\Phi(\mathbf{z}_i + \mathbf{p}_i) < \Phi(\mathbf{z}_i) \Rightarrow \lambda_{i+1} = a_1 \lambda_i \quad (38a)$$

$$\Phi(\mathbf{z}_i + \mathbf{p}_i) > \Phi(\mathbf{z}_i) \Rightarrow \lambda_i = a_2 \lambda_i, \quad (38b)$$

mit  $0 < a_1 < 1$  und  $a_2 > 1$ . Für  $\lambda_i = 0$  wird das Verfahren zum Gauß-Newton Verfahren mit Schrittweite 1, für  $\lambda_i \gg 1$  wird das Verfahren zum Steepest-Descent Verfahren.

Bei der Wahl der Matrix  $\mathbf{D}_i$ , die die Form des zulässigen Gebietes beschreibt, bestehen zwei Möglichkeiten:

- additive Dämpfung:  $\mathbf{D}_i^T \mathbf{D}_i = \mathbf{I}$
- multiplikative Dämpfung:  $\mathbf{D}_i^T \mathbf{D}_i = \text{diag}(\mathbf{J}_i^T \mathbf{J}_i)$

Die additive Dämpfung beschreibt einen kreisförmigen Vertrauensbereich und ist deshalb sinnvoll für gut skalierte Probleme, d.h. die Optimierungsparameter in  $\mathbf{z}_i$  liegen in einer ähnlichen Größenordnung. Zudem wird die Konditionszahl von  $\mathbf{B}_i$  verbessert, was numerische Vorteile bei der Implementierung mit sich bringt. Die multiplikative Dämpfung ist bei schlecht konditionierten Problemen zu bevorzugen, da ein elliptischer Vertrauensbereich verwendet wird, der die unterschiedlichen Größenordnungen in den Elementen von  $\mathbf{z}_i$  berücksichtigt.

## 5 Simulationsbeispiel

In diesem Abschnitt wird die vorgestellte Theorie auf einen nichtlinearen durchmischten Rührkesselreaktor (CSTR) [URP74] angewandt. Der CSTR ist ein häufig benutzter Benchmark in der chemischen Verfahrenstechnik (vgl. [HS97]).

Ein CSTR (s. Abb. 5) beschreibt den Prozess einer irreversiblen, exothermen Reaktion  $A \rightarrow B$ . Ausgehend von einer Anfangskonzentration  $c_A(0)$  und einer Anfangstemperatur  $T(0)$  im Reaktor, wird die Komponente  $A$  mit der Konzentration  $c_{A0}$  und der Temperatur  $T_{A0}$  kontinuierlich hinzugefügt. Die dabei frei werdende Reaktionswärme wird mit Hilfe des Kühlstroms der Temperatur  $T_C$  abgeführt. Das Reaktorvolumen  $V$  wird durch den abfließenden Volumenstrom  $\dot{V}$  konstant gehalten. Die mathematische Beschreibung des CSTR lautet

$$\dot{c}_A = \frac{\dot{V}}{V} (c_{A0} - c_A) - k_0 c_A \exp\left(-\frac{E_A}{RT}\right) \quad (39a)$$

$$\dot{T} = \frac{\dot{V}}{V} (T_{A0} - T) + \frac{\Delta H_R k_0}{\rho c_p} c_A \exp\left(-\frac{E_A}{RT}\right) + \frac{\alpha A_c}{\rho c_p V} (T_C - T) \quad (39b)$$

$$y = T, \quad (39c)$$

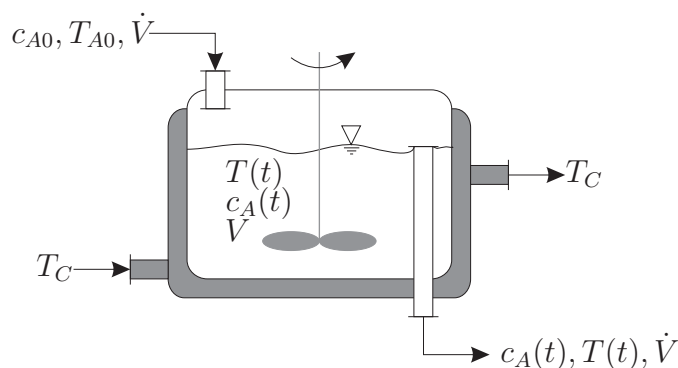


Abbildung 5: Schematischer Aufbau eines durchmischten Rührkesselreaktors (CSTR)

mit den Zuständen  $c_A, T \in \mathbb{R}$ , dem zu schätzenden Parameter  $k_0 \in \mathbb{R}$ , dem Eingang  $T_C \in \mathbb{R}$  und dem Ausgang  $T \in \mathbb{R}$ . Der Temperatursensor soll sich frei im Reaktionsgemisch bewegen können, und seine Datenpakete in Gegenwart von zufälligen Totzeiten  $\tau_i$  und zufälligen Paketverlusten zum Beobachter übertragen. Die Abtastung und damit das Senden eines Datenpaketes erfolgt immer dann, wenn sich die Temperatur um  $\Delta T = 10K$ , verglichen mit der letzten Übertragung, geändert hat, oder wenn  $\Delta t = 1.5s$  lang kein Datenpaket gesendet wurde. Für den Eingang  $T_C$  wurde eine flache Vorsteuerung so entworfen, dass die Konzentration  $c_A(t)$  einem gewünschten Sollverlauf  $c_{Af}(t)$  (hier: einem doppelten, stetigen Arbeitspunktwechsel) folgt. Das Ziel des Beobachters besteht nun darin, aus den Datenpaketen die Konzentration  $c_A(T)$ , die Temperatur  $T(t)$ , den unbekannt Parameter  $k_0$  und den unbekannt Zeitversatz  $t_o$  zu schätzen. Die Streckenparameter sowie die Anfangswerte und die Optimierungsparameter können aus den Tabellen 1 und 2 entnommen werden.

Systemparameter	$\frac{\dot{V}}{V}$	$\frac{\Delta H_R}{\delta \rho c_p}$	$\frac{\alpha A_c}{\rho c_p V}$	$\frac{E_A}{R}$	$c_{A0}$	$T_{A0}$
Wert	1	4250	2	5665	0.02	340

Tabelle 1: Streckenparameter für das Originalsystem und das Parallelsystem

Parameter (Strecke)	$c_A(0)$	$T_A(0)$	$k_0$	$t_o$
Wert	0.02	487.5635	$1 \cdot 10^6$	-1
Parameter (Beobachter)	$\hat{c}_A(0)$	$\hat{T}_A(0)$	$\hat{k}_0$	$N + 1$
Wert	0.01	400	$5 \cdot 10^5$	5
Optimierung (Gauß-Newton)	$c_1$	$c_2$	-	-
Wert	$10^{-4}$	0.9	-	-
Optimierung (Levenberg-Marquardt)	$\lambda_0$	$a_1$	$a_2$	<b>D</b>
Wert	0.2	0.2	10	multiplikativ

Tabelle 2: Anfangswerte und Optimierungsparameter

Abbildung 6 zeigt den Vergleich zwischen dem Konzentrationsverlauf (linker oberer Plot) und dem Temperaturverlauf (rechter oberer Plot) zwischen Originalsystem (durchgezogene Linie) und Parallelsystem (gestrichelte Linie). Die Abtastungen sind mit einem Kreuz gekennzeichnet, wenn das entsprechende Paket am Beobachter ankommt, bzw. mit einem Kreis, wenn ein Paketausfall statt gefunden hat. Der linke untere Plot zeigt den Paketverlauf im Netzwerk. Ein

mit einem Kreuz gekennzeichnetes Paket indiziert eine erfolgreiche Ankunft am Beobachter, ein Kreis hingegen einen Paketausfall. Die Sendezeitpunkte befinden sich auf der vertikalen Achse mit dem Wert 1, wohingegen die Ankunftszeitpunkte am Beobachter auf der vertikalen Achse mit dem Wert 0 sind. Ein Vergleich zwischen der geschätzten (weißer Balken) und der tatsächlichen Totzeit (schwarzer Balken) für ein Paket zeigt der rechte untere Plot.

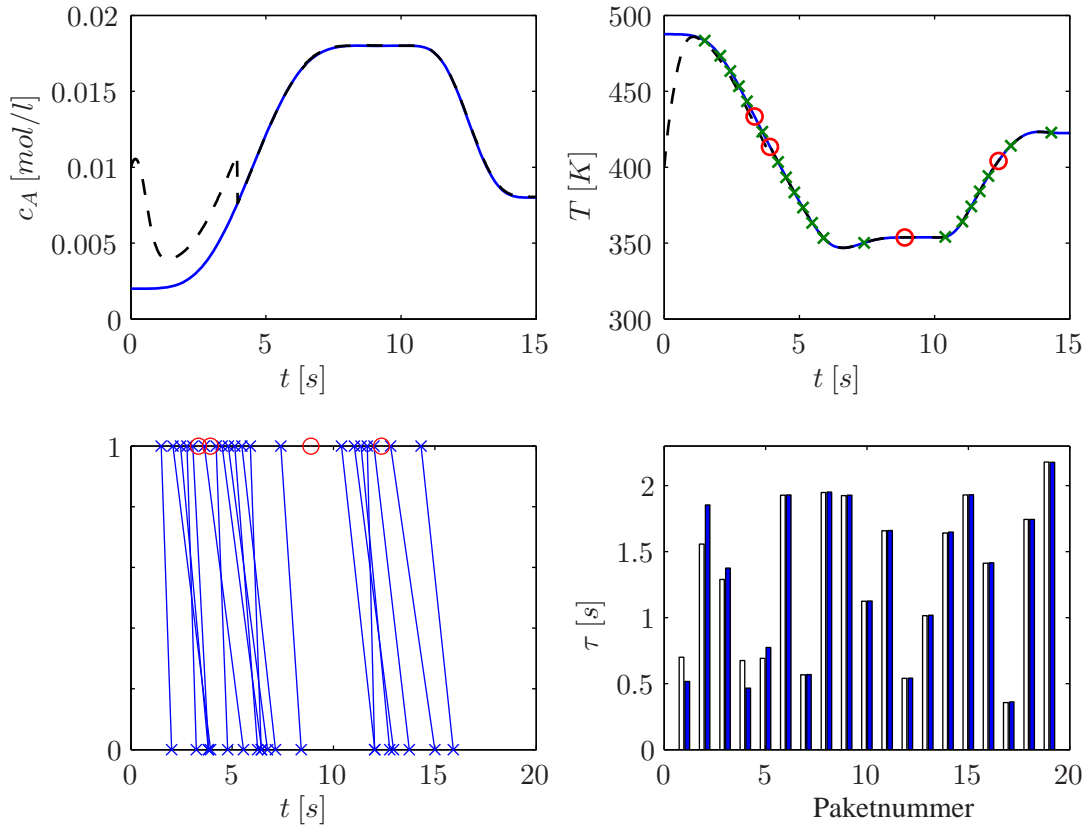


Abbildung 6: *oben links:* Vergleich der Konzentrationsverläufe, *oben rechts:* Vergleich der Temperaturverläufe, *unten links:* Paketverlauf im Netzwerk, *unten rechts:* Vergleich der Totzeiten

Der Unterschied zwischen den Verläufen der einzelnen Optimierungsverfahren sind minimal, so dass die Kurven in Abb. 6 für beide Optimierungsverfahren identisch sind. Ein Maß für die numerische Effizienz der Optimierungsverfahren ist die Anzahl der Gütemaßauswertung  $\#\Phi$  sowie die Anzahl der Jakobimatrixberechnungen  $\#J$  für den betrachteten Simulationszeitraum. Zudem geben die Performanceindexe

$$\gamma_1 = \frac{\int_{t_1}^{t_2} |c_A(t) - \hat{c}_A(t)| dt}{\int_{t_1}^{t_2} |c_A(t)| dt}, \quad \gamma_2 = \frac{\int_{t_1}^{t_2} |T(t) - \hat{T}(t)| dt}{\int_{t_1}^{t_2} |T(t)| dt}, \quad (40)$$

Aufschluss über die Qualität der Verfahren (s. Tab. 3), wobei  $t_1$  der Zeitpunkt der Ankunft des  $N + 1$ -ten Pakets und  $t_2$  der Zeitpunkt der Ankunft des letzten Pakets am Beobachter ist. Das Gauß-Newton Verfahren liefert hier qualitativ leicht bessere, dafür numerisch leicht ineffizientere Ergebnisse, verglichen mit dem Levenberg-Marquardt Verfahren.

	# $\Phi$	# $J$	$\gamma_1$	$\gamma_2$
Gauß-Newton Verfahren	84	40	1.257	0.091
Levenberg-Marquardt Verfahren	83	35	1.941	0.131

Tabelle 3: Vergleich der numerischen Güte der Optimierungsverfahren

## 6 Zusammenfassung

Dieser Beitrag behandelt den Entwurf eines optimierungsbasierten Beobachterverfahrens für nichtlineare NCS zur Zustands- und Parameterschätzung. Die übertragenen Datenpakete zwischen Sensor und Beobachter, die die Messwerte sowie die relativen Zeitstempel der unsynchronisierten Sensorzeit beinhalten, unterliegen dabei zufälligen und variablen Totzeiten sowie Paketverlusten. Dieses Beobachtungsproblem konnte als unrestringiertes Optimierungsproblem formuliert und anschließend effizient mit dem Gauß-Newton bzw. dem Levenberg-Marquardt Verfahren gelöst werden. Die dafür notwendige Modellberechnung mit der verbundenen Jakobimatrixberechnung wurde vorgestellt. Die Simulationsergebnisse bestätigen die Performance der vorgeschlagenen Methode.

## Literatur

- [HS97] HANSON, M. A. ; SEBORG, D.E: *Nonlinear Process Control*. Prentice Hall PTR, 1997
- [Lam97] LAMPTON, M.: Damping-undamping strategies for the Levenberg-Marquardt nonlinear least-squares method. In: *Computers in Physics* 11 (1997), Nr. 1, S. 110–114
- [MG95] MORAAL, P.E. ; GRIZZLE, J.W.: Observer design for nonlinear systems with discrete-time measurements. In: *IEEE Trans. Autom. Control* 40 (1995), March, Nr. 3, S. 395–404
- [MM95] MICHALSKA, H. ; MAYNE, D.Q.: Moving Horizon Observers and Observer-Based Control. In: *IEEE Trans. Autom. Control* 40 (1995), June, Nr. 6, S. 995–1006
- [NW99] NOCEDAL, J. ; WRIGHT, J.W.: *Numerical Optimization*. Springer-Verlag, 1999
- [RRM03] RAO, C.V. ; RAWLINGS, J.B. ; MAYNE, D.Q.: Constrained State Estimation for Nonlinear Discrete-Time Systems: Stability and Moving Horizon Approximations. In: *IEEE Trans. Autom. Control* 48 (2003), Nr. 2, S. 246–258
- [URP74] UPPAL, A. ; RAY, W.H. ; POORE, A.B: On the dynamic behavior of continuous stirred tank reactors. In: *Chemical Engineering Science* (1974)
- [Vor03] VORTISCH, J.: *Zustandsbeobachtung nichtlinearer Systeme auf bewegtem Horizont mit Hilfe des Gauss-Newton-Verfahrens*, Universität Bremen, Diss., 2003
- [Zim97] ZIMMER, G.: *Zustandsbeobachtung nichtlinearer Systeme*. Shaker Verlag, 1997