

Modellfreier, datenbasierter Ansatz zur Selbsteinstellung von Vorsteuerungen am Beispiel der Abgasturboladerregelung

René Noack, Universität Rostock
rene.noack@uni-rostock.de

Torsten Jeinsch, Universität Rostock
Nick Weinhold, IAV GmbH

Zusammenfassung

Dieser Beitrag stellt einen Ansatz zur Selbsteinstellung bestehender Vorsteuerungen basierend auf der Methodik iterativ lernender Regelungen (ILR) vor und vergleicht verschiedene ILR-Verfahren. Die Gegenüberstellung erfolgt beginnend von den klassischen ILR-Ansätzen (P-ILR und D-ILR) über normoptimierten und regulierten oder gradientenbasierten ILR-Verfahren bis hin zu stochastischen ILR-Methoden. Die gezeigten Ergebnisse werden am Beispielprozess: Ladedruckregelung eines Abgasturboladers ermittelt.

1 Einleitung

Mit steigender Rechenleistung lassen sich immer komplexere und leistungsfähigere Steuerungs- und Regelungsstrukturen und Verfahren realisieren. Oft ist die Anpassung oder Parametrierung der Regelungsstrukturen an den Prozess eine Herausforderung, welche viel Arbeitszeit und hohe Applikationskosten verursacht und ein hohes Maß an Expertenwissen voraussetzt. Verfahren zur Selbsteinstellung können diese Kosten verringern und zur Unterstützung der Applikateure dienen. Auf diesem Gebiet existieren viele verschiedene Ansätze mit zum Teil sehr unterschiedlichen Anforderungen und Anwendungsfällen. Ist der Aufwand für den Reglerentwurf z.B. auf Grund komplexer Modellbildung sehr hoch, werden die Parameter ausgehend von einer Startparametrierung durch den Applikateur iterativ in mehreren Versuchen eingestellt. Die Änderung der Parameter erfolgt dabei durch die Bewertung eines vorhergehenden Regelvorganges. Dies lässt sich auch automatisch durchführen. Dazu zählen Verfahren der Iterativ Lernenden Regelung (ILR) [3, 4], des Iterative Feedback Tuning (IFT) [5] und Methoden des Control Performance Monitoring (CPM) [1, 2].

Das in diesem Beitrag beschriebene Verfahren dient zur Unterstützung der Applikation von Parametern in Steuergeräten von Ottomotoren. Der betrachtete Teilprozess ist die Aufladung des Ottomotors durch einen Monoturbolader mit pneumatischem Wastegate. Dabei muss dieses Verfahren folgende Anforderungen erfüllen:

- Ein verständliches und intuitives Arbeitsprinzip der Selbsteinstellung ist wichtig, um die Akzeptanz zu erhöhen.
- Es darf kein zusätzlicher Arbeitsaufwand entstehen. Daher sind modellierungsfreie und datenbasierte Verfahren zu verwenden.
- Die Selbsteinstellung erfolgt mittels normaler Betriebsdaten. Spezielle Sollwertanforderungen wie beim Iterative Feedback Tuning (IFT) oder definierte Störungen wie bei der Verwendung von Methoden des Control Performance Monitoring (CPM) sind nicht erforderlich.
- Es sind hohe Lernraten bei gleichzeitiger Gewährleistung einer kontinuierlichen Parameterverbesserung zu erreichen, um die Applikationszeit zu minimieren.

Dieser Beitrag ist wie folgt aufgebaut. Abschnitt 2 stellt kurz den Prozess und das betrachtete Selbsteinstellungsverfahren vor. Danach beschreibt Abschnitt 3 verschiedene ILR-Ansätze, gefolgt von der Analyse dieser Ansätze in Abschnitt 4. Den Abschluss bildet die Zusammenfassung in Abschnitt 5.

2 Problemstellung und Vorbetrachtung

Dieser Abschnitt gibt einen kurzen Überblick zum betrachteten Prozess und dem Ansatz der Selbsteinstellung. Eine ausführlichere Darstellung des ILR-basierten Ansatzes zur Selbsteinstellung ist in [11, 12] beschrieben und soll im Folgenden kurz dargestellt werden.

2.1 Allgemeine Prozessbeschreibung

Der Ausgangspunkt ist die folgende allgemeine Beschreibung eines nichtlinearen diskreten Systems mit einem Ausgang und einem Eingang

$$x(i+1) = f_1(x(i), u(i), d(i)) \quad (1)$$

$$y(i) = f_2(x(i), u(i)) \quad (2)$$

mit den Zustandsgrößen x , der Eingangsgröße u , den Störungen d und der Ausgangsgröße y . Die Funktionen f_1 und f_2 stellen nichtlineare Zusammenhänge dar. Dieser Prozess verläuft im Zeitintervall $[0, T]$ und wird zyklisch wiederholt. Die nichtlineare Funktion f_3 repräsentiert die bestehende Vorsteuerung

$$u(i) = f_3(x(i), y_d(i), d(i), \hat{\boldsymbol{p}}) \quad (3)$$

mit dem Sollwert y_d und dem Parametervektor \hat{p} . Die Aufgabe ist die automatisierte iterative Einstellung von \hat{p} .

2.2 Idee der ILR-basierten Selbsteinstellung

Eine ILR ist die iterative Optimierung von Prozessen mit immer wiederkehrenden, sich wiederholenden Regelaufgaben. [6] weist als Beispiele Robotersteuerungen, Batch-Reaktoren und Festplattenantriebe aus. Hierzu erfolgt die Ausgabe des Verlaufes der Stellgröße für den wiederkehrenden Zyklus k im Zeitraum $[0, T]$ mit N Abtastungen. Nach dem Beenden des Zyklus erfolgt die Speicherung der aufgetretenen Regelabweichung und die Berechnung bzw. Anpassung des Stellgrößenverlaufes für den nächsten Zyklus $k + 1$. Der Standard-ILR-Ansatz ist

$$\mathbf{u}_{k+1} = \mathbf{Q}\mathbf{u}_k + \mathbf{L}\mathbf{e}_k \quad (4)$$

$$\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k \quad (5)$$

$$\mathbf{u}_{k+1} \in \mathbb{R}^{N \times 1}, \quad \mathbf{u}_k \in \mathbb{R}^{N \times 1}, \quad \mathbf{e}_k \in \mathbb{R}^{N \times 1}, \quad \mathbf{y}_k \in \mathbb{R}^{N \times 1}, \quad \mathbf{y}_d \in \mathbb{R}^{N \times 1},$$

mit den Lernmatrizen \mathbf{Q} und \mathbf{L} , der Anzahl der Abtastpunkte N , dem Stellgrößenvektoren \mathbf{u}_k und \mathbf{u}_{k+1} , dem Vektor der Regelabweichung \mathbf{e}_k , dem Ausgangsgrößenvektor \mathbf{y}_k und dem Sollwertvektor \mathbf{y}_d . Diese Vektoren beinhalten die gemessenen Daten des k -ten Zyklus.

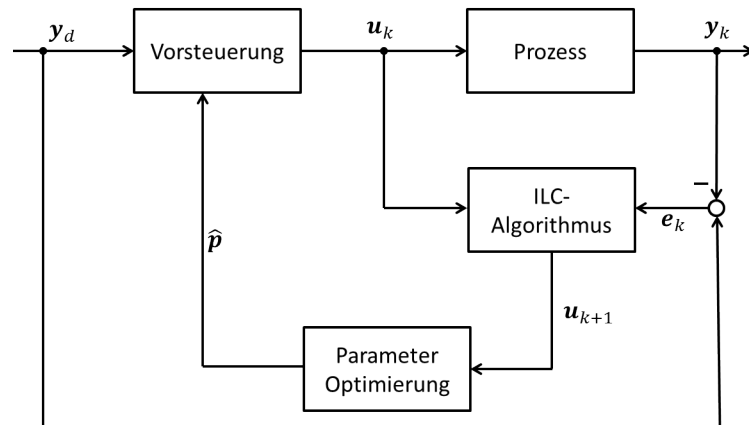


Abbildung 1: Strukturbild des ILR-basierten Selbsteinstellungsverfahrens

Die Grundidee des Verfahrens ist, die Optimierung des Stellgrößenverlaufes aus der ILR-Theorie zu verwenden, um die Einstellparameter der bestehenden Vorsteuerung zu identifizieren. Dadurch ist es möglich die Randbedingungen einer ILR (z.B. gleiche Sollwertverläufe, gleiche Intervalllänge und konstante Startbedingungen in jeder Iteration), unter der Voraussetzung einer monotonen Verbesserung der Parameter, zu umgehen. In Abb. 1 ist die Struktur des ILR-basierten Selbsteinstellungsverfahrens dargestellt. Der Index k kennzeichnet die aktuelle Messung und $k + 1$ den Optimierungsschritt der nächsten Iteration. Der ILR-Block berechnet nach dem gewählten ILR-Verfahren den optimierten Stellgrößenverlauf \mathbf{u}_{k+1} aus der gemessenen Größen \mathbf{u}_k , \mathbf{y}_k und \mathbf{y}_d . Dabei kann sich

\mathbf{y}_d auf Grund verschiedener Umgebungsbedingungen und Messfahrten in jeder Iteration ändern. Mit Hilfe von \mathbf{u}_{k+1} und y_d lässt sich eine Parameteroptimierung durchführen, die $\hat{\mathbf{p}}$ so bestimmt, dass die Differenz zwischen dem ILR-Stellgrößenverlauf \mathbf{u}_{k+1} und dem $\hat{\mathbf{p}}$ -abhängigen Stellgrößenverlauf der Vorsteuerung $\mathbf{u}_{vor}(\hat{\mathbf{p}})$ minimiert wird.

$$\hat{\mathbf{p}} = \arg \min_{\hat{\mathbf{p}}} (\|\mathbf{u}_{k+1} - \mathbf{u}_{vor}(\hat{\mathbf{p}})\|_2) \quad (6)$$

mit $\hat{\mathbf{p}} \in \mathbb{R}_+^*$.

3 Vorstellung verschiedener ILR-Verfahren

Dieser Abschnitt stellt verschiedene ILR-Verfahren vor und gibt einen Überblick über deren Ansätze und Rahmenbedingungen.

3.1 Klassische ILR-Ansätze

Zu den allerersten ILR-Verfahren zählt der PD-Typ ILR-Algorithmus [10, 3].

$$\mathbf{u}_{k+1} = \mathbf{u}_k + P\mathbf{e}_k + D\Delta\mathbf{e}_k \quad (7)$$

Bei diesen Ansätzen sind die Lernfaktoren P und D konstant über alle Iterationen und alle Zeitschritte. Daher erfolgt die Festlegung dieser Parameter relativ konservativ, um die Konvergenz zu gewährleisten.

3.2 Normoptimierte ILR-Ansätze

Der Ausgangspunkt der Betrachtung ist hier die Verwendung des Impulsantwortmodells mit der Eingangs-Ausgangs-Matrix \mathbf{G} , welche den Markov-Parametervektor enthält [3, 4].

$$\mathbf{y}_k = \mathbf{G}\mathbf{u}_k \quad (8)$$

Die Matrix \mathbf{G} lässt sich aus der Zustandsbeschreibung bzw. der Übertragungsfunktion herleiten. Da für das Selbsteinstellungsverfahren kein Modellierungsaufwand entstehen darf, erfolgt die Bestimmung der Matrix \mathbf{G} aus Messsignalen [3, 11, 12].

Verwendung der Pseudoinversen [3]: Der erste Ansatz ist die Bildung der Pseudoinversen des Systems [3]. Ausgehend vom allgemeinen ILR-Stellgesetz

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \delta(\mathbf{u}_k) \quad (9)$$

und dem verwendeten Gütekriterium

$$J(\delta(\mathbf{u}_k)) = \|\mathbf{G}\delta(\mathbf{u}_k)\|_2^2 \quad (10)$$

ergibt sich der ILR-Algorithmus

$$\mathbf{u}_{k+1} = \mathbf{u}_k + (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{e}_k. \quad (11)$$

Pseudoinverse mit Regularisierung [3]: Die Verwendung der Pseudoinversen führt zu großen Verstärkungen hoher Frequenzen, was insbesondere bei auftretendem Mess- und Prozessrauschen nachteilig ist. Daher ist es sinnvoll eine Regularisierung durchzuführen, was zu einer Erweiterung des Gütekriteriums (9) führt

$$J = \|\mathbf{G}\delta(\mathbf{u}_k)\|_2^2 + \eta_1 \|\delta(\mathbf{u}_k)\|_2^2 + \eta_2 \|\mathbf{E}\delta(\mathbf{u}_k)\|_2^2 \quad (12)$$

Durch den Wichtungsfaktor η_1 lässt sich die Änderung der Stellgröße berücksichtigen. Wichtungsfaktor η_2 wichtet die Änderungsrate. Die Matrix $\mathbf{E} \in \mathbb{R}^{N \times N}$ stellt dabei den diskreten Ableitungsoperator 1. Ordnung dar. Für den ILR-Algorithmus ergibt sich damit

$$\mathbf{u}_{k+1} = \mathbf{u}_k + (\mathbf{G}^T \mathbf{G} + \eta_1 \mathbf{I} + \eta_2 \mathbf{E}^T \mathbf{E})^{-1} \mathbf{G}^T \mathbf{e}_k. \quad (13)$$

Gradientenbasierter ILR-Ansatz [3, 12]: Eine weitere Möglichkeit ist ein gradientenbasierter Ansatz. Ausgehend vom ILR-Ansatz

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \epsilon_{k+1} \mathbf{G}^T \mathbf{e}_k, \quad (14)$$

lässt sich der Skalierungsfaktor ϵ_{k+1} so bestimmen, dass er in jeder Iteration den größtmöglichen Lernfaktor mit monotoner Konvergenz gewährleistet. Aus

$$J(\epsilon_{k+1}) = \|\mathbf{e}_{k+1}\|_2^2 + w\epsilon_{k+1}^2 \quad (15)$$

ergibt sich

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \frac{\|\mathbf{G}^T \mathbf{e}_k\|_2^2}{w + \|\mathbf{G}\mathbf{G}^T \mathbf{e}_k\|_2^2} \mathbf{G}^T \mathbf{e}_k \quad (16)$$

Der Lernfaktor ϵ_{k+1} ändert sich in jeder Iteration, ist aber für jeden Zeitschritt i konstant. Durch den Faktor w lassen sich Modellungenauigkeiten berücksichtigen.

3.3 MFAC basierte ILR-Ansätze

Model Free Adaptive Control (MFAC) gehört zu den datenbasierten adaptiven Regelverfahren für nichtlineare Systeme [9]. Die Idee hinter diesem Verfahren ist die Verwendung der Pseudo-Partiellen-Ableitung (engl. PPD) mit der das nichtlineare System als zeitvarianten lineares System betrachtet wird. In [9] sind drei Formen der PPD aufgeführt. Es han-

delt sich dabei um die Compact Form Dynamic Linearization (CFDL), die Partial Form Dynamic Linearization (PFDL) und die Full Form Dynamic Linearization (FFDL). Neben adaptiven Regelalgorithmen sind in [9] prädiktive MFAC-Verfahren und ILR-Verfahren für Ein- und Mehrgrößensysteme aufgeführt. An dieser Stelle soll die einfachste MFAC-ILR-Form (CFDL) dargestellt werden, die ein Spezialfall der PFDL und FFDL ist. Der allgemeine CFDL-Modellansatz lautet wie folgt:

$$\Delta y_k(i+1) = \phi_{C,k}(i)\Delta u_k(i), \forall i \in \{0, 1, \dots, N\} \quad (17)$$

mit der Änderung der Ausgangsgröße $\Delta y_k(i+1) = y_k(i+1) - y_k(i)$, der Änderung der Eingangsgröße $\Delta u_k(i) = u_k(i) - u_k(i-1)$, dem iterationsabhängigen zeitvariablen PPD-Parameter $\phi_{C,k}(i)$, dem diskreten Zeitschritt i und der Iteration k . Der MFAC-Algorithmus besteht aus 2 Schritten je Iteration. Im ersten Schritt erfolgt die rekursive Berechnung der PPD-Schätzung $\hat{\phi}_{C,k+1}(i)$ für jeden Zeitpunkt i . Ausgehend vom Gütekriterium

$$J(\hat{\phi}_{C,k+1}(i)) = \left(\Delta y_k(i+1) - \hat{\phi}_{C,k+1}(i)\Delta u_k(i)\right)^2 + \mu \left(\hat{\phi}_{C,k+1}(i) - \hat{\phi}_{C,k}(i)\right)^2 \quad (18)$$

lässt sich die Berechnungsvorschrift für den PPD-Faktor $\hat{\phi}_{C,k+1}(i)$ herleiten

$$\hat{\phi}_{C,k+1}(i) = \hat{\phi}_{C,k}(i) + \frac{\Delta u_k(i)}{\mu + (\Delta u_k(i))^2} (\Delta y_k(i+1) - \Delta \hat{y}_k(i+1)) \quad (19)$$

mit $\Delta \hat{y}_k(i+1) = \hat{\phi}_{C,k}(i)\Delta u_k(i)$. Der zweite Schritt ist die Berechnung der ILR-Stellgröße $u_{k+1}(i)$ für die nächste Iteration. Durch die Festlegung des Gütekriteriums

$$J(u_{k+1}(i)) = (e_{k+1}(i+1))^2 + \lambda (u_{k+1}(i) - u_k(i))^2 \quad (20)$$

ergibt sich der MFAC-ILR Algorithmus zu

$$u_{k+1}(i) = u_k(i) + \frac{\hat{\phi}_{C,k}(i)}{\lambda + (\hat{\phi}_{C,k}(i))^2} e_k(i+1). \quad (21)$$

Die Faktoren μ und λ stellen Tuning-Faktoren dar. Für den Nachweis der Stabilität sei auf [9] verwiesen. Es ist zu erkennen, dass es sich hier um einen zeitvariablen ILR-Ansatz handelt, da sich die Bewertung der Regelabweichung in jedem Zeitschritt i ändert.

3.4 Stochastischer ILR-Ansatz

Eine weitere Möglichkeit des ILR-Entwurfes sind stochastische Methoden und Algorithmen. Solche stochastischen iterativ lernenden Regler sind ILR für Systeme mit stochastischen Signalen wie Prozessrauschen, Messrauschen und Initialisierungsfehlern [7, 8]. Ein Beispiel ist der Kalman-Filter basierte Ansatz. Der Prozess wird hier als ein diskretes,

lineares, zeitvariantes System betrachtet:

$$x_k(i+1) = A(i)x_k(i) + B(i)u_k(i) + w_k(i) \quad (22)$$

$$y_k(i) = C(i)x_k(i) + v_k(i) \quad (23)$$

mit dem Zustandsvektor $x_k(i) \in \mathbb{R}^n$, dem Prozessrauschen $w_k(i) \in \mathbb{R}^n$, dem Ausgangssignal $y_k(i) \in \mathbb{R}^p$, dem Messrauschen $v_k(i) \in \mathbb{R}^p$ und dem Systemeingang $u_k(i) \in \mathbb{R}^m$. Die Rauschsignale werden als mittelwertfreies weißes gaußsches Rauschen angenommen. Dabei ist $\mathbb{E}(\mathbf{w}_k, \mathbf{w}_k^T) = \mathbf{Q}$ eine positive semidefinite Matrix und $\mathbb{E}(\mathbf{v}_k, \mathbf{v}_k^T) = \mathbf{R}$ eine positive definite Matrix. Die Kovarianzmatrix zwischen dem Prozessrauschen und dem Messrauschen ist null, diese Rauschsignale sind unabhängig voneinander. [8] und [7] definieren den Zustandsfehlervektor $\delta x_k(i) = x_d(i) - x_k(i)$ und den Eingangsfehlervektor $\delta u_k(i) = u_d(i) - u_k(i)$ ausgehend von der realisierbaren Referenztrajektorie $y_d(i)$ und dem eindeutigen Eingangssignal $u_d(i)$. Ziel ist die Minimierung der Kovarianzmatrix des Eingangsfehlervektors $\delta u(i, k)$, welche trotz auftretender Rauschsignale gegen null konvergieren soll. In diesem Fall wäre die realisierbare Referenztrajektorie $y_d(i)$ vom Prozess erreicht. Ausgehend von $P_{u,k+1} = \mathbb{E}(\delta u_{k+1}(i)\delta u_{k+1}^T(i))$ lässt sich laut [7, 8] folgende rekursive Berechnung finden

$$K_k(i) = P_{u,k}(i)B(i)^T C^T(i+1)[C(i+1)B(i)P_{u,k}(i)B^T(i)C^T(i+1) + C(i+1)QC^T(i+1) + R] \quad (24)$$

$$u_{k+1}(i) = u_k(i) + K_k(i)e_k(i+1) \quad (25)$$

$$P_{u,k+1}(i) = [I - K_k(i)C(i+1)B(i)] P_{u,k}(i) \quad (26)$$

Neben der Kenntnis des Prozessrauschens, des Messrauschens, der zeitvariablen Matrizen $B(i)$ und $C(i)$ ist die Definierung der Initialisierungskovarianzmatrix des Eingangsfehlervektors $P_{u,0}(i)$ notwendig.

4 Analyse und Vergleich der ILR-Verfahren

Dieser Abschnitt vergleicht die Ergebnisse der aufgeführten ILR-Algorithmen. Dazu wird das vorgestellte Selbsteinstellungsverfahren mit den jeweiligen ILR-Algorithmen an einem Simulationsmodell eines Ottomotors mit Abgasturbolader und pneumatischem Wastegate getestet [11, 12]. In jeder Iteration variieren der Anfangsparameter Startdrehzahl und der Verlauf des Sollwertes. Weiterhin sind alle gemessenen Signale von einem Messrauschen überlagert.

Die Abbildungen 2 bis 6 zeigen die Ergebnisse der Optimierung nach 50 Iterationen. Diese Abbildungen sind wie folgt aufgebaut. Das linke obere Diagramm zeigt den Verlauf des Sollwertes y_d (grün), den Verlauf der Regelgröße y_0 bei Beginn der Optimierung und den Verlauf der Regelgröße y_Z nach Ende der Optimierung. Das linke untere Diagramm

stellt die dazugehörigen Stellgrößenverläufe u_0 und u_Z dar. Im mittleren Diagramm ist der Verlauf der Regelfehler e_0 und e_Z abgebildet. Das rechte Diagramm stellt den Ausgangsparametervektor \hat{p}_0 und das Optimierungsergebnis dar. Im Zeitpunkt $t = 20s$ erfolgt der Start des Regelvorganges durch einen Sprung des Sollladedruckes. Zu Beginn ist das Stellglied in seiner maximalen Begrenzung, so dass hier nicht die Regelabweichung minimiert werden kann. Zu sehen ist, dass alle Verfahren ein ähnliches Optimierungsziel erreichen.

Die Abbildung 7 stellt die Lernrate und die Konvergenz der einzelnen ILR-Ansätze dar. Bei dem dargestellten Verlauf handelt es sich um die Entwicklung der 2-Norm des Regelfehlers in jeder Iteration. Für eine bessere Vergleichbarkeit sind die Werte auf den Ausgangsfehlerwert normiert und in Prozent dargestellt. Es zeigt sich, dass die klassische PD-ILR sehr viele Iterationen benötigt und der stochastische Ansatz eine zu hohe Streuung im Optimierungsbereich aufweist. Der gradientenbasierte ILR-Ansatz und der MFAC(CFDL) basierte ILR-Ansatz benötigen die wenigsten Iterationen und verursachen dadurch den geringsten Zeitaufwand beim Einstellen des gesuchten Parametervektors \hat{p} .

Neben den Konvergenzeigenschaften sind weitere Eigenschaften für eine Bewertung wichtig. Die Tabelle 1 zeigt eine Vergleichsübersicht aller hier angeführten ILR-Ansätze über die geforderten Eigenschaften des Selbsteinstellungsverfahrens. Die Einteilung beginnt bei '---' für sehr ungeeignet endet bei '++' für sehr geeignet.

Tabelle 1: Vergleich der ILR-Verfahren

| Verfahren | Lernrate | Konvergenz | Tuning-Parameter | Systemwissen | Rausch-anfälligkeit |
|------------------------------------|----------|------------|------------------|--------------|---------------------|
| Klassischer P-ILR | -- | - | ++ | + | - |
| Klassischer PD-ILR | -- | O | + | + | - |
| ILR mit Pseudoinverse | ++ | ++ | ++ | O | -- |
| ILR mit Regularisierung | + | + | + | O | + |
| Gradienten-basierter ILR-Ansatz | ++ | ++ | ++ | O | O |
| MFAC-ILR (CFDL) | ++ | + | O | ++ | + |
| MFAC-ILR (PFDL) | O | + | -- | ++ | + |
| MFAC-ILR (FFDL) | O | + | -- | ++ | + |
| Stochastischer ILR-Ansatz (Kalman) | + | O | O | - | ++ |

- **Lernrate:** Aussage wie viele Iterationen für die Selbsteinstellung notwendig sind.
- **Konvergenz:** Bewertet die erreichte Genauigkeit und die Streuung im Zielbereich.
- **Tuningparameter:** Wieviel einstellbare Tuningparameter hat das Verfahren?

- **Systemwissen:** Wieviel Prozesswissen ist notwendig?
- **Rauschanfälligkeit:** Bewertet den Einfluss von Mess- und Prozessrauschen.

Neben der Betrachtung der Konvergenzeigenschaften geht in die Bewertung die Anzahl der einzustellenden Tuningparameter, sowie das benötigte System- und Prozesswissen des jeweiligen ILR-Ansatzes ein. Hier zeigt der stochastische Ansatz den größten Nachteil, da die zeitvariable Eingangsmatrix $B(i)$ und die zeitvariable Ausgangsmatrix $C(i)$ einen notwendigen Bestandteil darstellen. Hier haben die MFAC-basierten Ansätze den größten Vorteil, da der PPD in jeder Iteration rekursiv geschätzt wird. Alle normoptimierten Ansätze benötigen die Eingangs-Ausgangs-Matrix \mathbf{G} , welche datenbasiert und ohne zusätzlichen Modellierungsaufwand aus den Messsignalen ermittelt werden muss. Bei den MFAC-basierten ILR höherer Ordnung (PFDL und FFDL) nimmt mit der Ordnung auch die Zahl der einzustellenden Tuningparameter zu, was den Einsatz des Selbsteinstellungsverfahrens kompliziert.

Die Ansätze mit den meisten Vorteilen für das beschriebene Selbsteinstellungsverfahren stellen somit der gradientenbasierte ILR-Ansatz und der MFAC(CFDL)-ILR-Ansatz dar.

5 Zusammenfassung

Dieser Beitrag stellt ein ILR-basiertes Einstellungsverfahren für bestehende Vorsteuerungen vor und vergleicht dabei verschiedene ILR-Ansätze. Die dargestellten Simulationsergebnisse zeigen die Möglichkeiten dieses Selbsteinstellungsverfahrens und die Auswirkungen des jeweiligen ILR-Ansatzes. Die empfohlenen ILR-Ansätze sind der gradientenbasierte ILR-Ansatz und der MFAC(CFDL)-ILR-Ansatz.

Literatur

- [1] M. Jelali und B. Huang: *Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods*, Springer Verlag, 2010.
- [2] B. Huang und R. Kadali: *Dynamic Modeling, Predictive Control and Performance Monitoring: A-Data-driven Subspace Approach*, Springer Verlag, 2008.
- [3] H.-S. Ahn, K.L. Moore und Y.Q. Chen: *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*, Springer Verlag, 2007.
- [4] E. Rogers, K. Galkowski und D.H. Owens: *Control Systems Theory and Applications for Linear Repetitive Processes*, Springer Verlag, 2007.
- [5] H. Hjalmarsson, M. Gevers, S. Gunnarson und O. Lequin: *Iterative feedback tuning: theory and applications*, Control Systems IEEE, 18:4, S. 26-41, 1998.

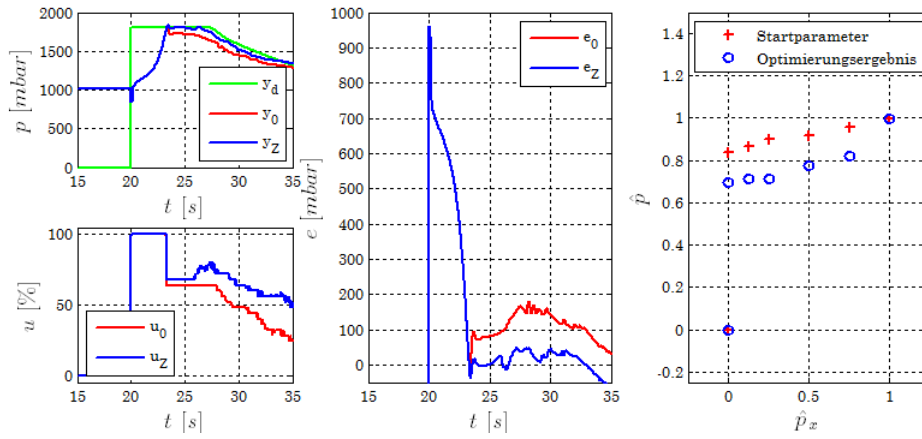


Abbildung 2: Ergebnis PD-ILR

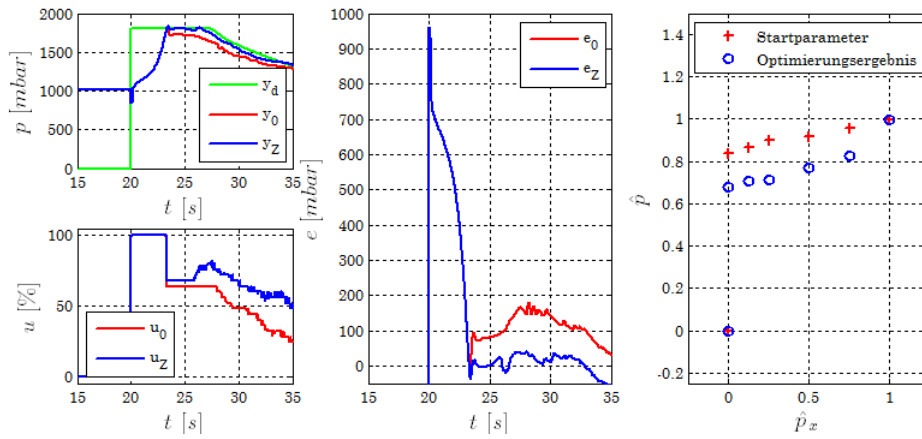


Abbildung 3: Ergebnis ILR: Pseudoinverse mit Regularisierung

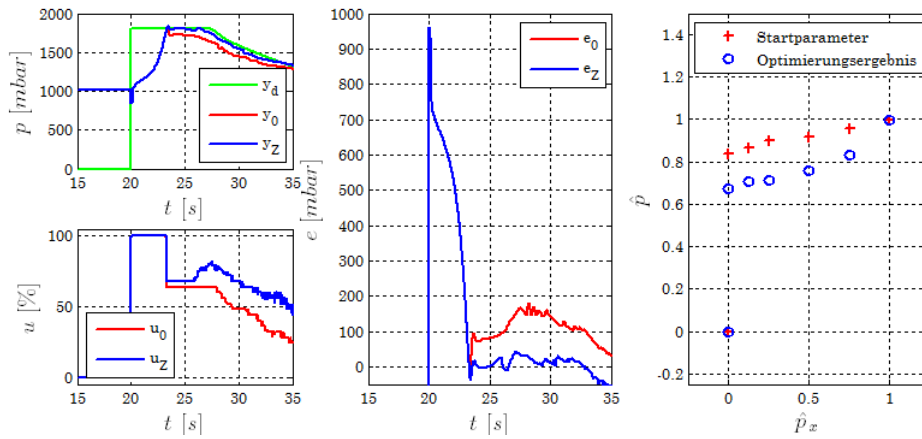


Abbildung 4: Ergebnis gradientenbasierte ILR

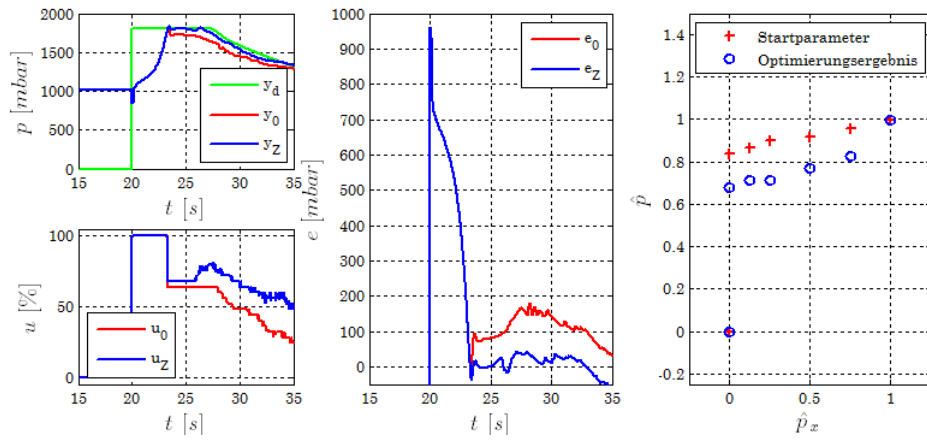


Abbildung 5: Ergebnis MFAC (CFDL) basierte ILR

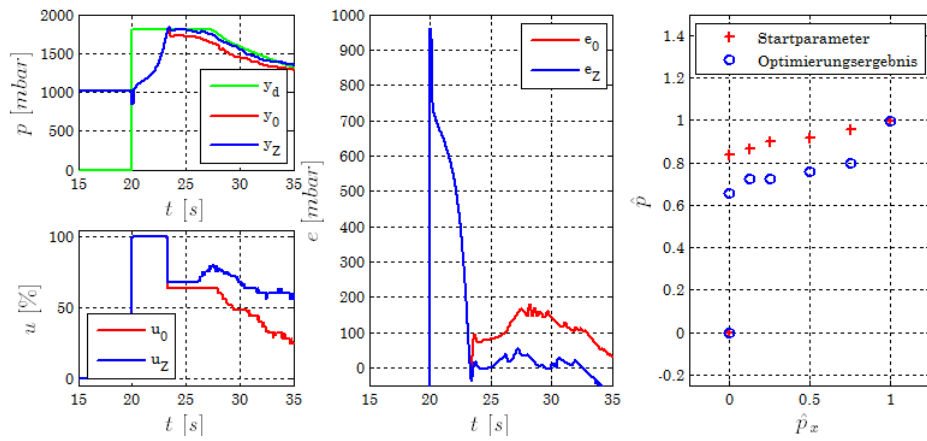


Abbildung 6: Ergebnis stochastische ILR (Kalman-Ansatz)

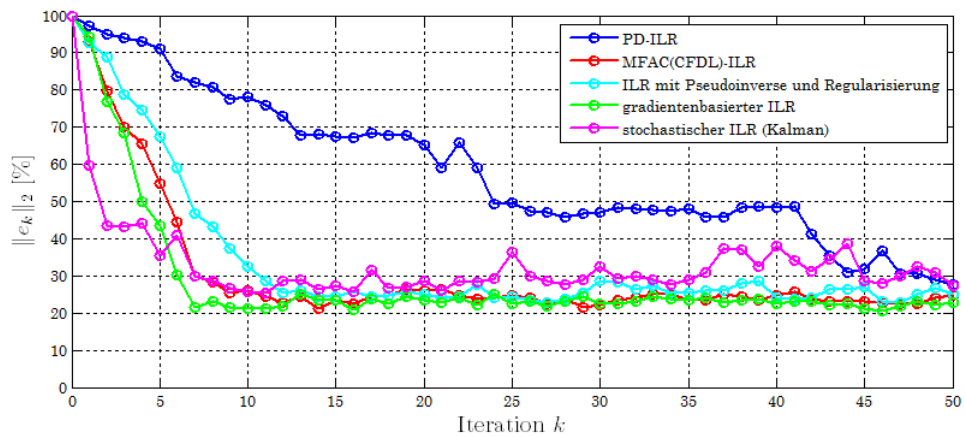


Abbildung 7: Konvergenzvergleich

- [6] K.L. Moore, Y. Chen und H.-S. Ahn: *Iterative learning control: A tutorial and big picture view*, IEEE Conference on Decision and Control, S. 2352-2357, 2006.
- [7] W. Shen und Y. Wang: *Survey on stochastic iterative learning control*, Journal of Process Control, 2014.
- [8] S.S. Saab: *Stochastic P-type/D-type iterative learning control algorithms*, International Journal of Control, 2003, 76:2, S. 139-148, 2003.
- [9] Z. Hou und S. Jin: *Model Free Adaptive Control: Theory and Applications*, CRC Press, 2013.
- [10] S. Arimoto, S. Kawamura und F. Miyazaki: *Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronic systems*, IEEE Conference on Decision and Control, S. 1064-1069, 1984.
- [11] R. Noack, T. Jeinsch und N. Weinhold: *Datengestützte selbsteinstellende Regelungen mittels Verfahren iterativ lernender Regelungen*, AALE - Angewandte Automatisierungstechnik in Lehre und Entwicklung (Regensburg), Deutscher Industrie-Verlag GmbH, ISBN: 978-3-8356-7142-3, S. 51-59, 2014.
- [12] R. Noack, T. Jeinsch, A. Haghani und N. Weinhold: *Data-Driven Self-Tuning Control by Iterative Learning Control with Application to Optimize the Control Parameter of Turbocharged Engines*, ISIE - International Symposium on Industrial Electronics (Istanbul), IEEE Catalog Number: CFP14ISI-USB, ISBN: 978-1-4799-2398-4, S. 2435-2440, 2014.