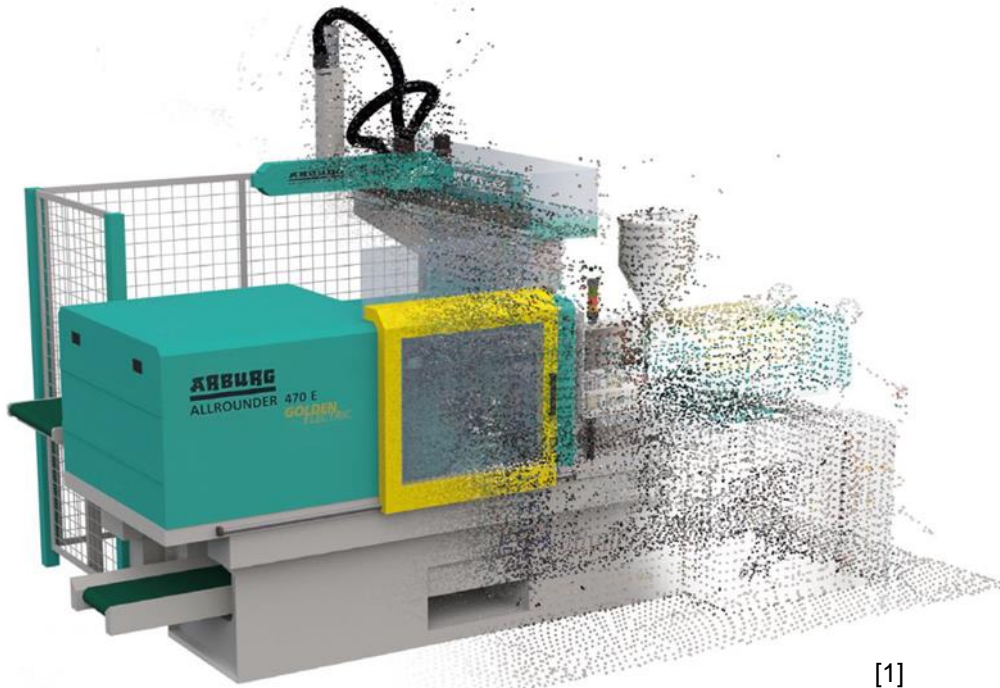


Digital Twin of Injection Molding



[1]

Leitfaden

Modellbasierte Optimierung von Spritzgießprozessen

Alexander Rehmer

Universität Kassel

Fachgebiet Mess- und Regelungstechnik

Optimierungsproblem

In dem vorliegenden Leitfaden wird davon ausgegangen, dass das Ziel der Optimierung die Ermittlung einer optimalen Eingangsgröße \mathbf{u}^{opt} sei, sodass ein vom Nutzer spezifizierter Soll- oder Referenzwert für die Ausgangsgröße \mathbf{y}^{ref} erreicht wird. Ziel der Optimierung ist somit den mittels einer Kostenfunktion \mathcal{L} gewichteten Fehler $e = \mathbf{y}^{\text{ref}} - \mathbf{y}$ zu minimieren. Im Rahmen einer modellbasierten Optimierung wird das Modell verwendet, um das Ausgangssignal \mathbf{y} zu präzisieren. Der prädisierte Systemausgang ist $\hat{\mathbf{y}}$. Da das identifizierte Modell den funktionalen Zusammenhang zwischen Eingangsgröße \mathbf{u} und Ausgangssignal \mathbf{y} herstellt $\hat{\mathbf{y}} = \hat{\mathbf{y}}(\mathbf{u})$, kann dieses verwendet werden, um das Optimierungsproblem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u}} \mathcal{L}(\mathbf{y}^{\text{ref}} - \hat{\mathbf{y}}(\mathbf{u}))$$

unter den Nebenbedingungen $\mathbf{u}_{\text{lb}} \leq \mathbf{u} \leq \mathbf{u}_{\text{ub}}$

$$\mathbf{g}_{\text{lb}} \leq \mathbf{g}(\mathbf{u}) \leq \mathbf{g}_{\text{ub}}$$

zu lösen. Die durch Lösung des Optimierungsproblems ermittelten optimalen Eingangsgrößen \mathbf{u}^{opt} werden dann auf den realen Prozess appliziert. Im Rahmen der Modellgenauigkeit wird \mathbf{y} dann \mathbf{y}^{ref} entsprechen. Der gesamte Ablauf ist übersichtsartig in Abbildung 1 dargestellt.

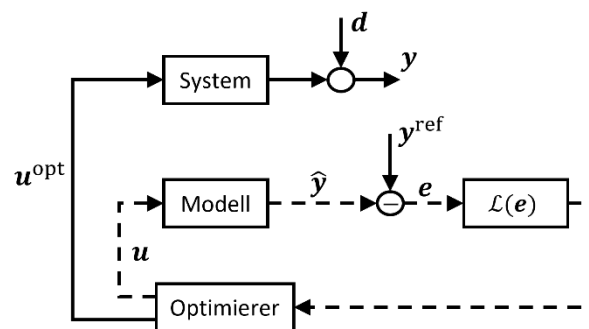


Abbildung 1: Ablauf der modellbasierten numerischen Optimierung

Bei dem Modell kann es sich entweder um ein statisches oder ein dynamisches Modell handeln. Bei einem dynamischen Modell sind die Eingangs- und damit Optimierungsgrößen Trajektorien, d.h. $\mathbf{u} = \{\mathbf{u}_k\}_{k=1}^N$ mit $\mathbf{u}_k \in \mathbb{R}^{n_u}$ wobei n_u die Anzahl der Eingangsgrößen ist, k ist die diskrete Zeit und N die Anzahl an Abtastpunkten, d.h. $\mathbf{u} \in \mathbb{R}^{N \times n_u}$. Bei einem statischen Modell hingegen sind die Eingangs- und damit Optimierungsgrößen nur vektorielle Größen, d.h. $\mathbf{u} \in \mathbb{R}^{n_u}$.

Dynamische und statische modellbasierte Optimierung

Abhängig davon, ob ein statisches oder ein dynamisches Modell vorliegt, sind unterschiedliche Methoden für die Lösung des Optimierungsproblems erforderlich. In Abbildung 1 wurden die Modelleingangsgrößen allgemein mit \mathbf{u} und Modellausgangsgrößen mit \mathbf{y} . bezeichnet. Im Folgenden werden die für die unterschiedlichen Modellierungsansätze verwendeten Ein- und Ausgangsgrößen konkret benannt.

Dynamisch

Ein dynamisches Modell des geregelten Spritzgießprozesses von den manipulierbaren Prozessparametern s bis zur resultierenden Bauteilqualität \hat{Q} ist in Abbildung 2 dargestellt. Ein dynamisches Prozessmodell bildet die durch die Prozessparameter s parametrisierten Sollwerttrajektorien $\{\mathbf{r}_k\}_{k=1}^N$ auf die Prozessgrößentrajektorien $\{\mathbf{p}_k\}_{k=1}^N$ ab, welche wiederum von einem dynamischen Qualitätsmodell auf die Bauteilqualität $Q \in \mathbb{R}^{n_a}$ abgebildet werden.

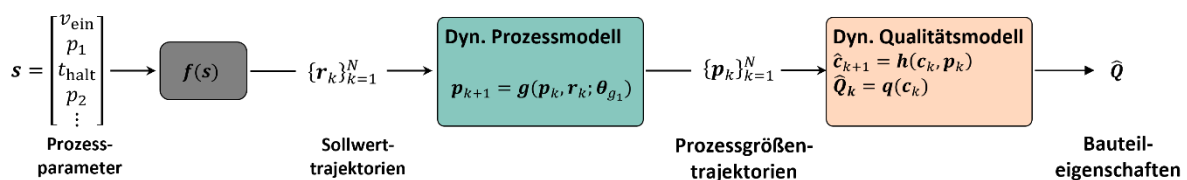


Abbildung 2: Dynamisches Modell des geregelten Spritzgießprozesses bestehend aus dynamischem Prozess- und Qualitätsmodell

An den Modellgleichungen ist erkennbar, dass der Ausgang eines dynamischen Systems zum Zeitpunkt $k + 1$ von dem Zustand des Systems im vorigen Zeitschritt k abhängt. Dies kann am Beispiel des dynamischen Qualitätsmodells erläutert werden:

$$\hat{\mathbf{c}}_{k+1} = \mathbf{h}(\mathbf{c}_k, \mathbf{p}_k)$$

$$\hat{Q}_k = \mathbf{q}(\mathbf{c}_k)$$

$\hat{\mathbf{c}}_k \in \mathbb{R}^{n_c}$ ist hierbei ein modellinterner Zustandsvektor ohne physikalische Interpretation, \mathbf{p} die in das Qualitätsmodell eingehenden Prozessgrößen und \hat{Q} die prädiizierte Bauteilqualität. Der modellinterne Zustand evolviert gemäß der nichtlinearen Zustandsgleichung $\mathbf{h}: \mathbb{R}^{n_c} \rightarrow \mathbb{R}^{n_c}$ und wird durch die Ausgabegleichung $\mathbf{q}: \mathbb{R}^{n_c} \rightarrow \mathbb{R}^{n_a}$ auf die Ausgangsgröße \hat{Q}_k abgebildet. Durch die zeitliche Abhängigkeit kann das System bspw. nicht von einem zum nächsten Zeitpunkt beliebige Werte annehmen, dies muss bei der Optimierung der Prozessgrößen \mathbf{p} berücksichtigt werden.

Liegen ein dynamisches Prozess- und Qualitätsmodell vor, können verschiedene Optimalsteuerungsprobleme formuliert werden, um die optimalen Prozessparameter s^{opt} für eine gegebene Soll-Bauteilqualität Q^{ref} zu ermitteln.

Lösung der Teilprobleme

Eine Möglichkeit besteht darin, zunächst unter Verwendung des Qualitätsmodells die optimalen Prozessgrößentrajektorien $\{\mathbf{p}_k^{\text{opt}}\}_{k=1}^N$ zu ermitteln, um eine geforderte Soll-Bauteilqualität \mathbf{Q}^{ref} zu realisieren. Unter der Annahme einer quadratischen Kostenfunktion und einer Gleichgewichtung aller Qualitätsmerkmale Q_i^{ref} $i = 1, \dots, n_q$ lautet das zu lösende Optimierungsproblem:

$$\{\mathbf{p}_k^{\text{opt}}\}_{k=1}^N = \arg \min_{\mathbf{p}_1, \dots, \mathbf{p}_N} \frac{1}{2} \|\mathbf{Q}_N^{\text{ref}} - \widehat{\mathbf{Q}}_N(\{\mathbf{p}_k\}_{k=1}^N)\|^2$$

Dieses Optimierungsproblem kann mittels Einfach- oder Mehrfachschießverfahren (Single-Shooting bzw. Multiple-Shooting) gelöst werden. In der DIM-Toolbox wird dieses Optimierungsproblem in der Klasse `QualityMultiStageOptimization` formuliert und mittels eines Einzelschießverfahrens gelöst.

Anschließend können unter Verwendung des Prozessmodells die entsprechenden Prozessparameter \mathbf{s}^{opt} ermittelt werden, um den hinsichtlich der Bauteilqualität optimalen Prozessgrößenverlauf $\{\mathbf{p}_k^{\text{opt}}\}_{k=1}^N$ zu ermitteln. Unter Annahme einer quadratischen Kostenfunktion muss hierzu das Optimierungsproblem

$$\mathbf{s}^{\text{opt}} = \arg \min_{\mathbf{s}} \frac{1}{2} \|\{\mathbf{p}_k^{\text{opt}}\}_{k=1}^N - \{\widehat{\mathbf{p}}_k(\mathbf{s})\}_{k=1}^N\|^2$$

gelöst werden. In der Klasse `ProcessMultiStageOptimization` wird dieses Optimierungsproblem formuliert und mittels eines Mehrfachschießverfahrens gelöst.

Nachteilig an dieser Lösungsstrategie ist, dass die optimale Lösung der Teilprobleme nicht zwangsläufig der optimalen Lösung des Gesamtproblems entspricht. Insbesondere können die bei der Lösung des Qualitäts-Teilproblems ermittelten Prozessgrößentrajektorien $\{\mathbf{p}_k^{\text{opt}}\}_{k=0}^N$ von der Maschine nur näherungsweise realisierbar sein. Die Prozessparameter \mathbf{s}^{opt} , welche zur näherungsweisen Realisierung dieser Trajektorien führen, müssen aber nicht zwangsläufig zu der besten Bauteilqualität führen (im Sinne der kleinsten Abweichung von $\mathbf{Q}_N^{\text{ref}}$).

Lösung des Gesamtproblems

Eine Möglichkeit bestünde darin, Prozess- und Qualitätsmodell verkettet auszuwerten, also ineinander einzusetzen. Auf diese Weise werden $\widehat{\mathbf{Q}}_N$ und \mathbf{s} , direkt miteinander verknüpft und die Prozessdynamik eingehalten:

$$\widehat{\mathbf{Q}}_N(\{\mathbf{p}_k\}_{k=1}^N) = \widehat{\mathbf{Q}}_N(\{\widehat{\mathbf{p}}_k(\mathbf{s})\}_{k=1}^N)$$

Anschließend können unter Verwendung des verketteten Prozess- und Qualitätsmodells die optimalen Prozessparameter s^{opt} ermittelt werden, indem das Optimierungsproblem

$$s^{\text{opt}} = \arg \min_s \frac{1}{2} \left\| Q_N^{\text{ref}} - \widehat{Q}_N(\{\widehat{p}_k(s)\}_{k=1}^N) \right\|^2$$

gelöst wird. Es wurde hierbei wieder von einer quadratischen Kostenfunktion ausgegangen. Die Lösung des Optimierungsproblems kann mittels Single-Shooting oder Multiple-Shooting erfolgen (siehe [Seminar & Workshop: Prozessoptimierung mittels numerischer Methoden](#) für weitere Details).

Statisch

Kann die Prozessdynamik des Spritzgießprozesses weitgehend vernachlässigt werden, kann ein statisches Modell zur Prädiktion der Bauteilqualität verwendet werden, wie bspw. in Abbildung 3 dargestellt. Das Optimierungsproblem zur Ermittlung von s^{opt} vereinfacht sich durch die entfallende zeitliche Abhängigkeit dann erheblich.

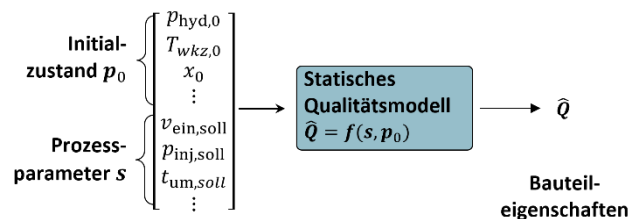


Abbildung 3: Statisches Modell des Spritzgießprozesses

Auf den zeitlichen Index wurde hierbei bewusst verzichtet, um das Entfallen der zeitlichen Komponente auszudrücken, d.h. $\widehat{Q}_N = \widehat{Q}$. Da die Optimierungsgrößen keine Trajektorien mehr sind, reduziert sich zum einen die Anzahl der zu optimierenden Werte. Zum anderen ist das Optimierungsproblem weniger stark nichtlinear, weil keine rekurrenten Modellauswertungen in Abhängigkeit der Optimierungsgrößen mehr erforderlich sind. Ausgehend von einer quadratischen Kostenfunktion lässt sich das Optimierungsproblem wie folgt formulieren:

$$s^{\text{opt}} = \arg \min_s \frac{1}{2} \left\| Q^{\text{ref}} - \widehat{Q}(s, p_0) \right\|^2 \quad (1)$$

Erwähnenswert ist hierbei lediglich, dass der initiale Prozesszustand p_0 eine gegebene Größe und damit nicht manipulierbar ist. Er muss also gemessen werden, bevor das Optimierungsproblem für s gelöst werden kann. Diese und weitere Funktionalitäten sind in der Klasse `StaticProcessOptimizer` der Toolbox implementiert.

Praktische Implikationen

Aufgrund des großen Erfolgs bei der Modellbildung mit statischen Modellen bei gleichzeitig vergleichsweise geringer Komplexität wurde dieser Weg im weiteren Projektverlauf für einen Einsatz im laufenden Betrieb weiterentwickelt. Um die modellbasierte Optimierung in der Praxis einsetzen zu können, mussten im Wesentlichen drei Herausforderungen adressiert werden:

- Initialisierungsabhängigkeit der Lösung: Die Ermittlung von s^{opt} stellt ein nichtlineares Optimierungsproblem dar. In Abhängigkeit der gewählten Initialwerte für s resultieren daher i.d.R. unterschiedliche Lösungen.
- Gültigkeit des datengetriebenen Modells: Das datengetriebene Modell ist zum einen nur für Bereiche gültig, in dem Daten bei der Modellbildung genutzt wurden. Zum anderen ist der Spritzgießprozess zeitvariant, sodass das Modell nicht auf unbeschränkte Zeit gültig ist.
- Darstellung der Lösung: Es müssen mehrere Lösungskandidaten in dem mehrdimensionalen Raum der Prozessparameter so dargestellt werden, dass der Maschinenführer geeignete Lösungskandidaten schnell erfassen kann.

Die realisierten Lösungsstrategien für den Umgang mit diesen Herausforderungen werden im Folgenden kurz skizziert.

Initialisierungsabhängigkeit der Lösung

Um die Initialisierungsabhängigkeit der Lösung zu berücksichtigen, wird der Löser in allen Betriebspunkten s_i mit $i = 1, \dots, S$ initialisiert, für die bereits Messdaten vorliegen. Diese Heuristik wird motiviert durch die Annahme, dass das Modell in Bereichen in denen Daten vorliegen besonders vertrauenswürdig ist und daher Lösungen für s_i^{opt} die in der Nähe dieser Bereiche tendenziell vertrauenswürdiger sein sollten. Insgesamt wird das Optimierungsproblem (1) also S mal gelöst.

Gültigkeit des datengetriebenen Modells

Bei der Lösung des Optimierungsproblems (1) muss berücksichtigt werden, dass das statische Modell $\hat{Q}(s, p_0)$ nur in Bereichen gültig ist, in denen Daten für dessen Parameterschätzung vorlagen. Um diesen Umstand zu berücksichtigen, wird das Optimierungsproblem (1) um folgende Nebenbedingung erweitert:

$$\min(s_i) < s_i^{\text{opt}} < \max(s_i)$$

Hierdurch wird sichergestellt, dass jede Komponente der i Lösungen s_i^{opt} sich innerhalb des Bereiches, in dem Daten aufgenommen wurden, befindet. Anzumerken hierbei ist, dass es

sich um eine univariate Betrachtung handelt. Im Allgemeinen kann man nicht davon ausgehen, dass sich s_i^{opt} in einem Bereich befindet in dem Daten vorliegen, nur weil in die individuellen Koordinatenrichtungen die Unter- und Obergrenzen eingehalten wurden. Im speziellen Anwendungsfall war die Datengrundlage aber ein vollfaktorieller Versuchsplan, sodass die Daten im Eingangsdatenraum in Form eines Würfels vorliegen.

Des Weiteren muss beachtet werden, dass sich das Übertragungsverhalten des Prozesses im Laufe der Zeit durch nicht messbare Störgrößen, Aufwärmvorgänge, Verschleiß, Materialablagerungen, etc. ändert. Durch diese Änderungen verliert das statische Modell seine Gültigkeit, sodass dessen Modellparameter nachgeschätzt werden müssen. Für die Nachschätzung der Modellparameter existieren rekursive Verfahren wie bspw. Recursive Least Squares, welche die Modellparameter ständig, d.h. mit jedem neuen Messwert, aktualisieren. Eine praktische Implementierung einer automatischen Nachschätzung in einer Echtzeit-Umgebung kann aufgrund von Phänomenen wie bspw. dem sogenannten „Blow-Up“ der Kovarianzmatrix, sehr aufwändig sein. Aus diesem Grund wurde beschlossen, eine Nachschätzung der Modellparameter anlassbezogen anzustoßen statt mit jedem neuen Messwert automatisch durchzuführen. Die Aktualität des Modells wird durch eine geeignete Auswahl der Messdaten, die für die Parameterschätzung verwendet werden, sichergestellt. Die Methode für die Datenauswahl erzeugt eine Datenbank, in der die Messdaten von s_i $i = 1, \dots, S$ verschiedenen Betriebspunkten abgelegt sind. Diese Daten stellen eine Untermenge aller Messdaten dar, die Methode für die Datenauswahl ist in der Klasse `Data_Manager` implementiert. Der Prozess der Datenauswahl folgt den folgenden Prinzipien:

- Werden neue Daten in einem neuen Betriebspunkt s_i erhoben, der überdurchschnittlich weit weg von den S Betriebspunkten in der Datenbank liegt, wird die Datenbank um die Daten dieses Betriebspunktes erweitert, d.h. $S = S + 1$. Die Distanz wird hierbei mittels der euklidischen Metrik gemessen, überdurchschnittlich bezieht sich auf die Distanz der S Betriebspunkte s_i in der Datenbank zueinander.
- Werden neue Daten in einem Betriebspunkt s_i erhoben, der maximal durchschnittlich weit weg von den S Betriebspunkten in der Datenbank liegt, werden die Daten des zu dem neuen Betriebspunkt nächsten Betriebspunktes in der Datenbank durch diese ersetzt.

Anlassbezogene Nachschätzung bedeutet, dass die Parameter erst dann nachgeschätzt werden, wenn das Modell eine bestimmte Modellgüte unterschritten hat. Hierbei wird das Modell regelmäßig auf allen S Betriebspunkten in der Datenbank ausgewertet und das Bestimmtheitsmaß R^2 ermittelt. Fällt dieses unter einen vom Nutzer spezifizierten Grenzwert, wird das Modell neu geschätzt.

Darstellung der Lösung

Durch die S -fache Mehrfachinitialisierung bedingt gibt das Optimierungsverfahren letztendlich S Lösungsvorschläge aus. Um die Lösungsvorschläge für den Maschinenführer in einer unterstützenden Weise zu präsentieren, werden folgende Vorverarbeitungsschritte durchgeführt:

- „Schlechte“ Lösungen werden gelöscht. Ist die zu einer Lösung s_i gehörige Prädiktion $\hat{Q}(s_i, p_0)$ zu weit von der gewünschten Bauteilqualität entfernt, d.h. $\|Q^{\text{ref}} - \hat{Q}(s, p_0)\|^2 > d$, wird sie gelöscht. Der Schwellenwert d wird vom Nutzer spezifiziert.
- „Ähnliche“ Lösungen werden zusammengefasst, d.h. wenn $\|s_i - s_j\|^2 < d$ mit $i \neq j$, wird $s_i = s_j$ gesetzt.
- In der Visualisierung werden mehrere zu einer Lösung zusammengefasste Lösungen entsprechend visuell größer dargestellt, siehe Abbildung 4.

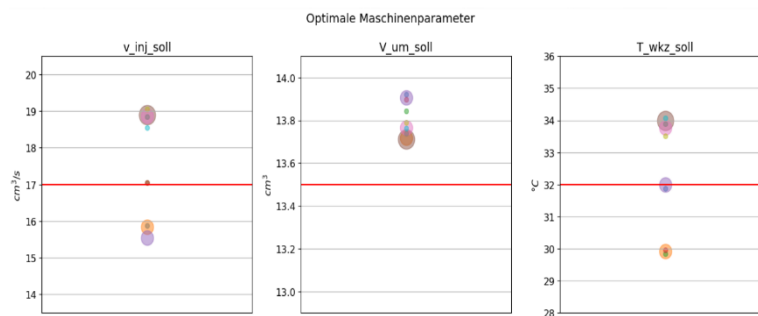


Abbildung 4: Visualisierung der Lösung in einem dreidimensionalen Prozessparameterraum

So wird dem Maschinenführer gezeigt, zu welchen Parameterwerten der Löser öfter konvergiert ist. Diese Werte korrespondieren dann mit lokalen Minima der Kostenfunktion mit einem größeren Einzugsgebiet. Ein direkter Zusammenhang zwischen der Güte der Lösung und der Häufigkeit der Konvergenz in diese Lösung besteht allerdings nicht.

Lessons Learned

Anhand der vorliegenden Ergebnisse und der während der Versuchsdurchführung gesammelten Erfahrungen lassen sich folgende Schlussfolgerungen ableiten:

- Die modellbasierte Optimierung bei statischen Modellen dauert auch bei Mehrfachinitialisierungen nur wenige Sekunden und ist damit für die Echtzeitanwendung geeignet.
- Auch die Nachschätzung der Modellparameter dauert, insofern es sich um ein statisches Modell handelt, nur wenige Sekunden und ist damit für die Echtzeitanwendung geeignet.
- Das Nutzen eines statischen Modells, welches die initiale Werkzeugtemperatur $T_{\text{wkz},0}$ als Prädiktor verwendet, wirkt sich einerseits positiv auf die Prädiktionsgüte des Modells aus. Andererseits wurde aber die Erfahrung gemacht, dass sich die

Werkzeugtemperatur im laufenden Betrieb stets, wenn auch nur um wenig, ändert. Hierdurch ermittelt der Optimierer in jedem Zyklus neue Lösungsvorschläge, was dem Maschinenführer das Ablesen einer geeigneten Lösung stark erschwert. Aus diesem Grund wurde die initiale Werkzeugtemperatur $T_{\text{w kz},0}$ daraufhin nicht mehr als Modelleingangsgröße verwendet.

Weiterführende Literatur

Projektbezogene Literatur

Rehmer, A. & Klute, M. & Kroll, A. & Heim, H.-P. *An internal dynamics approach to predicting batch-end product quality in plastic injection molding using Recurrent Neural Networks*, IFAC-PapersOnLine 6th IEEE Conference on Control Technology and Applications (CCTA), Vol. 53, Elsevier, 2022.

Rehmer, A. & Klute, M. & Kroll, A. & Heim, H.-P. *A Digital Twin for Part Quality Prediction and Control in Plastic Injection Molding*, Modeling, Identification, and Control for Cyber-Physical Systems Towards Industry 4.0, Elsevier, 2023, submitted.

Rehmer, A. & Kroll, A. *Eine Python-Toolbox zur datengetriebenen Modellierung von Spritzgießprozessen und Lösung von Optimalsteuerungsproblemen zur Steuerung der Bauteilqualität*, 32. Workshop Computational Intelligence, KIT Scientific Publishing, Berlin, Deutschland, 2022.

Grundlagenliteratur

Gros, S. & Diehl, M. [Numerical Optimal Control \(draft manuscript\)](https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf), Abrufdatum 14.12.2022, <https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf>, 2022.

Nelles, O. *Nonlinear system identification: from classical approaches to neural networks, fuzzy models, and gaussian processes*. 2. Auflage, Springer Nature, 2020.

Bildquellen:

- [1] <https://hmq-3d.ch/galerie/40/3d-modell-spritzgiessmaschine-arburg.html>
zuletzt aufgerufen am 21.02.2023