



Veranstaltungsort:

Heinz Nixdorf MuseumsForum
Fürstenallee 7, 33102 Paderborn

- **Mechatronische Produkte**
(neue Funktionalitäten, Industrie 4.0, Kosteneffizienz, Zuverlässigkeit, ...)
- **Ressourceneffizienz**
(Energie, Material, ...)
- **Automatisierte Mobilität**
(Konzepte, Modelle, Sensorik)
- **Systems Engineering und Entwicklungsmanagement**
(Prozesse, Verfahren, Software, ...)
- **Innovative Konzepte und digitale Geschäftsmodelle**
(Modelle, Regelung, Optimierung, Eco-Systeme)
- **Nutzerfreundlichkeit und Akzeptanz**
(Assistenzsysteme, Schnittstellen, Interaktion, Gesetzgebung, ...)
- **Systemvernetzung und Systemintegration**
(Konzepte, Verfahren, ...)
- **Smarte Aktoren**
(Konzepte, Beispiele, ...)
- **Serienfertigung mechatronischer Produkte**
(Fallbeispiele, Komponenten, Architektur, Qualitätsmanagement, ...)

www.VDI-Mechatroniktagung.de

Entwicklung eines stochastischen Werkstücktrackings für (teil-)automatisierte Produktionsprozesse

Development of a Stochastic Workpiece Tracking within (Semi-)Automated Production Processes

Karl-Philipp Kortmann, M. Sc.[†], Dr. Alexander Seel[‡], Johannes Zumsande, M. Sc.[†], Mark Wielitzka, M. Sc.[†] und Prof. Dr.-Ing. Tobias Ortmaier[†]

[†] Leibniz Universität Hannover, Institut für Mechatronische Systeme, Appelstraße 11a, 30167 Hannover, Deutschland, karl-philipp.kortmann@imes.uni-hannover.de

[‡] IAV GmbH, Dr.-Ludwig-Kraus-Str. 4, 85080 Gaimersheim, Deutschland, alexander.seel@iav.de

Kurzfassung

In dieser Arbeit wird ein neues Verfahren zur datenbasierten Schätzung des zeitlichen Aufenthalts einzelner Werkstücke oder anderer diskreter Zwischenerzeugnisse in einem komplexen mechatronischen Produktionsprozess vorgestellt. Das gezeigte Verfahren eignet sich insbesondere für teilautomatisierte Produktionsprozesse, die nicht vollständig digitalisiert sind oder bei denen ein hardwarebasiertes Tracking (z. B. mittels RFID) durch die Umgebungsumstände nicht möglich ist. Hierzu wird der Produktionsprozess zunächst in ein stochastisches Modell auf Basis eines gerichteten Graphen überführt. Der Optimierungsalgorithmus nutzt die Zeitstempel der zur Verfügung stehenden Feldbusdaten der Anlagensteuerungen, um den wahrscheinlichsten Pfad einzelner Objekte durch den Produktionsprozess zu rekonstruieren. Das Verfahren konnte in ersten Testphasen an simulierten Prozessdaten exemplarisch validiert werden.

Abstract

This paper presents a new method for the data-based estimation of the temporal location of individual workpieces or other discrete intermediate products in a complex mechatronic production process. The method shown is particularly suitable for semi-automated production processes that are not completely digitalized or where hardware-based tracking (e. g. using RFID) is not possible due to environmental conditions. For this purpose, the production process is first converted into a stochastic model based on a directed graph. The optimization algorithm uses the time stamps of the available fieldbus data of the shopfloors digital process controllers to reconstruct the most probable path of individual objects through the production process.

In the first test phase, the method has been validated successfully with simulated process data.

1 Einleitung

Die Digitalisierung komplexer mechatronischer Produktionssysteme im Sinne der „Industrie 4.0“ ermöglicht die Analyse und Prädiktion produktionsrelevanter Zielgrößen (z. B. Prozessstabilität, Produktqualität, Ausschussminimierung) und birgt folglich erhebliche Mehrwerte hinsichtlich der Optimierung flexibler Produktionsprozesse. Die Grundlage bilden hierbei digitale Mess- und Steuerungsdaten einzelner Subprozesse (z. B. Feldbusdaten, externe Sensorik oder auch Laborergebnisse), die bei einer Prozessdigitalisierung in der Regel in einer zentralen Datenbank („Data Warehouse“) zusammengeführt werden. Für die Analyse dieser meist heterogenen Zeitreihendaten (etwa im Zuge eines prädiktiven Prozessmonitorings) finden neben klassischen statistischen Verfahren auch immer häufiger Methoden des maschinellen Lernens Anwendung [1]. Für diese Methoden stellen Prozesse, die eine größere zeitliche Verzögerung zwischen der Datenerfassung und dem Eintritt der zu analysierenden Zielgrößen aufweisen, eine aktuelle Herausforderung dar [2].

Insbesondere in nicht vollständig digitalisierten und/oder nur teilautomatisierten Produktionssystemen können auch

mit Zeitstempel vorliegende Prozessdaten häufig nicht eindeutig einzelnen Werkstücken zugeordnet werden, sodass bei einer abschließenden Qualitätsbeurteilung des Werkstückes nicht auf den Einfluss einzelner Prozessvariablen rückgeschlossen werden kann. Speziell bei manuellen, gegebenenfalls ungeplanten Eingriffen oder Störungen (z. B. bei Aus- oder Umsortierung von Werkstücken oder nicht-deterministischen Lager- oder Stillstandszeiten) kann die erforderliche Zuordnung zwischen Werkstücken und Daten nur unvollständig erfolgen. Abhilfe schaffen hier beispielsweise Werkstückmarker (z. B. RFID-Chips oder QR-Tags) [3], deren Einsatz allerdings in einigen Anwendungsfällen, wie etwa bei heißen Gusserzeugnissen, nicht möglich oder aufgrund von geringen Stückkosten bei hohem Produktionsdurchsatz ökonomisch nicht sinnvoll ist.

In dieser Arbeit wird ein Verfahren präsentiert, das die nachträgliche Rückverfolgung der zeitlichen Verläufe (Pfade) diskreter Werkstücke durch den Produktionsprozess ermöglicht. Der Algorithmus nutzt zum einen die Zeitinformationen der diskreten Steuerungsdaten aus den vorhandenen Anlagensteuerungen (SPS), die beispielsweise den Beginn einer neuen Bearbeitungssequenz oder den Eingang

eines Werkstückes in einem Subprozess repräsentieren und zum anderen ein identifiziertes oder geschätztes stochastisches Modell des gesamten Produktionsprozesses.

2 Stand der Technik

Aktuelle Konzepte zum Einsatz von Methoden des (überwachten) maschinellen Lernens im industriellen Produktionsumfeld setzen in der Regel eine Digitalisierung insoweit voraus, dass eine relationale Verknüpfung (siehe **Bild 1**) zwischen bearbeitetem Werkstück oder einer Charge und gespeicherten Prozessdaten deterministisch gegeben ist [1, 4]. Dies vereinfacht die Analyse der Einflussfaktoren innerhalb der Prozessvariablen auf relevante Prozesszielgrößen (wie etwa die Produktqualität) ungemein, da Korrelationen auch über große zeitliche Verzögerungen hinweg sichtbar werden.

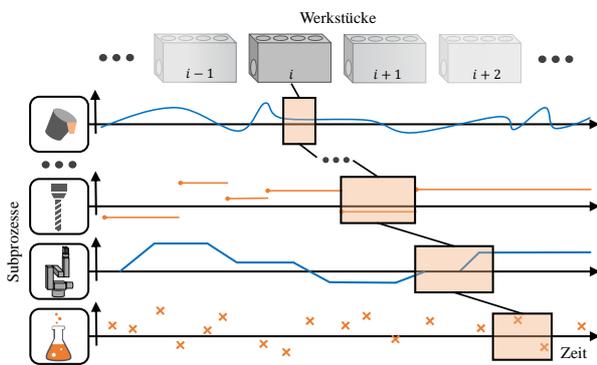


Abbildung 1 Schematische Darstellung des zeitlichen Zusammenhangs zwischen Werkstück und Prozessdaten. Die diskreten Werkstücke durchlaufen eine Reihe von Subprozessen, die individuelle, heterogene Prozessdaten erzeugen. Diese können den Werkstücken für ihre Aufenthaltsdauer (oranger Kasten) zugeordnet werden.

Darüber hinaus etablieren sich auch im Bereich der Zeitreihenanalyse Methoden, die in der Lage sind, die beschriebenen Latenzen zu beachten [5]. Allerdings sind diese, meist auf rekurrenten neuronalen Netzen (RNN) fußenden Methoden, eingeschränkt, was sowohl die Dauer zurückliegender Einflüsse [2], als auch die Interpretierbarkeit für die Produktionsleitung und folglich den aus der Analyse resultierenden Erkenntnisgewinn anbelangt [6].

Für eine vor diesem Hintergrund notwendige Werkstückverfolgung schlagen Zhou et al. den bewährten Einsatz der Radio-Frequency-Identification Technologie (RFID) vor [7]. In [8] wird der Einsatz von ebenfalls hardwareseitigen optischen QR-Codes innerhalb eines Schmiedeprozesses vorgestellt.

Im Bereich der softwareseitigen Verfolgung existieren übertragbare Lösungen in Form von Petri-Netzen [9, 10] oder deterministischen endlichen Zustandsautomaten (engl. deterministic finite automaton, DFA) [11], die beiderseits voraussetzen, dass sämtliche logischen Schaltsignale, die den Zustand (und hier insbesondere den Aufenthalt) eines Werkstückes beeinflussen, vollständig bekannt sind.

Eine Erweiterung im Bereich der Zustandsautomaten stellen sogenannte stochastische Zustandsautomaten (engl. probabilistic automaton, PA) dar, die erfolgreich zum Anlernen von hybriden mechatronischen Produktionssystemen, also unter Verwendung von sowohl kontinuierlichen als auch diskreten Prozessvariablen, verwendet wurden [12].

Ebenfalls stochastisch motiviert ist die Verwendung eines Hidden Markov Modells (HMM) zum Abbilden von nicht-deterministischen Abläufen in Produktionsprozessen [13, 14].

Die im Folgenden vorgestellte Methodik grenzt sich insoweit von der Automatentheorie ab, als dass sie speziell auf die zeitliche Schätzung des Aufenthaltsortes eines Werkstückes abzielt und nicht den Anlagenzustand und dessen Prozessvariablen zu modellieren versucht. Lediglich die Beschreibung des Produktionssystems erfolgt angelehnt an den endlichen Zustandsautomaten in Form eines gerichteten Graphen.

3 Methodik

Das generelle methodische Vorgehen ist in **Bild 2** skizziert. Relevante Prozessvariablen innerhalb der Produktion werden zentral in einer Datenbank geloggt, um sie anschließend sowohl dem Algorithmus zur Optimierung der Werkstückpfade als auch der nachgeschalteten Systemanalyse oder dem Monitoring zugänglich zu machen. Letzterer Schritt wird innerhalb dieser Arbeit nicht näher betrachtet.

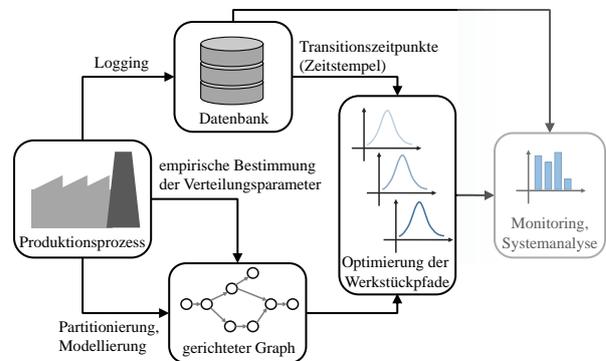


Abbildung 2 Strukturierte Übersicht der verfolgten Methodik innerhalb eines Produktionsprozesses

Der Produktionsprozess wird unter Zuhilfenahme von Expertenwissen in einzelne Subprozesse partitioniert. Diese stellen in sich geschlossene Bearbeitungs-, Transport- oder Lagerprozesse dar, welche durch das Werkstück in einer gewissen Zeit durchlaufen werden. Die Subprozesse werden in Form eines gerichteten Graphen miteinander verknüpft, wobei die Transitionen mögliche Übergänge eines Werkstückes zwischen zwei Subprozessen darstellen. Anschließend werden die wichtigsten Verteilungsparameter für die Verweildauer von Werkstücken für jeden Subprozess entweder empirisch aus den geloggtten Anlagendaten bestimmt (zum Beispiel durch das Anlernen eines Zu-

standsautomaten [15]) oder geschätzt.

3.1 Zeitliche Prozessmodellierung mithilfe gerichteter Graphen

Eine geeignete mathematische Struktur zur Beschreibung eines Produktionsprozesses mit beliebiger Topologie ist der gerichtete Graph, welcher für die in dieser Arbeit notwendigen Schritte wie folgt erweitert wurde:

Definition (stochastischer Produktionsgraph). Ein stoch. Produktionsgraph ist ein Tupel $N = (G, T, s, p, M, \Sigma, c)$, wobei

- $G = (V, E)$ einen gerichteten Graphen darstellt. V ist die geordnete Menge aller Knoten (Subprozesse) und $E \subseteq V \times V$ die Menge aller Kanten (Transitionen) in Form geordneter Paare (i, j) von Elementen $i, j \in V$, wobei $i \neq j$ (schleifenfrei).
- T ist eine Menge an Marken (Werkstücke), die zu Beginn über $s : T \rightarrow V$ einem Ausgangsknoten zugeordnet sind.
- Die Funktion $p : E \rightarrow [0, 1]$ ordnet jeder Kante eine Pfadwahrscheinlichkeit zu, wobei $\sum_j p_{i,j} = 1$ für alle $i \in V$ gilt.
- Die Funktion $M : E \rightarrow \mathbb{R}_+$ ordnet jeder Kante $(i, j) \in E$ den Erwartungswert $\mu_{i,j}$ für die Transitionszeit und $\Sigma : E \rightarrow \mathbb{R}_+$ die zugehörige Standardabweichung $\sigma_{i,j}$ zu.
- Die Kapazität eines Knotens $c : V \rightarrow \mathbb{N}$ gibt die maximale Anzahl an Marken an, die sich gleichzeitig an diesem Knoten befinden können.

Die wesentlichen Analogien zwischen mathematischem Formalismus und realer Anwendung sind in **Tabelle 1** aufgeführt.

Tabelle 1 Analogien zwischen formellem Prozessgraphen und realem Produktionsprozess

Graph	realer Produktionsprozess
Knoten $v \in V$	Subprozess (Bearbeitungs-, Transport- oder Lagerprozess)
Kante $e \in E$	Übergang zwischen Subprozessen
Marke $t \in T$	Werkstück

Die gewählte Struktur stützt sich auf die wichtigsten Restriktionen, Kausalitäten und Nichtdeterminismen, die für einen abstrakten Produktionsprozess identifiziert wurden. So sind die Transitionsmöglichkeiten eines Werkstückes, analog zu einer sequenziellen Produktionsstraße, restriktiv durch den Graphen vorgegeben. Ein zufälliger Werkstückverlust kann wiederum als Übergang in eine Senke über die Pfadwahrscheinlichkeiten $p_{i,j}$ als stochastisch unabhängiges Zufallsereignis abgebildet werden. Die nichtdeterministische Transitionszeit eines Werkstückes zwischen den Subprozessen i und j wird über die Verteilungsparameter $\mu_{i,j}$ und $\sigma_{i,j}$ mit einer geeigneten Wahrscheinlichkeitsverteilung (i. d. R. eine Gammaverteilung) modelliert.

Die softwareseitige Repräsentation des Graphen erfolgt objektorientiert in MATLAB, weshalb zur vereinfachten Darstellung im Folgenden die Punkt-Notation verwendet wird.

Tabelle 2 Wichtige Variablen, auf die in den Algorithmen 1-4 Bezug genommen wird

Variablen	Beschreibung
$vertices(i)$	Instanz eines Knoten i
$vertices(i).pings()$	Array aller Zeitstempel eines Knotens i
$\phi()$	Array paralleler Pfade
$slack$	Differenz von Anzahl zuzuordnender vorhandener Zeitstempel zweier Knoten

Die Darstellung der datenbasierten Schätzung des zeitlichen Aufenthalts von Werkstücken an den Knoten erfolgt anhand des Beispiels in **Bild 3**, die Rekonstruktion anhand eines Pfades im folgenden Sinn:

Definition (Pfad). Eine zusammenhängende, gerichtete Verbindung über n Knoten $v_1, \dots, v_n \in V$ auf G heißt Pfad $\phi(t) : V^n \rightarrow \mathbb{R}^n, (v_1, \dots, v_n) \mapsto (z_1, \dots, z_n)$ zur Marke $t \in T$, wenn z_i den Zeitstempel der Ankunftszeit von t an den Knoten i bezeichnet. Die Transitionszeit zwischen zwei aufeinanderfolgende Knoten im Pfad ist dann $\tau_m = z_{m+1} - z_m$.

Die synthetischen Zeitstempel $z_i = z_i(t)$ wurden in einer ersten Simulation anhand der vorgegebenen Statistik (konstantes $\mu_{i,j}$ und $\sigma_{i,j}$ für jede Transition von Knoten i nach j) zufällig über eine Gammaverteilung erzeugt und in den Knoten $vertices(i).pings$ gespeichert. Um die Verteilungsdichten der wirklichen Übergangszeiten nicht durch Wartezeiten zu deformieren, beschränken wir uns auf ausreichende Kapazitäten $c(i) \geq 1$ an den Knoten i , sodass jedes Werkstück bei Ankunft sofort bearbeitet werden kann. Weitere Variablen in der folgenden algorithmischen Darstellung sind in **Tabelle 2** zusammengefasst.

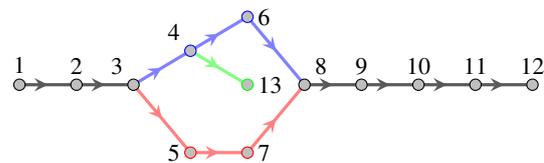


Abbildung 3 Produktionsgraph mit sowohl rein sequenziellen als auch parallelen Teilstücken. Über Knoten 13 kann beispielsweise Ausschuss aus Knoten 4 modelliert werden.

3.2 Optimierungsalgorithmus

Grundidee: Suche zwischen benachbarten Knoten eine Schar von Pfaden und optimiere die Wahl von Start- und Endzeitstempel hinsichtlich eines Abstandmaßes zu einer vorgegebenen Statistik. Die Einbeziehung mehrerer paralleler Pfade stabilisiert dabei die Berechnung des Abstandsmaßes, da große Fluktuationen über die Statistik der verbleibenden Werte kompensiert werden können.

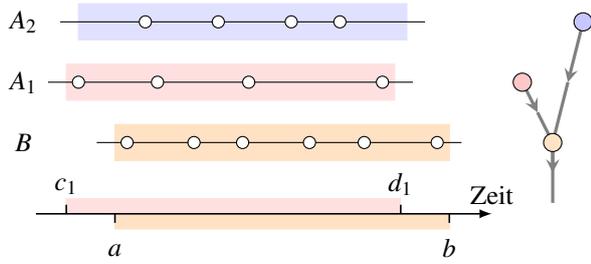


Abbildung 4 Auswahl möglicher Zeitstempel einer Zusammenführung über das Abbilden des Ausgangsintervalls $[a, b]$ des Endknotens B entsprechend der Transitionszeiten auf die Intervalle $[c, d]$; aller möglicher Startknoten A_i ; z. B. A_1 .

Der Optimierungsalgorithmus lässt sich auf die Rückverfolgung von Pfaden über eine Verzweigung hinweg, wie in **Bild 4** gezeigt, reduzieren und beginnt mit der Auswahl der Endzeitstempel auf Knoten B anhand eines vorgegebenen Intervalls $[a, b]$. Mit der Statistik der Transitionszeiten (TRANSITIONS) von A_i nach B werden in **Algorithmus 1** mögliche Startzeitstempel auf A_i aus den berechneten Intervallen $[c, d]$ so bestimmt, dass ihre Anzahl im Vergleich zu den Endzeitstempeln geringer ist.

Algorithmus 1 Auswahl möglicher Startzeitstempel

```

1: procedure MAPINTERVALS( $a, b$ )
2:    $slack \leftarrow -1$   $\triangleright$  difference number start/end pings
3:    $q \leftarrow 2$   $\triangleright$  downsizing interval by multiples of std
4:   while  $slack < 0$  &&  $q > -2.0$  do
5:     for  $sv$  in  $startVertices$  do
6:        $mean \leftarrow TRANSITIONS(sv, endVertex).mean$ 
7:        $std \leftarrow TRANSITIONS(sv, endVertex).std$ 
8:        $c \leftarrow a - mean - q \cdot std$ 
9:        $d \leftarrow \min(b - mean + q \cdot std, b)$ 
10:       $pings(sv) \leftarrow vertices(sv).pings(c < \tau < d)$ 
11:    end for
12:     $pings(endVertex)$ 
       $\leftarrow vertices(endVertex).pings(a < \tau < b)$ 
13:     $slack \leftarrow pings(startVertices).length -$ 
       $pings(endVertex).length$ 
14:    reduce  $q$ 
15:  end while
16:  if  $slack < 0$  then
17:    rerun while-loop along with symmetrically
    downsizing the interval  $[a, b]$  step by step
18:  end if
19:  return  $pings, StartVertices, endVertex$ 
20: end procedure

```

Von diesen möglichen Zeitstempeln der Startknoten A_i werden mit einem einstellbaren Parameter $maxSlack$, wie z. B. in **Bild 5** angedeutet, je $maxSlack = 2$ Zeitstempel ausgenommen, um die Robustheit gegen fehlerhafte Zeitstempeln zu gewährleisten, die etwa durch fehlerhafte Steuersignale o. ä. unbeabsichtigt gespeichert wurden.

*Alle Möglichkeiten, um von n_{level} Startzeitstempeln k_{level} Elemente auf Knoten $startVertices(level)$ auszuwählen.

†Alle Möglichkeiten, um aus n Endzeitstempeln $k_1 + k_2 + \dots + k_m$ Elemente (m verschiedene) auszuwählen. Für z. B. $n = 5$, $k_1 = 2$, $k_2 = 1$ sind alle möglichen Permutationen der Abfolge 2 2 1 0 0 zu bestimmen.

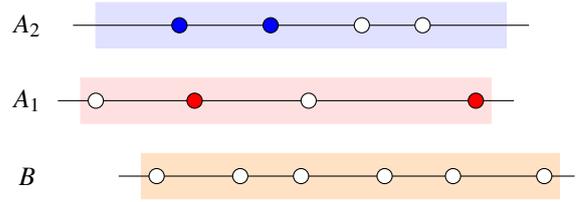


Abbildung 5 Eine von 36 Permutationen der ausgewählten Zeitstempel des Startknotens A_i .

Algorithmus 2 permutiere alle Startzeitstempel

```

1: procedure ALTERPINGS( $pings, level$ )
2:   if  $level < startVertices.length$  then
3:     for  $p \in Permutation(k_{level}, n_{level})^*$  do
4:       restrict  $pings(startVertices(level))$  to  $p$ 
5:        $(\delta, \phi) \leftarrow ALTERPINGS(pings, level + 1)$ 
6:        $(\delta_{opt}, \phi_{opt}) \leftarrow CHOOSE(\delta_{opt}, \phi_{opt}, \delta, \phi)$ 
7:     end for
8:   else
9:     for  $p \in Permutation(k_{level}, n_{level})$  do
10:      restrict  $pings(startVertices(level))$  to  $p$ 
11:       $(\delta, \phi) \leftarrow OPTIMIZEPERMUTATIONS(pings)$ 
12:       $(\delta_{opt}, \phi_{opt}) \leftarrow CHOOSE(\delta_{opt}, \phi_{opt}, \delta, \phi)$ 
13:    end for
14:   end if
15:   return  $\delta_{opt}, \phi_{opt}$ 
16: end procedure

```

Die Suche in **Algorithmus 3** ermittelt (Zeile 9, CAUSALITYCHECK) bei fixierten Startzeitstempeln des **Algorithmus 2** aus allen kausal möglichen Kombinationen die Konstellation der Endzeitstempel, die ohne Überholungen das kleinste Abstandsmaß

$$\delta = \frac{1}{n} \sum_{i=1}^n \frac{(\tau_i - \mu_i)^2}{\sigma_i^2} \quad (1)$$

über alle n Transitionen aufweist. Das Maß leitet sich aus der Mahalononis-Distanz ab, welche maßgeblich innerhalb der multivariaten Statistik Verwendung findet [16]. Der aufrufende **Algorithmus 2** startet mit $level = 1$ als Laufvariable für den ersten Startknoten. Aus den so optimierten Pfaden wird die Konstellation der Startzeitstempel mit dem kleinsten Abstandsmaß ermittelt. Die Auswahl CHOOSE einer neu berechneten Schar ϕ von parallelen Pfaden zu der bis dahin als optimal vorgehaltenen Schar ϕ_{opt} erfolgt dann, wenn die Differenz der Abstandsmaße den ursprünglichen Absolutwert übersteigt. Im Gegensatz zu einer Zusammenführung (**Bild 3**) erfolgt die Behandlung einer Verzweigung analog mit umgekehrter zeitlicher Betrachtung.

Abschließend lässt sich der Parameter $maxSlack = 2$ als Obergrenze falscher Zeitstempel minimieren, um möglichst viele parallele Pfade zu ermitteln.

Die gesamte Optimierung zwischen zwei benachbarten Knoten fasst **Algorithmus 4** zusammen. Somit können Pfade entsprechend der Topologie des Graphen rekursiv rekonstruiert werden. Ein beispielhaftes Teilergebnis für den anfangs in **Bild 3** gezeigten Graphen zeigt **Bild 7**. Für ein

Algorithmus 3 permutiere alle Endzeitstempel

```

1: procedure OPTIMIZEPERMUTATIONS(pings)
2:   for  $i = 1 : startVertices.length$  do
3:      $k_i \leftarrow pings(startVertices(i)).length$ 
4:   end for
5:    $\delta_{opt} \leftarrow 0, \phi_{opt} \leftarrow \{\}$  ▷ init
6:    $n \leftarrow pings(endVertex).length$ 
7:   for  $p \in \text{Permutation}(k, n)^\dagger$  do
8:     set up all pathes  $\phi$  connecting all the
       pings(startVertices) with pings(endVertex)
       without overtaking inside of edges, c. f. fig.6
9:     if CAUSALITYCHECK( $\phi$ ) then
10:       $\delta \leftarrow 0$  ▷ deviation w.r.t. statistic data
11:       $start \leftarrow \phi().startVertex, end \leftarrow \phi().endVertex$ 
12:      for  $i = \phi.length$  do
13:         $mean \leftarrow \text{TRANSITIONS}(start, end).mean$ 
14:         $std \leftarrow \text{TRANSITIONS}(start, end).std$ 
15:         $duration \leftarrow transitionTime(\phi(i))$ 
16:         $\delta \leftarrow \delta + (duration - mean)^2 / std^2$ 
17:      end for
18:       $\delta \leftarrow \delta / \phi.length$  ▷ normalize per path
19:       $(\delta_{opt}, \phi_{opt}) \leftarrow \text{CHOOSE}(\delta_{opt}, \phi_{opt}, \delta, \phi)$ 
20:    end if
21:  end for
22:  return  $\delta_{opt}, \phi_{opt}$ 
23: end procedure

```

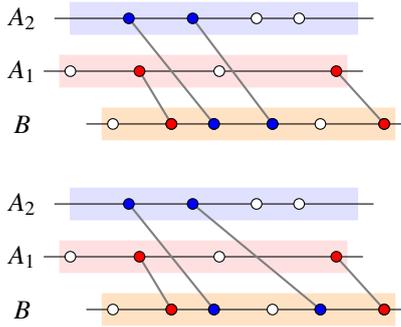


Abbildung 6 Zwei der 90 Permutationen der Endzeitstempel des Endknotens B zur Überprüfung kausaler Pfade – bei fixierter Konfiguration der Startknoten A_i .

Ausgangsintervall in Knoten 12 sind die vollständig rekonstruierten Pfade (roter und blauer Zweig), nicht vollständig rekonstruierten Pfade, sowie Ausschlusspfade (grün) dargestellt.

3.3 Auswertungsmethodik

Zur Validierung einer abgeschlossenen Zuordnung von n Zeitstempeln auf die Werkstücke sollen zwei Fehlermaße genutzt werden. Zum einen der Fehlerquotient

$$ERR = \frac{1}{n} \sum_{i=1}^n I(y(i), \hat{y}(i)), \quad (2)$$

wobei für die Indikatorfunktion

$$I(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & \text{sonst} \end{cases} \quad (3)$$

gilt und die Funktion y das tatsächliche Werkstück und \hat{y}

Algorithmus 4 Minimierung der Fehlstellen

```

1: procedure REDUCEGAPS( $a, b$ )
2:   while  $maxSlack \geq 0$  do
3:      $(pings, startVertices, endVertex)$ 
        $\leftarrow \text{MAPINTERVALS}(a, b)$ 
4:      $(\delta_{opt}, \phi_{opt}) \leftarrow \text{ALTERPINGS}(pings, 1)$ 
5:      $maxSlack = maxSlack - 1$ 
6:   end while
7: end procedure

```

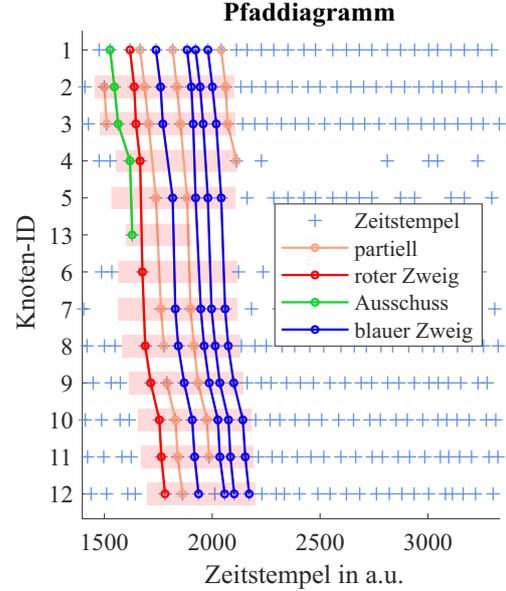


Abbildung 7 Datenbasierte Schätzung des zeitlichen Aufenthalts von Werkstücken an Knoten. Die Verbindungen benachbarter Knoten (partiell) werden zwischen Start- und Endknoten zu durchgehenden Pfaden (roter und blauer Zweig, Ausschuss) zusammengeführt. Ist ein Teilstück nicht vorhanden, wie z. B. zwischen den Knoten 8 und 9, kann kein durchgehender Pfad ermittelt werden.

das zugeordnete Werkstück angeben. Da der Fehlerquotient nur den Anteil an vollständig richtigen Zuordnungen angibt, bietet sich zum anderen die Wurzel der mittleren quadratischen Abweichung

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [\hat{z}(i) - z(i)]^2} \quad (4)$$

als zeitliches Fehlermaß an, wobei $\hat{z}(i) - z(i)$ die zeitliche Differenz zwischen geschätzter und tatsächlicher Ankunftszeit eines Werkstückes für den i -ten Zeitstempel ausgedrückt.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Konzept zur stochastischen Werkstückverfolgung in einem nicht vollständig digitalisierten Produktionsumfeld vorgestellt. Die Methodik lässt sich auf beliebige Produktionstopologien anwenden und wurde auf größtmögliche Robustheit gegenüber fehlerhaften Anlagendaten oder ungenauer statistischer Parame-

tern ausgelegt. Das Konzept wurde in der Simulation erfolgreich überprüft und soll im Folgenden mit Daten aus mehreren realen Industrieanlagen verifiziert werden. Hierfür stehen sowohl eine Forschungsanlage mit einstellbaren Taktzeiten der mechatronischen Subprozesse [17], als auch eine Industrieanlage einer Gießerei mit nicht vollständiger Digitalisierung der Produktionsstrecke zur Verfügung. Innerhalb der Methodik muss weiterhin untersucht werden, wie sich Kapazitätsrestriktionen auswirken und in welcher Weise der Ausfall oder das Blockieren von Subprozessen abgebildet werden kann.

Hinweis Diese Forschungsarbeit wurde finanziell durch Mittel des Europäischen Fonds für regionale Entwicklung (EFRE) unterstützt.

5 Literatur

- [1] Qin, S. J.: *Survey on Data-Driven Industrial Process Monitoring and Diagnosis*. Annual Reviews in Control. (2012) 36(2), ISSN 1367-5788, S. 220-234.
- [2] Schmidhuber, J.: *Deep learning in neural networks: An overview*. Neural Networks. (2015) 61, ISSN 0893-6080. S. 85-117.
- [3] Jana, R.; Rahmati, A.; Zhong, L.; Hiltunen, M.: *Reliability Techniques for RFID-Based Object Tracking Applications*. In: 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks 2007. 25./28. Juni 2007 Edinburgh, UK. ISBN 0-7695-2855-4. S. 113-118.
- [4] Qi, Q.; Tao, F.: *Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison*. IEEE Access. (2018) 6, ISSN 2169-3536. S. 3585-3593.
- [5] Tax, N.; Verenich, I.; La Rosa, M.; Dumas, M.: *Predictive Business Process Monitoring with LSTM Neural Networks*. Lecture Notes in Computer Science. (2017) 10253, ISSN 0302-9743. S. 477-492.
- [6] Wu, M.; Hughes, M. C.; Parbhoo, S.; Zazzi, M.; Roth, V.; Doshi-Velez F.: *Beyond Sparsity: Tree Regularization of Deep Models for Interpretability*. In: 32th AAAI Conference on Artificial Intelligence (AAAI-18) 2018. 2./7. Februar 2018 New Orleans, Louisiana, USA. AAAI 2018. ISBN 978-1-57735-800-8. S. 1670-1678.
- [7] Zhou, G.; Jiang, P.; Zheng, M.: *Design of an RFID-Based Manufacturing Data Tracking System in Plant Production*. In: Intelligent Robotics and Applications (ICIRA) 2008. 15./17. Oktober 2008 Wuhan, CHN. ISBN 978-3-540-88516-0. S. 688-696.
- [8] Liewald, M.; Karadogan, C.; Lindemann, B.; Jazdi, N.; Weyrich, M.: *On the tracking of individual workpieces in hot forging plants*. CIRP Journal of Manufacturing Science and Technology. (2018) 22, ISSN 1755-5817. S. 116-120.
- [9] Wu, N.; Chu, F.; Chu, C.; Zhou, M.: *Petri Net Modeling and Cycle-Time Analysis of Dual-Arm Cluster Tools With Wafer Revisiting*. IEEE Transactions on Systems, Man, and Cybernetics: Systems. (2013) 2168-2216, ISSN 2168-2216. S. 196-207.
- [10] Cabasino, M. P.; Giua, A.; Seatzu, C.: *Fault detection for discrete event systems using Petri nets with unobservable transitions*. Automatica. (2010) 46, ISSN 0005-1098. S. 1531-1539.
- [11] Hashtrudi, S.; Raymond, Z.: *Fault Diagnosis in Discrete-Event Systems: Framework and Model Reduction*. IEEE Transactions on Automatic Control. (2003) 48, ISSN 0018-9286. S. 1199-1212.
- [12] Niggemann, O.; Stein, B.; Vodencarevic, A.; Maier, A.; Büning, H. K.: *Learning Behavior Models for Hybrid Timed Systems*. In: 26th AAAI Conference on Artificial Intelligence (AAAI-12) 2012. 22./26. Juli 2012 Ontario, Kalifornien, USA. AAAI 2012. ISBN 978-1-57735-568-7. S. 1083-1090.
- [13] Esmael, B.; Arnaout, A.; Fruhwirth, R. K.; Thonhauser, G.: *Improving Time Series Classification Using Hidden Markov Models*. In: 12th International Conference on Hybrid Intelligent Systems (HIS) 2012. 4./7. Dezember Pune, IN. ISBN 978-1-4673-5115-7. S. 502-507.
- [14] Hofbaur, M. W.; Williams, B. C.: *Mode Estimation of Probabilistic Hybrid Systems*. In: Hybrid Systems: Computation and Control (HSCC) 2002. 25./27. März 2002 Stanford CA, USA. ISBN 978-3-540-43321-7. S. 253-266.
- [15] Niggemann, O.; Maier, A.; Just, R.; Jäger, M.: *Anomaly Detection in Production Plants using Timed Automata: Automated Learning of Models from Observations*. In: 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2011. 28./31. Juli 2012 Noordwijkerhout, NL. ISBN 978-989-8425-74-4. S. 363-369.
- [16] De Maesschalck, R.; Jouan-Rimbaud, D.; Massaer, D. L.: *The mahalanobis distance*. Chemometrics and intelligent laboratory systems. (2000) 50, ISSN 1755-5817. S. 1-18.
- [17] Maurer, I.; Riva, M.; Hansen, C.; Ortmaier, T.: *Cloud-based Plant and Process Monitoring based on a Modular and Scalable Data Analytics Infrastructure*. In: Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter 2017. ISBN 978-3-662-54441-9. S. 33-42.