

Simulativer Vergleich zentraler und dezentraler Steuerungen einer Getriebefertigung

Simulative Comparison of Central and Decentralized Controls of a Transmission Production

Nils Willeke, Benjamin Kuhrke, Felix Kuschicke, Opel Automobile GmbH,
Rüsselsheim (Germany), nils.willeke@opel.com, benjamin.kuhrke@opel.com,
felix.kuschicke@opel.com

Robert Scheffermann, flexis AG, Stuttgart (Germany),
robert.scheffermann@flexis.de

Eckart Uhlmann, TU Berlin, Berlin (Germany), eckart.uhlmann@iwf.tu-berlin.de

Abstract: This paper deals with the development of a simulation system for the evaluation of control algorithms in production systems. The simulation system is capable of performing both high-performance studies of control algorithms in different scenarios as well as studies of real-time interactions between a human and a control system. The outcome of the investigation on structural parameters depends strongly on the scenarios. The inherent advantage of centralised control is the planning scope and the ability to carry out setup operations ahead of schedule. A high number of set-up procedures combined with a low frequency of disturbances will lead to a better performance of centralised control.

1 Einführung

Die industrielle Produktion sieht sich in der heutigen Zeit mit mehreren Megatrends konfrontiert (Abele und Reinhart 2011). Die Globalisierung eröffnet auf der einen Seite neue Absatzmärkte, aber verstärkt auf der anderen Seite den Wettbewerb und erhöht zugleich den Kostendruck der einzelnen Unternehmen (Niehues 2016; Brackel 2009). Zusätzlich fordern Kunden individuellere Produkte zu geringeren Preisen, aber gleichzeitig eine hohe Qualität (Brackel 2009). Die Folge sind volatile Märkte, eine Verkürzung der Produktlebenszyklen sowie eine Zunahme der Variantenvielfalt (Nyhuis und Wiendahl 2009).

In einem Wirtschaftsstandort wie Deutschland, in dem die Produktion eine hohe Bedeutung genießt (Abele und Reinhart 2011), ist es dabei umso wichtiger, Produkte effizient zu fertigen. Als Grundvoraussetzung müssen die Unternehmen ihre eigenen Prozesse beherrschen und stetig mittels Prozessinnovationen verbessern (Scholz-Reiter und Höhns 2012).

Um der Dynamik des Marktes zu begegnen, müssen Unternehmen ihre Produktionssteuerung befähigen, jederzeit den Zustand der Fertigung erfassen, bewerten und gegebenenfalls Anpassungen einleiten zu können. Nach Scholz-Reiter und Höhns (2012, S. 294) haben die Informations- und Kommunikationssysteme eine „sprunghafte Entwicklung“ vollzogen und ermöglichen eine permanente Datenverfügbarkeit. Neben der Datentransparenz ist die Geschwindigkeit, in der die Daten erfasst und verarbeitet werden, ein wesentlicher Faktor um eine echtzeitfähige Produktionssteuerung umzusetzen (Schmidt et al. 2007).

Zunächst werden noch einmal die Arten von Steuerungsstrukturen vorgestellt. Es kann zwischen zentral-hierarchischen, dezentral-hierarchischen und dezentral-heterarchischen Steuerungsstrukturen unterschieden werden (Abb. 1 sowie Scholz-Reiter und Freitag 2007)

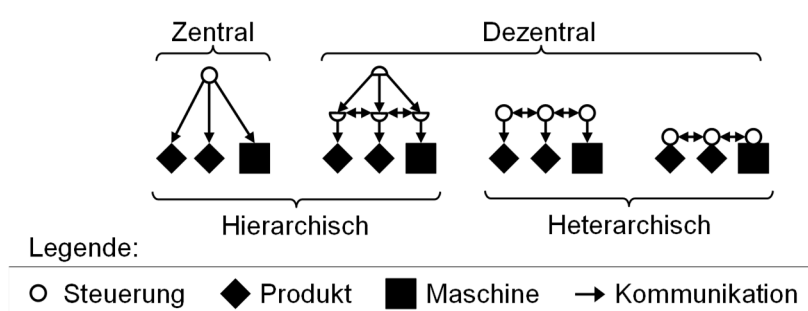


Abbildung 1: Klassifikation von Steuerungsstrukturen (Scholz-Reiter und Freitag 2007, S. 715)

Nach Lödding (2016, S. 2) sind die Voraussetzungen für eine effiziente Fertigung „ein realistischer und guter Produktionsplan“ und eine „funktionierende Fertigungssteuerung“. Fertigungssteuerungen, in denen Steuerungsentscheidungen anhand eines vorher festgelegten Produktionsplans getroffen werden, sind zentral-hierarchisch strukturiert. Diese Art wird in der Folge als *zentrale Steuerungen* bezeichnet. Sofern die Vorhersagequalität hoch bzw. die Auftragsschwankungen gering sind, lassen sich mit zentralen Steuerungen eine hohe Planungssicherheit und Planungsgenauigkeit erzielen (Grundstein et al. 2013; Schukraft et al. 2015). Auch wenn die beiden Attribute nur unter den beschriebenen Voraussetzungen gelten, sorgen sie für eine weite Verbreitung der zentralen Steuerung in der Praxis (Lödding 2016).

In turbulenten Fertigungsumgebungen stellt das Konzept der dynamischen Ablaufplanung (dynamic scheduling) ein Lösungsansatz in zentral-hierarchischen Systemen dar. Bei Abweichungen vom Produktionsplan (z. B. infolge von Störungen) wird auf Basis des aktuellen Zustands eine Neuplanung (Rescheduling) angestoßen. Die neu berechnete Lösung wird entweder in den bisherigen Produktionsplan integriert oder ersetzt diesen. In dynamischen Umfeldern neigen Rescheduling-Ansätze zu einem Übersteuern, was als Systemnervosität bezeichnet wird und zu Instabilitäten führen kann (Ouelhadj und Petrovic 2008).

Dezentrale Ansätze in der Fertigungssteuerung zeichnen sich durch kurze Entscheidungswege und einen geringen Koordinationsaufwand aus (Schuh et al. 2012). Sie

sind dadurch sehr reaktionsschnell und flexibel einsetzbar (Ramsauer 1997). Mehrere Studien stützen die These, dass dezentral-heterarchische Entscheidungsstrukturen (nachfolgend als *dezentrale Steuerungen* bezeichnet) in komplexen und dynamischen Produktionsumfeldern ähnlich gute oder sogar bessere Ergebnisse als ausgewählte zentrale Steuerungsansätze erzielen können (Windt et al. 2010a; Windt et al. 2010b; Scholz-Reiter et al. 2010).

Die logistische Zielerreichung setzt sich aus Durchlaufzeit (DLZ), Auslastung und dem Work in Process (WIP) zusammen. Der entscheidende Einflussfaktor auf die logistische Zielerreichung von zentraler oder dezentraler Steuerung ist demnach die Komplexität eines Anwendungsszenarios (Philipp et al. 2006). Nach Schuh (2005) ist die Komplexität nur durch mehrere unabhängige Parameter zu beschreiben (Schuh 2005). Hierzu liefern Philipp et al. (2006) und Grundstein et al. (2015) eine Liste an möglichen Komplexitätstreibern.

Bei der Auslegung eines Komponentenwerks in der Automobilindustrie ist der zu erzielende Durchsatz vorgegeben. Das Maschinenmengengerüst sollte so aufgebaut werden, dass in Bezug auf die Maschinenkapazität ein Engpass am Anfang der Fertigung entsteht. Während der Fertigung wird die Maschinenkapazität durch deterministische (Rüst- und Werkzeugwechsellaufwände) und stochastische Ereignisse (Störungen) reduziert. Diese Einflüsse können durch die folgenden Parameter beschrieben werden:

- Rüstzeit,
- Werkzeugwechselzeit,
- Werkzeugstandmenge und
- Stördauer.

Die Werte der einzelnen Parameter können sich im Laufe der späteren Produktion durch kontinuierliche Verbesserungsprozesse verändern. Daher ist es unerlässlich, verschiedene Kombinationen möglicher Ausprägungsspektren der Parameter in Bezug auf die logistische Zielerreichung (Auslastung, DLZ und WIP) zu evaluieren. Eine Simulationsstudie mit einer zentralen und dezentralen Steuerung wird dabei Aufschluss darüber geben, ob die genannten Parameter die logistische Zielerreichung der beiden Steuerungen unterschiedlich stark beeinflussen.

Um die Fragen zu beantworten, wurde eine Simulationsumgebung entwickelt, die in den nachfolgenden Abschnitten näher beschrieben wird.

2 Simulationsumgebung

Im Rahmen des Forschungsprojektes iWePro wurde eine Simulationsumgebung entwickelt werden, welche in der Lage ist, einerseits hoch performante Simulationsstudien verschiedener Konstellationen aus Fertigungsstrukturen und Steuerungs-lösungen sowie andererseits Untersuchungen von menschlichen Steuerungseinflüssen, die über Smart Devices in Echtzeit getroffen werden, durchzuführen (Uhlmann et al. 2015; Otto und Uhlmann 2016). Erstere Untersuchung wird nachfolgend als *Simulator* bezeichnet und letztere als *hybrider Demonstrator*. Der hybride Demonstrator ist für die hier vorgestellten Ergebnisse zwar weniger relevant, sollte aber der Vollständigkeit halber erwähnt werden.

2.1 Gesamtsystem und Kommunikation

Das Gesamtsystem koppelt diverse Software- und Hardwarekomponenten an den Simulator, welcher einmal im Umfeld des hybriden Demonstrators und einmal als Teil der Simulationsumgebung zur Durchführung von Simulationsexperimenten dient. Abbildung 2 zeigt die Zusammensetzung des Gesamtsystems:

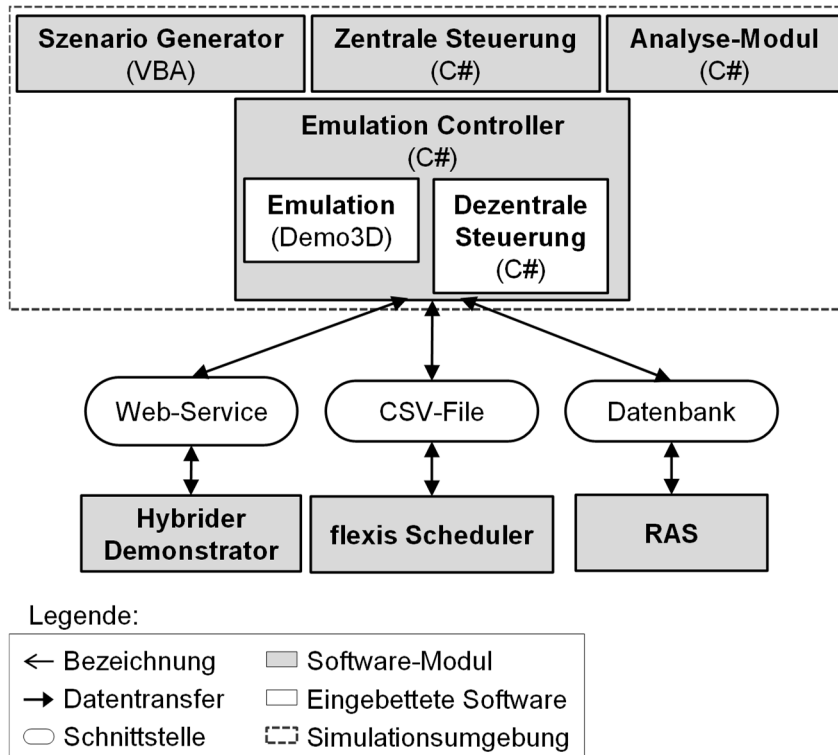


Abbildung 2: Einbettung des Simulators und des hybriden Demonstrators in das Gesamtsystem

Zur Demonstration des Gesamtsystems auf Messen wurde der hybride Demonstrator entwickelt. In diesem können Anwender interaktiv durch gekoppelte Hardware mit der Simulation interagieren (selbst Entscheidungen treffen). Darüber hinaus ermöglicht die 3D-Fähigkeit der Simulationssoftware Demo3D eine realitätsnahe Erprobung des Arbeitens in der Werkstattfertigung. Innerhalb des Forschungsprojektes iWePro wurde das System an diverse Komponenten gekoppelt, hierbei wurden unterschiedliche Kommunikationswege untersucht. Daraus resultierte u. a. eine Kommunikation über einen Web-Service, dessen Performanz die Geschwindigkeit des Simulators auf maximal 10-fache Echtzeit einschränkt.

Zur Bewertung der Lösungsqualität der simulatorinternen Steuerungsalgorithmen wurde eine Schnittstelle über CSV-Dateien zu dem Scheduler der flexis AG geschaffen, welcher mit mehr Rechenaufwand qualitativ hochwertige Lösungen generiert und

somit als Referenz dient. Der Einsatz dieses Schedulers für längere und/oder viele Simulationsläufe ist aufgrund der langen Rechenzeiten nicht praktikabel.

Das Fraunhofer IPK aus Berlin hat ein eigenes Rechner-Agenten-System (RAS) implementiert, welches alternativ die Simulation steuern kann. Die Schnittstelle zu diesem RAS wurde über eine Datenbank implementiert – die Performanz hierüber kann die Simulationsgeschwindigkeit bis zu dem 100-fachen der Echtzeit erreichen.

2.2 Simulator

Die Software-Architektur des Simulators und der direkt verbundenen Komponenten, welche für die Durchführung von Simulationsläufen relevant sind, wird in der Abbildung 3 dargestellt.

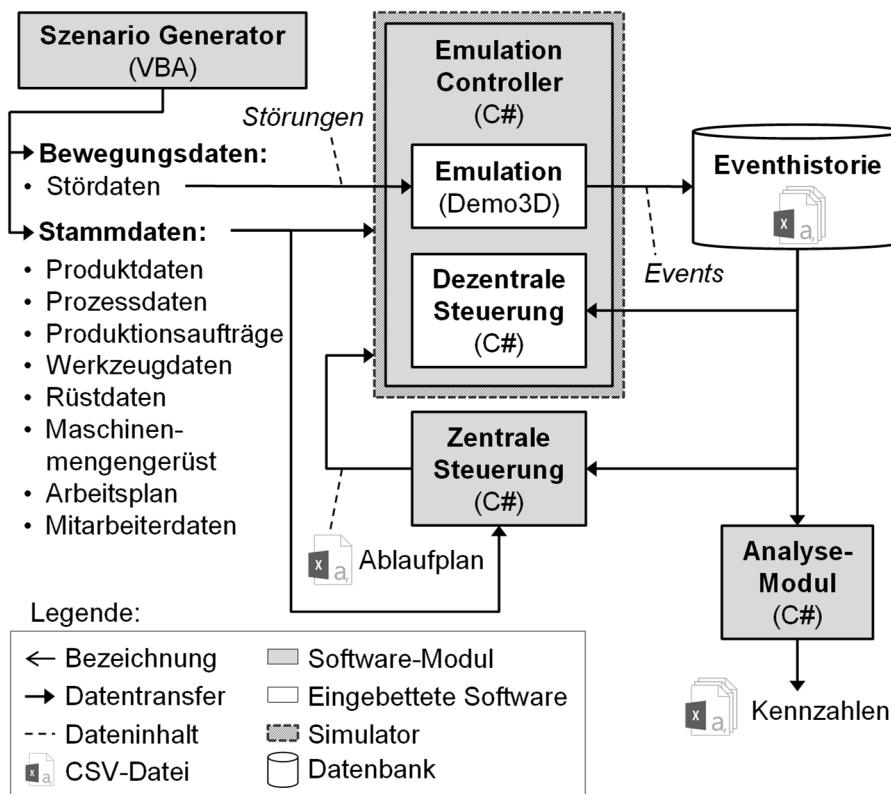


Abbildung 3: Software-Architektur für die Simulation

2.2.1 Emulation

Der Simulationskern enthält nur minimale Steuerungsintelligenz und führt primitive Befehle des Emulation Controllers aus. Es werden lediglich rudimentäre Restriktionen berücksichtigt (z. B. Anzahl freier Pufferplätze, Werkzeugstandmengen, Rüstrestriktionen). Die wesentliche Aufgabe des Simulationskerns besteht darin, die Befehle des Emulation-Controllers (insofern regelkonform) abzuarbeiten und zu den

unterschiedlichen Status der Prozesse entsprechend Events zu generieren und diese zu kommunizieren (Mitarbeiter-, Maschinen- und Auftragsstatus). Des Weiteren obliegt der Demo3D-Komponenten die Visualisierung des zu simulierenden Systems.

2.2.2 *Emulation-Controller*

Der Emulation-Controller (EC) ist innerhalb von Demo3D in C# implementiert und hat die Aufgabe der Steuerung des Gesamtsystems. Er gibt zu einem entsprechenden Zeitpunkt Befehle zur Auftragsbearbeitung an den Simulationskern und verfolgt über die Verarbeitung der Statusmeldungen (Events) aus diesem den Verlauf der Abarbeitung. Zu den Befehlen gehören

- der Transport von Korbstapeln zu den Bereichspuffern der Operationen,
- das Beladen einer dedizierten Fertigungszelle mit einem Korbstapel,
- das Entladen der Fertigungszelle,
- das Rüsten einer Fertigungszelle,
- den Werkzeugwechsel einer Fertigungszelle und
- die Behebung einer Störung an einer Fertigungszelle.

Neben der Steuerung der Simulation (man könnte hier auch von einer Emulation sprechen, da die Logik innerhalb des Demo3D-Moduls auf das reine Ausführen von Kommandos reduziert wird) bereitet der Emulation-Controller die Eventhistorie auf und generiert die Datenbasis zur Berechnung der Kennzahlen durch das Analyse-Modul (KPI-Manager). Hierfür werden die Statusmeldungen, die zu den entsprechenden Zeitpunkten der Abarbeitung generiert werden, konsolidiert und zu Zeitdauern und Typen zusammengefasst und exportiert.

Für die Steuerung der Simulation wird entweder eine dezentrale Steuerungslogik verwendet, die innerhalb des ECs implementiert ist und basierend auf dem aktuellen Zustand des Systems und einer einfachen Logik Entscheidungen trifft, oder es wird ein (in Visual Studio eigens für die Simulation entwickeltes) Software-Modul zur Ablaufplanung (Scheduling) gestartet.

2.2.3 *Dezentrale Steuerung*

Die dezentrale Steuerung wurde auf Basis eines in Scholz-Reiter et al. (2005) vorgestellten generischen Algorithmus realisiert, welcher in anderen Publikationen als Queue Length Estimator (QLE) bezeichnet wird (Scholz-Reiter et al. 2010; Windt et al. 2010a). Bei diesem wird vor jedem anstehenden Arbeitsgang und der damit verbundenen Entscheidung für eine geeignete Maschine, die jeweiligen aktuell möglichen Fertigstellungszeitpunkte aller zur Verfügung stehenden Maschinen berechnet. Um diese berechnen zu können, müssen alle sich in der Warteschlange vor einer Maschine befindlichen Arbeitsaufwände und die daraus resultierenden Rüst- und Werkzeugwechsellaufwände berechnet werden. Am Ende wählt das logistische Objekt die Maschine mit dem frühestmöglichen Fertigstellungstermin in dem betrachteten Arbeitsgang aus.

Die Auswahl des QLE wurde getroffen, weil sowohl innerhalb einer umfassenden Studie zum Vergleich von dezentralen Algorithmen in unterschiedlichen Anwendungsszenarien (Windt et al. 2010a) als auch in der Analyse von Studien zentraler Algorithmen in unterschiedlichen Anwendungsszenarien (Scholz-Reiter et al. 2010) vielversprechende Ergebnisse erzielt wurden.

2.2.4 Zentrale Steuerung

Zu Beginn eines jeden Simulationslaufs wird ein Schedule für einen definierten Zeitraum berechnet, welcher zusätzlich sowohl zyklisch als auch ereignisgesteuert aktualisiert wird (Rescheduling). Der Einsatz in einer Umgebung, die mit ca. 1.000-facher Echtzeit läuft, stellt besondere Anforderungen an die Performanz eines solchen Schedulers.

Der im Rahmen des Projektes iWePro entwickelte Scheduler beruht auf dem Prinzip des Beam-Search (Sabuncuoglu und Bayiz 1999). Dabei werden wie beim Dynamic-Programming sukzessive von einem Startzustand aus Nachfolgezustände erzeugt – je nach Anforderungen an die Laufzeit werden mehr oder weniger viele mögliche Zustände erzeugt bzw. es wird mehr oder weniger stark gefiltert. So kann die Laufzeit beliebig genau gesteuert werden.

Im Extremfall wird in jedem Schritt nur genau ein Nachfolgezustand erzeugt, die Entscheidung, welcher der möglichen Zustände gewählt wird, kann analog der Logik des Queue-Length-Estimators getroffen werden. Es entsteht quasi eine Queue-Length-Estimator-Depth-First-Search-Heuristik. Diese wurde innerhalb der Simulationsumgebung implementiert und erlaubt ein Rescheduling im zeitlichen Bereich von deutlich unter einer Sekunde.

3 Erkenntnisse

Viele umfangreiche Simulationsläufe bilden die Voraussetzung für statistisch fundierte Ergebnisse. Für einen vertretbaren Zeitaufwand eines Simulationslaufs ist die Performanz der Simulationsumgebung ausschlaggebend. Bei der Entwicklung der Simulationsumgebung wurden zwei wesentliche Erfolgsfaktoren zur Erreichung der nötigen Performanz identifiziert. Neben der reinen Simulation der Prozesse zeigen sich zwei wesentliche Aspekte für einen Mehraufwand an Rechenzeit (Overhead):

- Kommunikation mit externen Komponenten und
- das Rescheduling.

Die Kommunikation über Web-Services hat sich als sehr langsam erwiesen, sie drosselte die Geschwindigkeit des Gesamtsystems ca. auf ein 10-faches der Echtzeit. Dagegen erreichte die Kommunikation über eine Datenbank mit Triggern oder über CSV-Dateien Geschwindigkeiten von ca. 100-facher Echtzeit. Die Integration der dezentralen Steuerung und des initialen Schedulers als eigene Software-Module erlaubt aktuell eine Simulationsgeschwindigkeit mit ca. 1.000-facher Echtzeit.

Bezüglich des Scheduling hat der Rescheduling-Zyklus signifikanten Einfluss auf die Ergebnisse der zentralen Steuerung, sobald Störungen das System beeinträchtigen. Die Häufigkeit und Dauer des Rescheduling wirken sich direkt auf die Dauer eines Simulationslaufes aus (die Simulation wird während des Rescheduling gestoppt). Zusätzlich hat es sich als außerordentlich schwierig erwiesen, ein Full-Scale-Rescheduling zu integrieren, ohne die Laufzeit so stark zu reduzieren, dass die Simulationsumgebung noch sinnvoll nutzbar ist.

Dieses Problem kann an vielen Stellen angegangen werden: Zunächst sollte man einen Algorithmus wählen, der auch bei verkürzter Rechenzeit noch sehr gute Lösungen erzielt. Wenn der notwendige Rescheduling-Zyklus bekannt ist, sollte man den Zeithorizont des Scheduling einschränken und nur die naheliegende Zukunft planen.

Mit diesen zwei Maßnahmen kann die Rechenzeit akzeptabel kurzgehalten werden, aber es wird der Einsatz eines speziell für das Problem zugeschnittenen Algorithmus nötig.

4 Ausblick

Eine quantitative Analyse der Vor- und Nachteile einer zentralen bzw. dezentralen Steuerung sind allgemein nicht möglich, und eine umfassende Untersuchung steht noch aus. Es können aber grundsätzliche Erkenntnisse gewonnen werden, die hier zusammengefasst dargestellt werden.

Grundsätzlich kann gesagt werden, dass sich die Ergebnisse nicht stark unterscheiden, wenn eine Simulation mit stark deterministischem Umfeld und hoher Auslastung durchgeführt wird, was nicht verwundert, da beide Steuerungsarten die gleichen Entscheidungsmuster befolgen.

Die typische Auslegung in der Praxis sieht allerdings vor, in der ersten Bearbeitungsstufe einen Engpass zu installieren und in nachfolgenden Schritten „Puffer“ vorzusehen, welche „Lücken“ in der Bearbeitung zur Folge haben. Wenn diese Grundauslegung stark ausgeprägt ist, hat die zentrale Planung einen starken Vorteil. Durch den größeren Planungshorizont der zentralen Steuerung kann ein nötiges Rüsten innerhalb der Leerlaufzeiten vorgezogen werden kann (Belegungsplan ist schon vorhanden). Bei der dezentralen Planung erfolgt das Rüsten immer erst, wenn der Korbstapel sich an der Fertigungszelle „anmeldet“ bzw. zum Zeitpunkt der Steuerungsentscheidung – dies hat Einbußen bei der Durchlaufzeit der Aufträge bei der dezentralen Steuerung zur Folge.

Offensichtlich hat die dezentrale Steuerung einen Vorteil, je mehr und länger Störungen im System auftreten, und je seltener ein Rescheduling angestoßen wird. Dieser Effekt verstärkt sich durch zwei Faktoren: Je stärker das Gesamtsystem ausgelastet wird, desto länger dauert es, bis die Korrektur des Reschedulings greift. Zum anderen hat der Prozess einen Einfluss: Je früher eine irreversible Zuordnung eines Korbstapels zu einer Fertigungszelle getroffen wird (z. B. bei Verlassen des Bereichspuffers) und je größer die Puffer/Zeiten von dieser Entscheidung bis zur Bearbeitung sind (z. B. durch die Anzahl der Pufferplätze vor den Fertigungszellen oder die Integration einer automatisierten Beladezelle mit mehreren Stellplätzen), desto stärker hat eine „späte“ Entscheidung der Zuordnung eines Korbstapels zu einer Fertigungszelle einen positiven Einfluss auf die Vermeidung von starken Verspätungen, was eher einer dezentralen Steuerung zugutekommt.

Grundsätzlich kann gesagt werden, dass bei bisherigen Analysen die Komplexität des Gesamtsystems (heterogene Rüstzeiten, Werkzeugwechsel, Störgruppen) zu keinen deutlichen Unterschieden in der Leistungsfähigkeit von dezentralen bzw. zentralen Steuerungssystemen geführt hat.

Auf Basis dieser grundsätzlichen Erkenntnisse werden als nächste Schritte umfangreiche Simulationsläufe auf konkrete zu untersuchende Szenarios mit dem Ziel einer quantitativen Analyse durchgeführt, um für praktische Überlegungen zur Auslegung und Steuerung ähnlicher Systeme in der Praxis eine Hilfestellung zu bieten.

Danksagung

Die Autoren danken dem Bundesministerium für Bildung und Forschung (BMBF) für die Förderung des Verbundforschungsvorhabens iWePro „Intelligente selbstorganisierende Werkstattproduktion“ im Rahmen der Fördermaßnahme „Intelligente Vernetzung in der Produktion – Ein Beitrag zum Zukunftsprojekt Industrie 4.0“.

Literatur

- Abele, E.; Reinhart, G.: Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen. München: Carl Hanser 2011.
- Brackel, T.v.: Adaptive Steuerung flexibler Werkstattfertigungssysteme: Nutzung moderner Informations- und Kommunikationstechnologien zur effizienten Produktionssteuerung unter Echtzeitbedingungen. Wiesbaden: Gabler 2009.
- Grundstein, S.; Schukraft, S.; Görges, M.; Scholz-Reiter, B.: Interlinking central production planning with autonomous production control. In: Marascu-Klein, V. (Hrsg.): Advances in Production, Automation and Transportation Systems, 2013, S. 326-332.
- Grundstein, S.; Schukraft, S.; Scholz-Reiter, B.; Freitag, M.: Evaluation system for autonomous control methods in coupled planning and control systems. In: Teti, R. (Hrsg.): 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Red Hook, NY, USA: Curran, 2015, S. 121-126.
- Lödding, H.: Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration. Berlin, Heidelberg: Springer Vieweg 2016.
- Niehues, M.: Adaptive Produktionssteuerung für Werkstattfertigungssysteme durch fertigungsbegleitende Reihenfolgebildung. Technische Universität München, Dissertation 2016.
- Nyhuis, P.; Wiendahl, H.-P.: Fundamentals of production logistics: Theory, tools and applications. Berlin [u. a.]: Springer 2009.
- Otto, F.; Uhlmann, E.: iWePro – Intelligente selbstorganisierende Werkstattproduktion. In: Krüger, J.; Uhlmann, E.; Verl, A.; Klemm, P.; Brecher, C. (Hrsg.): Industrie 4.0. Düsseldorf: VDI Verlag 2016, S. 71-85.
- Ouelhadj, D.; Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling 12 (2008) 4, S. 417-431.
- Philipp, T.; Böse, F.; Windt, K.: Autonomously Controlled Processes – Characterisation of Complex Production Systems. In: Cunha, P.; Maropoulos, P. (Hrsg.): Proceedings of 3rd CIRP Conference in Digital Enterprise Technology, Setubal, Portugal, 2006, (ohne Seitenzahlen).
- Ramsauer, C.: Dezentrale PPS-Systeme: Neue Strukturen bei hoher Innovationsdynamik. Wiesbaden: Gabler 1997.
- Sabuncuoglu, I.; Bayiz, M.: Job shop scheduling with beam search. European Journal of Operational Research 118 (1999), S. 390-412.
- Schmidt, M.; Fronia, P.; Fisser, F.; Nyhuis, P.: Decentralized planning and control for assembly areas driven by Gentelligent® parts. In: Helander, M. (Hrsg.): IEEM 2007. [Piscataway, NJ]: IEEE 2007, S. 1088-1092.
- Scholz-Reiter, B.; Freitag, M.: Autonomous processes in assembly systems. CIRP Annals 56 (2007) 2, S. 712-729.

- Scholz-Reiter, B.; Freitag, M.; Beer, C. de; Jagalski, T.: Modelling and analysis of autonomous shop floor control. In: Proceedings of 38th CIRP International Seminar on Manufacturing Systems, Florianopolis, Brasilien, 2005, S. 1-6.
- Scholz-Reiter, B.; Höhns, H.: Selbststeuerung logistischer Prozesse mit Agentensystemen. In: Schuh, G.; Stich, V. (Hrsg.): Produktionsplanung und -steuerung 2. Berlin: Springer Vieweg 2012, S. 294-335.
- Scholz-Reiter, B.; Rekersbrink, H.; Görge, M.: Dynamic Flexible Flow Shop Problems - Scheduling Heuristics vs. Autonomous Control. CIRP Annals - Manufacturing Technology 59 (2010) 1, S. 465-468.
- Schuh, G.: Produktkomplexität managen: Strategien - Methoden - Tools. München [u. a.]: Carl Hanser 2005.
- Schuh, G.; Brosze, T.; Meier, C.: Gestaltung der innerbetrieblichen Produktionsplanung und -steuerung. In: Schuh, G.; Stich, V. (Hrsg.): Produktionsplanung und -steuerung 1. Berlin: Springer Vieweg 2012, S. 297-305.
- Schuh, G.; Stich, V. (Hrsg.): Produktionsplanung und -steuerung 1: Grundlagen der PPS. Berlin: Springer Vieweg 2012.
- Schukraft, S.; Grundstein, S.; Scholz-Reiter, B.; Freitag, M.: Evaluation approach for the identification of promising methods to couple central planning and autonomous control. International Journal of Computer Integrated Manufacturing (2015), S. 1-24.
- Uhlmann, E.; Schallock, B.; Otto, F.: Intelligente Werkstattfertigung. wt-online 105 (2015) 4, S. 184-189.
- Windt, K.; Becker, T.; Jeken, O.; Gelessus, A.: A classification pattern for autonomous control methods in logistics. Logistics Research 2 (2010a) 2, S. 109-120.
- Windt, K.; Philipp, T.; Böse, F.; Becker, T.: Application of the 3-component evaluation system for autonomous control in logistics. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture (2010b) 224, S. 1267-1276.