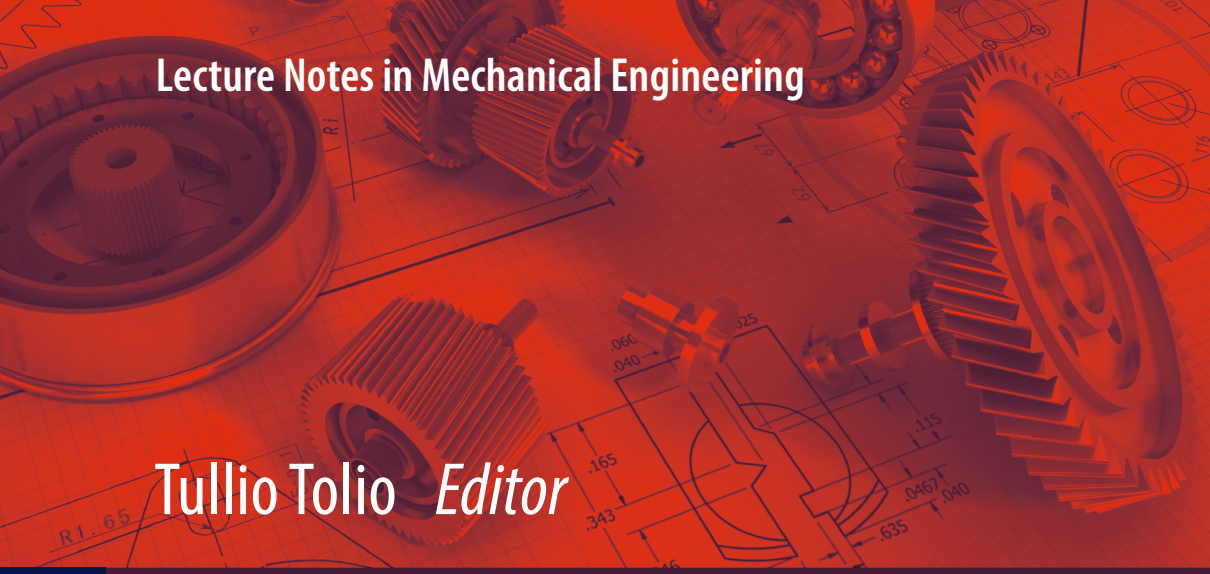
Lecture Notes in Mechanical Engineering



Tullio Tolio *Editor*

CIRP Novel Topics in Production Engineering: Volume 2

Springer

Lecture Notes in Mechanical Engineering

Lecture Notes in Mechanical Engineering (LNME) publishes the latest developments in Mechanical Engineering—quickly, informally and with high quality. Original research or contributions reported in proceedings and post-proceedings represents the core of LNME. Volumes published in LNME embrace all aspects, subfields and new challenges of mechanical engineering.

To submit a proposal or request further information, please contact the Springer Editor of your location:

Europe, USA, Africa: Leontina Di Cecco at Leontina.dicecco@springer.com
China: Ella Zhang at ella.zhang@cn.springernature.com
India, Rest of Asia, Australia, New Zealand: Swati Meherishi at swati.meherishi@springer.com

Topics in the series include:

- Engineering Design
- Machinery and Machine Elements
- Mechanical Structures and Stress Analysis
- Automotive Engineering
- Engine Technology
- Aerospace Technology and Astronautics
- Nanotechnology and Microengineering
- Control, Robotics, Mechatronics
- MEMS
- Theoretical and Applied Mechanics
- Dynamical Systems, Control
- Fluid Mechanics
- Engineering Thermodynamics, Heat and Mass Transfer
- Manufacturing Engineering and Smart Manufacturing
- Precision Engineering, Instrumentation, Measurement
- Materials Engineering
- Tribology and Surface Technology.

Indexed by SCOPUS, EI Compendex, and INSPEC.

All books published in the series are evaluated by Web of Science for the Conference Proceedings Citation Index (CPCI).

To submit a proposal for a monograph, please check our Springer Tracts in Mechanical Engineering at https://link.springer.com/bookseries/11693.

Tullio Tolio
Editor

CIRP Novel Topics in Production Engineering: Volume 2

Springer

Editor
Tullio Tolio
Department of Mechanical Engineering
Politecnico di Milano
Milan, Italy

ISSN 2195-4356 ISSN 2195-4364 (electronic)
Lecture Notes in Mechanical Engineering
ISBN 978-3-032-04438-9 ISBN 978-3-032-04439-6 (eBook)
https://doi.org/10.1007/978-3-032-04439-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Editorial Board

Preface

Le but d'une encyclopédie est de rassembler les connaissances éparses sur la surface de la terre; d'en exposer le système général aux hommes avec qui nous vivons, et de les transmettre aux hommes qui viendront après nous

—Denis Diderot

The evolution of Encyclopedias mirrors the growth of human knowledge. From the papyrus scrolls of antiquity to the digital libraries of today, these compendia have served as the bedrock of learning and intellectual exploration.

Starting from the most famous Encyclopedia in modern Western history, the "*Encyclopédie, ou dictionnaire raisonné des sciences, des arts et des métiers*" by Didier Diderot and Jean le Rond d'Alembert, Encyclopedias have been developed "*to gather the scattered knowledge across the surface of the earth and to expose its general system to the men with whom we live, and to transmit it to the men who will come after us*".

Throughout history, Encyclopedias have been created by Academies. An Academy is "*a society or institution of distinguished scholars and artists or scientists that aims to promote and maintain standards in its particular field*" *(Oxford Dictionary of English).* As a result, academia has come to represent the cultural accumulation of knowledge and its development and transmission across generations.

In modern society, access to documents and knowledge repositories is more open and easier. Nevertheless, knowledge is still limited to limited groups of people in scientific and technological areas.

The extreme specialisation of some areas has driven the emergence of groups of specialists sharing common background knowledge, which consists of language, methodologies, technologies and tools. People external to these groups cannot access the whole set of knowledge, nor do they fully understand the available literature.

Publications tend to be very specific. They are typically focused research papers presenting and demonstrating advances in the field or keynote papers defining the state of the art in a given scientific area and providing information about future development possibilities. These contributions are written, revised and read by the

members of that specific community, and the typical audience is the community of researchers in the field.

Acquiring the background knowledge to understand the language and contents of these publications should be grounded in intermediate sources of knowledge, which provide the reader with the background concepts, language, theory, etc.

When referring to general scientific and technological areas (e.g. mathematics and mechanical engineering), these sources of knowledge are usually books (reference books, handbooks, etc.).

On the contrary, when moving to more novel areas, intermediate books, bringing the reader to a more advanced level of knowledge, are not available.

Given that the availability, accessibility and diffusion of knowledge have drastically changed in the last centuries, the original mission of Encyclopedias to bring knowledge to a broader range of people still applies today to the dissemination and consolidation of knowledge in novel scientific and technological specialisation areas.

As a recognised academy in production engineering, CIRP, after completing a significant endeavour towards the publication of the CIRP Encyclopedia of Production Engineering, is now presenting the CIRP Novel Topics in Production Engineering (CNTPE), a collection of specialised essays in novel areas of production engineering. The present book is the second volume of this editorial initiative.

In an era marked by rapid technological changes and increasing specialisation, the need for a comprehensive, up-to-date resource in production engineering is more pressing than ever. CIRP, through CNTPE, rises to this challenge, offering a perspective on current research, methodologies and advancements to serve the needs of students, researchers and practitioners in the field.

Unlike traditional research, keynote or review papers, the essays in CNTPE are designed to offer a comprehensive, systematic exposition of novel research areas, including original contributions of the authors. Each essay, written to address both the experienced professional and the young researchers approaching a technological area, combines a deep dive into specific topics with a broad contextual understanding of their place within the larger field of production engineering. Whenever possible, hands-on examples and real cases are presented to allow one to enter the details of the field without the need for previous experience. This approach ensures that each contribution is not only a presentation of findings but also a guide for the reader through the complexities of the subject and the specificity of the jargon. Therefore, the essay represents a first landing point for readers interested in understanding the topic and entering the field.

The contributors to the essays represent a diverse, global community of experts rooted within the CIRP but also involving experts from other scientific organisations and institutions. Their insights and expertise bring the highest academic and industrial excellence standards to the CNTPE.

The editorial process is led by the Editorial Board, whose members belong to the various Scientific Technical Committees (STCs) of CIRP to guarantee coverage of all the scientific and technological areas of interest of the CIRP Academy. Their

professional and dedicated effort has been essential to ensuring the accuracy, relevance and clarity of the contributions, maintaining the integrity and quality that the CIRP community is renowned for.

Each volume contains essays whose topics are selected by the various STCs during the meetings at the CIRP General Assembly. The authors of the essays are specialists in the topic and may belong to the CIRP community or be external scientists. By this process, the community defines the novel topics that are important to cover in the various areas of interest of CIRP.

Therefore, even if the editorial process requires that essays be published in separate books, the series of volumes can be seen as a unique endeavour to cover novel topics in production engineering.

This second volume features (see the Table of Contents—ToC) contributions addressing the emerging fields of research in manufacturing linked to technical language processing exploiting AI-enhanced and natural language processing (ToC/1); the implementation of in-line control techniques and real-time adaptive processes to improve quality (ToC/2); methods and tools fir milling stability (ToC/3); and the assessment of the impact of release control policies in manufacturing systems using discrete-event simulation tools (ToC/4).

As we look forward, regular updates, new volumes and the inclusion of emerging topics are all part of our ongoing effort to ensure that the CNTPE remains an essential resource for the production engineering community. As you delve into this volume, it is our hope that it will serve as a valuable guide in your scientific and/or professional career in production engineering, inspiring new ideas, collaborations and advancements.

Milan, Italy
December 2024

Prof. Tullio Tolio
Editor in Chief—CIRP Novel Topics
in Production Engineering

Contents

STC A—Life Cycle Engineering and Assemble

Technical Language Processing in Manufacturing Applications

Fazel Ansari, John Ahmet Erkoyuncu, Julian Kölbl, and Rok Vrabič

Abstract The manufacturing data space represents multi-modal and multi-structural data. Over the past years, the majority of research has been focusing on analysis of structured data captured from IT or operational technology (OT) systems. However, the industrial knowledge-base encompasses unstructured data sources, in particular text, which features documented human and organizational knowledge. Yet, the body of knowledge in manufacturing research and also industrial innovation lag behind in untapping the potentials of textual data. This essay aims at deepening insights into technical language processing (TLP), as an emerging field of research in manufacturing. TLP is a human-in-the-loop and AI-enhanced approach to tailor methods and tools in natural language processing (NLP) using manufacturing data. The essay discusses the foundations of TLP and NLP, with a special attention to the emergence of large language models (LLMs) and associated challenges for industrial applications.

Keywords Artificial intelligence · Knowledge management · Natural language processing

F. Ansari (✉) · J. Kölbl
TU Wien, Vienna, Austria
e-mail: fazel.ansari@tuwien.ac.at

J. Ahmet Erkoyuncu
Cranfield University, Bedford, UK
e-mail: j.a.erkoyuncu@cranfield.ac.uk

R. Vrabič
University of Ljubljana, Ljubljana, Slovenia
e-mail: rok.vrabic@fs.uni-lj.si

T. Tolio (ed.), *CIRP Novel Topics in Production Engineering: Volume 2*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-3-032-04439-6_1

1 Technical Language Processing (TLP)

1.1 Background and Terminologies

Natural language processing (NLP) is "…a field that addresses various ways in which computers can deal with natural–that is, human–language" [25]. Human language is represented in a written form as text. From a computational linguistic perspective, text is an unstructured form of data that can be interpreted as human-readable text, consisting of a sequence of strings, called words [22]. Text, however, is not limited to a certain amount of strings. It can even be composed of a collection of unstructured documents [44]. Since text consists of word strings, it can only be interpreted and understood by the appliance of a wide range of rules, known as grammar [22].

NLP has a long-term history, as depicted in Fig. 1 In 1950s, the historical evolution of NLP approaches was begun. Early NLP approaches relied on rule-based and template-based systems, where linguists and language experts hardcoded patterns and rules for tasks like translation and information extraction. These methods utilized expert knowledge. However, human language's diversity and ambiguity posed challenges, limiting the rule-based method's generalization ability [25]. In the 1980s, the introduction of statistical approaches and machine learning (ML) algorithms marked significant progress in NLP tasks, leveraging language data and statistics to improve performance [25]. While rule-based approaches in NLP rely on precise but inflexible rules, statistical approaches learn from data without assumptions, allowing prediction flexibility [25]. However, statistical methods require large, high-quality datasets to perform well. Eventually, with the continuous development of algorithms and the growth in available data, novel approaches capable of learning efficiently from vast datasets were required. In the 2010s, advancements in computer hardware enabled the adoption of powerful DL techniques, which gained significant prominence in the last couple of years [25].

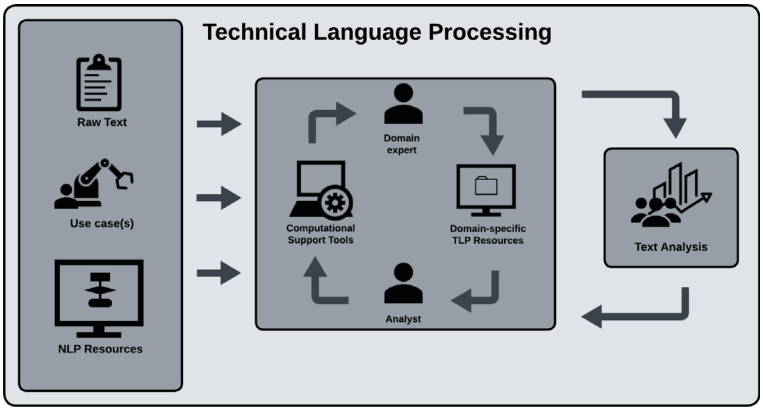


Fig. 1 NLP Evolution over the past decades

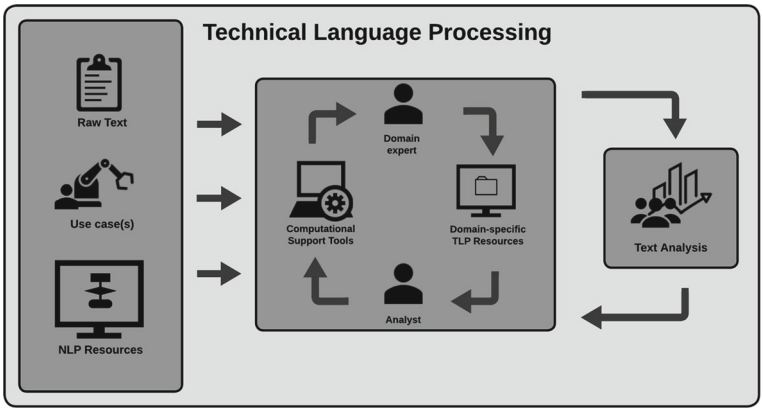


Fig. 2 TLP Framework according to NIST

Besides, NLP aims at explicating knowledge and extracting information from textual data. Knowledge extraction usually involves analyzing and processing textual data, using information retrieved from storage [1, 22]. Thereby the expressions "Knowledge Discover from Text (KDD)" and "data mining on text (aka text mining)" are often used interchangeably in literature referring to the NLP.

Application of (universal) NLP methods in engineering domain confronts significant problems in terms of identifying and understanding engineering, context-, sector- or company-specific terms. Hence the term "Technical Language Processing" (TLP) is coined as and AI-enhanced and "…a human-in-the-loop, iterative approach to tailor NLP tools to engineering data." (Brundage et al., 2021). In particular, TLP is to tailor NLP methods and tools to engineering text-based data, as illustrated in the framework proposed by the National Institute of Standards and Technology (NIST) (cf. Fig. 2).

Notably, the term "NLP resource" in Fig. 2 refers to NLP libraries, dictionaries, taxonomies etc. used for processing and analyzing textual data.

TLP is a socio-technical system maintained by individuals rather than as an algorithmic pipeline [13]. It aims to utilize domain-specific taxonomies and data dictionaries to ensure that industrial tools understand and process all relevant technical terms correctly [15]. Further, it encourages the choice of the most practical techniques for developing industrial applications, achieving a thoughtful balance between analytical performance and available resources. Eventually, TLP should help mitigate the accumulation of systemic technical bias [13].

Table 1 Sample report for pre-processing

Original sample text	Load point interface error! shuttle is not in!

1.2 Text Preprocessing

In TLP, alike NLP, processing and analyzing text depends on quality of raw data and thus it involves pre-processing steps. Pre-processing typically includes tokenizing, removal of stop-words, lowercasing, lemmatizing (stemming), part of speech tagging, parsering and dependency analyzing [8]. Considering a sample text (cf. Table 1), the pre-processing steps are exemplified and applied in Python using the nltk (natural language tool kit) module as follows:

Tokenizing is usually the first step, done in text pre-processing. By applying tokenizers to text, the text is split into tokens. The step of splitting up text into tokens is essential for further knowledge extraction. Each token is an instance of a type, for this reason, the number of tokens outnumbers the number of types [44]. Applying the word tokenizer in python, the following output is provided (see Table 2):

Stop-Words are the words, that almost never have any important meaning, such as articles like 'a' and 'the' or pronouns such as 'you' and 'its'. Therefore, in many cases they are removed. Removing stop words in python leads to the following output, displayed in Table 3.

Lowercasing is a simple pre-processing step and labels the technique of lowercasing the tokens (cf. Table 4). Because of its simplicity, lowercasing is often used in deep learning packages and word embedding packages. Besides that, the sparsity and the vocabulary size are reduced. However, it has to be considered that lowercasing can also impact the system's performance negatively by increasing the ambiguity. For instance, "Apple", the company and "apple" the fruit would be considered as the same entity [8].

Lemmatizing, also known as stemming, is the art of converting tokens to its lemma as shown in Table 5. For document-classification, it brings some advantages. However, for neural network-based approaches, this technique is used rarely. With stemming the sparsity and therefore the number of distinct types in a text is reduced drastically. In contrast, important syntactic nuances can be missed. Some algorithms benefit from stemming techniques, i.e. algorithms that take the frequency under account, while for others this pre-processing step does not bring specific benefits [8].

Table 2 Tokenized sample report

Original	Load Point Interface Error! Shuttle is not in!
Processed	'Load', 'Point', 'Interface', 'Error', '!', 'Shuttle', 'is', 'not', 'in', '!'

Table 3 Removal of stop-words from sample report

Original	Load Point Interface Error! Shuttle is not in!
Processed	Load Point Interface Error! Shuttle!

Table 4 Lowercasing of sample report

Original	Load Point Interface Error! Shuttle is not in!
Processed	Load point interface error! shuttle is not in!

Table 5 Stemming of sample report

Original	Load Point Interface Error! Shuttle **is** not in!
Processed	Load Point Interface Error! Shuttle **be** not in!

Table 6 POS of the test report

Tags	Meaning
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
POS	Possessive ending
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
WDT	Wh-determiner

Tokens are basically allocated to its part of speech (noun, adverb, etc.). There are as many as 179 speech tags, the most important tags are displayed in Table 6. Because relations of entities are usually defined through verbs, the art of **part of speech tagging (POS)** is crucial for information extraction from text. POS applied on the sample report is shown in Table 7.

By applying a **parser**, a hierarchical structure from every sentence can be detected (see Fig. 3). The hierarchical structure is presented in the form of a parse tree, where the lower subtrees group parts of speech into syntactically coherent phrases, i.e. noun

Table 7 POS of the test report

Original	Load Point Interface Error! Shuttle **is** not in!
Processed	(Load, NNP), (Point, NNP), (Interface, NNP), (Error, NNP), (!, .), (Shuttle, NNP), (is, VBZ), (not, RB), (in, IN), (!, .)

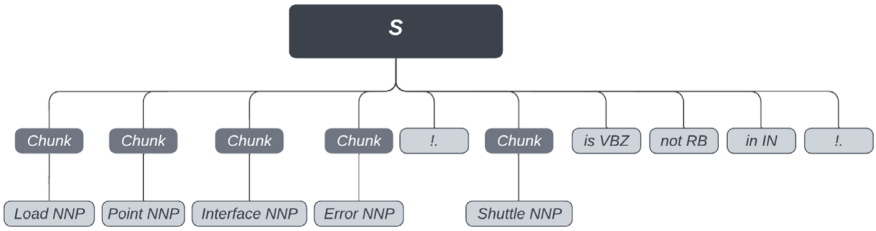


Fig. 3 Parser applied on a sample report

phrases and verb phrases. This is useful in case of the named entity recognition as well as the relationship extraction [2].

The **dependency analyser** is able to check the dependency from words to each other. The dependencies are displayed in the form of dependency graphs, where nodes represent words and direct edges represent dependencies.

A summary of pre-processing techniques, including their application areas is provided in Table 8.

1.3 Overview of NLP Methods and Tasks

NLP is basically classified into two sub-areas [23], namely:

- **Natural language understanding** (NLU) to comprehend and analyze human language by extracting concepts, entities, emotions, and keywords.
- **Natural language generation** (NLG) to produce meaningful phrases, sentences, and paragraphs from an internal representation.

The methods supporting NLU and NLG originates from the overarching field of AI and overlapping with subareas of AI, particularly with machine learning (ML) and deep Learning (DL) as depicted in Fig. 4.

Novel NLP approaches heavily rely on ML and DL techniques. However, specific conventional NLP methods, like word counting and text similarity measurements, are essential building blocks for ML-based models but are not always categorized as ML [18]. In NLP, rule-based, statistical, and DL methods are all utilized based on specific task requirements [25]. The state-of-the-art methods and approaches in

Table 8 Pre-processing techniques

Pre-processing	Summary	Application
Tokenizing	Text is split into tokens–mostly these tokens are words	For almost every text mining algorithm
Removal of stop-words	So called stop-words are removed from the text, because they usually do not provide any further information	Depends on the use case
Lowercasing	Tokens/words are being lowercased, sparsity and vocabulary size is reduced, ambiguity is reduced	For most of deep learning algorithms
Lemmatizing / Stemming	Tokens are converted to its lemma, sparsity and distinct types are reduced, syntactic nuances can be missed	Algorithms that take token frequency under account, document classification
Part of speech tagging	Tokens are assigned to their part of speech	Information Extraction
Parser	Detects the hierarchical structure of a sentence and displays are parse tree	Named entity recognition, relationship extraction
Dependency analyser	Checks dependency of words to each other	Depends on the use case

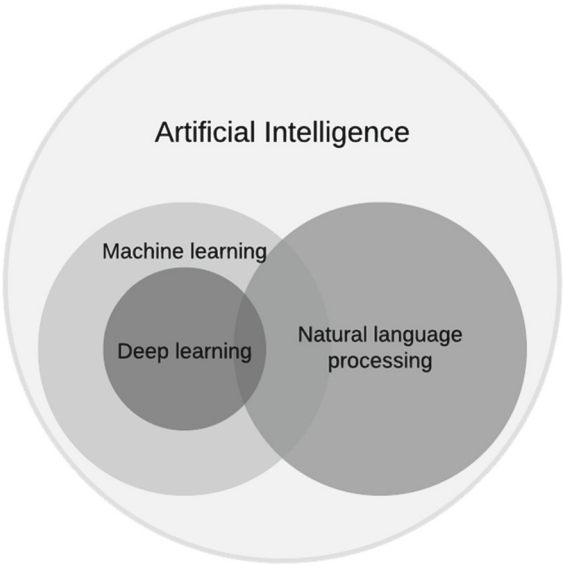


Fig. 4 NLP as a subdomain of AI [18]

NLP are clustered into three categories [42], namely (i) Heuristic and statistic-based NLP, (ii) Machine learning for NLP, and (iii) Deep learning for NLP.

Heuristic and statistic-based NLP: A large body of heuristic-based methods are based on digitizing resources like dictionaries and thesauruses. For instance, lexicon-based sentiment analysis utilizes counts of positive and negative words to predict text sentiment [42]. Focusing on lexical information, semantic relationships between words such as synonyms, hyponyms, and meronyms represented e.g. by ontologies or knowledge graphs facilitates the NLP tasks, especially for extracting and understanding concepts in various application context and domains [42]. Furthermore, regular expressions (regex), also called rational expressions, are tools for searching and manipulating text strings. A regex is a filter that consists of a sequence of characters and metacharacters, accepting a set of strings and rejecting the rest [20]. Regexes enable the integration of domain knowledge into NLP systems. For instance, in customer complaints received via email, regexes can facilitate automating the identification of products [42]. They are used for building deterministic rule-based frameworks like TokensRegex [17], which aid in defining regular expressions to identify text patterns and create rules. The sub-branch of probabilistic regexes introduces probabilities for matching, addressing the deterministic limitations [42].

Machine learning for NLP: ML techniques, including supervised methods like classification and regression, are widely used in NLP tasks such as news article categorization and stock price estimation based on social media posts and inherited networks. One supervised method is the Naive Bayes classifier, which calculates class label probabilities. It uses Bayes' theorem and assumes independence among features [9]. This makes it suitable for text classification tasks, like news article classification based on word counts [42]. In contrast, unsupervised clustering algorithms are mainly deployed for grouping text documents [42].

A support vector machine (SVM) is a binary classifier that implicitly transforms data from the original feature space to a higher-dimensional space using a kernel function. This transformation disentangles the data, making it linearly separable. The process involves mapping the input space to an alternative representation with a higher dimension, enabling the creation of a hyperplane for classification [37]. For instance, SVM is used to enhance failure detection and availability optimization in production systems [4]. Through the training of an SVM using vectorized textual failure reports and associated downtimes, they were able to give early downtime predictions for improving industrial maintenance.

Deep learning for NLP: In recent years, deep neural network architectures in NLP have gained prominence due to the requirement of dealing with complex, unstructured data by processing natural language. Recurrent neural networks (RNN), designed to process sequential data, are capable of understanding language flow and maintaining memory. This memory is temporal, and the information is stored and updated at every time step as the RNN processes each word in the input sentence. This aspect makes them very efficient in solving various NLP tasks such as text classification, named entity recognition, and machine translation. However, traditional RNNs are fundamentally limited by their short memory and struggle with longer contexts [42].

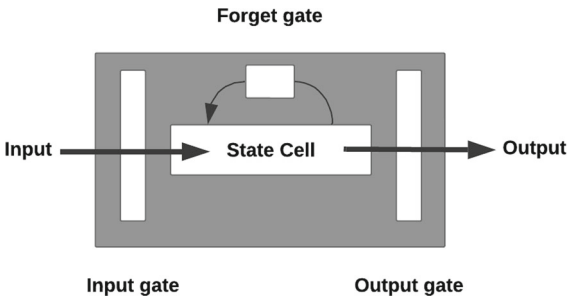


Fig. 5 Simplified architecture of an LSTM cell, showing how relevant information is maintained throughout the encoding process [37]

The long short-term memory networks (LSTM) address the limitations of simple RNNs by including gating operations that control the flow of historical network information into the present [37]. The essential components of LSTM networks are seen in Fig. 5. The input gate applies a weight matrix to the new input. The state cell processes the weighted input and applies a forget gate that weighs information from the previous state cell and also weighs the input using a separate matrix. Finally, the output gate selectively transmits information from the current state cell. Because of their improved memorization capabilities and their ability to focus on relevant contexts, LSTMs have largely replaced RNNs in NLP applications today [37].

Even though LSTM networks have made significant progress in memorization capabilities and performance in common NLP tasks, transformer networks represent the forefront of DL models in the field of NLP. Over the past two years, these models have excelled in nearly all significant NLP tasks, achieving state-of-the-art performance. Unlike sequential methods, transformers do not linearly process textual context. Instead, when given a word as input, they employ a specific self-attention mechanism to consider all words close to it, allowing each word to be represented in relation to its surrounding context [42].

Several common NLP tasks, including NLU and NLG tasks, are solved in various applications today with heuristic-based, ML, and DL methods. The spectrum of such tasks is very diverse, and several tasks are intertwined with each other, such that there are sub-tasks used for higher-level tasks. Depending on the NLP task, it might require solving only NLU or NLG sub-tasks or both. This prohibits clear categorization as summarized below:

- **Information retrieval** aims at representing queries and documents as vectors. This vector representation visualizes queries and documents in a geometrical space. In this space, objects are defined by coordinates, and their proximity determines the similarity between objects. By leveraging this geometrical space representation, the system identifies the document most relevant to the query by locating the one with the most similar content using different NLU methods [25].
- **Text classification (TC)** involves categorizing pieces of text into distinct classes. One notable application is spam filtering, where emails or other text types like web pages are classified as either spam or not [18]. Another essential type of text classification is sentiment analysis, which detects subjective information such

as opinions, emotions, and feelings within the text using various NLU methods. Given these capabilities, sentiment analysis plays a vital role in understanding public sentiment, user reviews, and social media interactions [18].

- **Information extraction** involves extracting information from text sources, such as calendar events from emails or individuals' names in social media posts. In the literature, it is also often referred to as named entity recognition [42].
- **Question-answering (QA)** depends on input and output formats. Extractive QA involves extracting answers from contexts, including text, tables, or HTML. Open generative QA generates free text responses directly from given contexts. On the other hand, closed generative QA generates answers without any context provided, relying entirely on the model's generation capabilities [14].
- **Summarization** condenses lengthy texts while retaining their core meaning, producing shorter text output. This method is particularly valuable for simplifying complex documents such as legislative bills, legal papers, and scientific articles. The two fundamental approaches in summarization are extractive summarization, which involves selecting and extracting important sentences directly from the source text, and abstractive summarization, which generates a summary, potentially incorporating new words not present in the source document [14].
- **Machine translation** deals not only with complexity of translating sentences based on the word translation but also with preserving sentence meaning, grammar, and tenses. This is solved using statistical ML techniques that gather parallel data from multiple sources, calculating the likelihood that language A corresponds accurately to language B or by employing artificial neural networks and DL methods [23].
- **Text generation (TG)** creates natural language text from various sources. If the input data consists of text or other data types, TG can be further divided into text-to-text and data-to-text generation. For instance, dialog systems produce natural utterances based on ongoing conversations, enabling the generation of news text from events such as sports game outcomes and weather conditions [18].

The following SWOT (strengths, weaknesses, opportunities, and threats) in Tables 9 and 10 summarizes the major considerations concerning the state of the art, conventional NLP approaches.

2 TLP in the Era of Generative AI

Generative AI refers to a subset of artificial intelligence techniques and models that focus on generating new content or data that is similar to the data they were trained on [19]. This can include a wide range of outputs such as text, images, music, and even video. The goal of generative AI is not just to analyze and understand data (as is common in other types of AI), but to use this understanding to create new, original content that is indistinguishable from human-generated content.

Table 9 SWOT analysis on conventional NLP/TLP approaches

Strengths	• Text mining makes it possible to extract implicit and explicit information from unstructured and semi-structured data • Unstructured and semi-structured data is converted to structured data and is therefore made accessible to further data mining and knowledge discovery steps • By extracting knowledge from text, informed decisions can be made depending on the extracted data • Via e.g. sentiment analysis, a knowledge extraction technique for text mining, even the sentiment (intrinsic opinion of the authors) of a text can be extracted
Weaknesses	• Further analysis of a pre-processed/tokenized text is usually limited to a specific language. Different languages usually mean different linguistic syntax and grammar rules , i.e. when dealing with a different language many of the applied rules should be reconstructed • The unstructured form of text makes multiple processes necessary in order to come even close to extracting entire knowledge stored in textual data. While a database is usually organized in tables and therefore structured, entities and their attributes in texts are hard to extract. The main reason is that those elements do not have a fixed position in a given text and even the labels of the entities and attributes differ from text to text • ML approaches for NLP consist of tokenizing a string of characters into structures such as words, phrases, sentences and paragraphs and applying then a statistical analysis on the pre-processed text. However, human language includes imitation, comprehension and remembrance. In other words, the underlying knowledge defines the understanding of text. Thus, conventional text mining techniques are still lagging behind capabilities of human in text understanding • Multilingual text mining is still off the norm, since most of text mining tools are using monolingual supervised ML-algorithms. Therefore, e.g. the ML-algorithm used for the sentiment analysis is trained on monolingual labelled training data. If multilingual texts are analyzed the sentiment analysis will present inaccurate results • Context specificity is important, since depending on the application field different expressions are used. In order to still analyze a text accurately, even if the text contains special terms, these expressions must be defined in a context or company specific dictionary • In order to analyze a text at all, the text must be present in form of a digital text or a document, so the text analysis can take place. This is yet a challenge mainly in manufacturing companies

Two key components of generative AI are (1) generative models (e.g. Generative Adversarial Networks (GANs) [16], Variational Autoencoders (VAEs) [24], and Transformers [43]), and (2) the learning process. Generative models are typically trained on a large dataset, learning the underlying distribution of this data. Once trained, they can generate new data points that follow this learned distribution. The quality and realism of the generated content depend heavily on both the model architecture and the breadth and quality of the training data.

Applications of generative AI are diverse and growing, including areas like art creation, realistic video game environments, personalized content generation, deep-

Table 10 SWOT analysis on conventional NLP/TLP approaches cont.

Opportunities	• Conventional text mining is still behind other new techniques, like face recognition or speech recognition. Text mining, therefore, has the opportunity to catch up. This identifies the main advantage of large language models for text analysis and understanding • Text understanding and generation could facilitate reproducing human knowledge and also facilitating human work assistance • Efficient and productive TLP is achieved through establishing shared/open access infrastructure for industrial text mining systems • A huge amount of experiential knowledge of human and organizations in the industrial sector is often stored in previously written reports. By being able to extract the stored knowledge from these reports an improved knowledge management and knowledge protection in industrial value-chains could be achieved • Human failure rate in manufacturing context can be reduced by analyzing previously written reports and provision of feedback or multi-modal learning materials for the purpose of assistance or work-based learning
Threats	• Data quality and availability are major problems for implementation of conventional NLP, especially in industrial context where poor data or lack of data could lead to unreliable outcome of NLP systems • Lack of explainability affects reliability of NLP approaches, i.e. proper interpretation of analyzed text data depends highly on the capability of black box NLP algorithms on text understanding and integration of human prior knowledge in the analytical pipeline

fakes, drug discovery, and more. The field is rapidly evolving, with new models and techniques continually being developed to improve the realism and capabilities of the generated outputs.

Generative AI is also making significant strides in the realm of technical language processing. This involves the application of advanced models to understand, interpret, and generate human language in a way that is contextually and semantically accurate. These advancements are opening up novel applications in areas such as automated content creation, language translation, summarization, and even in the development of interactive AI conversational agents. The ability of these models to process and generate language has vast implications for industries ranging from education to customer service, and even in complex fields like legal and medical document analysis.

The following subsections will go deeper into the technical aspects of generative AI, particularly in the context of text generation. We will start by exploring the fundamentals of neural networks, the backbone of modern AI models. Understanding neural networks is essential to grasp how these models learn from data and how this learning translates into the generation of new, coherent, and contextually relevant textual content. Subsequent sections will cover advanced topics such as the architecture of specific models like Transformers, training methodologies, challenges in language model development, and emerging trends in the field of technical language processing.

2.1 Neural Networks and Their Architectures

The Perceptron Neural networks are a class of powerful algorithms that draw inspiration from the complexities of the human brain. They can be thought of as computational models that are adept at recognizing and interpreting patterns in diverse types of data. The foundational principle behind neural networks is their ability to process sensory data, much like the human brain, enabling them to perform tasks such as categorization, labeling, or understanding raw input. At the core of neural networks lies the concept of pattern recognition. These patterns, inherently numerical in nature, are embedded in vectors. Vectors serve as the universal language for neural networks, translating various forms of real-world data into a format that these algorithms can understand and analyze. This data can range from visual imagery and auditory signals to textual content and time series data. Through this process of translation and interpretation, neural networks learn to make sense of the complex, multifaceted information that characterizes our world and thus provide a framework for machines to not only recognize patterns but also to derive meaningful insights and make informed decisions based on the data they process.

The perceptron, conceptualized by Frank Rosenblatt in 1958 [38], is the initial and seminal model in the evolution of neural networks, mirroring the functional attributes of a singular neuron within the human brain. At its core, the perceptron is a linear binary classifier, functioning as a decision-making unit that computes a weighted sum of its input features. The mathematical representation of a perceptron can be formalized as shown in Eq. 1.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here, x represents the input vector, w denotes the vector of weights, b is the bias term, and $\cdot$ signifies the dot product. The function $f(x)$ yields an output of 1 or 0, based on whether the weighted sum $w \cdot x + b$ is greater than zero or not, respectively. Perceptrons can be connected into a multilayer perceptron architecture

by fully connecting the outputs of perceptrons of the previous layer to the inputs of the perceptrons of the next one.

Despite its pioneering status, the perceptron has inherent limitations. Its most notable constraint is the inability to process data that is not linearly separable. This means that the multilayer perceptron can only find a solution when there exists a linear boundary that can distinctly segregate the data points into separate classes. Consequently, while the multilayer perceptron laid the groundwork for more complex neural network architectures, its application is confined to relatively simple, linearly classifiable tasks.

How Do Neural Networks Learn? Neural networks learn through a process called backpropagation, which is essentially a method for adjusting the weights of the network in response to the error in its output. Initially, input data is fed into the network (forward pass), and a prediction is made. This involves calculating the output of each neuron in each layer, starting from the input layer and moving towards the output layer. The output of each neuron, in a generalized form with respect to the perceptron, is determined by a function of the sum of its inputs, as shown in Eq. 2.

$$a_i^{(l)} = \sigma\left(\sum_j w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)}\right) \tag{2}$$

Here, $a_i^{(l)}$ is the activation of the i-th neuron in the l-th layer, σ is the activation function (like sigmoid, ReLU, etc.), $w_{ij}^{(l)}$ are the weights, and $b_i^{(l)}$ is the bias. The sum is over all neurons j in the previous layer.

Once the output is obtained, the error is calculated. The error is typically the difference between the predicted output and the actual output. The most common way to measure this error is through a loss function, such as Mean Squared Error (MSE) for regression problems, as shown in Eq. 3.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

Here, y_i is the actual value, and $\hat{y}_i$ is the predicted value. Backpropagation is then used to calculate the gradient of the loss function with respect to each weight in the network. This involves applying the chain rule of calculus to compute gradients backwards from the output layer to the input layer, given by Eq. 4

$$\frac{\partial \text{Loss}}{\partial w} = \frac{\partial \text{Loss}}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w} \tag{4}$$

Here, $\frac{\partial \text{Loss}}{\partial a}$ is the derivative of the loss function with respect to the activation, $\frac{\partial a}{\partial z}$ is the derivative of the activation function, and $\frac{\partial z}{\partial w}$ is the derivative of the neuron's output with respect to its weight. Finally, the weights are updated in the direction

that reduces the error. This is typically done using an optimization algorithm like gradient descent, shown in Eq. 5.

$$w_{\text{new}} = w_{\text{old}} - \eta \frac{\partial \text{Loss}}{\partial w} \tag{5}$$

Here, η is the learning rate, a small positive value that determines the size of the step we take in the direction of the negative gradient.

Through iterative application of these steps, the neural network 'learns' by adjusting its weights to minimize the error in its predictions, effectively improving its performance on the given task.

Neural Network Architectures By extending and refining the basic principles of the perceptron, researchers have developed a variety of neural network types, each suited to different applications.

Convolutional Neural Networks (CNNs) are a class of deep neural networks, primarily used in processing data with a grid-like topology, such as images [26]. They are particularly known for their ability to detect and extract hierarchical spatial features from images, which makes them highly effective for tasks like image classification, object detection, and image segmentation. CNNs are structured in layers, with each layer performing specific transformations on its input data. The key components of a CNN include convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a set of filters to the input to create feature maps, highlighting various features of the input. Pooling layers reduce the spatial dimensions (width and height) of the input volume for the next convolutional layer, helping in reducing the computational load and the number of parameters. Finally, fully connected layers, similar to traditional neural networks, perform classification based on the extracted features.

Recurrent Neural Networks (RNNs) are a type of neural network designed for processing sequential data, like time-series data, speech, or text [39]. The fundamental feature of RNNs is their ability to maintain a 'memory' of previous inputs in their internal state, which influences the network's output. This makes them particularly suitable for tasks where context and order of data are important, such as language modeling and speech recognition. In RNNs, connections between nodes form a directed graph along a temporal sequence. This allows them to exhibit temporal dynamic behavior. Unlike traditional neural networks, where each input is processed independently, RNNs share the weights across different time steps, which aids in learning sequences of data.

In the context of language, however, the most important breakthrough is the Transformer architecture, introduced in 2017 [43]. The Transformer represents a significant shift in the approach to sequence-to-sequence tasks such as natural language processing. Distinct from its predecessors that relied on recurrent or convolutional layers, the Transformer adopts a purely attention-driven approach, using mechanisms known as self-attention and multi-head attention to process sequences of data in parallel. This design allows for more efficient handling of dependencies within the data,

irrespective of their distance in the sequence. The Transformer's ability to handle long-range dependencies and its scalability has made it a cornerstone for many subsequent advancements in the field of generative AI, leading to state-of-the-art models in language translation, text generation, and beyond.

2.2 The Transformer Architecture

The Transformer architecture, introduced in [43], marked a significant departure from previous sequence learning methods. Unlike RNNs, Transformers use self-attention mechanisms to process all elements of the input sequence in parallel. This approach allows the model to weigh the importance of different parts of the input data irrespective of their positions in the sequence. Since Transformers do not use recurrence or convolution, they require a means to incorporate the order of the sequence. This is achieved through positional encoding, added to the input embeddings at the bottoms of the encoder and decoder stacks. The Transformer model has become a foundation for many subsequent developments in NLP, leading to state-of-the-art models like BERT, GPT, and others that have revolutionized the field. Its ability to handle long-range dependencies and its parallelizable structure make it highly efficient and effective for a wide range of sequence learning tasks. In the following, the Transformer architecture is explained in the context of Large Language Models (LLMs).

Input Embedding In LLMs, input embeddings are essential for transforming raw text data into a numerical format that the models can process. These embeddings serve as a bridge between the human-readable text and the model's internal numerical vector representations. The process begins with tokenization, where the input text is broken down into smaller units called tokens. These tokens might be words, subwords, or characters. Each token is then mapped to a vector via an embedding lookup. This embedding vector is retrieved from an embedding matrix, which is a learnable component of the model. Positional encodings are added to these embeddings to provide information about the position of each token in the sequence. In some models, token embeddings are combined with other types of embeddings, such as segment embeddings in BERT, to provide additional contextual information.

Embeddings are typically high-dimensional, encompassing hundreds or thousands of dimensions, allowing them to encode substantial information about each token. Contrasting with sparse representations like one-hot encoding, embeddings are dense, with each dimension contributing some information about the token. Initially random, the vectors in the embedding matrix are refined during the training process to capture semantic and syntactic relationships between words.

Multi-head Self-Attention Multi-head self-attention is a mechanism within the Transformer architecture that allows the model to focus on different positions of the input sequence. This mechanism is crucial for understanding the relationships and dependencies between tokens in the sequence. The self-attention mechanism

computes the attention scores for each element in the input sequence, relative to all other elements. For a single attention head, the computation involves three vectors for each input element: Query (Q), Key (K), and Value (V). These vectors are derived from the input by multiplying it with weight matrices W^Q, W^K, and W^V, respectively, as shown in Eq. 6.

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{6}$$

Here, X represents the input sequence. The attention weights are computed using the dot product of Q and K, followed by a softmax operation, as shown in Eq. 7.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Here, d_k is the dimension of the key vectors, and the division by $\sqrt{d_k}$ is a scaling factor. In multi-head attention, the above process is repeated multiple times (heads), each with different, learnable linear transformations (weight matrices). This allows the model to jointly attend to information from different representation subspaces. The multi-head self-attention mechanism allows the Transformer to capture different types of dependencies in the input sequence, making it more powerful and flexible than single-head attention.

The Transformer Block A Transformer block is a key component in Transformer-based models. This block is composed of several key elements that work together to process input data. The first component of a Transformer block is the Multi-head self-attention mechanism. Following the attention mechanism, the output is passed through a position-wise feed-forward neural network, as shown in Eq. 8.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{8}$$

Here, W_1, W_2, b_1, and b_2 are the weights and biases of the feed-forward network, and $\max(0, x)$ represents the ReLU activation function. Each sub-layer in the Transformer block, including the attention and feed-forward layers, has a residual connection around it, followed by layer normalization, as shown in Eq. 9

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{9}$$

This step helps in mitigating the vanishing gradient problem and allows for deeper models. Layer normalization is applied after each sub-layer to stabilize the learning process, as shown in Eq. 10.

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sigma} \odot \gamma + \beta \tag{10}$$

Here, μ and σ are the mean and standard deviation of the inputs, (γ and β) are learnable parameters, and $\odot$ represents element-wise multiplication. The Transformer block, through these components, can effectively process sequences, capturing complex relationships in the data.

2.3 The State-of-the-Art of LLMs and Relevance to Production Engineering

As of January 2024, the state-of-the-art in Large Language Models (LLMs) has seen significant advancements and diversification. Key developments include:

- **ChatGPT, GPT-4, GPT-4.5 Turbo, and GPT-4o** [6]: ChatGPT was released in November 2022. Since then, GPT-4 was released in March 2023, featuring an expanded context window, multimodal processing capabilities, enhanced creativity, and faster training and execution, making it more versatile for a broader range of applications. GPT-4.5 Turbo and GPT-4o are an iteration of ChatGPT with a new data cutoff windows of April and May 2023, supporting up to 128,000 tokens of context, enabling the creation of extremely long and detailed prompts.
- **GPT-4o1** is a family of large language models are trained with reinforcement learning to perform complex reasoning, launched in September 2024. The distinguishing feature of the o1 family is their unique "think-before-you-speak" approach, where models generate extensive internal chains of thought prior to providing user responses. Unlike their predecessors, o1 models dedicate significantly more processing time to information analysis, resulting in more thoughtful and comprehensive outputs.
- **Bard**: Google's response to OpenAI's ChatGPT, Bard, based on PaLM 2 [10], is a 137B parameter LLM capable of generating various creative text formats. It demonstrates sophisticated performance in answering questions in open-ended, challenging, or strange scenarios.
- **Gemini**: Google's Gemini is a highly capable and general family of models with state-of-the-art performance across many leading benchmarks [40]. They includes versions optimized for different sizes: Ultra, Pro, and Nano. Gemini Ultra, in particular, has outperformed human experts on massive multitask language understanding (MMLU) and shown significant capabilities in multimodal tasks.
- **Claude**: Developed by Anthropic, Claude is an LLM-based generative AI model family with capabilities in text generation, language translation, question answering, and creative content creation.
- **Bloom**: Developed by a collaboration involving HuggingFace's BigScience team, Microsoft DeepSpeed team, NVIDIA Megatron-LM team, and others, Bloom is an open-access, multilingual LLM optimized for text generation and language exploration. It can generate text in 13 programming languages and 46 natural languages, showcasing the emphasis on multilingual capabilities in LLMs.

- **Llama2**: Developed by Meta, Llama2 is a series of large language models (LLMs) with parameter sizes ranging from 7 billion to 70 billion [41]. It uses an auto-regressive language model framework and is specifically designed for dialogue applications.
- **Mistral**: Developed by Mistral AI, Mistral 7B is a powerful language model with 7.3 billion parameters [21]. It stands out for its performance and efficiency, outperforming Llama 2 13B on all benchmarks and Llama 1 34B on many benchmarks. Mistral 7B approaches CodeLlama 7B performance on coding tasks while remaining proficient in English language tasks.
- **Falcon**: Falcon models were developed by the Technical Innovation Institute (TII) [35]. They include Falcon-7B, with 7 billion parameters, and Falcon-40B, with 40 billion parameters. These models are causal decoder-only models trained on large datasets, including 1,500 billion tokens and 1 trillion tokens of the RefinedWeb dataset.
- **StableLM**: StableLM, developed by Stability AI, is a suite of open-source language models available in various sizes, including 3 billion and 7 billion parameters. These models are designed to generate text and code efficiently and are trained on a new experimental dataset built on The Pile, but three times larger, with 1.5 trillion content tokens.
- **Grok**: Grok is an AI chatbot developed by xAI, an offshoot of the team that remained after Elon Musk's acquisition of X (formerly known as Twitter). Introduced in November 2023, Grok is positioned as a direct competitor to popular AI models. Grok's distinct approach includes a more unfiltered and edgy communication style compared to other AI chatbots, aiming to offer witty and rebellious interactions. Grok2, an upgraded model, was release in August 2024.

The LMSYS Arena is an open-source platform designed to evaluate and compare large language models (LLMs) through human preference judgments [28]. It hosts a diverse array of AI models from various organizations, allowing users to interact with these models and provide feedback on their responses. The platform then aggregates these human judgments to create a competitive ranking system, similar to chess Elo ratings. The leaderboard is regularly updated to reflect the latest models and user evaluations. Table ?? presents a snapshot of the current LMSYS Arena leaderboard (August 2024), showcasing the top-performing models along with their scores, confidence intervals, and other relevant information (Table 11).

Since the introduction of open LLMs that can be fine-tuned for specific tasks, several areas of research have emerged in the context of production engineering:

- **Process Optimization and Documentation**: LLMs can analyze and generate technical and process documentation, making it easier to update manuals, create training materials, or document quality control procedures [32]. They can help streamline the documentation process, ensuring that all materials are clear, up-to-date, and comprehensible.
- **Predictive Maintenance and Analytics**: By analyzing maintenance logs and reports, LLMs can assist in predictive maintenance [29]. They can process large

Table 11 Model comparison Table, LMSYS Arena snapshot from September 2024

Rank	Model	Score	95% CI	Org.	Lic.	Cutoff
1	o1-preview	1355	+12/–11	OpenAI	Prop.	2023/10
2	ChatGPT-4o-latest	1335	+5/–6	OpenAI	Prop.	2023/10
3	o1-mini	1324	+12/–9	OpenAI	Prop.	2023/10
4	Gemini-1.5-Pro-Exp	1299	+5/–4	Google	Prop.	2023/11
5	Grok-2	1294	+4/–4	xAI	Prop.	2024/03
6	GPT-4o-2024-05-13	1285	+3/–3	OpenAI	Prop.	2023/10
7	GPT-4o-mini	1273	+3/–3	OpenAI	Prop.	2023/10
8	Claude 3.5 Sonnet	1269	+3/–3	Anthropic	Prop.	2024/04
9	Gemini-1.5-Flash-Exp	1269	+4/–4	Google	Prop.	2023/11
10	Grok-2-Mini	1267	+4/–5	xAI	Prop.	2024/03
11	Gemini Advanced App	1267	+3/–3	Google	Prop.	Online
12	Meta-Llama-3.1-405b-fp8	1266	+4/–4	Meta	Comm.	2023/12
13	Meta-Llama-3.1-405b-bf16	1264	+6/–8	Meta	Comm.	2023/12
14	GPT-4o-2024-08-06	1263	+4/–3	OpenAI	Prop.	2023/10
15	Gemini-1.5-Pro-001	1259	+3/–3	Google	Prop.	2023/11
16	GPT-4-Turbo-2024-04-09	1257	+3/–2	OpenAI	Prop.	2023/12

volumes of text data from machine logs to predict when equipment might fail or require servicing, thereby reducing downtime.

- **Training and Skill Development**: LLMs can be used to create customized training programs. By analyzing existing training materials and procedures, they can generate educational content tailored to specific roles or equipment within the manufacturing process.
- **Innovation and Research**: LLMs can aid in research and development by aggregating and summarizing relevant research papers, patents, and technical documents, providing insights and sparking innovation in manufacturing processes and product development.
- **Customer Support and Engagement**: These models can automate and enhance customer support by generating responses to customer inquiries, creating detailed product descriptions, and providing personalized assistance.

However, LLMs currently posses several drawbacks, inhibiting their use in real environments:

- **Reliability and Accuracy Concerns**: One of the primary concerns with LLMs is their reliability. While they can generate coherent and seemingly accurate responses, there's no guarantee that the information provided is correct. In critical situations, such as medical advice, legal counsel, or safety-related issues, reliance on LLMs can be risky due to the possibility of generating incorrect or misleading information. This limitation arises from the fact that these models generate responses based on patterns in data rather than understanding or reasoning.

- **Bias and Ethical Issues**: LLMs are trained on vast datasets sourced from the internet, which often include biased and discriminatory language. As a result, these models can inadvertently perpetuate and amplify biases present in the training data. This is particularly concerning in scenarios where fairness and neutrality are paramount, such as in hiring processes, law enforcement, or financial services. The use of biased language models in these contexts can lead to ethical issues and unfair treatment of individuals based on gender, race, or other characteristics.
- **Lack of Explainability**: LLMs, especially those based on deep learning, are often described as 'black boxes' due to their lack of explainability. Understanding how and why a model has arrived at a particular output is crucial in critical applications. The inability to interpret the decision-making process of LLMs poses a significant challenge in contexts where transparency and accountability are essential, such as in healthcare diagnostics or autonomous vehicle decision systems.
- **Data Privacy and Security**: The use of LLMs in real-life applications raises concerns about data privacy and security. These models require access to vast amounts of data, some of which may be sensitive or personal. Ensuring the confidentiality and security of this data is paramount, particularly in fields like healthcare or finance, where data breaches can have severe consequences.
- **Dependence and Skill Erosion**: Over-reliance on LLMs for tasks like writing, coding, or decision-making might lead to skill erosion among professionals. This dependency could reduce the ability of individuals to perform these tasks without AI assistance, potentially leading to a skills gap in the long term.
- **Resource Intensity**: Training and running LLMs require significant computational resources, which has environmental and economic implications. The energy consumption and carbon footprint associated with training large-scale models are substantial, raising concerns about the sustainability of these technologies.

2.4 A Framework for Integrating Knowledge Representation

For industry practitioners looking to implement LLMs in production engineering a guide outlining the key steps and considerations for deployment and effective use is presented next.

1. Understanding the application domain
    - Domain analysis: Before diving into LLMs, understand your industry's specific needs and challenges.
    - Model selection: Choose an LLM that aligns with your domain needs. Factors like model size, training data, and performance benchmarks are crucial.
2. Data preparation and fine-tuning
    - Data collection: Gather domain-specific data for fine-tuning. This data should represent the kind of language or problems the model will encounter in production.

- Structured knowledge sources: Introduce structured knowledge sources like databases, ontologies, or knowledge graphs during the data preparation phase. This structured information can provide context and domain-specific insights.
- Data annotation: Annotate training data with semantic tags or labels that correspond to the structured knowledge. This helps the LLM understand the relationships and hierarchies within the domain knowledge.
- Fine-tuning process: Use your collected data to fine-tune the LLM. This step customizes the model to better understand and respond to domain-specific queries or tasks.
- Consider LORA: Layerwise Relevance Analysis (LoRA) can be an effective way to adapt large models to specific tasks with minimal additional training data and compute resources.

3. Model architecture and training

- Incorporate external knowledge bases: During model training, consider integrating external knowledge bases. Techniques like entity linking can be used to connect text in the training data to entities in these knowledge bases.
- Hybrid models: Consider hybrid models that combine the generative capabilities of LLMs with the structured reasoning of knowledge-based systems.

4. Model quantization and optimization

- Quantization: Implement model quantization to reduce the model size without significantly sacrificing performance. This step is necessary for deploying large models in resource-constrained environments.
- Optimization techniques: Use techniques like pruning or knowledge distillation to make the model more efficient for operational use.
- Retain knowledge integrity: Ensure that the quantization and optimization process preserves the integrity of the incorporated knowledge. Test the model to ensure that the knowledge representation is still effective after optimization.

5. Deployment strategies

- Deployment platform: Choose a suitable platform for deployment. Cloud platforms offer scalability, but edge computing might be necessary for low-latency applications.
- API integration: Develop APIs to interface the LLM with existing systems and applications in your production environment.
- Knowledge-enriched interfaces: Design interfaces that can utilize the structured knowledge. For instance, a query system in production engineering could pull data from a knowledge graph to provide more accurate responses.

6. Risk management and compliance

- Ethical considerations: Address potential biases in the model and ensure compliance with industry regulations and ethical guidelines.

- Risk assessment: Conduct a thorough risk assessment, particularly focusing on the consequences of incorrect predictions or advice given by the model.

7. Operational use and monitoring

   - User training: Train your staff on how to interact with and leverage the LLM for optimal results.
   - Continuous monitoring: Set up systems for continuous monitoring of the model's performance and accuracy. Incorporate user feedback for ongoing improvements.
   - Monitoring knowledge utilization: Monitor how the LLM uses the integrated knowledge to ensure that it is being applied correctly and effectively in real-world scenarios.
   - Update and maintenance: Regularly update the model based on new data, user feedback, and evolving industry trends.

8. Evaluation and iteration

   - Performance metrics: Regularly evaluate the model against key performance metrics.
   - Iterative improvement: Use evaluations to inform iterative improvements of the model and its deployment strategy.
   - Feedback loop for knowledge refinement: Consider establishing feedback loop where the model's outputs are used to refine the knowledge base. This can be particularly valuable in identifying gaps or inaccuracies in the current knowledge representation.

9. Documentation and reporting

   - Maintain transparency: Keep detailed documentation of the model's capabilities, limitations, and any alterations made during fine-tuning and deployment.
   - Reporting: Establish reporting mechanisms for any issues or anomalies encountered in the operational use of the LLM.

10. Scalability and expansion

    - Scalability considerations: As your operations grow, ensure that the LLM can scale up accordingly, both in terms of handling increased load and adapting to new tasks.
    - Exploring new applications: Stay informed about advancements in LLMs and explore new applications within your industry.

3 Manufacturing Application of TLP

LLM is increasingly gaining traction across manufacturing. There are a vast range of use cases that have considered LLM across the life cycle in areas such as process efficiencies, product quality, and knowledge capture. The literature review that has been conducted in this section focuses on three key search terms: 'TLP and Manufacturing', 'LLM and Manufacturing' and 'NLP and manufacturing' and Science Direct was used as the platform to identify research contributions. The results of this review have been classified based on different manufacturing life cycle stages, considering core challenges, approaches, and impact of research. Further details are shared in the following sub-sections. Overall, LLMs are offering numerous benefits in terms of: (1) creating a conversational gateway between humans and machines, which could unlock productivity gains through symbiotic relationships, (2) new insights can be captured through LLM to tackle complex, and domain-specific needs, (3) new opportunities emerge to better manage databases that humans have not managed to at scale, and (4) significant gains to be made by offering ways to capture and transfer tacit knowledge (from e.g. retiring workforce).

3.1 Good Practice Application in Product Design

Challenges tackled in design: Numerous challenges have been noted in literature linked to product design processes. Mingdong Li et al. (2024) focus on the context of new product development. They highlight the process of generating new concepts is a critical step that affects the whole life cycle. However, they highlight challenges related to incomplete information, chaotic design constraints, and complicated design evolutions. They consider deep learning based methods relevant to deal with these challenges, but highlight gaps and inconsistencies in how design related documents (e.g. patents, user manuals) and causal spoken language (e.g. thinking process for concept generation) use terms. In contrast, Powell et al. [36] focus on the context of product customization in design. They highlight that consumer preferences can be challenging to elicit, which influences the time and effort to create novel products. They also promote the growing role that prediction models can have in replacing traditional means (e.g. elicitation, focus groups) of collecting consumer preferences, which will ultimately lead to speeding up the production cycle and saving cost. However, the core challenge they raise is that current prediction models require extensive data, and more efficient methods are needed to cope with this. Thus, they propose a clustering approach that can cope with different product types and features. Finally, Akay et al. [3] focus on the challenges related to contextual data and knowledge exchange across operational and functional units in a manufacturing enterprise. They highlight that in order to achieve smart manufacturing there is a need for precise control and agility under unexpected disruptions. The challenge that they focus on is how to obtain information from a decision maker at the point of decision making in

order to embed a systematic way to extraction of decision reasoning and functional information.

Methodologies followed for design: Numerous LLM based approaches have been proposed in the context of design. Mingdong Li et al. (2024) developed a process for key concepts derivation using cognitive analysis to effectively incorporate prior knowledge. They achieved this by developing a pretrained language model to extract information from multiple sources of textual data, which gets represented within a vector space. This involved a Design-by-Analogy based verbal protocol, which was tested in the context of high-speed elevators. Based on this, the paper shows strong results in the effectiveness for key concepts derivation in product design particularly considering specific customer- required functionality. In alignment, Powell et al. [36] developed a proof-of-concept consumer preference prediction methodology focusing on design for product customization. They develop a sample of 307 individual consumer preferences from a survey about thirty-seven training products and three validation products. The paper applied ChatGPT for user-generated clustering variables along with features. Further considerations were offered by Akay et al. [3], which proposed the use of LLM to address limitations in the design of smart manufacturing systems by building a semantically searchable and sharable knowledgebase. They apply deep learning-based (NLP) models such as Google's BERT in order to perform the task of Question-Answering. This involves extracting and structuring conceptual design reasoning information from textualized descriptions of existing designs. They highlight the need for past design and decision reasonings to be well described in textualized design documents to achieve high accuracy.

Impact observations for design: There are a range of observations that have been made about the impact that LLM can have on design. Mingdong Li et al. (2024) made significant contributions in terms of cognitive analysis-based key concepts derivation using a pretrained natural language model. The design by analogy based verbal protocol demonstrated that it is powerful in eliciting how different inspirations impact the design solutions. It was impactful that NLM was able to match embedded representations to create key concepts capturing the design thinking process. The paper demonstrated that LLM can offer reasonable assistance in product design. In contrast, the recommendation model developed by Powell et al. [36] for customization decisions, achieved an average accuracy of 70% across three different products. This is a promising result that could be further improved by considering larger training datasets, and further developments in ChatGPT. As an another example, by applying the concept of 'Design reading', Akay et al. [3], demonstrated promising results for extracting functional reasoning in an automated manner. In this process, they extracted and represented contextual design reasoning data and functional requirements from past design documents. They introduced the concept of the Push-Pull Digital Thread, which makes an impact in three ways: (1) extracting contextual knowledge and reasoning data from the decision maker for design, (2) representing functional aspects of decision reasoning through distributed feature vectors using statistical language models which can be semantically searched and pulled when necessary, and (3) enabling security using enterprise-wide intranet to share

functional representation of decisions enabling the exchange of data without leaving out any key stakeholders in the loop. The contribution of the approach is in keeping the know-hows and past decision reasonings within the enterprise in perpetuity.

3.2 Good Practice Application in Production Management

Challenges tackled in manufacturing: Across literature, a range of challenges have been outlined for manufacturing processes and how LLM can assist. Badini et al. [5] focus on the use of LLM in the context of additive manufacturing (AM). They specifically aim to contribute to the Gcode generation process, which has a critical role to control the movements of the printer's extruder, and the process of layer-by-layer building. They target to address challenges in how to optimize the Gcode with the ambition to ensure the quality of the final product is in place. They also aim to reduce the print time and waste. As for the proposed approach, the paper proposes the use of ChatGPT for the existing Gcode data to generate optimized Gcode for specific polymeric materials, printers, and objects. They also consider ways to analyse and optimise the Gcode based on various printing parameters such as printing temperature, printing speed, bed temperature, fan speed, wipe distance, extrusion multiplier, layer thickness, and material fiow. As an alternative perspective, Malburg et al. [31] focus on the need for intelligent manufacturing systems that can flexibly react to unexpected circumstances in order to minimise their impact. They consider the use of AI to move towards automated planning in flexible production processes. They recognise that such approaches require contextual knowledge, and there are challenges with the scale of effort required to create and maintain AI modules, and also they are error prone. They propose an approach for existing knowledge to be reused and transformed automatically into planning descriptions. In a similar vain, Xiao et al. [45] focus on the assembly process, and highlight challenges linked to inefficiencies due to high reliance on the knowledge of process related personnel, which often is not captured effectively. This causes delays in assembly process evaluation and the optimal assembly sequence cannot be determined easily. They propose a knowledge management and inference tool using Knowledge Graphs, which can handle these problems well.

Methodologies followed for manufacturing: There are numerous approaches that have been taken for manufacturing, which often have focused on a particular type of manufacturing process. Badini et al. [5] applies ChatGPT to improve the efficiency of the Gcode generation process in AM considering warping, bed detachment, and stringing. They structured the input into ChatGPT in an open-ended format in the following sequence: 1. Resolution of a specific high level 3D printing issue; 2. Resolution of the detailed 3D printing issue for alternative filament material (e.g., Thermoplastic polyurethane polymer); and 3. Goes into further detail considering also the specific boundary conditions (e.g. values of printing parameters, type of printer). In contrast, Malburg et al. [31] focus on addressing the challenge of reusing knowledge for dynamic planning by developing a converter that transforms existing

knowledge derived from literature. They used the SWS2PDDL converter in order to transform the knowledge into formal Planning Domain Definition Language (PDDL) descriptions. This involved building a semantic model consisting of the Semantic Web Services (SWSs). The corresponding domain ontology as well as the planning problem in the context of the smart factory are inputs of the converter and the formal PDDL domain, whilst the outputs are the problem descriptions. The planning domain, relies on the development of the domain ontology FTOnto and the developed SWSs, which are processed by the Domain Ontology Parser and the Semantic Web Service Parser respectively. In this process, they use the open-source framework Apache Jena10 for utilising the semantic model.

Impact observations for manufacturing: There are numerous research outputs that have demonstrated significant impact from applying LLM in manufacturing. Badini et al. [5] demonstrated how ChatGPT could assist in performing complex tasks related to AM process optimization. Their approach particularly specialised in evaluating printing parameters and bed detachment, warping, and stringing issues for Fused Filament Fabrication (FFF) methods using thermoplastic polyurethane polymer as feedstock materials. They highlight the importance of having a user-friendly interface for ChatGPT with the promise for AM process optimisation through the enhanced Gcode generation process and optimal printing parameters. It is observed that the real-time optimization capabilities of ChatGPT can enable significant time and material savings, which would ultimately make AM more accessible and cost-effective for manufacturers and industry more widely. Along these lines, Malburg et al. [31] showed the usefulness of the SWS2PDDL converter for dynamic planning using a near real-world application scenario in the context of reacting to failures in a physical smart factory and evaluating the generated re-planned production processes. They illustrated that the SWS2PDDL based automated approach creates updated plans that are comparable or even better than the manual modelling efforts.

3.3 Good Practice Application in Service Engineering

Challenges tackled in service engineering: Across service based literature, there are numerous examples highlighting the importance of human knowledge. May et al. [33] focus on limitations caused by data analysis being typically constrained to numerical data in the context of digitization of manufacturing. They particularly highlight this causes an impact on the synthesis of knowledge that could be accumulated from text based input from experienced employees. Accordingly, they propose NLP to leverage available text data from machine providers to realise better plans to reduce failure downtime. They also outline ways to formalise knowledge as a basis for optimizing manufacturing processes more widely. In a similar vain, Naqvi et al. [34] argue that there is a need to leverage untapped human knowledge in Maintenance Work Orders (MWOs). They highlight NLP as a suitable approach to handle complex challenges and enable wider adoption of digital twins to deliver maintenance. In their paper they propose a mechanism for human knowledge centred

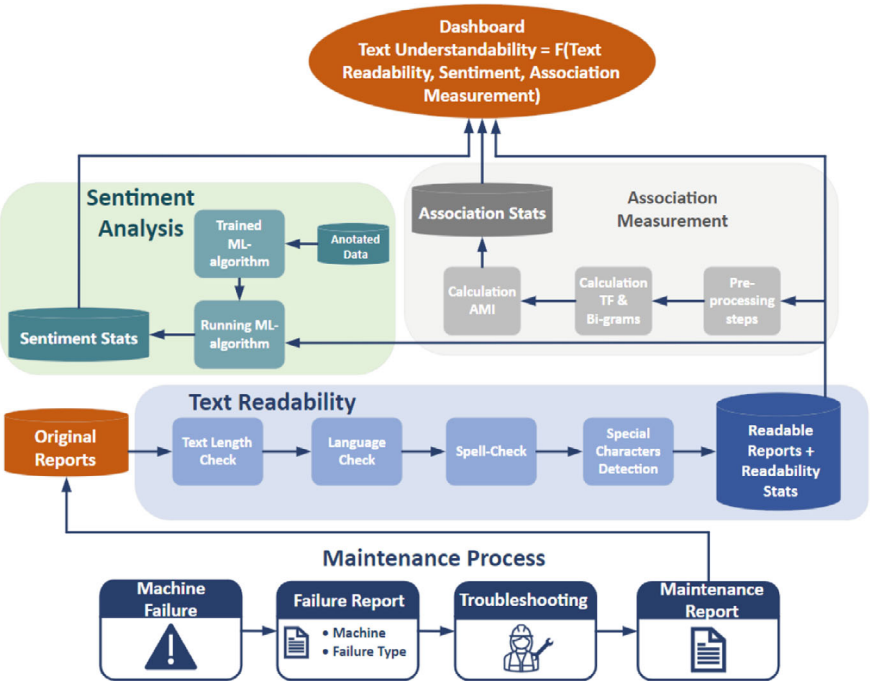


Fig. 6 A holistic methodology for integrating TLP in maintenance and service engineering [7, 30]

intelligent maintenance decision support. The approach consists of finding solutions to emerging maintenance challenges based on past maintenance records using a digital twin environment. These considerations are also shared by Brundage et al. [7], which make an important point about how NLP has to be developed with contextual knowledge. They also highlight that across the life cycle of asset management valuable information contained in documents are typically under utilised in analysis. They also pointed out that industrial implementations of NLP are often built on tools intended for non-technical use cases, suffering from a lack of verification validation, which lead to a lack of personnel trust. Lee et al. [27] focus on challenges related to formalising knowledge and specifications that simplify and automate the design and operation of safety equipment and investigated how the failure classification process can be made more efficient (Fig. 6).

Methodologies followed for service engineering: Brundage et al. [7] highlight that in the process of developing NLP based approaches, there is a need to apply a holistic, domain-driven methodology within a technical engineering setting. This is schematically sketched in the context of maintenance and could be transfered to the other application domains as well. The important point here is about how to achieve a holistic approach given multiple views and perspectives, and how to develop a level of normalisation across those data sources. Along these lines, May et al. [33] applied NLP on a case study linked to downtime prediction in manufacturing envi-

ronments. They used text based comments from machine operators in combination with alarm and events data from production lines to predict the downtime of machine stops. They also classified faults based on their severity with regard to experienced downtime. In contrast, Naqvi et al. [34] developed an architecture for NLP use for improved maintenance in digital twin environments using its connections with Physical Space, Virtual Space and Digital Twin Data are presented in this paper. They evaluate the performance of the service through a case study on an open-source dataset of real maintenance work orders from mining excavators. Lee et al. [27] developed a semi-automated process using TLP, which can incorporate classification rules for processing for failure classification. They make recommendations for how the work can be applied more widely in Industry 4.0 particularly in the context of digital representations to monitor the performance of safety instrumented systems.

Impact observations for service engineering: Brundage et al. [7] highlight that in order to make a significant impact using NLP in the industrial asset management world, there is a need to have consensus around key domain-specific resources, such as: (1) the way data representations are created to be goal-driven, (2) approaches to enable flexible entity type definition and dictionaries, and (3) achieving improvements to access data-sets–raw and annotated. To support these points, Naqvi et al. [34] demonstrated that state-of-the-art NLP techniques are able to process human knowledge in maintenance work orders and produce impactful predictions. They made a significant contribution towards the application of TLP in a smart manufacturing context. In order to make a wide impact in NLP, Lee et al. [27] highlights the need for groundwork where classification rules are generated from international standards and commonly agreed by industry. They considered NLP could play a role in the context of failure mode and detection method in order to decide if a failure is dangerous and undetected. They highlighted that in this context data is often provided as un-structured text in notifications captured in maintenance management system. They highlighted the need for further work in classifying failures in order to reduce the considerable manual effort in reading and analysing the texts. As an example application of NLP with a formalised way to structure data, May et al. [33], presented promising results. They applied numerous vectorisation methods such as Bag-of-Words for vectorization, Word2Vec, TF-IDF vectorization,in order to classify text based input about failure downtime prediction. From their comparisons TF-IDF vectorization achieved the highest F1 score at 58.87%, and demonstrated potential for further development. Here it is important to note the real time nature of predictions, and realising classifications in text.

4 TLP Compliance for Industrial Applications

4.1 Regulative Considerations for Development and Deployment of LLM-Based TLP System: Example of EU AI ACT

The European Union (EU) AI Act, introduced in April 2021, aims to establish conditions for developing and using trustworthy AI systems and innovative applied AI technologies. Since the scientific community lacks a universally accepted definition of AI, the EU AI Act establishes a legal definition of AI-System as [11]: *"…software that is developed with [specific] techniques and approaches and can, for a given set of human-defined objectives, generate outputs such as content, predictions, recommendations, or decisions influencing the environments they interact with"*.

This involves careful consideration of black box models, in particular deep learning methods and transformer models, i.e. large language models (LLMs) and LLM-based TLP. Thus, the EU AI Act (as a notable example and also similar guidelines in other markets worldwide) would directly affect the development, deployment, and use of TLP and all applications that leverage deep learning models, such as chatbots, text generation systems, and virtual agents. In this context, the regulation defines several key subjects. The ones that are most important to the training and use of LLMs in an industrial setting are [11]:

- "Harmonized rules for the placing on the market, the putting into service and the use of artificial intelligence systems ('AI systems') in the Union,
- prohibitions of certain artificial intelligence practices, and
- specific requirements for high-risk AI systems and obligations for operators of such systems".

While, currently, most pre-trained LLMs are trained, developed, and deployed outside of the EU, like GPT-3 [6], Llama-2 [41], and BERT [12], the EU AI Act addresses this issue by extending its scope to [11]:

- "Providers, placing on the market or putting into service AI systems in the Union, irrespective of whether those providers are established within the Union or in a third country, and
- Providers of AI systems that are located in a third country, where the output produced by the system is used in the Union".

The realization of the EU AI Act should achieve its objectives, such as ensuring the safety and compliance of AI systems in the EU market with existing EU laws providing legal certainty to encourage investment and innovation. Further, governance and enforcement of EU laws related to AI systems' fundamental rights and safety requirements are improved, and a unified market for trustworthy AI applications is developed [11].

To efficiently enforce regulations and achieve the goals outlined in the act, the act defines mandatory requirements applicable to the design and development of

AI systems based on the level of the system's risk. This risk-stratified framework classifies AI systems into four distinct tiers [11]:

- **Unacceptable risk**: This category prohibits harmful AI practices that threaten individuals' safety, livelihoods, and rights. AI systems falling under this classification cannot be placed on the EU market, deploy their services, or be used within the EU.
- **High risk**: This tier includes AI systems with considerable impact on individuals' safety or fundamental rights. It includes explicitly systems designated as safety components within products and high-risk AI systems operating in eight specified areas, such as biometric identification and categorization of natural persons, management and operation of critical infrastructure, and law enforcement.
- **Limited risk**: AI systems within this category present modest risks, such as interacting with humans (e.g., chatbots), emotion recognition systems, biometric categorization systems, and systems generating or manipulating multimedia content (e.g., deepfakes). These systems are subject to a restricted set of transparency requirements.
- **Low or minimal risk**: All other AI systems characterized by low or minimal risks are permitted for development and use within the EU without additional legal constraints.

In this context, LLM-based TLP models and systems are associated with the limited risk category. These models generate and manipulate text exceptionally, closely resembling human-like output. Moreover, they engage with humans through integration into various frameworks such as chatbots, question-answering systems, and virtual assistants. Consequently, they are subject to specific transparency obligations [11]:

- **Informing natural persons**: Providers are required to ensure that AI systems designed for interaction with individuals are developed in a way that informs users of their engagement with an AI system unless this is evident from the context of use.
- **Disclosure of artificial content**: Users of AI systems generating or manipulating multimedia content resembling existing persons or events to a degree that could deceive users (e.g., deep fakes) must disclose that the content has been artificially generated or manipulated.

However, considering the evolving nature of the LLM field and its uncertain trajectory, the EU has additionally introduced amendments to the EU AI Act, specifically addressing foundation models [11]. Thus, comparable to high-risk systems, LLMs must comply with similar requirements in addition to lower category requirements [11]:

- **Database registration**: Providers must register LLMs in a centralized EU database before market placement or service deployment.

- **Comprehensive criteria**: LLMs must fulfill various criteria encompassing risk management, technical robustness, data governance, human oversight, and cybersecurity.

Further, LLMs must follow the transparency requirements such as [11]:

- **Content source disclosure**: Indicating that AI has generated the content.
- **Prevention of illegal content**: Ensuring the model is designed to prevent generating illegal or prohibited content.
- **Publication of training data summaries**: Providing summaries of copyrighted data used during the model's training process.

Given these comprehensive and stringent regulations on the horizon, establishing an LLM assessment process is necessary. It ensures that LLMs, developed and deployed across different industrial sectors, follow existing and future regulations without restricting the adoption of novel technologies to stay competitive.

4.2 Universal Compliance Assessment Process (UCAP)

Considering global regulatory frameworks for AI systems, and in particular the EU AI ACT, this section proposes a universal compliance assessment process (UCAP) for LLMs, in particular LLM-based TLP models and systems. The UCAP should act as a safeguard, systematically evaluating such models for legal adherence, transparency, and accountability. By addressing these aspects comprehensively, such a process guarantees that LLMs are deployed responsibly, fostering trust among users and companies. The universal compliance assessment approach is tailored explicitly for LLMs within the EU AI Act. The approach centers on the technical dimensions of compliance, emphasizing adherence to guidelines, standards, and protocols established for developing and deploying LLMs. While ethical and social considerations are crucial, they require separate attention beyond the UCAP.

The UCAP approach unfolds across three distinct phases, ensuring transparency and precision throughout the entire development cycle of a custom LLM for industrial applications, namely: (i) Pre-development phase, (ii) Development phase including pre-training and fine-tuning, and (iii) deployment. The entire process is visualized in Fig. 7. The UCAP is explained using two distinct swim lanes, i.e. (i) the business, and (ii) development units. This organizational structure is important to address and incorporate into the assessment process, considering that the development of an LLM-based application may involve outsourcing to external companies or a dedicated software or IT division functioning as a separate business unit within a larger corporate framework. Generally, assessment tasks included in the pre-development phase are shared between both units. However, responsibilities for tasks from the development phase lie exclusively within the scope of the development unit. Finally,

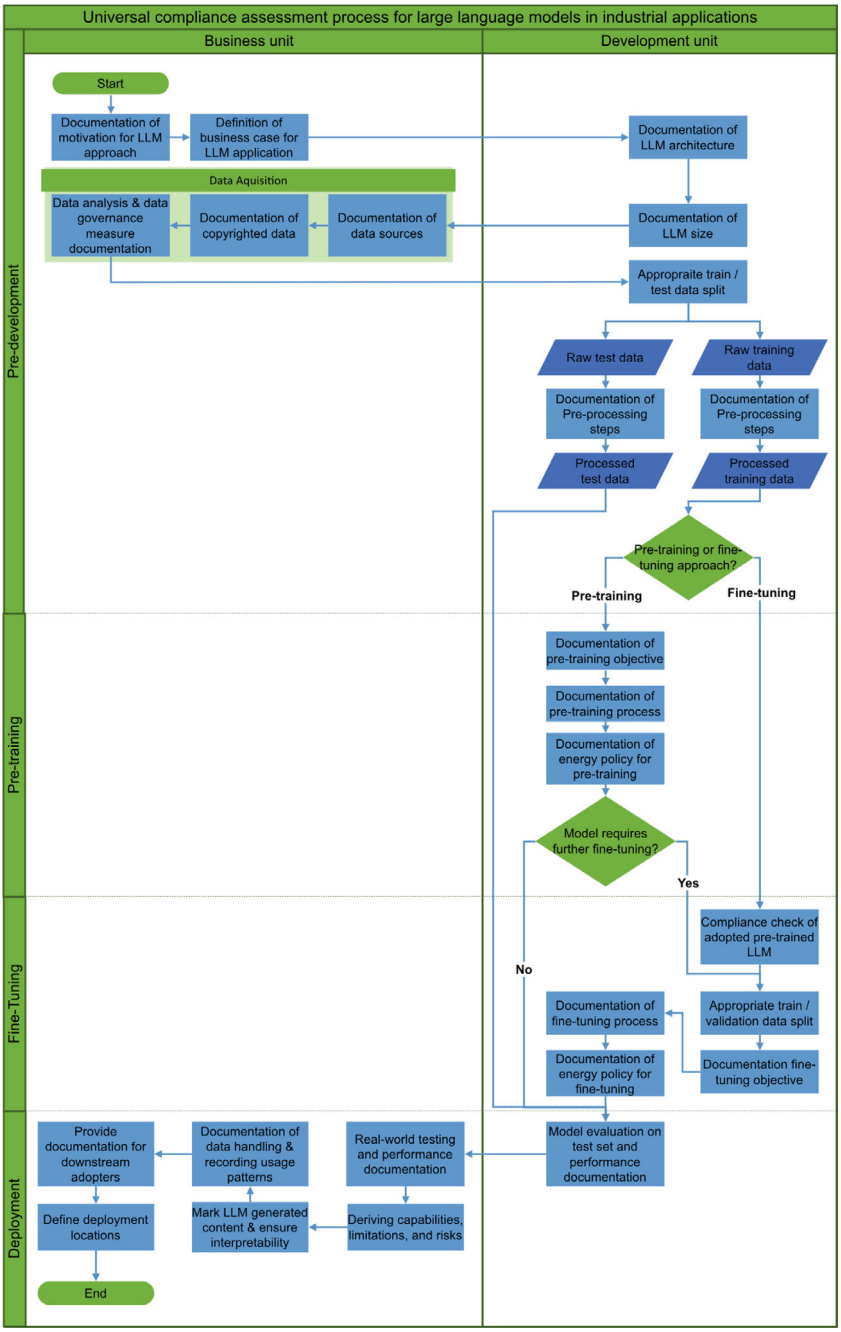


Fig. 7 UCAP for the development of domain-specific LLMs for industrial applications

the deployment phase is shared between both units, comprising activities such as field testing, risk assessments, downstream documentation, and providing information regarding the deployment location.

5 Conclusion

This essay aims at deepening the insights into the emerging filed of TLP by elaborating on (i) basic terminologies and concepts, (ii) definition of TLP and its coherent relation to NLP, (iii) pre-processing tasks, (iv) opportunities and challenges for implementation of NLP/TLP methods in industrial applications, (v) TLP in the era of generative AI and LLMs covering neural networks and their architectures, and later by discussing (vi) manufacturing application of TLP across product life-cycle, including product design, production management and service engineering, as well as (vii) regulative consideration for LLM-based TLP system, and related compliance assessment processes.

The paper, therefore, addresses both scientists and industrial experts and provides in-depth explanation of the emerging field of TLP especially in manufacturing applications. Further, it lays the ground for future research on methodical and methodological enhancements of TLP and its integration, as a tool, into manufacturing applications. The paper also encourages further investigation on (i) KPIs for assessing efficiency, effectiveness and accountability of TLP in manufacturing, and (ii) on data privacy and ethics in particular dealing with black-box models like neural networks to establish transparent and explainable TLP systems of the future.

References

1. Aggarwal CC (2015) Mining text data. Springer
2. Aggarwal CC (2018) Machine learning for text: An introduction. Springer
3. Akay H, Sang HL, Sang-Gook K (2023) Push-pull digital thread for digital transformation of manufacturing systems, CIRP Annals
4. Ansari F, Kohl L, Giner J, Meier H (2021) Text mining for ai enhanced failure detection and availability optimization in production systems. CIRP Ann 70(1):373–376
5. Silvia B, Stefano R, Emanuele F, Raffaele P (2023) Assessing the capabilities of chatgpt to improve additive manufacturing troubleshooting. Advanced Industrial and Engineering Polymer Research
6. Tom B, Benjamin M, Nick R, Melanie S, Kaplan JD, Prafulla D, Arvind N, Pranav S, Girish S, Amanda A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
7. Brundage MP, Sexton T, Hodkiewicz M, Dima A, Lukens S (2021) Unlocking maintenance knowledge. Technical language processing Manuf Lett 27:42–46
8. Camacho-Collados J, Pilehvar MT (2017) On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *arXiv preprint* arXiv:1707.01780

9. Nilesh C, Denis L, Gaurav M, Priyansh T, Jens L, Asja F (2019) Introduction to neural network based approaches for question answering over knowledge graphs. *arXiv preprint* arXiv:1907.09361
10. Aakanksha C, Sharan N, Jacob D, Maarten B, Gaurav M, Adam R, Paul B, Hyung Won C, Charles S, Sebastian G, Parker S, Kensen S, Sasha T, Joshua M, Abhishek R, Parker B, Yi Tay, Noam S, Vinodkumar P, Emily R, Nan D, Ben H, Reiner P, James B, Jacob A, Michael I, Guy G-A, Pengcheng Y, Toju D, Anselm L, Sanjay G, Sunipa D, Henryk M, Xavier G, Vedant M, Kevin R, Liam F, Denny Z, Daphne I, David L, Hyeontaek L, Barret Z, Alexander S, Ryan S, David D, Shivani A, Mark O, Dai AM, Pillai TS, Marie , Aitor L, Erica M, Rewon C, Oleksandr P, Katherine L, Zongwei Z, Xuezhi W, Brennan S, Mark D, Orhan F, Michele C, Jason W, Meier-Hellstern K, Douglas E, Jeff D, Slav P, Noah F (2022) Palm: Scaling language modeling with pathways
11. European Commission. Briefing on eu ai act
12. Jacob D, Ming-Wei C, Kenton L, Kristina T (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805
13. Alden D, Sarah L, Melinda H, Thurston S, Brundage MP (2021) Adapting natural language processing for technical text. Appl AI Lett 2(3):e33
14. Hugging F. What is questions answering?
15. Flare. Difference between natural & technical language processing
16. Goodfellow I, Pouget-Abadie J, Mirza M, Bing X, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
17. Stanford NLP Group. Stanford tokensregex
18. Masato H (2021) Real-world natural language processing: practical applications with deep learning. Simon and Schuster
19. Harshvardhan GM, Gourisaria MK, Pandey M, Rautaray SS (2020) A comprehensive survey and analysis of generative models in machine learning. Comput Sci Rev 38:100285
20. Hock-Chuan C. Regular expression (regex) tutorial
21. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, de las Casas D, Bressand F, Lengyel G, Lample G, Saulnier L, Lavaud LR, Lachaux MA, Stock P, Scao TL, Lavril T, Wang T, Lacroix T, El Sayed W (2023) Mistral 7b
22. Taeho J (2019) Text mining, vol 45. Springer
23. Khurana D, Koli A, Khatter K, Singh S (2023) Natural language processing: State of the art, current trends and challenges. Multimedia Tools Appl 82(3):3713–3744
24. Kingma DP, Welling M (2013) Auto-encoding variational bayes. *arXiv preprint* arXiv:1312.6114
25. Ekaterina K (2022) Getting started with natural language processing. Simon and Schuster
26. LeCun Y, Haffner P, Bottou L, Bengio Y (1999) Object Recognition with Gradient-Based Learning. Springer, Berlin Heidelberg, pp 319–345
27. Shenae L, Maria Vatshaug O, Stein H, Mary Ann L (2023) Towards standardized reporting and failure classification of safety equipment: Semi-automated classification of failure data for safety equipment in the operating phase. Process Safety Environ Protect 177:1485–1493
28. LMSYS O (2024) Lmsys arena leaderboard. https://chat.lmsys.org/?leaderboard. Accessed: August 16 2024
29. Lowin M (2024) A text-based predictive maintenance approach for facility management requests utilizing association rule mining and large language models. Mach Learn Knowl Extraction 6(1):233–258
30. Theresa M, Linus K, Fazel A (2021) A text understandability approach for improving reliability-centered maintenance in manufacturing enterprises. In: Dolgui A, Bernard A, Lemoine D, von Cieminski G, Romero D (eds) Advances in Production Management Systems. Springer International Publishing, Artificial Intelligence for Sustainable and Resilient Production Systems. Cham, pp 161–170
31. Lukas M, Patrick K, Ralph B (2023) Converting semantic web services into formal planning domain descriptions to enable manufacturing process planning and scheduling in industry 4.0. Eng Appl Artif Intell 126:106727

32. Mandvikar S (2023) Augmenting intelligent document processing (idp) workflows with contemporary large language models (llms). Int J Comput Trends Technol 71(10):80–91
33. May MC, Neidhöfer J, Körner T, Schäfer L, Lanza G (2022) Applying natural language processing in manufacturing. Procedia CIRP 115:184–189
34. Naqvi SMR, Ghufran M, Meraghni S, Varnier C, Nicod J-M, Zerhouni N (2022) Human knowledge centered maintenance decision support in digital twin environment. J Manuf Syst 65:528–537
35. Penedo G, Malartic Q, Hesslow D, Cojocaru R, Cappelli A, Alobeidli H, Pannier B, Almazrouei E (2023) and Julien Launay. Outperforming curated corpora with web data, and web data only, The refinedweb dataset for falcon llm
36. Powell C, Zhu E, Xiong Y, Yang S (2024) A data-driven approach to predicting consumer preferences for product customization. Adv Eng Inform 59:102321
37. Stephan R (2022) Deep Learning for Natural Language Processing. Simon and Schuster
38. Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408
39. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
40. Gemini Team (2023) Gemini: A family of highly capable multimodal models
41. Hugo T, Louis M, Kevin S, Peter A, Amjad A, Yasmine B, Nikolay B, Soumya B, Prajjwal B, Shruti B, et al (2023) Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint* arXiv:2307.09288
42. Vajjala S, Majumder B, Gupta A (2020) and Harshit Surana. A comprehensive guide to building real-world NLP systems. O'Reilly Media, Practical natural language processing
43. Ashish V, Noam S, Niki P, Jakob U, Llion J, Aidan NG, Ł ukasz K, Illia P, (2017) Attention is all you need. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in Neural Information Processing Systems, vol 30. Curran Associates Inc
44. Weiss SM, Indurkhya N, Zhang T (2015) Fundamentals of predictive text mining. Springer
45. Xiao Y, Zheng S, Shi J, Xiaodong D, Hong J (2023) Knowledge graph-based manufacturing process planning: A state-of-the-art review. J Manuf Syst 70:417–435

STC F—Forming

Towards "First-Time-Right" Production in Metal Forming Processes

Enrico Simonetto, Evripides G. Loukaides, Till Clausmeyer, Jos Havinga, and Andrea Ghiotti

Abstract The increasing demand for customized and sustainable products is driving the manufacturing industry toward the adoption of the 'First-Time-Right' principle. This approach focuses on maximizing process efficiency, minimizing waste, and reducing operational costs. To achieve these objectives, the implementations of in-line control techniques and real time adaptive processes is crucial. These methods aim to minimize the reliance on trial and error approaches and compensate for dispersion of processes and material parameters. However, the systematic and comprehensive integration of all these systems still encounters resistance in traditional metal forming systems, where machines and equipment have often been designed and optimized over time without full consideration of these aspects. The purpose of this paper is therefore to analyse the advancements in the field, illustrating the main advantages and limitations of integrating control systems into metal forming processes, and to explore how these innovations can drive the industry towards achieving 'First-Time-Right' production in metal forming processes.

Keywords Control · Metal forming · System architecture

E. Simonetto (✉) · A. Ghiotti
Department of Industrial Engineering, University of Padova, Padua, Italy
e-mail: enrico.simonetto.1@unipd.it

A. Ghiotti
e-mail: andrea.ghiotti@unipd.it

E. G. Loukaides
Department of Mechanical Engineering, University of Bath, Bath, UK
e-mail: E.Loukaides@bath.ac.uk

T. Clausmeyer
Institute for Machine Tools and Production Processes, Chemnitz University of Technology, Chemnitz, Germany
e-mail: till.clausmeyer@mb.tu-chemnitz.de

J. Havinga
Faculty of Engineering Technology, University of Twente, Enschede, The Netherlands
e-mail: jos.havinga@utwente.nl

T. Tolio (ed.), *CIRP Novel Topics in Production Engineering: Volume 2*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-3-032-04439-6_2

1 Introduction

Metal forming operations are part of a family of capital-intensive manufacturing processes integrated into globally distributed supply chains, characterized by a high level of technology. Given the complex economic, corporate, and technological structures within which companies in this sector operate, innovations typically advance incrementally and are driven by the necessity to enhance efficiency or broaden the range of producible components [1]. Nevertheless, even within this context, certain innovation trends continue to play a pivotal role in shaping the technological evolution of metal forming processes. Over the past 70 years, their evolution, spurred by market expansion and driven by increasingly sophisticated consumer demands, has transformed the way many components are manufactured [2]. This transition has enabled the shift from the paradigm of mass production to that of mass customization [3]. Figure 1 [4] illustrates the movement from high-volume production of standardized parts with high efficiency to the manufacturing of a greater variety of products in smaller batches. This transition made possible by the third, fourth, and fifth industrial revolutions, manages increased complexity without sacrificing efficiency. This trend, which continually emphasizes the enhancement of efficiency and flexibility to increase each company's revenue and market share, is now facing additional pressures. These stem from, on one hand, the global reorganization of supply chains post-pandemic [5] and, on the other, the demands of the green transition [6].

An example can be provided by the transition in the transportation market, where the introduction of electric vehicles alongside internal combustion engine vehicles leads to an increase in the number of models produced, along with a redistribution of market shares [7]. In this context, an additional technological step is required in metal forming processes to increase operational flexibility while maintaining high efficiency and reducing waste. However, metal forming processes are characterized by a cost structure, as illustrated in Fig. 2, where the cost per part is significantly influenced by batch sizes [8]. Tool investments are typically substantial, and setup

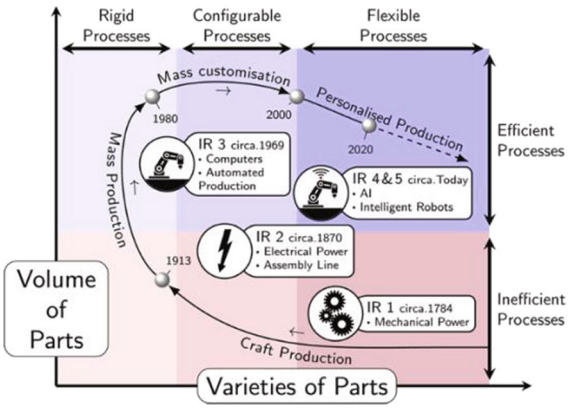


Fig. 1 Chronological demand for volumes and varieties of parts [4]

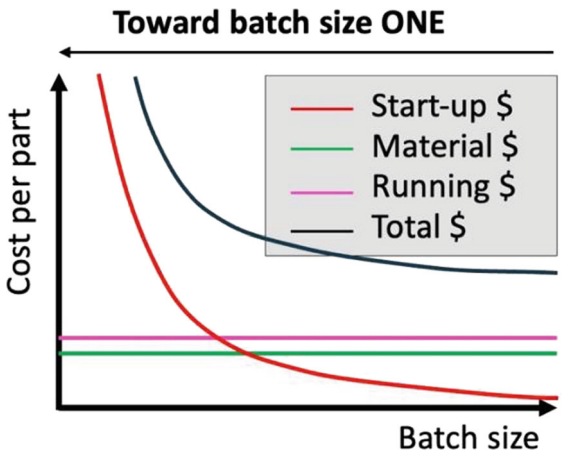


Fig. 2 Typical cost structure for a metal forming process

costs need to be partially repeated each time a new batch enters production. These costs include not only die change operations but also the production ramp-up phase to tune the process parameters to achieve the target quality, which is usually both material and time-consuming [9].

One way to address this cost structure is to reduce the tools cost by using more flexible processes, e.g. the incremental processes for sheet metal forming [10]. These solutions use non-dedicated tools with multiple degrees of freedom moved along complex paths to form parts, rather than using complex shape dies. However, these approaches have lower productivity, require longer times to adjust process parameters and involve changes in the deformation mechanism if compared with traditional processes.

A second way is to reduce setup costs. These can be addressed by using predictive models to enhance the tools and parameter design [11], and by implementing control systems for quickly tuning the process parameters [12]. The success of the first approach hinges primarily on the reliability of the models and the availability of input data, which may not fully accommodate the variability present in the industrial environment. On the other hand, the second approach requires the integration of actuators, sensors, and control algorithms into the forming line, which often encounters various practical obstacles. Among these, key issues include difficulties in directly measuring variables due to the lack of appropriate sensors, space constraints for sensor integration, and the cost–benefit imbalance of such systems. Consequently, variables are often estimated indirectly using other measurements and estimating models, introducing uncertainties and disturbances that can significantly impact the performances. It's important to note that these approaches are complementary rather than competitive, as each one addresses the shortcomings of the other. However, if successfully applied, these methods may lead to an overall improvement in the quality of the formed components and the efficiency of the process chain, resulting in cost reduction while increasing product value. At the same time, it's essential to be aware that the target parameters for metal-formed parts often account not only

for geometric aspects, but also other functional characteristics, such as mechanical, microstructural, and surface properties, among others [2].

In recent years, the field of metal forming processes has witnessed remarkable advancements enhancing the potential for integrating sophisticated control systems into metal forming processes. The impact of these advancements spans various areas including sensors, actuators, and control systems.

A key area of progress is in the realm of sensors [13]. The latest generation of sensors showcases notable improvements in several aspects, such as a reduction in their size and energy requirements, making them more efficient and easier to integrate into production systems [14]. Additionally, their overall measurement performance have been a significant enhancement not only thanks to hardware improvements but also thanks to software improvements for error compensation and disturbance reduction [15].

In the case of actuators, a significant trend is the shift towards electrification [16], with electric actuators increasingly replacing traditional pneumatic and hydraulic ones in many applications. This trend is driven by the improved performance and the wide range of types of electric actuators available [17], as well as the possibility to easily integrate them with sensors and feedback systems, to enable high dynamic performance and broad possibilities for control and monitoring.

The market for controllers has observed the advent of new, economically viable players coupled with an escalation in computational capabilities, facilitating real-time solutions. Concurrently, the latest developments in AI algorithms [18] and in Digital Twins [19] to achieve the real-time link between the sensor feedback and numerical models appear promising. These advancements are auspicious for incorporating process models into control systems tasked with processing substantial data volumes within timeframes congruent with process operations. However, the systematic and comprehensive integration of all these systems still encounters resistance in traditional metal forming systems. Machines and equipment have often been designed and optimized over time without full consideration of these aspects. The purpose of this essay is, therefore, to analyse the advancements in the field and to illustrate the main advantages and limitations of integrating control systems into metal forming processes.

2 Scope

This essay seeks to address the question: can the competitiveness of metal forming processes be enhanced by adjusting process settings based on measurement data? The answer is to be sought within the scope of traditional, industrially accepted metal forming processes, which are the result of decades of continuous improvement. A revamp of existing forming lines will therefore not be suggested, but it will be identified and discussed what modifications and features are required to adopt advanced metal forming control technologies, given the currently available sensor and actuation systems. Consequently, the primary limitation imposed in this essay is

that only the traditional concept of (cold) deformation through contact between the product and the tool is considered. The development of alternative methods, such as heat-assisted forming, is not considered here, even though these approaches may also have potential to enhance product accuracy. Secondly, this essay focuses mainly on product dimensions as the target properties of interest, although it is common that other properties, such as local hardness, microstructural properties or surface roughness may be specified as well.

As discussed in the review paper on metal forming control by Allwood et al. [12], the need to meet customer specifications is the main driver for the development of metal forming control systems. Different business cases for implementing metal forming control can be formulated by considering required production tolerances and allowable process disturbances: firstly, the already discussed trend towards mass customization drives the development of flexible forming systems such as incremental sheet forming [10]. Such systems deliver lower geometric accuracy, and closed-loop control systems are required to get them to par with conventional forming processes. Besides, solid business cases can be made for traditional forming processes as well. For example, reducing the scrap rate whenever a forming process fails to consistently work within set customer specifications. Or reducing the number of test products in the try-out phase of a new process. Otherwise, it may be effective to apply metal forming control for deliberately tightening specifications, to either omit additional finishing steps or to reduce the amount of material removal during finishing of forged products. Another reason to apply metal forming control is to avoid being dependent on a single material supplier, but to ensure accurate production while using materials from different suppliers, despite the larger spread in material properties.

From the aforementioned business cases, several objectives for metal forming control systems can be derived. Firstly, the objective may be to use a single tooling set for producing different products, either in flexible forming or in conventional forming, for example for sheet bending with different sheet thicknesses and materials. Secondly, it may be the objective to rapidly find the process settings for a new product and to reduce or eliminate the try-out phase. This target of First-Time-Right production is especially relevant for small batch production. Thirdly, metal forming control can be employed to reduce variation in product properties, especially for larger production runs. These variations are caused by changes in process conditions, such as tool heating, differences in material and friction properties, or tool wear. Depending on the cause, the variations manifest in different ways: the effects of tool heating are noticeable during the ramp-up phase and then stabilize, whereas variations in material batches introduce disturbances that are typically step-like. In contrast, tool wear can result in a gradual drift in process parameters over the tool's lifespan, potentially leading to more abrupt changes in the event of tool failure.

While this concept of compensating for external disturbances is the closest to what is conventionally understood by process control, it must be noted that many studies on metal forming control are focused on the first two objectives, which can be rephrased as compensating for lack of knowledge on the transfer function between process settings and product properties. Knowledge of this transfer function is crucial for improving process stability.

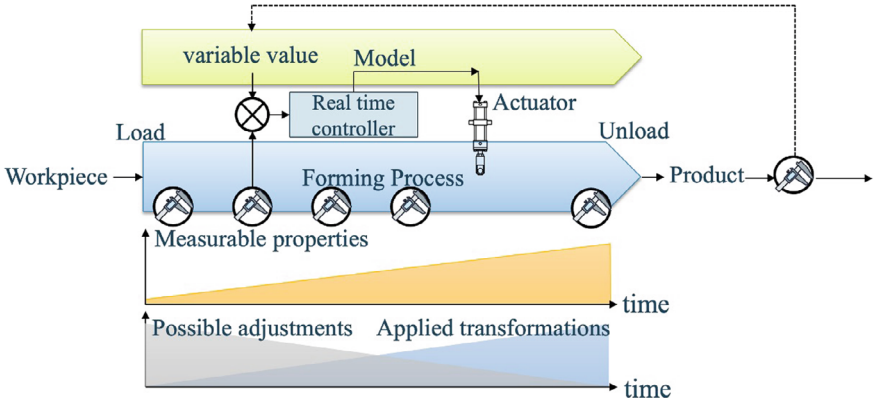


Fig. 3 Schematic of controllability window during forming of a single product

In this essay, a distinction is made between (i) single step forming processes where limited control actions can be applied during forming, and (ii) processes that are continuous or have multiple forming steps. In the former case, process settings must be defined prior to the start of the deformation processes, leaving little room for in-line adjustments. In the latter case, control actions can be determined during the deformation process. Figure 3 shows how during a metal forming process, the properties of a component are transformed (applied transformations, Fig. 3), leading to an increase in measurable properties as time progresses, while simultaneously the window for possible adjustments actions narrow.

The question of whether set production targets can be achieved through process control depends on a complex interplay. This includes product specifications, process deviations (whether uncertainties in the transfer function from process settings to product properties, or temporal/spatial variations in process conditions), and the bandwidth of sensor and actuation systems. In other words: can deviations from the target properties be identified through measurements and corrected through subsequent control actions? Furthermore, are sufficiently fast and accurate models available that define the relations between measurements, control actions, and product properties within the control system?

Figure 4, which shows a sequence of the calculated distortion values during the production of 1000 inner car door sheets [20], is illustrative when considering process variations during mass production. It is relevant to distinguish between product-to-product variations and long-term variations [21]. While product-to-product variations are dominant in the given automotive stamping example, also gradual drifts in distortion values can be observed. Such drifts can be considered as long-term variations, which require gradual adjustment of process settings. Measurements on product properties of finished products can be used in such a control system. On the other hand, when dealing with product-to-product variations, it is crucial to define control actions using measurements from the current product. A final notable type of process variation is the sudden change of product properties at the start of each

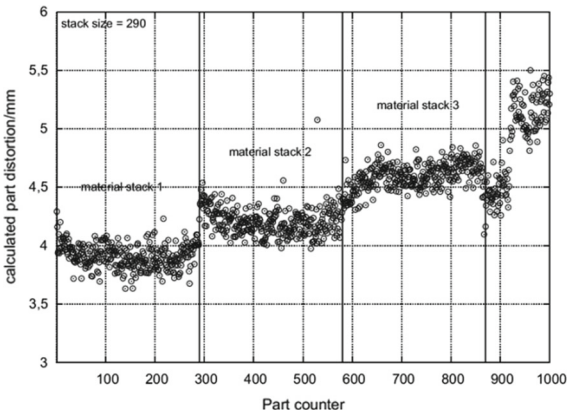


Fig. 4 Calculated part distortion during the production of inner door sheets [20]

new batch of material. Such jumps in product properties can for example occur at production restart after a standstill, or when changing to a new batch of material, as seen in the example.

Given the playing field defined by process, disturbances, sensors and actuators, a key ingredient for an effective control system is the ability to make sufficiently accurate predictions on the expected product properties, using the available measurements and process models. In that sense, the availability of sufficiently fast and accurate models plays a pivotal role in the development of effective metal forming control systems. In the light of aforementioned control objectives and disturbance types, it must be noted that various types of models may be needed: models that use process measurements during the production of a single product to estimate the final product properties and models that predict the effects of future control actions on the final product properties.

In the next section of this essay, the components of metal forming control systems will be discussed in detail: the control architecture, process disturbances, sensors, actuators and models. Together, these can be employed for closed-loop control of various forming processes, which will be extended upon in Sect. 4, using many examples from metal forming literature.

3 Controls Architecture

A review of the literature highlights that various types of controls have long been used in metal forming, applied both to a single process or to entire process chains. The approaches found in the literature range from open-loop systems to closed-loop systems performed both off-line, based on feedback from the quality control steps, and in-line based on feedback from data measured by sensors during the deformation process [12]. Despite this variety of approaches, a general hierarchy [22] of metal forming control systems can be specified, as depicted in Fig. 5. The

core is the machine, with a set of tools that come in contact with the workpiece, causing deformation. In the figure, two machine stages are considered, but they can be reduced to a single stage or expanded to multiple stages. Or, in the case of a continuous forming process such as incremental sheet forming, each tool pass or time period can be regarded as a single stage, and the figure can be interpreted in an equivalent way.

Forming process parameters (indicated with vertical double-lined arrows) are defined in terms of required loads or tool positions, and a machine control system must operate the actuators according to these settings. Often, an additional model is required to convert control parameters into signals for actuator management. For instance, achieving a specific position through hydraulic or electrical actuation requires a unique signal. Then the machine model plays a role in compensating

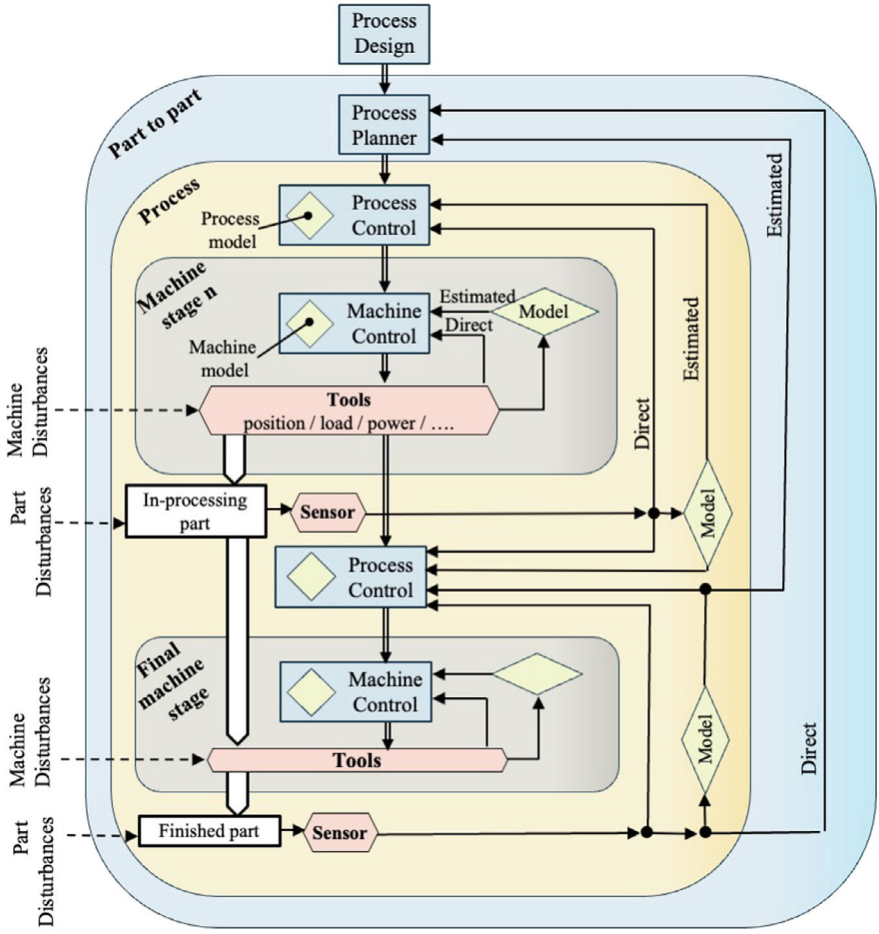


Fig. 5 Control system hierarchy for a metal forming control system

for disturbances, using feedback from sensors within or external to the actuators. Machine control systems can use position sensor signals to correct errors caused, for example, by deformations and mechanical play. The extent of these corrections is determined by the machine model and a closed-loop machine control system.

The machine control system is used to ensure that the prescribed parameters, such as tool positions and loads, are properly applied. The process parameters for a specific product are initially defined in the process design phase, based on simulations or experience. It is important to understand that not all designed parameters directly translate to process variables. For example, in sheet metal stamping, while component thickness is a key parameter, it is not directly controllable by an actuator, but results from the interaction of variables like the blank holder force, friction conditions and material properties. However, some design parameters, like the thickness of a rolled product, can be directly controlled by adjusting the roll gap.

Whenever a production process fails to produce parts within specifications, process settings must be adjusted. The traditional form of control is captured in the top-left cell in Table 1: offline inspection of finished products is used by an operator to manually adjust process settings, based on experience and process knowledge. An alternative form of process parameter adjustment using offline measurements is through prior inspection of the initial blank or billet, for example through non-destructive electromagnetic sensing or measurement of the blank thickness. This is illustrated in the top-right cell of Table 1 and has the potential to be automated and executed inline.

When employing an automated closed-loop control system instead, inline measurements are used to adjust actuator settings. A key factor for categorizing metal forming control systems is whether the control actions are based on measurements of previous products (left column in Table 1) or on previous forming stages of the current product (right column in Table 1). That is also visualized in Fig. 5, where sensor measurements are either used for control of the current stage for the following product, or for control of the following stage for the current product. Following the distinction between long-term and product-to-product variations discussed in Sect. 2,

Table 1 Overview of control approaches as function of measurement type

		Measurement source	
		Previous product	Current product
Measurement type	Offline	**Example** Offline quality control **Action** Operator adjustment	**Example** Material properties with electromagnetic sensing **Action** Feedforward 'control'
	Inline	**Example** Final geometry camera system **Action** Feedback control	**Example** Process forces or intermediate geometry **Action** Feedforward control

feedback control with measurements from previous products can be used to compensate for long-term variations. In contrast, measurements from the current product are required to control product-to-product variations through feedforward control.

In general, the unloaded product geometry is the primary target property. Due to springback after forming, it is usually only possible to measure the achieved geometry after finishing the forming process. This means that other measurements must be used if the aim is to control the final geometry of the current product. Measurements for example process forces or intermediate geometry can potentially be used to estimate the final product properties. Consequently, a model is required to relate the measured quantities with the predicted final properties, as shown in Fig. 5. For example, in sheet metal stamping, final sheet thickness may be a target property. However, while a direct global measurement of thickness is not possible, it can be estimated at specific points by measuring the part draw in.

If an actual or estimated product property strays from its target value, the control system must adjust the actuator input to correct it. For that purpose, a process model is vital for calculating these adjustments, mindful that corrective actions may alter other component parameters. For instance, in stamping, changing the blank holder force affects thickness, springback, and wrinkle formation. Despite the ideal scenario where the process model accounts for all effects, its capabilities are often limited due to computational constraints, modelling challenges, and uncertainties in boundary conditions.

An alternative control architecture to address the inability to measure the target properties of the current product involves defining target values for measurable quantities based on data from good reference parts. For example, in a deep drawing operation the punch displacement can serve as a reference for a closed-loop control system to control the sheet draw-in with adjustable blank holder forces [23]. This approach significantly improves production accuracy, even with disturbances like friction variations.

To enhance competitiveness in metal deformation processes, especially for smaller batches, it is crucial to implement advanced control systems that streamline setup, reduce waste, and improve quality. Polyblank et al. [24] discuss key concepts for evaluating these systems: (i) observability the system's ability to detect relevant changes; (ii) controllability indicates whether target properties are within the subset of achievable product states within a finite period, given limitations in actuation or tooling; (iii) model uncertainty reflects the limited accuracy of process models, and its proper quantification can enhance control performance. The following paragraphs will address each component, followed by examples of metal forming control systems.

3.1 Process Disturbances

Part of the metal forming control studies are concerned with the design of control systems that compensate for process disturbances such as variations in material properties, lubrication properties, tool heating or tool wear. Even though the performance of such a control system is dependent on the severity of process disturbances, it is not common to quantify or study process disturbances in metal forming. This is due to several factors, including the difficulty in identifying the main disturbances and quantifying their range of application, as well as the economic cost that physical tests may entail. For example, while it may be possible to vary the amount or type of lubricant used in a forming process, it is challenging to quantify the effects of such variations on the coefficient of friction. Instead, many approaches are validated with numerical studies, by studying the system performance when subjected to fictive disturbances, such as sudden jumps in material property values or friction coefficient.

A few studies together draw a picture of process disturbances to be expected in the context of metal forming control. In a short review from 2003, the origins of scatter in sheet metal stamping were presented, listing sources of variation such as material (uneven material properties), tooling (clearance, wear, roughness), process (blank holder force variation, stiffness of the press), lubrication (difficult to keep constant) and other unpredictable variations [25]. In the context of robust optimization, extensive studies have been performed on the variation of material properties for DP600 [26] and for DX54D+Z [27]. In these studies, correlations between different material parameters have been quantified, which should be accounted for when characterizing statistical parameters of material variability. In other studies, the magnitude of material property variations within batches, from coil to coil [28] and from supplier to supplier [29] have been analysed, reporting up to 20% variation in material properties between different suppliers.

As discussed in Sect. 2, process disturbances can be classified based on their temporal dynamics. Besides distinguishing between product-to-product and long-term variations, different types of long-term variations can be differentiated. The ramp-up stage refers to gradual changes in process conditions during process start-up, for example caused by tool heating and changes in tribological conditions [30] or by a new tool set. Tool wear causes a slow and unidirectional change in process conditions that can be observed over many production runs. Alternatively, changes in material batches cause a sudden step-like jump in process conditions, as observed in Fig. 4. Although control systems for different types of process disturbances are discussed in this paper, it must be noted that the target of "First-Time-Right" production specifically relates to control during the ramp up phase (besides compensating for lack of knowledge on the transfer function between process settings and product properties when producing a product for the first time, see Sect. 2).

The abovementioned studies reveal a small fraction of the process disturbances to be accounted for in the design of metal forming control systems. The lack of studies on process variations in metal forming can be attributed to the complexity of

measuring in-process variations, and to the experimental burden of quantifying statistical parameters for the properties that can be properly measured, such as material parameters. Given the growing amount of data acquisition systems in modern factories, a potential solution may be to use process models to infer statistical parameters of process variations using large datasets of production data [31].

3.2 Sensors

Various types of sensors and actuators are employed in forming processes to measure and control the physical quantities, which determine the shape and properties of parts. The first categorization of sensors and actuators is related to the aim of the control strategy. The primary objective of forming processes is to set the shape of a part. Nowadays, the secondary aim of determining the physical properties by forming is almost as important [2]. However, the most common approach is to control the shape. Therefore, this essay focuses on the applications to control geometry. Further, sensors are categorized based on the primary output they generate, i.e. regardless of the physical measuring principle (e.g. tactile with induction coil or laser-optical to determine displacement). A sensor which determines the position or displacement is considered a sensor for the kinematics of the process.

Related to the control strategy, sensors directly measure a part-based quantity, i.e. the shape or a mechanical property such as the local strength or a process-based quantity such as a force or the velocity of a tool (see Fig. 6). In the latter case an estimator-based control strategy is necessary, while in the former case the directly measured physical quantity may be used for control or the directly measured physical quantity is an input for a model-based estimator. The term estimator refers to the definition of Havinga et al. [32]. Considering that there may be no unique mapping from process measurements to estimated state variables, in combination with significant modelling uncertainties, recent efforts have been directed at developing estimators in a probabilistic framework [33].

Taking into account the necessity of controlling the part geometry, part-based sensors of the shape as direct sensors and process-based sensors for kinematic quantities or dynamic quantities such as force are most relevant. Regardless of the measured physical quantity, to implement an effective control system, sensor sampling frequency should be high such that it is consistent with the actuation frequency of actuators. To minimize the need for transfer functions or to improve the accuracy of the transfer function, the measurement of position or derivatives thereof such as strain should be in close proximity to the workpiece or tool part of interest (see Fig. 7a).

The different sensing principles can be discussed best for cases of sheet forming at room temperature. In applications that focus on part-to-part control, measurements of the parts are performed after the forming process. An example of this is the approach by Garcia-Romeu et al. [34], where punch displacement and bending forces are measured with sensors during the process. Similarly, an offline goniometer

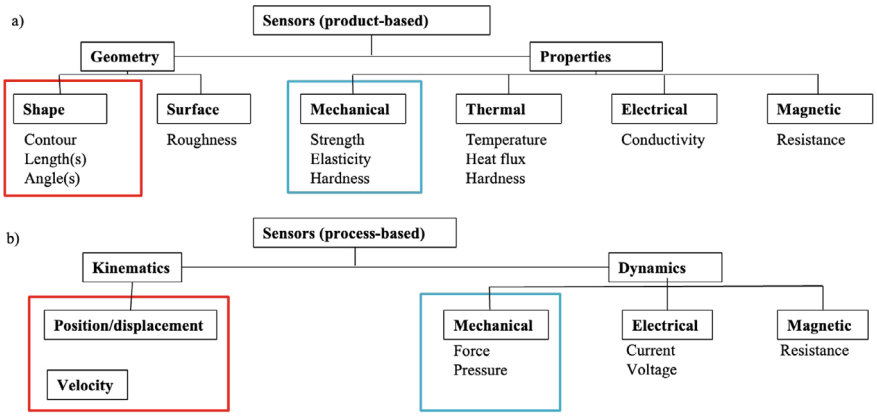
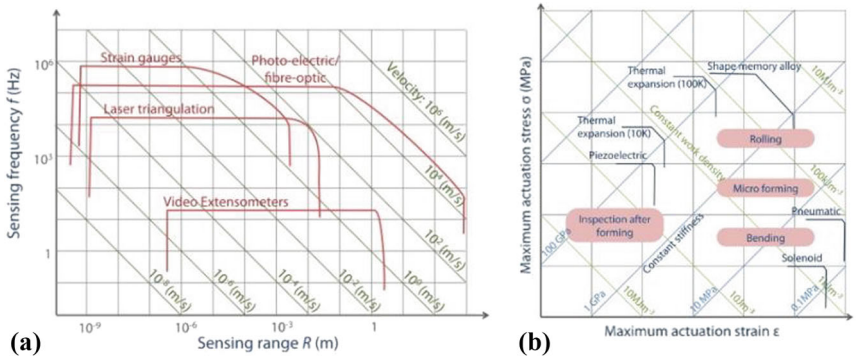


Fig. 6 Categorization of sensors into product-based or process-based



	Actuator type	Characteristics
Force/displacement	Motor + ball screw Hydraulic Piezo-electric Electromagnetic/plasma pulse Explosion Pneumatic (+switching valves)	Stiffness Maximum force/pressure Torque Power Stroke (range) Resolution/accuracy Resonant frequency Speed and acceleration

(c)

Fig. 7 **a** Correlation of sensing frequency versus sensing range for positional sensors, **b** Selection chart in metal forming for actuators **c** Actuator types in metal forming [12]

and hardness measurement were employed by Forcellese et al. [35]. The geometry of the bent sheet is then measured offline with a coordinate measurement machine. An improved strategy used additional in-line angular transducers to measure the bending angle [36]. This allows for a closing of the loop and a step-wise improvement of geometrical accuracy. The final geometry including springback compensation is obtained by attaining different punch displacement, measurement of the geometry after springback and continued forming to the final geometry. The requirements for the sensor in terms of frequency are low here. In a completely flexible sheet forming process such as single point incremental forming with counter pressure, Thiery et al. [37] apply a closed-loop control making use of an axial force sensor and a laser-distance sensor. So here a combination of workpiece based sensors and process based sensors are applied. Despite the challenges posed by industrial environments, such as vibrations, fluctuating lighting conditions, contaminants, and electrical interference, camera measurement systems are increasingly used in inline control. This is made possible by the analysis and real-time integration of data from various sources, supported by the continuous evolution of analysis algorithms [38].

3.3 Actuators

Given the scope of the current essay on reducing shape deviations, actuators aim at setting the displacement or load at a given point, line or surface of the workpiece. The current work adopts the definition of Huber et al. [39] that a mechanical actuator is a work-producing machine or device. Most actuators in this scope "operate in a linear fashion, causing a finite change in length". Analogously, a thermal actuator produces heat. The actuators produce work or heat upon a controllable signal. Performance characteristics of a mechanical actuator are stress, strain, volumetric power, and strain resolution. In integrated fashion, these result in forces and displacements. A selection of available actuators is given in Fig. 7c [12]. The tool geometry mainly sets the shape of sheet metal components. In cases with punch-die contact, the actuators are the movable tools. Depending on the drive type, the actuator can set a force or displacement. The typical actuator in this case performs a translatory motion. More flexible forming operations such as incremental forming, where the actuators act in multiple spatial directions need a more complex modelling approach in the control strategy. A combination of both scenarios is when additional actuators are used in the punch-die scenario.

With decreasing batch size, the importance of the transfer function, the model function for the process and the quality of the estimators in sensors increases compared to control applications, where fluctuations in workpiece material are the main reason for shape deviations.

Having this in mind, fluctuations in workpiece material, ambient conditions from one batch run to the next become less important.

A categorization can be made based on the physical principle, response times, accuracy, and possible feedback. The selection based on the performance criteria is

made with actuation stress versus strain diagrams as proposed by Huber et al. [39] and Isermann [40]. Figure 7b presents such a chart specifically for metal forming applications [12]. In case multi-step processes with multiple subsequent actuators are chosen, Duncan et al. proposed a strategy to select the number of actuators and the type [41].

3.4 Processes and Machines

Metal forming processes can be categorized in various ways, based on the type of production, the applied deformation states, temperatures, the type of component being processed, and so on. However, regarding the possibilities offered by process control, in this essay, we will distinguish two main categories. The first (i) includes discrete forming processes carried out in a single forming step. The second (ii) comprises forming processes executed both in a fixed and predetermined or in a variable number of steps.

The first case, as depicted in Fig. 8, involves processes such as sheet metal stamping, deep drawing, stretching, stretch-bending, extrusion, forging, etc. In these processes, equipment is generally used with a limited number of degrees of freedom, where most of the geometric properties of the component depend on the design of the tools. Generally, in these processes, the contact between the dies and the component occurs over extended surfaces, which makes it difficult both to insert sensors for direct measurements of the component and to incorporate actuators for potential process corrections. Furthermore, these processes are usually carried out at high production rates, reducing the time available for control systems to process and apply corrections to the component being worked on.

Figure 9 highlights the scheme of forming processes carried out in multiple steps. This category includes continuous processes, such as rolling, profiling of bulk elements or metal sheets, drawing, cold forging, or cold stamping performed in multi-station or progressive presses. In this case, we consider components processed within the same machine with the application of similar deformation states between successive stations. Here, the number of deformation steps is multiple, but the total

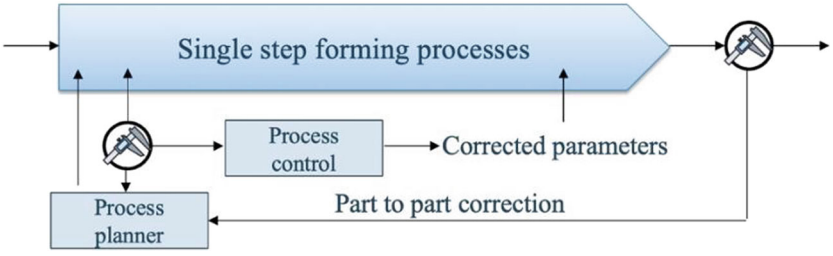


Fig. 8 Control for single step processes

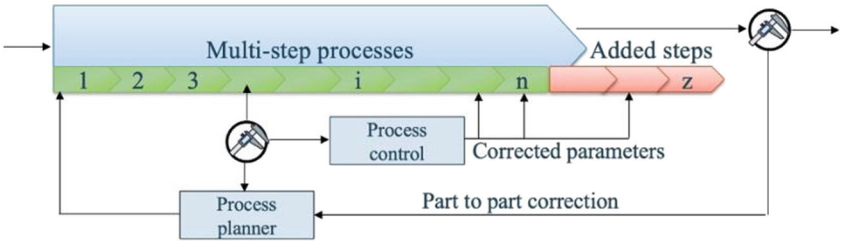


Fig. 9 Control for multi-step processes

number of steps is generally predetermined by the characteristics of the equipment, for example, by the number of available stations and dies. In this scenario, sensors can access the component between successive stations, and similarly, it's possible to correct the deviations of the properties of a workpiece during the subsequent stations. In this case, it is necessary that the subsequent stations are equipped with the necessary actuators to potentially apply such corrections to the component. The last scenario, includes those processes that allow for some variability in the number of steps, as shown by the red additional step from n to z in Fig. 9. Typical examples are incremental forming processes, such as ring rolling for bulk components, or single- or double-point incremental forming processes in the case of sheet metal working. These processes are generally carried out using one or more tools of relatively simple shape but moved according to complex kinematics, and are therefore executed on flexible machines, with multiple actuators and with the possibility of easily integrating sensors within them. Typically, the tools are moved along a predetermined path; however, control strategies can be used to real-time control the tool path, possibly adding additional steps, to adjust the target workpiece properties. It is important to note that, even in this case, despite the high flexibility, not all possible deviations can be corrected by subsequent steps.

3.5 Models

Process models are essential for the implementation of any control or optimization strategy, as mentioned in Sect. 2. Uses for such process models are diverse and include: (i) the creation of a starting set of process parameters (e.g., an initial toolpath strategy in forming processes with a high number of degrees of freedom); (ii) estimators when process control relies on indirect measurement from limited available sensors; and (iii) correcting the ongoing actions of the machine (closed-loop control).

The nature of the models also takes diverse forms. In conventional control theory, analytical models are usually required. "Analytical models" here refer to closed-form mathematical expressions that provide the relationship between process parameters and process outputs. The evolution of simulation accuracy and speed also allows

for numerical models, most often in the form of Finite Element Analysis (FEA), to be employed instead in most use cases. Further, with the increased availability of data-based approaches and improved computation, there are now also "black box" models which exclusively rely on data. The recent proliferation of Machine Learning (ML) methods provides diverse techniques for data-based models. Finally, it is also possible to consider hybrid process models that are two or more of the aforementioned approaches. These can take the form of ML models which receive some or all the underlying data from simulation, or physics-guided ML models.

In the context of First-Time-Right" production, the required capabilities of the process model can be expressed in terms of the nature of the process as described in the previous Sect. 3.4. In single step processes where, typically, a single mechanical action is taken in a very short period, the process model needs to provide a "perfect" prediction or a fully optimized set of process parameters. In continuous processes and incremental processes, subsequent steps can correct or compensate for earlier actions, therefore the models themselves need to provide a sufficient approximation for a control strategy to be implemented. In the latter scenario, even a perfect model does not ensure a successful outcome because of external uncertainties such as the variation of material properties and environmental factors. A stochastic view of processes is therefore more appropriate. Undoubtedly, extensive knowledge of the process and sensing limitations, as well as corresponding model availability, are essential for a successful choice of modelling strategy. Here, we briefly present both the uses of process models and their nature through a set of examples and insights from the literature. These findings are not limited to successful First-Time-Right" implementation, which is not yet common.

Further, models can be categorized by the aspect of the process they attempt to describe. Volk et al. [42], for example, explore a broad range of models aimed at identifying process limits. These include models of metal fracture, necking, wrinkling and dimensional accuracy. The same paper presents a diagrammatic procedure for selecting an appropriate model for process limit identification shown in Fig. 10. Analytical models are usually only possible for simple geometries (both tools and workpieces) and follow several simplifications and assumptions for material properties, friction, and other factors. In other words, they follow deterministic continuum mechanics approaches. One of the simplest processes to model is the bending of sheet metal. By assuming plane strain conditions (i.e. a sufficiently wide workpiece), equilibrium, an idealized material law and frictionless dies [42] provide a simple equation to predict bending springback based on tool angle. The same authors, with similar assumptions, produce an analytical description of the sheet stamping process.

In this instance [43], mathematical expressions linking the shape of the die with the forces at the blank holder and at the punch can be useful both for process design and control. For another bending process, incremental tube bending, Nazari et al. [44] produce a system of differential equations which can be solved numerically and allows a prediction of the required bending moment to achieve a desired amount of bending. In bulk forming processes, such as ring rolling, we can sometimes find that volume conservation alone is sufficient to provide a reasonable analytical constraint for curvature control, as in [45]. Ring rolling can be thought of as a

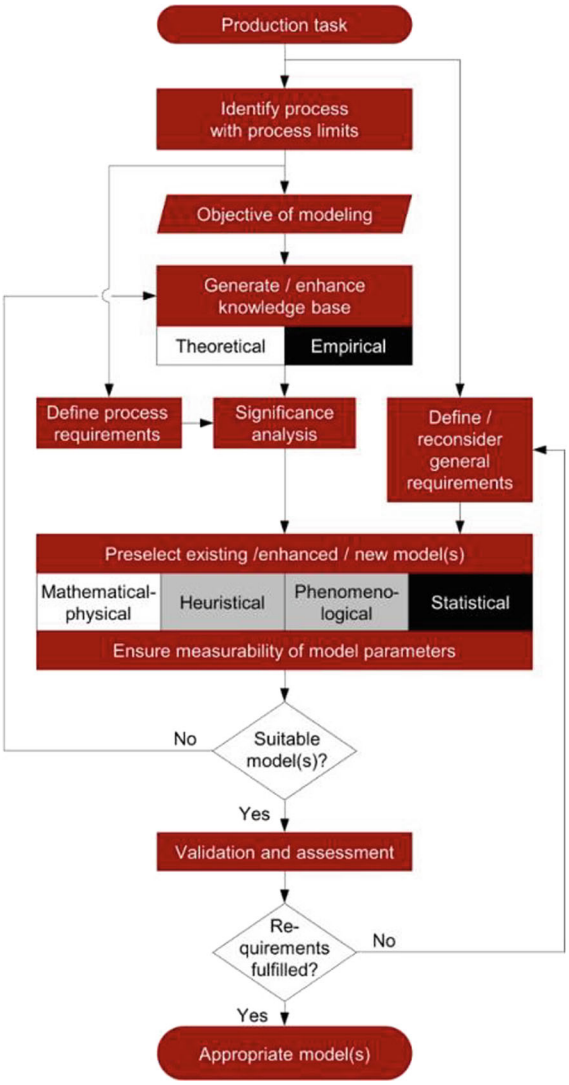


Fig. 10 A decision flowchart for process limit models [42]

continuous process which allows multiple passes, therefore it is not surprising that it affords fertile ground for control, as indicated previously. Incremental sheet forming takes many forms and provides extreme process flexibility, but the core deformation mechanics involve stretching which is difficult or impossible to reverse. Here, the use of volume conservation alone results in a well-known sine law, which deviates from observed deformation behaviour [46] because of material shearing. Therefore, more nuanced geometrical models had to be devised to predict thinning [47] and, more recently, force prediction models helped understand process mechanics [48]. In all the examples above, the analytical description is only possible by considering

the local deformation of the workpiece and assuming a closed-form description of the local geometry.

A specific analytical approach is the concept of model linearization around a nominal toolpath, that originates from control research and has been applied to incremental forming processes [49]. The idea is that the distributed effect of a control action on a product can be defined as an impulse function [50]. By defining a process model as a sequence of linear impulse functions, such model can be easily implemented in a Model Predictive Control system and used for closed-loop control of flexible forming processes [49].

Simulation-based models play an increasingly important role in metal forming. Numerical methods, especially Finite Element Analysis, are now essential tools in metal forming operations. They are integral to necessary activities such as the design of tooling but can also act as surrogate models for process control. For example, FEA is used in [51] to develop and validate a numerical model of roll forming which is then used to control the process inline. In [52] a virtual model of the strip bending process is used to demonstrate a control approach for springback compensation. However, the non-linear nature of metal forming processes and the complex physical phenomena involved such as surface interactions translate to a substantial computational cost. Despite continuous improvement in both algorithmic and computational efficiency over recent decades, it is still impractical to employ FEA in real-time control of most industrial processes.

More recently, data-based models have gained traction. These rely on the proliferation of sensing, data storage and computation technologies and can take advantage of both physical and numerical data. A recent review by Liewald et al. [53] details this trend and the associated challenges. They highlight that the increased volume and variety of data might open new opportunities, but it also introduces the demand to isolate the genuinely useful measurements from noise. Efficient data manipulation with increasing data availability also requires the development of new, more efficient algorithms. Perhaps the most prominent challenge in purely data-based, or "black box" models is the loss of "explainability", i.e. the ability to gain reliable knowledge and insight from the model rather than quantitative outcomes alone. For many critical applications, relying on such models represents an intolerable risk.

Some specific ML or AI approaches that show promise in the context of metal forming models have emerged. These include artificial neural networks, which are typically used to model a limited subset of process outcomes. The literature includes examples of using ANNs for predicting wrinkling [54], springback [55], friction [56] and for producing a forming limit diagram [57]. Metal forming benefits, of course, from the wider success that ANNs have had in relevant areas. For example, [58] used ANNs to model ductile damage in sheet metal forming while [56] showed that recurrent neural networks can model plastic behaviour accurately. Other concepts from the Machine Learning community, such as co-operative learning with multiple co-evolving ANNs have not been explored sufficiently in the context of metal forming, but perhaps offer a way to bring together the individual capabilities listed above.

The concept of the Digital Twin—the virtual representation of the physical process—is another recent addition to the manufacturing arsenal. These can rely on

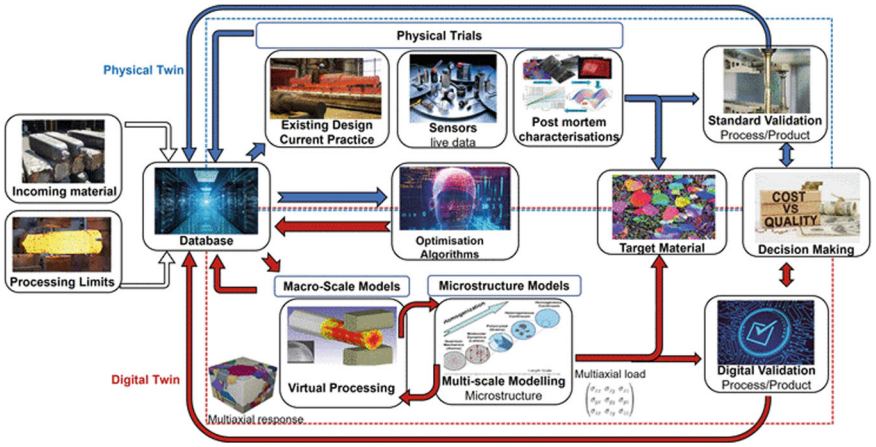


Fig. 11 A proposed configuration of a Digital Twin for a forging process [59]

a combination of FEA, physical sensing, historical data, ML and analytical models to calculate, most often not in real time, how the manufacturing system will react to certain conditions. Contrary to the physical system, all aspects of the performance can be interrogated thus giving a wider range of information for operation decisions. An illustration of this for a forging operation by the Advanced Forming Research Centre [59] is shown in Fig. 11.

It operates in conjunction with physical testing and encompasses multiscale simulation, optimization of process parameters and decision making. Another effort to construct a hybrid model, based on both simulation and physical data is shown by Tatipala et al. [60] in a workflow concept. These two illustrative examples highlight a major conclusion about the state-of-the-art: despite the excitement around the possibilities of Digital Twins, there are few, if any, successful implementations in the real world.

4 New Concepts and Applications

This chapter examines practical applications with respect to the different metal forming process categories identified in Sect. 3. It presents successful cases and ongoing integrations of control systems, addressing the challenges and limitations in these contexts. The chapter highlights a significant gap in the literature: while the integration of complex, real-time control systems into metal forming processes is a growing interest, comprehensive system studies are limited. Most research focuses on individual components of the control system, such as models, actuators, or sensors, rather than on their complete integration, indicating a disparity between theoretical development and practical implementation.

4.1 Single Step Processes

As identified in Sect. 3.4, this category encompasses forming processes that rely on complex geometry dies to deform components in a single action. A prime example is sheet metal stamping, where the main inline control systems involve adjusting the force of the blank holder. This adjustment aims to alter the material flow and the applied stress states, optimizing the stamping process.

An intriguing example of a control system applied to the cold stamping of steel sheet metal was presented in [61], where the authors proposed a method to control the punch force–stroke path by adjusting the blank holder force, operated by 12 hydraulic actuators, each equipped with force sensors. The system was dynamically modelled using a Multi-input Multi-output (MIMO) approach. However, the target trajectory was established based on a series of trial-and-error experimental tests, while the process model was based on a simplified two-dimensional analytical model, severely limiting the application results. Nevertheless, the approach successfully integrates advanced control strategies in metal forming processes leading to notable improvements, as seen in Fig. 12, in reducing the wrinkling magnitude and avoiding the onset of tears.

The same authors [62] later advanced the system by implementing a Model Reference Adaptive Control logic. This enhancement allowed for auto-tuning [63] of the control system gains during the process, enabling it to better adapt to disturbances such as variations in lubrication conditions or changes in sheet metal thickness. This adaptive approach further refined the system's responsiveness and effectiveness in

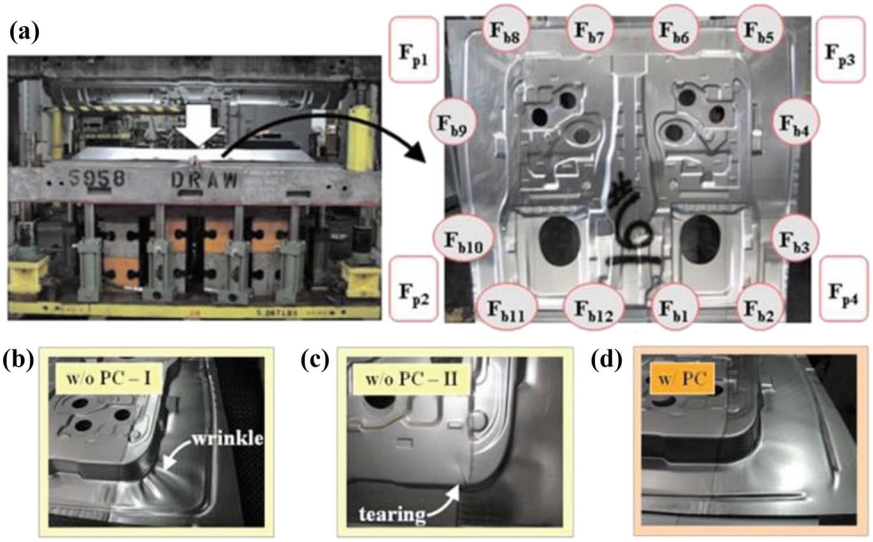


Fig. 12 **a** View of tools and stamped part with actuator and sensor positions **b** wrinkled part (8 ton constant blank holder force), **c** torn part (16 ton constant blank holder force), **d** improved part [61]

managing real-time process variations, ensuring consistent quality in the stamping operation. While these approaches are based on using the process forces as a real-time feedback signal, estimating the sheet metal flow and the potential onset of defects from force measurements can be extremely challenging with real time model, and is subject to various external disturbance variables. Consequently, in recent years, several authors have sought to develop integrated measurement systems in dies for the direct in-line measurement of the sheets draw in [15], to enhance control systems by avoiding errors introduced by estimation models. In most cases, the main issue in embedding sensors and actuators within tools lies in the limited space available, as the component is often entirely covered by the dies. Bäume et al. [64] demonstrated the effectiveness of using triangulation laser sensors combined with strain gauges embedded in the black-holder to monitor to directly measure the sheet draw-in. While Volk et al. [20] implemented Eddy current sensors to measure local distances between the tools and part, enabling the control of dimensional accuracy during production processes. The application limit of these systems may be due to the high cost of implementation, the poor compatibility with the industrial environment compared to the controlled laboratory setting, and the need to modify the architecture of the tools. More suitable for retrofitting current systems may be the implementation of piezoelectric sensors inserted within the fastening bolts to enhance the system's capabilities [65], or the implementation of a cost-effective optical sensor, for the draw in measurement as done in [66] and in [67]. It is worth highlighting, as emphasized in [68], that crucial for the proper use of sensors and actuators is their positioning within the system. Similar to sensors, the integration of actuators presents significant challenges. Beyond the primary actuator responsible for the die movement, the blank-holder units typically incorporate passive, uncontrolled actuators like mechanical and gas springs, primarily to minimize both spatial footprint and costs. However, these systems are inadequate for real-time inline control. An alternative is the adoption of hydraulic systems [69], possibly combined with segmented blank holders [70] as above discussed. However, these systems are usually expensive and voluminous, making it difficult to implement them in most tools. From this perspective, piezoelectric actuators appear more promising, as implemented in [64] to control the draw beads elements to control the material flow. However, their limited stroke and high sensitivity to shocks restrict their use in most industrial environments. A new promising solution appears to be based on passive systems that utilize magnetorheological fluids, controllable through the application of magnetic fields governed by a coil integrated into the system. These systems seem promising in terms of size and the range of applicable forces and have been applied in both blanking [71] and stamping operations [67]. Another magnetic-based solution, as proposed in [72], involves electromagnets embedded in the blank holder, providing variable pressure as a function of the power supply current.

While on one hand, the use of sensors that directly measure the quantities to be controlled complicates the system setup, on the other hand, they allow for a simplification of the models embedded in the control system. Nevertheless, it is still necessary to implement models that can calculate the correct response of the actuators.

Proportional-Integral-Derivative PID models remain the simplest, most cost-effective, and widely used. However, as the complexity of systems increases, they struggle to manage the growing number of variables involved. Multi-input Multi-output (MIMO) models appear more suited to handling complexities arising from the proliferation of sensors and actuators, as demonstrated in the initial example. Alternatively, models based on Model Predictive Control (MPC), which in their most advanced applications can evolve into Digital Twins (described in Sect. 3), seem promising despite the increased computational power required to manage predictive models. Indeed, Digital Twin (DT) models can quickly self-update through real-time data acquired from sensors, avoiding error drift over time. However, due to their complexity, they are often applied to investigate only specific aspects. For instance, Ihlenfeldt et al. [73] proposed the use of a DT to model the stiffness of the press and tools system in order to compensate for its flexions, according to the program flow shown in Fig. 13.

Similarly, Gan et al. [74] implemented these models to update the friction coefficients of the numerical model of a stamping process based on measured process forces, compensating for the effects of tool wear. In a similar application Link et al.

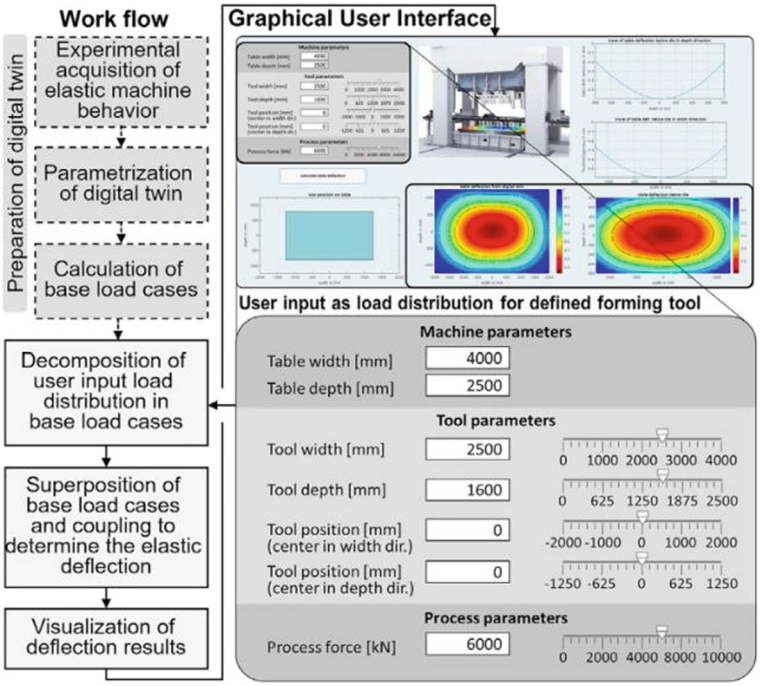


Fig. 13 Program flow for the tool stiffness compensation by using an interactive digital twin [73]

[75] proposed a Digital Twin model for sheet metal forming that predicts and optimizes contact conditions to improve thickness distribution. The model controls selective lubrication and uses a Machine Learning algorithm, trained via numerical simulation, to predict friction distribution, while receiving real-time feedback on process load and blank holder force.

Despite the recent increase in publication, the maturity level of these systems currently seems inadequate for their implementation in in-line control systems and is used for quality control activities or predictive maintenance. In the case of applications related to metal forming, the proposed Digital Twins are mostly used for preventive maintenance operations, as in [76] for bending systems, or for tool commissioning activities, as in [77] for stamping processes.

While AI-based models show great promise in managing multivariable control systems, as demonstrated in [78], their practical applications in the field covered by this paper are not very extensive. This highlights a gap between theoretical advancement and real-world implementation in this specific area. Indeed, in this field, most contributions based on Iterative Learning [79] and Deep Learning [80] techniques are often used as advanced tools for numerical optimization in tool design, rather than for implementing control systems. As identified by Endelt [81], the main challenges in applying these models, and directivity derive from the software and hardware architecture of most industrial stamping presses. In these, it can be extremely difficult to implement sensors and data acquisition and processing systems capable of working in real-time. At the same time, most of the actuators allowed to actively control the process parameters, such as those previously illustrated, remain the prerogative of prototype systems developed at the laboratory level.

These difficulties, which are evident in the above discussed case where there is a single main control parameter, are present in other single step processes as well, where the control variables are one, or worse, more than one. In the first case lies the simpler bending processes, where the primary objective is generally the control of the final bending angle. However, this problem is extremely challenging as springback only occurs upon unloading. Consequently, only in the simplest bending configurations, such as air bending, is it possible to overbend the piece, after a direct measurement of the bending angle, to compensate and correct the component. Whereas in most cases, the correction can only be introduced for subsequent components.

The literature analysis highlights how several contributes focus on the development of sensors [82] and algorithms [83] for the direct measurement or to estimate the springback magnitude, however cases of actual in-line control are effectively proposed almost exclusively for the production of simple geometries [84]. Then for more complex ones, where multiple control variables are involved, most of the proposed models aim to predict springback ex-ante [85], or possibly to introduce corrections on subsequent components [86].

While an effective solution to the in-line control of the final bent angle still seems challenging, in-line control systems applied to forming technologies carried out in a single step appear very promising when used to identify process drifts. As analyzed by Liewald et al. [53], data-driven models represent an effective way to identify defects due to tool wear, shown in Fig. 14 or variations in boundary conditions such

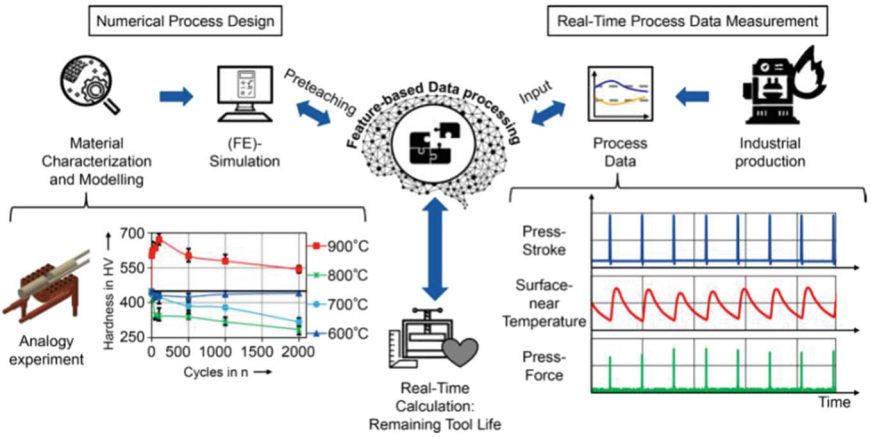


Fig. 14 Concept of a feature-based data processing for real time tool life calculation based on numerical process design and real time data measurements [53]

as die temperature during a production ramp-up stage. The former case has been successfully applied in monitoring tools for both the sheet blacking [87] process and the hot forging one [88], while the latter represents a typical problem in multi-stage cold forging productions.

These models enable process efficiency improvements, but they are generally not linked to an in-line control system that would automatically correct process parameters. It appears, therefore, that further effort is still required to implement them in a complete closed-loop control system.

4.2 Multi Step Processes

This paragraph focuses on the analysis of multi-step forming processes. Whereas the number of steps is constrained (es. the fixed number of stations in a cold forging press), often due to the architecture of the manufacturing system being used or unconstrained (such as in the ring rolling process or in sheet incremental forming).

4.2.1 Constrained Number of Steps

The first case addresses processes where the number of forming steps matches the number of forming stations, each applying a defined level of deformation. These processes are highly productive and typically performed by dedicated systems. Unlike the processes discussed earlier, this setup allows for direct measurement of certain part parameters between successive steps and enables correction of deviations in subsequent stages. A prime example is the rolling process, performed on

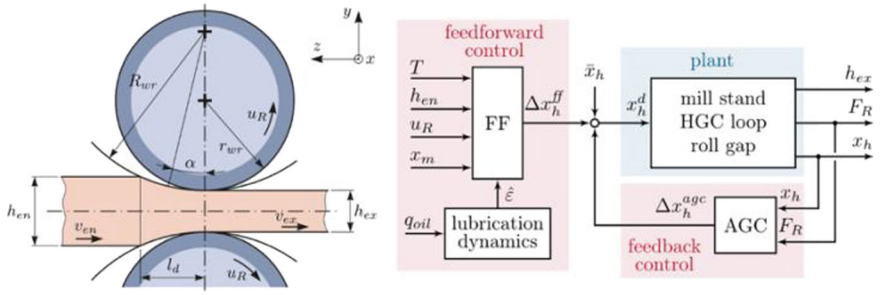


Fig. 15 Roll gap geometry and proposed control structure including the lubrication dynamics [91]

non-reversing mill, where the main control parameter is the sheet/plate thickness [89], which is determined by the roll gap. Other relevant parameters might include surface roughness and final mechanical properties [90]. The advantage of this process lies in the generally simple tool geometry and the automation of roll distance in most industrial plants, facilitating effective in-line corrections. For instance, in Fig. 15 Muller et al. [91], describe a feedforward controller model that accounts for the interaction between different rolling mill cages. This model uses advanced mathematical calculations to predict roll force and control inputs, minimizing deviations in strip thickness. It also integrates disturbances due to varying friction coefficients, using a lubrication dynamics model based on oil feed rate measurements. Accurate signal synchronization and reliable sensor data are critical for the feedforward controller's effectiveness. Similarly, Shulte [92] proposed an adaptive control model that dynamically adjusts deformation at each roll stand in a tandem rolling mill, optimizing thickness and strip roughness.

The roll forming process, used for producing bulk or sheet profiles with complex sections, presents additional challenges compared to rolling. While the overall plant size is smaller than that of a rolling mill, the architecture is similar, consisting of a series of stations with shaped rollers applying deformation to the component. However, the geometric complexity of the sections, such as diameter, roundness, and thickness distribution in a profiled tube, introduces more variables to manage, making it difficult to embed sensors for directly measuring control variables or gathering sufficient data for estimation. Additionally, since each roller pair applies localized deformation, determining corrective actions for subsequent stations becomes complicated. This issue is further exacerbated by the manual operation of many degrees of freedom necessary for setup adjustments.

To address these challenges, two approaches are found in the literature. The first, more comprehensive but complex, involves upgrading machine architecture by automating more degrees of freedom and incorporating in-line sensors. Ren et al. [93], exemplify this approach by developing a Digital Twin for a roll forming line, which monitors and controls aluminum profile production. This system includes actuators, sensors for process correction and monitoring, and a real-time laser measurement system, all integrated with COPRA modeling software and an AI algorithm for

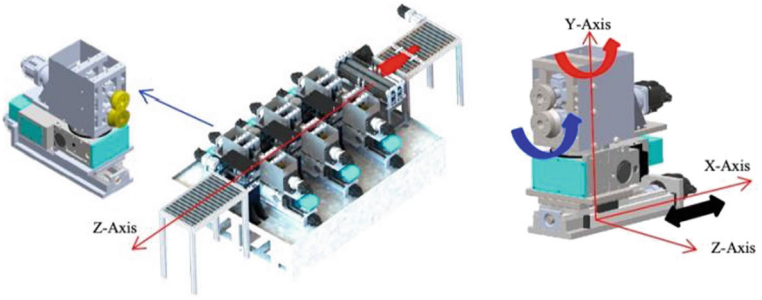


Fig. 16 Flexible roll forming machine (left) and forming module (right) [96]

process adjustments. However, this method requires extensive hardware and software modifications.

Groche et al. [94] initiated this concept of flexible roll forming, followed by others like Woo et al. [95] and Sheu et al. [96], who proposed stations with more automated actuators and redesigned tools to take full advantage of the new architecture. Another variation was proposed by Abeyrathna et al. [97], who adapted the concept for stand-alone equipment aimed at rapid prototyping, emphasizing increased geometric complexity of profiles with non-constant sections rather than enhanced control capabilities. [98] (Fig. 16). Consequently, few contributions focus on integrating these systems with advanced controls.

The second approach focuses on integrating specialized stations into the production line to control and correct common defects. For instance, as proposed in [99] this method targets specific defects such as bow and twist caused by uncontrolled longitudinal strains during the process. It involves inserting a station whose sole purpose is to straighten the already shaped profiles. While this reduces machine costs, it offers less flexibility in the range of corrections that can be applied. Similar systems, commonly used in industrial lines, are employed to control profile twist and align edges for in-line welding operations.

In cold forging processes, typically performed on multi-station machines at high production rates, accuracy issues [100] can arise both due to thermal variations during the ramp-up stage [101] and variations in the incoming material or the wear state of the tools. The modular distribution of stations further adds complexity to the integration of sensors and actuators, given the limited available space and the high process frequency [100]. To overcome these limitations, several solutions are proposed in the literature. Qin [102], for example, compares two different control approaches, passive and active, based on the use of smart materials to compensate for die expansion due to temperature increases during the ramp-up phases. A more comprehensive approach is proposed by Liwald et al. [103], where a hydraulic actuator, or alternatively an electric actuator [104], is integrated into the dies to control the tools position in cold forging operations. These actuators can be feedback-controlled based on force measurements using thin-film sensors [105] specifically developed to be integrated into cold forging dies, aiming to improve forgeability and compensate for geometrical defects caused

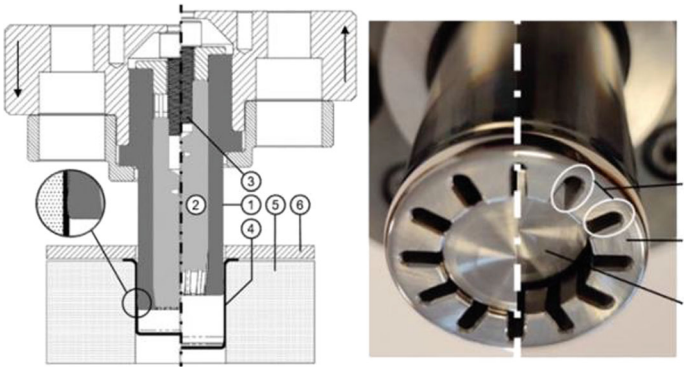


Fig. 17 Ironing punch with adjustable diameter: (left) sectional view, (right) inner mandrel position during ironing and retraction [107]

by tool stiffness and boundary conditions. A similar control system, which adjusts actuator position based on force measurements to compensate for friction variations, has been proposed for oscillating cold forging in gear manufacturing [106]. However, this system operates on dedicated equipment with fewer limitations. More recently, for the ironing step, a hollow punch equipped with an inner mandrel capable of expanding its diameter has been suggested. This system allows for small diameter adjustments on the order of micrometers and, despite its compact size, shows promise for process control, as illustrated in Fig. 17 [107].

However, the system is still in the prototype stage and hasn't been integrated into an inline control system. In cold forging control, Uribe et al. [108] discusses developing digital twin processes using Proper Orthogonal Decomposition models for a one-blow cold upsetting process in copper billets, validated with offline measurements. However, integrating these models into complex industrial applications with inline measurements remains largely unexplored.

Similar issues affect progressive die sheet stamping. As highlighted by Budnick et al. [109], the strip's transfer phase can introduce dynamic effects, such as vibrations, leading to positioning errors. In [110], an AI-based control system is proposed to mitigate these effects and improve positioning accuracy in the sheet feeding system. However, this system has only been validated numerically, with no experimental validation provided, leaving open the challenges of in-die measurement of these effects and the integration of a potential compensation system.

4.2.2 Unconstrained Number of Steps

The second case accounts for processes, where the number of steps is unrestricted, or rather not directly constrained by the machine's architecture or the number of forming stations. A clear example lies in incremental processes, where the workpiece is manipulated during several discrete or continuous steps. When the process does not

require part-specific tooling, it is also called flexible concerning target geometry. A recent review of flexibility in metal forming can be found in [10], where its definition is expanded beyond geometrical shape to include lot size, achievable accuracy, and workpiece material. Flexibility can vary within process families, often with simple modifications which usually carry a trade off with control. In other words, higher flexibility usually necessitates increased control. Incremental processes are particularly relevant to efforts to control metal forming and many noted efforts for open- and closed-loop control of metal deformation appear in the context of this category of processes. This is because they offer both the time and the opportunity to observe the changing workpiece but also typically involve more degrees of freedom and necessitate more complex intervention. Beyond Incremental Sheet Forming (ISF) and its many incarnations, prominent flexible incremental processes include incremental tube bending, incremental roll forming, ring rolling, and open die forging.

Incremental processes were the hallmark of the traditional job shop. Craft techniques like hammer and anvil forging, the English Wheel and manual spinning were ubiquitous in industry for many decades, because of their ability to serve diverse objectives and produce little waste. In that context, the human operator continually observed and assessed the progress of the workpiece towards the target part and adjusted the toolpath and other process parameters based on intuition and heuristics. The increasing scarcity of these skills and modern demands for repeatability, accuracy and high-volume production make it clear that understanding the underlying mechanics of these processes is a pressing need. At the same time, it is no surprise that a great number of proposed incremental processes draw inspiration from corresponding craft processes.

Bowen et al. [4] document the many techniques researchers have used to replicate the level of control that human operators are capable of in craft metal forming. These have included rule-based control systems based on known strategies, analytical models, and neural networks. To develop these approaches, an equally diverse set of sensors and knowledge capture have been documented: training manuals, human motion capture and haptics, tool trajectory capture and more.

Because the deformation in many incremental processes is local, additional complications are added to the challenge of closed-loop control. Hartmann et al. [111], focus on toolpath creation for the driving process. Depending on the tools used, this process can provide both local stretching and local compression. The human operator is replaced with a robotic arm that holds and actuates the workpiece. Automation of toolpath generation is achieved in this work by training an Artificial Neural Network. Their approach includes the need to discretize and parameterize both the workpiece geometry and the target geometry so that they can be mapped to each other (see Fig. 18). This is not a trivial task because the workpiece moves in 3D space with only one point of reference with respect to the tools and the workpiece is continually deforming.

Another example of increasing flexibility requiring increased control is spinning which can be performed with or without a mandrel. A neural network approach was developed in [112] to generate toolpaths for multi-pass spinning with a mandrel.

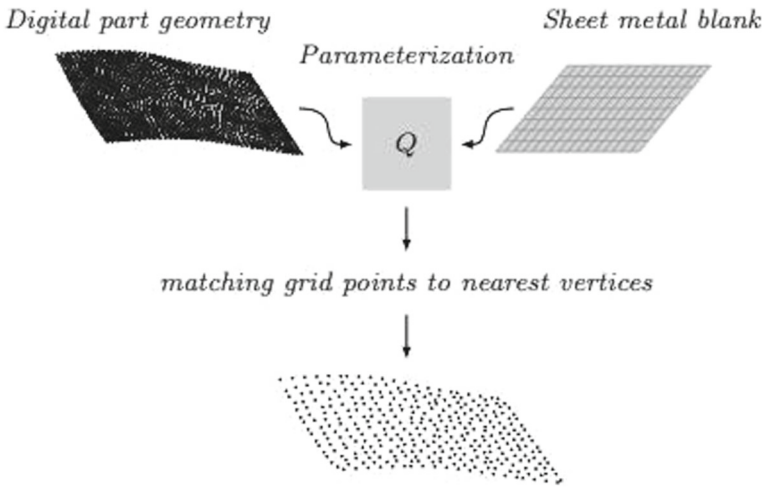


Fig. 18 Methodology to map the target geometry to the blank so that an Artificial Neural Network could model the process [111]

Asymmetric mandrel-free spinning substitutes the role of the mandrel with an additional moving tool on the inside of the workpiece [113]. In addition to removing a fixed cost for each part in the form of mandrel production, the mandrel-free setup allows for a range of asymmetric geometries to become available without dedicated tooling [114]. The trade-offs include the need to design more complex toolpaths and reduced stability of the workpiece in this highly dynamic process. The latter can be addressed to some extent by an additional roller at the edge of the workpiece, a technique often employed in the manual version of spinning and verified by FEA [115]. The design of the blank to facilitate formability and reduce waste is another challenge. In this context of mandrel-free spinning, this was addressed in [116] both through an iterative simulation approach and by relating the blank planform to the curvature of the target part. The evolution of a craft process like spinning into a more rigid industrial process and then towards increasing flexibility with an additional need for control highlights several of the recurring themes in this essay. The importance of hardware modifications on control requirements is one such challenge.

Despite the increasing interest in the automation of craft processes, the most prominent incremental process is the ISF process. ISF did not evolve from a craft process but was inspired by the availability of CNC which flourished in subtractive processes. In most versions of the process, a metal sheet is clamped at its boundaries and a moving stylus-like tool moves across its surface and deforms it by stretching and shearing. Formability and control can be improved by the presence of a second moving tool or some other supporting substrate or die on the other side of the sheet. Therefore, the main control input to the process is the trajectory of the tool(s). Because the ISF is primarily a stretching process, reversing mistakes or deviations from the target geometry can be difficult or impossible. Therefore, the

First-Time-Right" objective is crucial to the process viability. At the same time, the local nature of the deformation causes challenges not present in discrete processes discussed in previous sections. For example, Bambach et al. [117] explore the impact of local springback and address it through a combination of multi-stage forming and stress-relief annealing, producing better accuracy than a single-stage approach.

Early efforts for closed-loop control include the use of a stereoscopic camera and linearization of the process "impulse response" [49] to allow for on-line optimization—this approach is proposed by the authors for the broader category of processes with "mobile tools". Gooijer et al. [118] propose improving the linearization of the nominal path and subsequent optimization by making it *history aware*, i.e. maintaining a memory of past corrections. At the same time, many researchers employed neural networks to control versions of the ISF. Thiery et al. [37] employ an artificial neural network trained on experimental data for closed-loop control after augmenting the process with an active medium substrate. The control strategy incorporates both the actuation of the forming tool and the adjustment of the pressure in the active medium, see Fig. 19.

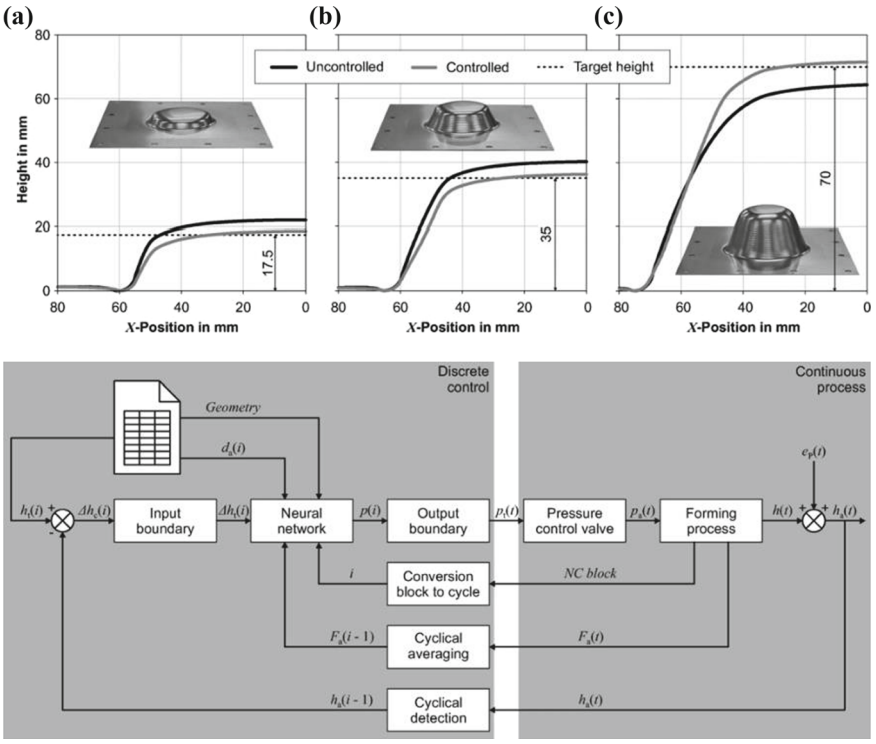


Fig. 19 Control of product height in Incremental Sheet Forming with active medium and the control scheme employed in [37]

A similar evolutionary history was followed by other incremental processes, such as the three-roll bending of tubes and sheet metal. In this case, the development of automatic controls is highly beneficial given that, on one hand, the process employs a relatively simple set-up with few degrees of freedom, while on the other, it is extremely difficult to pre-calculate the rollers positions that would allow for achieving the desired bending radius and angle, despite the process being carried out in successive bending steps [119]. However, while the implementation of closed-loop control cycles [120], based on analyst and FEM (Finite Element Method) models, combined with in-line force [121] or displacement [122] measurements are well-established, new interest arises from potential applications of machine learning to further refine the models used [123].

This evolution has not only impacted the sheet metal forming processes but, obviously, also bulk forming ones, such as the ring rolling process [124]. Jenkouk et al. have demonstrated the effectiveness of adaptive controls also when applied to the virtual process, to enhance the precision of the process model, thereby improving the overall process outcomes in simulations [125]. While Arthington et al. [45] focused on the use of in-line control to extend the process capabilities, allowing for the production of rings with variable radii using the standard process setup. This advancement highlights the potential for more flexible and efficient manufacturing processes through the strategic integration of real-time control systems. More recently, Liang et al. have worked on integrating Digital Twin models applied to both rectangular section rings [126] and conical elements [127]. Lastly, Lafarge et al. proposed the application of data-based models for process control through the use of soft-sensors for estimating and controlling the microstructural properties of the ring according to process parameters [128].

Finally, it is worth mentioning how the accelerating interest in incremental processes is reflected in the academic literature and large scale research projects such as HAMMER: Hybrid Autonomous Manufacturing Moving from Evolution to Revolution [129] (see Fig. 20a). The latter aims to develop the next generation of fabrication shops by creating flexible autonomous robotic processes. But at the same time, there is a new generation of commercial endeavours relating to incremental forming. The less flexible processes such as conventional spinning with a mandrel have existed as CNC industrial machines for several decades. More recently, however, several commercial attempts have been made to popularise the ISF. These include Ford's F3T [130], the startup Machina Labs [131] (see Fig. 20b) and the Figure process, currently sold by Desktop Metal [132]. The latter claims the capability for First-Time-Right production with 15 min setup time and 40 min forming time for a car fender demonstrator part [133]. There is no publicly available information on the underlying methodology for these presumably novel control approaches but the metal forming community should be encouraged by the wider impact on end-users.

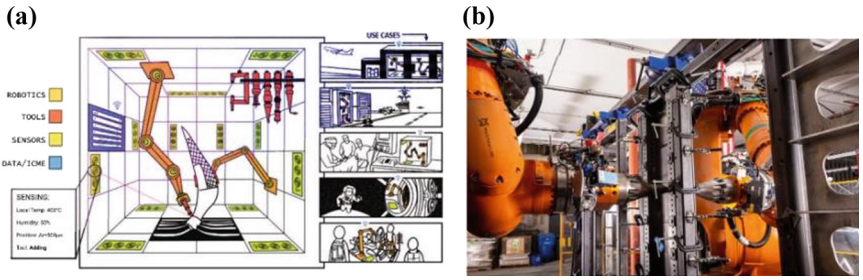


Fig. 20 Examples of recent, ISF based: **a** large scale research project [129], and **b** commercial systems [131]

5 Conclusions and Outlooks

This essay addresses the paradigm shift required in metal forming processes to achieve comprehensive control system implementation throughout the entire production chain. The goarl is to move towards metal forming production systems that enhance productivity by reducing setup times, costs, and production waste, in alignment with the First-Time-Right manufacturing concept. Achieving this requires a systematic, integrated approach where sensors, actuators, and cutting-edge models work together to monitor and adjust process parameters in real time, ensuring that deviations are promptly and effectively corrected, ultimately meeting the First-Time-Right objective.

Despite the well-established notion of controlled metal forming, the literature predominantly focuses on singular aspects, leaving a gap in detailing their integrated application. This identifies a vast domain for innovation, suggesting that while the conceptual groundwork is laid, the path towards practical, integrated control systems in metal forming remains largely unexploited, offering several opportunities for future research and development. The exploration of standard models and cutting-edge approaches like Digital Twins and AI-driven methodologies reveals a significant potential yet to be fully harnessed in industrial applications.

At the same time, the essay emphasizes that, to move toward the Firt-Time-Right paradigma, it is essential to integrate control systems operating at multiple levels, following a hierarchical structure, which allows for interventions at various stages of the production process. At the most basic level, control systems must offer solutions for immediate corrections within individual machines/tools, addressing issues such as disturbances or variations in boundary conditions. At a higher level, the system should implement part to part corrections, ensuring an integrated approach to adjust and optimize the process parameters.

The practical implementation of such hierarchical control systems is heavily dependent on the underlying machine architecture. A significant challenge is that many traditional manufacturing systems were designed decades ago, undergoing several stages of optimization but often based on architectures initially developed when sensors and automation systems underperforming and expensive. This

mismatch between existing machinery designs and the needs of contemporary control systems often limits the application of such technologies in traditional processes. However, this challenge also presents an opportunity for innovation in machine/process design and control. As a result, an important and actual challenge is to integrate such control systems within traditional metal forming processes, implementing these changes in terms of sensorization, actuation, and necessary control models. All this in an economically efficient manner by balancing the costs associated with this integration with the benefits observable from the process, in terms of: (i) increased flexibility, (ii) the possibility to expand process capacities, (iii) waste reduction, (iv) try-out time reduction, (v) tightening specification, (vi) improved process stability. In conclusion, achieving the First-Time-Right paradigm in metal forming requires not only the integration of advanced control systems but also a fundamental rethinking of traditional machinery architectures. While this is an ambitious goal and may not be fully attainable for all technologies, pursuing it can still lead to substantial improvements. However, several open challenges remain to be investigated, such as the practical integration of sensors and actuators, the development of suitable control models, and balancing the costs of implementation with the observable process benefits.

References

1. Duflou J, Sutherland J, Dornfeld D, Herrmann C, Jeswiet J, Kara S, Hauschild M, Kellens K (2012) Towards energy and resource efficient manufacturing: a processes and systems approach. CIRP Ann Manuf Technol 61:587–609. https://doi.org/10.1016/j.cirp.2012.05.002
2. Tekkaya A, Allwood J, Bariani P, Bruschi S, Cao J, Gramlich S, Groche P, Hirt G, Ishikawa T, Löbbe C, Lueg-Althoff J, Merklein M, Misiolek WL, Pietrzyk M, Shivpuri R, Yanagimoto J (2015) Metal forming beyond shaping: predicting and setting product properties. CIRP Ann Manuf Technol 64:629–653. https://doi.org/10.1016/j.cirp.2015.05.001
3. Hu S, Zhu X, Wang H, Koren Y (2008) Product variety and manufacturing complexity in assembly systems and supply chains. CIRP Ann Manuf Technol 57:45–48. https://doi.org/10.1016/j.cirp.2008.03.138
4. Bowen D, Russo I, Cleaver C, Allwood J, Loukaides E (2022) From art to part: learning from the traditional smith in developing flexible sheet metal forming processes. J Mater Process Technol 299
5. Ivanov D, Keskin B (2023) Post-pandemic adaptation and development of supply chain viability theory. Omega (United Kingdom) 116:102806. https://doi.org/10.1016/j.omega.2022.102806
6. Medini K, Da Cunha C, Bernard A (2012) Sustainable mass customized enterprise: key concepts enablers and assessment techniques. IFAC-PapersOnline 45:522–527
7. Peng R, Tang J, Yang X, Meng M, Zhang J, Zhuge C (2024) Investigating the factors influencing the electric vehicle market share: a comparative study of the European Union and United States. Appl Energy 355:122327. https://doi.org/10.1016/j.apenergy.2023.122327
8. Kalpakjian S, Schmid S (2017) Manufacturing processes for engineering materials. Pearson
9. Campi F, Favi C, Mandolini M, Germani M (2019) Using design geometrical features to develop an analytical cost estimation method for axisymmetric components in open-die forging. Procedia CIRP 84:656–661

10. Yang D, Bambach M, Cao J, Duflou J, Groche P, Kuboki T, Sterzing A, Tekkaya A, Lee C (2018) Flexibility in metal forming. CIRP Ann 67:743–765. https://doi.org/10.1016/j.cirp.2018.05.004
11. Yanagimoto J, Banabic D, Banu M, Madej L (2022) Simulation of metal forming—visualization of invisible phenomena in the digital era. CIRP Ann 71:599–622. https://doi.org/10.1016/j.cirp.2022.05.007
12. Allwood J, Duncan S, Cao J, Groche P, Hirt G, Kinsey B, Kuboki T, Liewald M, Sterzing A, Tekkaya A (2016) Closed-loop control of product properties in metal forming. CIRP Ann Manuf Technol 65:573–596. https://doi.org/10.1016/j.cirp.2016.06.002
13. Bleicher F, Biermann D, Drossel W, Moehring H, Altintas Y (2023) Sensor and actuator integrated tooling systems. CIRP Ann 72:673–696. https://doi.org/10.1016/j.cirp.2023.05.009
14. Weckenmann A, Jiang X, Sommer K, Neuschaefer-Rube U, Seewig J, Shaw L, Estler T (2009) Multisensor data fusion in dimensional metrology. CIRP Ann Manuf Technol 58:701–721. https://doi.org/10.1016/j.cirp.2009.09.008
15. Cao J, Brinksmeier E, Fu M, Gao R, Liang B, Merklein M, Schmidt M, Yanagimoto J (2019) Manufacturing of advanced smart tooling for metal forming. CIRP Ann 68:605–628. https://doi.org/10.1016/j.cirp.2019.05.001
16. Albertelli P, Strano M (2017) Tube bending machine modelling for assessing the energy savings of electric drives technology. J Clean Prod 154:83–93. https://doi.org/10.1016/j.jclepro.2017.03.212
17. Saidur R (2010) A review on electrical motors energy use and energy savings. Renew Sustain Energy Rev 14:877–898
18. Mypati O, Mukherjee A, Mishra D, Pal S, Chakrabarti P, Pal A (2023) A critical review on applications of artificial intelligence in manufacturing. Artif Intell Rev. https://doi.org/10.1007/s10462-023-10535-y
19. Leng J, Wang D, Shen W, Li X, Liu Q, Chen X (2021) Digital twins-based smart manufacturing system design in Industry 4.0: a review. J Manuf Syst 60:119–137
20. Maier S, Liebig A, Kautz T, Volk W (2017) Tool-integrated spring back measuring system for automotive press shops: a contribution to the quality control of complex car body parts. Prod Eng 11:307–313. https://doi.org/10.1007/s11740-017-0725-8
21. Havinga J (2016) Optimization and control of metal forming processes. University of Twente, Enschede, The Netherlands
22. Nakahata R, Seetharaman S, Srinivasan K, Tekkaya A (2022) A control strategy for incremental profile forming. J Manuf Process 79:142–153. https://doi.org/10.1016/j.jmapro.2022.04.034
23. Endelt B (2017) Design strategy for optimal iterative learning control applied on a deep drawing process: recognising that stamping and deep-drawing operations are repetitive processes—which can learn and improve based on process history. Int J Adv Manuf Technol 88:3–18. https://doi.org/10.1007/s00170-016-8501-z
24. Polyblank J, Allwood J, Duncan S (2014) Closed-loop control of product properties in metal forming: a review and prospectus. J Mater Process Technol 214:2333–2348. https://doi.org/10.1016/j.jmatprotec.2014.04.014
25. Col A (2003) Investigation on press forming scatter origin. In: Proceedings of the sixth international ESAFORM conference on material forming, Salerno, pp 183–186
26. Aspenberg D, Larsson R, Nilsson L (2012) An evaluation of the statistics of steel material model parameters. J Mater Process Technol 212:1288–1297. https://doi.org/10.1016/j.jmatprotec.2012.01.016
27. Wiebenga J, Atzema E, An Y, Vegter H, Van Den Boogaard A (2014) Effect of material scatter on the plastic behavior and stretchability in sheet metal forming. J Mater Process Technol 214:238–252. https://doi.org/10.1016/j.jmatprotec.2013.08.008

28. Hora P, Heingärtner J, Manopulo N, Tong L (2011) Zero failure production methods based on a process integrated virtual control. In: AIP conference proceedings, vol 1383, pp 35–47
29. Miranda S, Cruz D, Amaral R, Santos A, César de Sá J, Fernandes J (2021) Assessment of scatter on material properties and its influence on formability in hole expansion. Proc Inst Mech Eng Part L J Mater Des Appl 235:1262–1270. https://doi.org/10.1177/14644207219948868
30. Veldhuis M, Heingärtner J, Krairi A, Waanders D, Hazrati J (2020) An industrial-scale cold forming process highly sensitive to temperature induced frictional start-up effects to validate a physical based friction model. Procedia Manuf 47:578–585
31. Havinga J, Mandal P, Van Den Boogaard T (2019) Bayesian model-based state estimation for mass production metal forming. In: IOP conference series: materials science and engineering, vol 651
32. Havinga J, Van Den Boogaard T, Dallinger F, Hora P (2018) Feedforward control of sheet bending based on force measurements. J Manuf Process 31:260–272. https://doi.org/10.1016/j.jmapro.2017.10.011
33. Havinga J, Mandal P, Van Den Boogaard T (2020) Exploiting data in smart factories: real-time state estimation and model improvement in metal forming mass production. Int J Mater Form 13:663–673. https://doi.org/10.1007/s12289-019-01495-2
34. Garcia-Romeu M, Ciurana J, Ferrer I (2007) Springback determination of sheet metals in an air bending process based on an experimental work. J Mater Process Technol 191:174–177. https://doi.org/10.1016/j.jmatprotec.2007.03.019
35. Forcellese A, Gabrielli F, Ruffini R (1998) Effect of the training set size on Springback control by neural network in an air bending process, vol 80
36. Wang J, Verma S, Alexander R, Gau J (2008) Springback control of sheet metal air bending process. J Manuf Process 10:21–27. https://doi.org/10.1016/j.manpro.2007.09.001
37. Thiery S, Zein M, Abdine E, Heger J, Khalifa N (2020) Closed-loop control of product geometry by using an artificial neural network in incremental sheet forming with active medium. https://doi.org/10.1007/s12289-020-01598-1
38. Pierer A, Hauser M, Hoffmann M, Naumann M, Wiener T, de León M, Mende M, Koziorek J, Dix M (2022) Inline quality monitoring of reverse extruded aluminum parts with cathodic dip-paint coating (KTL). Sensors 22:9646. https://doi.org/10.3390/s22249646
39. Huber J, Fleck N, Ashby M (1997) The selection of mechanical actuators based on performance indices
40. Isermann R (2007) Mechatronische systeme. Springer, Darmstadt
41. Duncan S, Allwood J, Garimella S (1998) The analysis and design of spatial control systems in strip metal rolling, vol 6
42. Volk W, Groche P, Brosius A, Ghiotti A, Kinsey B, Liewald M, Madej L, Min J, Yanagimoto J (2019) Models and modelling for process limits in metal forming. CIRP Ann 68:775–798. https://doi.org/10.1016/j.cirp.2019.05.007
43. Hu Z Elasto-plastic solutions for spring-back angle of pipe bending using local induction heating
44. Nazari E, Staupendahl D, Löbbe C, Tekkaya A (2019) Bending moment in incremental tube forming. Int J Mater Form 12:113–122. https://doi.org/10.1007/s12289-018-1411-x
45. Arthington M, Cleaver C, Huang J, Duncan S (2016) Curvature control in radial-axial ring rolling. In: IFAC-PapersOnLine, vol 49, pp 244–249. Elsevier B.V.
46. Jackson K, Allwood J (2009) The mechanics of incremental sheet forming. J Mater Process Technol 209:1158–1174. https://doi.org/10.1016/j.jmatprotec.2008.03.025
47. Bambach M (2010) A geometrical model of the kinematics of incremental sheet forming for the prediction of membrane strains and sheet thickness. J Mater Process Technol 210:1562–1573. https://doi.org/10.1016/j.jmatprotec.2010.05.003
48. Chang Z, Li M, Chen J (2019) Analytical modeling and experimental validation of the forming force in several typical incremental sheet forming processes. Int J Mach Tools Manuf 140:62–76. https://doi.org/10.1016/j.ijmachtools.2019.03.003

49. Allwood J, Music O, Raithathna A, Duncan S (2009) Closed-loop feedback control of product properties in flexible metal forming processes with mobile tools. CIRP Ann Manuf Technol 58:287–290. https://doi.org/10.1016/j.cirp.2009.03.065
50. Music O, Allwood J (2012) The use of spatial impulse responses to characterise flexible forming processes with mobile tools. J Mater Process Technol 212:1139–1156. https://doi.org/10.1016/j.jmatprotec.2011.12.018
51. Wiebenga J (2014) Robust design and optimization of forming processes. ISBN 9789077172964
52. Havinga G, Van Den Boogaard A, Dallinger F, Hora P Inline control of a strip bending process in mass production
53. Liewald M, Bergs T, Groche P, Behrens B, Briesenick D, Müller M, Niemietz P, Kubik C, Müller F (2022) Perspectives on data-driven models and its potentials in metal forming and blanking technologies. Prod Eng 16:607–625. https://doi.org/10.1007/s11740-022-01115-0
54. Wang J, Wu X, Thomson P, Flitman A (2000) A neural networks approach to investigating the geometrical influence on wrinkling in sheet metal forming
55. Viswanathan V, Kinsey B, Cao J (2003) Experimental implementation of neural network Springback control for sheet metal forming. J Eng Mater Technol 125:141–147. https://doi.org/10.1115/1.1555652
56. Trzepieciński T, Najm S (2022) Application of artificial neural networks to the analysis of friction behaviour in a drawbead profile in sheet metal forming. Materials 15:9022. https://doi.org/10.3390/ma15249022
57. Kotkunde N, Deole A, Gupta A (2014) Prediction of forming limit diagram for Ti-6Al-4V alloy using artificial neural network. Procedia Mater Sci 6:341–346. https://doi.org/10.1016/j.mspro.2014.07.043
58. Abbassi F, Belhadj T, Mistou S, Zghal A (2013) Parameter identification of a mechanical ductile damage using artificial neural networks in sheet metal forming. Mater Des 45:605–615. https://doi.org/10.1016/j.matdes.2012.09.032
59. Rahimi S, Jackson M, Wynne B (2022) Digital Twins for high-value components. Material-World
60. Tatipala S, Wall J, Johansson C, Larsson T (2020) A hybrid data-based and model-based approach to process monitoring and control in sheet metal forming. Processes 8:89. https://doi.org/10.3390/pr8010089
61. Lim Y, Venugopal R, Ulsoy A (2010) Multi-Input Multi-Output (MIMO) modeling and control for stamping. J Dyn Syst Meas Control Trans ASME 132:1–12. https://doi.org/10.1115/1.4001332
62. Lim Y, Venugopal R, Ulsoy A (2012) Direct and indirect adaptive process control of sheet metal forming
63. Lim Y, Venugopal R, Ulsoy A (2012) Auto-tuning and adaptive control of sheet metal forming. Control Eng Pract 20:156–164. https://doi.org/10.1016/j.conengprac.2011.10.006
64. Bäume T, Zorn W, Drossel W, Rupp G (2016) Iterative process control and sensor evaluation for deep drawing tools with integrated piezoelectric actuators. Manuf Rev (Les Ulis) 3:3. https://doi.org/10.1051/mfreview/2016002
65. Jang I, Bae G, Kim H (2022) Metal forming defect detection method based on recurrence quantification analysis of time-series load signal measured by real-time monitoring system with bolt-type piezoelectric sensor. Mech Syst Signal Process 180:109457. https://doi.org/10.1016/j.ymssp.2022.109457
66. Doege E, Seidel H, Griesbach B, Yun J (2002) Contactless on-line measurement of material flow for closed loop control of deep drawing
67. Simonetto E, Ghiotti A, Brun M, Bruschi S, Filippi S (2023) Adaptive metal flow control in stamping through ferrofluidic actuators. CIRP Ann 72:209–212. https://doi.org/10.1016/j.cirp.2023.03.030
68. Fischer P, Heingärtner J, Duncan S, Hora P (2020) On part-to-part feedback optimal control in deep drawing. J Manuf Process 50:403–411. https://doi.org/10.1016/j.jmapro.2019.10.019

69. Cavone G, Bozza A, Carli R, Dotoli M (2022) MPC-based process control of deep drawing: an Industry 4.0 case study in automotive. IEEE Trans Autom Sci Eng 19. https://doi.org/10.1109/TASE.2022.3177362
70. Barthau M, Liewald M, Christian H (2017) Improved process robustness by using closed loop control in deep drawing applications. J Phys Conf Series, Institute of Physics Publishing 896
71. Ghiotti A, Regazzo P, Bruschi S, Bariani P (2010) Reduction of vibrations in blanking by MR dampers. CIRP Ann Manuf Technol 59:275–278. https://doi.org/10.1016/j.cirp.2010.03.111
72. Huang H, Sang H, Li L, Wang Y, Zhu L, Liu Z (2023) High-accuracy control of variable blank holding force driven by electromagnetics based on pulse width modulation with grading voltage and mode matching. J Mater Process Technol 322:118210. https://doi.org/10.1016/j.jmatprotec.2023.118210
73. Ihlenfeldt S, Tehel R, Reichert W, Kurth R (2023) Characterization of generic interactive digital twin for increased agility in forming. CIRP Ann 72:333–336. https://doi.org/10.1016/j.cirp.2023.04.061
74. Gan L, Li L, Huang H (2022) Digital twin-driven sheet metal forming: modeling and application for stamping considering mold wear. J Manuf Sci Eng 144:121003. https://doi.org/10.1115/1.4054902
75. Link P, Penter L, Rückert U, Klingel L, Verl A, Ihlenfeldt S (2025) Real-time quality prediction and local adjustment of friction with digital twin in sheet metal forming. Robot Comput Integr Manuf 91:102848. https://doi.org/10.1016/j.rcim.2024.102848
76. Mayr S, Gross T, Krenn S, Kunze W, Zehetner C (2024) Digital twin-based predictive maintenance for sheet metal bending. Procedia Comput Sci, Elsevier B.V. 232:504–512
77. Klingel L, Penter L, Mayer P, Ihlenfeldt S, Verl A (2022) Digital twins in deep drawing for virtual tool commissioning and inline parameter optimization. IOP Conf Ser Mater Sci Eng 1238:012072. https://doi.org/10.1088/1757-899x/1238/1/012072
78. Koyama H, Wagoner R, Manabe K (2004) Blank holding force control in panel stamping process using a database and FEM-assisted intelligent press control system. J Mater Process Technol 152:190–196. https://doi.org/10.1016/j.jmatprotec.2004.03.031
79. Dornheim J, Link N, Gumbsch P (2020) Model-free adaptive optimal control of episodic fixed-horizon manufacturing processes using reinforcement learning. Int J Control Autom Syst 18:1593–1604. https://doi.org/10.1007/s12555-019-0120-7
80. Biegel T, Jourdan N, Hernandez C, Cviko A, Metternich J (2022) Deep learning for multivariate statistical in-process control in discrete manufacturing: a case study in a sheet metal forming process. Procedia CIRP 107:422–427
81. Endelt B (2018) Numerical comparison of three different feedback control schemes applied on a forming operation. Appl Mech Mater 885:64–74. https://doi.org/10.4028/www.scientific.net/amm.885.64
82. Ghiotti A, Simonetto E, Bruschi S, Bariani P (2017) Springback measurement in three roll push bending process of hollow structural sections. CIRP Ann Manuf Technol 66:289–292. https://doi.org/10.1016/j.cirp.2017.04.119
83. Simonetto E, Ghiotti A, Bruschi S (2020) In-process measurement of Springback in tube rotary draw bending. https://doi.org/10.1007/s00170-020-06453-w
84. Fu Z, Gong P (2014) The study for stability of closed-loop control system based on multiple-step incremental air-bending forming of sheet metal. Int J Adv Manuf Technol 71:357–364. https://doi.org/10.1007/s00170-013-5289-y
85. Yang M, Liang P, Zhang Y, Fan L, Wang G (2023) Improvement of Springback prediction accuracy applying a new constitutive model considering damage and nonlinear elastic unloading-reloading behaviors. Int J Press Vessels Pip 204:104961. https://doi.org/10.1016/j.ijpvp.2023.104961
86. Molitor D, Arne V, Kubik C, Noemark G, Groche P (2024) Inline closed-loop control of bending angles with machine learning supported Springback compensation. Int J Mater Form 17:8. https://doi.org/10.1007/s12289-023-01802-y
87. Ubhayaratne I, Pereira M, Xiang Y, Rolfe B (2017) Audio signal analysis for tool wear monitoring in sheet metal stamping. Mech Syst Signal Process 85:809–826. https://doi.org/10.1016/j.ymssp.2016.09.014

88. Liewald M, Karadogan C, Lindemann B, Jazdi N, Weyrich M (2018) On the tracking of individual workpieces in hot forging plants. CIRP J Manuf Sci Technol 22:116–120. https://doi.org/10.1016/j.cirpj.2018.04.002
89. Prinz K, Steinboeck A, Muller M, Ettl A, Kugi A (2017) Automatic gauge control under laterally asymmetric rolling conditions combined with feedforward. IEEE Trans Ind Appl 53:2560–2568. https://doi.org/10.1109/TIA.2017.2660458
90. Li X, Schulte C, Abel D, Teller M, Hirt G, Lohmar J (2021) Modeling and exploiting the strip tension influence on surface imprinting during temper rolling of cold-rolled steel. Adv Ind Manuf Eng 3:100045. https://doi.org/10.1016/j.aime.2021.100045
91. Müller M, Prinz K, Steinboeck A, Schausberger F, Kugi A (2020) Adaptive feedforward thickness control in hot strip rolling with oil lubrication. Control Eng Pract 103:104584. https://doi.org/10.1016/j.conengprac.2020.104584
92. Schulte C, Li X, Stemmler S, Vallery H, Hirt G, Abel D (2023) Adaptive pass scheduling for roughness control in cold rolling. IFAC-PapersOnLine 56:2683–2688. https://doi.org/10.1016/j.ifacol.2023.10.1360
93. Ren Y, Dong J, He J, Zhang D, Wu K, Xiong Z, Zheng P, Sun Y, Liu S (2024) A novel six-dimensional digital twin model for data management and its application in roll forming. Adv Eng Inform 61:102555. https://doi.org/10.1016/j.aei.2024.102555
94. Groche P, Zettler A, Berner S, Schneider G (2011) Development and verification of a one-step-model for the design of flexible roll formed parts. Int J Mater Form 4:371–377. https://doi.org/10.1007/s12289-010-0998-3
95. Woo Y, Kang P, Oh I, Moon Y (2018) Flexible roll forming of double layered blank. Procedia Manuf 15:775–781
96. Sheu J, Liang C, Yu C, Hsu W, Lee P (2018) Flexible roll forming of U-section product with curved bending profile using advanced high strength steel. Procedia Manuf 15:782–787
97. Abeyrathna B, Rolfe B, Harrasser J, Sedlmaier A, Ge R, Pan L, Weiss M (2017) Prototyping of automotive components with variable width and depth. J Phys Conf Ser 896
98. Abeyrathna B, Ghanei S, Rolfe B, Taube R, Weiss M (2022) Optimising part quality in the flexible roll forming of an automotive component. Int J Adv Manuf Technol 118:3361–3373. https://doi.org/10.1007/s00170-021-08176-y
99. Groche P, Güngör B, Kilz J (2023) Straight roll formed profiles through partial rolling. CIRP Ann 72:229–232. https://doi.org/10.1016/j.cirp.2023.03.024
100. Kuzman K (2001) Problems of accuracy control in cold forming. J Mater Process Technol 113:10–15
101. Qin Y, Balendra R, Chodnikiewicz K (2000) A method for the simulation of temperature stabilisation in the tools during multi-cycle cold-forging operations
102. Qin Y (2006) Forming-tool design innovation and intelligent tool-structure/system concepts. Int J Mach Tools Manuf 46:1253–1260. https://doi.org/10.1016/j.ijmachtools.2006.01.013
103. Liewald M, Schiemann T, Mletzko C (2014) Automatically controlled (cold-) forging processes. Procedia CIRP 18:39–44
104. Böhm J, Liewald M, Clauß P (2024) Study on scrap reduction in cold forging during ramp-up phases through actuator control. Lecture notes in production engineering. Springer Nature, Part F1764, pp 326–334
105. Rekowski M, Grötzinger K, Schott A, Liewald M (2024) Thin-film sensors for data-driven concentricity prediction in cup backward extrusion. CIRP Ann 73:205–208. https://doi.org/10.1016/j.cirp.2024.04.035
106. Groche P, Heß B (2014) Friction control for accurate cold forged parts. CIRP Ann Manuf Technol 63:285–288. https://doi.org/10.1016/j.cirp.2014.03.012
107. Nielsen C, Arinbjarnar U, Ceron E, Madsen T, Møller B, Madsen K, Siimut K (2023) A novel ironing punch concept with adjustable tool diameter. CIRP Ann 72:213–216. https://doi.org/10.1016/j.cirp.2023.03.001
108. Uribe D, Baudouin C, Durand C, Bigot R (2024) Predictive control for a single-blow cold upsetting using surrogate modeling for a digital twin. Int J Mater Form 17:7. https://doi.org/10.1007/s12289-023-01803-x

109. Budnick D, Steinlehner F, Weinschenk A, Volk W, Melek W, Worswick M, Huhn S (2021) Simulation of dynamic effects in progressive die operation and control. IOP Conf Ser Mater Sci Eng 1157:012085. https://doi.org/10.1088/1757-899x/1157/1/012085
110. Budnick D, Ghannoum A, Steinlehner F, Weinschenk A, Volk W, Huhn S, Melek W, Worswick M (2022) Predicting dynamic process limits in progressive die sheet metal forming. IOP Conf Ser Mater Sci Eng 1238:012068. https://doi.org/10.1088/1757-899x/1238/1/012068
111. Hartmann C, Opritescu D, Volk W (2019) An artificial neural network approach for tool path generation in incremental sheet metal free-forming. J Intell Manuf 30:757–770. https://doi.org/10.1007/s10845-016-1279-x
112. Gondo S, Arai H (2022) Data-driven metal spinning using neural network for obtaining desired dimensions of formed cup. CIRP Ann 71:229–232. https://doi.org/10.1016/j.cirp.2022.04.044
113. Music O, Allwood J (2011) Flexible asymmetric spinning. CIRP Ann Manuf Technol 60:319–322. https://doi.org/10.1016/j.cirp.2011.03.136
114. Russo I, Cleaver C, Allwood J, Loukaides E (2020) The influence of part asymmetry on the achievable forming height in multi-pass spinning. J Mater Process Technol 275:116350. https://doi.org/10.1016/j.jmatprotec.2019.116350
115. Jawale K, Loukaides E (2019) An investigation of mandrel-free spinning. J Phys Conf Ser 29:145–152
116. Loukaides E, Russo I (2017) Toolpath generation for asymmetric mandrel-free spinning. Procedia Eng 207:1707–1712
117. Bambach M, Taleb Araghi B, Hirt G (2009) Strategies to improve the geometric accuracy in asymmetric single point incremental forming. Prod Eng 3:145–156. https://doi.org/10.1007/s11740-009-0150-8
118. de Gooijer B, Havinga J, Geijselaers H, van den Boogaard A (2021) Evaluation of POD based surrogate models of fields resulting from nonlinear FEM simulations. Adv Model Simul Eng Sci 8:25. https://doi.org/10.1186/s40323-021-00210-8
119. Hale M, Hardt D (1987) Dynamic analysis and control of a roll bending process
120. Jia C, Shan Z, Cui Y, Bai T, Cui F (2013) Modeling and simulations of hydraulic roll bending system based on CMAC neural network and PID coupling control strategy. J Iron Steel Res Int 20:17–22
121. Jing Y, Jiang S, Sun Q, Zhao Y, Song Z, Meng X, Li H (2023) Design and development of high precision four roll CNC roll bending machine and automatic control model. Sci Rep 13:12954. https://doi.org/10.1038/s41598-023-40204-7
122. Liu H, Sun Q, Zhao Y, Song Z, Wang J (2023) Design of three-roll bending machine tool and research on compensation algorithm. Adv Mech Eng 15. https://doi.org/10.1177/16878132231196090
123. Cao H, Yu G, Liu T, Fu P, Huang G, Zhao J (2023) Research on the curvature prediction method of profile roll bending based on machine learning. Metals 13:143. https://doi.org/10.3390/met13010143
124. Allwood J, Tekkaya A, Stanistreet T (2005) The development of ring rolling technology. Steel Res Int 76:111–120
125. Jenkouk V, Hirt G, Franzke M, Zhang T (2012) Finite element analysis of the ring rolling process with integrated closed-loop control. CIRP Ann Manuf Technol 61:267–270. https://doi.org/10.1016/j.cirp.2012.03.115
126. Liang L, Guo L, Wang Y, Li X (2019) Towards an intelligent FE simulation for real-time temperature-controlled radial-axial ring rolling process. J Manuf Process 48:1–11. https://doi.org/10.1016/j.jmapro.2019.09.032
127. Liang L, Guo L, Yang J, Zhang H (2022) Formation mechanism and control method of multiple geometric defects in conical-section profiled ring rolling. J Mater Process Technol 306:117628. https://doi.org/10.1016/j.jmatprotec.2022.117628
128. Lafarge R, Hütter S, Tulke M, Halle T, Brosius A (2021) Data based model predictive control for ring rolling. Prod Eng 15:821–831. https://doi.org/10.1007/s11740-021-01063-1
129. Daehn G (2023) Introducing NSF's Hammer Engineering Research Center: hybrid autonomous manufacturing moving from evolution to revolution

130. Ford Motor Company, F3T. https://www.prnewswire.com/news-releases/ford-develops-advanced-technology-to-revolutionize-prototyping-personalization-low-volume-production-214097851.html. Accessed 30 Jan 2024
131. Machia Labs. https://machinalabs.ai. Accessed 30 Jan 2024
132. Desktop Metal. https://figur.desktopmetal.com. Accessed 30 Jan 2024
133. Desktop Metal, Claim. https://www.youtube.com/watch?v=4fYqPxYRTts&t=1s. Accessed 30 Jan 2024

STC M—Machines

Bayesian Inference for Milling Stability Modeling

Jaydeep Karandikar, Tony Schmitz, and Friedrich Bleicher

Abstract This essay describes Bayesian learning for milling stability modeling. A non-model grid-based method and two model-based methods using random sample stability maps and Markov Chain Monte Carlo sampling for Bayesian learning are described. The three methods are compared using experimental results completed on Aluminum 6061-T6 workpiece. A test selection strategy to maximize the material removal rate is presented for the non-model and model-based approaches. The essay also describes recent advances in Bayesian learning and experimentations and provides future research directions and outlook.

Keywords Milling · Stability · Machine learning

1 Introduction

The goal for discrete part milling is to produce accurate parts in the required timeframe at the maximum profit. There are several factors that influence the desired production efficiency and minimum milling cost. These include workpiece loading/unloading from the machine, fixturing, toolpath generation and machining strategy, process parameters, tool wear, tool and workpiece vibrations, coolant management, chip evacuation, and machine accuracy [1]. One persistent challenge for selecting optimal process parameters is that the tool-holder-spindle-machine assembly and

J. Karandikar · T. Schmitz (✉)
Manufacturing Science Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA
e-mail: tony.schmitz@utk.edu

J. Karandikar
e-mail: KARANDIKARJM@ORNL.GOV

T. Schmitz
Machine Tool Research Center, University of Tennessee, Knoxville, Knoxville, TN, USA

F. Bleicher
Institute of Production Engineering and Photonic Technologies, TU Wien, Vienna, Austria
e-mail: bleicher@ift.at

T. Tolio (ed.), *CIRP Novel Topics in Production Engineering: Volume 2*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-3-032-04439-6_3

workpiece collectively represent a dynamic system with finite stiffness. The result is tool tip/workpiece deflections due to the dynamic cutting forces during material removal. The relative vibration between the tool and workpiece leads to regeneration of surface waviness that can cause unstable machining conditions due to self-excited vibration, or chatter. Unstable cutting conditions are avoided because they provide poor surface finish and can cause spindle damage. Due to the poor surface finish, parts may be scrapped or may require rework. In addition, the part accuracy may be reduced and the tool wear rate may be increased [2].

For a selected tool-holder-spindle-machine assembly and workpiece, a stability map defines the stable combinations of spindle speed and axial depth for a given radial depth of cut. The stability map separates stable {spindle speed, axial depth} combinations from unstable combinations using a deterministic stability boundary. Methods for predicting machining stability maps include frequency domain, time domain, and semi-discretization [3, 4].

Inputs to the stability map include the mechanistic force model (e.g., specific cutting force, K_s, and force angle, β) and the tool tip frequency response function, or FRF. The FRF can be represented using modal parameters, including the natural frequency, f_n, stiffness, k, and (dimensionless) viscous damping ratio, ζ. For example, Fig. 1a shows a four flute, 12.7 mm diameter solid carbide tool clamped in a thermal shrink fit tool holder and inserted in a CNC milling machine spindle. Figure 1b displays the corresponding tool tip FRFs, where it is observed that the FRFs are approximately symmetric in the X (feed) and Y directions (both directions are perpendicular to the tool axis, Z). The FRFs are measured by tapping the tool tip with an instrumented hammer and measuring the response with an accelerometer. The modal parameters for the most compliant mode are $f_n = 1998$ Hz, $k = 4.47 \times 10^6$ N/m, and $\zeta = 0.012$, where symmetry is assumed.

Figure 2 shows the stability map for the Fig. 1 setup calculated using the zero-order frequency domain model [5]. Test results are superimposed, where both stable and unstable (chatter) conditions are identified for a grid of {spindle speed, axial depth} pairs. The workpiece material was 6061-T6 aluminum. The values for K_s and

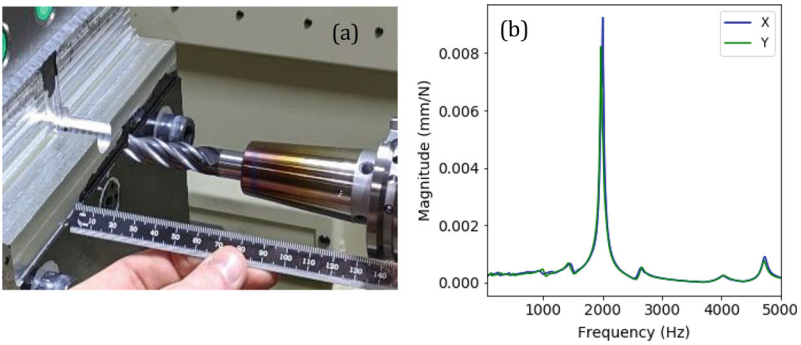


Fig. 1 **a** Experimental setup and **b** tool tip FRF magnitudes in the X and Y directions

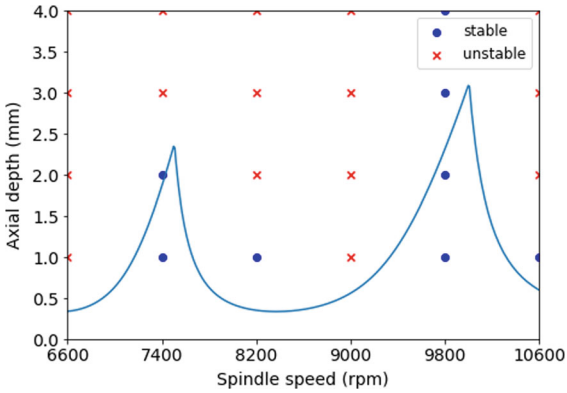


Fig. 2 Stability boundary and experimental results

β were 600 N/mm^2 and 68°, respectively. The radial depth of cut was 5 mm and the feed per tooth was 0.1 mm/tooth. Down milling tests were completed in {800 rpm, 1 mm} increments resulting in a 6 × 4 grid. The sound was recorded during each test using a microphone [6].

The audio frequency content from each test was compared to the tooth passing frequency (i.e., the spindle speed multiplied by the number of flutes) and its harmonics (integer multiples). A test was labeled as stable if significant content appeared only at these frequencies. A test was labeled as unstable if significant frequency content was observed at a different frequency, i.e., the chatter frequency. The significance of a peak was established using the ratio of the chatter frequency magnitude to the largest magnitude of the tooth passing frequency and its harmonics [7]. If this stability ratio was greater than 0.25, the cut was considered unstable.

Figure 3 shows the audio signal frequency spectra at two test points: {6600 rpm, 1.2 mm} and {9800 rpm, 4 mm}. As seen in Fig. 3a, the chatter frequency occurs at 2090 Hz for the unstable {6600 rpm, 1.2 mm} test. The {9800 rpm, 4 mm} test is stable because significant content is only observed at the tooth passing frequency (653.3 Hz) and its first harmonic. While a peak is observed at 2000 Hz for the {9800 rpm, 4 mm} test, its magnitude is much smaller than the first harmonic of the tooth passing frequency at 1306 Hz. As seen in Fig. 2, the test results nominally agree with the predicted stability boundary (blue curve). However, discrepancies between the prediction and test results exist. Specifically, stable results are seen within the predicted unstable zone at {8200 rpm, 1 mm}, {9800 rpm, 3 mm}, {9800 rpm, 4 mm}, and {10,600 rpm, 1 mm}.

While physics-based, deterministic models are valuable and used to significant advantage in both laboratory and production environments, their inputs include measurements and, therefore, uncertainty. In addition to measurement uncertainty, other uncertainty contributors include model assumptions, random noise, and unknown factors. The outcome is that model outputs also include uncertainty [8]. For the stability map, this means that the Fig. 2 deterministic boundary between stable and unstable axial depths at each spindle speed is better described as a distribution of potential axial depths that define the boundary, one of which provides the true

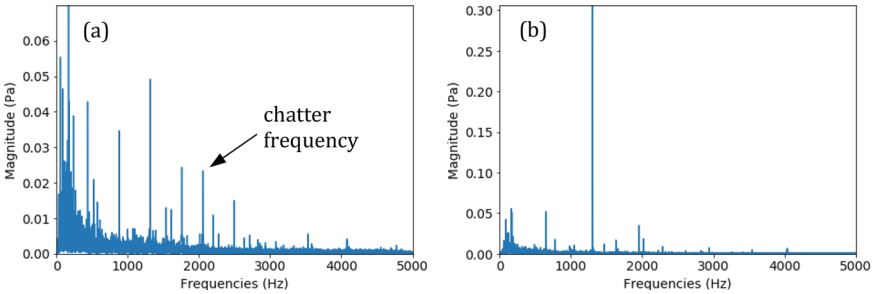


Fig. 3 **a** Sound pressure frequency spectrum for the unstable cut at {6600 rpm, 1.2 mm}; the chatter frequency is 2090 Hz. **b** Sound pressure frequency spectrum for a stable cut at {9800 rpm, 4 mm}

separation between stable and unstable performance. When the probability of an outcome and not just the deterministic solution is modeled, a predictive model is provided and improved decision making is possible [9].

Because a predictive model is based on a probability law, it is aligned with machine learning, which applies statistical algorithms to perform tasks without explicit instructions [10]. The value of machine learning is that initial data is used to establish the model and then new data is used to improve the model. The opportunity to collect new data and improve a predictive model is advantageous because it enables more accurate predictions and, therefore, improved parameter selection for optimized performance. The challenge in manufacturing is that data is generally expensive and time consuming to obtain. Stability testing, for example, incurs costs due to the tool, holder, work material, machine time, instrumentation, and machinist/engineer time. The best machine learning options for manufacturing, therefore, can leverage existing deterministic models and then improve prediction accuracy using limited data. In the following section, Bayesian inference is described, which provides a probabilistic machine learning approach that meets these criteria.

2 Bayesian Inference

Bayesian inference is a method for updating conditional probabilities when new information is made available. Bayes' rule is given by Eq. (1) [9], where:

- $p(A)$ is the prior probability, or initial beliefs, of an uncertain result A
- $p(B|A)$ is the likelihood of observing result B given result A has occurred
- $p(B)$ is the probability of observing result B.

Using Bayes' rule, the posterior conditional probability of result A given result B has occurred, $p(A|B)$, is calculated. This posterior represents new beliefs after testing is completed and the new information is incorporated.

$$p(A|B) = \frac{p(A)p(B|A)}{p(B)} \tag{1}$$

For milling, Bayes' rule can be used to update the probability of stability (or distribution of potential stability boundaries) at a given set of process parameters using test data. To demonstrate the approach, consider a stable test result at a certain combination of spindle speed, Ω_1, and axial depth, b_1. The result may be labeled as stable or unstable using a threshold for the chatter frequency magnitude [7] (as described in Fig. 3), surface finish, the repetition in once-per-revolution samples [11–13], or a combination of these. The probability that $\{\Omega_1, b_1\}$ is stable given a stable experimental result at $\{\Omega_1, b_1\}$ is 1. This is because no uncertainty is assigned to the stability result determination (although this is not explicitly required). Next, consider a different spindle speed and axial depth combination, $\{\Omega_2, b_2\}$. Using Bayes' rule, the probability of stability at this {spindle speed, axial depth} can be updated using the test result at $\{\Omega_1, b_1\}$.

Using Bayes' rule to update the probability of stability offers three main benefits. First, a test at any {spindle speed, axial depth} combination updates the probability of stability at all other combinations. This is a primary differentiator of Bayes' learning from traditional machine learning algorithms, such as neural networks and random forest. The traditional methods require many tests to learn the stability boundary. In an industrial environment with many tool-holder-spindle-machine combinations, completing stability tests is costly and time consuming and, therefore, limited.

Second, uncertainty in the stability model can inform the prior. This is completed by propagating uncertainty in the inputs through the deterministic stability model using Monte Carlo simulation [14]. The prior can also include user beliefs about the stability map for a given tool-holder-spindle-machine assembly. These beliefs can be based on experience or knowledge about the setup. As a result, Bayes' learning can be used even without a stability model by relying on the user's beliefs to generate the prior.

Third, modeling the stability map as a probability distribution (with a probability of stability at each {spindle speed, axial depth} combination) enables intelligent sampling of test parameters and faster convergence to an optimal condition. The goal of the stability tests could be the identification of stable process parameters that maximize material removal rate, *MRR*, or accurate identification of the entire stability map to separate all stable and unstable {spindle speed, axial depth} pairs. Based on the test goal, probabilistic characterization of the stability map using Bayesian inference (learning) enables a value to be placed on the information from a test before performing it; this is called the value of information [15, 16]. As a result, parameters are selected to maximize the value of information from each test.

3 Bayesian Inference for Milling Stability

Bayesian inference methods for stability updating can be categorized as non-model-based and model-based. In model-based methods, a physics-based deterministic model is used to establish the prior and update the probability of stability map. In non-model-based approaches, a physics-based stability model is not used. Instead, knowledge of the stability behavior is applied within the Bayesian framework to update the probability of stability map. Both methods are described in the following sections.

3.1 Non-model-Based Grid Method

In the non-model-based grid method, the {spindle speed, axial depth} domain is divided into equally spaced grid points with indices i (spindle speed) and j (axial depth). A test result at any point is used to update the probability of stability at all other points using Bayes' rule. Let G be any grid point in the domain. The spindle speed and axial depth at G are then Ω_i and b_j. Since G denotes any arbitrary grid point in the {spindle speed, axial depth} domain, the spindle speed and axial depth at G are denoted with indices i and j. Let T denote the grid point at which a stability test is completed. The corresponding spindle speed and axial depth are $_T$ and b_T. Equation (2) gives Bayes' rule for updating the probability of stability at G given a stable result at T, where:

- $p(s_G|+_T)$ is the posterior (updated) probability of stability (s) at grid point G given a stable (+) result at T
- $p(s_G)$ is the prior probability of G being stable
- $p(+_T|s_G)$ is the likelihood probability of observing a stable result at T given G is stable
- $p(+_T)$ is the probability of observing a stable result at T.

The $p(+_T)$ result is calculated using the law of total probability as shown in Eq. (3), where $p(u_G)$ is the prior probability of G being unstable (u) and $p(+_T|u_G)$ is the likelihood probability of a stable result at T given G is unstable.

$$p(s_G|+_T) = \frac{p(+_T|s_G)p(s_G)}{p(+_T)} \tag{2}$$

$$p(+_T) = p(+_T|s_G)p(s_G) + p(+_T|u_G)p(u_G) \tag{3}$$

The probabilities of a grid point being stable or unstable sum to 1 (i.e., $p(s_G) + p(u_G) = 1$). Therefore, the posterior probability of G being unstable is determined by subtracting $p(s_G|+_T)$ from 1. The denominator $p(+_T)$ in Eq. (2) is different at each grid point as defined in Eq. (3).

The first step of the Bayesian inference process is to determine the prior probability of stability. As noted, the prior represents the initial beliefs about the probability of stability. In the non-model-based approach, the prior probability of stability can be established, for example, using only the knowledge that it is more likely to observe unstable milling conditions at higher axial depths for a given spindle speed than it is at lower axial depths for the same spindle speed. For very small axial depths of cut (e.g., 0.1 mm), the cut is likely to be stable at all spindle speeds. As the axial depth is increased, however, the probability of stability reduces.

To calculate the posterior probability of stability at each grid point, the likelihood probabilities, $p(+_T|s_G)$ and $p(+_T|u_G)$, need to be determined. In the non-model-based grid approach, general knowledge about milling stability can be used to calculate the likelihood probabilities.

The following knowledge about milling stability may be applied. First, a stable result at a selected axial depth implies that all axial depths smaller than the selected depth at the same spindle speed are also stable. Similarly, an unstable result implies that all axial depths larger than the selected depth at the same spindle speed are unstable. Note that these assumptions neglect the special case of closed stability islands that can exist in low radial immersion milling [17, 18]. Second, there exists a critical axial depth, below which stable results are obtained at all spindle speeds. As a result, the width of the stable zones in the horizontal (spindle speed) direction increases at smaller axial depths.

This information was used to encode the influence of the test result at T on all grid points in the {spindle speed, axial depth} domain. Note that the influence of the test result at T reduces with increasing Euclidean distance from T. This implies that the prior probabilities at grid points beyond a certain distance from T will be unchanged. In other words, the posterior probability of stability is equal to the prior probability of stability at these remote points.

Consider a stable test result at T. As noted, based on the milling stability knowledge, the influence of the stable result at T does not propagate uniformly away from T in the {spindle speed, axial depth} domain. For example, a stable result will have a larger influence at the test spindle speed for axial depths less than the test axial depth because these points are assumed to be stable as well. Also, since the width of the stable zones increases at smaller axial depths, the smaller the axial depth of cut, the larger the influence of the stable result in the spindle speed direction. At axial depths larger than the test axial depth, the influence of the test result reduces as the distance from the test axial depth increases. The variation in influence with domain location is illustrated in Fig. 4.

The influence of a test result can be modeled using Gaussian probability density functions. Standard deviations in the axial depth, b, and spindle speed, Ω, directions are selected to establish the test result influence: (1) σ_b defines the test result influence in the axial depth direction (vertical in the probability of stability map); and (2) σ defines the test result influence in the spindle speed direction (horizontal). Due to the asymmetric nature of the test result influence in the axial depth direction, σ is described as a function of the axial depth. The value at the test axial depth of cut, b_T,

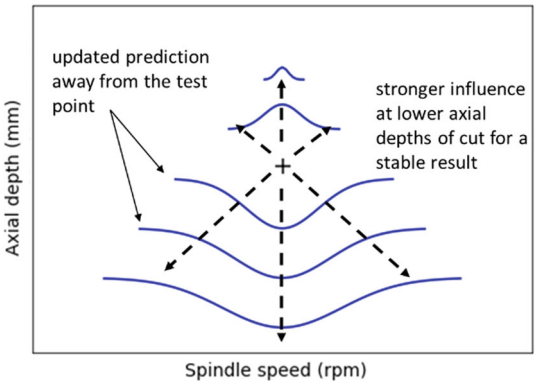


Fig. 4 Ripple effect influence of a stable result (+). A stable result implies that all axial depths below the test axial depth are also stable, resulting in a strong influence at the test spindle speed. The height of each blue curve indicates the local influence, where a larger height (up or down) means a stronger influence

is $\sigma_{\Omega_{b_T}}$. The distance beyond which the posterior probabilities are equal to the prior probabilities is influenced by σ_b and σ_Ω.

The procedure to calculate the posterior probability at all grid points given a stable test result is described here. First, the standard deviations that define the test result influence in the axial depth and spindle speed directions, σ_b and $\sigma_{\Omega_{b_T}}$ are selected. Second, σ is calculated for the stable test as a function of axial depth using Eq. (4). At larger axial depths, the σ_Ω value reduces linearly from $\sigma_{\Omega_{b_T}}$ at $b_j = b_T$ to 0 at $b_j = b_T + 3\sigma_b$. At axial depths smaller than b_T, σ_Ω increases at the same rate.

$$\sigma_{\Omega_{b_j}} = -\frac{\sigma_{\Omega_{b_T}}}{3\sigma_b} b_j + \frac{\sigma_{\Omega_{b_T}}(b_T + 3\sigma_b)}{3\sigma_b}, \text{ if stable } (+) \tag{4}$$

Third, the likelihood probabilities $p(+_T|s_G)$ and $p(+_T|u_G)$ are calculated at the test spindle speed and all axial depths using Eqs. (5) and (6).

$$p(+_T|s_G)_{\Omega_T, b_j} = 1 \tag{5}$$

$$p(+_T|u_G)_{\Omega_T, b_j} = \begin{cases} 0, & b_j \le b_T \\ e^{-0.5\left(\frac{(b_j - (b_T + 3\sigma_b))}{\sigma_b}\right)^2}, & b_T < b_j \le b_T + 3\sigma_b \\ 1, & b_j > b_T + 3\sigma_b \end{cases} \tag{6}$$

Fourth, likelihood probabilities $p(+_T|s_G)$ and $p(+_T|u_G)$ are calculated at other spindle speeds. Since the influence of a stable result is higher at axial depths smaller than the test axial depth, the calculation of $p(+_T|s_G)$ and $p(+_T|u_G)$ in the spindle speed direction is completed at two levels: (1) $b_j \le b_T$; and (2) $b_j > b_T$. The likelihood probabilities for $b_j \le b_T$ as a function of spindle speed are provided in Eqs. (7) and (8).

$$p(+_T|s_G)_{\Omega_i,b_j\le b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_{\Omega_{b_T}}}\right)^2}}{2} \tag{7}$$

$$p(+_T|u_G)_{\Omega_i,b_j\le b_T} = 1 - p(+_T|s_G)_{\Omega_i,b_T} \tag{8}$$

The likelihood probabilities are calculated for $b_j > b_T$ using Eqs. (9) and (10). Note that from Eq. (6), the value of $p(+_T|u_G)_{\Omega_T,b_j}$ increases from 0 to 1 in the interval $[b_T, b_T + 3\sigma_b]$. As a result, the $p(+_T|u_G)_{\Omega_i,b_j>b_T}$ needs to be normalized by the value at the test spindle speed and corresponding axial depth of cut to get the same value [19].

$$p(+_T|s_G)_{\Omega_i,b_j>b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_\Omega(b_j)}\right)^2}}{2} \tag{9}$$

$$p(+_T|u_G)_{\Omega_i,b_j>b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_\Omega(b_j)}\right)^2}}{\left(\frac{1}{p(+_T|u_G)_{\Omega_T,b_j}-0.5}\right)} \tag{10}$$

The likelihood probabilities for an unstable result are calculated using Eqs. (11)–(17). Note that for an unstable test result, the influence is higher at axial depth larger than the test axial depth. In other words, if a test cut is unstable, axial depths above the test value are also unstable for the same spindle speed.

$$\sigma_{\Omega_{b_j}} = \frac{\sigma_{\Omega_{b_T}}}{3\sigma_b} b_j - \frac{\sigma_{\Omega_{b_T}}(b_T - 3\sigma_b)}{3\sigma_b}, \text{ if unstable}(-) \tag{11}$$

$$p(-_T|u_G)_{\Omega_T,b_j} = 1 \tag{12}$$

$$p(-_T|s_G)_{\Omega_T,b_j} = \begin{cases} 1, & b_j < b_T - 3\sigma_b \\ e^{-0.5\left(\frac{(b_j-(b_T-3\sigma_b))}{\sigma_b}\right)^2}, & b_T - 3\sigma_b \le b_j < b_T \\ 0, & b_j \ge b_T \end{cases} \tag{13}$$

$$p(-_T|u_G)_{\Omega_i,b_j\ge b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_{\Omega_{b_j}}}\right)^2}}{2} \tag{14}$$

$$p(-_T|s_G)_{\Omega_i,b_j\ge b_T} = 1 - p(+_T|u_G)_{\Omega_i,b_T} \tag{15}$$

$$p(-_T|u_G)_{\Omega_i,b_j<b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_{\Omega_{b_j}}}\right)^2}}{2} \tag{16}$$

$$p(-_T|s_G)_{\Omega_i,b_j<b_T} = 0.5 + \frac{e^{-0.5\left(\frac{(\Omega_i-\Omega_T)}{\sigma_{\Omega_{b_j}}}\right)^2}}{\left(\frac{1}{p(-_T|s_G)_{\Omega_T,b_j}-0.5}\right)} \tag{17}$$

Algorithm 1 provides the pseudo-code for calculating the posterior probability of stability at G given a test result (stable or unstable) using the non-model-based grid method. For multiple test results, the updated posterior probability of stability at G after the first test result is the prior probability of stability at G for the second test result, and so on.

Algorithm 1 Non-model-based grid method Bayesian inference pseudo-code

Input: Prior probability of stability at grid point G with axial depth b_j, spindle speed Ω_i, standard deviation for axial depth σ_b, standard deviation for spindle speed at test axial depth $\sigma_{\Omega_{b_T}}$, test axial depth b_T, test spindle speed Ω_T, and test result (+ for stable and – for unstable)

Output: Posterior probability of stability at grid point G given test result

1. **if** test result = +, **then**
2. Calculate $\sigma_{\Omega_{b_j}}$ using Eq. 4
3. Calculate $p(+_T|s_G)_{\Omega_T,b_j}$ and $p(+_T|u_G)_{\Omega_T,b_j}$ using Eq. 5 and Eq. 6
4. **if** $b_j \leq b_T$, **then**
5. Calculate $p(+_T|s_G)_{\Omega_i,b_j}$ and $p(+_T|u_G)_{\Omega_i,b_j}$ using Eqs. 7 and 8, respectively
6. **elseif** $b_j > b_T$
7. Calculate $p(+_T|s_G)_{\Omega_i,b_j}$ and $p(+_T|u_G)_{\Omega_i,b_j}$ using Eqs. 9 and 10, respectively
8. **elseif** test result = –, **then**
9. Calculate $\sigma_{\Omega_{b_j}}$ using Eq. 11
10. Calculate $p(-_T|u_G)_{\Omega_T,b_j}$ and $p(-_T|s_G)_{\Omega_T,b_j}$ using Eqs. 12 and 13, respectively
11. **if** $b_j \geq b_T$, **then**
12. Calculate $p(-_T|u_G)_{\Omega_i,b_j}$ and $p(-_T|s_G)_{\Omega_i,b_j}$ using Eqs. 14 and 15, respectively
13. **elseif** $b_j < b_T$, **then**
14. Calculate $p(-_T|u_G)_{\Omega_i,b_j}$ and $p(-_T|s_G)_{\Omega_i,b_j}$ using Eqs. 16 and 17, respectively.
15. Calculate $p(u_G) = 1 - p(s_G)$
16. Substitute in Eq. 3 and Eq. 2 to calculate posterior probability of stability at G with axial depth b_j and spindle speed Ω_i
17. Set the prior probabilities equal to the posterior probabilities for the next update

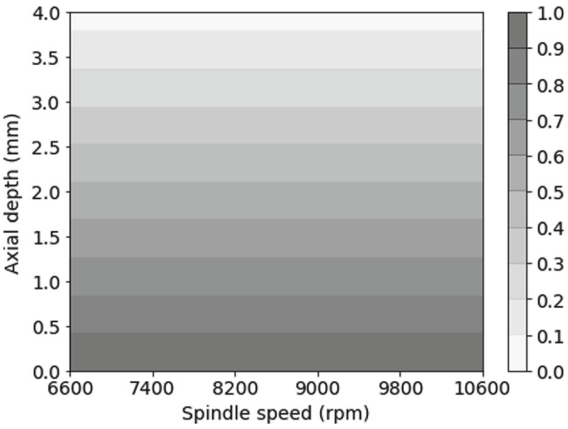


Fig. 5 Prior probability of stability

The test results shown in Fig. 2 were used to demonstrate the method. The spindle speed range was 6600 rpm to 10,600 rpm and axial depth range was 0–4 mm. The {spindle speed, axial depth} domain was discretized into a rectangular grid of points with 10 rpm and 0.1 mm spacing. This resulted in 16,000 (400 × 40) grid points. The prior probability of stability at each grid point was defined using the knowledge that it is more likely to observe an unstable test result at high axial depth values. The probability of stability was modelled as linearly decreasing from 1 at 0 mm to 0.05 at 4 mm for all spindle speeds. This is displayed in Fig. 5.

Figure 6 shows the progression of the posterior probability of stability after tests at six spindle speeds from 6600 to 10,600 rpm with 1 mm increments in axial depth. For comparison, the zero-order frequency domain stability boundary (shown in Fig. 2) is also displayed in Fig. 6. The $\sigma_{\Omega_{b_T}}$ and σ_b values were 120 rpm and 0.4 mm, respectively. Note that in the updating procedure, an unstable result at {6600 rpm, 1 mm} gives a posterior probability of 0 at all axial depths of cut larger than 1 mm at 6600 rpm. As a result, additional unstable results at {2, 3, and 4} mm axial depths at 6600 rpm do not change the posterior probability of stability and are not included in Fig. 6. As seen in Fig. 6, each experimental result updates the probability of stability, where the influence of a stability result is governed by the choice of $\sigma_{\Omega_{b_T}}$ and σ_b. In general, small values of $\sigma_{\Omega_{b_T}}$ and σ_b weaken the influence of the test result and represent a conservative choice, which will increase the number of test results required to converge to the true (unknown) stability boundary. Based on numerous tests completed by the authors, a value of 3% of the spindle speed range for $\sigma_{\Omega_{b_T}}$ and 10% of the axial depth range for σ_b is a suitable choice for most cases.

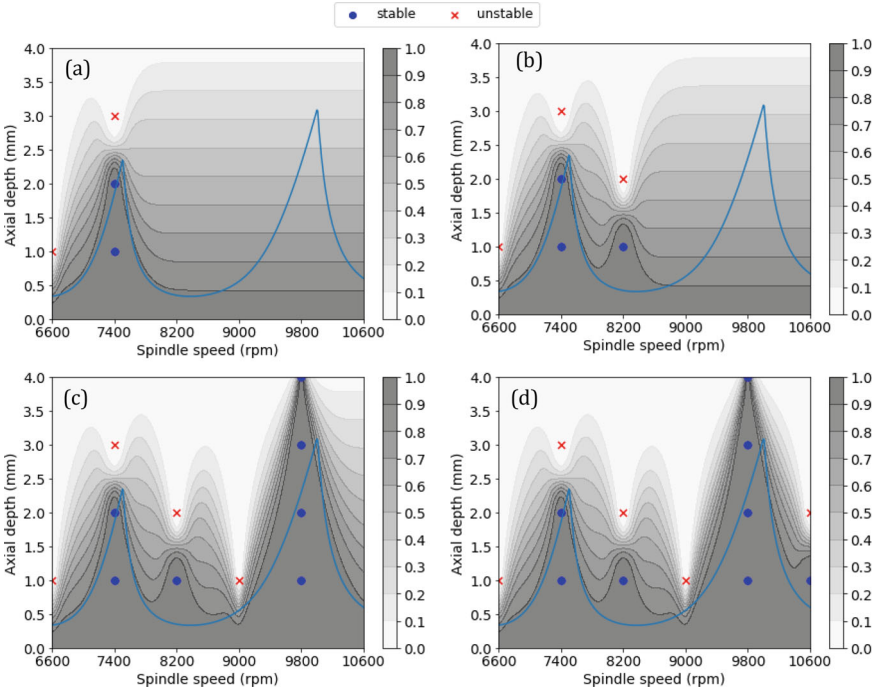


Fig. 6 Posterior probability of stability after **a** four tests; **b** six tests; **c** 11 tests; and **d** 13 tests. The stability boundary from Fig. 2 is also shown for comparison

3.2 Model-Based Random Sample Stability Map Method

The random sample path method for stability updating proceeds as follows. First, prior distributions for the stability model [1] inputs (force model and the tool tip frequency response function, FRF, or modal parameters that represent the FRF) are selected. The prior distributions are sampled and the stability map is calculated for each input sample. This is repeated many times using a Monte Carlo approach [12]. Before any test results are available, each prior input sample, and the associated stability map, is assumed to be equally likely to be the true value. Therefore, if N prior samples of the input parameters and corresponding stability maps are generated, each is assumed to be the true value with a probability of $1/N$. After a test result is made available, the probability of each input sample (and the associated stability map) is updated using Bayes' rule.

Let α denote a stability result. Let $\theta = \{K_s, \beta, f_n, k, \zeta\}$ be a vector containing values for the cutting force model, K_s and β, and modal parameters, f_n, k, and ζ. The probability of the nth sample being the true value given the test result, α, is defined by Eq. (18).

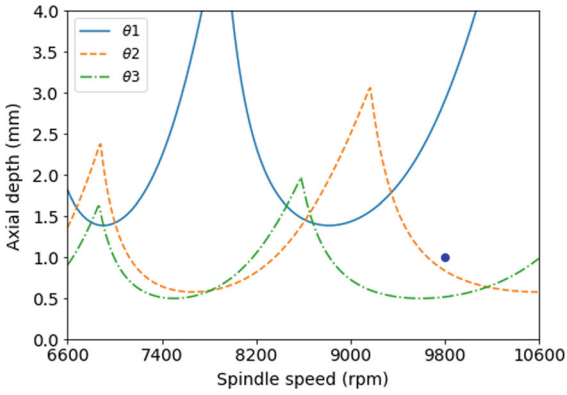


Fig. 7 Three candidate stability maps with different input parameters

$$p(\theta_n|\alpha) = \frac{p(\theta_n)p(\alpha|\theta_n)}{p(\alpha)} \tag{18}$$

In Eq. (18), $p(\theta_n|\alpha)$ is the posterior probability of the input sample θ_n given test result α, $p(\alpha|\theta_n)$ is the likelihood of observing the result α given input parameters θ_n, $p(\theta_n)$ is the prior probability of the stability input sample θ_n, and $p(\alpha)$ is the probability of observing the result α. The denominator in Eq. (18), $p(\alpha)$, is a normalizing constant and ensures the sum of all prior generated input samples is 1.

The likelihood function is defined as follows. Consider the three stability maps based on three input parameter sets, θ_1, θ_2, and θ_3, shown in Fig. 7. A stable result at {9800 rpm, 1 mm} shown as a blue dot. The stability map associated with θ_1 (blue solid line in Fig. 7) predicts the stable result because the limiting axial depth at 9800 rpm is 3.11 mm. As a result, the likelihood of observing the stability result given the stability map $p(\alpha|\theta_1)$ is 1. Stability maps associated with θ_2 (orange dashed line), and θ_3 (green dash-dot line) predict an unstable result at 9800 rpm because the limiting axial depths are 0.83 mm and 0.51 mm, respectively.

As noted, there exists some uncertainty in the stability boundary due to measurement uncertainty for input values, model assumptions, random noise, and unknown factors. This implies that stable results may occur slightly above the stability boundary and unstable results may occur below the boundary. This was observed in Fig. 2. As a result, the likelihood for stability maps that do not agree with the experimental result is not assigned as 0. In Fig. 7, the stability map with θ_2 input parameters (orange dashed line) could still be the true map. Therefore, a likelihood function is defined which assigns a smaller, but non-zero, likelihood value as the distance between the test result and the stability map limiting axial depth of cut prediction increases. In Fig. 7, this implies that the stability map with θ_2 has a higher likelihood of being the true map than the stability map with θ_3. The likelihood function is defined here using a Gaussian function with axial depth uncertainty, σ_b. Other functions, such as linear or sigmoid, may also be selected. For a stable result, the likelihood function is given by Eq. (19).

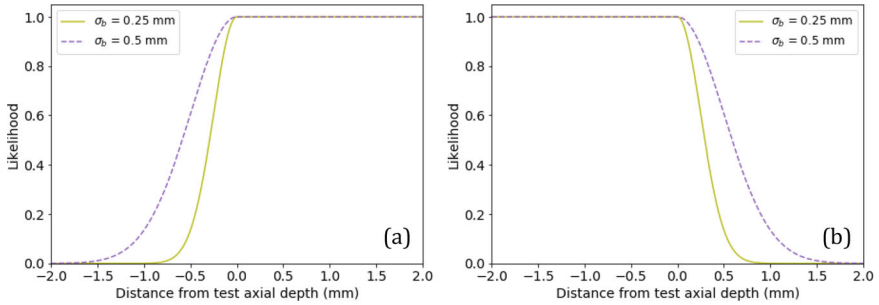


Fig. 8 **a** Likelihood for stable test result and **b** likelihood for unstable result with $\sigma_b = 0.25$ mm (yellow solid line) and $\sigma_b = 0.5$ mm (purple dashed line)

$$p(\alpha : +|\theta_n) = \begin{cases} e^{\frac{-(b_j - b_T)^2}{2\sigma_b^2}} & b_j < b_T \\ 1 & b_j \geq b_T \end{cases} \tag{19}$$

For an unstable result, the likelihood is shown in Eq. (20).

$$p(\alpha : -|\theta_n) = \begin{cases} 1 & b_j \leq b_T \\ e^{\frac{-(b_j - b_T)^2}{2\sigma_b^2}} & b_j > b_T \end{cases} \tag{20}$$

Figure 8 shows the likelihood function for a stable test result (Fig. 8a) and an unstable result (Fig. 8b) with $\sigma_b = 0.25$ mm (yellow solid line) and $\sigma_b = 0.5$ mm (purple dashed line). The likelihood probabilities are plotted as a function of the distance from the test axial depth of cut. The value of σ_b influences the level of disagreement between the test result and the stability map. A larger value of σ_b means that increased disagreement is allowed.

Since $\int_{\theta} p(\theta|\alpha)d\theta = 1$, the normalizing constant $p(\alpha)$ is calculated as the sum of probabilities for all samples as shown in Eq. (21).

$$p(\alpha) = \sum_{n=1}^{N} p(\theta_n|\alpha) \tag{21}$$

As noted, the posterior probabilities of each sample after the first update become the prior for the second update, and so on. The likelihood function also enables simultaneous updates for multiple test results using a product of the likelihood function values for each individual result. Given M tests, the posterior is calculated as shown in Eq. (22).

$$p(\theta_n|\alpha_{1:M}) = p(\theta_n) \prod_{m=1}^{M} p(\alpha_m|\theta_n) \tag{22}$$

The pseudo-code for updating the probability of each stability input sample (and the associated stability map) for a given test result is given in Algorithm 2. The updated posterior probabilities for sample paths are used to calculate the probability of stability at any selected {spindle speed, axial depth} combination, $\{\Omega_i, b_j\}$. The posterior probability at $\{\Omega_i, b_j\}$ is given by the sum of the posterior probabilities where the limiting axial depths of cut at Ω_i from the stability map samples is less than b_j.

Algorithm 2 Model-based random sample stability map Bayesian updating

Input: N stability maps with limiting axial depth values for each entry in Ω_i vector, spindle speed, vector of prior probability of stability maps $p(\theta_{1:N})$, test axial depth b_T, test spindle speed Ω_T, standard deviation for axial depth σ_b, and test result (+ for stable and – for unstable)

Output: Posterior probability of stability maps

1. $n = 1$
2. **while** $n \leq N$, do
3. **if** test result = +, **then**
4. Calculate $p(\alpha: +|\theta_n)$ using Eq. 19
5. **elseif** test result =–, **then**
6. Calculate $p(\alpha: -|\theta_n)$ using Eq. 20
7. Calculate $p(\theta_n|\alpha)$ using Eq. 18; the denominator value; $p(\alpha)$ is not yet calculated and ignored at this step
8. $n = n + 1$
9. Calculate normalizing constant $p(\alpha)$ using Eq. 21
10. Calculate the normalized posterior probabilities $\frac{p(\theta_n|\alpha)}{p(\alpha)}$
11. Set prior probabilities equal to the posterior probabilities for the next update

The test results shown in Fig. 2 were used to demonstrate the Bayesian updating method using the model-based random sample path method. The first step is to generate the prior by sampling from distributions of input variables and calculating the deterministic stability maps using a Monte Carlo approach. Consider a scenario where information on the FRF is not available. In this case, the prior distribution of stability input parameters, $\theta = \{K_s, \beta, f_n, k, \zeta\}$, can be selected using prior knowledge or experience. The prior distributions were selected as: $K_s = \text{U}(500, 800)$ N/mm^2 and $\beta = \text{U}(60, 75)^\circ$ for an aluminum alloy workpiece and $f_n = \text{U}(500, 2500)$ Hz, $k = \text{U}(1.0 \times 10^6, 1.5 \times 10^7)$ N/m, $\zeta = \text{U}(0.01, 0.03)$ for typical single mode tool tip FRFs, where U denotes a uniform distribution and the values in parentheses denote the minimum and maximum values for each variable. A uniform distribution implies that the true value of the input parameter is equally likely to be anywhere between the minimum and maximum values.

Recall that the spindle speed range was 6600–10,600 rpm and the axial depth range was 0–4 mm in Fig. 2. The {spindle speed, axial depth} domain was discretized into a rectangular grid of points with 10 rpm spindle speed and 0.1 mm axial depth spacing. The procedure to calculate the prior probability of stability at each grid point is as follows. First, N random samples are drawn from the stability map input parameter distributions. Note that the correlation between the input parameters is taken as zero

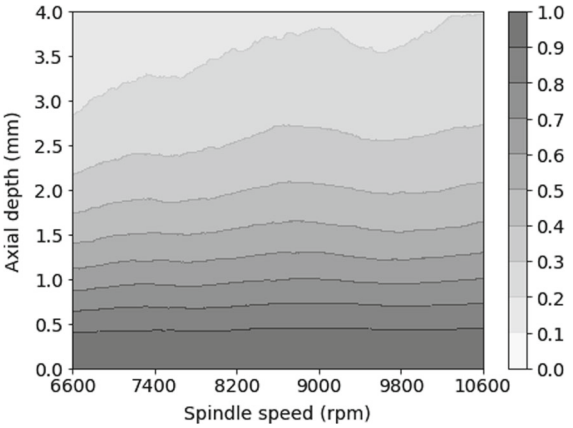


Fig. 9 Prior probability of stability using 1×10^4 stability maps generated by sampling from the prior distribution of input parameters

here, but this is not a strict requirement. Second, the stability map is calculated for each random sample of input parameters. Third, the probability of stability at each grid point, $\{\Omega_i, b_j\}$, is assigned the ratio of stability maps where the limiting axial depths of cut at $_i$ is less than b_j and the total number of stability maps N. Figure 9 shows the prior probability of stability using 1×10^4 stability maps generated by sampling from the prior distribution of stability input parameters. Since the prior distribution of input parameters is wide, the variation in stability with spindle speed is not evident in the prior (i.e., the prior resembles the non-model-based prior shown in Fig. 5 with a general decrease in the probability of stability as axial depth increases).

Figure 10 shows the progression of the posterior probability of stability after tests at the same spindle speeds as Fig. 6, where the σ_b value was 0.25 mm. The analytical stability map (Fig. 2) is also included for comparison. As stated, the posterior probability at $\{\Omega_i, b_j\}$ is given by the sum of the posterior probabilities where the limiting axial depths of cut at Ω_i from the stability map samples is less than b_j.

To establish the prior for Fig. 10, each stability map had a set of input parameters, $\theta = \{K_s, \beta, f_n, k, \zeta\}$. The posterior probability of stability can therefore be used to calculate the posterior distribution of input parameters. Equations (23) and (24) show the calculations for the mean and standard deviation of K_s, $\mu(K_s)$ and $\sigma(K_s)$, respectively. In these equations, p_n and K_{s_n} are the probability and the K_s value for sample n, respectively. Figure 11 displays the posterior distributions of the five input parameters after 13 tests with 1×10^4 samples. Note that in Fig. 11, the vertical axis is the discrete probability of each bin (the sum of all is equal to one). The posterior distribution for f_n converged to the measured value of 1998 Hz (Fig. 1b). There is some uncertainty for other modal parameters due to disagreements between the zero-order frequency domain prediction and test results as shown in Fig. 2.

$$\mu(K_s) = \sum_{n=1}^{N} p_n K_{s_n} \tag{23}$$

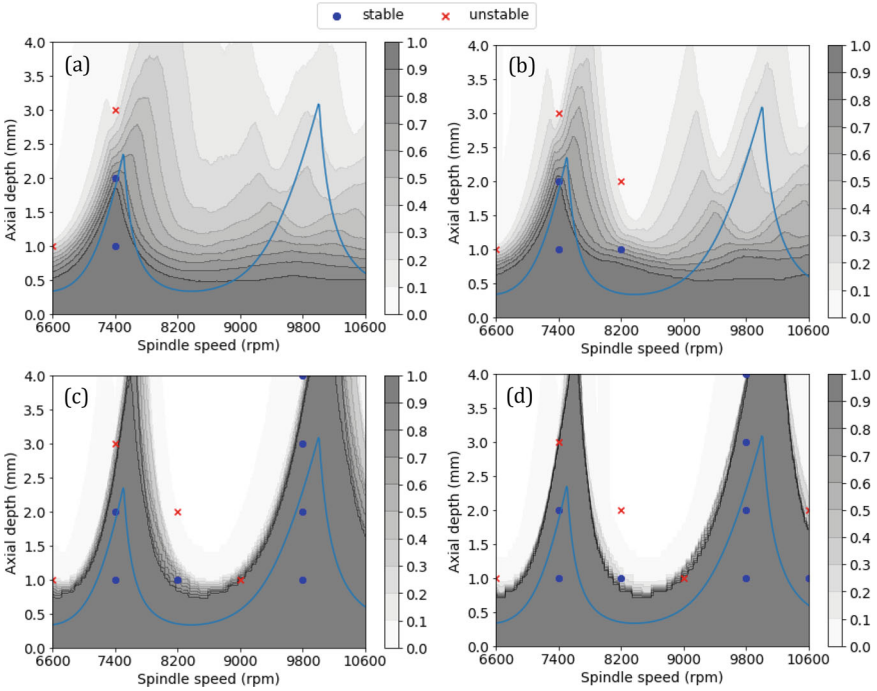


Fig. 10 Posterior probability of stability after **a** four tests; **b** six tests; **c** 11 tests; and **d** 13 tests. The stability map from Fig. 2 is also shown for comparison

$$\sigma(K_s) = \sum_{n=1}^{N} p_n \left(K_{s_n} - \mu(K_s)\right)^2 \tag{24}$$

3.3 *Model-Based Markov Chain Monte Carlo Updating*

The Markov Chain Monte Carlo (MCMC) method is a sampling strategy to draw samples from a known distribution [20]. Let the variable of interest be denoted as x, where $x \in X \subseteq \mathbb{R}$ and X is a real-valued input space. The distribution of interest is the target probability density function (pdf), $p(x)$. The MCMC method enables the generation of N samples from the target distribution using a Markov chain mechanism. The Metropolis Hastings (MH) algorithm is the most widely used MCMC method, where a candidate sample, x_c, is proposed from a proposal pdf, $q(x)$. The candidate sample is selected conditioned on the current value of sample x according to $q(x_c|x_n)$ where x_n is the nth sample. The candidate sample is either accepted or rejected depending on an acceptance ratio, A. The acceptance ratio is

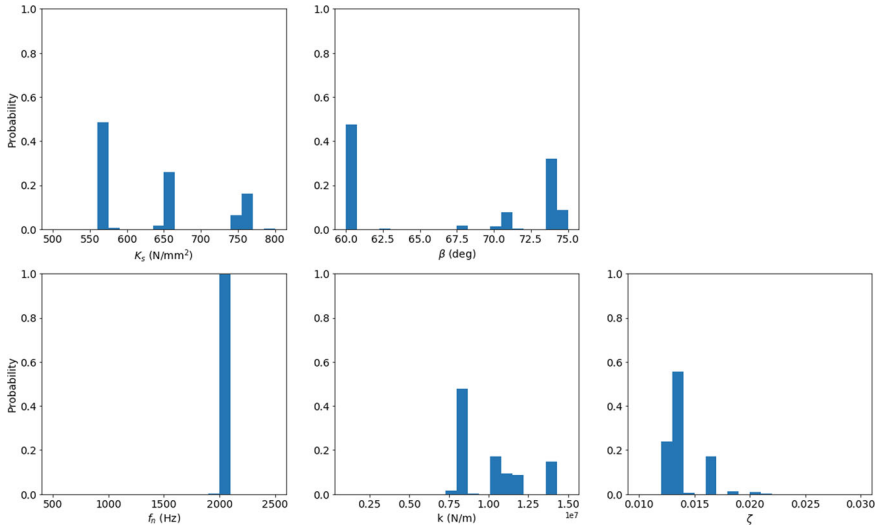


Fig. 11 Posterior distributions of stability input parameters after 13 tests

given by Eq. (25).

$$A = \min\left(1, \frac{p(x_c)q(x_n|x_c)}{p(x_n)q(x_c|x_n)}\right) \tag{25}$$

To draw N samples, the MCMC algorithm proceeds for $N-1$ iterations as follows.

1. Initialize the starting sample, $x_{n=1}$
2. For $n = 2$ to $N-1$ iterations, do:
   a. Generate candidate sample x_c from the proposal distribution conditioned on x_n, $q(x_c|x_n)$
   b. Randomly sample u from a uniform distribution of values between 0 and 1
   c. Compute acceptance ratio A using Eq. (25)
   d. If $u < A$, $x_{n+1} = x_c$, else $x_{n+1} = x_n$

The MCMC method is applied for Bayesian updating of the input parameters and the associated stability maps by treating the posterior distribution as the target distribution. The prior is a joint probability distribution of the inputs $\theta = \{K_s, \beta, f_n, k, \zeta\}$. For stability updating, the MCMC method is used to sample from the posterior joint distribution of the stability inputs given M test results. The posterior probability is calculated as the product of the likelihood and the prior. The advantage of the MCMC method is that the normalizing constant for the posterior pdf is not required for sampling. The acceptance ratio for MCMC sampling of the input parameters given M stability test results is shown in Eq. (26).

$$A = \min\left(1, \frac{p(\theta_c|\alpha_{1:M})q(\theta_n|\theta_c)}{p(\theta_n|\alpha_{1:M})q(\theta_c|\theta_n)}\right) \tag{26}$$

In Eq. (26), θ_c is the candidate sample of input parameters, θ_n is the nth input parameters sample, $p(\theta_c|\alpha_{1:M})$ and $p(\theta_n|\alpha_{1:M})$ are the posterior probabilities of θ_c and θ_n given M test results (see Eq. 22). The likelihood functions for stable and unstable results are given by Eqs. (19) and (20), respectively.

The updating procedure is described here. First, the prior distribution of the input parameters θ is selected. As noted, the prior distribution can be selected using all available information, including measurements, test results, physics-based simulations, and user beliefs. Second, an initial set of input parameters is selected, θ_1, which serves as a starting point for the Markov chain. Third, a joint proposal distribution for the input parameters, θ, is selected, where the choice of the proposal distribution influences the MCMC results. Although the Markov chain should converge to the underlying distribution for any proposal distribution, the mixing of the chain and the rate of convergence is dependent on the choice of the proposal distribution [21]. In general, the proposal distribution should be selected such that sampling is convenient. Therefore, a multivariate normal distribution with no correlation between parameters is a common choice. For the stability input parameters, a multivariate normal distribution is defined by the mean of the variables, θ_{mean}, and the covariance between the variables, θ_{cov}. For a symmetric proposal distribution (such as normal or uniform), the pdf values $q(\theta_n|\theta_c)$ and $q(\theta_c|\theta_n)$ are equal and cancel. In this case, the acceptance ratio is simplified as shown in Eq. (27).

$$A = \min\left(1, \frac{p(\theta_c|\alpha_{1:M})}{p(\theta_n|\alpha_{1:M})}\right) \tag{27}$$

Fourth, the MCMC algorithm enters a loop to sample N input parameters. Given M test results, the MCMC algorithm draws N samples from the posterior distribution proceeds as follows. A candidate set of input parameters, θ_c, is drawn from the proposal distribution, conditioned on the current set of parameters θ_n. This means that, for a normal proposal distribution, the mean is taken as the current value of the parameter set, θ_n. The acceptance ratio, A, is calculated using the posterior probabilities $p(\theta_c|\alpha_{1:M})$ and $p(\theta_n|\alpha_{1:M})$. For a uniform prior distribution, $p(\theta_n) = p(\theta_c)$, since all samples are equally likely between the minimum and maximum values. θ_c is accepted or rejected by comparing A to a random number between 0 and 1. If θ_c is accepted, $\theta_{n+1} = \theta_c$, else $\theta_{n+1} = \theta_n$. The MCMC loop is repeated for N iterations to draw samples from the posterior distribution given M test results. The posterior samples can be subsequently used to calculate the posterior probability of stability at any grid point $\{\Omega_i, b_j\}$. This is completed by first calculating the stability map for each posterior sample and then calculating the ratio of stability maps where the limiting axial depths at Ω_i is less than b_j and total number of stability maps N. The pseudo-code for MCMC sampling of input parameters given M experimental results is provided by Algorithm 3.

Algorithm 3 MCMC for stability map Bayesian updating

Input: Number of MCMC samples N, prior probability distribution for input parameters θ_{prior}, proposal distribution for the input parameters $\{\theta_{mean}, \theta_{cov}\}$ for a normal distribution and $\{\theta_{min}, \theta_{max}\}$ for a uniform distribution, M test results $\alpha_{1:M}$, standard deviation for axial depth σ_b

Output: Posterior samples $\theta_{1:N}$

1. Initialize the starting point $\theta_1 = \{K_s, \beta, f_n, k, \zeta\}_1$
2. $n = 1$
3. **while** $n \leq N$, do
4. Calculate the stability map with θ_n
5. Calculate the posterior probability $p(\theta_n|\alpha_{1:m})$ using Eq. 22
6. Generate candidate sample $\theta_c = \{K_s, \beta, f_n, k, \zeta\}$ using the proposal distribution
7. Calculate the stability map with θ_c
8. Calculate the posterior probability $p(\theta_c|\alpha_{1:m})$ using Eq. 22
9. Calculate the acceptance ratio A using Eq. 26 (or Eq. 27 for symmetric proposal distributions)
10. Randomly sample u from a uniform distribution of values between 0 and 1
11. **if** $u \leq A$, then
12. Set $\theta_{n+1} = \theta_c$
13. **elseif** $u >$ A, **then**
14. Set $\theta_{n+1} = \theta_n$
15. $n = n + 1$

The MCMC algorithm was applied to sample from the posterior distribution of input parameters using the test results shown in Fig. 2. The samples were subsequently used to calculate the posterior probability of stability. The prior distribution was selected as uniform as shown in Sect. 3.2, resulting in the prior probability of stability shown in Fig. 9. A normal proposal distribution was used with standard deviation values for each parameter equal to 10% of the prior uniform distribution range. σ_b was 0.25 mm. Figure 12 displays the posterior probability of stability after four, six, 11, and 13 tests using 1000 MCMC samples. Figure 13 shows the posterior distribution of input parameters.

3.4 Discussion

In Sects.3.1 to 3.3, different methods for updating the stability map using test results were described. This section presents a comparison between methods and considerations for method selection. Figure 14 shows the predicted stability map for the three methods, where the boundary represents a posterior probability of stability equal to 0.5. Results are provided for six tests (Fig. 14a) and 13 tests (Fig. 14b). The zero-order frequency domain stability map is also shown for comparison. As described in Sect. 3.1, the non-model-based method uses the knowledge of the stability boundary in the Bayesian learning procedure. As a result, the method provides a local update to

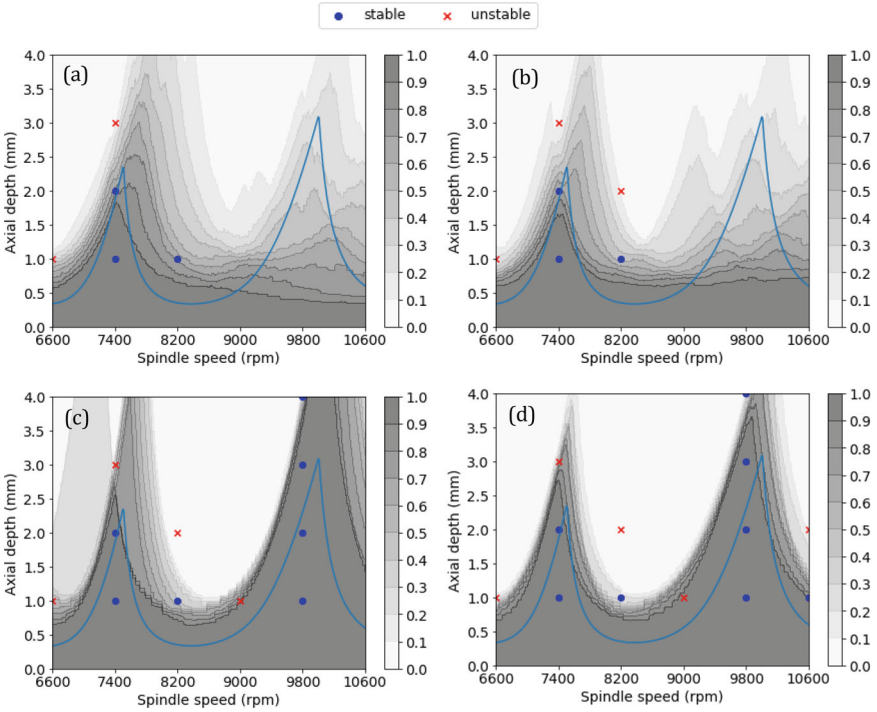


Fig. 12 Posterior probability of stability after **a** four tests; **b** six tests; **c** 11 tests; and **d** 13 tests using 1000 MCMC samples. The stability map from Fig. 2 is also shown for comparison

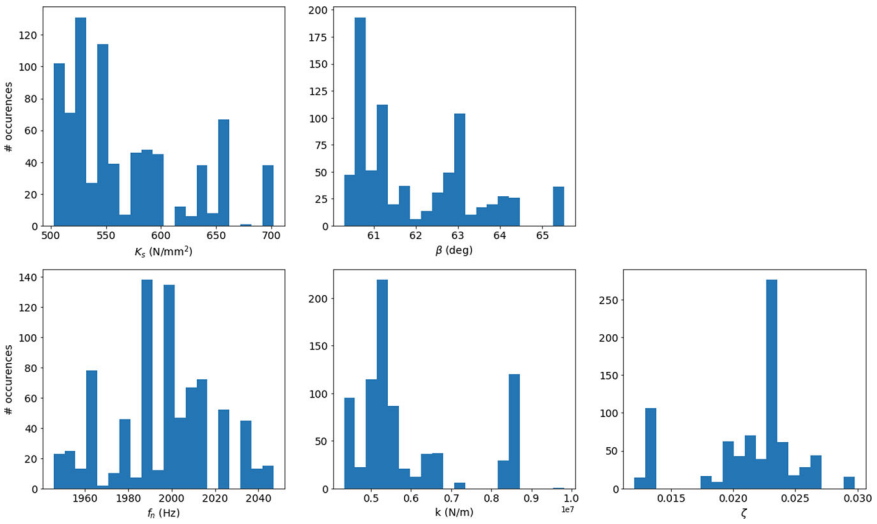


Fig. 13 Posterior distributions of input parameters after 13 tests

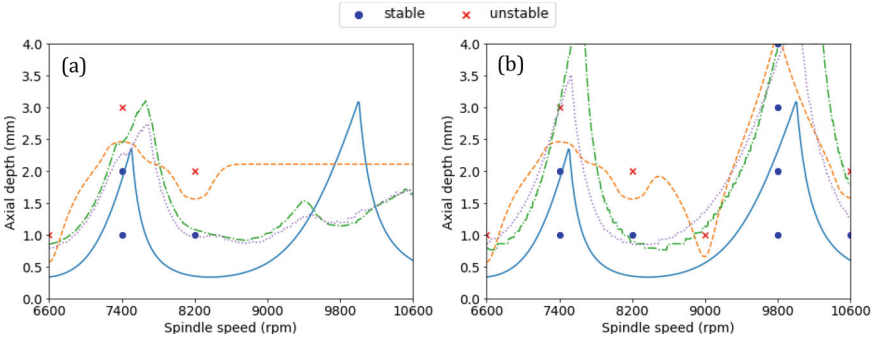


Fig. 14 Comparison of the predicted stability map for the non-model-based grid method (orange dashed), model-based random sample stability map (green dash dot), and model-based MCMC method (purple dotted) after six tests (**a**) and 13 tests (**b**). The stability map from Fig. 2 is shown as the blue solid line

the probability of stability. This is seen in Fig. 14a where the stability map prediction is constant at 2 mm after 8600 rpm. The model-based methods use the zero-order frequency domain stability model. Therefore, the methods provide an update at all spindle speeds given six test results. This is seen in the two stability boundaries predicted by both methods at 9400 and 10,600 rpm. After 13 tests, all methods correct for the initial discrepancy in the zero-order frequency domain prediction and the test results (stable results at {8200 rpm, 1 mm}, {9800 rpm, 3 mm}, {9800 rpm, 4 mm}, and {10,600 rpm, 1 mm}. Note that the non-model-based prediction does not resemble a standard stability map since it does not use the zero-order frequency domain model. The model-based random stability map and MCMC method fit the experimental results well.

Recall that for all three methods described in Sect. 3, the prior probability of stability did not use measurements for the FRF or cutting force model. For the non-model-based grid method, the knowledge of the stability map was used to establish the prior probability of stability (see Fig. 5). For the model-based methods, a wide distribution of the input parameters was selected. All the methods were able to learn the stability boundary using test results. Note that although the results were presented with a single mode, the methods can be extended to learn the stability map for multiple modes. The choice between the three methods can be based on the following considerations.

As noted, there exists uncertainty in the zero-order frequency domain stability predictions. In special cases, such as spindle-speed dependent FRFs [22] or special geometry tools with variable flute spacing, variable helix, multiple inserts, or specialized edge preparation [23], the zero-order frequency domain stability model is less accurate. In these cases, time domain numerical models can be applied to predict the stability map. However, these models are computationally expensive. Additionally, information on the tool geometry, such as the non-uniform flute spacing, may not be available. In such situations, the non-model-based method is a good choice to

learn the stability map. On a standard laptop with i7-1265U processor, the computation time to update the probabilities at each grid point (16,000 grid points) in the non-model-based method was approximately 1 min. As seen in Fig. 14, the stability map predictions using the model-based random sample method and MCMC method agree well. The difference between the two is the computational expense. In the model-based random sample method, the stability maps are generated first using a prior distribution of input parameters. Each test result updates the probability of each sample stability map. The time to generate the 1×10^4 initial sample stability maps can be significant, especially if time domain simulation is selected. However, the computation time can be substantially reduced by parallelizing the calculation for different inputs. In MATLAB, for example, the 'parfor' loop executes the for-loop in parallel using workers in a parallel pool [24]. In Python, the multiprocessing package can be used to parallelize stability map calculations by initiating a pool of workers; multiple stability input parameters can be passed as tuple arguments in the pool.starmap function [25]. On a standard laptop (Intel i7-1265U, 10 core processor), the time to generate 10^5 stability maps using parallel processing was approximately 15 min. The stability maps need to be generated only to establish the prior; the procedure to update the probability of each map and calculate the probability of stability at different grid points is computationally inexpensive; the time for updating the probabilities of 10^5 samples for each test result was less than 30 s. For the MCMC method, however, samples are drawn for each test result which can result in significant computation times. This is especially true for computationally expensive models since the stability map needs to be calculated for every candidate sample in the MCMC sampling loop. The MCMC computation time can be reduced using a parallelized sampling strategy where multiple samples are accepted at each iteration [26]. Furthermore, the proposal distribution can be recalculated at each step using an adaptive MCMC approach; this results in fewer samples being rejected [27]. The standard MCMC method required 5 min to generate 5000 stability samples on a standard laptop with an Intel i7-1265U processor.

However, the MCMC method is more flexible in generating new samples that agree with the stability test results. This is seen in the MCMC stability map prediction at 7400 and 10,600 rpm in Fig. 14. The model-based random sample method over-predicts the limiting axial depth at both spindle speeds relative to the MCMC method. Both methods have one important consideration. They will not converge if the true input parameters and the associated stability map that agrees with the test results are not included in the prior. However, this can be addressed by selecting a wide range of values for the input parameters, as shown in Sect. 3.2. The limitation is that more tests may be required for the wider distributions.

In an industrial environment, all the methods can be easily implemented on an edge-device or an industrial PC. As shown in Fig. 14, each method can learn the stability boundary in a small number of tests. Therefore, the method can be used to learn the stability map for a given tool-material combination in a high production or small batch environment. Section 5.3 details a closed-loop control system for system for automated stability testing in an industrial environment.

4 Test Parameter Selection

This section describes methods for parameter selection during stability testing. A common goal for stability tests is to identify the {spindle speed, axial depth} which maximizes the material removal rate, *MRR*. The advantage of the probabilistic stability map is that information to be gained from a test can be calculated prior to doing the test. As noted in Sect. 2, this is called the value of information [28]. As a result, test parameters can be selected that maximize the value from each test. This results in a convergence to the optimal parameter identification goal with a sequential selection of machining parameters. The value of information metric for test parameters selection is called an acquisition function [29]; the optimal test parameters maximize the acquisition function. This approach is more efficient than a traditional design of experiments where all the test parameters are selected prior to performing any tests. The procedure for sequential sampling of test parameters follows.

1. Start with a prior probability of stability.
2. Calculate the acquisition function using a value of information metric.
3. Select test parameters that maximize the acquisition function.
4. Complete the stability test at the selected parameters.
5. Update the probability of stability using the test result.
6. Repeat steps 2–5 for a desired number of tests or until a convergence criterion is met.

As noted, determining the stability map using the zero-order frequency domain model requires information on the tool tip FRF and the cutting force model for the tool-material combination. In an industrial environment, especially for small-to-medium size manufacturers, the tool tip FRF is often not known because the measurement capability is not available. In this case, the optimal stable machining parameters could be selected by completing a few tests and learning the stability map. As shown in Sect. 3, the probability of stability over the entire {spindle speed, axial depth} domain can be updated using each test result. The objective is to find the {spindle speed, axial depth} combination with the highest *MRR*, which is stable with certainty (i.e., the probability of stability is equal to one).

Using the prior probability of stability, let P denote the optimal grid point, or the {spindle speed, axial depth} combination with the highest *MRR* and a probability of stability equal to one. Let MRR_{prior} denote the *MRR* at P. Consider a test at any arbitrary grid point G in the {spindle speed, axial depth} domain. Let MRR_G denote the *MRR* at G. If the test at G is stable, it represents a feasible set of parameters with *MRR* equal to MRR_G. However, if the test at G is unstable, it is not a feasible parameter set. In this case, the optimal *MRR* remains equal to MRR_{prior} (since it is the last known optimal stable *MRR*). Therefore, the expected[1] *MRR* after a test at G is given by Eq. 28, where, $\mathbb{E}$ denotes expectation and $p(s_G)$ is the probability of stability at the grid point G.

[1] The expected value is the mean value obtained from many tests. It is alternately referred to as the "long-term mean".

$$\mathbb{E}[MRR]_G = p(s_G)MRR_G + (1-p(s_G))MRR_{prior} \tag{28}$$

The expected improvement in *MRR* after a test at *G* is given by Eq. (29), where I denotes improvement.

$$\begin{aligned}\mathbb{E}[\mathrm{I}(MRR)]_G &= \mathbb{E}[MRR]_G - MRR_{prior} \\ &= \left(p(s_G)MRR_G + (1-p(s_G))MRR_{prior}\right) - MRR_{prior} \\ &= p(s_G)\left(MRR_G - MRR_{prior}\right)\end{aligned} \tag{29}$$

The expected improvement in *MRR* can be expressed as an expected percentage improvement over MRR_{prior} as shown in Eq. (30).

$$\mathbb{E}[\%I(MRR)]_G = p(s_G)\frac{\left(MRR_G - MRR_{prior}\right)}{MRR_{prior}}100\% \tag{30}$$

Expressing the expected improvement in *MRR* as percentage improvement over MRR_{prior} provides an intuitive stopping criterion. For example, testing can be terminated when the maximum expected percentage improvement over MRR_{prior} is less than 5%. Figure 15 shows the sequence of the first three tests for the non-model-based grid method. Before any testing, the optimum parameters are taken as {10,600 rpm, 0.1 mm}. Note that 0.1 mm is the smallest grid point axial depth (since 0 mm is not a feasible parameter). Figure 15a shows $\mathbb{E}[\%I(MRR)]$ for the first test (using the prior probability of stability shown in Fig. 5). The optimal test parameters are {10,600 rpm, 2.2 mm} with $\mathbb{E}[\%I(MRR)] = 1002\%$ (shown as a yellow dot in Fig. 15a). Note that the value is large since the prior optimal axial depth is very small. The stability test at {10,600 rpm, 2.2 mm} is unstable. Figure 15b shows the posterior probability of stability after the unstable test at {10,600 rpm, 2.2 mm}. Since the test was unstable, the prior optimum *MRR* remains at {10,600 rpm, 0.1 mm}. Figure 15c shows $\mathbb{E}[\%I(MRR)]$ for the second test. The optimal test parameters are {10,230 rpm, 2.1 mm} with $\mathbb{E}[\%I(MRR)] = 962\%$. Figure 15d shows the posterior probability of stability after an unstable test at {10,230 rpm, 2.1 mm}. Figure 15e and g show $\mathbb{E}[\%I(MRR)]$ for the third and fourth test, respectively. Figure 15f and h show the posterior probability of stability given test results at the selected test parameters. Note that the third test at {9830 rpm, 2.1 mm} is stable. Therefore, the optimum *MRR* is updated with *MRR* at {9830 rpm, 2.1 mm}. This reduces the maximum $\mathbb{E}[\%I(MRR)]$ for the fourth test to 17.05%.

Figure 16 shows the posterior probability after eight tests. The test procedure was terminated when the maximum $\mathbb{E}[\%I(MRR)]$ was less than 5%. The method identified the optimal parameters at {9830 rpm, 3.9 mm} after eight tests. Testing completed using different stability maps shows that the method typically converges to the optimum within 10 to 15 tests [19]. Recall that these results did not require a model or additional information about the input parameters.

The test procedure was repeated using the model-based random sample method. Figure 17 shows the results. The prior probability of stability is shown in Fig. 9. As

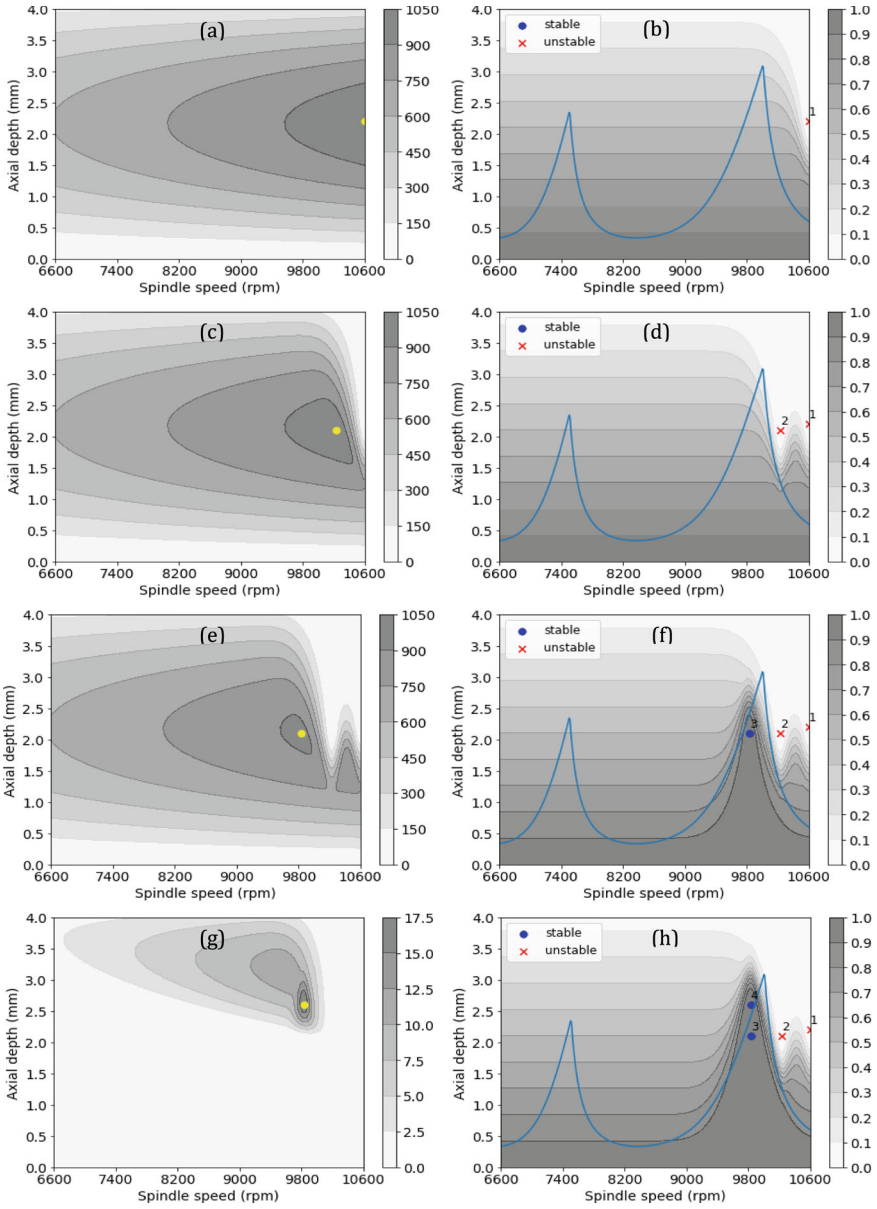


Fig. 15 Sequential test results using the non-model-based grid method. The left column shows the $\mathbb{E}[\%I(MRR)]$; the optimal parameters with maximum $\mathbb{E}[\%I(MRR)]$ are shown as a yellow dot and the $\mathbb{E}[\%I(MRR)]$ values are shown in the colorbar. The right column shows the posterior probability of stability after testing at the optimal parameters, where the grayscale indicates the probability values. The stability map from Fig. 2 is also included as the blue line

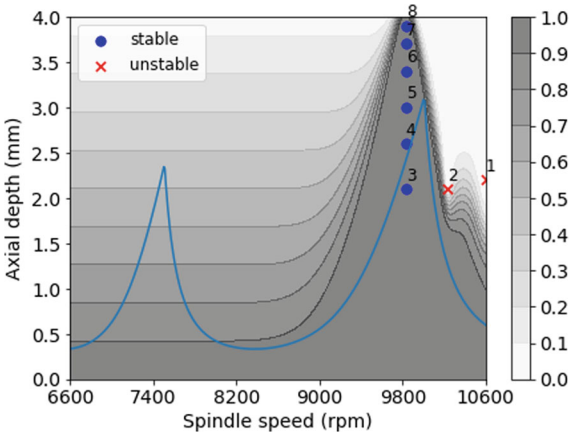


Fig. 16 Sequence of eight tests for optimal parameter identification for the non-model-based grid method

noted, before any testing the optimal parameters were {10,600 rpm, 0.1 mm}. The method converges to {10,140 rpm, 4 mm} in five tests. As expected, the model-based approach convergences to the optimal faster than the non-model-based method. This is because, as stated in Sect. 3.4, each stability test result updates the probability of the input parameters. In Fig. 17, this results in the selection of {9780 rpm, 3.4 mm} for the fourth test. In general, using the model-based method, convergence to the optimum (within 5% of the true optimal *MRR*) occurs within five to 10 tests. The model-based MCMC method shows similar results. Note that the expected improvement approach is a global optimization method which balances the trade-off between exploration (where the probability of stability is low, but the improvement in *MRR* is large) and exploitation (where the probability of stability is high, but the improvement in *MRR* is low). Therefore, the final convergence result does not depend on the initial starting point. The starting point will only influence the rate of convergence to the optimal.

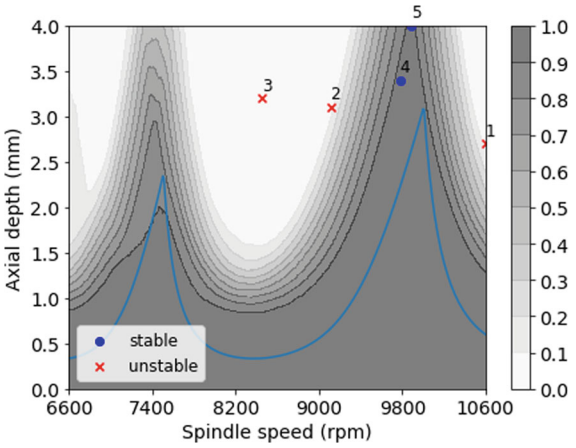


Fig. 17 Sequence of five tests for optimal parameter identification for the model-based random sample stability map method

5 Advanced Topics

5.1 Model-Based Prior Selection

As noted, the prior probability of stability is selected using all available information. In Sect. 3.2, it was assumed that the FRF information is not available. This results in a prior distribution selection of stability input parameters using prior knowledge or experience. To obtain FRF information, two options are available. First, the tool tip FRF can be measured using impact/tap testing, where an instrumented hammer is used to excite the tool tip and the response is measured with a vibration sensor, such as a low-mass accelerometer. Second, the tool tip FRF can be predicted using receptance coupling substructure analysis (RCSA) [30, 31]. In this frequency domain approach, models of the tool and holder (typically Timoshenko beam models) are coupled to a measurement of the spindle-machine. The spindle-machine receptances (FRFs) are identified by inserting a standard geometry artifact in the spindle, measuring the artifact-spindle-machine FRF, and then using inverse RCSA to isolate the spindle-machine receptances [32]. See Fig. 18.

In order to propagate RCSA input uncertainties to tool tip FRF prediction uncertainties, Monte Carlo simulation was employed. Normal distributions were defined for the connection stiffness and damping values between the tool and holder and were randomly sampled during the 5000 Monte Carlo iterations. The standard deviations for these distributions were set as 20% of the mean values, which were based on previous experience. The 5000 predicted FRFs were then fit to extract the single degree of freedom modal parameters. Figure 19a shows the prior distribution of the modal parameters using the RCSA approach. Given the tool tip FRFs, a stability map was generated for each using the zero-order frequency domain model while also including uncertainty in the force model. Normal distributions were selected for K_s and β with standard deviations equal to 10% of the mean values (600 N/mm^2 and 68°). Figure 19b shows the prior probability of stability using the model-based approach. The stability map from Fig. 2 is also displayed for comparison. As seen in Fig. 19b, a model-based prior results in reduced uncertainty in the probability of stability as compared to Fig. 9, where the broad modal parameter distributions were $f_n = \text{U}(500, 2500)$ Hz, $k = \text{U}(1.0 \times 10^6, 1.5 \times 10^7)$ N/m, $\zeta = \text{U}(0.01, 0.03)$ for Fig. 9.

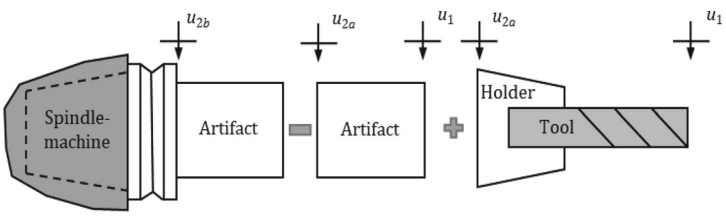


Fig. 18 RCSA method for tool tip FRF prediction

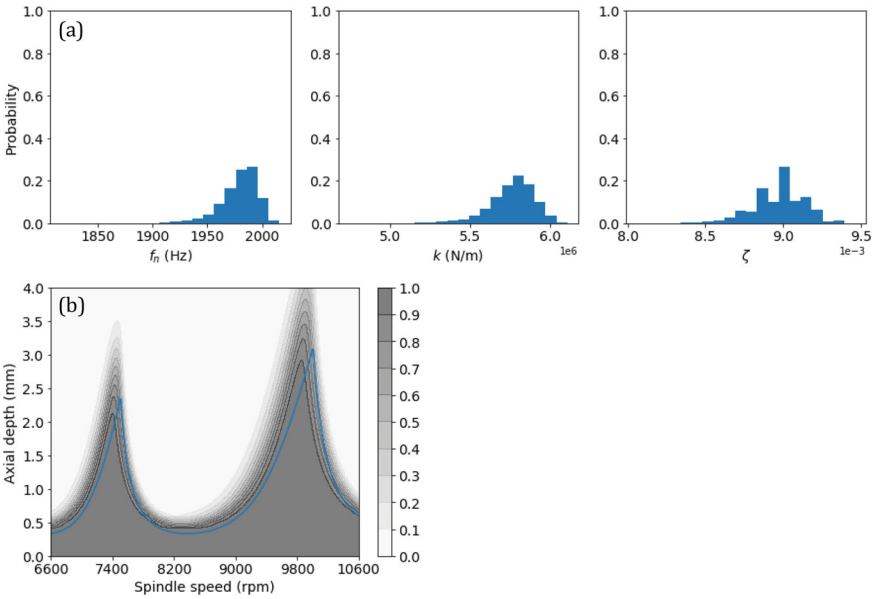


Fig. 19 **a** Prior distributions of modal parameters and **b** prior probability of stability from the model-based prior. The stability map from Fig. 2 is also shown for comparison

Figure 20 shows the posterior distribution of modal parameters (Fig. 20a) and the posterior probability of stability (Fig. 20b) using the same grid tests from Fig. 2. Figure 21 shows the sequence of cutting tests for optimal parameter identification using the expected percentage improvement in *MRR* criterion. The approach converges to the optimal in five tests. As seen in Fig. 21, the method converges to {9930 rpm, 3.7 mm} in four tests. Note that all four recommended optimal test parameters were stable. This is in contrast to the sequence of tests starting with the broad prior distribution in Fig. 9, which results in the first three tests being unstable. In general, a model-based prior will result in fewer tests for convergence with more stable test results than unstable.

5.2 Chatter Frequency for Learning

In Sect. 3, stability test results were used to update the prior probability of stability. For an unstable cut, significant frequency content in an audio signal is observed at the chatter frequency. The chatter frequency information can also be used to update the prior probability of stability in addition to the test result [33]. To illustrate, consider two new random sample stability maps shown in Fig. 22a; the two stability maps were generated by randomly sampling the prior stability input parameter distribution described in Sect. 3.2. Using the zero-order frequency domain model, the chatter

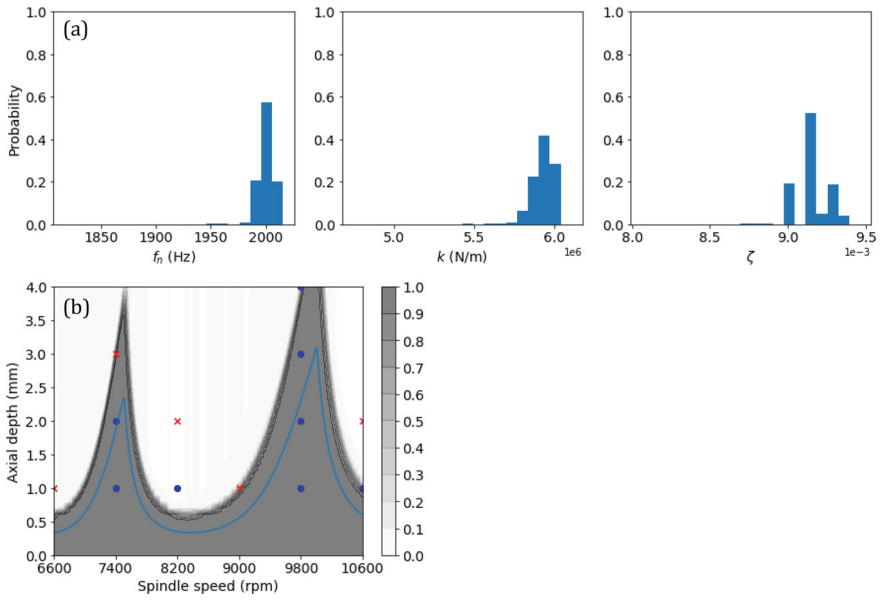


Fig. 20 **a** Posterior distributions of modal parameters and **b** posterior probability of stability from the model-based prior using grid test results. The stability map from Fig. 2 is also shown for comparison

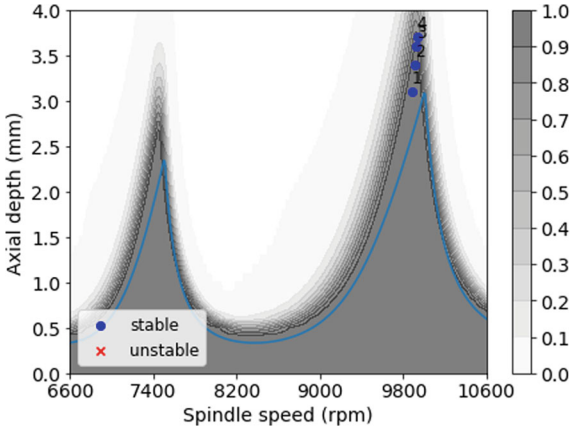


Fig. 21 Sequence of cutting tests with the expected percentage improvement in *MRR* criterion. The stability map from Fig. 2 is also shown for comparison

frequencies were also calculated for the two stability maps. The chatter frequencies are shown in Fig. 22b.

Consider an unstable result at {6600 rpm, 1 mm}. The frequency content of the audio signal is shown in Fig. 3a, where the chatter frequency is observed at 2054 Hz. The likelihood of an unstable result at {6600 rpm, 1 mm}, $p(\alpha : -|\theta_n)$, is equal to 1

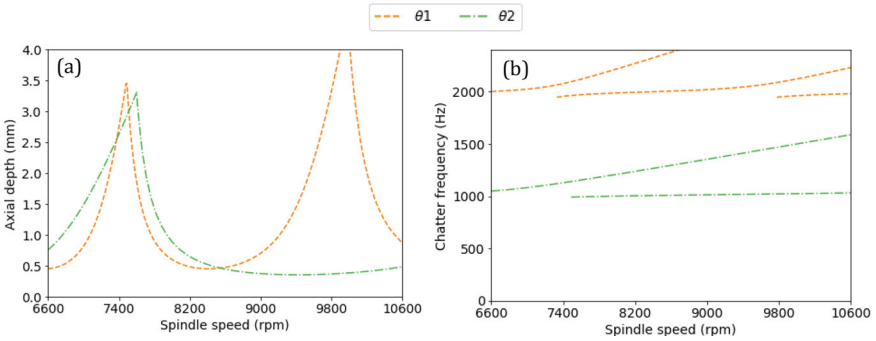


Fig. 22 **a** Two new random sample stability maps calculated by sampling from the prior stability input parameter distribution described in Sect. 3.2; and **b** chatter frequencies associated with the two sample stability maps

for the two sample stability maps (see Eq. 20). This is because both sample stability maps have a limiting axial depth less than 1 mm at the test spindle speed of 6600 rpm.

From Fig. 22b, the likelihood of observing a chatter frequency of 2054 Hz is higher for the stability map with θ_1 than θ_2. This is because the predicted chatter frequency at 6600 rpm for θ_1 is 2000.1 Hz and for θ_2 is 1047.9 Hz. The chatter frequency information for an unstable cut can be incorporated into the likelihood as shown in Eq. (31) [33].

$$p(\alpha : -|\theta_n) = e^{\left(-\frac{1}{2}\left(\frac{\left(b_j - b_T\right)^2}{\sigma_b^2} + \frac{(\Delta f_c)^2}{\sigma_{f_c}^2}\right)\right)} \tag{31}$$

In Eq. (31), f_c is the chatter frequency, σ_{f_c} is the uncertainty in chatter frequency, and Δf_c is the difference between the predicted and measured chatter frequency at b_T. Figure 23 shows the posterior probability of stability given an unstable result at {6600 rpm, 1 mm} without chatter frequency (Fig. 23a) and with chatter frequency (Fig. 23b) in the likelihood, where σ_{f_c} was 50 Hz. Like σ_b, σ_{f_c} allows for disagreement between the model prediction and the measured value of the chatter frequency. The zero-order frequency domain stability map from Fig. 2 is included for comparison. Figure 24 shows the posterior distribution of f_n for the two approaches after the unstable result at {6600 rpm, 1 mm}. As seen in Figs. 23 and 24, a single unstable test results in a rapid convergence to the underlying map and f_n when including the chatter frequency information in the likelihood function.

5.3 *Closed-Loop Control for Automated Testing*

Section 4 described a test strategy for optimal stable machining parameter identification. Results show that the expected percentage improvement in *MRR* strategy

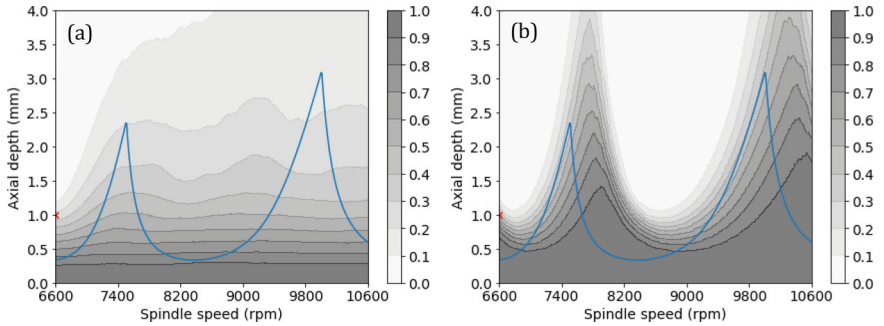


Fig. 23 Posterior probability of stability; **a** without using chatter frequency in the likelihood as shown in Eq. (20) and **b** using chatter frequency in the likelihood as shown in Eq. (31). The Fig. 2 stability map is shown for comparison

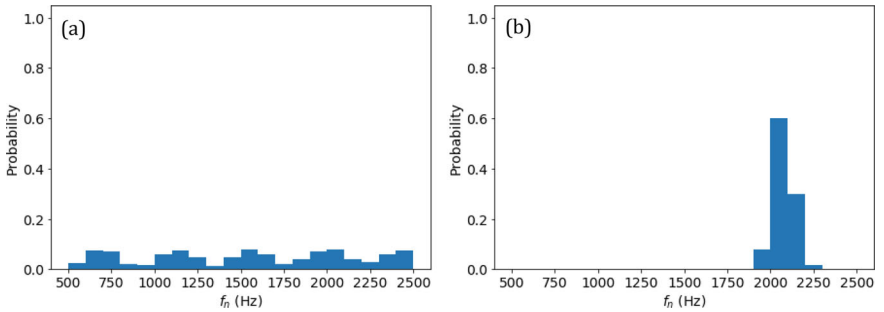


Fig. 24 Posterior distribution of f_n; **a** without using chatter frequency in the likelihood (Eq. 20) and **b** using chatter frequency in the likelihood (Eq. 31)

enables convergence to the optimal in 10 to 20 tests for the non-model-based grid method and five to 10 tests for the model-based random sample stability map method. The testing strategy can be automated in an industrial environment using a closed-loop control strategy. The closed-loop control system for stability testing consists of:

(1) an architecture to monitor the machine state and stability tests through communication protocols such as MTConnect or OPC-UA
(2) an analysis module to select the optimal test parameters using the acquisition function, automatically classify the test cut as stable/unstable using audio signal, and calculate the posterior probability of stability given test results
(3) a feedback mechanism to communicate the stability test parameters to the machine controller to perform a test [34].

The optimal test parameters can be saved in G-code instructions. The G-code is transferred to the CNC controller using timed updates. Figure 25 shows the timing diagram for the feedback mechanism to transfer the test parameters and update

machine instructions [34]. Controller flags (0 or 1) are used to determine when the G-code with the test parameters can be transferred to the machine. After the closed-loop control is initiated, the flag is set to one, signaling that commands can be sent to the machine, overwriting specific commands in the specified memory location on the machine controller. In Fig. 25, a time of 2000 ms is allocated to transfer the G-code to the machine; this time is user-defined. At the end of the allocated time, the flag is set to zero, signaling that no additional instructions can be sent. The updated G-code with the test parameters is parsed by the machine and the test is completed. The sound is recorded and is analyzed to determine stability. The CNC machine resets the flag to 1 after the cut is completed indicating that G-code for the next cut can be transferred. For the model-based random sample stability map, 2000 ms is sufficient time to calculate the posterior probability of stability and the next optimal test parameters using the expected percentage improvement in *MRR* criterion. Figure 26 shows the generic operational flow for the feedback mechanism; the flow is designed to work on any commercial CNC machine with a standard operating system [34].

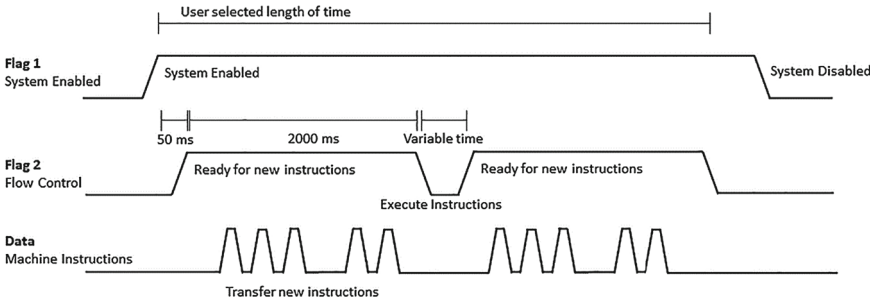


Fig. 25 Timing diagram for the feedback mechanism to update machine instructions [34]

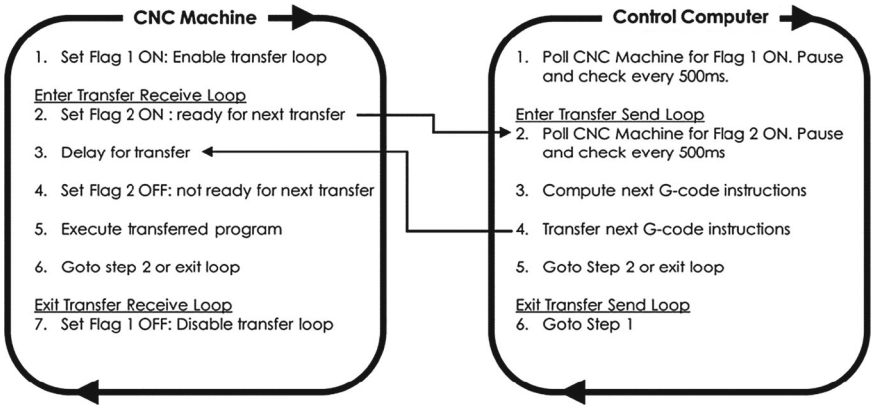


Fig. 26 Generic operational flow for the feedback mechanism [34]

The closed-loop control system can be used in an industrial environment to quickly identify optimal stable parameters prior to executing a part program and machining an actual part. Because a given part program can require multiple tools to execute difference operations, such as face milling with an indexable facemill or contour milling with a ball-nose solid carbide endmill clamped in a thermal shrink fit holder, testing is required for each tool-holder combination. The results for each tool can be archived for use in the current and future part programs, provided the tool setup is consistent (i.e., the tool extension length from the holder should not be changed). Furthermore, the tests can be completed periodically to capture new optimal parameters due to any changes in the spindle dynamics over time.

5.4 *Test Parameter Selection for Stability Map Identification*

In Sect. 4, a test strategy for optimal stable parameter identification using an expected percentage improvement in *MRR* criterion was described. In some applications, the goal for completing the tests may be to identify the entire stability map. The criterion for optimal stable parameter identification can be modified for stability map identification as follows.

In the model-based random sample path method described in Sect. 3.2, the prior distribution for input parameters (force model and tool tip FRF modal parameters) is used to generate sample stability maps. Each test result is used to update the probability of the stability map as well as the underlying input parameters. After each update, the posterior mean and standard deviation for each input parameter can be calculated; Eqs. (23) and (24) were used to calculate the mean, $\mu(K_s)$, and standard deviation, $\sigma(K_s)$, of K_s, for example. Each test updates the probabilities of the N sample stability maps and reduces the uncertainty in the stability input parameters.

To illustrate, consider K_s. Let $\sigma(K_s)_{prior}$ be the prior standard deviation for K_s, calculated using the prior probability of the N samples. The expected standard deviation in K_s derived from a test at G is given by Eq. (32).

$$\mathbb{E}[\sigma(K_s)]_{\mathrm{G}} = p(s_G)\sigma(K_s)_{G:+} + p(u_G)\sigma(K_s)_{G:-} \tag{32}$$

In Eq. (32), $\sigma(K_s)_{G:+}$ is the K_s standard deviation given G is stable and $\sigma(K_s)_{G:-}$ is the K_s standard deviation given G is unstable. The expected reduction in $\sigma(K_s)$ given a test at G is given by Eq. (33).

$$\begin{aligned}\mathbb{E}[R(\sigma(K_s))]_G &= \sigma(K_s)_{prior} - \mathrm{E}[\sigma(K_s)]_G \\ &= \sigma(K_s)_{prior} - p(s_G)\sigma(K_s)_{G:+} + p(u_G)\sigma(K_s)_{G:-}\end{aligned} \tag{33}$$

Like the expected improvement in *MRR* (shown in Eq. 30), the expected reduction in uncertainty can be expressed as a percentage reduction over $\sigma(K_s)_{prior}$ as shown

in Eq. (34).

$$\mathbb{E}[\%R(\sigma(K_s))]_G = \frac{\left(\sigma(K_s)_{prior} - \mathbb{E}[\sigma(K_s)]_G\right)}{\sigma(K_s)_{prior}} 100\% \tag{34}$$

In Eq. (34), $\mathbb{E}[\%R(\sigma(K_s))]_G$ is the expected percentage reduction in $\sigma(K_s)$ given a test at G. Recall that the input parameters include the force model (K_s, β) and the modal parameters (f_n, k, ζ) that represent the tool tip FRF. A criterion for test point selection could use the average of the expected percent reduction in uncertainty for all input parameters. To illustrate, for five stability input parameters, the average expected reduction in parameter uncertainty is given by Eq. (35), where $A(\mathbb{E}[\%R(\theta)])_G$ is the average expected reduction in parameter uncertainty.

$$A(\mathbb{E}[\%R(\theta)])_G = \frac{\mathbb{E}[\%R(\sigma(K_s))]_G + \mathbb{E}[\%R(\sigma(\beta))]_G + \mathbb{E}\left[\%R(\sigma(f_n))\right]_G + \mathbb{E}[\%R(\sigma(k))]_G + \mathbb{E}[\%R(\sigma(\zeta))]_G}{5} \tag{35}$$

For a given grid point G, the algorithm proceeds as follows. First, calculate the prior standard deviation for all input parameters (see Eq. 24). Second, assume the grid point is stable and calculate the posterior standard deviation for all input parameters. Third, assume the grid point is unstable and calculate the posterior standard deviation for all input parameters. Fourth, calculate the expected percentage reduction for all parameters (Eqs. 33 and 34 provide the expressions for K_s). Fifth, calculate the average expected reduction in parameter uncertainty (Eq. 25) and select the optimal test parameters where it is maximum. Figure 27a shows the $A(\mathbb{E}[\%R(\theta)])$; the optimal tests parameters are {7080 rpm, 1.4 mm}. Figure 27b shows the posterior probability of stability after the first unstable test at {7080 rpm, 1.4 mm}. The prior probability of stability is shown in Fig. 9. Figure 28 shows a sequence of 10 tests with the average expected reduction in parameter uncertainty criterion.

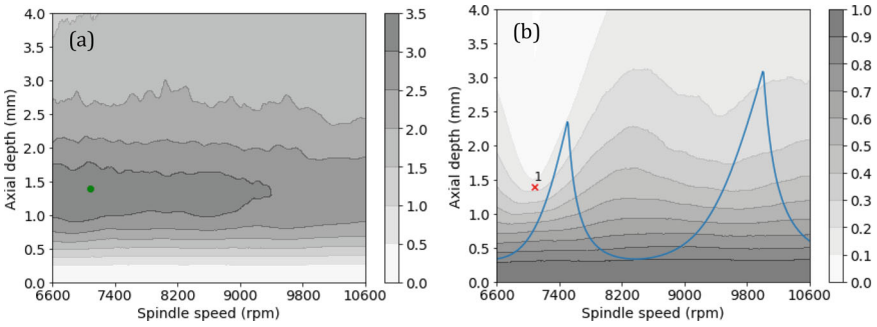


Fig. 27 **a** Average expected reduction in parameter uncertainty criterion for the first test; and **b** posterior probability of stability after the first unstable test at {7080 rpm, 1.4 mm}

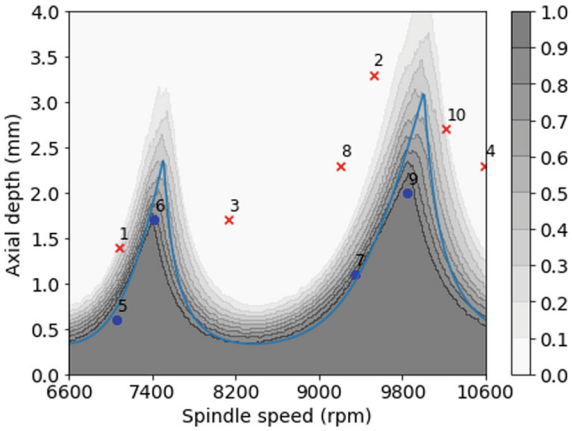


Fig. 28 Sequence of tests for stability map identification

6 Outlook

While machine learning (ML) and artificial intelligence (AI) techniques are well-established in data-rich domains, such as e-commerce (fraud prevention), education (personalized learning), lifestyle (facial recognition), navigation, and healthcare, to name a few, its widespread application to manufacturing processes is relatively new. Approaches can be loosely divided into purely data-driven, which is agnostic to physical laws and domain expertise, and physics-informed, where first principle models are leveraged to guide learning activities.

While ML/AI can provide tremendous advantage, its implementation is not without challenges. These include data collection, storage, and recovery; domain expertise to support algorithm development; high-performance computing, when required; trust; and data security [35]. Broad adoption, therefore, must address these challenges while continuing innovation in research and development. In the authors' opinion, a promising research direction is the intersection of established analytical and numerical progress models with data learning strategies, such as Bayesian inference, in the physics-informed approach. Key considerations are data cost, interpretation, and accuracy, as well as automated approaches to apply the models and associated decision making (such as parameter selection with real-time modification). As these requirements are met, the potential for increased productivity in manufacturing processes and systems is significant.

References

1. Schmitz T, Smith KS (2019) Machining dynamics: frequency response to improved productivity, 2nd edn. Springer, New York, NY
2. Kayhan M, Budak E (2009) An experimental investigation of chatter effects on tool life. Proc Inst Mech Eng Part B J Eng Manuf 223(11):1455–1463

3. Altintas Y, Stepan G, Budak E, Schmitz T, Kilic ZM (2020) Chatter stability of machining operations. J Manuf Sci Eng 142(11):110801
4. Altintas Y, Weck M (2004) Chatter stability of metal cutting and grinding. CIRP Ann 53(2):619–642
5. Altintas Y, Budak E (1995) Analytical prediction of stability lobes in milling. CIRP Ann 44(1):357–362
6. Delio T, Tlusty J, Smith S (1992) Use of audio signals for chatter detection and control. ASME J Eng Ind 114(2):146–157
7. Rubeo MA, Schmitz TL (2017) Amplitude ratio: a new metric for milling stability identification. Procedia Manuf 10:351–362
8. Schmitz T, Karandikar J, Kim NH, Abbas A (2011) Uncertainty in machining: workshop summary and contributions. J Manuf Sci Eng 133(5):051009
9. Hazelrigg GA (2012) Fundamentals of decision making for engineering design and systems engineering. GA Hazelrigg
10. Koza JR, Bennett FH, Andre D, Keane MA (1996) Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In: Artificial intelligence in design'96, pp 151–170
11. Honeycutt A, Schmitz TL (2017) Milling stability interrogation by subharmonic sampling. J Manuf Sci Eng 139(4):041009
12. Honeycutt A, Schmitz T (2017) A numerical and experimental investigation of period-n bifurcations in milling. J Manuf Sci Eng 139(1):011003
13. Honeycutt A, Schmitz TL (2018) Milling bifurcations: a review of literature and experiment. J Manuf Sci Eng 140(12):120801
14. Duncan GS, Kurdi M, Schmitz T, Snyder J (2006) Uncertainty propagation for selected analytical milling stability limit analyses. Trans NAMRI/SME 34:17–24
15. Karandikar JM, Schmitz TL, Abbas AE (2012) Spindle speed selection for tool life testing using Bayesian inference. J Manuf Syst 31(4):403–411
16. Karandikar JM, Tyler CT, Abbas A, Schmitz TL (2014) Value of information-based experimental design: application to process damping in milling. Precis Eng 38(4):799–808
17. Insperger T, Munoa J, Zatarain MA, Peigné G (2006) Unstable islands in the stability chart of milling processes due to the helix angle. In: CIRP 2nd international conference on high performance cutting, Vancouver, Canada, pp 12–13
18. Patel BR, Mann BP, Young KA (2008) Uncharted islands of chatter instability in milling. Int J Mach Tools Manuf 48(1):124–134
19. Karandikar J, Honeycutt A, Schmitz T, Smith S (2020) Stability boundary and optimal operating parameter identification in milling using Bayesian learning. J Manuf Process 56:1252–1262
20. Andrieu C, De Freitas N, Doucet A, Jordan MI (2003) An introduction to MCMC for machine learning. Mach Learn 50:5–43
21. Gelman A, Gilks WR, Roberts GO (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann Appl Probab 7(1):110–120
22. Cheng CH, Schmitz TL, Scott Duncan G (2007) Rotating tool point frequency response prediction using RCSA. Mach Sci Technol 11(3):433–446
23. Gomez M, No T, Smith S, Schmitz T (2020) Cutting force and stability prediction for inserted cutters. Procedia Manuf 48:443–451
24. https://www.mathworks.com/help/parallel-computing/parfor.html
25. https://docs.python.org/3/library/multiprocessing.html
26. Calderhead B (2014) A general construction for parallelizing Metropolis-Hastings algorithms. Proc Natl Acad Sci 111(49):17408–17413
27. Haario H, Saksman E, Tamminen J (2001) An adaptive metropolis algorithm, Bernoulli, pp 223–242
28. Howard RA (1968) The foundations of decision analysis. IEEE Trans Syst Sci Cybern 4(3):211–219
29. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13:455–492

30. Schmitz TL, Donalson RR (2000) Predicting high-speed machining dynamics by substructure analysis. CIRP Ann 49(1):303–308
31. Schmitz T, Betters E, Budak E, Yüksel E, Park S, Altintas Y (2023) Review and status of tool tip frequency response function prediction using receptance coupling. Precis Eng 79:60–77
32. Schmitz T, Duncan GS (2005) Three-component receptance coupling substructure analysis for tool point dynamics prediction. J Manuf Sci Eng 127(4):781–790
33. Schmitz T, Cornelius A, Karandikar J, Tyler C, Smith S (2022) Receptance coupling substructure analysis and chatter frequency-informed machine learning for milling stability. CIRP Ann 71(1):321–324
34. Karandikar J, Saleeby K, Feldhausen T, Kurfess T, Schmitz T, Smith S (2022) Evaluation of automated stability testing in machining through closed-loop control and Bayesian machine learning. Mech Syst Signal Process 181:109531
35. Plathottam SJ, Rzonca A, Lakhnori R, Iloeje CO (2023) A review of artificial intelligence applications in manufacturing operations. J Adv Manuf Process e10159

STC O—Production Systems and Organizations

A Modular Framework for Implementing Release Control Policies in Discrete-Event Simulation Models

Marcello Urgo, Walter Terkaj, Aydin Nassehi, and Qunfen QI

Abstract Release control policies support the management of factories by deciding when a job or part can enter a manufacturing system, cell, or a single machine. Effective release control policies are essential to ensure the smooth operation of a complex factory, but modelling and implementing them in performance evaluation models is a rather complex task. Although discrete-event simulation (DES) models can evaluate the performance of systems without imposing constraining hypotheses, the modelling of control mechanisms strongly depends on the specific commercial-off-the-shelf simulation package (CSP). This essay provides insights and a detailed description of implementing release control policies in DES models, leveraging a modular representation of policies and their relations with production systems and plans. Starting from the definition of a controller, modelled after the IEC 61499 standard, and defining how production resources are managed, a statechart-based representation of the control mechanism is generated and linked to a factory data model structured as a modular OWL ontology based on existing technical standards. This unified framework is exploited as a key enabler for generating a DES model of a manufacturing system and its control mechanisms, guaranteeing an unambiguous implementation of control policies not depending only on non-transparent assumptions. A detailed description of the approach is provided for different release control policies (e.g., CONWIP, Kanban), with example implementations in the PlantSimulation and AnyLogic environments, two well-established commercial DES software packages.

M. Urgo (✉)
Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy
e-mail: marcello.urgo@polimi.it

W. Terkaj
Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, Milan, Italy
e-mail: walter.terkaj@stiima.cnr.it

A. Nassehi · Q. QI
University of Bristol, Bristol, United Kingdom
e-mail: aydin.nassehi@bristol.ac.uk

Q. QI
e-mail: qunfen.qi@bristol.ac.uk

T. Tolio (ed.), *CIRP Novel Topics in Production Engineering: Volume 2*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-3-032-04439-6_4

Keywords Control · Simulation · Digital twin

1 Introduction

The control of a manufacturing system plays a crucial role in determining its performance and directly influences metrics such as throughput, cycle time, and resource utilization [12, 16]. Control policies are typically employed to control manufacturing systems and various performance evaluation methods can model control policies with differing level of flexibility.

Analytical methods usually impose stringent conditions on the control of manufacturing systems, with limited customization options for the user [17]. While approximate analytical methods offer more flexibility, they still accommodate only a subset of control policies, and the complexity of the associated mathematical model may quickly increase [7, 15]. Discrete event simulation (DES) models [6, 8] offer the capability to evaluate the performance of systems without imposing constraining hypotheses on the control policies, also leveraging functionalities offered by commercial-off-the-shelf simulation packages (CSP) [24]. Nevertheless, the modelling of control mechanisms strongly depends on specific functionalities and poses serious concerns about the generality of the derived models [10, 13]. As the control policy to be tested changes, small modifications (e.g., refactoring a rule or the associated variables) or, in many cases, more invasive revisions of the DES model are needed [3].

The rigidity of performance evaluation approaches for modelling control policies hinders the optimized design of manufacturing systems, considering that the huge variety of control decisions to be made in a manufacturing system also depends on the actual system configuration [25]. To overcome such limitations, formal modelling approaches can support a modular representation of control policies and their relations with production system configurations and production plans [20, 27, 31]. This is a key enabler for generating and fast reconfiguring performance evaluation models with different control options in a manufacturing system while guaranteeing an explicit definition of control policies that do not depend only on non-transparent assumptions at the implementation level [26, 28, 30].

Section 2 will present a formal modelling approach for a class of control policies, specifically release control policies that determine when a job (part) is allowed to enter a manufacturing system or a portion of it (e.g., a shop, a manufacturing cell, a subset of machines, etc.). Then Sect. 3 will address the representation of control policies in a DES model, while Sects. 4 and 5 will show how it can be done using two commercial tools. The use case presented in Sect. 6 is used to test the simulation of manufacturing system with control policies in Sect. 7.

2 Formal Modelling of Release Control Policies

The formal modelling of release control policies in a manufacturing system, proposed in [31] and expanded in [32], is centred on the concept of a *Controller* that determines how a production system and its resources are managed during manufacturing execution. In particular, the formal model consists of a modular OWL ontology [34] integrating data models such as UML Statechart [4] and W3C SSN/SOSA [9]. Figure 1 shows the classes modelling the controller, while Table 1 lists the subset of relevant modules of the ontology.

The *Controller* operates the manufacturing system while considering and interacting with the following elements:

- *Production Resources* (e.g. workstations, buffers) whose behaviour is managed by the controller. A controller can have an impact on one or more production resources.
- *Control Policy*, i.e. specific rules and algorithms adopted by the controller. A policy typically uses observed variables (e.g. buffer level) related to production resources.

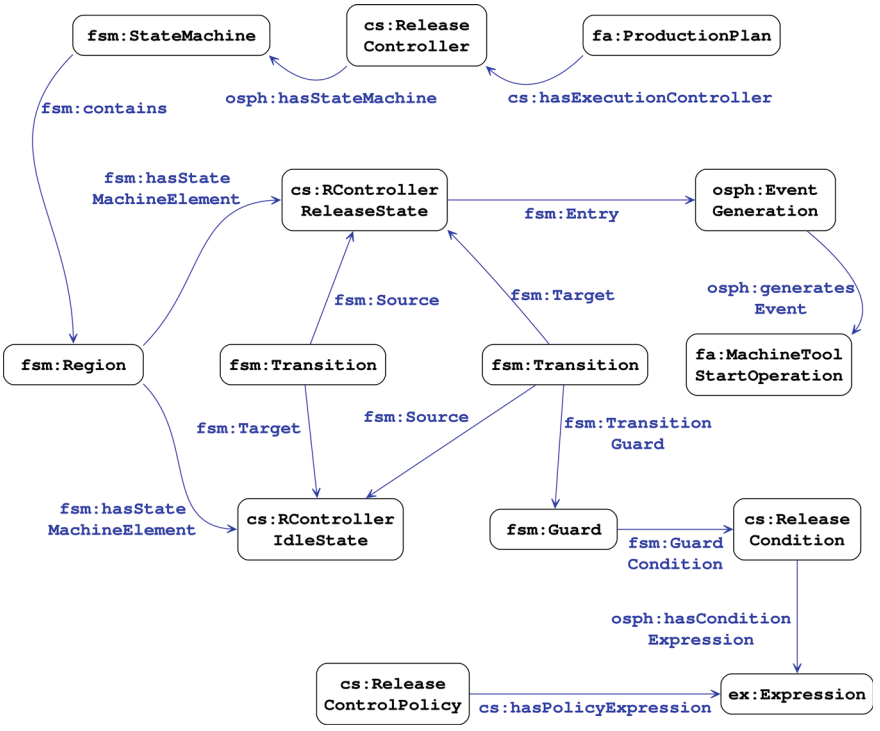


Fig. 1 Core ontology to model the release controller. Rounded boxes represent classes identified by their IRIs. Blue solid arrows denote object properties identified by their IRIs, linking classes in their `rdfs:domain` to classes in their `rdfs:range`

Table 1 List of ontology modules with prefix names. All modules are available online at the same address, except *fsm* (*fsm* module can be downloaded from https://w3id.org/ontoeng/fsm)

Ontology module	Prefix	Prefix IRI of ontology module
controlSystem	cs:	https://w3id.org/ontoeng/controlSystem#
expression	ex:	https://w3id.org/ontoeng/expression#
factory	fa:	https://w3id.org/ontoeng/factory#
fsm	fsm:	http://www.learninglab.de/\inlinealttext\\sim$\sim$dolog/fsm/fsm.owl#
osph	osph:	https://w3id.org/ontoeng/osph#
sosa	sosa:	https://www.w3.org/ns/sosa/
ssn	ssn:	https://www.w3.org/ns/ssn/

- *Production Plan* that consists of scheduled activities assigned to production resources within a given planning horizon. The chosen controller enforces the execution of the plan.

A production plan (`fa:ProductionPlan`) must select which controller is activated for its manufacturing execution. The release controller (`cs:ReleaseController`) is a specialization of the generic controller that manages the release of parts to a production resource. The release controller is characterized by a state machine (class `fsm:StateMachine`) that can be decomposed into several orthogonal regions (class `fsm:Region`). Each region models the control behaviour of a specific production resource (e.g. a machine tool) by containing two states: a release state (class `cs:RControllerReleaseState`) that allows parts to be released to the controlled production resource and an idle state (class `cs:RControllerIdleState`) that inhibits the release of parts.

An example is provided in Fig. 2 for a generic release controller. The state machine of the controller (`Controller_stM`) is composed of two regions (`C_M1_region` and `C_M2_region`) aimed at controlling two production resources (`M1` and `M2`, respectively).

A release control policy (class `cs:ReleaseControlPolicy`) is based on an expression (class `ex:Expression`) that defines the release condition (class `cs:ReleaseCondition`) to be satisfied to trigger the release transition (class `fsm:Transition`) through its release guard (state `fsm:Guard`). Although this expression can be arbitrarily complex, for release control policies it is usually based on a set of observable variable (e.g., the number of parts in a set of buffers or workstations) compared with a constant value acting as a threshold.

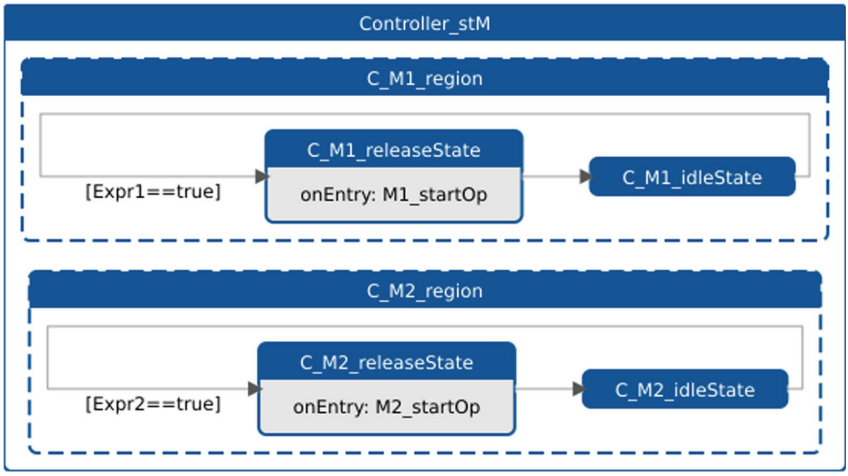


Fig. 2 UML statechart of a Release Controller of two machines

Referring to Fig. 2 and machine M1, as soon as the release condition is satisfied (i.e., `Expr1==true`), the controller leaves the idle state (`C_M1_idleState`) and enters the release state (`C_M1_releaseState`).

When the controller enters the release state, an entry action (class `osph:EventGeneration`) is started to generate an event (class `fsm:Event`) that will trigger the actual release of a part when the state machine of the controlled production resource intercepts this event. If the controlled production resource is a machine tool, then the event generated by the release controller will specify that an operation can be started (class `fa:MachineToolStartOperation`). In Fig. 2, the triggered event tells machine `M1` to start the operation (`onEntry: M1_startOp`). As soon as the entry action is run to completion, the release controller goes back to the idle state (`C_M1_idleState`).

3 Modelling Control Policies in DES Models

Implementing control policies in a discrete event simulation (DES) environment involves designing rules and mechanisms to manage the flow of entities (e.g., parts) through a system based on the occurrence of discrete events (e.g., the arrival of a new entity or the completion of a task). This has to pass through a sequence of modelling decisions and steps, strongly influenced by the developing environment and available functionalities.

The first step is defining a model for the manufacturing system to be implemented. This model should include all relevant entities, resources, processes, and flows. Ensure it accurately reflects the operational dynamics and constraints of your real-world system.

Once the model for the system has been defined, control mechanisms have to be designed and plugged into the existing model. Control policies generally take into consideration some variables (e.g., the number of parts in a buffer, the state of a production resource, etc.) to support decisions (e.g., the release of parts, the priority of the parts in a buffer, the routing of parts in the system, the assignment to production resources, etc.). The decision-making process is triggered by some events in the model (e.g., the arrival of a part, the completion of an activity, etc.). DES model intended to simulate the behaviour of policies must be capable of:

- providing and storing the values of variables and parameters that the policies must observe.
- model and intercept specific events that trigger the decisions.
- allowing the user to implement policies, e.g., through scripting languages.
- providing a set of control mechanisms interacting with the inner behaviour of the model to support the implementation of the decisions taken by the policies.

As the panorama of available DES packages is quite large, the focus will be on two of the most common packages, e.g., Tecnomatix Plant Simulation (Sect. 4) and Anylogic (Sect. 5). We will show how release mechanisms and the related control policies can be implemented for these two environments, matching the modular reference data model defined in Sect. 2.

4 Tecnomatix Plant Simulation

Tecnomatix Plant Simulation is a software solution developed by Siemens Digital Industries Software [21]. It is an integrated development environment for discrete event simulation, supporting the user in modelling, simulating, analyzing, and optimizing general systems, with a special focus on manufacturing systems.

With Tecnomatix Plant Simulation, users can create digital models of entire production facilities, including machines, material handling systems, logistics, and human resources. These models allow users to visualize and simulate various scenarios to understand how changes in the manufacturing environment might affect productivity, efficiency, and other key performance indicators.

4.1 Modelling a Production System in Plant Simulation

Plant Simulation provides a set of objects to model a flow of parts and their processing, i.e., stations, buffers, sources and drains. The *Station* object allows the modelling of

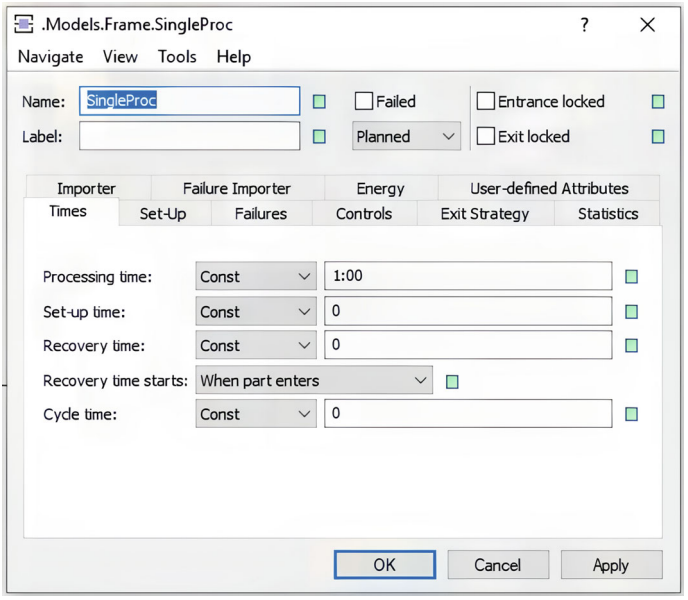


Fig. 3 Dialog window for the Station object in PlantSimulation

a server, e.g., a machine, while the *Buffer* object is used to model the behaviour of buffers.

Objects in this class allow the possibility to define their behaviour for a wide range of characteristics (e.g., processing time, set-up times, failures, etc.). These options also allow managing the way parts enter and exit the object (Fig. 3).

For both classes of objects, i.e., a station and a buffer, the mechanism supervising the arrival and exit of entities is managed through a set of options:

Entrance locked. It provides the possibility to close the entrance of the object. As the checkbox is selected, entities in the process object will complete their processing, but no additional entity will be allowed to enter until the check box is cleared.

Exit locked. It provides the possibility to close the exit of the object. As the checkbox is selected, entities in the object completing their processing and moving towards the successor object will not be allowed to move on. They will enter the Exit Blocking List of the object, and when the checkbox is cleared, the exit will be unlocked, and they will be able to move forward according to their position in the Exit Blocking List.

Plant simulation offers a flexible way of implementing complex behaviour of the objects, allowing the user to use a piece of code called *Method*. Specific events can trigger these scripts. Concerning the entrance and exit of parts in the objects, this can be defined through the interface in Fig. 4, through the following options:

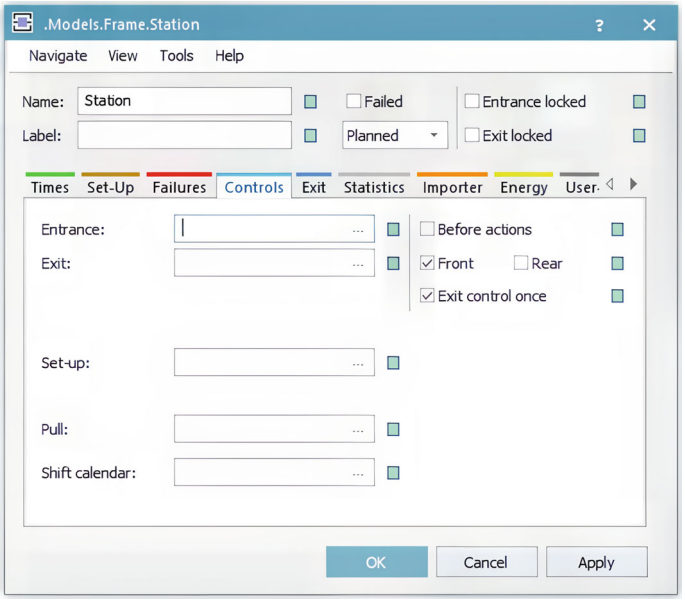


Fig. 4 Dialog window for the Station object in PlantSimulation

Entrance. Provides the possibility of executing a script when an entity enters the object. Depending on the behaviour of the object, an entity approaching its entrance can just enter the object, or its entering could be blocked, e.g., if the entrance is locked. To cope with this behaviour, the control mechanism for the entrance can have an additional option:

Before actions. If this option is activated, the script is executed only if the entity is actually entering the object.

Exit. Provides the possibility of executing a script when an entity exits the object. Also, in this case, the actual exit of an entity could depend on other mechanisms in the simulation model. To this aim, additional specifications can be provided to the control to define the specific sequence of events defined in the control mechanism. In the jargon of Plant Simulation, moving entities have a *front* and a *rear*. Thus, when exiting an object, the front portion first exits, followed by the rear portion, if the exit is actually completed. Thus:

Front. This option specifies that the script associated with the exit of an entity must be called when an entity is ready to exit, before the exit action has actually been completed. This control mechanism overrides the standard transfer control of Plant Simulation, hence, it must also take care of moving the entity if this has to happen.

Rear. Under this option, the indicated script is called when the rear portion of the entity is exiting the object. This means that the move action has been

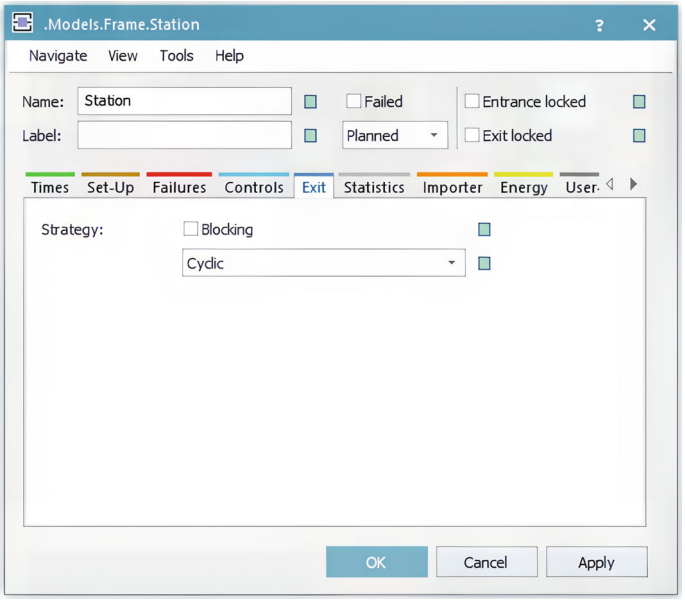


Fig. 5 Dialog window for the station object in PlantSimulation

already executed, thus the entity will surely leave the object towards its new destination. In this case, the control can block the entity, but as soon as the block is deactivated, it will proceed further towards its destination.

Exit control once. This option specifies whether the exit control has to be executed only once or more. It could be executed multiple times if a part is exiting the object, but a specific action blocks it. If the check box is cleared, the exit control is executed again when the part is unblocked.

Plant Simulation also provides a blocking control for the Station object (see Fig. 5). If the checkbox Blocking is selected, the entity is moved to the designated successor, and if that successor is not ready to receive it, the entity will be blocked. If this happens, the entity remains in the object and is listed in the Forward Blocking List of the designated successor. On the contrary, if the check box is cleared, the entity is moved only if any of its successors can receive it, and if none of them can receive it, it enters the forward blocking list of all successors

4.2 Implementing Release Control Policies in Plant Simulation

Release policies are enforced by taking advantage of the possibility to lock the entrance of an object upon control method is called.

Let us consider a general release control policy, controlling the release of entities to a machine $M1$ under the constraint that a function f of the values of attributes belonging to a subset of objects A is respected. In general terms:

$M1$: controlled object;
A: set of observed objects;
$f(...)$: function of attributes of the objects in A; if the return value is true, then it is possible to release entities to M1, if it is false then it is not possible.

At the beginning of a simulation, all the objects in the simulation model are initialized according to the following rules:

```
simobj.EntranceCtrl := ref(enterCtrl);
simobj.EntranceLocked := false;
simobj.ExitCtrlFront := true;
simobj.ExitCtrlRear := false;
simobj.ExitCtrlOnce := false;
simobj.exitctrl := ref(exitCtrl);
```

The script `enterCtrl` is called when a part enters an object. This script is also in charge of updating the values involved in the expressions linked to policies. If any of those expressions is greater than the associated thresholds, then the entrance of the machine associated with that control policy is locked, meaning that no part can be released on that machine until further notice.

The actual release of parts is managed by the `exitCtrl` script. Every time a part is about to leave an object (e.g., a station or a buffer), this script checks its destination and verifies that the release on that object is permitted. If yes, the part is authorized to proceed and leave the object. Otherwise, nothing happens. As the exit control is operated before the part actually leaves the object (`ExitCtrlFront := true`), and it can be run multiple times for the same part (`ExitCtrlOnce := false`) guarantees that the entity will not remain stuck in the object. As soon as an entity leaves an object, the release condition is checked again and, if respected, the entrance of the controlled objects (e.g., $M1$) is opened again.

It must be noted that the release condition is based on the function of the number of entities in the controlled objects, as for the vast majority of release policies. In this case, an entity leaving an object is the only event that can affect the release condition; thus, it triggers the policy check and, in this case, the opening of the entrance for the controlled machine.

5 AnyLogic

AnyLogic is simulation modelling software developed by the AnyLogic company [29]. It enables the modelling of complex environments using three main simulation methodologies: system dynamics, discrete event, and agent-based modelling. This versatility makes AnyLogic a powerful tool for tackling challenges in manufacturing, healthcare, logistics, supply chain management, transportation and more [5, 11, 14, 18, 19, 22].

Key features of AnyLogic include a wide range of built-in libraries and modules that cater to various industry needs, and the ability to extend models with custom Java code. It also supports cloud services, integration with other software tools and data sources, such as SQL databases, machine learning models and AI tools (e.g. Python scripts or TensorFlow models), and simulation optimisation software such as OptQuest, enhancing its utility for comprehensive analysis and scenario testing.

5.1 Modelling a Production System in AnyLogic

AnyLogic provides a wide range of process modelling libraries and tools, to model the flow of parts and their processing, i.e. Queue, Delay, Service, and Source. The *Queue* block is used to model the behaviour of buffers. Both *Delay* and *Service* blocks allow the modelling of a server. The *Delay* block models simple, fixed-time delays and is used for straightforward processes such as mandatory waiting. The *Service* block is designed for complex scenarios involving resource management and queuing, ideal for processes where the handling of entities depends on available resources.

5.1.1 Delay

Process time: *Delay* block delays agents for a given amount of time, i.e. process time. Entities exit the block after the delay time, which is evaluated dynamically. It can be a fixed duration, a statistical distribution, or a data-driven model, and may depend on the agent or on any other conditions. Multiple agents (up to the given *Delay* capacity) can be delayed simultaneously and independently. Users can also programmatically control the block's corresponding function. An example of delay time definition, as shown in Fig. 6, when multiple types of agents are being processed in the block and processing time differs by agent type, use `agent instanceof partTypeA? opA05:opB03` to determine the processing time.

Capacity: The capacity of a *Delay* block can be unlimited with *Maximum capacity* box clicked, a constant number, or change dynamically by executing the script in the *Capacity* box. Entities can enter the *Delay* block as soon as they arrive, if the block is not at capacity. For blocks with limited capacity, users can configure conditions

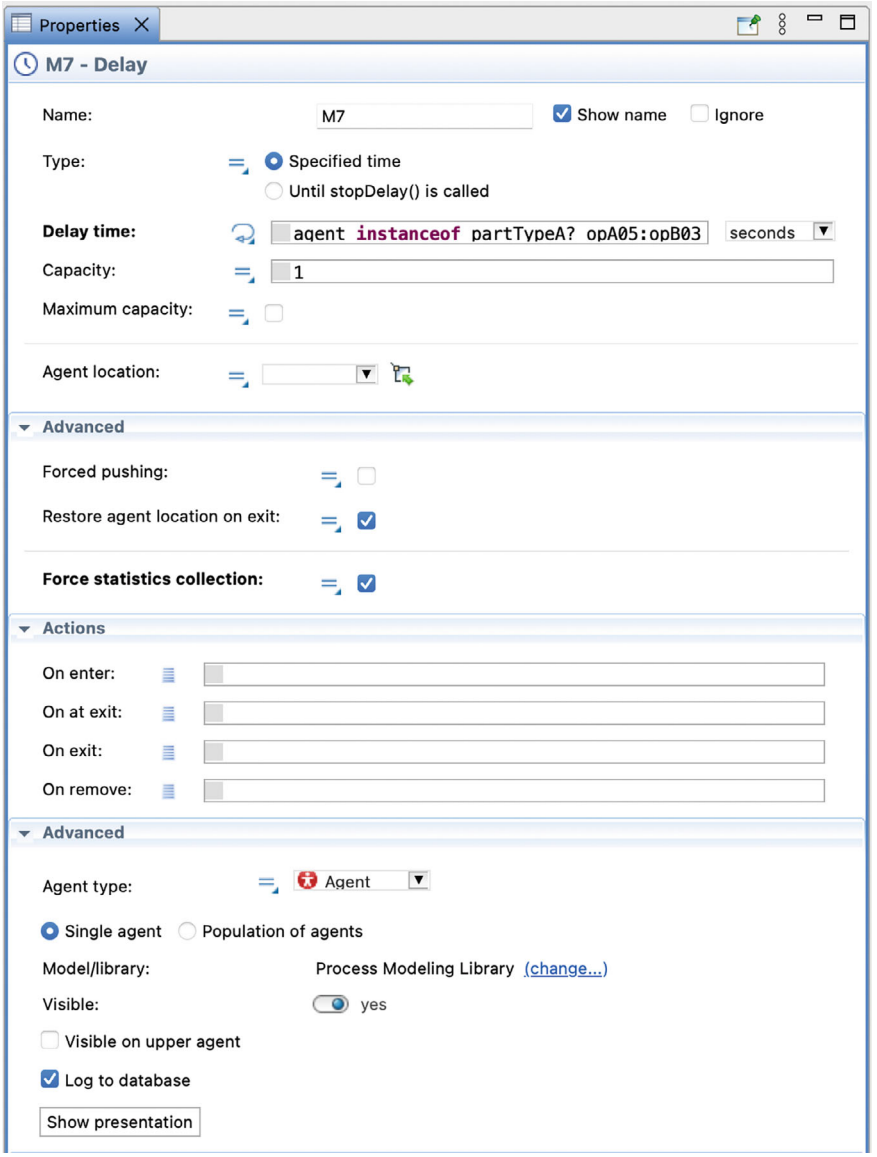


Fig. 6 Properties window for the delay block in AnyLogic

under which entities are accepted or rejected. For example, only entities meeting certain criteria may be allowed to initiate the delay. If the capacity of the block is adjusted dynamically, and the current number of agents exceeds this new limit, the block will allow those inside to finish their delay times but will prevent new agents from entering until the number falls below the new capacity.

Actions: allow configuration on customised behaviour and the processing of agents. These Actions include *On Enter*, *On Exit*, *On Remove*, and *On At Exit*. Each of these actions is triggered by different events and can be used to execute specific code affecting the agent, the delay block itself, or other parts of the model. *On Enter* is used for initialising or modifying agent properties at the start of their delay, such as recording entry time or adjusting attributes that influence their processing. *On Exit* is used to update the agent's state, or to prepare them for subsequent stages in the process, such as updating statistics or performing necessary logging. *On At Exit* allows for updates to an agent's state at a precise moment while still in the delay. *On Remove* handles situations where an agent's delay is interrupted, allowing for the reversal of initial modifications, updating interrupted process statistics, or freeing up allocated resources.

There are other advanced features, for example *Forced pushing*. If the option is selected, agents are immediately pushed forward upon completing their processing at the block, regardless of the state of the next block. If the option is not selected, agents are only pulled when the succeeding block is ready to accept another agent. The succeeding block requests the agent from the previous block, and only then does the agent move forward.

5.1.2 Statechart

In AnyLogic, a *Statechart* is an advanced tool that models the behaviour of agents or components in a system by defining various states and the transitions between those states. It effectively represents a finite state machine, where each state captures a specific condition or status of an agent, and transitions are the changes that occur due to events or conditions. Statecharts provide a graphical and intuitive way to design complex behaviors, making them particularly useful in simulation models that require detailed behavioral control.

A *Statechart* in AnyLogic includes several key elements:

- **States:** distinct conditions in which an agent or component of the system can exit;
- **Transitions:** links between states, triggered by specific events or conditions that cause the agent to move from one state to another;
- **Events:** external or internal occurrences that trigger transitions, which can be time-based, action-based, or condition-based;
- **Actions:** activities that occur either when entering or exiting a state or during the transition between states. Actions might include sending messages, updating variables, starting timers, etc.

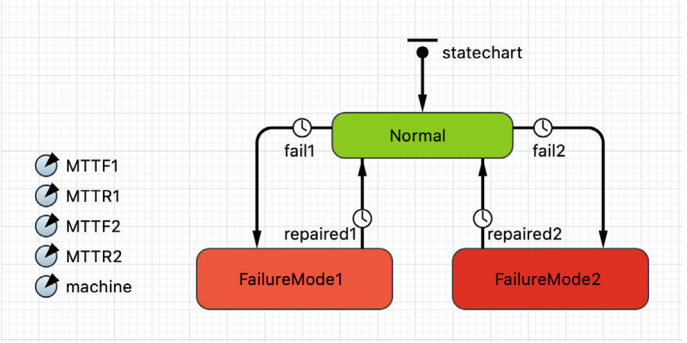


Fig. 7 State chart of MachineAgent in AnyLogic

Statecharts are particularly effective in modelling the behaviour of various components within a production system in AnyLogic. Here's how they can be applied:

1. **Machine Behavior:** Each machine in a production line can be modeled with a statechart, depicting states such as Idle, Operating, and Broken. Transitions between these states can be triggered by events such as the completion of a task or a failure occurrence.
2. **Process Control:** Statecharts can control the logic of more complex production processes. For example, a production system might need to switch between different operating modes based on production targets, availability of raw materials, or maintenance schedules.
3. **Human Resources**: Workers or robotic agents can also be modeled using statecharts, representing their shifts, tasks, breaks, or interactions with machinery.
4. **System Dynamics:** At a higher level, statecharts can govern the overall workflow of a production system, managing transitions between states like Start-Up, Normal Operation, Shutdown, and Emergency Stop.

Figure 7 indicates an example of using statechart to implement two failure modes for a set of Delay blocks. In this case, a custom agent type called *MachineAgent* is created. This agent type is configure with parameters such as *MTTF1*, *MTTR1*, *MTTF2* and *MTTR2* for both failure modes, along with a parameter named *machine* to link the agent type with the Delay block. Within *MachineAgent*, a state chart, as shown in Fig. 7, outlines three states: *Normal*, *FailureMode1* and *FialureMode2*. Transitions among the three states–'fail1', 'fail2', 'repaired1' and 'repaired2' are triggered by the specific MTTF and MTTR timeouts. As indicated in Fig. 8, during each fail transition, the machine ceases operation via `machine.suspend()`, and upon each repaired transition, it resumes operation with `machine.resume()`.

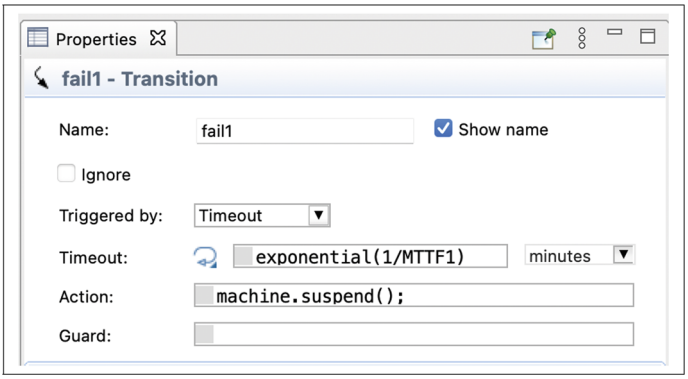
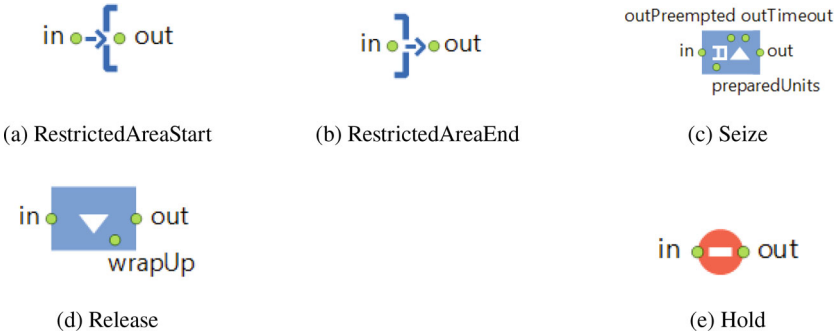


Fig. 8 Transition in state chart of MachineAgent in AnyLogic

5.2 Implementing Release Control Policies in AnyLogic

AnyLogic provides a wide range of methods to implement release control policies. Each of these methods offers unique capabilities and can be selected or combined based on the specific needs and complexities of a simulation model.

1. *Using RestrictedAreaStart and RestrictedAreaEnd Blocks:* A pair of *RestrictedAreaStart* block (Fig. 9a) and *RestrictedAreaEnd* block (Fig. 9b) manage how many entities can be in a defined area at one time. This method is for managing entity count within spatially or resource-restricted areas, ideal for scenarios such as in environments with limited capacity like hospital rooms, manufacturing areas, or public spaces.
2. *Using Seize and Release Blocks:* Manage entity flow by seizing (Fig. 9c) and releasing (Fig. 9d) resources, which control when entities can proceed. It is effective in models where entities' progression is dependent on resource availability, simulating real-world scenarios like staffed operations or machine usage.
3. *Conditional Holding Using Hold Blocks:* Employ *Hold* blocks (Fig. 9e) to pause entities until certain conditions are met. The method is useful for delay implementations where the release of entities depends on specific triggers or system conditions, such as waiting for an available service station, Shutdown, and Emergency Stop.
4. *Statecharts for Complex Logic:* Use statecharts to create sophisticated control logic based on multiple conditions and states. Statechart is suitable for complex decision-making processes where intricate rules or multi-condition scenarios govern entity flow.
5. *Programmatic Control Using Java Code:* Utilise Java code to dynamically control the flow based on conditions, system states, or changes in simulation variables. It is applicable for scenarios requiring highly dynamic and conditional control over entity flow, adaptable to complex systems.



(a) RestrictedAreaStart (b) RestrictedAreaEnd (c) Seize

(d) Release (e) Hold

Fig. 9 Blocks in AnyLogic that implement release control

5.2.1 Restricted Area Method

When the release control policies are implemented on *Delay* and *Queue* blocks where entities are processed, using *RestrictedAreaStart* and *RestrictedAreaStart* blocks is a simple way to manage how entities enter and exit *Delay* and *Queue*, especially when you need to manage capacity or implement specific control over the flow of entities in a model. These blocks help control the number of entities within a particular section of the model.

To set up the release policy, place *RestrictedAreaStart* block before the *Delay* or *Queue* block. This is the point where control over entering the restricted area begins. Then, place *RestrictedAreaStart* block after the *Delay* or *Queue* block. This marks the end of the controlled area. In the Properties window of the *RestrictedAreaStart* block, as shown in Fig. 10, set the capacity of the restricted area. The value defines how many entities can be within the restricted area (including *Delay* or/and *Queue* blocks) at any one time. The users can dynamically adjust the capacity of the restricted area based on simulation conditions. For instance, if additional resources become available or unavailable in response to other conditions in the model, you can change the capacity of the restricted area through code. If there are multiple possible paths through the restricted area, you can implement policies where certain types of entities have preferential access to the resources or space within the restricted area. The *RestrictedAreaStart* block typically doesn't require much configuration as its primary function is to mark the exit point of the restricted area and decrement the count of entities in the area.

6 Use Case

The application of the proposed approach to generate DES models (Sect. 4 and 5) can be demonstrated by employing a reference use case named *ControlPolicy03*, which formalizes the configuration and behaviour of a manufacturing system.

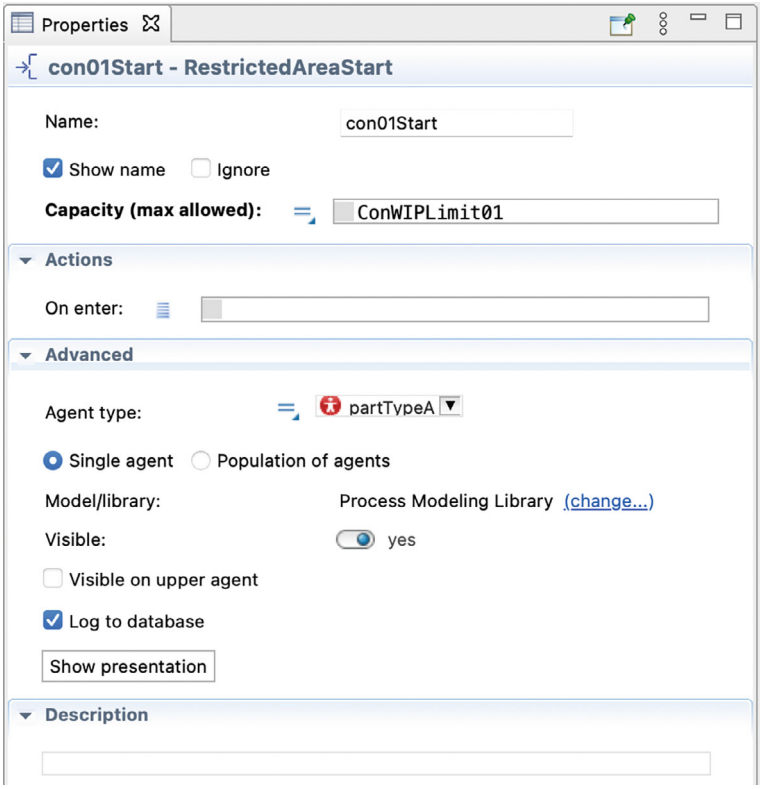


Fig. 10 Properties window for the RestrictedAreaStart block in AnyLogic

The configuration of the manufacturing system, represented in Fig. 11, is defined as follows:

- Eight machines (*M1*...*M8*) with associated failure modes (Table 2). Both the time to failure and the time to repair follow exponential distributions.
- Nine inter-operational buffers (*B1*...*B9*) are placed along the line. Each buffer has a capacity (Table 3).
- Two part types are produced (*parttypeA*, *parttypeB*).
- Each part type is associated with a process plan decomposed into sequenced manufacturing operations. Each operation is characterized by a deterministic processing time and is assigned to a machine. Table 4 lists the operations of process plan *pplanA* for *parttypeA*, whereas Table 5 the operations of process plan *pplanB* for *parttypeB*.

The manufacturing system operates as follows:

- A production plan (*prodPlan*) spanning 24 h is scheduled. The arrival rate of raw parts is set at 6 [parts/min] for both *parttypeA* and *parttypeB*.

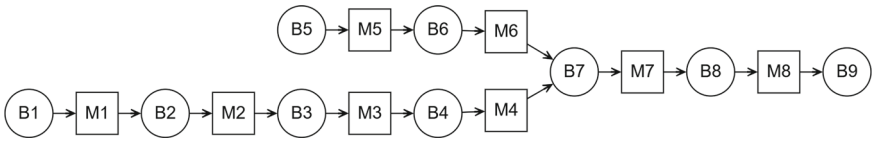


Fig. 11 Graphical representation of the manufacturing system of use case *ControlPolicy03*, where squares represent machines and circles represent buffers

- Three control policies (*conwip01*, *conwip02*, *conwip03*) of type CONWIP [23] are defined to regulate the work-in-progress of the system (Table 6). A CONWIP policy allows the release of a part if the total number of parts in the observed resources (i.e. machines and buffers) does not exceed the threshold value.
- The control policies are employed by the release controller *c01* to determine whether a part should be released. Specifically, the release of parts on machines

Table 2 Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR) of the machine failure modes

Machine	Failure mode 1		Failure mode 2	
	MTTF [min]	MTTR [min]	MTTF [min]	MTTR [min]
M1	101.02	13.85	52.56	9.63
M2	79.48	5.52	72.95	11.51
M3	81.10	10.82	116.30	3.89
M4	91.62	4.74	91.08	10.90
M5	84.01	9.39	78.03	8.59
M6	67.00	5.23	62.32	3.47
M7	83.23	10.37	79.27	12.19
M8	65.63	12.03	59.48	5.72

Table 3 Buffer capacity

Buffer	Capacity
B1	50
B2	6
B3	8
B4	6
B5	10
B6	6
B7	10
B8	14
B9	50

Table 4 Operations in process plan *pplanA*

Operation	Processing time [s]	Successor	Assigned to machine
opA01	4	opA02	M01
opA02	12	opA03	M02
opA03	14	opA04	M03
opA04	12	opA05	M04
opA05	8	opA06	M07
opA06	9		M08

Table 5 Operations in process plan *pplanB*

Operation	Processing time [s]	Successor	Assigned to machine
opB01	14	opB02	M05
opB02	16	opB03	M06
opB03	6	opB04	M07
opB04	8		M08

Table 6 CONWIP policies

CONWIP policy	Observed resources	Threshold	Controlled machine
conwip01	M1, M2, M3, M4, B2, B3, B4	18	M1
conwip02	M5, M6, B6	6	M5
conwip03	M7, M8, B8	12	M7

M1, *M5*, and *M7* is controlled according to the policies *conwip01*, *conwip02*, and *conwip03*, respectively.

The behaviour of machines and release controllers are both modelled as UML statecharts [1]. This enables linking a release controller with a controlled machine by defining trigger events. For instance, when the condition of *conwip01* is satisfied (i.e., $[M1\_wip + M2\_wip + M3\_wip + M4\_wip + B2\_lev + B3\_lev + B4\_lev <= 18]$) statechart *c01_stM* of controller *c01* (in Fig. 12) enters the release state *c01_M1_workComp_relSt* that on entry starts an action generating the event *M1_startOp*. In turn, this event triggers the transition from the idle state *M1_idle_opA01* to the working state *M1_opA01* in the statechart *M1_stM* of *M1* (see Fig. 13).

The use case has been modelled as an OWL ontology according to a specific factory data model [2, 31] and is available online.[1] The W3C standard language SPARQL [33] can be adopted to develop SPARQL queries to extract data from ontologies and support a semi-automatic generation of simulation models.

[1] https://difactory.github.io/repository/ontoeng/UC/ControlPolicy03.ttl.

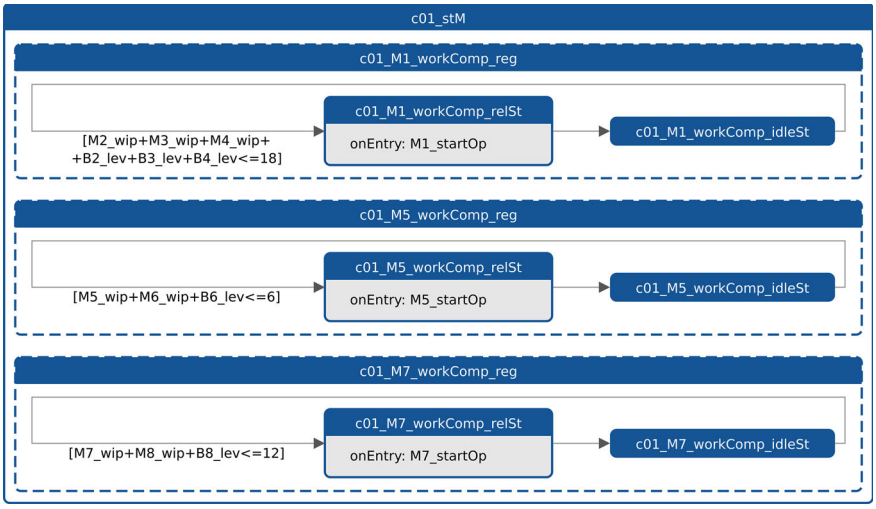


Fig. 12 UML statechart of release controller *c01*

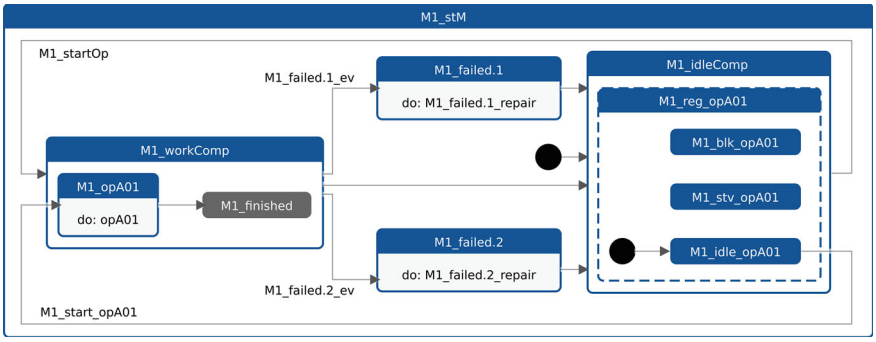


Fig. 13 UML statechart of machine *M1*

7 Implementation, Testing and Results

The selected use case has been implemented using the two DES environments, PlantSimulation and Anylogic. As described in Sects. 4 and 5, the two environments provide different classes of objects and approaches to implement control mechanisms; thus specific design choices have been made, leading to two simulation models depicted in Fig. 14 for Tecnomatix PlantSimulation, and Fig. 15 for Anylogic.

The models have been used to run simulation experiments and collect the results to compare their behaviour. An analysis was carried out comparing the average number of parts in each factory object, namely machines and buffers. The results for the two models are reported in Table 7, showing a good but not perfect alignment.

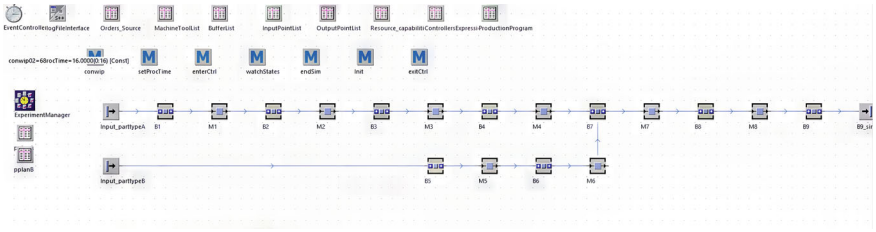


Fig. 14 Implementation of the use case in Plant Simulation

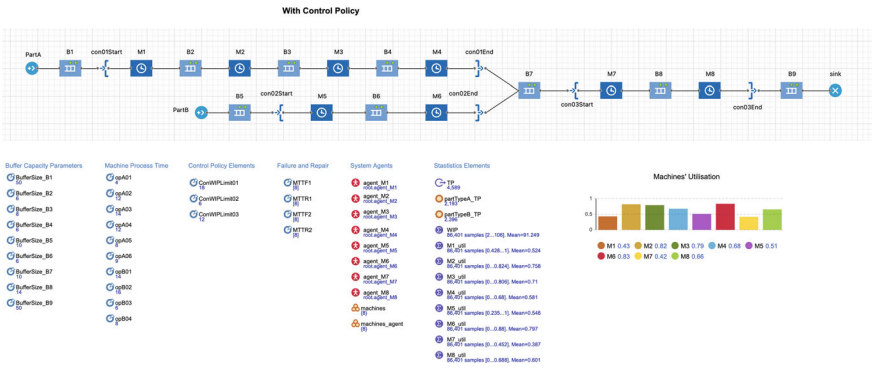


Fig. 15 Implementation of the use case in Anylogic

Although small, the differences between the two exist, and they are probably due to the internal mechanisms implementing failures and queue management policies for different part types staying in the same buffer, in the two simulation environments.

Finally, the average number of parts in the objects observed by the different control policies are reported in Table 8.

8 Conclusions

In this work, we proposed a structured and standardized approach for implementing release control policies in a discrete event simulation (DES) environment. This approach was validated and tested in two DES environments, AnyLogic and Plant Simulation, demonstrating the capability to obtain models that differ in terms of architecture and implementation details due to the distinct functionalities provided by each DES environment. However, the models were aligned in terms of results. Despite this alignment, some misalignment emerged due to how to control policies leverage the inner mechanisms of the simulation models, such as queue management rules and the modelling and simulation of failures (e.g., failure before/after service).

Table 7 Comparison of the results of the two simulation models concerning statistics on the number of parts in the factory objects

	Number of parts in factory objects								
	Anylogic			PlantSimulation			Difference		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
M1	0.43	0	1	0.42	0	1	0.01	0	0
M2	0.82	0	1	0.82	0	1	0.00	0	0
M3	0.79	0	1	0.78	0	1	0.01	0	0
M4	0.68	0	1	0.69	0	1	–0.01	0	0
M5	0.51	0	1	0.52	0	1	–0.01	0	0
M6	0.83	0	1	0.88	0	1	–0.04	0	0
M7	0.42	0	1	0.58	0	1	–0.16	0	0
M8	0.66	0	1	0.64	0	1	0.02	0	0
B1	49.70	0	50	49.52	0	50	0.18	0	0
B2	3.09	0	6	2.68	0	6	0.40	0	0
B3	5.86	0	8	5.54	0	8	0.32	0	0
B4	3.12	0	6	3.11	0	6	0.01	0	0
B5	9.99	0	10	9.99	0	10	0.00	0	0
B6	3.69	0	6	3.86	0	6	–0.17	0	0
B7	5.03	0	10	6.09	0	10	–1.06	0	0
B8	5.63	0	12	4.56	0	12	1.08	0	0
B9	0.00	0	1	0.00	0	1	0.00	0	0

Table 8 Average number of parts for CONWIP policies

	Anylogic	Plant simulation
conwip01	14.35	13.62
conwip02	5.04	5.26
conwip03	6.71	5.77

These discrepancies warrant further analysis. The models and data on the described use case are available in a public data repository available online.[2]

Acknowledgements This research has been partially funded by the EU Horizon Europe programme under GA 101058505, 101092073, 101138930, and Royal Society International Exchanges IEC\R2\212033.

[2] https://github.com/murgo-polimi/CNTPE-O25.

References

1. About the Unified Modeling Language Specification Version 2.5.1. https://www.omg.org/spec/UML/2.5.1/About-UML
2. Berardinucci F, Colombo G, Lorusso M, Manzini M, Terkaj W, Urgo M (2022) A learning workflow based on an integrated digital toolkit to support education in manufacturing system engineering 63:411–423. https://doi.org/10.1016/j.jmsy.2022.04.003
3. Buy U (2018) Control reconfiguration of discrete event systems with dynamic control specifications
4. Dolog P (2004) Model-Driven Navigation Design for Semantic Web Applications with the UML-Guide. In: Proc. of ICWE, pp. 75–86. Munich, Germany
5. Ershova I, Totmyanin A (2023) Simulating production tasks using the anylogic program. Autom Model Design Manag 2023:32–39
6. Furian N, O'Sullivan M, Walker C, Vössner S, Neubacher D (2015) A conceptual modeling framework for discrete event simulation using hierarchical control structures. Simul Model Pract Theory 56:82–96
7. Gershwin SB, Werner LM (2007) An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers. Int J Prod Res 45(14):3085–3111. https://doi.org/10.1080/00207540500385980
8. Golbasi O, Turan MO (2020) A discrete-event simulation algorithm for the optimization of multi-scenario maintenance policies. Comput Ind Eng 145:106514. https://doi.org/10.1016/j.cie.2020.106514
9. Janowicz K, Haller A, Cox SJ, Le Phuoc D, Lefrançois M (2019) Sosa: A lightweight ontology for sensors, observations, samples, and actuators. J Web Semant 56:1–10
10. Kamach O, Chafik S, Pietrac L, Niel E (2004) Generalization of des multi-modeling. IFAC Proceed 37(4):271–278
11. Karakikes I, Hofmann W, Mitropoulos L, Savrasovs M (2019) Integrating logistics and transportation simulation tools for long-term planning. In: Data Analytics: Paving the Way to Sustainable Urban Mobility: Proceedings of 4th Conference on Sustainable Urban Mobility (CSUM2018), 24-25 May, Skiathos Island, Greece. Springer, pp 807–814
12. Kuhnle A, May MC, Schäfer L, Lanza G (2022) Explainable reinforcement learning in production control of job shop manufacturing system. Int J Prod Res 60(19):5812–5834. https://doi.org/10.1080/00207543.2021.1972179
13. Kuipers B (1994) Qualitative reasoning: modeling and simulation with incomplete knowledge. MIT press
14. Magilton D, Mahdavi A (2017) Multimethod simulation and analytics for the entire business lifecycle. In: 2017 Winter Simulation Conference (WSC), pp 4409–4409. https://doi.org/10.1109/WSC.2017.8248146
15. Magnanini MC, Tolio T (2020) Restart policies to maximize production quality in mixed continuous-discrete multi-stage systems. CIRP Ann 69(1):361–364. https://doi.org/10.1016/j.cirp.2020.03.021
16. Magnanini MC, Tolio T (2020) Switching-and hedging-point policy for preventive maintenance with degrading machines: application to a two-machine line. Flex Serv Manuf J 32:241–271
17. Mascolo MD, Frein Y, Dallery Y (1996) An analytical method for performance evaluation of kanban controlled production systems. Oper Res 44(1):50–64
18. Özceylan E, Çetinkaya C, Demirel N, Sabırlıoğlu O (2017) Impacts of additive manufacturing on supply chain flow: A simulation approach in healthcare industry. Logistics 2(1):1
19. Possik J, Gorecki S, Asgary A, Solis AO, Zacharewicz G, Tofighi M, Shafiee MA, Merchant AA, Aarabi M, Guimaraes A, et al (2021) A distributed simulation approach to integrate anylogic and unity for virtual reality applications: Case of covid-19 modelling and training in a dialysis unit. In: 2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT). IEEE, pp 1–7

20. Qi Q, Terkaj W, Urgo M, Jiang X, Scott P (2022) A mathematical foundation to support bidirectional mappings between digital models: an application of multi-scale modelling in manufacturing. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **478**(2264). https://doi.org/10.1098/rspa.2022.0156
21. Siemens Digital Industries Software: Tecnomatix Plant Simulation: Discrete-Event Simulation Software (2024). https://plm.sw.siemens.com/en-US/tecnomatix/plant-simulation-software/. Version 16.0
22. Sokolov S, Antonova A (2024) Application of simulation modeling in ship construction processes. Intellectual Technologies on Transport **0**, 44–51
23. Spearman ML, Woodruff DL, Hopp WJ (1990) CONWIP: A pull alternative to kanban. Int J Prod Res 28(5):879–894. https://doi.org/10.1080/00207549008942761
24. Taylor SJ, Mustafee N, Turner SJ, Pan K, Strassburger S (2009) Commercial-off-the-shelf simulation package interoperability: Issues and futures. In: Proceedings of the 2009 Winter Simulation Conference (WSC), pp. 203–215. IEEE
25. Terkaj W, Annoni M, Martinez B, Pessot E, Sortino M, Urgo M (2024) Digital twin for factories: Challenges and industrial applications. Lecture Notes in Mechanical Engineering pp. 255–274. https://doi.org/10.1007/978-3-031-41163-2_13
26. Terkaj W, Gaboardi P, Trevisan C, Tolio T, Urgo M (2019) A digital factory platform for the design of roll shop plants. CIRP J Manuf Sci Technol 26:88–93. https://doi.org/10.1016/j.cirpj.2019.04.007
27. Terkaj W, Qi Q, Urgo M, Scott P, Jiang X (2021) Multi-scale modelling of manufacturing systems using ontologies and delta-lenses. CIRP Ann 70(1):361–364. https://doi.org/10.1016/j.cirp.2021.04.047
28. Terkaj W, Tolio T, Urgo M (2015) A virtual factory approach for in situ simulation to support production and maintenance planning. CIRP Ann Manuf Technol 64(1):451–454. https://doi.org/10.1016/j.cirp.2015.04.121
29. The AnyLogic Company: AnyLogic: Simulation Modeling Software (2024). https://www.anylogic.com/. Version 8.9.1
30. Tolio T, Sacco M, Terkaj W, Urgo M (2013) Virtual factory: An integrated framework for manufacturing systems design and analysis. pp 25–30. https://doi.org/10.1016/j.procir.2013.05.005
31. Urgo M, Terkaj W (2020) Formal modelling of release control policies as a plug-in for performance evaluation of manufacturing systems. CIRP Ann 69(1):377–380. https://doi.org/10.1016/j.cirp.2020.04.007
32. Urgo M, Terkaj W, Liu L (2025) Towards digital twin-enhanced control policies: A knowledge-based classification of release and dispatching policies in manufacturing systems. CIRP J Manuf Sci Technol 63:310–335. https://doi.org/10.1016/j.cirpj.2025.08.006
33. W3C: SPARQL 1.1 Query Language. W3C Recommendation (2013). https://www.w3.org/TR/sparql11-query/. Accessed: March 15, 2024
34. W3C OWL Working Group: OWL 2 Web Ontology Language. https://www.w3.org/TR/owl2-overview/ (2012). Recommendation, World Wide Web Consortium (W3C), Cambridge, USA