Linköping Studies in Science and Technology
Dissertation No. 2505

Adaptive Automation Strategies for Increasing Variability in Design and Production

Sanjay Nambiar

LINKÖPING
UNIVERSITY

Linköping Studies in Science and Technology
Dissertations, No. 2505

Adaptive Automation Strategies for Increasing Variability in Design and Production

Sanjay Nambiar

Linköping University
Department of Management and Engineering
Division of Product Realization
SE-581 83 Linköping, Sweden

Linköping 2026

Typeset using LaTeX
Language and grammar improved using ChatGPT

Cover designed using Google Gemini

Printed by LiU-Tryck, Linköping 2026

Edition 1:1

ISBN 978–91–8118–461–7 (print)
ISBN 978–91–8118–462–4 (PDF)
https://doi.org/10.3384/9789181184624
ISSN 0345-7524

To my wife, Madhu.

POPULÄRVETENSKAPLIG SAMMANFATTNING

Den ökande efterfrågan på kundanpassade produkter, tillsammans med behovet av flexibla, människocentrerade och motståndskraftiga tillverkningssystem, har intensifierat behovet av automatiseringslösningar som kan fungera i dynamiska och ostrukturerade industriella miljöer. Denna avhandling visar hur automatiseringsmetoder utvecklas och var de misslyckas i takt med att komplexitet och variation ökar inom design- och produktionsområdena.

Forskningen börjar med att hantera tidskrävande och iterativa tekniska uppgifter genom designautomation. Med hjälp av kunskapsbaserad teknik (KBE) utvecklades automatiserade ramverk för att effektivisera tekniska arbetsflöden och stödja konsekvent beslutsfattande i strukturerade industriella miljöer. Men när fokus utvidgas till verklig produktion skapar den växande komplexiteten och osäkerheten i miljön betydande utmaningar för både design- och produktionsautomation.

Medan KBE ger struktur och konsekvens, kräver dess beroende av fördefinierade regler standardisering, vilket resulterar i ett rigid designutrymme och begränsad anpassningsförmåga. För att övervinna dessa inneboende begränsningar integrerar forskningen kompletterande verktyg och tekniker som möjliggör mer flexibel och adaptiv automatisering. Kameraseende fångar verkliga förhållanden och spårar förändringar i miljön, medan stora språkmodeller, i kombination med en agentbaserad metod, ger resonemangsförmåga som tolkar variationer i produkter eller processer och genererar adaptiva beslutsstrategier. Digitala tvillingsimuleringar validerar och förutsäger resultaten av dessa variationer i en virtuell miljö, vilket gör att systemet kan reagera proaktivt och säkert genom att förena realtidsdata med simuleringsresultat.

Sammantaget bidrar detta arbete med en holistisk och skalbar automationsmetodik som förenar designautomation, adaptiva digitala tvillingar och kunskapsdrivet resonemang. Resultaten visar hur strukturerad ingenjörskunskap, i kombination med resonemang och adaptiva teknologier, möjliggör utveckling av motståndskraftiga automationslösningar för det alltmer ostrukturerade landskapet inom framtidens industri.

ABSTRACT

The increasing demand for customized products, together with the need for flexible, human-centric, and resilient manufacturing systems has intensified the need for automation solutions capable of operating in dynamic and unstructured industrial environments. This dissertation shows how automation methodologies evolve and where they fail as complexity and variability increase across the design and production domains.

The research begins by addressing time-consuming and iterative engineering tasks through design automation. Using Knowledge-Based Engineering (KBE) approaches, automated frameworks were developed to streamline engineering workflows and support consistent decision-making in structured industrial settings. However, when extending the focus to real-world production, the growing complexity and uncertainty of the environment create substantial challenges for both design and production automation.

While KBE provides structure and consistency, its reliance on predefined rules necessitates standardization, resulting in a rigid design space and limited adaptability. To overcome these inherent restrictions, the research integrates complementary tools and techniques that enable more flexible and adaptive automation. Camera vision captures real-world conditions and tracks changes in the environment, while large language models, combined with an agent-based approach, provide reasoning capabilities that interpret variations in products or processes and generate adaptive decision-making strategies. Digital twin simulations validate and predict the outcomes of these variations in a virtual environment, allowing the system to respond proactively and safely by reconciling real-time data with simulation outcomes.

Overall, this work contributes a holistic and scalable automation methodology that unifies design automation, adaptive digital twins, and knowledge-driven reasoning. The results demonstrate how structured engineering knowledge, combined with reasoning and adaptive technologies, enables the development of resilient automation solutions for the increasingly unstructured landscape of future Industry.

Acknowledgments

The research presented in this dissertation was carried out at the Division of Product Realisation, Linköping University. Completing this journey would not have been possible without the contribution, guidance, and encouragement of many individuals.

I would like to extend my deepest gratitude to my supervisors for their unwavering support throughout this long road. To Marie Jonsson, Anton Wiberg, and Johan Persson, thank you for your insightful feedback and for always being available when I needed guidance. A very special mention goes to my main supervisor, Mehdi Tarkian. Your belief in my potential has been a constant source of motivation (along with the occasional 'mehdi talk'). Your mentorship has truly shaped my growth as a researcher.

To my colleagues and PhD peers at the Division of Product Realisation, thank you for the coffee breaks, the brainstorming, and the many chats that made the challenges of research into an enjoyable shared experience. I also wish to acknowledge the assistance of ChatGPT, which served as a valuable tool in providing technical and language assistance through most part of my research process.

Above all, my beloved Madhu, you have been a true foundation of this journey. This thesis is as much a testament to your patience and sacrifice as it is to my research. I would like to express my gratitude to Achan, Amma, Anju, Avinash, Nandu and Neeru for your support. And finally to my friends for supporting me through this journey.

Appended Publications

Papers I-VI presented below comprise the foundation of this thesis.

Paper I

Nambiar, S., Ananno, A. A., Titus, H., Wiberg, A., & Tarkian, M. (2024). Multidisciplinary Automation in Design of Turbine Vane Cooling Channels. International Journal of Turbomachinery, Propulsion and Power, 9(1), 7.

Contribution: Nambiar initiated and conducted literature research, contributing to framework development. Annano and Titus developed the remaining framework. Nambiar is the main author of the paper. Wiberg and Tarkian provided support with ideas, structure, and enhancing paper quality.

Paper II

Nambiar, S., Albert, A. P., Rimmalapudi, V. V. R. C., Acharya, V., Tarkian, M., & Kihlman, H. (2022). Autofix–automated design of fixtures. Proceedings of the Design Society, 2, 543-552.

Contribution: Nambiar initiated the paper, conducted literature research, and is the primary author. Nambiar and Albert equally contributed to framework development. Rimmalapudi and Acharya supported in setting up initial part of the methodology. Tarkian provided support with ideas, structure, and enhancing paper quality.

Paper III

Nambiar, S., Wiberg, A., & Tarkian, M. (2023). Automation of unstructured production environment by applying reinforcement learning. Frontiers in Manufacturing Technology, 3.

Contribution: Nambiar initiated the paper and is the main author, responsible for the development of framework. Wiberg and Tarkian supported with ideas, improving the quality of the paper.

Paper IV

Nambiar, S., Jonsson, M., & Tarkian, M. (2024). Automation in unstructured production environments using Isaac sim: A flexible framework for dynamic robot adaptability. Procedia CIRP, 130, 837-846.

Contribution: Nambiar initiated the paper and is the main author of the paper. Majority of the framework is developed by Nambiar, Jonsson supported in the development of automation involving hardware. Jonsson and Tarkian supported with ideas, structure and improving the quality of the paper.

Paper V

Nambiar, S., Paul, R. C., Ikechukwu, O. C., Jonsson, M., & Tarkian, M. (2025). Digital Twin-Enabled Adaptive Robotics: Leveraging Large Language Models in Isaac Sim for Unstructured Environments. Machines, 13(7), 620.

Contribution: Nambiar initiated the paper and is the main author of the paper. Majority of the framework is developed by Nambiar, Ikechukwu and Paul supported in the development of camera vison and large language model. Jonsson and Tarkian supported with ideas, structure and improving the quality of the paper.

Paper VI

Nambiar, S.,Ikechukwu, O. C. , Paul, R. C., Jonsson, M., & Tarkian, M. (2026). Digital Twin-Enabled Adaptive Robotics: Multi-Agent Reasoning over Language, Vision, and Structured Database. Production & Manufacturing Research, (Submitted).

Contribution: Nambiar initiated the paper and is the main author of the paper. Majority of the framework is developed by Nambiar, Ikechukwu and Paul supported in the development of simulation and large language model. Jonsson and Tarkian supported with ideas, structure and improving the quality of the paper.

Abbreviations

AI	Artificial Intelligence
AR	Action Research
BIW	Body-in-White
CAD	Computer Aided Design
DA	Design Automation
DRM	Design Research Methodology
DT	Digital Twin
EGM	Externally Guided Motion
HLCt	High-Level CAD templates
HRI	Human-Robot Interaction
IK	Inverse Kinematics
KBE	Knowledge-Based Engineering
KBS	Knowledge-Based System
LLM	Large Language Model
MAS	Multi-Agent Systems
MDO	Multidisciplinary Optimization
OOP	Object-Oriented Programming
RL	Reinforcement Learning
RMP	Riemannian Motion Policy
VBA	Visual Basics for Applications

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Motivation

Manufacturing is undergoing a transformation driven by shifting market expectations and societal priorities. Over the past decade, demand for personalized and customized products has grown steadily across sectors [1, 2, 3]. Customers increasingly expect products that reflect their individual needs, preferences, and usage contexts [4]. This trend challenges the long-established model of stable, high-volume design and production lines optimized for long product cycles and minimal variation. Instead, manufacturers are facing an environment in which product mixes change rapidly, batch sizes shrink, and new product variants must be introduced with minimal disruption [5]. Increasing product variability places significant demands on design processes also, where engineers must repeatedly adapt geometries, constraints, and process assumptions to accommodate new variants. As a result, manufacturing systems must evolve from rigid, pre-defined configurations to operations capable of responding to variability as a built-in characteristic rather than an exception.

Importantly, the challenges introduced by variability are not confined to the production floor. Design and production automation are tightly coupled stages within a continuous engineering lifecycle, where decisions made during design directly constrain or enable flexibility with efficiency in production. Automation solutions that address only downstream execution without considering upstream design adaptability risk shifting complexity rather than resolving it. Consequently, achieving robustness in varying manufacturing environments requires coordinated advances in both Design Automation (DA) and production automation, rather than isolated optimization of either stage.

In conventional manufacturing, the economic justification for industrial robots has historically been tied to production scale [6]. As illustrated in Figure 1.1, robot deployment becomes cost-effective primarily in medium- and high-volume settings, where the costs of automation can be distributed across large numbers of units [7]. Under these conditions, the unit cost decreases as volume increases, making automation an attractive and financially sound choice. However, when production volumes are low, the unit cost associated with industrial robots remains comparatively high. This cost imbalance often discourages companies from adopting automation for small-batch or customized production [8, 9, 10]. Instead, firms tend to rely on manual labour or shift production to regions with lower labour costs, reinforcing a dependence on practices that may be economically or socially unsustainable in the long term.

A key motivation for this thesis is to challenge this traditional cost–volume relationship. If industrial robots can be made sufficiently adaptive, capable of handling diverse tasks, responding to variability, and operating in less structured conditions, the economic viability of automation could extend into low-volume and customization intensive manufacturing. This shift would not only close the gap between low- and medium-volume scenarios but also support more resilient and regionally grounded manufacturing models.

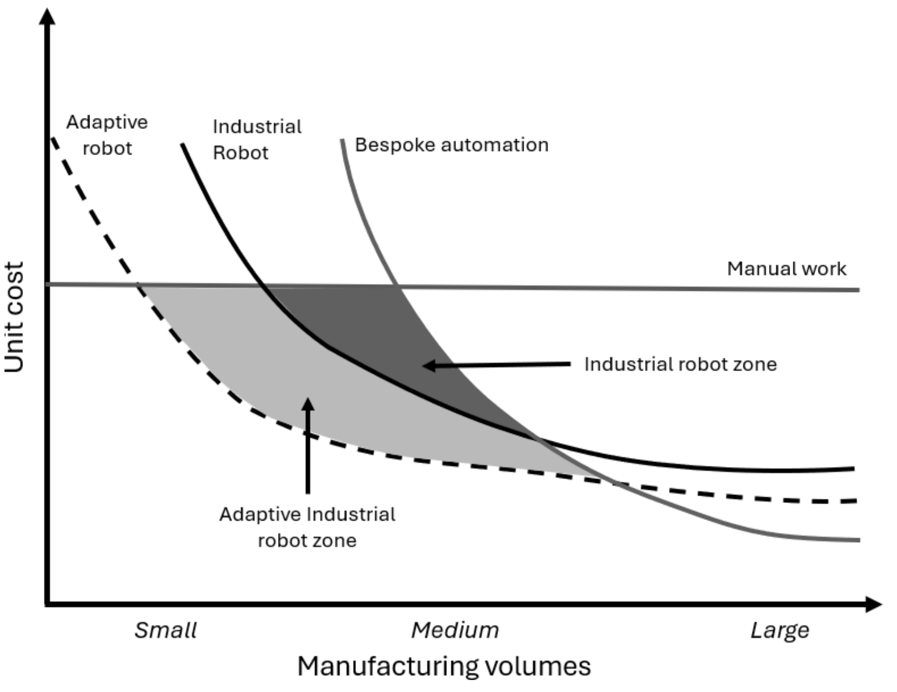


Figure 1.1: Use of industrial robot with respect to unit cost and manufacturing volumes (adapted from Pires, Loureiro and Bölmsjo[7])

At the same time, the ongoing shift toward Industry 5.0 places additional demands on manufacturing. Beyond productivity and efficiency, the emphasis now extends to sustainability, circular resource flows, and meaningful collaboration between humans and technology [2, 11]. Circularity oriented strategies such as remanufacturing, refurbishment, recycling, and reuse are becoming integral parts of industrial practice [12]. These activities inherently involve higher levels of uncertainty: components may return in different conditions, assemblies may require selective disassembly, and material properties may vary from batch to batch. Such operations rarely conform to standardized workflows, further amplifying the unstructured nature of modern shop floors. To navigate this complexity, recent research in smart and adaptive manufacturing leverages cyber-physical systems, Artificial Intelligence (AI), and Digital Twins (DT) to enhance real-time decision-making and resilience [13].

Together, customized products and circular manufacturing practices create a manufacturing landscape that is progressively less predictable, less uniform, and more dynamic [14]. Conventional automation solutions, designed around fixed sequences, rules and well-controlled environments, struggle to accommodate this reality. Even small deviations such as variations in part annotations, geometry, or condition, can result in automation failures, forcing manual intervention and limiting the scalability of automated solutions. Engineers are therefore confronted with a fundamental question: how can automation be reimagined to operate reliably in environments characterized by variability, incomplete information, and continuous change?

This challenge forms the core motivation of this thesis. Rather than treating variability as a problem to be minimized, there is a growing need for design and production automation solutions that can adapt, learn, and collaborate: systems that can incorporate human expertise, respond to unstructured situations, and flexibly reconfigure their behaviour. Adaptive automation, viewed from this broader perspective, is not tied to a single technology; it is a design philosophy aimed at enabling manufacturing systems that remain robust and effective despite the complexities introduced by customization, circularity, and human involvement.

1.2 Terminology

The terminology defined in this section has been deliberately selected because these concepts recur throughout the thesis and form the conceptual foundation of the research. As several of these terms are used differently across research communities and industrial contexts, explicitly defining them is necessary to avoid ambiguity and to ensure a consistent interpretation of key concepts.

- **Design Automation:** In this thesis, the term Design Automation (DA) is used following the definition by Cederfeldt and Elgh [15], which describes it as the engineering support provided through the use of information and knowledge embedded in tools, systems, or solutions that are designed for reuse. This definition covers the computerized automation of tasks that are directly or indirectly related to the design process, ranging from individual components to complete products, thereby facilitating the overall progress of the design workflow.
- **Production Automation:** According to Satchel [16], “automation is the replacement of human activity by machine activities”. In this thesis, production automation refers to the use of mechanical, electronic, and computer-based systems such as control systems, robotics, sensors, and software to carry out

inspection, control, and execution of manufacturing operations [17]. These technologies enable production processes to be performed with minimal human intervention while enhancing efficiency, consistency, and flexibility.

- **Adaptivity:** In this thesis, adaptivity in the context of production systems and components is defined based on Hribernik et al. [18] as the capability of a system or component to modify its behaviour in response to unpredicted goals in order to achieve one or more objectives. Adaptivity is considered a core objective of Industry 4.0 for smart factories, where it refers to "manufacturing solutions that provide flexible and adaptive production processes capable of addressing problems arising in production facilities with dynamic and rapidly changing boundary conditions in an increasingly complex environment" [19].
- **Unstructured Environment:** In this thesis, an unstructured environment refers to a design and production context characterized by high variability, incomplete information, and evolving constraints, making the environment dynamic rather than static. From a design perspective, unstructured environments arise when product definitions, functional requirements, geometries, or process assumptions cannot be fully specified in advance and are subject to frequent change, often due to customization or reconfiguration. From a production perspective, such environments manifest as variability in annotation, geometry, material condition, task sequencing, or workspace configuration. Unlike traditional structured settings, in which automation is governed by fixed rules, predefined models, and stable workflows, unstructured environments require automation solutions that can accommodate increasing variability through flexible adaptation, continuous reconfiguration, and effective handling of unforeseen conditions.

1.3 Aim

The aim of this thesis is to advance the understanding of how automation strategies can be designed and adapted to address the growing variability of products and manufacturing environments. Conventional automation approaches, as illustrated in Figure 1.2, typically follow a Knowledge-Based Engineering (KBE) workflow comprising the stages of identification, justification, knowledge capture, formalisation, packaging, and activation [20]. In a manufacturing context, this process begins with identifying requirements or boundary conditions, followed by validating observations and transforming them into informal models. These models are then formalised into executable representations, modularised for reuse, and finally deployed as automated solutions.

A critical limitation of this traditional methodology is that the early stages, particularly identification, justification, and knowledge capture are predominantly manual processes. As a result, automation systems built through this workflow struggle to respond effectively when product configurations or environmental conditions vary, making them unsuitable for dynamic or unstructured environments. This limitation is closely related to the challenge of when and how an automation system should adapt its behaviour. In adaptive automation, changes may be triggered by variations in design, environment, task, system state, or interaction with human operators [21]. How these triggers are detected and acted upon fundamentally influences whether adaptation is initiated autonomously by the system or deliberately by a human operator [21].

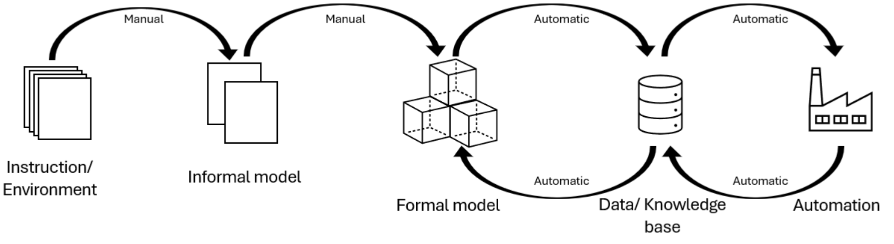


Figure 1.2: Iterative life-cycle model of 'Methodology and tools Oriented to Knowledge-based engineering Application'.

This distinction highlights an important design consideration: adaptive systems that modify their behaviour automatically versus adaptable systems in which humans retain direct control over when and how adaptation occurs [22]. While system-driven adaptation has received substantial attention in automation research [23, 24, 25], there is growing recognition that preserving human involvement in the adaptation process can enhance situational awareness, trust, and overall system performance, particularly in complex manufacturing environments [26, 27, 28, 29]. From this perspective, the challenge is not only to automate downstream execution, but to DA architectures that can sense relevant triggers, reason about their implications, and support appropriate levels of human oversight across both design and production stages. To guide the systematic exploration of this problem, the research is structured around the following question:

RQ *What are the limitations associated with implementing adaptive automation techniques in unstructured environments, and how can these challenges be mitigated?*

The research question is intentionally framed at a broad level to reflect the dual focus of this thesis on both DA and production automation. Rather than addressing a single phase of the product lifecycle, the question encompasses the full span from early-stage design activities to downstream production. This scope is necessary because variability and complexity in modern manufacturing do not originate in a single domain, instead, they emerge from the strong interdependencies between product design choices, production system configurations, and operational conditions on the shop floor.

The research question therefore serves as a guiding framework rather than a problem to be exhaustively solved within the scope of this thesis. It establishes the premise that achieving adaptive automation in unstructured environments requires first identifying the fundamental limitations present at both the design and production stages, and then exploring how these limitations may be mitigated through emerging technologies and architectural approaches. Rather than aiming to enumerate all potential limitations or to prescribe a single, universal solution, the question frames a structured inquiry into the incremental realization of adaptability. By systematically recognizing and addressing key limitations, this approach lays the groundwork for developing more robust and adaptive automation architectures capable of operating effectively in unstructured environments.

1.4 Thesis Scope

The research presented in this thesis investigates a series of solutions aimed at addressing variability in products and manufacturing environments, both in design and in production. The work has evolved through several interconnected stages, each building on the limitations and insights of the previous phase.

The initial phase focused on design-related challenges, particularly the implementation of DA and automated structural simulation for turbine vane cooling. KBE served as the foundational methodology, enabling the formalisation and reuse of engineering knowledge to automate design tasks [30, 31]. This methodology was subsequently applied to a second industrial case involving fixture design for Body-in-White (BIW) in automotive manufacturing. In this context, both DA and design optimization were conducted using KBE as the underlying framework, further demonstrating the method's potential for reducing manual workload and increasing consistency in static and well-structured design domains.

While the early stages of the research addressed variability from a design perspective, the subsequent phases shifted toward production environments, with a particular focus on industrial robotics. Observations from these settings highlighted significant limitations of conventional KBE automation when deployed in unstructured or highly dynamic environments. These constraints motivated an exploration of physics-based simulation environments and Reinforcement Learning (RL) as potential means of enabling robots to make decisions based not solely on predefined rules but on learned behaviour and accumulated experience.

Although RL did not yield the desired level of modularity or reliability, this phase was instrumental in establishing the value of physics-based simulation platforms capable of modelling complex interactions and variability. The insights gained directed the research toward the development of adaptive automation strategies that could be deployed on physical robotic systems.

To bridge the gap between simulated adaptability and real-world implementation, the research advanced toward utilising DT technology. A real-time DT was developed to enable responsive control, continuous monitoring, and informed decision-making in environments characterised by high variability. Building on this foundation, the resulting framework was further enhanced using state-of-the-art technologies, including Large Language Models (LLM), vision-based perception, and multi-agent architectures. During the implementation of these advanced solutions, it was observed that the absence of a structured knowledge base could reduce system reliability and operational efficiency. To address this, the proven methodology of KBE was integrated with agent-based decision-making and adaptive simulation. This hybrid approach facilitated the development of more robust solutions, combining the strengths of structured knowledge representation with dynamic, experience-driven decision-making.

Overall, the research trajectory reflects a progression from automation solutions suited to static, rule-based environments toward the development of adaptive automation capable of handling variability and unstructured conditions through the integration of knowledge-based methodologies, simulation, and AI-driven technologies.

1.5 Research Methodology

Research, in its broadest sense, is a systematic process aimed at generating knowledge, understanding phenomena, and developing solutions to identified problems [32]. Scientific research advances this goal by producing new fundamental insights, whereas applied research combines knowledge generation with the resolution of practical challenges [33]. Across disciplines, scientific research is commonly categorised into quantitative, qualitative, and mixed-method approaches [34]. Qualitative methods explore complex, context-dependent phenomena that cannot be reduced easily to numerical description, such as human–machine interaction or organisational constraints within manufacturing environments. Quantitative research, in contrast, relies on numerical data, objective measurement, and statistical analysis to evaluate system behaviour, an approach closely aligned with engineering research, where controlled experiments, performance assessments, and computational modelling provide empirical validation. Mixed-method approaches integrate these perspectives, enabling a comprehensive understanding that accounts for both measurable system characteristics and contextual factors. Given the technical nature of the present work and its emphasis on model development, system validation, and empirical testing, this thesis is situated primarily within the quantitative research paradigm.

Because the research is closely aligned with engineering design and manufacturing, the methodology must also be grounded in the rigorous traditions of disciplines that study design as a human and technical activity [33]. The development and evaluation of artefacts such as frameworks, algorithms, digital tools, and adaptive automation solutions necessitate specialised methodological support beyond general quantitative practices. Design Science Research offers one established approach by providing a structured process for creating artefacts that address real-world problems while contributing to theoretical knowledge [35]. However, scholars have identified the need for an even more rigorous and comprehensive methodology for engineering design research, one that supports empirical investigation and the systematic development, validation, and introduction of design methods and tools [36]. This raises the broader question of whether design research requires its own methodology or a framework capable of integrating multiple disciplinary approaches [33]. The work in this thesis aligns with the latter perspective, adopting the Design Research Methodology (DRM) developed by Blessing and Chakrabarti, which provides a structured, iterative, and empirically grounded framework well suited to research that spans both design and production automation [37].

1.5.1 Design Research Methodology

Design Research Methodology (DRM), developed by Blessing and Chakrabarti, is a structured and systematic approach tailored specifically for conducting rigorous research in engineering design [37]. It provides a coherent framework for studying design activities, developing new methods or tools, and evaluating their impact in a scientific manner. DRM is organised into four major stages as shown in Figure 1.3 and these stages are:

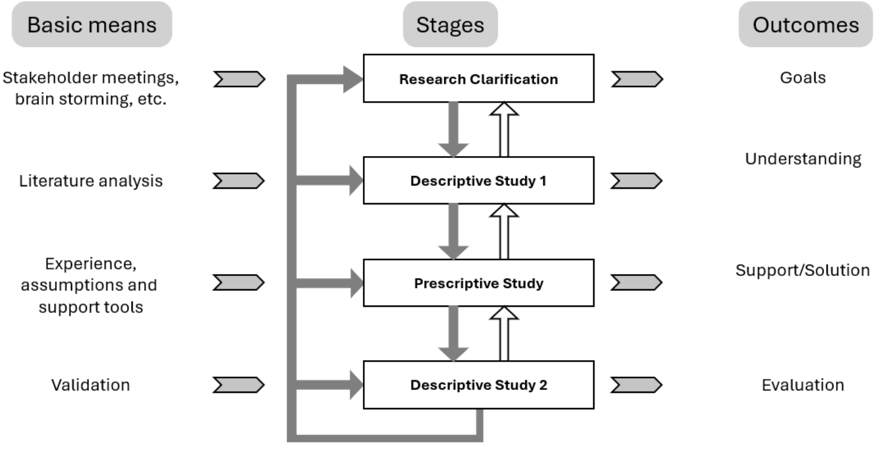


Figure 1.3: The Design Research Methodology, adapted from Blessing and Chakrabarti [37].

1. **Research Clarification (RC):** The RC stage establishes the foundation for the entire study. Its goal is to define the research problem clearly, justify its relevance, and identify the desired improvements in design practice or system performance. During this phase, the existing literatures are reviewed, industrial practices are examined, and the scope and objectives of the research is determined. The outcome of RC is a well-formulated research aim supported by preliminary theoretical understanding and evidence of the problem's significance.
2. **Descriptive Study - I (DS1):** DS1 focuses on analysing the current situation to understand the existing process, challenges, and influencing factors. This stage involves gathering empirical data through observations and literature reviews of state-of-the-art. DS1 seeks to develop an explanatory model that clarifies why the identified problems occur, what variables influence system performance, and how current methods or tools support or fail to support different activities.
3. **Prescriptive Study (PS):** Here the study stage aims to develop solutions intended to improve the issues identified in DS1. These solutions may take the form of frameworks, methodologies, design tools, automation approaches, or decision-support systems. PS involves both conceptual development and initial testing, ensuring that the proposed support is theoretically justified and practically feasible. The output of this stage is a support or solution which typically act as prototypes or demonstrators.
4. **Descriptive Study - II (DS2):** DS2 evaluates the effectiveness of the prescriptive support developed in PS. This evaluation is conducted through empirical studies such as experiments, case studies, or validation in industrial contexts. DS2 assesses whether the proposed solution leads to the intended improvements and identifies any limitations, unexpected outcomes, or necessary refinements. This stage provides the evidence needed to validate the contribution of the research and determine its practical impact.

1.5.2 Research Approach

While the four stages of research presented by Blessing and Chakrabarti may appear sequential, it is important to recognize that each stage is inherently iterative, allowing multiple cycles of refinement to achieve the desired objectives. Furthermore, within each stage, different research approaches can be applied depending on the nature of the problem and the type of knowledge sought. One such approach employed is Action Research (AR)[38]. AR is a participatory and iterative research approach that seeks to generate knowledge while simultaneously promoting practical change.

The methodology of AR is commonly conceptualized as a spiral of iterative cycles, with each cycle typically involving the following elements: planning, implementation (action), and evaluation (reflection) [38] as shown in Figure 1.4. In the planning phase, the problem is defined, objectives are set, and strategies or interventions are designed. The implementation phase involves executing the planned interventions in a practical setting. The evaluation and reflection phase focuses on interpreting the observed results from implementation, assessing the effectiveness of the plan, identifying limitations, and generating insights for the next cycle. This cyclical process is further structured using the LOOK-THINK-ACT framework [38].

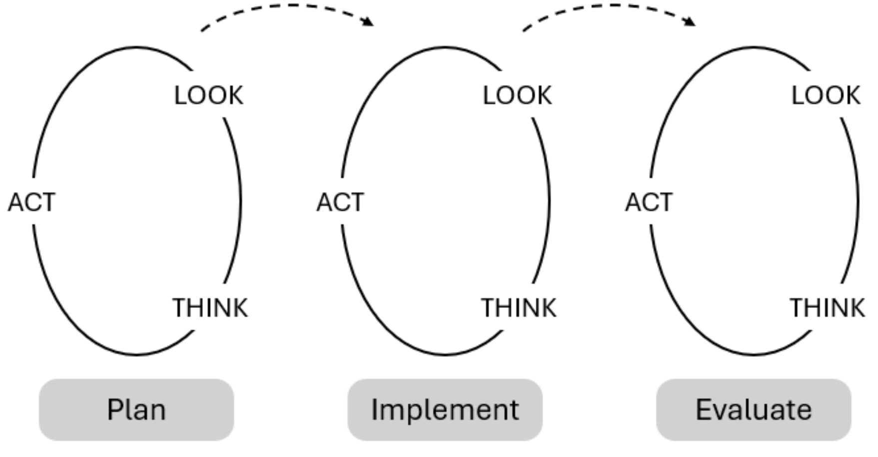


Figure 1.4: AR interacting spiral (adapted from Stringer and Aragón [38]).

During the LOOK phase, data is systematically observed and collected to evaluate the effects of the interventions, including monitoring outcomes, interactions, and contextual factors through observations, ensuring that findings are grounded in empirical evidence rather than assumptions. The THINK phase involves reflecting on and analysing the observed data to identify successes, limitations, and potential improvements. It also includes interpreting the findings in the context of existing theory, previous cycles, and research objectives, and using these insights to inform the planning of the next cycle, emphasizing learning, knowledge generation, and iterative refinement. In the ACT phase, the planned interventions or actions are carried out. Strategies are implemented to address the identified problem while ensuring the process is structured, documented, and repeatable, emphasizing practical engagement and active initiation of change.

1.5.3 Applied Methodology

The research presented in this thesis follows the DRM outlined in sub-section 1.5.1, integrated with an AR approach. The work adopts an application-driven perspective, where successive application developments are used to investigate, refine, and ground the overarching research question. For papers included in this thesis, the initial stage of DRM was carried out through stakeholder meetings and discussion sessions with individuals closely connected to the application domain or research topic. This was followed by a state-of-the-art review or background analysis to establish the problem context and clarify the research direction. While these early steps were largely consistent across the studies, the subsequent PS varied depending on the support tools available, the assumptions made, and the practical experience guiding each project. Multiple iterations between the PS and the DS1 were conducted to identify, refine, and evaluate the tools and technologies capable of addressing the research objectives. Similarly, the DS2 was primarily focused on validating the developed tools, methods, or architectures, with the level of validation determined by both the feasibility of testing and the scope of each individual project. Figure 1.5 illustrates how each paper presented in this thesis aligns with the different stages of DRM, and their scope and contributions are explained in detail below.

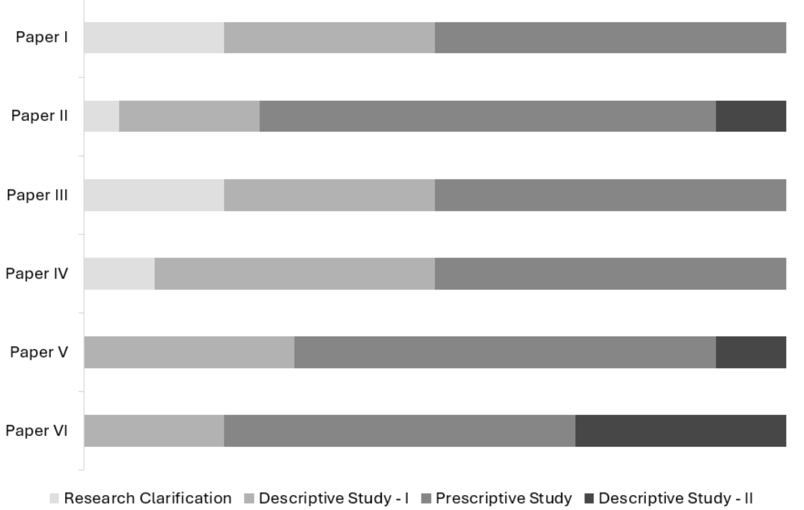


Figure 1.5: Summarized connection of published papers against different stages of DRM.

1. **Paper I:** This study spans the early DS1 and the PS stages of the DRM. It begins with a detailed literature review to identify gaps in automated cooling channel design, forming the basis of DS1, and then advances into a quantitative investigation focused on developing a multidisciplinary automation solution. The paper's main contribution is a novel framework that integrates DA, meshing, and structural analysis for turbine blade cooling channels, streamlining iterative and labour-intensive tasks and improving overall workflow efficiency in gas turbine development.
2. **Paper II:** This study extends the KBE principles used in Paper I but places stronger emphasis on a comprehensive PS, supported by targeted validation

activities characteristic of an initial DS2. The core contribution is a flexible and partially automated framework for fixture design and optimization in BIW structures, integrating DA, parametric modelling, optimization, and robotic simulation.

3. **Paper III:** Building on insights from the DS1 phase of the previous studies, this paper addresses identified research gaps through a review based RC, followed by a comprehensive DS1 and an initial PS. The main contribution is the development of two RL applications within a lightweight game engine: a mobile robot application for planning and controlling robots in unstructured industrial environments, and a mannequin application for automated simulation of unstructured assembly processes. The work demonstrates the promise of RL in adaptive automation and provides a foundation for future validation and integration into more complex industrial environments.
4. **Paper IV:** Opening a new direction for adaptive automation in manufacturing, this paper begins with a literature-based RC followed by a comprehensive DS1 to analyse the state-of-the-art in DT and simulation technologies. Building on insights from the PS of Paper III, the study develops a framework for establishing a digital shadow by integrating a physics-based simulation engine with robotic systems. The framework enables dynamic, real-time decision-making, providing a robust foundation for exploring DT technologies in unstructured industrial settings.
5. **Paper V:** Building on the limitations and observations from Paper IV, this study begins with a comprehensive DS1 supported by an in-depth literature analysis on DTs and human-centric production in collaborative robotics. The subsequent PS develops an enhanced adaptive DT architecture, incorporating a real-time perception module and integration with a local LLM to enable intuitive verbal task instructions. An initial DS2 validates the architecture through experimental runs, assessing synchronization, efficiency, and practical limitations. The paper's main contribution lies in advancing a robust, human-centric adaptive DT framework for collaborative robotic application, enabling dynamic adaptation to unstructured environments.
6. **Paper VI:** Extending the work from Paper V, this study begins with a comprehensive DS1 that includes a detailed literature analysis on state-of-the-art technologies such as multi-agent based automation and structured knowledge management. The subsequent PS implements these technologies within the previously developed DT architecture to address limitations identified during the DS2 of Paper V. A thorough DS2 validates the enhanced architecture, assessing the integration of multi-agent coordination, structured databases, and camera-based perception within adaptive robotic automation in unstructured environments.

1.6 Thesis Outline

The remainder of this thesis is structured as follows. Chapter 2, Frame of Reference, provides the theoretical and technological background necessary to understand the research. It covers key topics such as DA, multi-agent systems, DT technologies, and human–robot collaboration, setting the stage for the application-focused studies. Chapter 3, Application, presents the main contributions of this thesis through a series of individual publications. Each case explores specific implementations, methodologies, and findings, showing how the developed solutions and tools address challenges in adaptive automation and unstructured environments. Chapter 4, Discussion, evaluates the contributions of the research, highlights its limitations, and identifies directions for future work based on observed gaps and emerging opportunities. Finally, Chapter 5, Conclusion, synthesizes the research by answering the research question posed in Chapter 1 and reflecting on the broader implications of the study, demonstrating how the developed methods and frameworks advance adaptive automation solutions.

CHAPTER 2

Frame of Reference

This chapter presents the frame of reference for the thesis by outlining the core research areas that underpin the development of adaptive automation in unstructured manufacturing environments. Design automation and knowledge-based engineering provide methods for capturing and reusing engineering knowledge to support product customization. Multi-agent systems enable modular, distributed decision-making and coordination in complex automation architectures. Digital twin technologies integrate models, simulation, perception, and physical systems, enabling continuous alignment between the digital and physical domains. Together, these areas establish the theoretical and technological foundation for the approaches developed in this thesis and clarifies how these research areas relate to the papers included in this thesis

2.1 Design Automation

The roots of modern Design Automation (DA) trace back to the early days of Computer-Aided Design (CAD). In 1957 Patrick J. Hanratty (Father of CAD/CAM) developed PRONTO (Programme for Numerical Tooling Operations), one of the first commercial numerical-control software packages [39]. Hanratty later founded manufacturing and consulting services and released ADAM (Automated Drafting and Machining) in the early 1970s. Around the same time (1963), Ivan Sutherland created Sketchpad, the first interactive graphics system with a light-pen interface[40]. These innovations proved that complex geometric shapes and design data could be created and manipulated on computers. Over the following decades CAD became indispensable across industries, laying the groundwork for DA, the systematic use of software to automate design tasks.

DA is broadly defined as the computerized automation of tasks related to the design process through the implementation of design knowledge in software tools [15]. In practice, DA research and applications focus on two complementary areas: automating the design artifact (the product geometry and features) and automating the design process (the sequence of steps and decisions)[41]. For example, developing a customized product requires a detailed breakdown of every processing step in the design workflow. A DA framework captures these process steps and encapsulates the design knowledge so that routine activities can be replayed automatically to generate new product variants. In effect, DA "reuses" pre-planned knowledge (rules, templates, constraints) to accelerate variant design. The potential for DA grows as a product's design rules become well defined and as customization requirements increase [42]. In other words, highly standardized products with many variants are prime candidates for automation. At the same time, DA systems can be simple or elaborate: they range from libraries of standard parts or family templates, all the way to fully knowledge-intensive CAD platforms.

Key motivations drive companies to adopt DA. An efficient DA framework can significantly reduce development time and effort while improving consistency and flexibility. Industry studies consistently report effects such as shortening lead times for design and delivery, reducing labour-intensive repetitive tasks, reusing existing design knowledge, and ensuring consistent quality across all variants [15]. Automating routine tasks frees engineers to concentrate on novel, creative or critical aspects of the design, leveraging human intuition and skill where it adds the most value. Importantly, DA also improves quality assurance by automating rule-following tasks, every output follows the same standardized procedure, so designs from different engineers or different runs are identical in terms of correctness and standards compliance. In practice, a DA system can take inputs such as customer requirements, constraints and parameters, and then automatically generate output artifacts. With the ability to adapt easily to new requirements, a mature DA framework enhances the efficiency and maintainability of the product development process[43].

Many modern DA approaches are built on Knowledge-Based Engineering (KBE). KBE involves formally capturing engineering expertise (rules, relations, algorithms) so that software can make design decisions. As Craig and Pinfold describe, "KBE represents an evolutionary advancement" that "merges Object-Oriented Programming (OOP), Artificial Intelligence (AI), and CAD technologies to facilitate customized or variant DA solutions" [44]. A robust DA framework therefore depends on a rich, structured knowledge base, especially one capable of handling geometry. One approach of such implementation is through High-Level CAD templates (HLCt) which encapsulates a geometric shape along with all its design parameters[45]. This can ensure both topological changes as well as morphological changes depending on parameters as well as the geometry features.

Moreover, implementing a DA framework establishes a solid foundation for subsequent design optimization, particularly when refining geometries based on topological or morphological criteria. In this thesis, Paper I and Paper II apply DA to two distinct engineering applications. Building on the DA approach used in Paper I, Paper II extends the approach by integrating morphological optimization of the CAD model, demonstrating how automated design generation can seamlessly support and enhance geometry-focused optimization processes.

2.2 Knowledge-Based Engineering

Knowledge-Based Engineering (KBE) has evolved into a foundational methodology for automating and enhancing engineering activities across domains such as design and production. Traditionally rooted in design-centric applications, KBE integrates AI, OOP, and CAD technologies to formalize engineering knowledge and reuse it efficiently [44, 31]. Craig and Pinfold describe KBE as an engineering method that merges these technologies to enable customized or variant DA solutions [44]. Similarly, La Rocca characterizes KBE as a means of capturing product and process engineering knowledge and systematically reusing it to reduce lead time and cost while improving consistency [46]. Although KBE has traditionally been associated with design engineering, some adopt a broader perspective, viewing KBE as encompassing all activities related to engineering knowledge capture, management, and reuse. This central idea, codifying expertise so that it can be reused iteratively remains the driving force behind KBE's adoption in modern engineering practice.

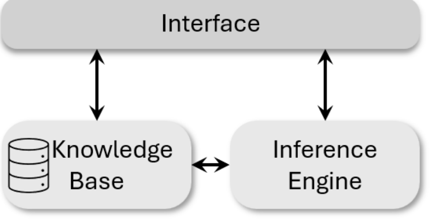


Figure 2.1: Basic structure of KBS.

At the core of KBE lies the Knowledge-Based System (KBS) (Figure 2.1), whose principal purpose is the automation of repetitive or rule-driven design tasks. A defining feature of a KBS is the division between the knowledge base, where engineering rules, relations, constraints, and data are stored, and the interface engine, which activates and executes this knowledge [47]. This separation is not merely architectural; it provides long-term flexibility, allowing the knowledge base to evolve independently of the computational mechanisms that use it. Such modularity supports faster updates, higher maintainability, and easier extension of automation frameworks [48].

In a production setting, early KBE functioned as rule-based expert systems designed to automate the lifecycle from product engineering design to downstream activities like production planning, analysis and cost estimation, particularly in stable, high-volume manufacturing [49]. Diagnostic expert systems and early rule-driven robotics such as Kak et al.'s knowledge-based assembly cell demonstrated the feasibility of automating structured tasks through inference engines [50, 51]. These implementations delivered significant benefits in consistency and repeatability but lacked the flexibility required to handle dynamic or changing production scenarios.

As production environments have shifted toward mass customization, unstructured tasks, and human–robot collaboration, purely rule-based KBE approaches have proven too rigid. In response, contemporary research has expanded such rule-based architectures into more adaptive and semantically rich approaches, such as ontology-driven task planning, cognitive robotics and multi-agent systems. One such shift in production system is shown by Plug and Produce paradigm, which effectively decentralize the "knowledge base" from a single system into the physical assets/agents themselves [52, 53]. In a Plug and Produce environment, the engineering knowledge such as kinematic constraints, operational limits, and process logic is no longer hard-coded into a master controller but is encapsulated within the manufacturing resources (e.g., robots, grippers, or AGVs). This encapsulation allows production modules to "know" their own functionalities and declare them to the system, thereby replacing manual reprogramming with automated self-configuration when the production line changes [54]. To enable this flexible orchestration, the industry has increasingly adopted skill-based control architecture which elevates the abstraction level of control code by formalizing low-level machine operations into standardized "skills" (grasp, weld, transport) that are semantically described and vendor-neutral [55, 56].

Recent literature highlights that skill-based control functions as a dynamic inference engine, instead of executing static IF-THEN rules, the system performs "capability matchmaking." It compares the requirements of a product against the capabilities of available resources, automatically identifying valid production paths without explicit human intervention [57, 58]. Crucially, the interoperability required for such intelligent automation is now being standardized through the Asset Administration Shell (AAS). Serving as information framework for the digital twin, the AAS acts as a standardized container that wraps physical assets with their associated technical data, behavioural models, and interaction rules [59, 60]. While Plug and Produce and skill-based architectures significantly reduce manual engineering effort and improve modularity, they largely assume that the relevant knowledge about system capabilities, constraints, and operational logic can be predefined, formalized, and encapsulated at design time. In practice, however, some production scenarios exhibit variability that cannot be fully anticipated, such as unforeseen object configurations, partial occlusions, human interventions, or evolving task goals. Under such conditions, capability matchmaking and skill orchestration remain constrained by the completeness and correctness of the underlying models and semantic descriptions.

Although decentralizing knowledge into assets enhances reconfigurability, decision-making is still predominantly rule-driven and reactive. These systems can select among known skills but lack mechanisms to reason about novel situations, handle ambiguous perception data, or adapt when available knowledge is incomplete or inconsistent. This limitation becomes particularly evident in environments requiring continuous interpretation of context and coordination among multiple subsystems. These limitations motivate a shift from static knowledge representation toward distributed and deliberative decision-making mechanisms. Multi-agent systems provide a natural extension by decomposing complex automation problems into interacting agents responsible for perception, reasoning, planning, and execution. Such architectures support localized adaptation and coordination, enabling more robust responses to dynamic and uncertain conditions.

2.3 Multi-Agent System

Multi-Agent Systems (MAS) are collections of autonomous entities, software processes or robots that perceive, reason, and act in a shared environment [61]. Each agent operates independently (controlling its own internal state and actions) and makes decisions to achieve its goals. Crucially, no single agent has a global view or control of the entire system. Instead, the overall behaviour emerges from local interactions, communication, and coordination among agents [62]. In this thesis, the agent takes up the adapted definition from Wooldridge and Jennings [63] as a function/function within a system, that operates autonomously within its designated scope to achieve its assigned objectives and produce a desired result. Key MAS properties include: autonomy, decentralization, communication and cooperation, modularity and robustness.

These characteristics contrast with monolithic control architectures, where a single controller would manage all functions. Instead, MAS decompose complex tasks into interacting sub-problems [64]. Classic MAS models include reactive agents (take observation from environment and react/take action) and planning agents (advanced decision making and reasoning) [65]. The Belief–Desire–Intention (BDI) model, in particular, formalizes an agent's mental state: its beliefs about the world, desires (goals to achieve), and intentions (current commitments), which drive decision-making [66, 67]. BDI agents provide a framework for rational planning under resource constraints, and have been influential in agent-oriented software.

More recently, MAS research has converged with advances in AI and learning-based methods. Large Language Models (LLMs) have emerged as a new class of reasoning components that can be embedded within agent architectures. Owing to their capacity for natural language understanding, contextual reasoning, and plan generation, LLMs can satisfy many of the criteria traditionally associated with intelligent agents [68, 69]. In other words, an LLM can interpret its environment (via text prompts), formulate plans, and generate outputs that pursue a goal, effectively acting as an autonomous agent. Techniques like chain-of-thought prompting allow LLMs to break down problems into sub-steps, mirroring symbolic reasoning [70]. These capabilities enhance the autonomy of LLM-based agents as they can make intermediate decisions, reflect on outcomes, and revise their approach without hard-coded instructions. Tools such as LangGraph, AutoGPT, and MetaGPT operationalize these ideas by structuring LLMs into coordinated, multi-agent workflows, where specialized agents manage perception, planning, execution, and validation [71, 72]. Despite their generative nature, such systems still rely on classical MAS principles, including explicit role allocation, message passing, and internal state maintenance.

The theoretical richness of MAS makes it valuable in robotics and manufacturing as a conceptual framework. Instead of modelling each robot or controller as an agent, one can decompose the production tasks themselves into autonomous operational agents. Each agent represents a specific operation or function within the production system, equipped with local knowledge, goals, and the ability to act based on observations from the environment. The manufacturing environment serves as a shared world in which these operational agents interact. From this perspective, MAS theory provides an epistemological lens as it describes how knowledge, decision authority, and task responsibilities are distributed and exchanged among agents. MAS frameworks including formal role and organization models specify how information flows, how operations are allocated, and how agents coordinate or negotiate when objectives conflict or resources overlap [73]. This view aligns with recent works demonstrating that MAS can support human-centric and flexible manufacturing by enabling distributed decision-making and collaborative problem-solving [74, 75, 76].

MAS theory provides a principled foundation for managing distributed cognition and interaction in unstructured environments that demand dynamic decision-making. By abstracting operations and tasks as agents equipped with local knowledge and goals, MAS offers a clear conceptual basis for understanding how coordinated and adaptive behaviour can emerge. This perspective supports the use of MAS in manufacturing as a structured way to organize, reason about, and execute complex and evolving tasks in unstructured settings, as demonstrated in Paper VI.

2.4 Digital Twin

The concept of the Digital Twin (DT) has evolved from niche aerospace applications to become a central pillar of Industry 4.0 and smart manufacturing. While the terminology has gained massive popularity in recent years, the foundational idea dates back to NASA's Apollo program. During this era, NASA utilized "living models" (simulated replicas of spacecraft on Earth) to mirror the condition of assets in space. These early precursors ingested telemetry data to simulate hardware behaviour and diagnose failures that were physically inaccessible [77].

However, the formal conceptualization of the DT in the context of manufacturing and product lifecycle management is credited to Michael Grieves [78]. In 2002, Grieves introduced the concept as the "Mirrored Spaces Model," emphasizing the necessity of bridging the physical and virtual worlds to improve design, operational efficiency, and decision-making throughout a product's life cycle [79]. Since then, the definition has expanded. In the modern landscape, a DT is broadly defined as a virtual model of a physical object, process, or system that mirrors its real-world counterpart, facilitating simulation, analysis, and control [80, 81].

Despite the widespread adoption of the term, there is a common misconception in both academia and industry where any 3D model or simulation is erroneously labelled a "Digital Twin." This ambiguity necessitates a strict classification. A 3D CAD model or a stand-alone simulation does not inherently constitute a DT. One main differentiating factor is the flow of data. According to the classification by Kritzinger et al. [82], digital representations should be categorized into three distinct levels based on integration and data flow: the Digital Model, the Digital Shadow, and the DT as shown in Figure 2.2.

1. **Digital Model:** A Digital Model is a digital representation of an existing or planned physical object that lacks automated data exchange [83]. In this scenario, the data flow between the physical and virtual states is manual or insufficient. Any changes made to the physical object are not automatically reflected in the model, and vice-versa, simulations performed on the model do not automatically influence the physical object. This is typical of the design phase, where CAD models are used for geometric representation but remain static regarding the operational state of the physical asset.

2. **Digital Shadow:** A Digital Shadow represents a more advanced level of integration between physical and digital domains. It is characterized by an automated, typically one-way data flow between the physical system and its digital counterpart. Changes in the physical system are immediately reflected in the digital representation or, in some implementations, digital updates may be mirrored in the physical object. This structure is common in monitoring and visualization systems, where sensor data from the factory floor is continuously streamed into dashboards or analytic tools. Despite this real-time linkage, the interaction remains fundamentally unidirectional as one side observes while the other is observed. Although the digital model can analyse incoming data, it cannot autonomously initiate actions or modifications in the physical system without human intervention [82, 84].
3. **Digital Twin:** A "True" DT is defined by its fully integrated, bi-directional data exchange between the physical system and its digital counterpart. In this architecture, data flows automatically from the physical asset to the digital model and from the digital model back to the physical system. This bi-directionality elevates the system from a passive monitoring tool to an active control mechanism. The DT not only replicates the physical state in real time but also performs simulations, predictions, and optimizations in which outputs are used to generate control commands for the physical equipment [82]. This closed-loop interaction enables advanced capabilities such as predictive maintenance, autonomous reconfiguration, and real-time adaptation within unstructured and rapidly changing environments [85, 86].

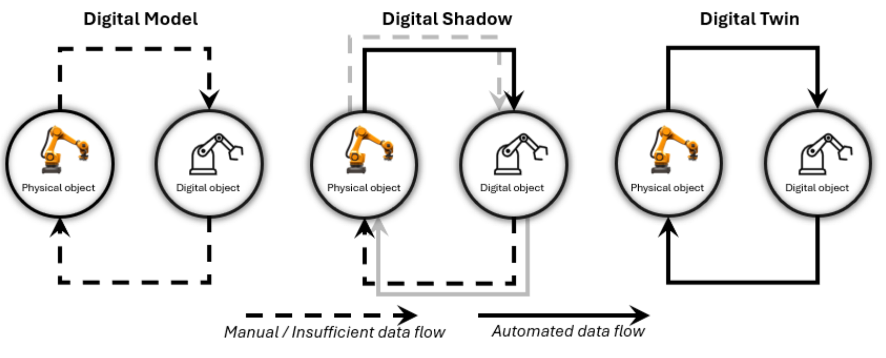


Figure 2.2: Overview of data flow for different digital concepts related to DT (adapted from Kritzinger et al. [82]).

While bi-directional data flow defines the structural maturity of a DT, its functional maturity is determined by its ability to act intelligently in dynamic and uncertain environments. As Industry 4.0 transitions toward Industry 5.0 with its emphasis on human-centricity, resilience, and adaptability, DTs must evolve from passive, expert-operated analytical tools into active, intelligent components of the manufacturing ecosystem. Hribernik et al. [18] propose that DTs must demonstrate three advanced capabilities to fulfil this role: context-awareness, adaptivity, and autonomy.

Context-awareness refers to the DT's capability to sense, interpret, and leverage information about its operational situation. In modern manufacturing, "context" extends far beyond geometric positioning to include system tasks, process goals, human presence, material states, and collaboration constraints [87]. A context-aware

DT understands the semantic meaning of incoming data [88]. For example, instead of merely logging an increase in temperature, it interprets this change within the context of a sensitive assembly operation and recognizes the associated production risks. This semantic grounding is essential for intelligent decision-making, ensuring that the virtual model remains aligned with evolving real-world shop-floor conditions.

Adaptivity enables the DT not only to understand changing conditions but also to respond to them. It is defined as the system's ability to modify its behaviour, parameters, or internal models in reaction to contextual shifts [18]. In unstructured environments particularly those involving human-robot collaboration, rigid or pre-programmed logic rapidly becomes inadequate. An adaptive DT dynamically reconfigures its simulations or control strategies when confronted with unforeseen variables, such as fluctuating sensor quality, irregular part placement, or deviations from standard human workflows [89, 90]. This ensures that the DT remains a reliable predictor and controller even when the physical system diverges from its nominal assumptions.

Autonomy marks the shift from a supportive tool to an active decision-making agent. An autonomous DT has the capability to pursue defined goals, individually or collaboratively without continual human oversight [18]. This transforms the DT from a reactive display into a proactive orchestrator. For instance, upon detecting a production bottleneck, an autonomous DT could re-route workflows, adjust robot task allocations, or initiate system reconfiguration, notifying human operators only when actions exceed its authority or safety boundaries.

2.5 Thesis Frame

This thesis is positioned at the intersection of KBE, DTs, and AI. Although these three disciplines originate from distinct research traditions, the work presented across the six papers demonstrates how they increasingly converge when addressing the challenges of customized products, unstructured environments, and adaptive robotic systems. Figure 2.3 illustrates how the individual papers contribute to these three domains and how the research trajectory evolved over time.

The first phase of the thesis (Papers I–II) is grounded in KBE, where DA and later design optimization were implemented using explicit engineering rules and parametrized models. As the research moved beyond structured design environments, the focus shifted to AI-based methods (Paper III). Reinforcement Learning was introduced to address decision-making in unstructured and dynamic production settings, conditions under which classical KBE approaches became insufficient.

The next phase expanded into DT technologies introducing a Digital Shadow, enabling one-way synchronization with the physical system. Paper V advanced this further by developing a full DT with two-way communication and embedding LLMs for interpreting user instructions and supporting perception-driven adaptation. Finally, Paper VI reconnects with the principles of KBE, but now within a DT context. Instead of relying solely on manually encoded rules, the knowledge base integrates observations and AI-derived interpretations.

Together, these three disciplines frame the methodological progression of the thesis: KBE provides structured knowledge, DTs provide synchronized virtual–physical representations, and AI contributes perception, and language-based reasoning. Their convergence establishes the theoretical basis for the work.

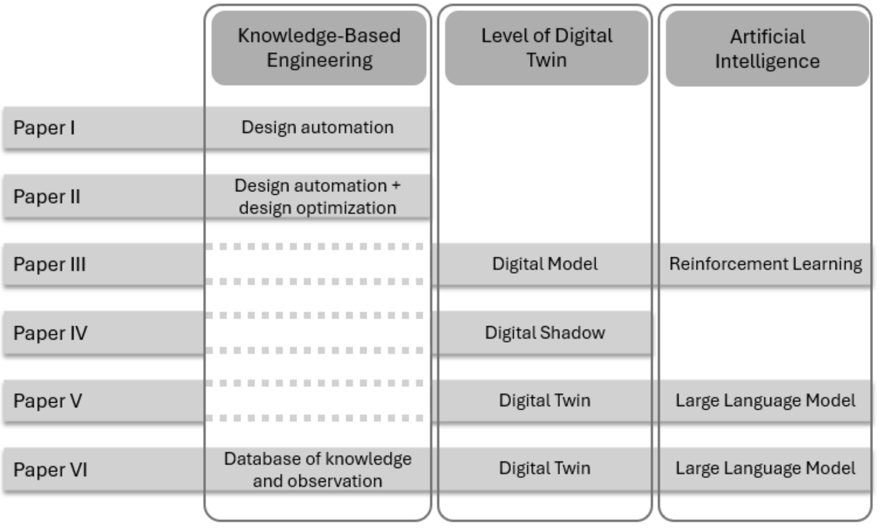


Figure 2.3: Papers positioned with respect to KBE, DT and AI.

CHAPTER 3

Applications

This chapter provides an overview of the papers and research undertaken during this PhD, highlighting their relevance and contribution to the aim of this thesis.

3.1 Paper I: Design Automation of Turbine Vane Cooling Channels

Problem Context:

Paper I represents the earliest stage of the thesis and originates from a product-development setting focused on gas turbine cooling technologies. The work targeted the creation of an automation solution for designing and analysing turbine vane cooling components, a domain that is highly structured but traditionally dependent on extensive manual engineering judgment. The motivation was to formalize this expert knowledge into a repeatable workflow capable of generating cooling geometries, preparing analysis models, and running simulations automatically, thereby reducing manual intervention and ensuring traceability across design iterations.

Workflow Architecture:

Paper I presents an automated workflow capable of generating cooling channel designs, producing meshes, executing structural simulations, and storing all related data for future reuse. A central enabler of the workflow is the High-Level CAD Template (HLCT), which captures morphological and topological variations through parametrized templates enriched with logical rules and embedded design intent.

Built on a Knowledge-Based Engineering (KBE) foundation, this approach establishes the structural basis required for future Multidisciplinary Optimization (MDO). At its core, the architecture links three major software tools: Siemens NX for CAD automation, Hypermesh for mesh generation, and Abaqus for finite element analysis through a shared database module and an inference script as shown in Figure 3.1.

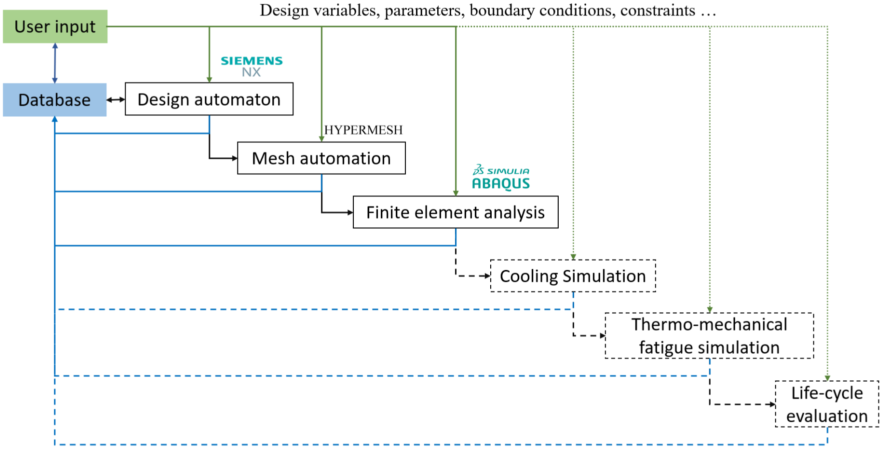


Figure 3.1: Overview of automation setup developed for design of turbine vane cooling channels.

To generate different cooling-channel configurations, HLCTs of turbine vane cooling components are first created in Siemens NX. A custom Visual Basics for Applications (VBA) automation script encodes the rules required to instantiate these HLCTs on the turbine blade. During execution, the script prompts the user for key topological and parametric inputs, which are then used to instantiate user-defined features and perform assembly operations. Although these parameters are specified at runtime, the Design Automation (DA) process relies on predefined geometric rules and annotations established in advance, as summarized in Table 3.1. Two-way communication between NX and a third-party database is enabled through the NX Journal programming feature, allowing scripts to run externally via BATCH execution and 'RunJournal.exe' application. Feature information, including types, names and unique IDs, is automatically extracted and stored in a back-end Excel database, supporting reuse in subsequent meshing and optimization steps. One of the generated CAD model with different cooling components after DA process is shown in Figure 3.2.

The next phase consists of mesh automation in Hypermesh and structural simulation in Abaqus. The resulting CAD file is imported into Hypermesh, where a fully automated TCL script accesses the database to retrieve mesh specifications, boundary conditions, and material properties. Once the model is imported, individual solid bodies are united for ease of downstream process. The meshing workflow includes: generating a volume tetrahedral mesh, assigning feature-dependent element sizes (e.g., finer meshes around cooling pins and holes), and applying material and physical properties. Boundary conditions are assigned automatically through a process that relies on the consistency of the geometric geometric annotations. The surface area computed by the meshing script is compared against the surface area documented in the database by the automated CAD model. Once a match is confirmed, the corresponding pressure or static load is automatically assigned to that surface.

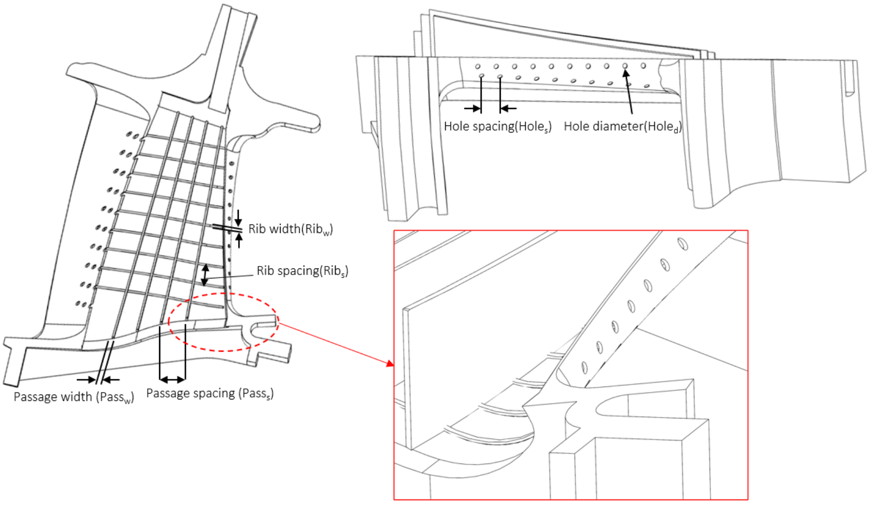


Figure 3.2: Final result of DA of cooling channels on turbine vane.

Table 3.1: Geometric annotations required for different user-defined features during DA

User-Defined Features	**Geometry Parameters**	**Annotations Required**
Passage Channel Walls	Number of channels	Point definition on tip and root chord
	Wall width	Plane definition for support
	Wall spacing	Two limiting bodies (suction side and pressure side)
	Location of closing wall along the chord at hub and tip	
Rib Structures	Number of ribs	Curve definition along chord
	Rib width	Plane definition for support
	Rib spacing	Two limiting surface definition (LE and TE inner geometry face)
	Location of rib as % of span	
Impingement Holes (LE and TE)	Number of holes in LE and TE	Line definition along LE
	Hole diameter	Plane definition for support
	Hole spacing	Two limiting bodies (LE and TE inner geometry)
	Location of hole as % LE and TE length	

Finally, the meshed model, with automatically assigned boundary conditions, is exported to the FEA solver. The simulation is executed through a batch script from the solver's native terminal. Upon completion, a Python script is used within the solver to analyse the results, recording key performance indicators such as maximum von Mises stress and deflection, which are then passed back to the back-end database for visualization and potential use in the upcoming MDO process.

Contribution to the Thesis:

Although Paper I is situated within a controlled engineering environment, characterized by stable part definitions and well-defined constraints, it provides a crucial foundation for the overall thesis. The study demonstrates that when domain knowledge is explicitly captured and structured, it becomes efficient to automatically generate design variants featuring both topological and morphological changes. While the framework is sufficiently flexible to generate variants for the turbine vane studied, it remains sensitive to changes in annotations or modifications in modelling standards. In such cases, the automation scripts must be updated to accommodate new geometric annotations or design rules.

Another important insight from Paper I concerns the integration of software from different disciplines. The use of a central database to manage and store data from each stage proved to be an efficient solution, enabling the various software components to operate independently while remaining coordinated within a unified pipeline. This approach to data handling would later be reused and extended in Paper VI, highlighting its relevance to the thesis trajectory.

Although the groundwork for multidisciplinary optimization was established, the full optimization cycle was not completed in this study. However, Paper I lays the methodological foundation for subsequent research, particularly Paper II. It demonstrates how KBE can formalize domain knowledge, enable automated design generation, and facilitate cross-disciplinary workflows, setting the stage for more complex, modular, and optimization-driven applications explored later in the thesis.

3.2 Paper II: Automation and Optimization of Fixture Planning

Problem Context:

Paper II builds upon the DA and KBE principles established in Paper I, applying them to a more modular and complex application: the design and optimization of fixtures in the Body-in-White (BIW) industry. The primary goal is to reduce design cycle time, minimize manual effort, and enhance the efficiency of fixture design through the integration of KBE, DA, robotic simulation, and design optimization.

Workflow Architecture:

The DA of modular fixture elements was executed in CATIA V5. Instead of using CATIA's built-in Visual Basic editor, the automation was accessed via Microsoft Excel, utilizing VBA. This external interface facilitated the control flow, starting with the automatic creation of product files and inserting the user-selected BIW model. The control flow for the DA process is illustrated in Figure 3.3 (left), demonstrating how the system converts user requirements into automated, configurable CAD models. Fixture elements were designed as modular, reusable components and assembled automatically. For instance, a single gripper/clamping unit consists of five distinct components: clamp, L-block, actuator arm, cylinder, and riser. Pick-up/locator units are simpler, featuring only a pin/locator and a riser, but share the same underlying code structure.

A critical aspect of the pick-up unit automation was the creation of a secondary axis system. While the primary axis was of the Euler type, the secondary axis was a Standard type, with its Z-axis aligned with the centre-line of a specific slot or hole in the BIW. The automation script used a forward chaining interface to measure the minimum distance between the Master Locating System input and all centre-lines in the BIW. The centre-line resulting in a measured distance of zero was dynamically selected as the reference for the secondary Z-axis. This rules-based approach ensured the correct and automated placement of locators based on geometric features.

A key characteristic of fixture design in BIW is its tight coupling with robotic operations. 3DExperience, in conjunction with the VBA, was used as a simulation platform for the spot-welding operation. The process leverages a Process, Product, and Resource (PPR) Context to define the manufacturing environment, which includes products (BIW parts), resources (robots, weld-guns, MFP, fixture), and operations (assembly directives). The automation of the robot spot-welding simulation follows a structured, four-step flow. First, a robot spot trajectory and a robot task is created, and a loop automatically adds all predefined Spot Weld Positions (SWP) into a new robot spot trajectory. Secondly, within the loop, the robot motion and spot-welding operations for each SWP are created and attached as the robot path. Before the final "Teach" task operation, the system checks for kinematic singularities by jogging the robot and modifying joint values to avoid operational issues. Finally, the robot task simulation is executed for visualization. Following the simulation, the "robot sweep", the volume covered by the robot's movement is extracted for use in the subsequent optimization process.

Following the DA and robotic simulation, the methodology employs a design optimization stage using modeFRONTIER to refine fixture layouts and parameters. The primary challenge addressed during optimization is minimizing clashes between fixture elements while ensuring efficient and safe manufacturing operations. The optimization setup is structured into two nested loops. The outer loop determined the total number of flagged elements in the assembly and selected a specific clamping unit's Product ID to be sent to the inner loop. The inner loop using the SIMPLEX algorithm focused on a single target (clamping unit), receiving the Product ID from the outer loop along with two optimization variables: 'Angle' (range 0° to 359°) and 'Clamp Length' (range 100 to 200). An Excel interface was used to bridge modeFRONTIER and CATIA. With each iteration, VBA scripts performed a clash test between the selected clamping unit and all other components, displaying the total number of clashes and the intersection volume between the fixture elements and the extracted

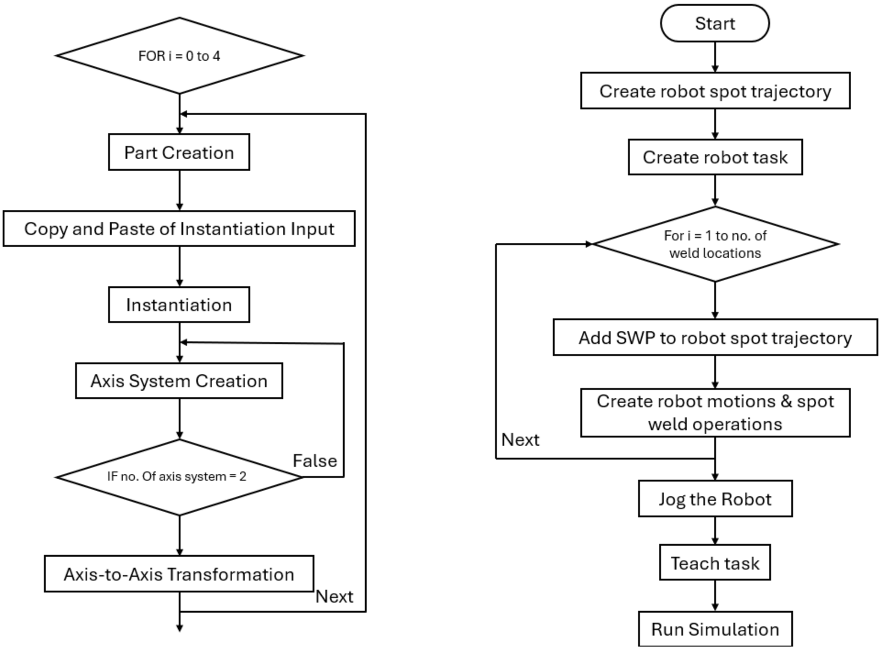


Figure 3.3: Illustration of the DA framework (left) and the corresponding automated spot-welding simulation process (right).

robot sweep volume. This clash information served as the key objective, allowing the system to identify the optimal fixture configuration that minimizes interference, particularly with the robot's operational space.

The combined framework from DA to robotic automation to optimization demonstrates the benefits of KBE in structured semi-static production scenarios (as shown in Figure 3.4). By automating repetitive tasks, ensuring consistency in design, and integrating multi-disciplinary workflows, the system reduces manual effort, improves precision, and accelerates the fixture design process.

Contribution to the Thesis:

While highly effective within well-structured engineering domains, the KBE-driven automation developed in Paper II remains limited in its adaptability to high variability in dynamic conditions. Its reliance on predefined templates, fixed robot trajectories, and rule-based optimization restricts its applicability when major variations or unforeseen task-level changes occur. If there is any topological change in the fixture configuration, change in the operation sequence of the robot, or the trajectory the robot takes while performing the spot welding operation, the optimization process has to be redone. Moreover, the limitations identified in paper I regarding the error prone nature of the rule-based approach with mismatched geometric annotations are present in the results presented in Paper II as well.

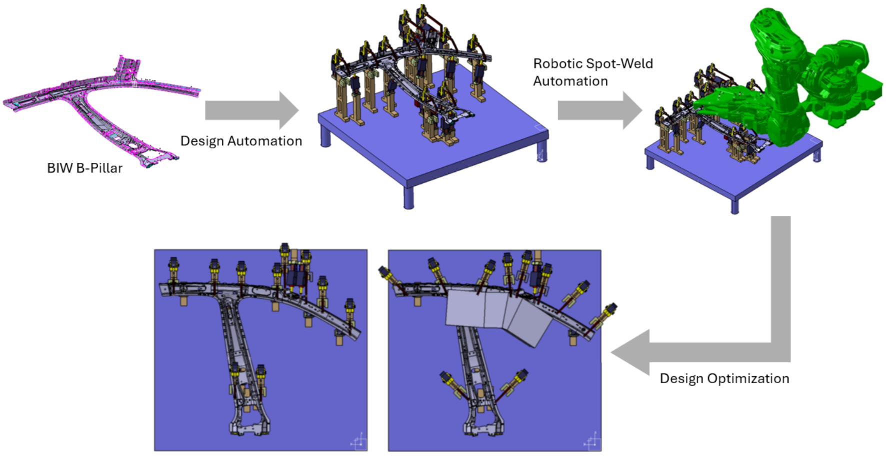


Figure 3.4: The final output generated through the combined processes of DA, robotic simulation, and optimization on BIW fixture assembly.

More fundamentally, classical KBE systems struggle in environments where tasks shift frequently, new product variants emerge unpredictably, or contextual decisions rely on tacit knowledge that is difficult to formalize. These constraints underscore a key gap between traditional rule-based automation and the needs of modern flexible manufacturing. Recognizing these limitations motivates the progression of the thesis toward adaptive, perception-driven, and learning-based automation approaches explored in Papers III–VI, where sensing, reasoning, and environmental feedback become central to managing variability and Human-Robot Interaction (HRI).

3.3 Paper III: Automation in Unstructured Production Environment Using Reinforcement Learning

Problem Context:

Publication III represents a turning point in the progression of this thesis. While Papers I and II demonstrated the strengths of KBE and rule-driven optimization in structured engineering environments, they also exposed a critical limitation such methods presuppose, stability. Rule-based automation functions effectively when geometries, workflows, and constraints remain predictable, but it rapidly breaks down when manufacturing environments become dynamic, when layouts shift, obstacles appear unexpectedly, or human–robot interactions influence task execution. Even minor deviations can require rewriting rules, reconfiguring templates, or rebuilding automation logic.

Paper III explores reinforcement learning (RL) as an alternative approach capable of overcoming this restriction. Instead of encoding task execution logic explicitly through rules, constraints, or parametrized workflows, RL enables agents to learn task-relevant behaviours directly from environmental interactions. This shift reframes automation from a rule-driven paradigm to a data-driven, experience-driven one. In this sense, Paper III can be viewed as "ahead of its time" in the overall trajectory of the thesis, acting as a conceptual precursor to the final move toward multi-agent systems and reasoning-capable agents discussed later in the dissertation.

Application Cases:

To investigate the feasibility of RL for industrial tasks, Paper III introduces two proof-of-concept applications implemented in a lightweight game engine (Unity):

1. A mobile robot performing path-finding in a dynamic workspace, and
2. A mannequin executing sheet-metal assembly operations using a combination of RL and inverse kinematics (IK).

These two applications were intentionally selected to reflect challenges that rule-based methods cannot easily handle: dynamic obstacles, non-deterministic sequences of actions, and tasks involving human-like motions or reasoning.

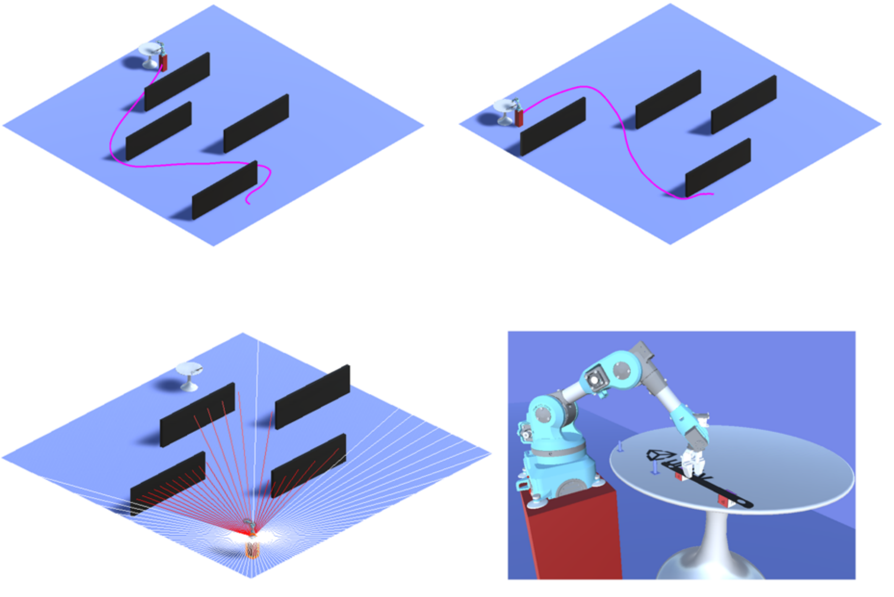


Figure 3.5: RL-driven path-finding for a mobile platform operating in a dynamic environment (top), informed by ray-perception observations (bottom left), enabling autonomous pick-and-place operations of sheet metal onto a fixture (bottom right).

The first application focuses on training an RL agent to navigate a mobile robot through an unstructured environment. Unlike classical path-planning algorithms, which require static maps or fixed graph structures, the RL agent learns its navigation strategy through trial-and-error interactions. As illustrated in Figure 3.5, the robot is placed on a mobile platform equipped with ray-perception sensors that serve as observations, detecting obstacles, table boundaries, and potential collisions. Vector observations such as the distance between the robot and the fixture, the distance to the pick-up object, and the velocity of the robot platform is used. Motion of the robot platform was controlled using continuous force vectors. Since the movement was restricted to a flat plane, a force vector of $(X, 0, Z)$ was used, where X and Z were continuous vector actions controlling the movement along the plane. During training, the starting position of the robot and the table is randomized along with the positioning of obstacles in the environment. This constant change ensures that the environment remains stochastic and unpredictable. The task concludes when the robot successfully approaches the table and performs the pick-and-place operation with the onboard manipulator. RL agent learns navigation policies that generalize across a wide distribution of scenarios without requiring a pre-labelled dataset or predefined rules.

The second application focused on controlling the hand and torso of a humanoid mannequin to perform a sheet metal assembly operation (pick a sheet metal and place it on a fixture). This required implementing IK using Unity's Fast IK Fabric constraint to achieve human-like motion. Since Unity's IK constraints cannot be directly manipulated by physical forces, a dummy game object with a rigid body component was created. This dummy object's transform position and rotation were linked to the mannequin's palm, allowing the agent to control the motion of the rigid body, which the mannequin's hand then followed. The agent was intended to perform three sequential operations in a single episode: reach the sheet metal target and create a fixed joint (pick-up), move the hand and the attached sheet metal to the drop-off location (on top of the fixture), and align the sheet metal accurately at the drop-off location. The entire workflow is illustrated in Figure 3.6.

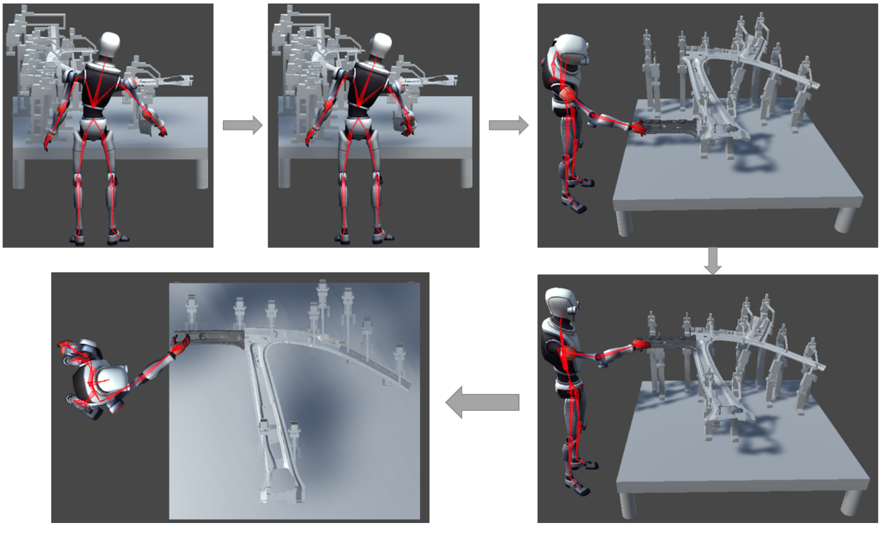


Figure 3.6: Stepwise operations of a trained mannequin performing a pick-and-place task: detecting and approaching the object, grasping it, transporting it to the designated drop-off area, positioning at the drop-off point, and finally releasing the object.

Due to the mannequin's multiple degrees of freedom and the challenge of camera-based observation failing in non-line-of-sight scenarios, a set of five Ray-Perception sensors was attached to the dummy game object. These sensors were configured with specific radii, lengths, and offsets to detect objects within the mannequin's arm's length. Tags were used to identify the sheet metal, fixture, and table separately. The observations also included conditional vector data: the distance between the palm and the sheet metal (recorded until pick-up), followed by the distance between the sheet metal and the drop-off location. A Boolean value tracked the completion of the pick-up operation. The actions were defined as discrete values (ranging from 1 to 6) corresponding to the six degrees of freedom. These discrete actions were used to generate a force vector to apply a fixed velocity to the rigid body of the dummy object, controlling the palm's motion.

The Proximal Policy Optimization algorithm was selected for both environments. Due to the mannequin environment's higher number of observations and multiple sequential rewards, it required a higher buffer size to collect more complex experiences before updating the policy. Consequently, the mannequin control training required 30 million iterations, significantly more than the 2.55 million steps used for the mobile robot, which had a single primary objective.

Contribution to the Thesis:

Firstly, Paper III empirically reinforces the observation that conventional KBE and rule-based DA excel only in stable, well-structured conditions as they cannot reliably adapt when environmental boundaries shift. Second, it introduces RL as an alternative automation paradigm in which agents learn behaviours autonomously rather than relying on pre-structured knowledge bases. This transition from "automation by rules" to "automation by experience" marks an important conceptual shift in how unstructured tasks are approached.

Finally, the work points toward a future in which multiple agents/robots, digital humans, and sensors collaborate in dynamic environments, perceiving changes, interpreting the scene, and acting without strict dependence on predefined workflows. These ideas directly influence the later chapters, where RL-inspired reasoning is combined with multi-agent decision networks and real-time digital twin environments. In this sense, Paper III serves not only as an early prototype but as a conceptual stepping stone toward adaptive, reasoning-oriented automation solutions capable of handling the unpredictability of dynamic manufacturing contexts.

3.4 Paper IV: Digital Shadow for Adaptive Robot Control

Problem Context:

Paper IV moves the thesis forward by shifting the focus from purely rule-based digital modelling to the control of physical robot systems using simulation-driven adaptability. Earlier papers highlighted the limitations of static KBE methods and showed how

RL agents can learn behaviour in unstructured virtual environments. In contrast, Paper IV tackles a different but equally important challenge: understanding how physical behaviour can be captured in a virtual model and then used to guide adaptive decisions in real robotic systems. As a result, this work sits at a methodological crossroads connecting automation logic, physics-based simulation, and the growing role of digital twins (DTs) in unstructured production environments.

The paper performs an extensive background review on DT technologies, unstructured production environments, and the requirements for connecting virtual and physical systems. A key insight from this review is that, in unstructured environments, adaptability depends not only on decision-making logic but also on the ability to replicate physical interactions (contact, force, collisions etc.) inside a digital environment. Without such fidelity, the system cannot evaluate or react to the uncertainties inherent in real production. This motivated the use of a physics-based simulation engine (Isaac Sim) as the core representation of reality.

Architecture and Experimental Setup:

Based on these insights, Paper IV introduces a flexible and modular framework that integrates NVIDIA Isaac Sim with ABB YuMi robot to form a digital shadow, a precursor to a fully realized DT. The target application is drawn from a hospital test laboratory, where microscope slides must be stacked from a slide fixture to a slide holder. The process involves fine manipulation of small objects with varying poses, exactly the type of task that becomes difficult to handle when performed through predefined rules or traditional offline programming. The full framework, shown in Figure 3.7, is built around three interconnected modules: a simulation module containing the physics-based virtual environment, a communication module using ROS to exchange data between simulation and hardware, and a robot controller module for YuMi.

The digital environment is built in Isaac Sim using CAD models converted to USD (Universal Scene Description). To achieve the high precision required for handling thin glass slides, the framework utilizes Signed-Distance-Fields (SDF) for collision detection. Unlike standard bounding boxes, SDF meshes provide high-detail triangular approximations for dynamic rigid bodies, allowing the simulation to accurately calculate the interaction between the robot's fingers and the slides. The robot itself is imported via a Unified Robotics Description Format (URDF) file, which was manually modified to include an extra link with a sphere mesh at the end-effector to facilitate precise tracking.

The main enabler for the adaptive behaviour in this work is the decider network with Isaac Cortex architecture as shown in Figure 3.8. This network acts as the 'brain' of the simulation, governed by Robust Logical Dynamic Systems (RLDS). The RLDS algorithm employs a backward-chaining logic: it starts from the desired end-goal (e.g., "Drop the slide") and evaluates the current environmental conditions to determine which prerequisite behaviour is currently 'runnable'. The network is structured into hierarchical nodes, beginning with the Dispatcher/Assembly Logic, which serves as the central monitor for the assembly's overall state. Based on this data, the Dispatcher determines whether the system should initiate a 'Pick' or 'Drop' cycle. When a pick cycle is triggered, the system shifts to the Pick Logic, which operates through a series of prioritized internal nodes. Within this hierarchy, the 'Open Gripper' node

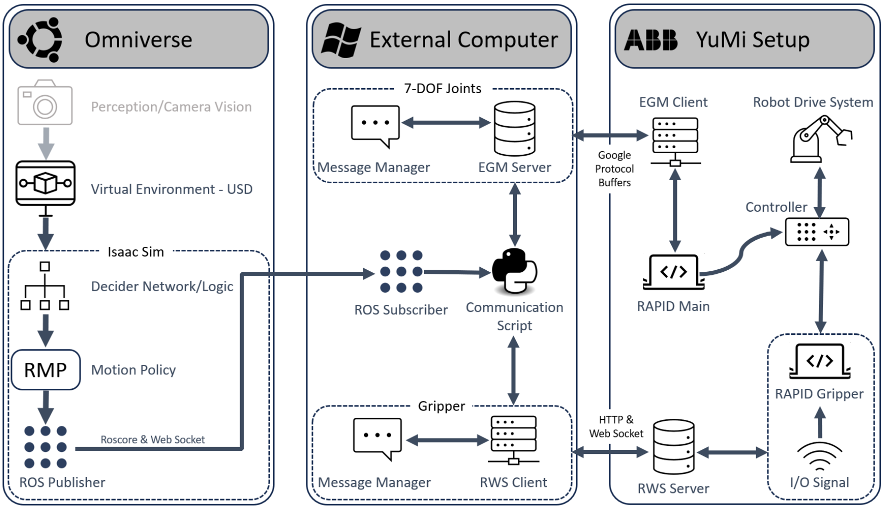


Figure 3.7: Overview of the digital shadow framework, integrating the simulation module (Omniverse), communication module (external computer), and robot controller (YuMi).

holds the highest priority to ensure the hardware is prepared for engagement, followed by the 'Pick Slide' node and finally the 'Reach to Slide' node.

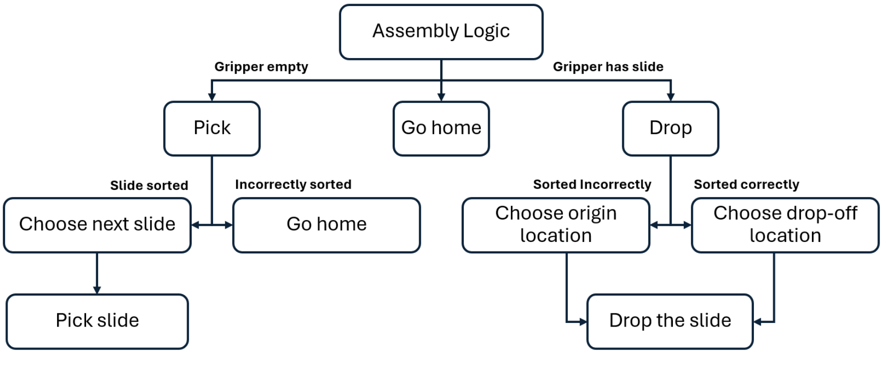


Figure 3.8: Decider network responsible for determining robot actions based on environmental observations, robot status and the operation status.

While the decider network handles the logic, the physical movement is generated by a Riemannian Motion Policy (RMP). This is a collision-aware motion generation algorithm that calculates the necessary joint actions for every frame of the simulation. To ensure the robot does not collide with other objects in the environment or itself, the framework uses a supplementary robot description file. This file maps 'collision spheres' onto the robot's kinematics. These spheres act as a protective envelope around the 7-DOF arms. The RMP engine continuously solves for the target position while ensuring these spheres do not intersect with any other objects in the environment, allowing the robot to navigate complex, cluttered spaces safely.

During simulation, the joint positions and velocities calculated by the RMP are published as ROS topics. A Python-based communication bridge subscribes to these topics and serializes the data using Google Protocol Buffers. This data is then streamed via UDP to the ABB controller's Externally Guided Motion (EGM) module. By utilizing the EGM Position Guidance feature, the framework bypasses the robot's native path planner, allowing the physical arms to follow the simulated movements with a latency of 10–20 ms. Since EGM cannot natively control YuMi's smart grippers, a secondary channel was developed using Robot Web Services (RWS). The system monitors the gripper's joint states within Isaac Sim. When the decider network triggers a close/open gripper action, a signal is sent via a RESTful API call to the controller. This modifies a Digital Input on the robot, which in turn triggers a RAPID trap routine to actuate the physical gripper. To demonstrate dynamic behaviour, the positions and orientations of both the slide fixture and the slide holder were varied in simulation, along with changes to the required sorting sequence. The system successfully sorted slides under all these variations, confirming the adaptive capability of the digital model.

Contribution to the Thesis:

The primary contribution of this paper lies in establishing a foundational framework for adaptive robotic automation in unstructured production environments, effectively bridging the gap between traditional rule-based automation and a dynamic responsive systems. By integrating NVIDIA Isaac Sim with the ABB YuMi robot, the study demonstrates the potential to replicate physical behaviours in the digital domain, simulate complex interactions, and enable dynamic decision-making through a decider network. However, the absence of real-time perception of the scene and the inability to monitor user actions limit this work to a digital shadow rather than a fully realized DT. Without real-time perception and environment awareness, automation remains confined to predefined scenarios, even when supported by physics-based simulation platforms and adaptive decision-making. The study therefore clarifies that digital shadows alone are insufficient unless coupled with robust perception, reasoning and feedback mechanisms.

By making these limitations explicit, Paper IV plays a critical role in shaping the trajectory of the thesis. Rather than presenting a complete solution, it functions as an exploratory step that identifies essential requirements for adaptive automation in unstructured settings, namely real-time perception, structured feedback loops, robust communication handling, easy HRI and scalable system architectures. These insights directly motivate the subsequent integration of perception systems, knowledge structuring, and higher-level reasoning mechanisms in later studies (Paper V and VI). In this way, the paper contributes both by demonstrating the potential of DT–based automation and by clarifying the limitations that must be addressed to move towards a truly adaptive and reliable automation solution for robot control.

3.5 Paper V: Digital Twin for Adaptive Robot Control

Problem Context:

Building on limitations identified in earlier work where adaptability existed only in simulation and no real-world changes could be synchronized, this paper advances the overarching vision of the thesis: to develop adaptive automation architectures capable of operating reliably in unstructured, variability-intensive environments with minimal human intervention. Hence, this paper introduces two major components essential for real-world adaptability: a real-time vision-based perception module and a natural-language task interface powered by a locally deployed Large Language Model (LLM). The study is motivated by a real-world case in a hospital laboratory, where robots must interact with medical staff who lack robotics expertise. The task of sorting microscope slides is repetitive and time-consuming, and its execution is subject to frequent variations introduced by human operators. Such a setting exemplifies the central challenge of unstructured environments: tasks cannot be defined exclusively through static, predefined rules, and continuous adaptation is required as both the task description and physical state of the environment evolve.

Architecture and Experimental Setup:

The paper implements a camera-based perception module that captures the positions and orientations of objects in the physical workspace and updates the digital model accordingly. The system uses an Intel RealSense D435i depth camera and a custom computer vision algorithm to reconstruct the workspace in the DT. The perception workflow follows a structured sequence that begins with Tray Detection, where the initial RGB image undergoes preprocessing including greyscale conversion, Gaussian blurring, and binary thresholding to identify tray boundaries. By fitting a minimum area rectangle to these contours, the system extracts the four corners to define a precise Region of Interest (ROI). Once the ROI is established, the process moves to Slide Segmentation, utilizing colour segmentation within the HSV colour space to identify slides based on predefined colour masks and calculate their centroids for localization. This is followed by Spatial Mapping, where the system calculates midpoints along the tray edges to accurately map each detected slide to its specific segment. Finally, to ensure alignment, the algorithm detects two specific screws on the robot's base to compute a direction vector between these fixed points and the tray. This calculation determines the precise relative position and angle, ensuring the virtual tray is correctly oriented for the robot's movements. All captured data including tray orientation, slide count, positions, and colours is published as a ROS node, which the Isaac Sim environment subscribes to for real-time scene initialization.

Humans alter workflows through ad hoc decisions, shifting priorities, and inconsistent execution patterns. By incorporating a speech interface supported by an LLM, the system can interpret natural-language instructions directly from users and convert them into actionable robotic tasks. Traditional offline programming approaches rely on predefined rules, making them unsuitable for continuously changing environments. To address this Paper V incorporates a locally hosted LLM (Mistral 7B) via the Ollama runtime, enabling intuitive Natural Language HRI through voice commands. The speech-to-task pipeline begins with Speech Transcription, where user input is captured and converted into text using the Google Speech Recognition API. This text is then processed and embedded into a structured prompt designed to minimize hallucinations. The LLM is strictly instructed to output a specific format, such as "Sorting; [colour, colour, ...]" as shown in Figure 3.9. To maintain system stability and performance, the LLM is not executed asynchronously, which prevents it from degrading simulation performance, frame rates, or physics calculations. Instead, the process is triggered via a button-based interaction prior to the simulation start, ensuring the task logic is fully defined before the robot begins moving. Finally, this structured sorting sequence is passed to the decider network, which updates the robot's pick-and-drop logic to reflect the user's specific preferences.

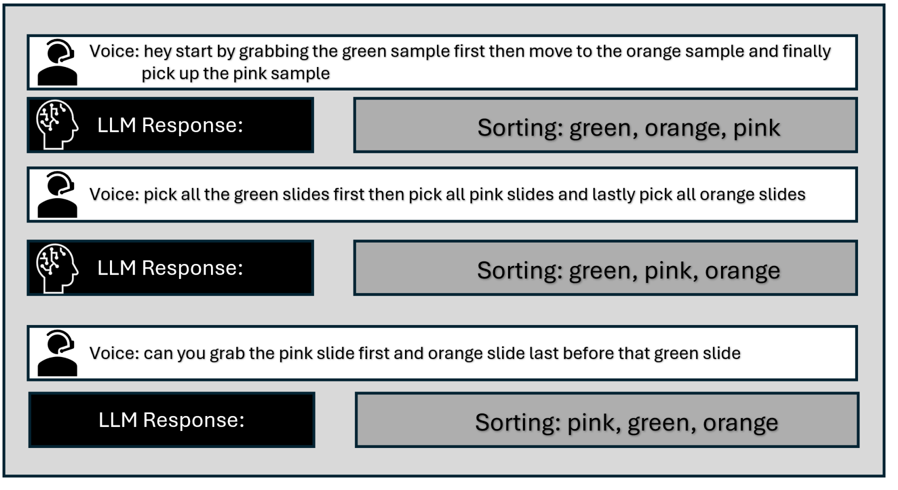


Figure 3.9: Illustration demonstrating the LLM's core function: interpreting human verbal inputs (natural language) and translating them into formal, actionable sorting instructions for the simulation model.

The virtual environment in Isaac Sim serves as the synchronization point for all modules. Once the perception module aligns the virtual scene with the real-world setup and the LLM provides the sorting sequence, the simulation is initialized. The framework maintains the RLDS-based decider network and RMP based motion planning used in Paper IV. The decider network uses the LLM-generated sequence to determine which slide to pick next, while the RMP ensures all movements are collision-free relative to the detected tray position. The communication bridge remains consistent with the previous study, utilizing EGM for high-frequency 7-DOF joint control and RWS for gripper actuation. By integrating these modules, the framework achieves a DT that "sees" through the perception module, "understands" through the LLM, and "acts" through the physics-aware simulation and EGM control as illustrated in Figure 3.10.

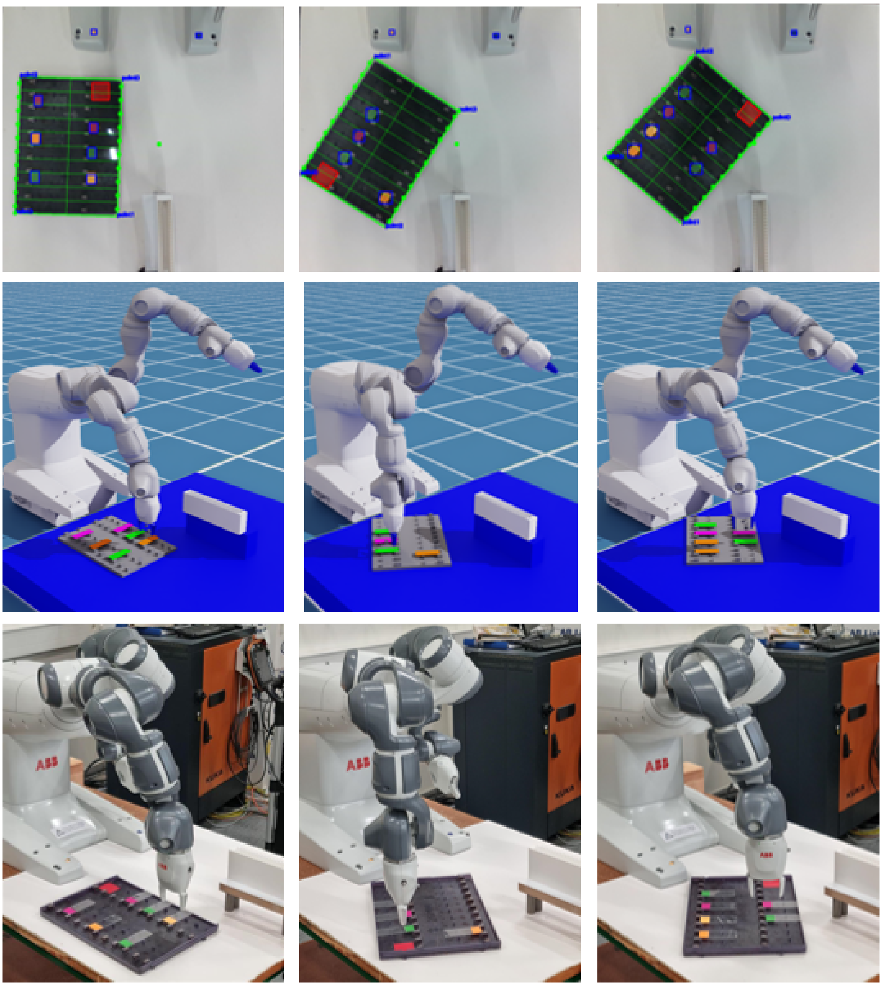


Figure 3.10: Illustration of the picking sequence demonstrating the real-time synchronization between the physical and digital environments with randomized positions, orientations, and quantities of the microscope slides within the tray.

Contribution to the Thesis:

Although achieving only 60% success rate while validating the whole setup, the current work extends the foundations of Paper IV by connecting the digital model to the real world realising a DT. Despite the advancements demonstrated, several limitations shaped the direction of subsequent studies in the thesis. One major challenge was the susceptibility of the LLM to hallucinations, which introduced inconsistencies in task interpretation and limited the system's reliability. Additionally, running all modules (simulation, perception, reasoning, and control) on a single machine created performance bottlenecks that constrained scalability. These limitations highlighted the need for a more structured and distributed architecture. The paper VI builds directly on these identified challenges, addressing them through a more modular, knowledge-driven, and multi-agent DT architecture that advances the system toward the goal of fully adaptive automation in unstructured production environments.

3.6 Paper VI: Digital Twin Enabled Adaptive Robot Control Using Database and Multi-Agent Approach

Problem Context:

The work presented in this paper represents an important step in advancing the overarching aim of the thesis, to develop automation strategies capable of handling the variability, uncertainty, and complexity inherent in unstructured manufacturing environments. Across the thesis, each preceding study gradually addressed different limitations of traditional automation workflows, ranging from rule-based programming and the absence of perception, to challenges in handling unstructured data and unreliable reasoning. Paper VI extends this trajectory by transforming the earlier DT and LLM-driven approach into a more modular, data-centric, and multi-agent architecture.

While this paper builds on the same application explored in Papers IV and V, it also introduces a second case involving the stacking of blocks of different shapes and colours to illustrate the scalability of the proposed automation solution. Limitations identified in Papers IV and V, particularly issues related to scalability, computational load, and system latency, motivated a reorganization of the architecture. Another key observation from earlier work, especially the exploration of LLM-based interaction, was the persistent challenge of hallucination. Addressing this issue became central to the development of a more structured and verifiable system.

Architecture and Experimental Setup:

Compared with the setup used in Paper IV (Fig. 3.7), the framework evolved significantly as the research progressed. The new architecture involved several resource-intensive modules being offloaded to separate computers as shown in figure 3.11. This reduced the burden on the simulation machine and enabled more stable real-time operation.

A major advancement introduced in Paper VI is the integration of a structured database that acts as a shared knowledge base for the entire system. This database not only stores all information required for the automation to operate but also separates parameters, operational metadata, and interaction logs into dedicated tables. Parameter tables contain all operation-specific variables generated automatically during simulation, while still allowing user-defined modifications where necessary. Interaction tables serve as the interface for the LLM agents, storing inputs, processed outputs, and intermediate reasoning steps.

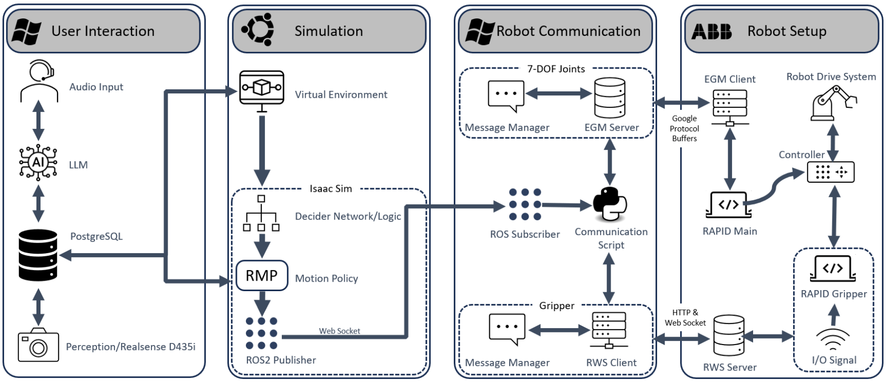


Figure 3.11: Overview of the DT framework, integrating user interaction module (LLM agent, perception and database), simulation module (Omniverse), communication module (external computer), and robot controller (YuMi).

To address the lack of modularity, robustness, and scalability in earlier LLM and object-detection components, Paper VI introduces a multi-agent architecture using LangGraph, structured around state- and node-based programming as shown in figure 3.12. The resulting system allows flexible activation of perception modules depending on the task requirements, while simultaneously supporting natural-language instructions from the user. The LLM agents interpret these instructions, decompose them into executable tasks, generate a task plan compatible with the decider network (figure 3.8) in the simulation, and retrieve relevant information from the database to respond to user queries. Because all data from robot status to perception outputs is stored in a structured manner, the LLM agents can reliably query the database and provide accurate, context-aware feedback in natural language. By grounding all information in a structured and query-able format, the system reduces ambiguity and provides the LLM agents with a reliable source of truth.

In previous approaches, modifying operation parameters required embedding new rules directly into the algorithms, which became increasingly unmanageable as complexity grew. In the new setup, once the task planner creates a sequence of operations, users can adjust parameters of individual steps through natural-language commands. For example, the direction of approach for picking a specific object can be altered by having the LLM agent modify the corresponding entry in the parameter tables. However, at this stage, the agents do not yet verify whether such modifications are feasible or safe to execute. Ensuring the validity of parameter changes therefore still relies on the user.

Contribution to the Thesis:

The results of Paper VI demonstrate that while the proposed architecture represents a marked improvement, it does not yet reach the level of robustness required for industrial deployment. A success rate of around 75% shows clear progress compared with earlier studies, but remains below the threshold expected in real-world scenario. The identified bottlenecks such as agent-coordination overhead, SQL scalability limitations, camera occlusions, and simulation-derived inaccuracies illustrate

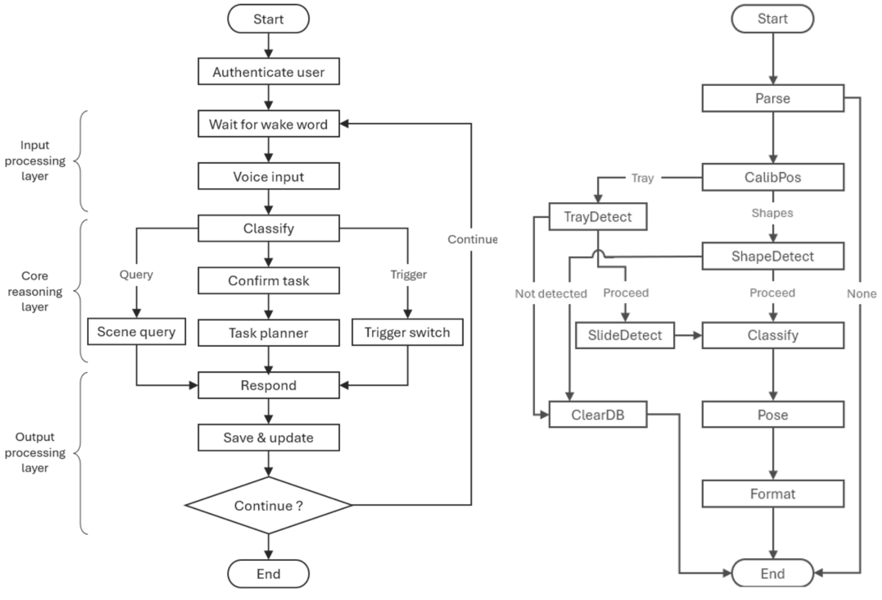


Figure 3.12: Stateful execution graph of the multi-agent LLM architecture (left), highlighting its input-processing, reasoning, and output-processing layers, alongside the execution graph of the multi-agent camera-vision system used across both applications (right).

that adaptive automation for unstructured environments remains an open research challenge. Nonetheless, the work provides both an advanced architectural solution and a clearer understanding of the gaps that must be addressed. In this sense, Paper VI represents not the conclusion of the thesis, but an opening towards new perspectives. It highlights that achieving highly adaptive automation requires more than improved perception or stronger models, it requires an integrated ecosystem that unites structured knowledge, perceptual intelligence, multi-agent reasoning, and human-centric interaction. The findings point toward future research directions centred on enhancing agent coordination, improving real-time perception, and incorporating safety-aware control mechanisms to support reliable deployment in industrial settings.

CHAPTER 4

Discussion

This chapter discusses the results and limitations of the work performed within this thesis. It discusses the methodological approach taken, details the research contributions, and concludes with potential future work based on the insights gained throughout the studies.

4.1 Analysis of Results

The progression of studies presented in this thesis illustrates a gradual transition from conventional automation techniques toward adaptive, simulation-driven, and reasoning-enabled automation frameworks capable of supporting highly customized and dynamically changing manufacturing environments. Across the papers, the results reveal both the technical viability of the proposed approaches and the constraints that shaped the evolution of the research, ultimately guiding the direction toward adaptive Digital Twins (DTs) and perception-driven robotic systems.

4.1.1 Rule-Based Design and Production Automation

Work from Paper I and Paper II highlights the strengths and fragilities of traditional Knowledge-Based Engineering (KBE), rule-based reasoning, and multidisciplinary Design Automation (DA) in structured industrial contexts. Paper I demonstrated that KBE frameworks can yield highly accurate, flexible and repeatable results for complex design tasks, such as automating topology- and morphology-varying cooling channel configurations, provided that geometric variability is moderate and the relationships among design, meshing, and analysis are well-defined. However, the automation setup in Paper I, while effective for creating topological and morphological cooling design variants, is rule-based and model-specific to the turbine vane geometry. As explained in Section 3.1, the automation algorithm is parametrized by the annotations and geometric properties of a particular turbine vane model. Because the automation pipeline is tightly coupled to the specific annotations and geometric parameters of the turbine vane, deploying the setup on a variant with significantly different labelling or geometric characteristics leads to errors. Such deviations require extensive reconfiguration of predefined rules, annotated objects, and conditions across the entire workflow. While the approach provides good return on investment for optimization when applied to a single, consistently modelled turbine vane, its dependence on rigid annotation structures makes adaptation to other vane models labour-intensive and error-prone.

Paper II further highlighted the inherent dependency between DA and production automation when rule-based logic is used to configure robotic operations and fixture design. In Paper II the optimization works effectively for a model with a fixed robot trajectory and operation sequence. However, if either of these production parameters is varied, the entire optimization process must be performed again. Similarly, any topological changes in the fixture design configuration also necessitate a complete re-optimization. This strong reliance on the rule-based approach restricts the solution's flexibility when variability increases, whether due to a change in the product design, annotations or a change in the production process. Such restrictions underscore the need for automation solutions that offer flexibility in decision-making and are not constrained solely by predefined rules.

4.1.2 Learning-Based Adaptation with Reinforcement Learning

Insights from Paper I and II established the need for more adaptive and learning-based methods, leading to the exploration of Reinforcement Learning (RL) implemented within a physics-based game engine. The experience- and learning-driven approach of RL was applied to the fixture assembly task (from Paper II), training an agent (mannequin) to assemble a sheet metal part. By training the agent across a varying design space, the resulting agent could make assembly decisions based on environmental observations to achieve its goal. Unlike rule-based systems requiring modification of rules or a complete re-run of optimization, the trained RL agent exhibited adaptability when the environment changed. For reasonable changes (e.g., a new layout or geometric change in fixture), the agent could be retrained to incorporate the new experience, offering greater flexibility. However, the agent's performance decreased as environmental variability increased substantially. The agent's behaviour, driven by a reward system, was also susceptible to exploitation. For instance, negative rewards for collision between mannequin and fixture caused the agent to keep its arms stationary or distant to avoid punishment, often failing to pursue the primary assembly goal. Such unpredictable nature of this approach make it unreliable in scenarios requiring high degrees of control, safety, and determinism. While pure RL has limitations, the experience-based decision-making capability could be valuable when applied to a small and modular level with layers if validated to ensure robustness. Most importantly, the use of the physics-based simulation engine to replicate real-world physical behaviour showed promising quality, giving insights and direction for the subsequent research.

4.1.3 Simulation-Driven Automation and Need for Closed-Loop Digital Twins

The initial exploration using Unity for industrial simulations proved insufficient for achieving the necessary fidelity in robot operations, largely because Unity is optimized for game development rather than high-precision robotic control. Nevertheless, Paper III demonstrated the value of using a physics-based simulation engine to replicate real-world behaviour and validate automation concepts rapidly. Building on this insight, Papers IV-VI transitioned to NVIDIA Isaac Sim, a platform explicitly designed for robotics, enabling higher realism and tighter integration with industrial workflows. Paper IV developed a simulation-driven adaptive framework incorporating a decider network that allowed the simulated robot to perform the microscope-slide sorting task adaptively within the virtual environment. However, the system operated solely within the simulation, lacking closed-loop integration with the physical world. As a result, changes in the physical environment such as repositioning of slide fixture were not reflected in the simulation, leading the robot to act on old state information. This limitation highlighted a critical requirement: an adaptive system must continuously gather real-world data, synchronize it with the digital environment, and react accordingly to maintain robust operation under variability.

Additionally, the setup could only be operated by users with substantial programming and simulation expertise. Minor adjustments, such as modifying a gripper orientation, required editing the underlying decider-network code. This not only restricted usability but also hindered scalability and operator autonomy. The resulting insight was that any practical automation solution must provide an intuitive interface that allows non-expert users to modify operational parameters without interacting with the underlying algorithms. These limitations and insights from Paper IV directly shaped the objectives of Paper V, motivating the integration of perception modules and natural-language-based interaction to move toward a fully closed-loop DT capable of adaptive behaviour both in simulation and in the physical environment.

4.1.4 Perception and Reasoning Challenges in Early Adaptive Digital Twin

The limitations identified in Paper IV shaped the objectives for Paper V, particularly the need to integrate a perception system capable of capturing real-world information and reflecting those changes within the digital environment. In parallel, a reasoning module was required to interpret user instructions and translate them into inputs for the decider network developed earlier. While this implementation allowed the system to see the environment and understand user intent, it also introduced new challenges. The most significant was the tendency of the integrated Large Language Model (LLM) model to hallucinate. Because the LLM had only restricted access to information, often limited to code blocks rather than structured data, it was unable to validate user inputs against ground-truth observations. As a result, incorrect or misleading user instructions often went unverified, producing unreliable or inconsistent outputs.

The perception system, based on contour-based object detection, showed similar constraints. It's rule-based nature meant that it worked well only when objects matched predefined characteristics. Variations caused by occlusion, poor lighting, or the introduction of unfamiliar objects led to detection failures, breaking the link between the physical and digital environments. These issues reflect a broader limitation that both the LLM and the perception system relied heavily on rigid, case specific rules, which limited adaptability and scalability. Applying such solution to a new task would require rewriting prompts, perception algorithms, and validation logic. Although this setup enabled an early realization of a DT capable of closed-loop operation, the overall reliability remained insufficient, achieving only a 60% success rate in validation studies, highlighting the need for more robust, scalable, and data-driven approaches in subsequent work.

4.1.5 Structured Knowledge-Base and Multi-Agents as Mitigation Strategies

Implementing a structured database to manage perception and simulation data in Paper VI significantly reduced hallucinations in the LLM module. By limiting the LLM's access to clean, structured data and parametrized information rather than entire code blocks, the model's reasoning became more consistent and the decision-making process more transparent. To further address the lack of validation observed in Paper V, Paper VI introduced a LangGraph-based multi-agent setup. In this setup, dedicated agents generate information, verify it against the database, and ensure that responses to the user remain reliable and grounded in the system's actual state. The node- and state-based programming architecture enabled a dynamic, non-linear workflow, allowing the system to iteratively refine queries, re-run searches, or correct inconsistencies, a capability that rigid, rule-based systems from earlier papers could not support. Because the control flow is shaped by the LLM agent's reasoning, the architecture is inherently better at handling uncertainty and adapting to unexpected user inputs. For example, if a user requests a sorting order involving an object that is not present, the LLM verifies this against the database, identifies the mismatch, and returns a clear corrective response.

A similar programming architecture guides the perception module introduced in Paper VI. The modular architecture makes it easier to extend the system to new applications by adding new detection modules without rewriting the entire pipeline. However, perception challenges remain. The static, top-down camera position introduces recurring occlusion issues during robot motion, limiting the system's ability to maintain an accurate real-time representation of the scene. Future improvements may involve repositioning the camera or using additional viewpoints to minimize occlusion and achieve more reliable perception in dynamic or cluttered environments.

4.1.6 Insights on Adaptive Automation Solutions

Reflecting on the approaches used in Paper I and II, it becomes clear that the rule-based automation developed in those early studies could be significantly strengthened by the reasoning-driven principles demonstrated in Paper VI. For example, in Paper I the automation pipeline requires more investment when a new turbine vane model is introduced as rule-based systems struggle when confronted with situations that fall outside their encoded constraints. A reasoning-enabled model could serve as a complementary component that enhances error handling within the automation workflow. For example, when a new turbine vane model is introduced, potentially with different annotation schemes or geometric labelling, the reasoning model could actively search for the required geometric features based on contextual understanding rather than only relying on predefined annotation names. By interpreting the CAD model's structure relative to the task requirements, it could identify missing or mismatched features, communicate these issues to the user in a constructive and transparent manner, or even autonomously adjust annotations to satisfy the input requirements of the automation algorithm. Such capabilities would reduce the need to manually modify the automation architecture for each new variant. The architecture introduced in Paper VI shows how combining structured rules with contextual reasoning enables automation systems to manage variability more effectively and respond to conditions that were not explicitly anticipated in advance.

Across the thesis, a broader insight emerges regarding the foundational elements required for a true adaptive automation in manufacturing. As Isaac Asimov said [91], a robot needs more than sensing, it also requires memory and understanding. The framework developed in the paper VI embodies these three capabilities in an industrial context. The vision system provides perceptual awareness of an uncertain changing environment, the structured database serves as the system's memory, and the integration of LLM-based reasoning introduces an early form of machine "understanding," enabling the system to interpret user intent, contextualize environmental data, and take dynamic decisions. Together, these components represent a shift toward more adaptive automation pipelines and context-aware systems capable of operating effectively in unstructured environments.

4.2 Method Discussion

As described through the industrial use cases in Chapter 3, the close collaboration with industry partners provided not only context but also evolving requirements that consistently guided the research direction. Owing to this dynamic and problem oriented nature of the work, the research approach aligns strongly with the principles of Action Research (AR), where iterative cycles of problem identification, solution development, evaluation, and refinement are central.

The methodological backbone of the thesis is rooted in Design Research Methodology (DRM). Although DRM originates primarily from the domain of design research, its structured approach to problem analysis, solution synthesis, and evaluation proved equally applicable to the broader scope of this thesis. In the early stages focusing heavily on DA and design optimization, the application of DRM was naturally justified. As the research gradually transitioned toward production focused challenges, other methodological frameworks such as industry-as-laboratory [92] or system-oriented problem-solving models [17] could also have been suitable. Yet the interdependence between design and production challenges presented in this thesis made the adoption of a single narrowly defined methodology inappropriate. Instead, DRM provided the required structure, transparency, and validity across diverse phases of the research while still allowing the flexibility needed for iterative, exploratory development.

The AR mindset played a critical operational role in the overarching DRM structure. Each application explored in the thesis involved formulating hypotheses based on observed challenges, implementing solutions, and refining those solutions based on iterative testing and outcomes. The adjustments made throughout the research were often responsive, emerging from practical constraints, system behaviour, validations or insights gained during implementation. This iterative, ad hoc evolution of hypotheses and solutions is characteristic of AR and aligns closely with the practical problem-solving orientation of the work.

4.3 Ethical Reflections

The research presented in this thesis goes beyond purely technical development and inevitably touches on broader ethical questions about how adaptive automation systems should behave, how they should be used, and how they may affect people working alongside them. As robots become more capable of perceiving their environment, reasoning about tasks, and acting with increasing autonomy, the focus can no longer remain solely on performance or efficiency. Instead, it becomes equally important to reflect on safety, responsibility, transparency, and the evolving role of humans within automated systems.

One of the most immediate ethical considerations concerns safety in human–robot interaction. Automation solution developed in Paper IV-VI rely on external controllers and simulation-based DTs which bypasses some of the safety mechanisms embedded in industrial robot controllers, such as certified collision detection or motion constraints. From an ethical standpoint, this raises questions regarding acceptable risk, particularly in collaborative or semi-collaborative settings. While the research is conducted in controlled experimental environments, it highlights the need for future solutions to embed safety assurance as a first-class design constraint rather than a post hoc addition.

Another important ethical dimension relates to transparency and interpretability of decision-making. The introduction of LLMs and learning-based components into robotic control pipelines shifts automation from fully deterministic behaviour toward probabilistic and data-driven reasoning. This shift can obscure why a system takes a particular action, making it more difficult for human operators to anticipate, understand, or challenge system behaviour. The thesis partially addresses this concern through modular and agent-based architectures, which improve traceability by separating perception, reasoning, and execution. However, the ethical implication remains that adaptive automation systems must be designed in a way that preserves human oversight and allows operators to question system decisions. Without such transparency, trust in Human–Robot Collaboration (HRC) may decrease in safety-critical or high-responsibility contexts.

A further ethical consideration concerns the potential impact of adaptive automation on employment and the nature of human work. While automation is often associated with job displacement, the research presented in this thesis is not aimed at eliminating human roles but at addressing situations where traditional automation fails due to high variability, incomplete information, or frequent human intervention. In such contexts, adaptive automation is positioned as a means to support and augment human workers rather than replace them, by reducing repetitive, error-prone, or cognitively demanding tasks and enabling humans to focus on supervision, decision-making, and exception handling. From this perspective, the ethical responsibility lies not in maximizing autonomy, but in designing systems that preserve meaningful human involvement and facilitate skill transformation rather than workforce substitution. The thesis aligns with this perspective by emphasizing human-in-the-loop supervision and human-centric interaction, rather than fully autonomous operation without oversight.

4.4 Research Contributions

Drawing on the analyses presented in Papers I, III, and IV, the research provides a systematic synthesis of an otherwise fragmented body of work on adaptive automation. Rather than offering a conventional literature review, the thesis formalizes a central limitation of prevailing DA approaches: while rule-based and parameter-driven systems are effective in well-structured settings, they become brittle when design assumptions, product configurations, or operating conditions change. By showing how rigid knowledge representations in upstream design activities constrain flexibility in downstream production systems, the work reframes design and production as a coupled, adaptive loop rather than as isolated stages in an automation pipeline.

Building on this conceptual grounding, the thesis introduces a methodological shift through the integration of KBE, DT, and LLM as presented in Paper VI. This hybrid approach moves beyond deterministic programming by combining the explicit structure of KBE with the contextual reasoning capabilities of LLMs and the dynamic synchronization offered by DTs. In doing so, the research demonstrates how complex industrial tasks can be formalized and executed under real-world constraints, even when those tasks involve uncertainty, variability, or incomplete specifications. This methodological contribution establishes a foundation for learning-enabled design and execution, enabling automation solutions to handle situations that are difficult or infeasible to capture using traditional rule-based logic alone.

These ideas are operationalized through a series of proof-of-concept implementations that demonstrate adaptive robotic behaviour within simulation-driven environments. The thesis shows how real-time camera-based perception, structured data management, and LLM-driven reasoning can be orchestrated into a unified system capable of interpreting evolving requirements and making situated decisions. While robotics serves as the primary application domain, the contribution extends beyond robot control by illustrating how adaptive reasoning can compensate for rigid assumptions embedded earlier in the design process. In this sense, the robotic case studies function as a concrete test-bed for evaluating how design decisions propagate through, and can be mitigated within, adaptive production systems.

In parallel, the thesis makes an open-science contribution by releasing the developed tools, workflows, and architectures as open-source resources. This includes simulation scenes, agent definitions, perception modules, reasoning pipelines, and DT orchestration frameworks. By doing so, the work addresses reproducibility challenges and provides a shared experimental baseline that other researchers can build upon, extend, or critically evaluate. These contributions offer a forward-looking perspective on the role of reasoning-capable systems in industrial automation. By examining both the opportunities and the limitations of integrating language-based intelligence into engineering workflows, the thesis expands the theoretical horizon of adaptive automation research.

4.5 Future Work

The research presented across papers collectively highlight that the integration of physics-based simulation, vision-based perception, structured data management, and LLM-driven reasoning can provide a promising foundation for achieving real-time adaptability, context awareness, and autonomous decision-making. Building on these findings, several avenues for future research emerge.

A primary direction for future work concerns enhancing the reliability and accuracy of LLM-based agents. Although early results show clear promise in using agentic workflows for interpreting user commands and reasoning over structured data, their performance remains constrained by prompt dependency and domain-specific sensitivity. Future work will therefore investigate systematic prompt modelling, domain-adaptive fine-tuning, and evaluation pipelines to minimize hallucination, improve interpretability, and ensure stable agent behaviour during different manufacturing operations. Strengthening these reasoning components is essential for developing trustworthy HRC interfaces that maintain consistency across tasks and applications.

In parallel, advancing the perception capabilities of the system represents another key direction. While the current approach relies on contour-based object detection with a static camera setup, this method remains sensitive to occlusions, lighting disturbances, and workspace constraints. Future developments will therefore explore more robust vision strategies, particularly 6D object detection and pose estimation, enabling richer geometric understanding and more reliable real-world perception. Transitioning to such methods is expected to significantly strengthen the fidelity of the DT and improve the consistency of perception–action loops in dynamic environments.

An equally important direction for future work is to extend the developed reasoning-, perception-, and memory-based methodology to DA and optimization tasks. Applying this approach to design-centric problems similar to those addressed in Papers I and II would enable a direct evaluation of its effectiveness in domains that have traditionally relied on rigid, rule-based workflows. In particular, the recurring limitations observed in earlier studies provide a clear baseline against which the proposed methodology can be assessed. Recasting these DA pipelines within a reasoning-enabled framework would allow the system to interpret design intent, adapt to geometric variation, and manage incomplete or inconsistent input information more robustly. Such investigations would not only validate the transferability of the proposed approach beyond production automation but also clarify its potential to reduce modelling effort, improve scalability, and increase robustness in early-stage engineering design processes.

Expanding the system to address a broader set of industrial applications will be a critical step. By deploying the developed framework on new case studies, involving different product characteristics, task constraints, and environmental uncertainties will help further validate the modularity and transferability of the solution. Such extensions will naturally enrich the shared database with new robot operations, perception templates, and semantic structures, thereby increasing its generality and reusability.

Ultimately, the long-term trajectory of this research is directed toward developing a self-adapting system capable of perceiving real-world conditions, understanding the context of those conditions, reasoning over multimodal data, and taking safe and effective actions with minimal human intervention. Achieving this vision requires the seamless integration of reliable perception, robust reasoning capability, structured data handling, and safe control policies. Such a system should enable humans to transition from operators to supervisors, overseeing processes while the system dynamically manages variability, learns from new conditions, and ensures safe operation within unstructured industrial environments.

CHAPTER 5

Conclusion

This final chapter addresses the research question mentioned in introduction by synthesizing the findings and limitations identified throughout the thesis. By integrating design and production perspectives, it reflects on the challenges of implementing adaptive automation in unstructured environments and the strategies developed to mitigate them.

RQ. What are the limitations associated with implementing adaptive automation techniques in unstructured environments, and how can these challenges be mitigated?

In design automation, adaptability is primarily driven by rising product customization, where changes in geometry, parameters, or annotations frequently invalidate rigid rule-based workflows. In production, adaptability is challenged not only by upstream design variability but also by dynamic shop-floor conditions, human interaction, and shifting operational objectives. Across both domains, the work demonstrates that legacy rule-based automation struggles when faced with variability, as it relies on fixed assumptions that are difficult to maintain in practice.

Achieving adaptability in both design and production depends on three fundamental capabilities: perception, memory, and understanding. Perception enables the system to observe its environment, whether by interpreting geometric features in design models or by sensing objects and states in physical production settings. Memory refers to the structured storage of relevant information, such as design parameters, task constraints, environmental states, and user inputs, in a form that supports reuse and validation.

The most critical limitation identified across the papers is the lack of understanding/context-awareness. In conventional automation, understanding is approximated through predefined rules, conditions, and sequences. Although this approach yields reliable performance in well-defined scenarios, it is not flexible when inputs deviate from what was anticipated during system design. This limitation becomes evident in both design automation, where small changes in geometry or annotations can break automated pipelines, and in production automation, where unstructured environments demand contextual interpretation and flexible decision-making.

The mitigation explored in this thesis lies in designing perception and memory with reasoning-enabled mechanisms, particularly through the use of digital twins, structured data representations, and language-based reasoning models. These elements allow automation systems to interpret context, validate information, and adapt behaviour beyond static rule sets. At the same time, the research highlights that such approaches introduce new challenges related to reliability, grounding, and system integration. As a result, adaptive automation should not be viewed as a fully solved problem, but as an evolving capability that requires careful architectural design and continued research.

A central contribution of this thesis is the demonstration that adaptive automation does not arise from any single technology, but from the deliberate way in which heterogeneous technologies are combined and structured. The papers show how state-of-the-art methods drawn from different domains can be integrated alongside legacy engineering tools and automation setups to extend their adaptive capabilities rather than replace them. The thesis shows that such integration must be architected as a closed-loop system in which exchange of information is seamless and continuous, with the specific realization shaped by application requirements.

Bibliography

[1] S Jack Hu. “Evolving paradigms of manufacturing: From mass production to mass customization and personalization”. In: *Procedia Cirp* 7 (2013), pp. 3–8. DOI: 10.1016/j.procir.2013.05.002.

[2] Amr Adel. “Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas”. In: *Journal of Cloud Computing* 11.1 (2022), p. 40. DOI: 10.1186/s13677-022-00314-5.

[3] Luis M Camarinha-Matos, Andre Dionisio Rocha, and Paula Graça. “Collaborative approaches in sustainable and resilient manufacturing”. In: *Journal of Intelligent Manufacturing* 35.2 (2024), pp. 499–519. DOI: 10.1007/s10845-022-02060-6.

[4] Flavio S Fogliatto, Giovani JC Da Silveira, and Denis Borenstein. “The mass customization decade: An updated review of the literature”. In: *International Journal of production economics* 138.1 (2012), pp. 14–25. DOI: 10.1016/j.ijpe.2012.03.002.

[5] Xingyu Li, Ragu Athinarayanan, Baicun Wang, Wei Yuan, Quan Zhou, Martin Jun, Jose Bravo, Robert X Gao, Lihui Wang, and Yoram Koren. “Smart reconfigurable manufacturing: Literature analysis”. In: *Procedia CIRP* 121 (2024), pp. 43–48. DOI: 10.1016/j.procir.2023.09.228.

[6] Steven M Miller. “Impacts of robotics and flexible manufacturing technologies on manufacturing costs and employment”. In: *The management of Productivity and Technology in manufacturing*. Springer, 1985, pp. 73–110. DOI: 10.1007/978-1-4613-2507-9_4.

[7] J Norberto Pires, Altino Loureiro, and Gunnar Bölmsjo. *Welding robots: technology, system issues and applications*. Springer, 2006. DOI: 10.1007/1-84628-191-1_1.

[8] Sebastian Saniuk, Sandra Grabowska, and Bożena Gajdzik. “Personalization of products in the industry 4.0 concept and its impact on achieving a higher level of sustainable consumption”. In: *Energies* 13.22 (2020), p. 5895. DOI: 10.3390/en13225895.

[9] Violetta Giada Cannas, Antonio Masi, Margherita Pero, and Thomas Ditlev Brunø. “Implementing configurators to enable mass customization in the Engineer-to-Order industry: a multiple case study research”. In: *Production planning & control* 33.9-10 (2022), pp. 974–994. DOI: 10.1080/09537287.2020.1837941.

[10] Stéphanie Bouchard, Sébastien Gamache, and Georges Abdulnour. "Operationalizing Mass Customization in Manufacturing SMEs—A Systematic Literature Review". In: *Sustainability* 15.4 (2023). ISSN: 2071-1050. DOI: 10.3390/su15043028. URL: https://www.mdpi.com/2071-1050/15/4/3028.

[11] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. "Industry 4.0 and Industry 5.0—Inception, conception and perception". In: *Journal of manufacturing systems* 61 (2021), pp. 530–535. DOI: 10.1016/j.procs.2025.01.164.

[12] Michael Lieder and Amir Rashid. "Towards circular economy implementation: a comprehensive review in context of manufacturing industry". In: *Journal of cleaner production* 115 (2016), pp. 36–51. DOI: 10.1016/j.jclepro.2015.12.042.

[13] Anastasiia Rozhok, Rosa Abate, Elena Manoli, and Luigi Nele. "A Review of Recent Advanced Applications in Smart Manufacturing Systems". In: *Journal of Manufacturing and Materials Processing* 10.1 (2025), p. 1. DOI: 10.3390/jmmp10010001.

[14] Dimitris Mourtzis and Michael Doukas. "Design and planning of manufacturing networks for mass customisation and personalisation: challenges and outlook". In: *Procedia Cirp* 19.C (2014), pp. 1–13. DOI: 10.1016/j.procir.2014.05.004.

[15] Mikael Cederfeldt and Fredrik Elgh. "Design automation in SMEs–current state, potential, need and requirements". In: *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005*. 2005. URL: https://www.designsociety.org/publication/23007/design_automation_in_smes_%E2%80%93_current_state_potential_need_and_requirements.

[16] Paul Satchell. *Innovation and automation*. Routledge, 2018. DOI: 10.4324/9780429449888.

[17] Monica Bellgran and Kristina Säfsten. *Production development: design and operation of production systems*. Springer, 2010. DOI: 10.1007/978-1-84882-495-9.

[18] Karl Hribernik, Giacomo Cabri, Federica Mandreoli, and Gregoris Mentzas. "Autonomous, context-aware, adaptive Digital Twins—State of the art and roadmap". In: *Computers in Industry* 133 (2021), p. 103508. DOI: 10.1016/j.compind.2021.103508.

[19] Agnieszka Radziwon, Arne Bilberg, Marcel Bogers, and Erik Skov Madsen. "The smart factory: exploring adaptive and flexible manufacturing solutions". In: *Procedia engineering* 69 (2014), pp. 1184–1190. DOI: 10.1016/j.proeng.2014.03.108.

[20] Robert Plant and Rose Gamble. "Methodologies for the development of knowledge-based systems, 1982–2002". In: *The Knowledge Engineering Review* 18.1 (2003), pp. 47–81. DOI: 10.1017/S026988890300064X.

[21] Karen M Feigh, Michael C Dorneich, and Caroline C Hayes. "Toward a characterization of adaptive systems: A framework for researchers and system designers". In: *Human factors* 54.6 (2012), pp. 1008–1024. DOI: 10.1177/0018720812443983.

[22] Reinhard Oppermann. "Adaptively supported adaptability". In: *International Journal of Human-Computer Studies* 40.3 (1994), pp. 455–472. DOI: 10.1006/ijhc.1994.1021.

[23] Marco Bortolini, Maurizio Faccio, Francesco Gabriele Galizia, Mauro Gamberi, and Francesco Pilati. "Adaptive automation assembly systems in the industry 4.0 era: A reference framework and full–scale prototype". In: *Applied Sciences* 11.3 (2021), p. 1256. DOI: 10.3390/app11031256.

[24] Margherita Bernabei and Francesco Costantino. "Adaptive automation: Status of research and future challenges". In: *Robotics and Computer-Integrated Manufacturing* 88 (2024), p. 102724. DOI: 10.1016/j.rcim.2024.102724.

[25] Lucia Vigoroso, Federica Caffaro, Massimo Tronci, and Mario Fargnoli. "Adaptive and Adaptable Automation in manufacturing: a bibliometric review". In: *Procedia Computer Science* 253 (2025), pp. 1024–1033. DOI: 10.1016/j.procs.2025.01.164.

[26] Christopher A Miller. "Trust in adaptive automation: the role of etiquette in tuning trust via analogic and affective methods". In: *Proceedings of the 1st international conference on augmented cognition*. 2005, pp. 22–27. URL: https://api.semanticscholar.org/CorpusID:17289588.

[27] Raja Parasuraman and Christopher D Wickens. "Humans: Still vital after all these years of automation". In: *Human factors* 50.3 (2008), pp. 511–520. DOI: 10.1518/001872008X312198.

[28] Marco Bortolini, Maurizio Faccio, Francesco Gabriele Galizia, Mauro Gamberi, and Francesco Pilati. "Design, engineering and testing of an innovative adaptive automation assembly system". In: *Assembly Automation* 40.3 (2020), pp. 531–540. DOI: 10.1108/AA-06-2019-0103.

[29] Gloria Calhoun. "Adaptable (not adaptive) automation: Forefront of human–automation teaming". In: *Human factors* 64.2 (2022), pp. 269–277. DOI: 10.1177/00187208211037457.

[30] Patricia Kuegler, Fabian Dworschak, Benjamin Schleich, and Sandro Wartzack. "The evolution of knowledge-based engineering from a design research perspective: Literature review 2012–2021". In: *Advanced Engineering Informatics* 55 (2023), p. 101892. DOI: 10.1016/j.aei.2023.101892.

[31] Wim JC Verhagen, Pablo Bermell-Garcia, Reinier EC Van Dijk, and Richard Curran. "A critical review of Knowledge-Based Engineering: An identification of research challenges". In: *Advanced Engineering Informatics* 26.1 (2012), pp. 5–15. DOI: 10.1016/j.aei.2011.06.004.

[32] Hossein Nassaji. *Qualitative and descriptive research: Data type versus data analysis*. 2015. DOI: 10.1177/1362168815572747.

[33] Claudia M Eckert, MK Stacey, and PJ Clarkson. "The spiral of applied research: A methodological view on integrated design research". In: (2003). URL: https://www.designsociety.org/publication/23944/the_spiral_of_applied_research_a_methodological_view_on_integrated_design_research.

[34] Maura Borrego, Elliot P Douglas, and Catherine T Amelink. "Quantitative, qualitative, and mixed research methods in engineering education". In: *Journal of Engineering education* 98.1 (2009), pp. 53–66. DOI: 10.1002/j.2168-9830.2009.tb01005.x.

[35] Jan Vom Brocke, Alan Hevner, and Alexander Maedche. "Introduction to design science research". In: *Design science research. Cases*. Springer, 2020, pp. 1–13. DOI: 10.1007/978-3-030-46781-4_1.

[36] SJ Culley. "Future Issues in Design Research (FIDR) Report". In: *EPRSC/DTI, Bath* (1999).

[37] Lucienne TM Blessing and Amaresh Chakrabarti. *DRM, a design research methodology*. Springer, 2009. DOI: 10.1007/978-1-84882-587-1.

[38] Ernest T Stringer and Alfredo Ortiz Aragón. *Action research*. Sage publications, 2020.

[39] Cheryl Baltes. "Patrick J. Hanratty: Pioneer in commercial CAD/CAM software systems". In: *IEEE Annals of the History of Computing* 46.4 (2024), pp. 50–57. DOI: 10.1109/MAHC.2024.3473389.

[40] Ivan E Sutherland. "Sketch pad a man-machine graphical communication system". In: *Proceedings of the SHARE design automation workshop*. 1964, pp. 6–329. DOI: 10.1145/800265.810742.

[41] Leon Peter Poot, Mehdi Tarkian, and Johan Ölvander. "Design and Production Automation for Mass Customisation–An Initial Framework Proposal Evaluated in Engineering Education and SME Contexts". In: *Transdisciplinary Engineering for Resilience: Responding to System Disruptions*. IOS Press, 2021, pp. 71–80. DOI: 10.3233/ATDE210084.

[42] Lars Hvam, Niels Henrik Mortensen, and Jesper Riis. *Product customization*. Springer, 2008. DOI: 10.1007/978-3-540-71449-1.

[43] Fredrik Elgh. "Decision support in the quotation process of engineered-to-order products". In: *Advanced Engineering Informatics* 26.1 (2012), pp. 66–79. DOI: 10.1016/j.aei.2011.07.001.

[44] Craig B Chapman and Martyn Pinfold. "The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure". In: *Advances in engineering software* 32.12 (2001), pp. 903–912. DOI: 10.1016/S0965-9978(01)00041-2.

[45] Kristian Amadori, Mehdi Tarkian, Johan Ölvander, and Petter Krus. "Flexible and robust CAD models for design automation". In: *Advanced Engineering Informatics* 26.2 (2012), pp. 180–195. DOI: 10.1016/j.aei.2012.01.004.

[46] Gianfranco La Rocca. "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design". In: *Advanced engineering informatics* 26.2 (2012), pp. 159–179. DOI: 10.1016/j.aei.2012.02.002.

[47] Adrian A. Hopgood. *Intelligent Systems for Engineers and Scientists*. CRC Press LLC, 2001. DOI: 10.1201/9781003226277.

[48] Mehdi Tarkian. *Design automation for multidisciplinary optimization: A high level CAD template approach*. Linkopings Universitet (Sweden), 2012.

[49] Pablo Bermell-García and Ip-Shing Fan. "A KBE System for the design of wind tunnel models using reusable knowledge components". In: *International Congress on Project Engineering, Barcelona*. Vol. 23. 2002, p. 24. URL: http://hdl.handle.net/10234/81867.

[50] DE Calkins, Nathaniel Egging, and Christian Scholz. "Knowledge-based engineering (KBE) design methodology at the undergraduate and graduate levels". In: *development* 21 (1999). URL: https://api.semanticscholar.org/CorpusID:18409624.

[51] Avinash C Kak, Kim L Boyer, Chien-Hue Chen, Robert J Safranek, and Hyun S Yang. "A Knowledge-Based Robotic Assembly Cell." In: *IEEE Expert* 1.1 (1986), pp. 63–83. DOI: 10.1109/MEX.1986.5006501.

[52] Mahmood Reza Khabbazi, Fredrik Danielsson, Bassam Massouh, and Bengt Lennartson. "Plug and Produce—a review and future trend". In: *The International Journal of Advanced Manufacturing Technology* 134.9 (2024), pp. 3991–4014. DOI: 10.1007/s00170-024-14379-w.

[53] Julius Pfrommer, Denis Stogl, Kiril Aleksandrov, Stefan Escaida Navarro, Björn Hein, and Jürgen Beyerer. "Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems". In: *at-Automatisierungstechnik* 63.10 (2015), pp. 790–800. DOI: 10.1515/auto-2014-1157.

[54] Christian Eymüller, Julian Hanke, Alexander Poeppel, and Wolfgang Reif. "Towards self-configuring plug & produce robot systems based on ontologies". In: *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE. 2023, pp. 23–27. DOI: 10.1109/ICARA56516.2023.10126075.

[55] Torben Wiese, Johannes Abicht, Christian Friedrich, Arvid Hellmich, and Steffen Ihlenfeldt. "Flexible skill-based control for robot cells in manufacturing". In: *Frontiers in Robotics and AI* 9 (2022), p. 1014476. DOI: 10.3389/frobt.2022.1014476.

[56] Julius Pfrommer, Denis Stogl, Kiril Aleksandrov, Viktor Schubert, and Björn Hein. "Modelling and orchestration of service-based manufacturing systems via skills". In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE. 2014, pp. 1–4. DOI: 10.1109/ETFA.2014.7005285.

[57] Aljosha Köcher, Luis Miguel Vieira da Silva, and Alexander Fay. "Automated Process Planning Based on a Semantic Capability Model and SMT". In: *arXiv preprint arXiv:2312.08801* (2023). DOI: 10.48550/arXiv.2312.08801.

[58] Eeva Järvenpää, Niko Siltala, Otto Hylli, and Minna Lanz. "Product Model ontology and its use in capability-based matchmaking". In: *Procedia CIRP* 72 (2018), pp. 1094–1099. DOI: 10.1016/j.procir.2018.03.211.

[59] Lucas Sakurada, Paulo Leitao, and Fernando De la Prieta. "Towards the digitization using asset administration shells". In: *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2021, pp. 1–6. DOI: 10.1109/IECON48115.2021.9589370.

[60] Tasnim A Abdel-Aty, Elisa Negri, and Simone Galparoli. "Asset administration shell in manufacturing: Applications and relationship with digital twin". In: *IFAC-PapersOnLine* 55.10 (2022), pp. 2533–2538. DOI: 10.1016/j.ifacol.2022.10.090.

[61] Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[62] Terrin Pulikottil, Luis Alberto Estrada-Jimenez, Hamood Ur Rehman, Jose Barata, Sanaz Nikghadam-Hojjati, and Leszek Zarzycki. "Multi-agent based manufacturing: current trends and challenges". In: *2021 26th IEEE international conference on emerging technologies and factory automation (ETFA)*. IEEE. 2021, pp. 1–7. DOI: 10.1109/ETFA45728.2021.9613555.

[63] Michael Wooldridge and Nicholas R Jennings. "Intelligent agents: Theory and practice". In: *The knowledge engineering review* 10.2 (1995), pp. 115–152. DOI: 10.1017/S0269888900008122.

[64] Jean-Pierre Müller and H Van Dyke Parunak. "Multi-agent systems and manufacturing". In: *IFAC Proceedings Volumes* 31.15 (1998), pp. 545–550. DOI: 10.1016/S1474-6670(17)40609-4.

[65] Mathijs De Weerdt and Brad Clement. *Introduction to planning in multiagent systems*. 2009. DOI: 10.3233/MGS-2009-0133.

[66] Anand S Rao, Michael P Georgeff, et al. "BDI agents: From theory to practice." In: *Icmas*. Vol. 95. 1995, pp. 312–319.

[67] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. "The belief-desire-intention model of agency". In: *International workshop on agent theories, architectures, and languages*. Springer. 1998, pp. 1–10. DOI: 10.1007/3-540-49057-4_1.

[68] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. "Large language model based multi-agents: A survey of progress and challenges". In: *arXiv preprint arXiv:2402.01680* (2024). DOI: 10.48550/arXiv.2402.01680.

[69] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. "The rise and potential of large language model based agents: A survey". In: *Science China Information Sciences* 68.2 (2025), p. 121101. DOI: 10.1007/s11432-024-4222-0.

[70] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[71] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. "MetaGPT: Meta programming for a multi-agent collaborative framework". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=VtmBAGCN7o.

[72] Edward Y Chang and Longling Geng. "SagaLLM: Context Management, Validation, and Transaction Guarantees for Multi-Agent LLM Planning". In: *arXiv preprint arXiv:2503.11951* (2025). DOI: 10.48550/arXiv.2503.11951.

[73] Andreas Schmidt Jensen, Virginia Dignum, and Jørgen Villadsen. "A framework for organization-aware agents". In: *Autonomous Agents and Multi-Agent Systems* 31.3 (2017), pp. 387–422. DOI: 10.1007/s10458-015-9324-2.

[74] Dongge Han, Trevor McInroe, Adam Jelley, Stefano V Albrecht, Peter Bell, and Amos Storkey. "Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots". In: *Proceedings of the 31st International Conference on Computational Linguistics*. 2025, pp. 1465–1474. URL: https://aclanthology.org/2025.coling-main.98/.

[75] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. "LLM multi-agent systems: Challenges and open problems". In: *arXiv preprint arXiv:2402.03578* (2024). DOI: 10.48550/arXiv.2402.03578.

[76] László Monostori, Botond Kádár, Thomas Bauernhansl, Shinsuke Kondoh, Soundar Kumara, Gunther Reinhart, Olaf Sauer, Gunther Schuh, Wilfried Sihn, and Kenichi Ueda. "Cyber-physical systems in manufacturing". In: *Cirp Annals* 65.2 (2016), pp. 621–641. DOI: 10.1016/j.cirp.2016.06.005.

[77] B Danette Allen. "Digital twins and living models at NASA". In: *Digital Twin Summit*. 2021. URL: https://ntrs.nasa.gov/citations/20210023699.

[78] Maulshree Singh, Evert Fuenmayor, Eoin P Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. "Digital twin: Origin to future". In: *Applied System Innovation* 4.2 (2021), p. 36. DOI: 10.3390/asi4020036.

[79] Michael Grieves. "Digital twin: manufacturing excellence through virtual factory replication". In: *White paper* 1.2014 (2014), pp. 1–7.

[80] Fei Tao and Meng Zhang. "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing". In: *IEEE access* 5 (2017), pp. 20418–20427. DOI: 10.1109/ACCESS.2017.2756069.

[81] Roland Rosen, Georg Von Wichert, George Lo, and Kurt D Bettenhausen. "About the importance of autonomy and digital twins for the future of manufacturing". In: *Ifac-papersonline* 48.3 (2015), pp. 567–572. DOI: 10.1016/j.ifacol.2015.06.141.

[82] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. "Digital Twin in manufacturing: A categorical literature review and classification". In: *Ifac-PapersOnline* 51.11 (2018), pp. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474.

[83] Valentina Villa and Bernardino Chiaia. "Digital twin for smart school buildings: state of the art, challenges, and opportunities". In: *Handbook of Research on Developing Smart Cities Based on Digital Twins* (2021), pp. 320–340. DOI: 10.4018/978-1-7998-7091-3.ch015.

[84] Fei Tao, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, and Ping Ji. "Digital twin modeling". In: *Journal of Manufacturing Systems* 64 (2022), pp. 372–389. DOI: 10.1016/j.jmsy.2022.06.015.

[85] Yuzhu Sun, Mien Van, Stephen McIlvanna, Nguyen Minh Nhat, Kabirat Olayemi, Jack Close, and Seán McLoone. "Digital Twin-Driven Reinforcement Learning for Obstacle Avoidance in Robot Manipulators: A Self-Improving Online Training Framework". In: *arXiv preprint arXiv:2403.13090* (2024). DOI: 10.48550/arXiv.2403.13090.

[86] Rotimi Ogunsakin, Nikolay Mehandjiev, and Cesar A Marin. "Towards adaptive digital twins architecture". In: *Computers in Industry* 149 (2023), p. 103920. DOI: 10.1016/j.compind.2023.103920.

[87] Paolo Bellavista, Nicola Bicocchi, Mattia Fogli, Carlo Giannelli, Marco Mamei, and Marco Picone. “Requirements and design patterns for adaptive, autonomous, and context-aware digital twins in industry 4.0 digital factories”. In: *Computers in Industry* 149 (2023), p. 103918. DOI: 10.1016/j.compind.2023.103918.

[88] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. “Smart factory-a step towards the next generation of manufacturing”. In: *Manufacturing Systems and Technologies for the New Frontier: The 41 st CIRP Conference on Manufacturing Systems May 26–28, 2008, Tokyo, Japan*. Springer. 2008, pp. 115–118. DOI: 10.1007/978-1-84800-267-8_23.

[89] Xin Ma, Qinglin Qi, and Fei Tao. “A Digital Twin–Based Environment-Adaptive Assignment Method for Human–Robot Collaboration”. In: *Journal of Manufacturing Science and Engineering* 146.3 (2024). DOI: 10.1115/1.4064040.

[90] Igiri Onaji, Divya Tiwari, Payam Soulatiantork, Boyang Song, and Ashutosh Tiwari. “Digital twin in manufacturing: conceptual framework and case studies”. In: *International journal of computer integrated manufacturing* 35.8 (2022), pp. 831–858. DOI: 10.1080/0951192X.2022.2027014.

[91] Isaac Asimov and Jesús Fernández Zulica. *Opus 100*. Houghton Mifflin Boston, 1969.

[92] Colin Potts. “Software-engineering research revisited”. In: *IEEE software* 10.5 (2002), pp. 19–28. DOI: 10.1109/52.232392.

[93] Sanjay Nambiar, Anan Ashrabi Ananno, Herman Titus, Anton Wiberg, and Mehdi Tarkian. “Multidisciplinary Automation in Design of Turbine Vane Cooling Channels”. In: *International Journal of Turbomachinery, Propulsion and Power* 9.1 (2024), p. 7. DOI: 10.3390/ijtpp9010007.

[94] Sanjay Nambiar, Albin Parappilly Albert, Veeravenkatamanikanta Virupaksh Raja Chowdary Rimmalapudi, Vinayak Acharya, Mehdi Tarkian, and Henrik Kihlman. “Autofix–automated design of fixtures”. In: *Proceedings of the Design Society* 2 (2022), pp. 543–552. DOI: 10.1017/pds.2022.56.

[95] Sanjay Nambiar, Anton Wiberg, and Mehdi Tarkian. “Automation of unstructured production environment by applying reinforcement learning”. In: *Frontiers in Manufacturing Technology* 3 (2023), p. 1154263. DOI: 10.3389/fmtec.2023.1154263.

[96] Sanjay Nambiar, Marie Jonsson, and Mehdi Tarkian. “Automation in unstructured production environments using isaac sim: A flexible framework for dynamic robot adaptability”. In: *Procedia CIRP* 130 (2024), pp. 837–846. DOI: 10.1016/j.procir.2024.10.173.

[97] Sanjay Nambiar, Rahul Chiramel Paul, Oscar Chigozie Ikechukwu, Marie Jonsson, and Mehdi Tarkian. “Digital Twin-Enabled Adaptive Robotics: Leveraging Large Language Models in Isaac Sim for Unstructured Environments”. In: *Machines* 13.7 (2025), p. 620. DOI: 10.3390/machines13070620.

[98] Sanjay Nambiar, Oscar Chigozie Ikechukwu, Rahul Chiramel Paul, Marie Jonsson, and Mehdi Tarkian. “Digital Twin-Enabled Adaptive Robotics: Multi-Agent Reasoning over Language, Vision, and Structured Database”. In: *Robotics and Computer Integrated Manufacturing (Submitted)* (2025).

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

https://doi.org/10.3384/9789181184624

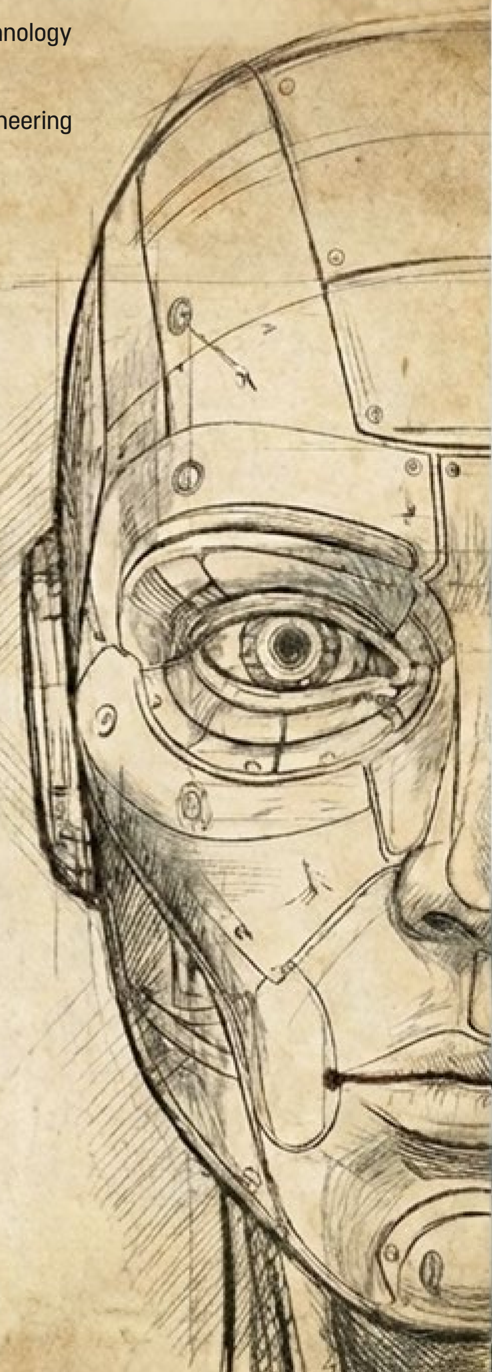
FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology
Dissertation No. 2505, 2026

Department of Management and Engineering
Division of Product Realisation

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se



LINKÖPING
UNIVERSITY