MEMOIRE DE DOCTORAT DE

L'ECOLE CENTRALE DE NANTES

ECOLE DOCTORALE N° 602
Sciences de l'Ingénierie et des Systèmes
Spécialité : Génie Industriel

Par

Oswaldo Armando PÉREZ LARA

Integration of drones to support the agility of production systems in the context of industry 4.0

Projet de recherche doctoral présenté et soutenu à l'École Centrale de Nantes, le 8 décembre 2025
Unité de recherche : UMR 6004, Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Sofiene Dellagi	Professeur des universités, Université de Lorraine
Damien Trentesaux	Professeur des universités, Université Polytechnique Hauts-de-France

Composition du Jury :

Présidente :	Maria Di Mascolo	Directrice de recherche, CNRS – École Nationale Supérieure de Génie Industriel – Grenoble INP
Examinateurs :	Sofiene Dellagi	Professeur des universités, Université de Lorraine
	Damien Trentesaux	Professeur des universités, Université Polytechnique Hauts-de-France

Directeur de recherches doctorales :	Jean Jacques Loiseau	Directeur de recherche, CNRS – École Centrale de Nantes
Co-dir. de recherches doctorales :	Abdelhamid Chriette	Maître de Conférences, École Centrale de Nantes
Co-enc. de recherches doctorales :	Maroua Nouiri	Maître de Conférences, Nantes Université

Contents

List of Figures . . . 7
List of Tables . . . 8

Introduction **9**
Problem Statement . . . 10
Research Questions . . . 10
Objectives and limitations . . . 11
Contributions . . . 12
Thesis structure . . . 13
Conferences and Scientific publications . . . 15

1 State of the art **17**
1.1 Industry 4.0 . . . 18
1.1.1 Flexibility in the Production Systems . . . 20
1.2 Intra-logistics 4.0 . . . 20
1.3 Mass-customized manufacturing systems . . . 22
1.3.1 Intra-logistics in Mass Customization systems . . . 23
1.3.2 Intralogistics trends . . . 24
1.4 Drones in the Industry 4.0 . . . 26
1.4.1 Types of drones . . . 26
1.4.2 Main applications of Drones within the Industry . . . 27
Construction . . . 29
Agriculture . . . 29
Maritime . . . 30
1.5 Challenges for integrating drones into the Industry 4.0 . . . 31
1.5.1 Strategic level . . . 32
1.5.2 Operational level . . . 34
Path planning . . . 36
Scheduling . . . 37
Navigation . . . 38
Routing . . . 41

1.5.3 Organizational level . . . 45
1.6 Synthesis and research gap . . . 46

2 Problem formalization within an agile manufacturing system 49
2.1 Scenario conceptualization . . . 50
2.1.1 From conceptual model to mathematical formalization . . . 52
2.2 Problem definition: Drone Delivery Routing Problem . . . 53
2.2.1 Hypothesis and constraints . . . 55
2.2.2 Model validation and numerical experiments . . . 59
2.2.3 Energy consumption . . . 60
2.3 Conclusion . . . 61

3 Pre-adoption: Economic and strategic factors 63
3.1 Human Drone Interaction . . . 64
3.1.1 Evaluation of the Human-Drone ergonomics . . . 64
3.1.2 Quadrotor-human safety distance . . . 65
3.2 Discrete Event System Simulation . . . 67
3.2.1 Envisioned scenario . . . 70
3.2.2 Scenario definition . . . 71
3.2.3 FlexSim . . . 73
3.2.4 System overview . . . 73
3.2.5 Batch Production System . . . 76
3.2.6 Drone Delivery System . . . 76
3.3 Economic benefits . . . 81
3.3.1 Calibration of the economic model . . . 83
3.3.2 Operational modes and profit functions . . . 85
3.3.3 Model validation . . . 87
3.3.4 Dynamic Programming formulation . . . 89
3.3.5 Numerical results . . . 90
3.4 Conclusion . . . 91

4 Early Adoption: operational and supply chain factors 95
4.1 Conflict-Management Algorithm: Integration with a supervisor . . . 96
4.1.1 Types of conflicts and how to manage them . . . 96
4.1.2 Integration of the conflict-management algorithm . . . 97
4.1.3 Related work on conflict management for drone scheduling . . . 98
4.2 System Architecture . . . 101
4.2.1 Drone platform . . . 101
4.3 openTCS: Drone Fleet Manager . . . 103

4.3.1 Plant model elements . . . 104
4.3.2 Plant operation elements . . . 105
4.3.3 openTCS integration . . . 106
4.4 Conclusion . . . 111

5 Adoption: Organizational and behavioral factors 113
5.1 Risk Management . . . 114
5.1.1 Ground impact . . . 116
5.1.2 Mid-air collision . . . 117
5.1.3 System reliability: Risk management framework . . . 118
Mission definition . . . 119
Risk evaluation . . . 122
Risk reduction . . . 125
Risk acceptance . . . 128
5.2 Working Situation Model . . . 129
5.2.1 Defining safety protocols . . . 130
5.3 Conclusion . . . 132

General Conclusion 135

Appendix 139

A Working Situation Model 139

B /vehicle 140

C build.gradle 142

D SentinelCommAdapter.java 143

E SentinelCommAdapterDescription.java 146

F SentinelCommFactory.java 147

G SentinelCommAdapterModule.java 149

H SentinelProcessModel.java 150

I Emergency protocols 152

List of Figures

1.1 Maturity level graph (Modrák and Šoltysová, 2020) 23
1.2 Autonomous Flying Platform (Gavilán (2021)) 28
1.3 Drone market share per industry sector (StartUs-Insights, 2023) 29
1.4 Framework for Advanced Manufacturing Technology adoption (Maghazei et al., 2022) 33
1.5 Drone flying space boundary 38
1.6 System architecture indoor drone (Khosiawan et al., 2019) 39
1.7 Block diagram for motion estimation: a) VO and b) SLAM (Yousif et al., 2015) 40

2.1 Supervisor interaction with the system 50
2.2 Example of a conflict in P1, P2, and P3 54
2.3 Linear approximation of the power consumption model 60

3.1 Process flow by part 68
3.2 Current layout - Batch Production System 69
3.3 Updated layout - Drone integration 70
3.4 Conflict Management Algorithm 71
3.5 Standard deviation of demand per hour 85
3.6 Production performance when demand/(total capacity) varies significantly and standard deviation $\geq 30\%$ of system mean 86
3.7 Piecewise-linear fit of profit functions 88
3.8 Bootstrap distribution of estimated switching thresholds 89
3.9 Value of flexibility 90

4.1 Path reservation 99
4.2 UAV System Architecture 101
4.3 openTCS system components 103
4.4 openTCS Plant overview 105

5.1 Primary and secondary accidents from indoor drone operation 115

5.2 Sentinel image sensors 120
5.3 UAV navigation system architecture 121
5.4 Cause-Consequence relationship 123
5.5 UAV risk assessment 124
5.6 Risk evaluation framework 127
5.7 UAV risk assessment 127

A.1 Working Situation Model based on Hasan (2002) 139

List of Tables

1.1 Intralogistics technology trends 24
1.2 Drones classification (Krystosik-Gromadzińska, 2021) 27
1.3 Drones within the Maritime Industry (Krystosik-Gromadzińska, 2021) . . . 31
1.4 TSP-D Contributions (Macrina et al., 2020) 43
1.5 DDP Contributions (Macrina et al., 2020) 44
1.6 Drone-feature approach models (Macrina et al., 2020) 45

2.1 Proposed model 61
2.2 DDP model Dorling et al. (2016) 61

3.1 Working heights per group 65
3.2 Simulation parameters 74
3.3 Linear regression analysis 75
3.4 Throughput and Utilization Rate - Batch Production System 76
3.5 Throughput and Utilization Rate - Drone Delivery System Scenario 77
3.6 Drone utilization rate 78
3.7 Battery level at Charging station 79
3.8 DDS Performance Comparison — Warehouse (left) and Maintenance (right) 79
3.9 System performance compared to total demand 84
3.10 System performance with stable demand 84
3.11 Piecewise-linear analysis - parameter results 88
3.12 Comparison of operational modes across variability levels. 93

5.1 Population density scenarios 117
5.2 VIO estimated error using Sentinel onboard navigation system (Dias et al., 2024) 122
5.3 Population density scenarios - Aircraft & drone 126

Introduction

In recent years, the increasing complexity and customization of manufacturing systems have placed significant pressure on internal logistics. Modern production environments no longer operate under stable, predictable demand but instead must adapt to fluctuating markets, shorter product life cycles, and the growing need for personalized goods. These shifts challenge conventional material handling solutions, which were primarily designed for repetitive, large-batch production. As industries transition toward agile and highly responsive systems, intralogistics must evolve to ensure that material flow keeps pace with the dynamic requirements of the shop floor.

Within this context, drones have emerged as a promising complement to ground-based technologies. Their ability to exploit unused aerial space, coupled with advancements in autonomy, navigation, and localization, makes them well-suited to address the responsiveness gap in traditional material handling systems. For example, recent pilot projects at companies such as ZF Friedrichshafen (Veil, 2019) and SEAT Martorell (Gavilán, 2021) have demonstrated the potential of UAVs for indoor material handling, yet their widespread adoption remains constrained to unresolved operational and organizational challenges.

While their industrial use seems limited compared to other automated solutions, drones offer unique advantages in scenarios where speed, flexibility, and single-order responsiveness are essential. The motivation of this work is therefore rooted in exploring how aerial vehicles can be systematically integrated into manufacturing systems, not as replacements for existing solutions, but as support tools that enhance agility, resilience, and continuous production in the face of variable demand.

The current work has been carried out at the "Laboratoire des Sciences du Numérique de Nantes" with the collaboration of the École Centrale de Nantes and the Institut Carnot MERS. The focus of this manuscript is on the analysis of a drone fast-pace delivery platform integrated within manufacturing settings. Specifically, drone integration is envisioned as a support tool for internal logistics tasks, where the transport of light-weighted items is a priority to enable continuous processing in highly customized production setups. In fact, enabling a continuous production line when demand varies considerably through time depends on two factors; supplying the right items on time to the shop-floor

and keeping the normal functioning of the machines.

While ground robots have shown to be an efficient solution as flexible material handling systems, their behavior and properties are limited to environments where the demand is known beforehand, so agents can be scheduled accordingly using their maximum capacity. Scenarios where the process depends on single-order deliveries to optimally use plant capacity cannot benefit from a cost-efficient way to integrate such technologies.

In this sense, this study evaluates current drone technology present in the market, and design an Advance Manufacturing Technology adoption framework that managers can use as a basis for the creation of profitable business models.

Problem Statement

Although ground-based robots have proven effective as flexible material handling systems in manufacturing, their efficiency relies on predictable demand scenarios where agents can be scheduled to maximize capacity, leaving gaps in environments that require single-order or one-piece flow deliveries. This limitation highlights the need for alternative intralogistics solutions that can operate in agile and dynamic production settings. Drones, originally developed for surveillance tasks outside the industrial domain, present a promising yet underexplored support tool for material handling within indoor manufacturing systems. However, their adoption raises significant operational, organizational, economic, and legislative challenges that must be addressed before they can be integrated into agile manufacturing and warehousing processes. The problem lies in the absence of a structured framework that enables managers to assess the feasibility and profitability of drones as complementary intralogistics tools, ensuring effective routing, scheduling, communication, and alignment with broader business strategies while maintaining safety and regulatory compliance.

Research Questions

Based on the challenges identified in the problem statement, this thesis is guided by the following research questions:

- **Operational feasibility:** How can drones be effectively integrated into agile manufacturing systems to support one-piece flow deliveries without disrupting existing material handling solutions?
- **Organizational adoption:** What managerial and organizational factors influence the feasibility and strategic alignment of drone-based intralogistics solutions within manufacturing environments?

- **Economic evaluation:** Under what conditions can drone adoption provide measurable economic benefits compared to conventional intralogistics technologies?

- **Safety aspects:** What are the main safety constraints for indoor drone operations, and how can they be addressed to ensure safe and compliant integration?

These questions form the foundation of the adoption framework developed in this thesis, and they are addressed through a combination of simulation-based performance evaluation, system architecture design, and organizational analysis.

Objectives and limitations

The objective of this thesis is to model an Advanced Manufacturing Technology adoption framework with the aim of helping managers evaluate if this technology is suitable as a material handling system to enable one-piece flow transportation. In general, feasibility analysis consists of four main axis that should be evaluated prior to piloting a project: Economical, organizational, operational, and legislative. The limitations of this work concern the legislative analysis of the integration of drones within the industry. However, legal aspects related to technical and organizational factors, such as speed limits and safety protocols are not excluded.

Considering that drones, as described in this work, are intended to function as a fast-paced material handling system, the industries that are envisioned to benefit the most are those working under agile systems, either linked to manufacturing or warehousing applications. Agility should be understood as a business strategy adopted to rapidly respond to demand variation, customer needs, or custom orders. With this in mind, the assumptions, and hence limitations for the drone delivery system are as follows:

- Drones can execute one delivery task at the time, once they deliver an item, they must return to a picking or charging location.

- Navigation is fully autonomous. Drones are equipped with forward and rear stereo cameras to handle obstacle avoidance and path modifications in real-time. However, mission path planning and scheduling is computed off-board.

- Drones have fixed paths that should follow without exception. Each time a path is assigned to a drone, it becomes blocked and it cannot be used or crossed by another vehicle.

- Drones can communicate their position in real-time with the scheduler. Resources cannot be liberated unless a drone has sent a confirmation of departure.

- Paths are pre-defined following an obstacle-free strategy. In case obstacles exist at the working height of drones, they are assumed to be fixed, hence paths should avoid any obstacle encounter from advance.

- It is assumed that in every case, the weight of any item to be transported is already known. Drones cannot exceed their payload capabilities.

- Drones are intended to be a support tool that works in synchronization with other material handling technologies, not as a standalone solution.

As the focus of this work will be given to the operational and organizational aspects of integrating drones within agile systems, the adoption framework encompasses the routing, scheduling, and tracking of drones, as well as the communication and interface between each vehicle, the supervisor responsible for task management, and the manufacturing system. Moreover, organizational aspects included in the proposed framework aim to establish the basic components of the business strategy needed to optimally adopt the system.

Contributions

To address the absence of a structured framework for drone adoption in industrial settings, this thesis contributes by developing a comprehensive evaluation approach that integrates both economic and technical dimensions of feasibility. Using a simulation-based methodology, the study assesses drone performance not only as an isolated technology but as an embedded component of a larger production system. The simulation environment replicates the full dynamics of a manufacturing process, including stochastic demand variations, resource constraints, and unplanned events to provide a realistic representation of how drones behave under actual operating conditions. This enables decision-makers to quantify system performance, identify operational bottlenecks, and evaluate profitability under different scenarios. By doing so, the work establishes a foundation for evidence-based adoption strategies that align drone deployment with organizational objectives and process requirements, effectively bridging the gap between technological potential and managerial applicability.

Furthermore, to overcome the lack of realistic operational models and architectures for drone-enabled systems, this thesis also proposes a modular system architecture specifically designed for agile manufacturing environments. The proposed framework delineates distinct functional layers, covering communication protocols, scheduling and routing algorithms, conflict-management mechanisms, resource allocation, and task-management methods, ensuring interoperability between autonomous drones and supervisory systems.

This architecture goes beyond evaluating drones as isolated devices and instead positions them as integral components of a coordinated intralogistics network. By introducing structured communication and control modules, the work provides a scalable framework for drone integration that supports conflict-free operations, efficient resource use, and responsive material flow. In doing so, it contributes a holistic perspective that unites system engineering, logistics management, and autonomous architecture toward the realization of agile, drone-supported manufacturing systems.

Thesis structure

The thesis is organized into four main chapters, each of which progressively develops the discussion from the theoretical foundations of advanced manufacturing to the practical integration of drones as support tools for intralogistics in agile manufacturing systems.

Chapter 1, State-of-the-Art, provides a comprehensive overview of the evolution of industrial paradigms, beginning with the first industrial revolution and culminating in the current era of Industry 4.0 and Smart Manufacturing. The chapter traces how each industrial era has shaped organizational processes, technological development, and value creation.

It then situates drones within the broader context of Information and Digital Technologies (IDTs), Cyber-Physical Systems (CPS), and real-time data acquisition infrastructures, emphasizing the role of digitalization and smartization in enabling autonomous systems. This foundation establishes the conceptual framework needed to understand drones as potential contributors to the digital transformation of intralogistics.

Chapter 2, Problem Formalization within an Agile Manufacturing System, shifts the focus to the specific challenges of integrating drones in operational environments. It formalizes the problem by considering drones as complementary tools to existing ground-based material handling systems, particularly in scenarios where responsiveness and agility are essential.

This chapter introduces the two-stage optimization model used to ensure safe and energy-efficient navigation, highlighting the need for conflict-free trajectories, autonomous resource management, and supervisory control. By defining the operational constraints and technical requirements, this chapter sets the stage for evaluating drones not only as technological novelties but as feasible agents within agile manufacturing systems.

Chapter 3, Pre-Adoption: Economic and Strategic Factors, addresses the early-stage considerations that managers must take into account before deciding whether to invest in drone technologies for intralogistics. The discussion centers on the economic, organizational, and strategic parameters that influence adoption, such as system limitations, safety protocols, and performance metrics.

With it, it is emphasized the role of simulation models in capturing realistic operating conditions, incorporating stochastic production dynamics, and evaluating Key Performance Indicators (KPIs). It also discusses how drone adoption can be aligned with broader business strategies to create competitive advantages, thus positioning drones as enablers of agile and cost-effective operations.

Chapter 4, Early Adoption: Operational and Supply Chain Factors, presents the operational framework required for conflict-free drone integration within manufacturing and warehousing systems. It introduces a conflict-management algorithm supported by centralized supervision, which allocates resources, plans trajectories, and mitigates both resource-based and collision-based conflicts.

The chapter further elaborates on how transfer points, position markers, and real-time localization technologies contribute to system stability and efficiency. By embedding this conflict-management mechanism within a multi-layer UAV fleet management architecture, the chapter demonstrates how drones can be coordinated to operate safely and reliably as part of a larger intralogistics system.

Finally, Chapter 5, Adoption: Organizational and Behavioral Factors, explores the conditions necessary for ensuring the safe and sustainable integration of drones in industrial intralogistics. Here, it is introduced a structured risk management framework that addresses hazard identification, evaluation, reduction, and acceptance, emphasizing the adaptation of aviation safety principles such as the Equivalent Level of Safety (ELOS) to the unique risks of indoor UAV operations. Key considerations include ground impact, mid-air collision avoidance, and system reliability, with risk mitigation strategies ranging from restricted operational zones to advanced guidance and failsafe protocols.

The discussion highlights how supervisory platforms like openTCS, in combination with robust hardware such as the Sentinel drone, reinforce both operational reliability and organizational trust. By linking technical safeguards with organizational readiness, the chapter positions risk management as a central enabler of acceptance and long-term adoption in Industry 4.0 ecosystems.

Together, these chapters provide a structured progression from theory to application, guiding the reader through the historical, technical, economic, and operational dimensions of drone adoption. Overall, this work proposes an adoption framework that managers can use to assess the feasibility of drones as intralogistics support tools, with the goal of enhancing agility, responsiveness, and strategic competitiveness in modern manufacturing systems.

Conferences and Scientific publications

International conferences

- 2024 IEEE International Conference on Technology Management, Operations, and Decisions
  Title: "Towards a Behaviour Model for Drones Integration in Agile Manufacturing Systems"

- 2025 IFAC International Conference on Manufacturing Modelling, Management and Control
  Title: "Integrating drones as intralogistics solutions in the factory of the future"

National conferences

- 2023 LAAS-CNRS Modélisation des Systèmes Réactifs
  Title: Les drones sont-ils efficaces pour supporter l'agilité des processus de l'industrie 4.0 ?

- 2024 SAGIP 2ème Congrès Annuel de la SAGIP
  Title: Les drones sont-ils efficaces pour supporter l'agilité des processus de l'industrie 4.0 ?

Journal articles

- Autonomous Drones in Internal Logistics: A Framework for Drone Adoption in Smart Manufacturing Setups
  submitted to IJPR

Conference papers

- Towards a Behaviour Model for Drones Integration in Agile Manufacturing Systems
  O. A. P. Lara, M. Nouiri, F. Belkadi and A. Chriette, "Towards a Behaviour Model for Drones Integration in Agile Manufacturing Systems," 2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Sharjah, United Arab Emirates, 2024, pp. 1-6
  doi: 10.1109/ICTMOD63116.2024.10878220.

- Integrating Drones As Intralogistics Solutions in the Factory of the Future
  Lara, P., Armando, O., Farouk, B., Maroua, N., and Abdelhamid, C. (2025).

Integrating drones as intralogistics solutions in the factory of the future. IFAC-PapersOnLine, 59(10):2201–2206. 11th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2025
doi: https://doi.org/10.1016/j.ifacol.2025.09.370

Chapter 1

State of the art

As manufacturing systems evolve toward higher levels of digitalization and customization, new logistical challenges emerge, particularly in environments that require rapid, small-batch, or one-piece material flows. Within this context, the integration of unmanned aerial vehicles (UAVs) for internal logistics has gained attention as a potential means to enhance responsiveness and spatial efficiency, though their industrial application remains insufficiently structured and evaluated. This gap motivates the present research, which seeks to establish a coherent understanding of the strategical, operational, and organizational factors influencing drone adoption in smart manufacturing environments.

To situate this problem within the broader industrial transformation paradigm, the chapter introduces how production systems achieve and sustain flexibility. It then clarifies the conceptual foundations that determine a system's ability to adapt to uncertainty, thereby linking flexibility to strategic competitiveness in dynamic markets.

The following section, dedicated to Mass-Customized Manufacturing Systems, expands this analysis by exploring how increasing product variety and personalization reshape intralogistics requirements, leading to the need for more autonomous, real-time material handling solutions.

Building on these theoretical and technological foundations, the chapter aims to address the emergence of drones as industrial assets, outlining their classifications, current applications, and potential to support next-generation intralogistics. With this information, the main challenges for integrating drones into Industry 4.0 are synthesized, identifying the main barriers in the literature.

Together, these sections aim to provide a comprehensive understanding of the state of the art, delineate the multi-dimensional nature of the research problem, and establish the analytical basis for developing a structured framework to assess the feasibility and strategic value of drones in agile manufacturing systems.

1.1 Industry 4.0

Since the first industrial revolution, each industrial era (steam power, mass production, digital transformation, and, lastly, the IIoT) has had a significant impact on the way companies operate, not only in terms of processes and technological development, but also in the perspective and value added to each product or service offered.

The term Industrie 4.0, as we know it today, was first coined in 2011 during the Hannover Messe trade fair, and was later publicly introduced in 2015 at the World Economic Forum by Klaus Schwab (Effoduh, 2016). Since then, it has become one of the main topics of discussion in universities, governments, and industries around the globe.

While the concept has evolved since its inception, mainly due to the desire for a one-size-fits-all definition, it is now generally interpreted as the digital transformation of industrial chains. This transformation ranges from the digitalization of manufacturing and value creation processes to the smartization of plants, suppliers, customers, and distribution channels (Ghobakhloo, 2020).

In other words, the process of digitalization and smartization encompasses both smart technologies and smart value chain networks to provide real-time information on manufacturing systems, components, and products (Zhong et al., 2017).

Other authors, such as Chiarini et al. (2020), add that the main pillar of Industry 4.0 is built on Information and Digital Technologies (IDTs), leading companies toward industrial transformation. On this topic, most researchers agree that IDTs consist of nine core technologies: autonomous robots, simulation, horizontal and vertical system integration, the Industrial Internet of Things (IIoT), cybersecurity, the cloud, additive manufacturing, augmented reality, and big data and analytics (Rüßmann et al., 2015).

Although many of these technologies have been used for some time, recent advancements have made them more efficient and affordable. In addition, the current focus on integrating emerging technologies such as artificial intelligence and blockchain deserves special attention due to their high potential in the digitalization and transformation of value chains (Ilin et al., 2019). These technologies are especially promising as drivers for the development of autonomous and fully integrated smart systems.

On the other hand, the term Smart Manufacturing originated in the United States and was first fully described in 2011, outlining its vision and goals. Although some authors attempt to distinguish between the two terms, it can be argued that both Industrie 4.0 and Smart Manufacturing refer to the same concept: to drive and facilitate the adoption of manufacturing intelligence. The key characteristic is enabling smart factories with vertically and horizontally integrated production systems (Thoben et al., 2017).

To achieve this level of integration, a real-time information-sharing system must exist between machines, storage systems, and utilities within a company. At the same time,

these three elements should act and control each other autonomously. Such systems are called Cyber-Physical Systems (CPS) (Baheti and Gill, 2011).

According to Thoben et al. (2017), the main difference between conventional production systems and CPS is that the latter can be understood as a "system of systems", encompassing complex networks of embedded systems, application systems, and infrastructure. In other words, the goal of CPS is to integrate the physical reality with the digital environment, allowing communication between them (Bagheri et al., 2015).

Researchers such as Ustundag and Cevikcan (2017) emphasize that two main functions are necessary for an embedded system to be considered a CPS. First, it should be designed with an advanced level of networking, enabling real-time data provision from the physical facilities while simultaneously processing feedback from the digital platform. Second, it should be capable of autonomously performing intelligent data processing, decision-making, and providing the computational support required by physical facilities (Lee et al., 2015).

While CPS creates the necessary digital infrastructure for communication within the process, having the proper devices responsible for providing accurate real-time data is critical for effective system deployment. For this purpose, radio-frequency-based technologies such as RFID (Radio-Frequency Identification) or more advanced systems like Real-Time Location Systems (RTLS) are ideal for gathering essential data on most process components. Uckelmann (2008) classified real-time monitoring systems based on their primary function as follows:

- **Identification:** Especially RFID with single and bulk reading
- **Locating:** RTLS like GPS and others
- **Sensing:** For example temperature and humidity sensors

By implementing a combination of the three listed technologies, a high degree of data transparency can be expected, along with enhanced flexibility due to faster decision-making processes, and increased overall process efficiency through continuous monitoring of each step in the value creation chain.

In general, the Fourth Industrial Revolution has brought a shift in mindset regarding production processes: from the concept of automated manufacturing to that of intelligent manufacturing. The vision behind this shift is the co-development of the physical and virtual worlds, where elements within the company, including machines, are equipped with sensors capable of gathering critical data on process status (Klocke et al., 2011).

As stated by Thoben et al. (2017), by collecting relevant information in real time, it becomes possible to meet customer requirements without waste, thanks to the reconfiguration of assembly lines and the elimination of set-up times through dynamic business and engineering processes.

1.1.1 Flexibility in the Production Systems

In the context of today's rapidly evolving market, the inability of firms to adapt to dynamic demand conditions often leads to organizational failure. Having a certain degree of organizational and operational flexibility to respond to these changes is crucial to maintain process stability. According to Duguay et al. (1997), flexibility is the capacity to deploy or redeploy production resources efficiently in response to changes in the environment. Thus, it is an inherent characteristic of any production process, regardless of its maturity level.

On this topic, Duguay et al. (1997) clarifies that the term environment does not refer solely to fluctuations in demand but also includes any factor that might disrupt the normal flow of the process, such as machine breakdowns, variable lead times from suppliers, or operator absenteeism. This concept can be translated into having an efficient trade-off cost when a variable changes. Similarly, the MIT Commission defines flexibility as "the ability to deliver high-quality products tailored to each customer at mass production prices" (Dertouzos et al., 1989).

This latter definition was then adopted by Agile Manufacturing Systems (Elkins et al., 2004). The integration of the term agility was simply an evolution of the Lean Manufacturing concept, shifting from an idealized cost-cutting but fixed production system, to one that is both cost-efficient and adaptable. The work of Elkins et al. (2004) provides a clear perspective on what an Agile Manufacturing System is, defining it as a production model that enables the reallocation of production line capacity to products experiencing higher-than-expected demand, while still maintaining resource availability for products with lower-than-expected demand.

1.2 Intra-logistics 4.0

It is clear that the Fourth Industrial Revolution has impacted not only production systems but also everything around them, including material scheduling and handling. Consequently, as industry began introducing more technologies to improve output on the shop floor, logistics professionals started to explore how the same principles could be applied to supply chain management. In other words, the changes and improvements in production lines brought with them new requirements for supply chains (Bukova et al., 2018).

Given that companies are increasingly focused on achieving higher performance on the shop floor, the question of how to coordinate and synchronize production and logistics processes has become a significant research problem (Li and Huang, 2021). One clear example of this integration can be found in the work of Pan et al. (2021), who describes a CPS- and IIoT-based integration framework for these processes, featuring a self-organizing configuration and a data-driven model designed to reduce both waiting time and energy

consumption.

In this context, the term logistics 4.0 can be understood as the link between production and consumption through the use of IIoT tools (Bukova et al., 2018). As stated by Bukova et al. (2018), the digitization and smartization of supply chains lay the foundation for Logistics 4.0. Other authors, such as Winkelhaus and Grosse (2020), also attempt to define the framework of Logistics 4.0 by synthesizing current understandings of logistics as follows:

> Logistics 4.0 is the logistical system that enables the sustainable satisfaction of individualized customer demands without an increase in costs, and supports this development in industry and trade using digital technologies.

In addition, their work also defines an accurate framework for Logistics 4.0, dividing it into three dimensions:

1. **External:** It relates to every change caused by customer demands of individualized products, the current needs of the Industry and the associated aspects of globalisation and sustainability (Kagermann et al., 2013).
2. **Technological:** This dimension consists on the technological drivers that enable the switching from traditional logistics to Logistics 4.0, taking advantage of the development of the Industry 4.0 but not only limited to it.
3. **Logistical:** Here there is a need to distinguish three different elements; Tasks, domains and human factors. The tasks involve the management activities and the execution activities, which at the same time are split into four categories (called domains) based on the natural flow of the Supply Chains: Supply logistics, intra-logistics, distribution logistics and reverse logistics. Finally, the human factors refer to the psycho-social interactions that influence the performance of the logistical system.

Although there is a clear link between the terms Industry 4.0 and Logistics 4.0, it is important to highlight that the literature lacks a precise account of how the concept of Logistics 4.0 emerged. Nevertheless, there is no doubt that it appeared as a response to, and in support of, Industry 4.0, not the other way around (Radivojević and Milosavljević, 2019).

In this context, the literature review conducted by Winkelhaus and Grosse (2020) sheds light on this issue by concluding that, while the term "4.0" has been widely adopted in the field of production research, logistics has not systematically adopted the same denomination.

This evolution towards intralogistics 4.0 is particularly demanded in highly customized production environments. As manufacturing systems move from standardized production

toward greater product personalization, the need for flexible, responsive, and data-driven material handling solutions becomes increasingly critical. In mass-customized contexts, production lines must frequently adapt to small batch sizes and rapid product changes, which in turn place significant pressure on internal logistics to supply the right materials, at the right time, and in the right quantity.

1.3 Mass-customized manufacturing systems

The growing adoption of digital and interconnected production systems has enabled firms to pursue higher levels of customization while maintaining competitive costs. However, this shift toward individualized production fundamentally alters the dynamics of material flow within the factory. As product variety increases, batch sizes decreases consequently. And so, traditional logistics systems, often optimized for predictable, large-scale operations, struggle to maintain efficiency under fluctuating and highly variable demand conditions.

By nature, a custom product is designed to satisfy the specific requirements of a particular customer (Duray et al., 2000). Based on this idea, Lampel and Mintzberg (1996) established the notion that the degree of customization of any production system is determined by customer involvement within the production cycle. McCutcheon et al. (1994) took a similar perspective, arguing that the main variable determining the design of the production process is the stage at which a product is differentiated.

Combining both ideas, it can be argued that the degree of customization of any production system is directly linked to the processes that are impacted by, and hence should be adapted to, customer requirements. In this context, Mintzberg (1988) classifies customization into three forms: pure, tailored, and standardized. A pure customization strategy considers customer needs throughout the entire supply chain, from product design to the delivery of the finished good. Tailored customization strategies focus on producing basic sub-components and personalizing the final product with unique elements at a later assembly stage. Finally, in a standardized customization strategy, goods are differentiated based on a standard set of options; hence, the customer is not involved in the production cycle.

More recently, Modrák and Šoltysová (2020) proposed a maturity-level framework for mass-customized manufacturing systems (Fig. 1.1). They divided organizational capabilities into five stages: conventional, starting, moderate, advanced, and optimized. At the conventional stage, there is no degree of mass customization; it represents a traditional production approach offering standard products to similar markets.

The starting stage is the first level of a mass customization system. At this stage, massively produced items are delivered to different markets based on their needs, with

an emphasis on optimizing logistics operations. At the moderate stage, the emphasis is placed on product modularization. This means that core elements of the products are mass-produced; however, they are assembled based on customer needs. Integration of customized solutions across the supply chain is required to reach this level.

The advanced stage adapts the entire organizational production model to accommodate one-piece flow processes. At this level, automatic identification technologies for monitoring and tracking resources and parts within the production system and supply chains play a key role. Finally, the optimized stage focuses on real-time communication and synchronization across the entire supply chain. In other words, the supply chain becomes an autonomous, real-time demand chain.

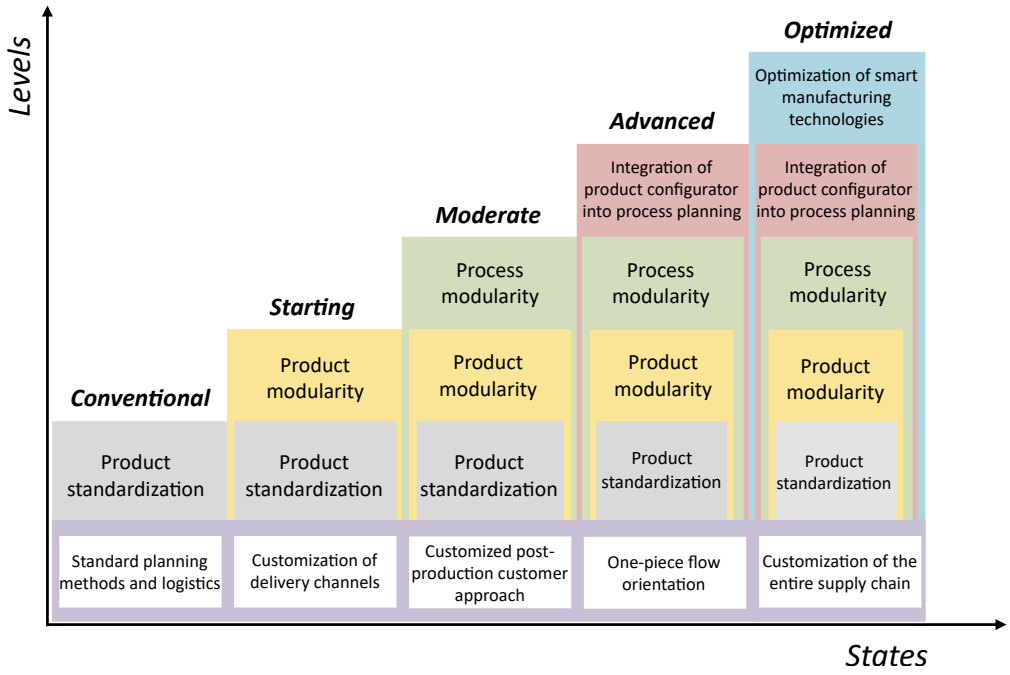


Figure 1.1: Maturity level graph (Modrák and Šoltysová, 2020)

While implementing a mass customization strategy can bring benefits for both producers and manufacturers, a critical consideration is the identification and selection of the degree that best fits the target market. As Pine (1999) states, even if a firm has the flexibility in manufacturing and the capability to gather customer needs, if it cannot transfer customer preferences efficiently into a suitable product, the best available manufacturing technology is meaningless.

1.3.1 Intra-logistics in Mass Customization systems

In addition to adding complexity to production processes, mass customization also negatively affects the internal supply chain by increasing costs and slowing down its velocity.

As shown in the Maturity Level graph, the higher we go on the scale, the further we move from the production line. In fact, a higher degree of customization involves a more complex internal and external logistics system.

In the literature, many authors emphasize logistics (and intralogistics) as a key factor in achieving competitive advantage (Porter, 1985). Unlike modifications or improvements to a product, attaining flexibility in logistics is challenging to replicate due to the complex changes in personnel, technology, facilities, and strategic corporate business models (Sarder et al., 2006).

Higher degrees of customization also require a higher degree of process and product modularity, so implementing high-customization production systems often results in highly distributed shop floors across the plant layout (Galasso et al., 2009). Being able to manage dispersed processes under uncertain demand presents a significant challenge in achieving low-cost, high-variety production (Cedillo-Campos and Sánchez-Ramírez, 2013).

Rungtusanatham and Salvador (2008) referred to these challenges as hindrance factors that impede companies' transition from mass production to mass customization. In a mass production environment, having a large enough safety stock to respond to fluctuations in demand, as well as a buffer to continuously feed the production line, is essential. However, in a mass customization scenario, surplus inventory and buffers would create an enormous stock of highly volatile products that may not meet customer needs. By minimizing decoupling inventory along the supply chain, the urgency of having a synchronized delivery system across the entire chain, both external and internal, becomes crucial (Rungtusanatham and Salvador, 2008).

1.3.2 Intralogistics trends

With the increasing need for flexibility, logistics professionals have been seeking supply solutions that can be efficiently adapted to different scenarios. Based on the research of Lakner and Winkler (2018) regarding trends in the literature on automating intralogistics processes, 48 publications summarize the most mentioned technologies for optimizing the internal supply of items, as shown in Table 1.1.

Trend	% mentions	% people would invest
Self-organizing conveyor systems	52.08%	35.4%
Automated Guided Vehicles	43.75%	43.9%
Collaborative Robotics	29.17%	63.4%
Automated Picking Systems	27.08%	35.4%
Drone technology	8.33%	18.3%

Table 1.1: Intralogistics technology trends

The study encompasses additional trends; however, concerning delivery systems, the

technologies shown in Table 1.1 are the only ones identified. On the other hand, the percentages are calculated based on the number of times each technology was mentioned across the 48 articles reviewed, meaning that some trends may have appeared together in the same research.

Following the work of Lakner and Winkler (2018), Winkler and Zinsmeister (2019) carried out a survey to determine the most influential trends in the industry for logistics managers in terms of willingness to invest. The survey was completed by 105 individuals from diverse backgrounds within the logistics sector. The results are summarized also in Table 1.1.

In a more recent study, Dabić-Miletić and Raković (2023) used the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) to compare and evaluate the most suitable technologies for automating intralogistics activities. The research encompasses an eight-criteria ranking system, defined as follows:

1. Sensitivity of the warehouse system [min]
2. Possibility of mistake [min]
3. Dependence on labor force [min]
4. Operating costs [min]
5. Flexibility [max]
6. Productivity [max]
7. Carrying capacity [max]
8. Navigation system level [max]

Each criterion has an optimal expected result, either the maximum (max) or minimum (min) value found among the evaluated technologies. Dabić-Miletić and Raković (2023) selected four technologies for the analysis: forklifts, AGVs, AMRs, and drones. The findings revealed that Autonomous Mobile Robots (AMRs) were the best fit for carrying out intralogistics tasks. Drones ranked second, while Automated Guided Vehicles (AGVs) came third. Although forklifts offer numerous benefits in material handling, they were deemed the least effective for automating the warehouse system.

In a similar manner, de Paula Ferreira et al. (2022) provide a framework to enable companies to identify Industry 4.0 tools that can efficiently improve their processes. The framework serves as a guideline to assist in project conception, portfolio selection, and planning, depending on different scenarios during the development phase. Among the 34 use cases proposed by the authors, drones are identified as a feasible tool to improve the lead time and response time of decentralized and autonomous transportation systems.

1.4 Drones in the Industry 4.0

Unmanned Aerial Vehicles (UAVs), or "drones" as they are better known, represent an emerging technology that has attracted significant interest from manufacturing and logistics companies due to their flexibility, having virtually no limitations in terms of degrees of freedom and working space (Deja et al., 2020). However, their practical and profitable applications still present a barrier for researchers and practitioners.

Maghazei and Netland (2020) described this issue, explaining that there is currently a significant gap in understanding and differentiating between what drones are capable of and what actually makes sense for them to do. In fact, the recent technological maturity of drones has enabled industrial applications in fields such as agriculture, construction, and mining, which benefit from their inherent ability to reach areas that may be dangerous, costly, or time-consuming for humans.

As can be inferred, most of these applications take place outdoors, whereas the majority of manufacturing operations are performed indoors. The main reason for this is that outdoor applications can be efficiently managed using GPS for routing, localization, and positioning. In contrast, indoor drones require more complex systems to achieve the same level of autonomy, such as LiDAR, infrared rangefinders, sonar, ultra-wideband radio signals (UWB), simultaneous localization and mapping (SLAM), and motion capture systems, to name a few.

1.4.1 Types of drones

Currently, it is possible to find semi-autonomous and even fully autonomous drones on the market for some specific tasks. Depending on the application, different types of drones may perform better in particular scenarios. Although drones can be classified in various ways, such as by size or application, this work focuses on medium-sized drones suitable for industrial use.

Based on the work of Krystosik-Gromadzińska (2021), drones can be classified into four distinct categories according to their flight design. These categories are presented in Table 1.2.

Among the drones listed in Table 1.2, the most commonly used for industrial applications are multi-rotor UAVs, due to their adaptability to various scenarios as well as their low cost and complexity. These drones can be further sub-classified based on the number of propellers they have, such as tricopters, quadcopters, pentacopters, hexacopters, and octocopters.

The widespread use of multi-rotor drones is largely due to their ability to hover and perform vertical take-offs and landings, which facilitates deployment and landing in confined or hard-to-reach areas.

Drone classification		
Drone type	**Pros**	**Cons**
Multi-rotor	Maneuverability Easy use Suitable for cameras Suitable for indoors	Short flight Small payload capacity
Fixed-wing	Long endurance Large area coverage Fast flight speed	Need large areas Hard to fly High-end training required Expensive
Single rotor	Long endurance High payload capacity	Dangerous Hard to maneuver High-end training required Expensive
Fixed-wing hybrid	Long endurance	Not perfect either for hovering or forward flight

Table 1.2: Drones classification (Krystosik-Gromadzińska, 2021)

Additionally, increasing the number of rotors directly enhances their reliability. While adding more than four rotors does not always result in improved performance, it significantly improves safety. For instance, in configurations with more than four rotors, the failure of a single motor does not necessarily lead to a crash; the drone can often still execute a safe landing or be flown carefully to a secure area. This would be nearly impossible in the case of a rotor failure on a quadcopter.

1.4.2 Main applications of Drones within the Industry

Thanks to the rapid development of technology and the growing interest of companies in becoming safer, more sustainable, and more responsive, there is extensive research available on the technical capabilities of drone technology. However, limited attention has been given to its practical applications in manufacturing (Maghazei and Netland, 2020).

On one hand, the lack of technological maturity creates a risky environment, which leads companies to postpone any consideration to adopt a new technology in their facilities. On the other hand, the general perception of drones in society and industry has been negatively influenced by their association with non-industrial applications, particularly in military and surveillance contexts.

In modern facilities, fixed and mobile installations such as cranes, conveyors, and vehicles could potentially be replaced by drones. In fact, some companies have already begun this transition. For example, ZF Friedrichshafen (Germany) became the first company in the country to receive government approval for the use of an automated drone (Veil, 2019).

In brief, the drone system developed at the German facility consists of an automated delivery platform designed to support intra-logistics by interfacing the warehouse and manufacturing area. As a prototype, the drone was used to deliver items such as sensors and control cards to maintenance workshops. Technologically, the UAV was designed as a hexacopter with a total payload capacity of 5 kg. Considering that the grippers and container used to carry the parts weighed 2 kg, the available payload capacity for deliveries was 3 kg, with a flight autonomy of 30–40 minutes (Veil, 2019).



Figure 1.2: Autonomous Flying Platform (Gavilán (2021))

Another example of intralogistics drones can be found at SEAT. In 2021, the facility located in Martorell, Spain, in collaboration with Eurecat launched the project "Indoor: Autonomous Flying Platform" (Figure 1.2), which focused on mapping, programming, and developing obstacle avoidance algorithms to train the vehicle for every possible situation (Gavilán, 2021). As stated by Gavilán (2021), the goal was to connect nearby suppliers and maximize their supply chain efficiency by establishing a direct endpoint to the production line, thereby eliminating the need for warehousing.

Although companies are beginning to realize the full potential of autonomous drone systems, only a few in the manufacturing sector are actively pursuing fast-paced, highly adaptable supply systems. This contrasts with other industries, such as construction and agriculture, where drone applications translate more quickly into short-term financial benefits. To provide better insight into the current use of drones, Figure 1.3 shows the drone market share within various industries, based on technology usage in the USA during 2023.

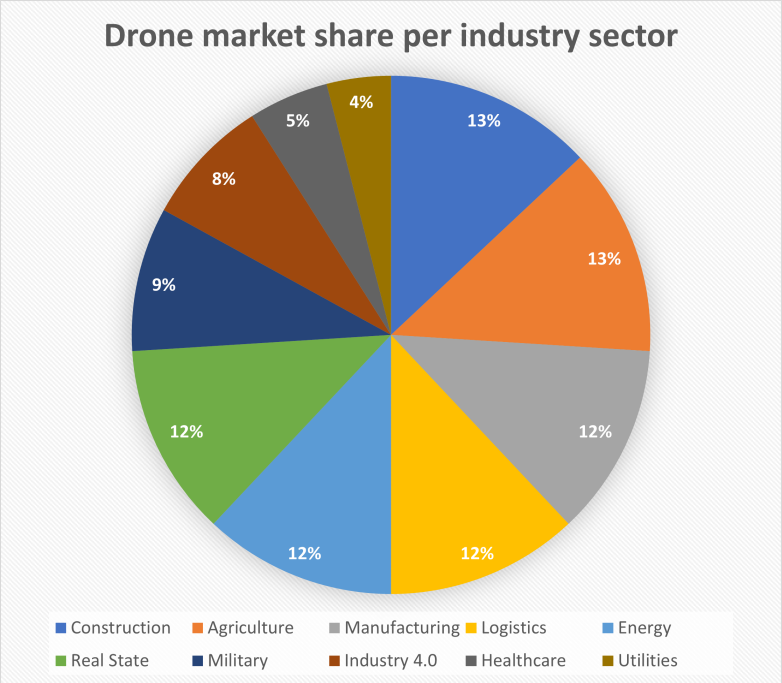


Figure 1.3: Drone market share per industry sector (StartUs-Insights, 2023)

Construction

In the construction sector, drones can easily take over the tasks typically performed by specialized technicians or engineers responsible for mapping and land surveys of a given area. The great success of UAVs in this sector is primarily due to their ability to access large or hard-to-reach facilities, as well as complex environments (Zaychenko et al., 2018). Additionally, they can capture accurate images and videos, along with detailed information about the surroundings, which can be used for:

- Land surveying
- Inspections
- Provide visual tools for customers and employees
- Monitor the progress
- Security surveillance
- Mapping

On top of that, advanced drones can perform multiple tasks simultaneously, directly saving a significant amount of time compared to manual methods or older technologies.

Agriculture

Similar to the construction sector, experts in agriculture primarily use UAVs to monitor their crops. However, they also go further by adapting these flying vehicles to spread

herbicides, fertilizers, and seeds across the fields. Veroustraete (2015) precisely listed and described some of the most useful current drone applications in this sector. More specifically, the author identified the following uses:

Mid-season Crop health monitoring This can be done by evaluating the Normalized Difference Vegetative Index (NDVI) with the use of specialized sensors.

Irrigation equipment monitoring Added to being a tedious task to perform manually, the complexity of monitoring the irrigation system could become a problem when the crops reach certain height.

Mid-field weed identification With the use of NDVI sensors, it is also possible easy to identify areas where there is a high-density weed proliferation, helping to remove the problem before the season of harvesting arrives.

Maritime

The use of drones in the maritime sector is becoming a point of interest for smart factories. One of the most important applications in this sector is their role as the "eyes and ears" of stakeholders, flying easily into areas that could be dangerous or difficult for humans to access. In a 2021 study by Krystosik-Gromadzińska (2021), it was estimated that using drones could reduce the total duration of inspection surveys by 90 to 95%, meaning that a task that normally would take three days to finish, drones could complete it in just two or three hours.

In other words, the integration of UAVs in the maritime sector has been shown to help companies save between €20,000 and €30,000 per large asset survey (Krystosik-Gromadzińska, 2021). Although aerial surveys show great potential in this field, there are many other applications that autonomous or semi-autonomous drones can perform to improve sector performance. Some successful examples already implemented in the industry are listed in Table 1.3.

Despite these promising applications, several major challenges hinder the industrial adoption of drones. While the potential of UAVs in enhancing efficiency, safety, and responsiveness is widely recognized, their integration into manufacturing environments remains limited due to various technical, organizational, and societal barriers. Many companies remain cautious, not only because of the current immaturity of drone technologies but also due to uncertainties surrounding their operational reliability, regulatory compliance, and economic viability. Consequently, understanding these obstacles is essential for bridging the gap between conceptual research and practical implementation.

Drone applications in the Maritime sector	
Company	**Application**
Airbus	Delivering packages onto ships anchored offshore
Wilhelmsen Ship Service	Delivering spare parts and documents to ships
The UK Royal Navy	Identifying ship defects
EMSA	Border control, pollution monitoring, detection of illegal activities

Table 1.3: Drones within the Maritime Industry (Krystosik-Gromadzińska, 2021)

1.5 Challenges for integrating drones into the Industry 4.0

Nowadays, integrating new technologies into industry involves analyzing various entry barriers. In this context, Maghazei and Netland (2020) clearly identified five generic categories that limit the feasibility and practical applications of industrial drones, naming them as follows:

- Technological
- Operational
- Organizational
- Legislative
- Societal

Logically, the current state of technology and its limitations are the main drawbacks for drone applications. Even if control and localization problems are solved, battery constraints significantly limit the efficient use of drones. Based on the specifications of drones available on the market, the average autonomy of UAVs is about 30 minutes, dropping to less than 10 minutes depending on the payload carried (Deja et al., 2020).

Regarding operational challenges, safety algorithms and protocols should provide companies with the necessary tools to ensure that drone integration into the shop floor does not disrupt existing processes. Basically, they should enable drone autonomy in case of malfunctions or signal loss, avoidance systems when near objects, and, more importantly, when flying around people.

Within economical factors, Maghazei and Netland (2020) states that a business case containing sufficient information about an acceptable return on investment is of major importance. While the costs of acquiring and adapting processes to fit UAV integration are easy to identify, proving the potential savings that drones can bring is a more complex task.

Concerning legislative barriers, particularly for indoor applications, these are generally less problematic compared to other categories. Mainly because regulations predominantly apply to outdoor scenarios. However, this does not mean they can be ignored; the rapid technological advancements and growing understanding of real-world applications will soon lead to a well-defined regulatory framework for their consolidation in industry (Maghazei and Netland, 2020).

Finally, societal concerns that may interfere with drone adoption largely stem from public perception. It is not surprising that the general public is wary of UAVs, given their historical use in military contexts. Privacy is another major concern. As revealed by Opray (2016) in a study conducted in Australia, drones can be used to supervise operators' activities in real-time.

Additionally, the noise generated by drones, combined with fears of malfunctions causing the vehicle to fall or crash into someone while the blades are still spinning, creates further barriers. Even if technological challenges are overcome, psychological concerns will remain and must be addressed.

As discussed in the work of Maghazei et al. (2022), the main problem companies face when attempting to implement fast-emerging technologies is that they do not follow a linear adoption pattern based on a clearly defined business case. Instead, companies often evaluate and pilot "use cases" that merely have the potential to evolve into business cases over time. Current literature on indoor drones, for example, tends to focus on identifying various use cases and addressing the technical and operational challenges associated with their implementation (Askerbekov et al., 2024). In this context, it is crucial that researchers and practitioners understand the technology's maturity level and distinguish between use-case and business-case adoption (Maghazei et al., 2022).

As shown in Figure 1.4, before considering the adoption of a technology, managers should focus on its strategic and economic potential (Hayes and Jaikumar, 1991). However, current methodologies do not support drone performance evaluation in discrete event systems.

1.5.1 Strategic level

When assessing a new technology, the value of long-term benefits is often underestimated, creating a bias against novel technologies (Baldwin, 1991). In most cases, easily identifiable implementation costs and short-term productivity inefficiencies overcome any

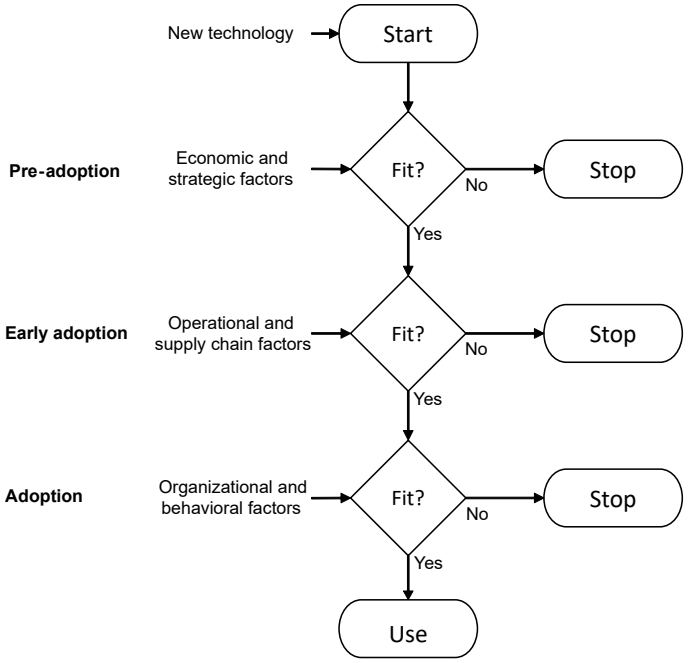


Figure 1.4: Framework for Advanced Manufacturing Technology adoption (Maghazei et al., 2022)

potential interest (Chew et al., 1991). In this regard, Small and Yasin (1997) state that these drawbacks in the financial assessment can be addressed by aligning firm strategies in advance with technology adoption. However, following a strict alignment process to suit current market trends and the existent degree of customization only enhances the current strengths of a firm, rather than creating disruptive innovation (Maghazei et al., 2022).

Based on the work of Hayes and Jaikumar (1991), companies tend to follow "diagonal cases", in which products are matched with their "natural" process structure. This means that for a determined volume, standardization, and degree of customization of a product structure, a likely process structure would offer manufacturing advantage. However, when companies seek competitive advantage, they may move "off the diagonal" to a more flexible process structure. As can be expected, the more a company moves away from its "diagonal", the more unique it becomes compared to its competitors. Nevertheless, implementing this shift without understanding the real implications of such a drift will lead to failure.

Integrating a new Advanced Manufacturing Technology should be done by understanding and evaluating the Key Performance Indicators necessary to judge the performance of such technology before its adoption (Small and Yasin, 1997). Establishing appropriate performance measures can help managers discern which technology to adopt based on the process structure they plan to align with (Swamidass, 1987).

On this topic, Suresh and Meredith (1985) point out that even if the performance

measures are known, evaluating the quantitative benefits arising from integrating new technologies should not only follow a direct benefit analysis. In most cases, the so-called indirect savings can amount to several times those of direct savings. In general, by using inadequate performance evaluation systems, many major saving areas are not included in the economic analysis, resulting in longer or even negative return on investment periods. Following a classical quantitative-oriented single-valued criterion procedure, such as ROI, do not reflect the real impact of the technology adoption within the system, meaning that other benefits that may lead to strategic advantage are easily overlooked.

Assessing strategic advantages for technology adoption should be based on a variety of factors. Criteria such as flexibility, feasibility, and cost should be included and justified in the evaluation models (Suresh and Meredith, 1985). Although most most of these factors follow a stochastic behavior, until now, stochastic approaches have seen limited application in economic analysis. However, the evolution of modern manufacturing systems has made their use increasingly essential. While stochastic simulation techniques are commonly employed to assess the effects of different scheduling strategies on production cells and to assist in system configuration, they are still not fully embedded in the economic justification process (Suresh and Meredith, 1985).

In fact, using simulation tools that not only rely on linear methods, but on stochastic approaches has demonstrated how several strategic and tactical objectives can be addressed, and hence effectively measure their economical benefits (Swamidass and Waller, 1990).

1.5.2 Operational level

Achieving technology integration within the company's operational model considers the technology alignment with internal and external resources and processes in the system (Maghazei et al., 2022). In fact, the adaptation process for implementing a new technology should not occur as an isolated event, but as an interconnected solution, where technical factors, as well as human and operational resources are interdependent, and none is of prime importance (Frohlich and Dixon, 1999).

Leonard-Barton (1988) refers to this interaction as "mutual adaptation". In his work, the concept is defined as a dynamic process of adaptation between the technology and its environment. Similarly, Van de Ven (1986) states that:

> "Innovations not only adapt to existing organizational and industrial arrangements, but they also transform the structure and practice of these environments"

In this context, Leonard-Barton (1988) highlights that adaptation processes are unavoidable since, in most cases, new technologies are not perfectly compatible with a pre-

structured user environment. Although some uncertainties linked to the new processes can be reduced by prototyping and testing, "as soon as the technology gets into the hands of the production department, the complexity will increase again" (Carter, 1979). Complexity can be understood as any misalignment in one of the following criteria:

1. Technical requirements
2. System that allows proper functioning and interaction with the technology
3. Performance

The loop of "adaptation" can be done by aligning the technology to the system or changing the environment in which it is implemented, and in some cases, changing both. As long as this mutual adaptation process remains dynamic, technology may dictate process structure, or vice versa (Gerwin, 2007).

Regarding technical misalignment, it is expected that the technology has some level of readiness to reach operational feasibility, at least in a laboratory setting (Leonard-Barton, 1988). However, scale-up problems may appear during the implementation process: simulating an end-user environment in laboratories does not always accurately represent the behavior of the final application. For this reason, running all the modules of the system through interconnected tests are of main importance. Without this, simulations cannot have enough data to evaluate whether the technology meets its technical specifications (Leonard-Barton, 1988).

On the other hand, aligning technology with its delivery system may seem simple. However, in some cases, users have difficulties to distinguish between which characteristics, and hence which issues are inherent in the technology and which belong to the system (Leonard-Barton, 1988). One clear example can be the response time of a system after receiving a new command to be executed. In this case, it can be implied that any delay in response time is due to software issues. However, misalignment in the hardware of the technology used, or in the communication protocols may be the reason for failure.

Finally, achieving performance optimality with high significance and positive impact requires managers to foresee critical changes that may be exploited with the new technology (Leonard-Barton, 1988). According to Das and Narasimhan (2001), performance can be evaluated once it is achieved what they called "process integration". In this context, process integration occurs when there is a suitable environment in which internal and external resources, as well as processes, work under optimal conditions.

One important indicator to foresee operational integration is based on the type of technology and how it interacts with the system. Meredith and Suresh (1986) distinguish three different types of technology based on this interaction:

1. Stand-alone

2. Intermediate
3. Integrated systems

Although the integration of stand-alone technologies has been characterized by the use of mathematical models that narrow down problems to single-valued criteria (Suresh and Meredith, 1985), integrated systems are in need of new models that can integrate multiple-criteria analysis to help decision-making. In this topic, Hutchinson (1976) states that modeling integrated systems in flexible environments should, in fact, consider the higher variable costs compared to transfer lines, but also indirect benefits that can impact the capacity of the production line. In this case, the benefits of flexibility are mainly due to the ability to increase production capacity gradually, rather than all at once, and to use this increase in capacity to dynamically produce more than one product.

Following a traditional product life cycle pattern where demand starts low, gradually increasing over time to its peak level, where it starts to decrease until it is stable, Hutchinson (1976) simulates the performance of a manufacturing system in a stochastic environment to compare the benefits of flexibility. First, the author assumes that all manufacturing technologies are inflexible and tailored to transfer lines. For the second scenario, the author models an environment with technologies that enable Flexible Manufacturing Systems. After running 192 simulations, results suggest that Flexible Manufacturing Systems are better suited, and hence provide higher benefits when the rate of new product introductions and the maximum capacity of the system increases. On the other hand, inflexible transfer lines showed a better performance when few products in high volume are produced, reducing the average interest rate and increasing the total output.

Path planning

Regardless of the operational environment in which drones are deployed, their ability to operate autonomously mainly depends on the efficiency of the path planning algorithms that guide them through the layout (Mourtzis et al., 2024). In practical terms, path-planning is typically formulated as an optimization problem, where algorithms determine an optimal route by accounting for various constraints and decision criteria, including collision avoidance, travel distance, energy management, and mission duration (Mourtzis et al., 2024).

Within two-dimensional applications, path planning is often addressed through algorithms such as A* (A-star) and Dijkstra, both of which are frequently cited in the literature for their ability to compute optimal routes. However, drone navigation should be enabled by developing more complex approaches that include 3-D (three dimensional) motion (Mourtzis et al., 2024).

Unlike outdoor applications, the complexity added to indoor path planning due to overcrowded layouts, resource sharing, and tools needed for accurate positioning, negatively impacts the research around them, reducing the amount of publications on this topic (Mourtzis et al., 2022).

Han et al. (2022) proposed one of the few works on indoor path planning for UAVs based on a grid-optimized algorithm. As stated by Wu et al. (2022), the grid method partitions the airspace into uniform two- or three-dimensional grids, thereby offering an efficient data model for conflict detection and path-planning tasks during the preflight stage. In confined indoor environments, path planning must guarantee precise and safe navigation throughout the UAV's trajectory in three-dimensional space (Han et al., 2022).

Based on the work of Han et al. (2022), current path planning solutions do not offer a perfect fit for indoor flight. In general, algorithms do not integrate a standalone solution for the most important issues to enable autonomous flight:

1. Fast path planning in complex 3D indoor airspace
2. Avoidance of local optima or deadlocks in non-optimal conditions
3. Accurate and realistic flight path results

On this topic, Mourtzis et al. (2024) highlights that an optimal path-planning process for UAVs generally involves two key steps: first, representing or visualizing the UAV within a three-dimensional environment to identify objects and obstacles in the free workspace; and second, constructing a map or graph that incorporates the UAV's configuration and operational requirements within that same environment.

Scheduling

Drone scheduling encompasses a range of considerations, including launch points, customer visits, delivery tasks, monitoring activities, and battery management (Pasha et al., 2022). With this, the available tasks in the system can be assigned to efficiently utilize the vehicles. However, in the case of drones, the number of works found in the literature on the scheduling of UAVs in indoor applications is scarce (Khosiawan et al., 2016).

In fact, schedulers can be understood as the supervisor that organizes and provides commands to the different resources within the system to accomplish the different tasks (Khosiawan et al., 2019). To achieve this, Khosiawan et al. (2019) propose a three-layer architecture that conceptually represents the operation of an UAV in indoor environments. Figure 1.6 shows the different layers and its components.

The first layer corresponds to the "indoor environment". It includes all infrastructure in which the UAV operates. This means that any object that may cause uncertainty during the drone's trajectory is considered. To ensure safe operation, it is necessary to

define a flying boundary around the drone. Unexpected obstacles, such as hanging wires or uncontrolled closed doors that could make the flight path unfeasible, should also be removed. The drone's flight boundary is shown in Figure 1.5.

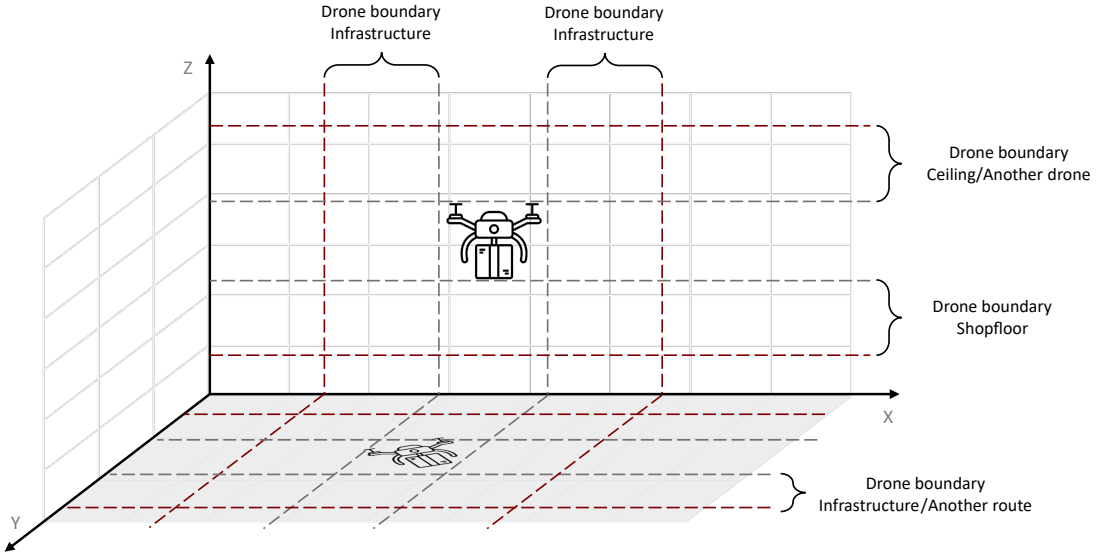


Figure 1.5: Drone flying space boundary

Furthermore, this layer should contain zones and resources already occupied by drones, fixed obstacles, and any other known sources of disturbance. Collision avoidance should be managed in real-time. Although no dynamic obstacles are expected along the flight path, an obstacle avoidance system is still necessary to ensure navigation autonomy.

The second layer consists of the drone operation system. This layer functions as an interface between the drone and the control center. Positioning tools, such as cameras, rangefinders, and LiDAR, should track and report the drone's real-time location at all times. In the event of communication issues with the scheduler, the drone must be programmed with emergency protocols that allow for safe landing.

Finally, the third layer, or "Task layer", contains every command to be executed. This layer contains information regarding the type of task, as well as the pick-up and drop-off locations. It also manages the sequence in which tasks are executed and assigns the vehicle responsible for each task.

Navigation

For a drone to carry out autonomous or semi-autonomous missions, it is required that internal positioning and tracking devices communicate continuously with external systems, such as a ground control station. These systems must identify the location of the drone in real time and relay these data to its navigation controller to sustain autonomous

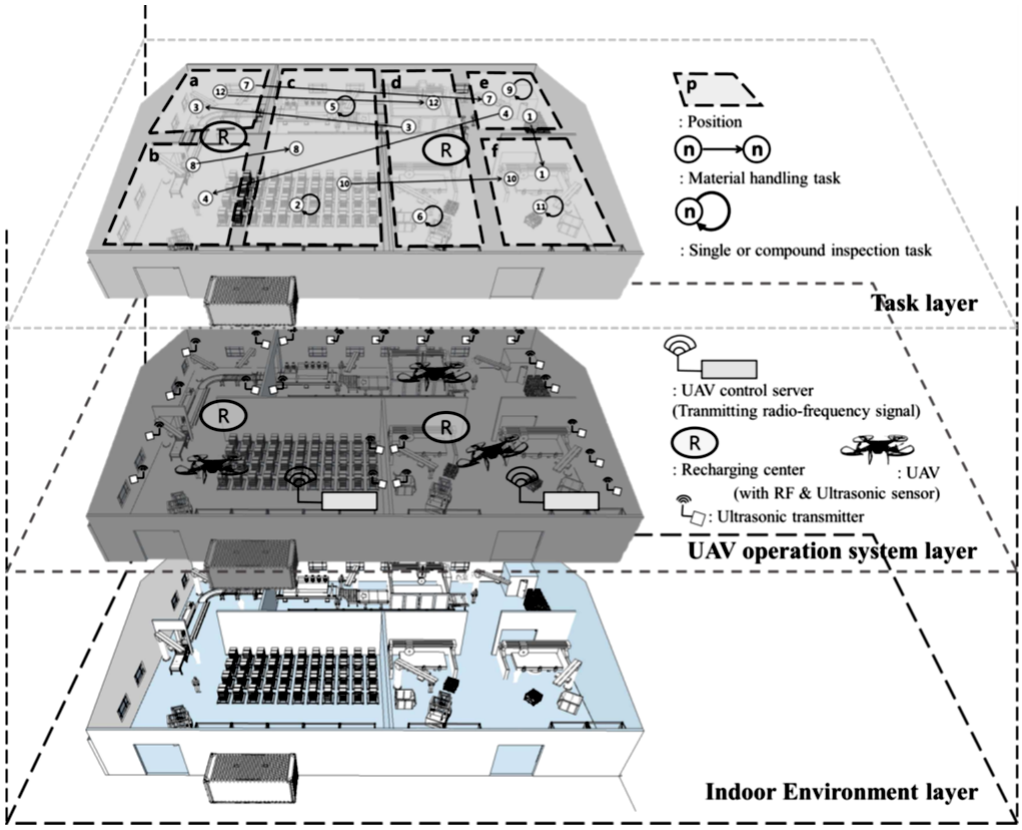


Figure 1.6: System architecture indoor drone (Khosiawan et al., 2019)

flight. In outdoors, this task is typically handled through GPS signals combined with data from the drone's Inertial Measurement Units (IMUs). Indoors, or in areas where GPS signals are unavailable, this method cannot be used, requiring alternative localization strategies (Famili and Park, 2020).

To address the challenge of UAV navigation in GPS-denied environments, researchers have introduced a range of vision-based navigation techniques (Balamurugan et al., 2016). This approach relies on a mounted camera that scans the surroundings as the vehicle moves through the environment. The images features obtained from this process can then be compared to estimate a motion and find the UAV trajectory (Balamurugan et al., 2016). In general, there are two visual-based techniques that have proven to be reliable:

1. Simultaneous Localization and Mapping (SLAM)
2. Visual Odometry (VO)

SLAM navigation is achieved by incrementally constructing a map and using it as a basis for comparison to localize itself within the environment (Yousif et al., 2015). In most applications, SLAM navigation requires the integration of different sensors that can increase the cost of the solution. These sensors range from infrared sensors (Abrate et al., 2007), to sonars (Kleeman, 2003) and laser scanners (Cole and Newman, 2006).

More recent research has focused on developing visual SLAM (V-SLAM) using cameras to compensate for the high cost of laser scanners. However, the computational cost for processing images and extracting the necessary information increases considerably, compared to laser technology (Yousif et al., 2015).

On the other hand, Visual Odometry works by estimating the egomotion of a vehicle. Pose estimation is achieved by analyzing a sequence of images that are used to translate changes in recognized features into translational and rotational movement with respect to a reference frame (Yousif et al., 2015).

Depending on the device used, feature recognition can be divided into monocular (Campbell et al., 2005) and stereo camera methods (Matthies and Shafer, 2003). Furthermore, they can be divided into three categories depending on the methodology used; feature matching (Talukder et al., 2003), feature tracking (Dornhege and Kleiner, 2006), and optical flow techniques (Zhang et al., 2009).

One disadvantage of visual odometry is that it is subject to cumulative estimation errors over time, similar to those encountered in wheel odometry (Yousif et al., 2015). Nevertheless, empirical studies demonstrate that VO achieves substantially higher accuracy and robustness in long-term localization when compared with wheel-based approaches (Helmick et al., 2004). Furthermore, VO remains unaffected by wheel slippage, a common source of error in odometric systems operating on uneven or low-friction terrain.

Overall, Visual Odometry and SLAM differ in the methodology followed to obtain motion estimation. Visual Odometry emphasizes local accuracy by incrementally estimating pose states at a continuous rate, often applying optimization within a limited time or spatial window (Yousif et al., 2015). SLAM, on the other hand, seeks a trajectory and map that are consistent at a global scale. This is achieved through loop closure, wherein the system identifies previously visited regions and integrates this information to constrain and correct accumulated drift. Figure 1.7 provides a schematic comparison of VO and SLAM frameworks.

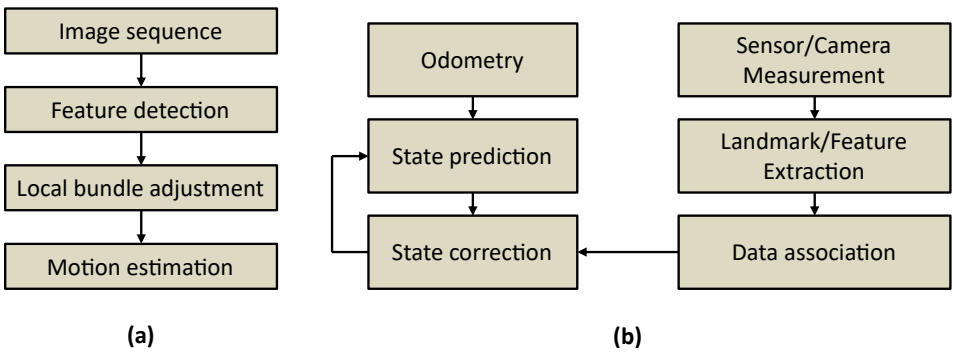


Figure 1.7: Block diagram for motion estimation: a) VO and b) SLAM (Yousif et al., 2015)

Regarding Visual Odometry, reconstruction of features for motion estimation depends on the technology used. VO-based implementations incorporate either stereo or monocular vision. In stereo vision systems, 3D structures can be recovered in a single-time step through triangulation, which leverages feature correspondences between the left and right images separated by a known baseline. Building upon this principle, Stereo Visual Odometry estimates motion by tracking these features across successive stereo pairs, thereby capturing both spatial and temporal relationships. A typical stereo VO pipeline proceeds as a 3D-to-2D motion estimation process, consisting of feature detection, correspondence matching, triangulation, and subsequent motion refinement (Yousif et al., 2015).

In monocular Visual Odometry, reliable reconstruction requires feature points to be tracked across at least three frames: features are first detected in the initial frame, then re-observed and triangulated into 3D points in the second frame, and finally used to estimate the motion transformation in the third frame. A key limitation of monocular VO is the inherent scale ambiguity. Unlike stereo systems, where the relative transformation (rotation and translation) between two camera views can be determined directly from the known baseline, monocular systems cannot fully recover the translation magnitude between the first two frames. As a result, scale is typically initialized with an arbitrary or predefined value, and all subsequent 3D reconstructions and motion estimates remain relative to this chosen scale (Yousif et al., 2015).

Routing

Routing algorithms utilizing drones are mainly tailored to delivery (node routing) and inspection (arc routing) problems (Poikonen and Campbell, 2021). Modeling delivery and inspection problems is usually done through networks represented in a 2D space, regardless the fact that drones move in a three-dimensional space (Poikonen and Campbell, 2021). Network-based drone routing approaches are typically composed of arcs that represent travel paths between nodes. Nodes usually represent drop-off and pick-up locations, charging stations, or parking places.

Current research on drone network routing often relies on simplifying assumptions, such as treating the distance, travel time, or cost between nodes as deterministic values, without explicitly specifying how these quantities are derived. Similarly, the time or cost associated with node related tasks, such as delivery, retrieval, or recharging is typically modeled as fixed, while the underlying processes are not explicitly considered (Cheng et al., 2020). In practice, however, these activities involve varying energy expenditures for operations such as hovering, landing, take-off, and battery recharging, which are seldom captured in existing models.

Keeping in mind that drone performance is characterized by several key attributes, including range, payload capacity, flight speed, energy consumption or battery life, per-

missible package dimensions, and feasible launch and landing locations, models need to represent them accurately. While some of these characteristics are constrained by the physical design of the drone and aerodynamics, others are influenced by environmental operating conditions such as weather, temperature, and elevation (Maghazei and Netland, 2020).

Among those attributes, range, payload, speed, and energy consumption are particularly critical, as they are inherently interdependent: achieving longer flight ranges typically requires either reducing payload weight or increasing onboard energy storage, and both payload and speed directly impact energy usage. For small drones that rely on battery power, this interdependence often results in limited operational range, constraining their use in many practical applications (Poikonen and Campbell, 2021).

Accurate modeling of drone energy consumption is essential for assessing drone performance within industrial time-framed setups. Energy usage is determined not only by the drone's design characteristics but also by the payload it carries and the operational parameters, such as flight speed and trajectory planning (Dorling et al., 2016).

Based on the work of Macrina et al. (2020), drone routing problems can be divided into four categories:

1. Traveling Salesman Problem with Drones (TSP-D)
2. Vehicle Routing Problem with Drones (VRP-D)
3. Drone Delivery Problem (DDP)
4. Carrier Vehicle Problem with Drones (CVP-D)

Originally, Murray and Chu (2015) developed the first variant of the TSP combining the routing of trucks and drones in the same model. In their work, they introduce two new models to solve the problem. First, a Flying Sidekick Traveling Salesman Problem (FSTSP), representing the behavior of multiple deliveries to a set of customers. In this problem, deliveries can be done either by a single truck or a single drone. For the second model, the authors introduced a new concept, enabling deliveries by a fleet of drones, instead of a single vehicle. This problem was named Parallel Drone Scheduling-TSP (PDSTSP).

The main difference between them is that, for the latter, drones and trucks do not cooperate, hence there is no synchronization between them. Drones start and finish any determined route at the depot. Since the formulations presented by Murray and Chu (2015), several authors have developed new algorithms to solve the problems, introducing new extensions and variants to their proposal (Macrina et al., 2020). Table 1.4 summarizes the TSP-D contributions in the literature for the different variants.

Regarding VRP-D algorithms, Wang et al. (2017) initially proposed a model where drones are deployed from trucks to customer locations to deliver parcels. Once the parcel

Table 1.4: TSP-D Contributions (Macrina et al., 2020)

Author	Variant	Mathematical model	Approach	Problem size
Murray and Chu (2015)	FSTSP	yes	heuristic	new: up to 10 customers
Murray and Chu (2015)	PDSTSP	yes	heuristic	new: up to 10 customers
Mourelo Ferrandez et al. (2016)	FSTSP	no	heuristic	new: 100 customers
Ponza (2016)	FSTSP	yes	heuristic	new: up to 200 customers
Marinelli et al. (2018)	FSTSP	yes	heuristic	Bouman et al. (2018)
Carlsson and Song (2018)	FSTSP	no	heuristic	new: up to 500 customers
Agatz et al. (2018)	FSTSP	yes	heuristic	new: up to 100 customers
Mbiadou Saleu et al. (2018)	PDSTSP	yes	heuristic	new: up to 229 customers
Bouman et al. (2018)	FSTSP	no	exact	new: up to 20 customers
Yurek and Ozmutlu (2018)	FSTSP	yes	heuristic/exact	new: up to 20 customers
Tu et al. (2018)	FSTSP	yes	heuristic	Ha et al. (2018)
Chang and Lee (2018)	FSTSP	no	heuristic/exact	new: up to 30 customers
Ha et al. (2018)	FSTSP	yes	heuristic	new: up to 100 customers
Ha et al. (2020)	FSTSP	yes	heuristic	Ha et al. (2018)
Li et al. (2020)	PDSTSP	yes	heuristic	new: up to 20 customers
Poikonen and Campbell (2021)	FSTSP	no	heuristic	new: up to 200 customers
Kim and Moon (2018)	PDSTSP	yes	heuristic	new: up to 80 customers
Jeong et al. (2019)	FSTSP	yes	heuristic	new: 10 customers
De Freitas and Penna (2020)	FSTSP	no	heuristic	Ponza (2016)
Murray and Raj (2020)	FSTSP	yes	heuristic	new: up to 100 customers
Schermer et al. (2019)	PDSTSP	yes		new: up to 50 customers
Dell'Amico et al. (2020)	PDSTSP	yes	heuristic	Murray and Chu (2015)
Salama and Srinivas (2020)	FSTSP	yes	heuristic	new: up to 35 customers
Dayarian et al. (2020)	FSTSP	no	heuristic	new: up to 60 customers
Moshref-Javadi et al. (2020a)	FSTSP	yes	heuristic	new: up to 101 customers
Moshref-Javadi et al. (2020b)	FSTSP	yes	heuristic	new: up to 159 customers
Gonzalez-R et al. (2020)	FSTSP	yes	heuristic	new: up to 250 customers
Agárdi et al. (2019)	FSTSP	yes	heuristic	

is delivered, drones can return to the truck, or wait at the next customer location in case battery level does not allow the full trip. The work follows a worst-case scenario analysis, in which the authors define optimal environments where the use of drones can improve the routing time to complete all deliveries.

Similar to TSP-D problems, with the introduction of VRP-D algorithms, several authors have extended the literature by adding constraints related to battery performance and time-windows variants. Chiang et al. (2019), for example, propose a different approach by analyzing the environmental impact of delivering parcels with and without the use of drones in parallel with trucks. However, implementations using TSP-D and VRP-D model are focused on outdoor applications. Indoor approaches are not considered within these two groups.

Considering scenarios where short delivery times are essential to maintain process performance, DDP approaches have shown to be better adapted to represent the behavior of a fleet of drones tailored for urgent deliveries. In most cases, DDPs incorporate more

realistic considerations compared to other approaches, such as energy consumption, flying range, and battery capacity.

One of the first works on DDPs was carried out by Dorling et al. (2016), developing a MILP formulation considering the weight of the payload carried to calculate battery consumption. The formulation assumes constant speed and batteries are assumed to be replaced each time they reach a pre-defined threshold. Furthermore, battery level decreases linearly based on flight time and payload carried. The model aims to minimize the total distance flown, the amount of batteries replaced, and the total number of drones in the solution.

Contrary to other approaches, DDP literature is relatively low; however, it is clear that it incorporates a stronger emphasis on representing realistic aspects of drone performance. Table 1.5 summarizes contributions in the scientific literature for Drone Delivery Problems.

Table 1.5: DDP Contributions (Macrina et al., 2020)

Author	Mathematical model	Approach	Problem size
Dorling et al. (2016)	yes	heuristic	new: up to 45 customers
Yadav and Narasimhamurthy (2017)	yes	heuristic	new: up to 500 customers
Coelho et al. (2017)	yes	heuristic	new: up to 10 customers
Liu (2019)	yes	heuristic	new: up to 353 customers
Troudi et al. (2018)	yes		new: up to 10 customers

Finally, the Carrier Vehicle Problem with Drones combines the three previous approaches to model a fleet of drones as a delivery system of urgent items that are deployed from slow carriers, such as ships or large vehicles. In CVP-D algorithms, carriers work as a depot, so the range and energy of the drone is restricted to fully complete a round trip to the carrier.

Initially, Mathew et al. (2015) proposed a use case in which, using a large truck as a mobile warehouse, a fleet of drones is in charge of parcel delivery to customers. For this, drones deliver single-orders and fly back to the carrier. In this approach, flight range is the main limitation, so routing is based on efficiently using drone's maximum energy to minimize the total delivery time.

Other authors, such as Boysen et al. (2018) extended the model by assuming that the routing of the truck is fixed and already known. In this case, optimization is focused on the scheduling and deployment of drones to achieve maximum deliveries.

Following the work of Boysen et al. (2018), Poikonen and Golden (2020a) developed another variant in which carrier routing is dynamic and cannot be known in advance.

Considering his scenario, drones can either execute single-order deliveries or multiple ones, where optimization is focused mostly on total payload capacity.

Overall, most existing studies focus on theoretical analyses of drone routing problems and, as a result, frequently lack detailed information regarding the actual technological specifications of drones and the measurement data used in benchmark instances (Macrina et al., 2020). Of the studies listed above, only five incorporate accurate behavioral modeling related to drone performance. As described by Macrina et al. (2020), Table 1.6 shows the contributions that better adapt the mathematical models to specific drone features.

Table 1.6: Drone-feature approach models (Macrina et al., 2020)

Author	Type of drone	Speed	Payload capacity
Dorling et al. (2016)	Hexacopter	6 m/s	up to 3 kg
Jeong et al. (2019)	Octocopter		up to 3.5 kg
Troudi et al. (2018)	Quadcopter	13 m/s	up to 1.2 kg
Mathew et al. (2015)	Quadcopter	8.3 m/s	up to 3 kg
Poikonen and Golden (2020b)	Quadcopter	10 m/s	up to 20 kg

1.5.3 Organizational level

As long as technology is intended to work as an integrated system, understanding of "human fit" rather than only technological considerations should be prioritized (Trist, 1981). This is particularly crucial when disruptive technological change exposes workers to heightened uncertainty and risk (Maghazei et al., 2022). Classical approaches regard man just as an extension of the system, and therefore as a comparable spare part (Trist, 1981). However, current paradigms should include human resources as complement to technological resources; a resource that should be developed and optimized for his own benefit, rather than diminished and discarded (Trist, 1981).

Loch (2017) states that as long as industries intend to focus on process innovation, all the components and interactions between the different actors of the system cannot be fully known in advance. Because of this reason, Knightian uncertainty (not only risk) will always exist between actors and mutual dependencies within the system.

In general, drone technology appears to have a potential misfit when it comes to aligning with organizational standards in the industry (Maghazei et al., 2022). Partly because of the recent development of the technology, as companies do not have organizational structures where drones can be properly integrated. Furthermore, the uncertainty created around the system may generate not only managerial concerns, but also to work-

ers. In some cases, workers could have negative connotations about privacy and safety associated with the technology (Maghazei et al., 2022).

1.6 Synthesis and research gap

The preceding chapter examined the evolution of industrial paradigms and highlighted how the transition from mechanization to digitalization has shaped the development of modern production systems. Within this evolution, the emergence of Industry 4.0 has redefined industrial value creation through the adoption of Information and Digital Technologies (IDTs). Together, these technologies enable the creation of interconnected and intelligent systems capable of transforming both manufacturing processes and supply chain operations.

However, as industries move from mass production toward mass customization, new forms of complexity emerge. High levels of product variability and modularity demand greater flexibility in production and intralogistics operations. The management of dispersed shop floors and the synchronization of internal material flows under uncertain demand conditions represent critical challenges to achieving responsive and cost-effective manufacturing. This limitation has prompted growing interest in alternative intralogistics solutions, among which Unmanned Aerial Vehicles (UAVs), or drones, stand out for their potential to perform agile, single-delivery transport tasks with minimal infrastructure requirements.

Despite this potential, the integration of drones into industrial environments remains in an early stage. Current research emphasizes their technical capabilities but lacks comprehensive frameworks to assess their practical applicability and strategic value in manufacturing systems. Studies addressing drone implementation frequently focus on specific use cases without developing structured approaches to measure feasibility, profitability, or operational performance within complex industrial settings.

Furthermore, while many works address the technological aspects of drones, such as navigation, path planning, routing, and scheduling, these studies often rely on simplified or deterministic models that fail to capture real-world stochastic behaviors, such as fluctuating energy consumption, dynamic obstacle avoidance, or variations in payload and task duration. Consequently, the absence of realistic modeling limits the reliability of current evaluations for industrial deployment.

The state of the current literature reveals a clear research gap: while drones are widely recognized as promising tools for industrial paradigms, there is no unified methodology that enables managers to evaluate their feasibility and profitability within agile manufacturing systems. Specifically, the literature lacks structured frameworks that combine strategic evaluation with operational modeling, integrate stochastic elements reflecting

real-world variability, and account for the mutual adaptation between human and technological systems. Addressing these limitations requires an approach that not only captures the technical and operational aspects of drone deployment but also aligns them with broader organizational strategies and performance objectives.

Building on this, the next chapter focuses on the formalization of the problem within an agile manufacturing context as a DDP algorithm. As stated before, DDP algorithms show the best fit to model realistic drone behavior, and yet they are currently underexplored. The chapter introduces a two-stage optimization model that operationalizes the integration framework by defining drone behavior in terms of path planning, energy constraints, and real-time supervision. This transition from conceptual synthesis to formal modeling establishes the analytical groundwork required to simulate and evaluate drone-based intralogistics systems in subsequent chapters.

Chapter 2

Problem formalization within an agile manufacturing system

At the core of the Industry 4.0 framework discussed in the previous chapter, Cyber-Physical Systems (CPS) serve as the fundamental enablers of smart manufacturing by connecting physical assets with their digital counterparts. In this context, drones can be understood as autonomous cyber-physical agents that interact continuously with both the physical production environment and the digital supervisory layer.

The modeling developed in this chapter builds upon this CPS foundation by formalizing how these agents operate within agile manufacturing systems, translating real-time data, decision-making, and system constraints into a structured optimization framework.

Furthermore, the approach directly addresses several of the integration challenges identified earlier, particularly operational barriers by proposing a mathematical method to ensure safe routing, efficient energy use, and coordinated task assignment within drone-assisted intralogistics.

As stated before, drones are intended to work as a support for existent material handling systems. The integration of this technology aims to decrease response time in case of an unplanned event, thereby enabling continuous production and improving resource utilization. In this context, drones should have full navigation autonomy and manage their own resources efficiently.

Unlike ground vehicles, aerial systems cannot rely on passive protocols when encountering a shared resource. In most cases, hovering situations to force a waiting position should be avoided. If a landing platform or path is occupied, drones should not begin a delivery task until it becomes available. If these events are not properly managed, the risk of battery depletion, and potential crashes, increases significantly.

The integration of drones into agile manufacturing systems should incorporate path-planning algorithms based on conflict-free trajectories. Moreover, system stability and safety should be ensured by a multifunctional online supervisor. The problem can be

formulated as a two-stage optimization model. The first stage involves pre-computing the optimal path based on energy constraints and payload. The result should be a round-trip delivery path, from the pick-up point to the drop-off location and, in most cases, back to the starting position. Once computed, the required paths are reserved exclusively for the drone executing the task.

During the second stage, information on the current battery status is updated and used to recalculate the next mission once the vehicle arrives at a landing location. At this step, the online supervisor determines whether the vehicle can perform another delivery task or should recharge its battery at the nearest charging station.

Since the trajectories in the drone delivery system are envisioned to be free of other vehicles, strategies such as altitude separation can be implemented. The vehicles' obstacle-avoidance systems are designed to function as the final layer of risk management and serve as a communication tool to track their real-time positions, rather than to actively avoid other vehicles.

Figure 2.1 shows the functioning of the supervisor and its interaction with the system.

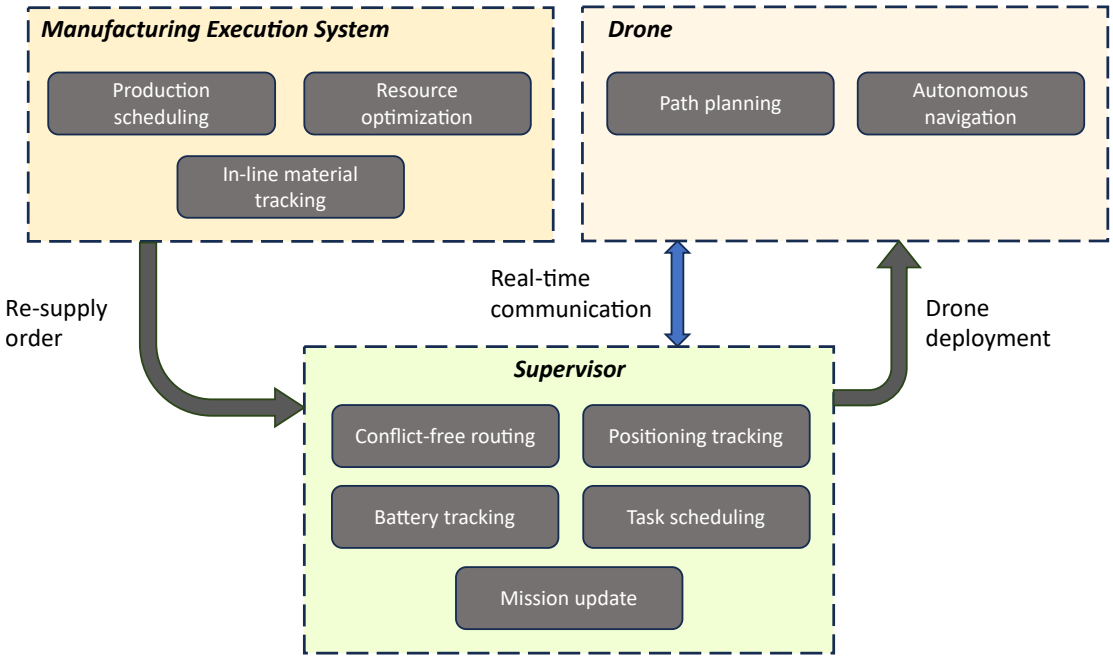


Figure 2.1: Supervisor interaction with the system

2.1 Scenario conceptualization

As production systems continue to evolve and escalate in terms of complexity, there is a need to have a tool that can describe the context around a specific situation. In this sense, a data model can offer a high level of abstraction, providing a logical understanding of

the scenario by clearly identifying the set of entities being part of a real system or meant to become so (Hasan, 2002).

As stated by Hasan (2002), by modeling it is expected to reach four goals:

- Visualize a system within its working situation as it is, or as we would like it to be
- Specify the structure and/or behavior of both, the system and the team working on it
- Provide a framework to guide the construction of a system, taking into account its final use
- Document the decisions taken during the design process and capitalize on the knowledge acquired over time

Following his work, a working situation can be defined by many possible ways depending on the context, however Hasan (2002) elaborated one of the clearest interpretations of the term by saying that it is:

> "...a set of relations between the main subject who produces the effort required for the work and, the production system itself (in the broadest sense), cooperating together to achieve a given performance in carrying out a well-defined mission"

Other researchers, such as Belkadi (2006) go beyond the concept of working situation by first clarifying the pure meaning of what is a situation in an Industrial context:

> "(the situation) is a set of entities and interactions of different natures that characterize the organization as a whole, from the point of view of an actor or an observer, of the external environment in which this actor is involved."

Hence, this definition introduces three fundamental ideas in line with the properties of systems (Belkadi, 2006):

- The situation is part of the actor's external environment. However, it is distinguished from the latter by the fact that it possesses an organized structure.
- This form of organization is relative to a particular point of view, either it is perceived by the actor or an observer, whose aim is to understand or intervene in the elements of the situation or its representation.
- Finally, the situation is the ground-floor where for each given objective, there is a task to accomplish.

Combining the work of both authors, Appendix A shows a UML-based model to describe the expected scenario where a Drone can be used to support intralogistics in Agile Manufacturing Systems. This model identifies the inputs and outputs of the system, as well as the risks and resources related to its implementation. With it, it is expected to provide a starting point that states the needs of the proposed system.

As can be seen, modeling the working situation can help to better define key components within the environment that can pose a potential risk for failure. Identification of actors that affect drone performance helps to optimize system integration since the designing phase, since internal and external variables are considered from the beginning.

Current literature on drone integration in manufacturing settings does not clearly state the interaction, as inputs and outputs, between the aerial vehicle and the manufacturing system, including personnel involved, technological resources, and working areas. With this tool, managers can define clear business models in which implementation is based on maintaining process efficiency while preventing risks.

2.1.1 From conceptual model to mathematical formalization

The conceptual model developed in the previous section provides a comprehensive understanding of the drone-assisted intralogistics scenario, describing the main entities, interactions, and relationships within the manufacturing system. By representing the system's structure through a UML-based approach, it becomes possible to visualize how drones, operators, and production resources coexist and interact in pursuit of specific performance objectives.

However, while such abstraction is essential for defining the scope and logic of the system, it remains descriptive by nature. To quantify performance and support decision-making, this conceptual representation must be translated into a mathematical form capable of capturing operational constraints and optimization goals.

Bridging this conceptual model with a mathematical framework requires formalizing the entities and relationships identified earlier into measurable parameters and decision variables. Each component of the conceptual diagram, such as pick-up and drop-off locations, as well as charging stations corresponds to an element within a mathematical structure.

This translation process involves redefining qualitative relationships (e.g., "interaction" or "dependency") into quantifiable terms such as distances, energy costs, and scheduling priorities. In doing so, the model not only preserves the system's logical structure but also enables its evaluation through optimization techniques that can assess the efficiency and feasibility of drone operations under realistic constraints.

The formalization allows the integration of operational factors such as energy consumption, vehicle routing, and task assignment within a unified optimization framework.

Consequently, the abstract relationships between system entities are transformed into operational decisions, such as which drone should execute a given task, when recharging should occur, and how flight paths should be scheduled to minimize conflicts and total travel distance.

In this context, the next section formalizes the previously defined working situation into a Drone Delivery Problem (DDP) representation. This mathematical representation enables the formulation of an optimization objective, minimizing total distance traveled while considering battery constraints, task priorities, and resource availability. Thus, the conceptual understanding of the intralogistics scenario is translated into a numerical decision-support model, forming the foundation for analyzing drone integration within agile manufacturing environments.

2.2 Problem definition: Drone Delivery Routing Problem

Considering a manufacturing system with $N = \{1, 2, ..., n\}$ drone-enabled landing locations with demand $D_i >= 0$, and $S = \{n+1, n+2, ..., n+s\}$ representing a set of charging stations where drones can land to recharge their batteries automatically for autonomous use, or manually for shorter downtime. $P = \{n + s + 1, n + s + 2, ..., n + s + p\}$ contains the set of drone-enabled pick-up locations, while $I = \{0\} \cup N \cup S \cup P$ contains the set of every drone-enabled location within the problem.

Using this data, a Drone Delivery Problem can be represented on a complete graph $G = (I, A, A')$, where A is the set of arcs $\{(i, j)|i, j \in I \text{ and } i \neq j\}$, and $A'\{(i, r, j)|i, j \in N \cup \{0\}, r \in S \cup P \text{ and } i \neq j\}$. Furthermore, each arc $(i, j) \in A$, and $(i, r, j) \in A'$ is associated with a positive distance d_{ij}, and $\hat{d}_{i,r,j}$, respectively.

In this manufacturing system, drones transport urgent light-weighted items which demand cannot be predicted or scheduled otherwise. To accomplish this, an online supervisor is in charge of managing the incoming tasks, as well as routing the vehicles through conflict-free areas. However, real-time navigation is managed completely by the drone.

Since drones are intended to be integrated as a fast-paced material handling system for high-priority deliveries, task scheduling is assumed to follow a First-In, First-Out (FIFO) strategy. Depending on the location, certain tasks may be assigned higher priority, meaning some locations are served first regardless of the delivery order's arrival time. However, if tasks share the same priority level, they follow FIFO.

As shown in Figure 2.1, drone scheduling and routing are managed externally by the supervisor. Accordingly, scheduling and task assignment are divided into two phases: first, pre-calculating the energy expenditure for the delivery round trip to ensure the

vehicle does not run out of battery; second, assigning the task and reserving resources for the chosen vehicle. Once a task is assigned, landing platforms, such as pick-up, drop-off, and idle/charging locations, along with the routes needed to reach them, are reserved for the assigned vehicle. This ensures that vehicles do not have to wait for a resource to be released and that routes remain unoccupied by other vehicles.

As drones fly over fixed paths and known environments to reduce risks, delivery times are impacted mainly by pick-up and drop-off tasks. For this reason, conflict-free strategies can be implemented without causing significant delays. As shown in Figure 2.2, the supervisor should manage routing to avoid conflict zones around P1, P2, and P3. If Drone B is assigned to Resource N_1 while Drone A is active at Resource N_2 or N_3, Drone B should remain idle until Drone A completes its round trip. However, if different flight altitudes are permitted within drone flight boundaries, Drone B may also begin its task.

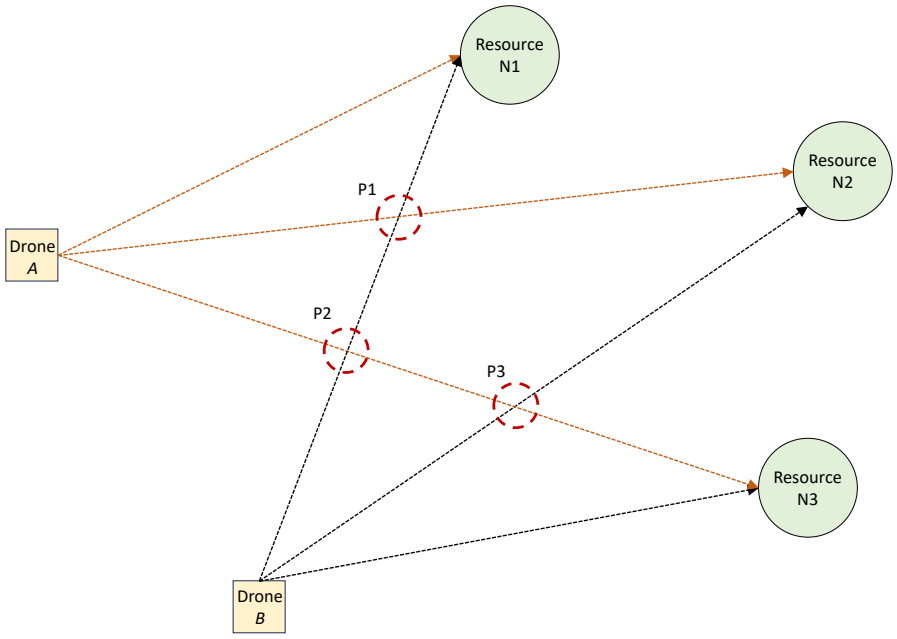


Figure 2.2: Example of a conflict in P1, P2, and P3

Considering that drones are capable of communicating in real-time their position, the supervisor should have the following characteristics:

- Fixed paths for each arc $(i, j) \in A$ and $(i, r, j) \in A'$
- Path differentiation for different heights
- Vehicle battery tracking
- Vehicle position tracking
- Task scheduling following FIFO strategy
- Resource management, including paths and landing platforms

- Item identification (for weight purposes)

Each time a delivery order arrives in the system, the supervisor assigns it to the closest available vehicle to the pick-up location. A return task is then created based on the vehicle's remaining battery after accounting for the energy required for the round trip. If the result is equal to or lower than the safety battery limit upon reaching the drop-off location, a recharge task is assigned; otherwise, the vehicle returns to its starting position.

Concerning routing, the objective of the supervisor is to minimize the total distance traveled in order to optimize energy consumption. When multiple routes can connect two resources and the shortest path is already reserved by another vehicle, the supervisor can assign a longer route if the vehicle has sufficient battery capacity to complete the trip. Mathematically, this can be expressed as follows:

$$\min \sum_{i,j \in N \cup \{0\}} d_{ij} x_{ij} + \sum_{i,j \in N \cup \{0\}} \sum_{r \in P} z_{irj} \hat{d}_{irj} + \sum_{i,j \in N \cup \{0\}} \sum_{r \in S} z_{irj} \hat{d}_{irj} b_r \tag{2.1}$$

Where d_{ij} represents the distance between a pick-up and a drop-off location, while x_{ij} is a binary variable equal to 1 if the arc $(i, j) \in A$ is active, and 0 otherwise. $\hat{d}_{irj}$ represents the total distance between a drone, a pick-up/charging station, and a drop-off location. In this scenario, the drone is located in a station different than the pick-up location, so the delivery trip would involve visiting the pick-up station first. z_{irj} is a binary variable equal to 1 when the arc $(i, r, j) \in A'$ is active, 0 otherwise. Finally, b_r represents the "cost of charging", forcing the vehicle to visit a charging station only when it is needed.

2.2.1 Hypothesis and constraints

Having defined the common notations, the Drone-Delivery Routing Problem is formulated as follows:

Data:

τ = Time in s for landing, (un)loading & taking-off
Q = Payload capacity of the drone in kg
H = Constant value of significant magnitude
$E_{\min}$ = Minimum allowed battery charge in kJ
$E_{\max}$ = Maximum battery charge in kJ
ξ = Battery energy density in kJ/kg
v = Speed of the drone in m/s
q = Battery weight in kg

Decision variables:

x_{ij} = Binary variable
z_{irj} = Binary variable
w_{ij} = Binary variable
d_{ij} = Distance travelled in m from location i to j
$\hat{d}_{irj}$ = Distance travelled in m if a stop r was visited
t_{ij} = Time spent in s to travel from location i to j
y_i = Payload carried (without battery) in kg
f_i = Energy consumed in kJ upon arrival to station i
t_i = Time spent in s upon arrival to station i
u_i = Represents the position of the node i in the tour

The binary variable x equals to 1 if and only if location j is visited immediately after location i, 0 otherwise, for all i, j in the set of $N \cup \{0\}$. On the other hand, the binary variable z_{irj} equals to 1 in the case where a drone flies from location i to a charging/pick-up station r, and then to location j, for all i, j in the set of $N \cup \{0\}$, r in $S \cup P$. Finally, the variable w_{ij} equals to 1 if any route from location i to j exists, whether direct or through an intermediate charging/pick-up station.

Flow constraints:

$$w_{ij} = x_{ij} + \sum_{r \in S \cup P} z_{irj} \qquad \forall i, j \in N \cup \{0\} \tag{2.2}$$

$$\sum_{j \in N} w_{0j} = K \tag{2.3}$$

$$\sum_{j \in N} w_{j0} = K \tag{2.4}$$

$$\sum_{j \in N \cup \{0\}} w_{ji} = 1 \qquad \forall i \in N \tag{2.5}$$

$$\sum_{j \in N \cup \{0\}} w_{ij} = 1 \qquad \forall i \in N \tag{2.6}$$

$$u_i - u_j + n(w_{ij}) = n - 1 \qquad \forall i, j \in N \tag{2.7}$$

Constraint (2.2) indicates that $w_{ij} = 1$ either a drone visits location j directly after location i, or after visiting a charging/pick-up station r. Constraint (2.3) ensures that exactly K drones leave from the depot, while constraint (2.4) forces the same amount of drones to return to the initial location. Constraints (2.5), (2.6) and (2.7) are the classical subtour elimination constraints.

Capacity constraints:

$$Q - y_j - D_j \leq H\,(1 - x_{0j}) \qquad \forall j \in N \tag{2.8}$$

$$y_i - y_j - D_j \leq H\,(1 - x_{ij}) \qquad \forall i, j \in N \tag{2.9}$$

$$Q - y_j - D_j \leq H\,(1 - z_{irj}) \qquad \forall i, j \in N \cup \{0\},\ r \in S \cup P \tag{2.10}$$

$$y_i \geq 0 \qquad \forall i \in I \tag{2.11}$$

Constraints (2.8) and (2.9) track the payload carried by the drone upon arrival at station j without visiting a charging station beforehand. In both cases, constraints (2.8) and (2.9) force y_j to equal the payload carried minus the demand of location j. Conversely, constraint (2.10) tracks the payload when the drone visits a charging or pick-up station between deliveries. The current formulation assumes the drone operates at maximum capacity to maximize deliveries. If only one delivery is allowed, Q equals the package weight. Furthermore, the charging station and pick-up station are co-located on the same node. Thus, after completing a battery charge, the drone can be re-loaded to begin a new delivery.

Energy constraints:

$$f_i - f_j - e(t_{ij}) \leq H\,(1 - x_{ij}) \qquad \forall i \in N, j \in N \cup \{0\} \tag{2.12}$$

$$E_{\max} - f_j - e(t_{0j}) \leq H\,(1 - x_{0j}) \qquad \forall j \in N \tag{2.13}$$

$$E_{\min} - f_i + e(t_{i0}) \leq H\,(1 - x_{i0}) \qquad \forall i \in N \tag{2.14}$$

$$E_{\max} - f_j - e(t_{rj}) \leq H\,(1 - z_{irj}) \qquad \forall j \in N \cup \{0\}, i \in N, r \in S \tag{2.15}$$

$$E_{\min} - f_i + e(t_{ir}) \leq H\,(1 - z_{irj}) \qquad \forall j \in N \cup \{0\}, i \in N, r \in S \tag{2.16}$$

$$f_i - f_j - e(t_{ir}) - e(t_{rj}) \leq H\,(1 - z_{irj}) \qquad \forall i \in N, j \in N \cup \{0\}, r \in P \tag{2.17}$$

Due to the nature of the MILP, the energy consumption function $e(t_{ij})$ should be linearized to fit the model. As shown in the work of Dorling et al.Dorling et al. (2016), the energy consumption of the system can be estimated by applying linear regression based on the payload carried by an n-rotor copter. This estimation assumes that the velocity is approximately linear or at least constrained to a reasonable maximum limit.

With this assumption, a linear regression model can be defined as $p(m) = \alpha m + \beta$, where α represents the energy consumed in kW per kilogram of payload, and m denotes the payload weight in kilograms (including the battery). β represents the energy consumed in kW to keep only the drone frame hovering, assuming that the battery weight and payload are zero.

On the other hand, the energy consumption function $e(t_{ij})$ represents the linearization of the kW consumed in relation with time (in seconds), having that:

$$e(t_{ij}) = p(m_j)(t_{ij})$$

With this in mind, constraint (2.12) tracks the energy spent by a vehicle upon arrival at location j, starting the route at location $i \in N$. In contrast, constraint (2.13) records the total energy consumed after leaving the depot and arriving at location i. Similar to the capacity constraints, constraints (2.12) and (2.13) force f_j to equal the energy consumed along the arc i, j. Constraint (2.14) guarantees that the drone has enough battery to return to the depot. Constraint (2.15) says that, if a drone visits a charging station, then the battery should be reset to its maximum level, while constraint (2.16) ensures that there is enough battery to arrive at a charging station after visiting a drop-off location. Constraint (2.17) tracks the energy the drone has consumed in case it visits a pick-up station before delivering a package.

Time constraints:

$$\tau + d_{0j}/v - t_j \leq H\,(1 - x_{0j}) \qquad \forall j \in N \tag{2.18}$$

$$t_i - t_j + \tau + d_{ij}/v \leq H\,(1 - x_{ij}) \qquad \forall i \in N, j \in N \cup \{0\}, i \neq j \tag{2.19}$$

$$t_i - t_j + \tau + \hat{d}_{irj}/v \leq H\,(1 - z_{irj}) \qquad \forall i \in N, j \in N \cup \{0\}, r \in S \cup P, i \neq j \tag{2.20}$$

Regarding time constraints, constraints (2.18) and (2.19) calculate the time spent traveling to location j. The formulation follows the same principle as the capacity and energy constraints, variable t_j is forced to equal the time consumed between location i and j. Additionally, the variable τ accounts for the time required for landing, loading/unloading, and taking off, depending on the task the drone is performing. Constraint (2.20) calcu-

lates the time spent when the drone visits a pick-up or charging station r in $S \cup P$ before continuing its route to location j.

2.2.2 Model validation and numerical experiments

The assumptions for this work related to the model are as follows: Q is equal to the total payload carried by a drone per trip (in most cases it refers to the total capacity). The depot also functions as a charging station, denoted by $\hat{0}$. The arcs z_{0rj}, $z_{i\hat{0}0}$ and $z_{0\hat{0}j}$ are forbidden. Finally, each time a drone visits a charging station, its battery is fully recharged.

In case the drone needs to supply several times the same location, copies of it are created, and its total demand is split between the copies and the real one based on the capacity of the vehicle. Furthermore, the data taken for the energy consumption are based on a Sentinel Drone by ModalAI (quadrotor), equipped with a battery GensAce 11.1V 60C 3S 5000mAh weighting 336 gr and a maximum payload of 1 kg excluding the battery. Using this data, the values obtained for α and β by applying linear regression in increments of 0.001 kg (from 0 to 1.5) to the energy consumption model are $\alpha = 0.0871$ kW/kg, and $\beta = 0.0397$ kW.

Following the work of Dorling et al. Dorling et al. (2016), the values obtained from the linear approximation were then compared with the exact equation:

$$P(m) = (M + m)^{3/2} \sqrt{\frac{g^3}{2\rho\varsigma n}} \tag{2.21}$$

where:

M = Frame weight of the drone in kg
m = Payload in kg (including battery)
g = gravity in m/s^2
ρ = air density in kg/m^3
ς = rotor disc area in m^2
n = number of rotors

For the present study, the frame weight M of the drone equals to 1.011 kg, while its rotor disc area ς is equal to 0.050671 m^2. Finally, the air density ρ used for the calculation was 1.204. kg/m^3.

Fig.2.3 shows that the linear approximation fits accurately equation (2.21). Based on the lower and upper bounds, being the battery weight and the maximum payload, respectively, the maximum difference between the exact method and the linear approximation is 1.95 W, with a mean percent error of 1.46%.

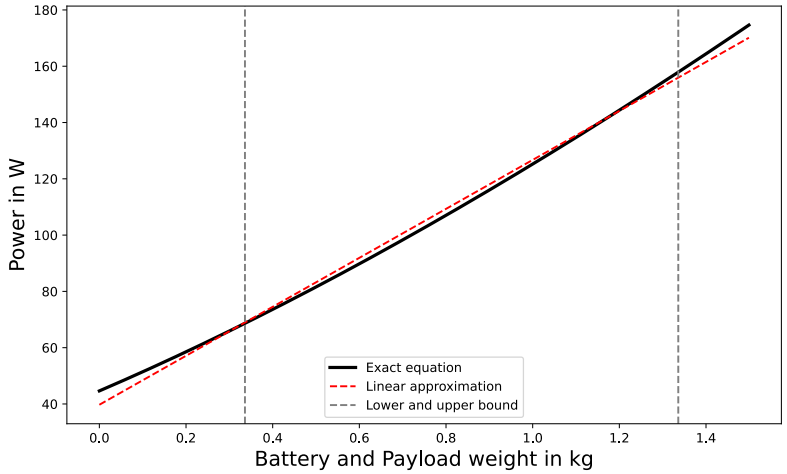


Figure 2.3: Linear approximation of the power consumption model

Similarly, the variables ξ and $E_{\max}$ were calculated, resulting in a value of 594.6428 kJ/kg for ξ and a battery capacity of 199.8 kJ for $E_{\max}$.

Regarding the scenarios, 10 randomly generated instances of 10 uniformly distributed drop-off stations and 3 charging stations which also work as pick-up stations (counting the depot $\hat{0}$ also as a charging station within them) were used for the study, this assuming a grid of 25m X 22.5m. Finally, each delivery was set on standard items of 0.5 kg, having a drone capacity of 1 kg.

2.2.3 Energy consumption

In the following tables, "O. Function" denotes the value obtained after solving the model for the total distance travelled in meters to fulfill all the supply tasks requested for each station. "Energy" indicates the total energy consumed after finishing the tour expressed in kJ.

Besides, K specifies the number of vehicles needed to complete all the deliveries and that were used in the solution model.

Tables 2.1 and 2.2 display the best solution values after running the instances on IBM ILOG CPLEX 22.1.1. All the instances were run using an Intel core i9-12900H PC with 32GB RAM and 2.50 GHz.

The results show that in general, the proposed model optimizes the energy consumption compared with the one of Dorling et al. Dorling et al. (2016) in all the instances tested but the third. In fact, the data obtained suggests that the proposed model in this paper has a higher scalability in terms of real life applications and drone's capabilities due to the adding of stopping nodes for picking-up new items and recharging the depleted

Table 2.1: Proposed model

Instance	$\|N\|$	$\|S\|$	K	O. Function[m]	Energy[kJ]
10n3rU1	10	3	1	116.593	106.068
10n3rU2	10	3	1	111.059	85.987
10n3rU3	10	3	1	114.825	112.105
10n3rU4	10	3	1	94.86	83.976
10n3rU5	10	3	1	87.402	76.692
10n3rU6	10	3	1	106.525	84.889
10n3rU7	10	3	1	102.868	91.583
10n3rU8	10	3	1	72.109	88.654
10n3rU9	10	3	1	90.718	72.480
10n3rU10	10	3	1	123.837	73.790

Table 2.2: DDP model Dorling et al. (2016)

Instance	$\|N\|$	$\|S\|$	K	O. Function[m]	Energy[kJ]
10n1rU1	10	1	1	189	115.833
10n1rU2	10	1	1	166.085	114.320
10n1rU3	10	1	1	137.082	92.706
10n1rU4	10	1	1	238.787	118.146
10n1rU5	10	1	1	225.737	117.971
10n1rU6	10	1	1	174.018	115.137
10n1rU7	10	1	1	263.435	114.031
10n1rU8	10	1	1	139.125	112.804
10n1rU9	10	1	1	169.642	94.757
10n1rU10	10	1	1	204.429	116.039

batteries.

As it can be seen, the distance traveled does not directly increase the energy spent either exponentially or geometrically, which can be explained by the time the drone takes to arrive to the different locations. Although the distances may look disproportional, in terms of time they do not differ greatly. Overall, the main reason for decreasing the energy consumption lies on decreasing the distances the drone flies with full capacity from one pick-up node to a drop-off station, which represents the greatest impact on the final result.

2.3 Conclusion

The integration of drones into agile manufacturing systems presents a promising advancement for material handling, particularly in scenarios that require urgent and unpredictable deliveries. As outlined in the chapter, their effectiveness hinges on the combination of au-

tonomous navigation and a well-structured supervisory framework capable of minimizing risks and ensuring continuous operation.

A central challenge in this integration lies in resource management and conflict resolution. Since drones cannot afford to waste energy in hovering or waiting situations, proactive scheduling and route blocking become critical strategies to safeguard both performance and safety. Path planning based on conflict-free trajectories, complemented by a supervisor that continuously monitors battery levels and task assignments, provides the necessary balance between autonomy and centralized oversight. This layered approach reduces the risk of failures while maximizing the overall system's responsiveness.

The chapter also highlights the importance of modeling the working situation as a foundation for system design. By adopting structured frameworks and scenario conceptualization methods, managers and engineers can identify key actors, resources, and risks from the early stages of implementation. This approach not only supports the technical design of drone operations but also provides clarity regarding the human, organizational, and technological interactions within the system. However, special attention should be given also to human-drone interaction protocols, enabling operators to intervene or override autonomous decisions when safety or efficiency is at stake.

Moreover, the problem definition presented provides a rigorous mathematical representation of the drone-enabled delivery system. Through the introduction of sets, graphs, and constraints, the model captures the complexity of balancing delivery demands, energy consumption, and available infrastructure. This structured representation serves as a roadmap for implementing practical solutions while supporting decision-making processes related to scheduling, routing, and priority management. It underlines the importance of combining theoretical models with real-time supervisory control for reliable drone operations.

Having formalized the problem within an agile manufacturing context, the next chapter moves beyond the basic operational aspects of drone integration to explore the economic and strategic dimensions of adoption. While the mathematical model presented here defines how drones can function as autonomous agents within a Cyber-Physical System, ensuring conflict-free navigation and efficient resource utilization, successful implementation ultimately depends on understanding the organizational readiness and financial feasibility of such systems.

Chapter 3 therefore examines the pre-adoption stage, identifying the economic and strategic factors that influence managerial decision-making to determine whether drone-enabled intralogistics can offer a sustainable competitive advantage under real industrial conditions.

Chapter 3

Pre-adoption: Economic and strategic factors

This chapter focuses on the economic and strategic considerations that influence drone adoption in intralogistics environments, providing insight into the key factors that shape managerial decision-making prior to implementation.

Unlike ground-based technologies, drones take advantage of the relatively open and typically stationary upper spaces within a facility, providing an environment better suited for autonomous navigation (Perez-Grau et al., 2021). For this reason, the attention given to this technology by manufacturing companies has increased considerably in recent years (Deja et al., 2020). However, the equipment required for a robust localization and positioning system often limits the applications to specific scenarios (Perez-Grau et al., 2021).

In warehouse environments, drones are currently used primarily for simple tasks that do not involve interaction with physical objects, such as inventory management and surveillance. However, it is envisioned that their applications will expand to more complex tasks, including intralogistics (Deja et al., 2020). Proper integration of drones as a support for intralogistics could seamlessly replace older technologies, such as conveyors, AGVs, or even cranes.

For this to happen, managers should first have a clear perspective on the limitations of the technology, and understand the specific operational conditions such as spatial layout, energy availability, and system responsiveness under which drones can perform optimally. Following a clear drone adoption framework can ease further integration, as well as set attainable objectives that lead to strategic advantage.

This chapter describes the first steps to consider towards drone adoption. Overall, system performance cannot be evaluated unless realistic features are considered. Parameters such as robot speed, safety distances for collision avoidance (which influence routing), and robot interaction with humans must be integrated in simulation models. Furthermore, it

is described how the stochasticity inherent to the production system can be added into the evaluation phase, so that Key Performance Indicators can reflect more realistic values rather than just patterns based on similar running conditions.

3.1 Human Drone Interaction

Human-Robot Interaction is a discipline focused on understanding, designing, and evaluating robotic systems intended for collaboration with or use by humans (Goodrich and Schultz, 2007). Similarly, Human-Drone Interaction (HDI) can be defined as the field of study dedicated to understanding, designing, and evaluating drone systems for use by or with humans (Tezza and Andujar, 2019).

While interacting with drones, humans play different roles that are directly linked to the UAV's application and level of autonomy. Tezza and Andujar (2019) define three distinct roles based on this relationship: active controller, recipient, and supervisor. An active controller can be understood as a pilot who controls the drone throughout the entire flight. A recipient is a user who benefits from the interaction with the aerial vehicle. For example, if the drone functions as a delivery platform, a recipient would act as a mediator between the drone and its load. Lastly, supervisors are responsible for ensuring the correct functioning of the drone by programming its behavior, supervising the flight, or modifying any components needed to initiate an autonomous task.

It is important to note that different risks may arise depending on the role the human plays during the interaction; therefore, the design of the drone system should focus on identifying potential countermeasures to mitigate these risks.

3.1.1 Evaluation of the Human-Drone ergonomics

As long as humans are still involved in the normal task sequence of a semi-autonomous or fully autonomous vehicle, a collaborative work zone should be designed in a way that does not introduce additional risks to the operator during manipulation. Proia et al. (2022a) propose a RULA assessment to evaluate the exposure of workers to risk factors associated with the upper limbs when interacting with drones. In their work, they use a Design of Experiments (DOE) approach, as in Proia et al. (2022b), to analyze the impact on human posture caused by the interaction between an operator fixed in a standing position and a work volume consisting of equidistant points reachable by the worker.

Proia et al. (2022a) simulated a supply drone that delivers its load directly to an operator while hovering. Based on this scenario, they evaluated the optimal altitude the drone should reach to create an ergonomic work zone for the operator. The experiment parameters were set for a right-handed male with a stature of 1.75 m and a weight of 75

kg, who handles a payload of 0.8 kg.

After solving the DOE, the results showed that the Human-Drone Interaction should be located within the following bounding dimensions (in meters): $X_{min} = 0.35$, $X_{max} = 0.5$; $Y_{min} = -0.2$, $Y_{max} = 0.1$; $Z_{min} = 1.25$, $Z_{max} = 1.35$.

Clearly, one can observe that collaborative work zones must adapt to a predetermined set of body parameters to ensure an optimal position for handling work contents. Nevertheless, country-specific differences and regional requirements must also be taken into account (Rexroth, 2012). For instance, in Germany, the norm DIN 33406 defines the criteria for designing ergonomic workstations, derived from classifying the body heights of the population into four groups. As stated in Rexroth (2012), the optimum working height based on the type of activity is summarized in Table 3.1.

Work contents	Working heights in mm			
	Group 1	Group 2	Group 3	Group 4
High reqs. for Visual inspection Motor skills	1100	1200	1250	1350
Medium reqs. Visual inspection Motor skills	1000	1100	1150	1250
Low reqs. for Visual inspection **High reqs. for** Elbow-room	900	1000	1050	1150

Table 3.1: Working heights per group

Regarding the classification, Group 1 includes the smallest women in the entire population, Group 2 comprises the average women and smallest men, Group 3 contains the largest women and average men, while Group 4 includes only the largest men. Similarly to the work of Proia et al. (2022a), the ergonomics guidebook by Bosch (Rexroth, 2012) suggests that the optimal height for a workstation should be set between 1250 mm and 1350 mm to accommodate most male operators performing highly detailed tasks.

3.1.2 Quadrotor-human safety distance

One fundamental aspect of HDI is ensuring the operator's safety while the aerial vehicle enters the collaborative zone (Proia et al., 2022a). To achieve this, it is crucial to control the robot's velocity once it has reached a defined distance boundary. In this context, the Technical Specification (TS) 15066, published by the International Organization for Standardization (ISO), outlines four distinct methods for determining the safety distance

between a human and a collaborative robot. These guidelines are designed to ensure secure working conditions in various scenarios (Belingardi et al., 2017).

1. SMS: The Safety-rated Monitored Stop depends on an external device or software to stop the robot's operation when a worker approaches, thereby preventing hazardous movement as long as the person is present in the workspace
2. HG: The Hand Guiding method relies on a manually operated device that transmits the motion commands to the robot
3. SSM: The Speed and Separation Monitoring method defines the minimum protective distance between an operator and a robot during interaction in a collaborative work space
4. PFL: The Power Force Limiting method allows contact between an operator and a robot, but it requires controlling the robot's momentum to prevent any injury

Regarding drone applications, SSM approaches are suitable for ensuring that the vehicle operates at a safe distance from any human. In contrast, SMS, HG, and PFL methods are not feasible for an autonomous aerial platform, as they rely on physical contact with the robot and manual activation of commands. Moreover, SSM methods can be adapted to both static and dynamic scenarios. While the static approach assumes constant speeds for both the human and the robot, the dynamic approach accounts for variations in their speeds (Belingardi et al., 2017). Following the work of Marvel and Norcross (2017), a dynamic approach of the SSM can be expressed as follows:

$$S(t_0) \geq \left(\int_{\tau=t_0}^{\tau=t_0+T_R+T_S} v_H(\tau)d\tau \right) + \left(\int_{\tau=t_0}^{\tau=t_0+T_R} v_R(\tau)d\tau \right) + \left(\int_{\tau=t_0+T_R}^{\tau=t_0+T_R+T_S} v_S(\tau)d\tau \right) + (C + Z_S + Z_R) \tag{3.1}$$

V_H and V_R represent the directed speeds of the human and the robot towards each other, respectively. V_S is the directed speed of the robot towards the operator, immediately after detecting any human presence within its surroundings. T_R is the time the robot needs to react to human presence. T_S is the time the robot takes from the initiation of the stop command to the moment when it fully halts motion along the desired axis. C represents the anticipated human reach, in case the operator extends an arm while the robot is moving. Z_R accounts for the uncertainty in the measurements provided by the sensors

regarding the robot's current position. Similarly, Z_S accounts for the uncertainty in the location of any human within the detection radius.

Furthermore, early versions of ISO/TS 15066 proposed a linearized version of equation (3.1), which has since become the standard approach in the literature for determining safety distances in static scenarios (Belingardi et al., 2017).

$$S = (v_H T_R + v_H T_S) + (v_R T_R) + B + (C + Z_R + Z_S) \tag{3.2}$$

Similarly to equation (3.1), equation (3.2) is calculated using the directed speeds of both the robot and the human. However, in the linearized form, the term B replaces v_s, where B represents the total distance the robot traveled from the moment the stop command was initiated until it fully halts its motion.

Other authors, such as Proia et al. (2022a), adapted the previous equation to determine the minimum safe distance between a quadrotor and a human in static conditions, where it is assumed that the quadrotor maintains a constant velocity. In this scenario, the operator is positioned at the center of the work zone while the quadrotor hovers above. The collaborative area begins when the drone starts its descent, limiting the safety distance calculation to the z-axis. Under these conditions, the SSM is reformulated as:

$$S(t_0) = (v_q T_q + v_q T_s) + B + C \tag{3.3}$$

Unlike equation (3.2), the work of Proia et al. (2022a) simplifies the model further to account for the quadrotor's behavior. Here, v_q denotes the quadrotor's velocity within the collaborative workspace, and T_q represents the response time required when human presence is detected. The stopping time of the drone is given by T_s, and the term B is determined as $v_q^2/2a$, where a is the maximum deceleration under worst-case conditions. Similarly, T_s is calculated as v_q/a, with $v_q = \dot{z}$, where $\dot{z}$ represents the quadrotor's maximum speed along the z-axis. Lastly, C corresponds to the safety margin accounting for anticipated human reach during potential intrusion.

3.2 Discrete Event System Simulation

As stated in Chapter 1, there is currently a gap in the literature regarding stochastic approaches to evaluate system performance aimed at determining strategic advantage. To overcome this, this work proposes the utilization of real machine performance data based on an Agile assembly platform. The environment used as a testbed to simulate the consumption of customizable items is based on the FESTO production line installed at the École Centrale de Nantes. This platform replicates an agile assembly line for smartphones and consists of nine assembly stations:

1. In-line Automated Storage and Retrieval System (AS/RS)
2. Measuring application module (INS-1)
3. Two assembly stations for electronics and fuse (ASSY-1/ASSY-2)
4. Labelling module (LM)
5. Camera application module (INS-2)
6. Back cover assembly station (B-ASSY)
7. Press (P)
8. Rework station (RS)
9. AS/RS of boxes (AS/RS-B)

The assembly of the smartphones requires 4 different components, which will be used throughout the whole production processes:

1. Front Cover (FC)
2. Back Cover (BC)
3. Circuit Board (CB)
4. Fuse (F)

In the described system, two types of products are assembled. Each product contains a Bill of Materials, represented by four sublevels: [FC, BC, CB, 1F] for product type A and [FC, BC, CB, 2F] for product type B. The difference between the two is based on the number of fuses installed on the circuit board. The process flow is depicted in Figure 3.1.

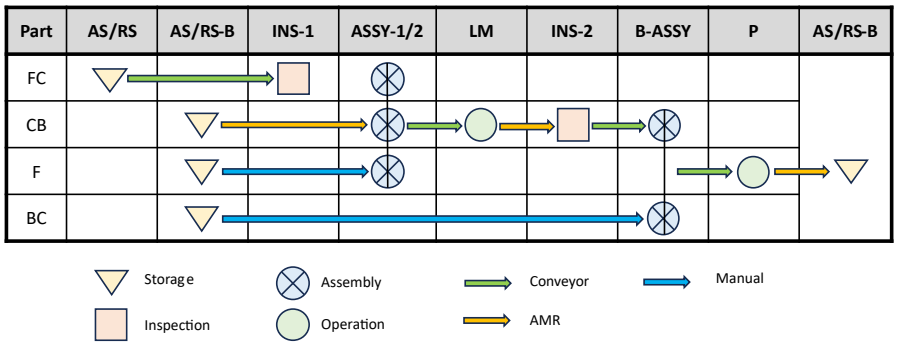


Figure 3.1: Process flow by part

Once the assembly process is finished, the products are sent to the quality area, where autonomous robots inspect the phone's functionality validating three aspects:

1. Physical external components

2. Software

3. Touch screen

If a phone malfunctions, it will be removed from the batch and prepared for rework. If the issue involves an external component or the screen, rework will be performed in the quality area. However, if the phone has a software problem, it will be sent to the engineering room for repair. In summary, the simulated environment consists of five areas: the warehouse, engineering room, maintenance department, production area (with two mirrored assembly lines), and quality area. Figure 3.2 illustrates the layout of the shop floor.

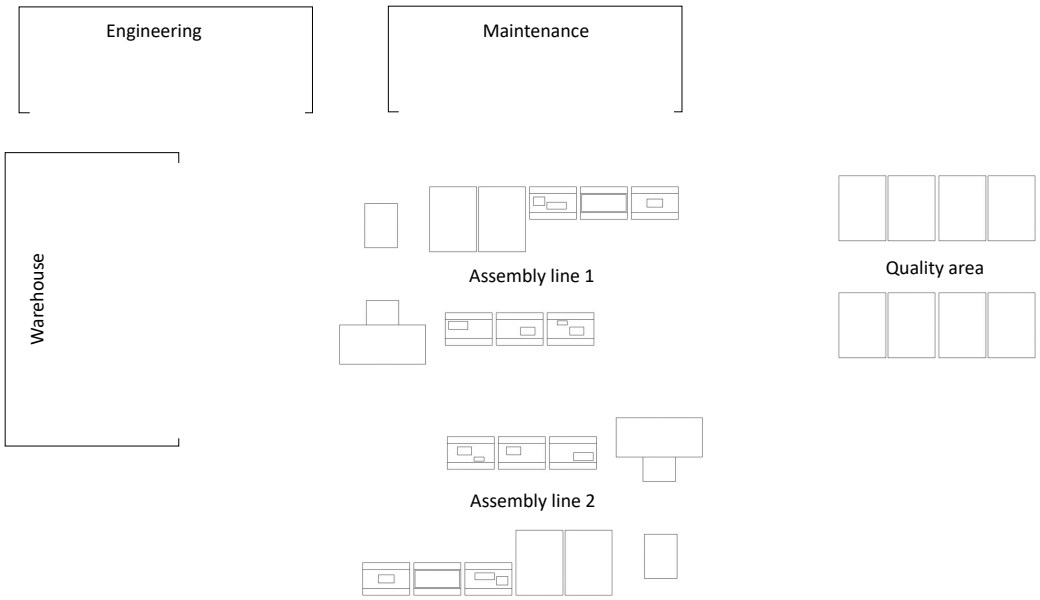


Figure 3.2: Current layout - Batch Production System

The current system operates using a pull strategy, which allows the production line to process a variety of products and produce only what is needed. However, the capacity of the in-line storage system, combined with the scheduling of new item deliveries to the production line, limits the system's reactivity. Since demand must be known in advance to schedule the delivery of raw materials, the system experiences a two-hour delay. This means that changes in demand must also account for this delay.

In other words, dynamic processing to accommodate changes in demand or urgent orders cannot be enabled due to the rigidity of the internal supply system. If demand decreases and schedulers do not fully utilize the production capacity, new orders will be scheduled for the next batch.

3.2.1 Envisioned scenario

The implementation of a highly reactive drone intralogistics system will involve two primary tasks. First, the delivery of urgent raw materials from the warehouse to the production and quality areas; second, the delivery of tools from the maintenance department to the production and quality areas. Additionally, a third drone is envisioned to interface between the quality area and the engineering room, enabling the automated transport of faulty devices.

The updated layout is shown in Figure 3.3. Three landing platforms have been added to the warehouse and maintenance department. Two of these platforms function as charging stations, while the third serves as a pick-up station for items. In the production area, each assembly line will have a landing platform to allow the drone to autonomously drop off items. At the quality area, two drone stations will be available. The first will be a landing platform reserved exclusively for drones coming from the warehouse or maintenance department, while the second will serve as both a charging and pick-up station for the drone transporting devices to the engineering room.

The goal is to increase the reactivity of the supply system and enable more flexible processing. If the stock of a specific item at the back-cover workstation reaches a reschedule point during production, a drone will deliver 5 pieces of the material directly to the line, where an operator will feed the machine once the items have arrived.

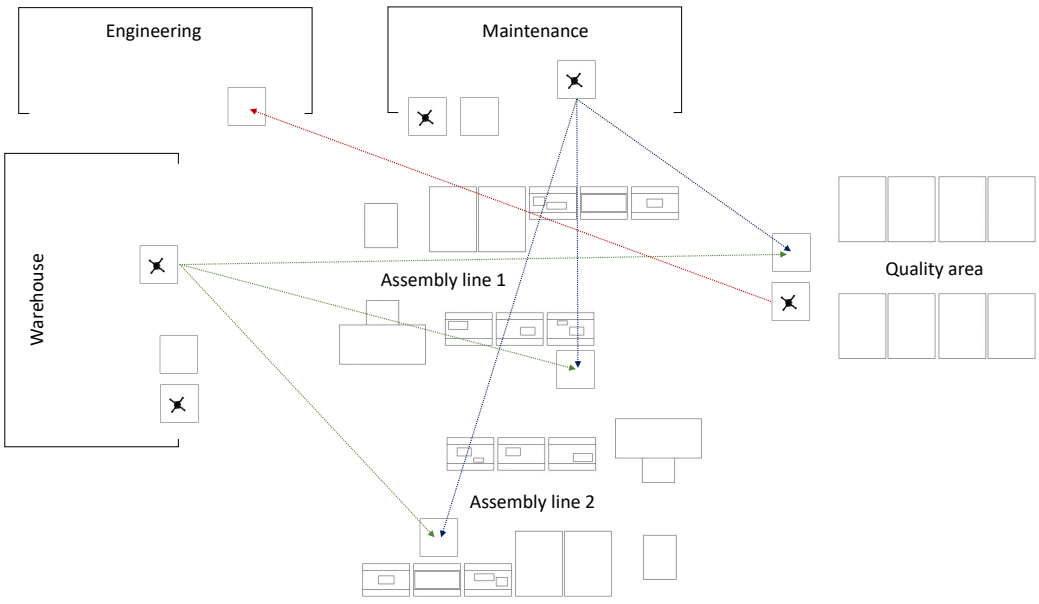


Figure 3.3: Updated layout - Drone integration

3.2.2 Scenario definition

To ensure efficient supply system performance, the evaluation is conducted using a drone platform built on the Sentinel VOXL 2 by ModalAI. The drone operates at a maximum speed of 2.0 m/s, following the most typical speed limit for industrial implementations of autonomous mobile robots and automated guided vehicles (Lin et al., 2021). The vehicle is powered by a GensAce 11.1V 60C 3S 5000mAh battery, which has a total energy capacity of 199.8 kJ. When the battery level drops to 25% of its capacity, the drone completes its final delivery before proceeding to the charging station. Each drone is assigned a dedicated platform for landing and recharging. However, only one vehicle per area can operate at a time, ensuring that a backup drone remains available while the other is charging. On the other hand, the algorithm shown in Fig. 3.4 will be used to prevent collisions and possible delays at shared drop-off locations.

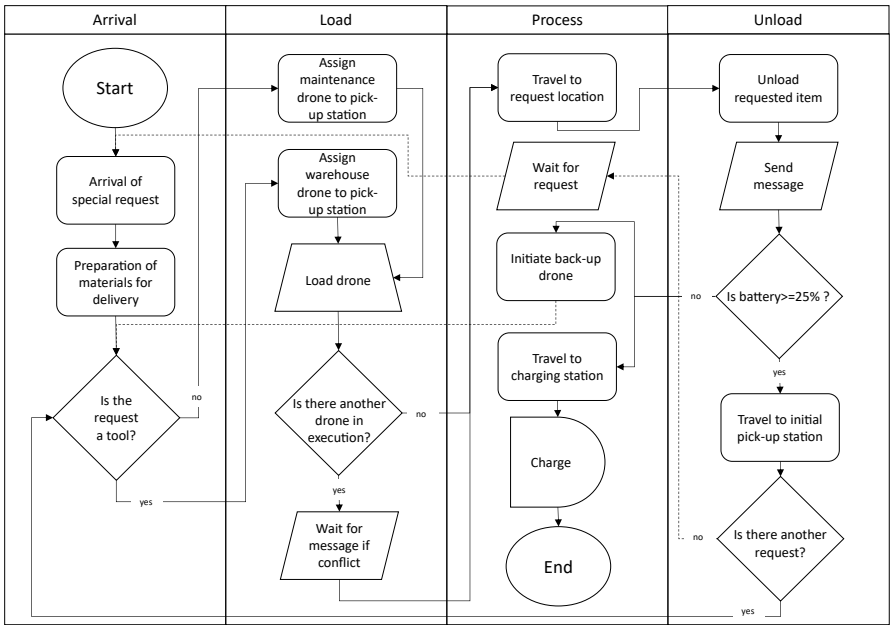


Figure 3.4: Conflict Management Algorithm

In this sense, the scheduling of ground and aerial vehicles differs in a key aspect: for ground vehicles, risk management can be implemented by stopping the vehicle when it encounters an obstacle or an unavailable shared space. In contrast, aerial robots cannot afford prolonged hovering, as this increases the risk of battery depletion. Moreover, extended airtime significantly reduces the drone's efficiency due to the need for frequent recharging, leading to suboptimal scenarios.

To address this issue, a conflict management strategy will be implemented. As part of this approach, drones will send messages whenever they are scheduled to visit a shared resource. Two types of messages will be used: first, a message indicating the target resource when a drone is assigned a task; second, a message notifying that the resource

has become available once the drone departs from the area. In the context of this study, resources refer to landing platforms used by drones during operation. If a resource is occupied and has a pending delivery request in the queue, the request cannot be scheduled until the previous drone has left the area.

In the simulated scenario, four variants will be available at the back cover station. The back covers are automatically fed to the production line by a retrieval system similar to the one used for supplying the front covers. This module has a capacity of 118 items. Each variant will be determined by the work order received at the start of the production line. The production rate for each variant is as follows: [55%, 20%, 15%, 10%].

Each back cover weighs 150 grams, while the weight of the tools will be a random value between 100 and 900 grams. For simplicity, each weight value must be a multiple of 100. In other words, once a tool delivery request is made, the weight of the item to be transported will be randomly selected from the following values: [100, 200, 300, 400, ..., 900] grams.

Regarding the layout, the Euclidean distance between the warehouse and the landing platform of assembly line 1 is 65 meters, while the distance to assembly line 2 is 75 meters. The distance between the warehouse and the landing platform at the quality area is 100 meters. On the other hand, the Euclidean distance between the maintenance department and assembly line 1 is 75 meters, while the distance to assembly line 2 is 105 meters. The distance between the maintenance department and the quality area is 70 meters. Lastly, the distance between the special platform in the quality area and the engineering room is 105 meters. Since human-drone interaction will only occur at the landing platforms, the height of each station will be set to 1250 mm, as defined in Table (3.1).

The algorithm used for obstacle avoidance is based on the A* algorithm. First introduced by Hart et al. (1968), the A* algorithm has become widely adopted in the literature for path-finding applications in 2D environments. Its popularity stems from its ability to find feasible solutions without performing unnecessary calculations, resulting in a fast search process (Tang et al., 2021). However, when applied to complex environments with multiple obstacles, the algorithm may fail to find a solution.

On the other hand, after conducting several tests, the average time for the drone to perform a controlled landing or take-off is 6 seconds. The pick-up and drop-off of items are estimated to take 8 seconds. Once the drone initiates charging, it will recharge at a constant rate of 5 amperes at a wireless station and remain there until fully charged. Upon arrival at the charging station, the backup drone will automatically proceed to the nearest pick-up station and wait for a new request.

3.2.3 FlexSim

FlexSim is a software designed for modeling production processes, with the advantage of creating 3D models to simulate real-life scenarios. It uses advanced statistical analysis to assess throughput and the overall behavior of discrete event systems. This tool will be used to test the conflict management algorithm and assess the efficiency of the delivery system by tracking four Key Performance Indicators (KPIs):

- On time deliveries
- Drone usage [Busy, Idle, Waiting]
- Conflict management [zero conflicts]
- Battery usage

The analysis will offer insights into how a drone delivery system behaves when path planning is focused on conflict management. Issues such as bottlenecks, battery depletion, and obstacle encounters can be highlighted to reveal the limitations of the proposed solution. However, if the system operates smoothly, scheduling algorithms for fleet management could be based on the methodology outlined in Fig. 3.4.

3.2.4 System overview

To gain a clearer understanding of how the system behaves under different conditions, ten instances are run for five distinct scenarios. Each instance is run for 40 hours, with a 2-hour warm-up period. The simulated scenarios differ in four key aspects. First, the probability of producing a non-standard item in the two mirror cells will increase. Second, the inter-arrival rate of production orders will become more frequent, reaching maximum capacity in the final scenario. Additionally, the preventive maintenance schedule for the machines and their breakdowns will vary depending on the scenario. Lastly, random tool requests for the different areas will appear at varying intervals. This setup will allow for an evaluation of the system's efficiency as the demand for urgent deliveries increases. All parameters are summarized in Table 3.2.

Since the aim is to increase production flexibility, priority will be given to the production cells. In cases where there are simultaneous requests from both the production and quality areas, warehouse orders destined for the production cells will be scheduled first. Regarding tool deliveries, technicians can send requests to perform preventive maintenance on the machines, change a broken component, repair a device, etc. Under normal conditions, operators must go to the maintenance department to collect the exact tools and components needed for maintenance tasks. This is because certain items must be

stored under specific conditions, while others are kept there to ensure traceability and prevent misuse.

Furthermore, Statistical Process Control (SPC) of entry items and semi-finished products, as well as rework tasks and machine recalibrations, often require tools and materials that must be preconditioned or calibrated prior to use. Accurately reproducing this behavior in a simulated environment would require an extensive amount of data, which would not significantly contribute to the insights of the current analysis. Therefore, for the purposes of this study, this behavior will be modeled using an exponential distribution, with an average rate that varies depending on the scenario.

	Distribution of special requests			
Scenario	Back-covers distribution[%]	Arrival rate	Breakdowns [minutes]	Tools request
1	[55,20,15,10]	EXP[8]	EXP[900]	EXP[45]
2	[50,25,15,10]	EXP[8]	EXP[900]	EXP[40]
3	[45,25,20,10]	EXP[7]	EXP[900]	EXP[35]
4	[40,25,20,15]	EXP[6]	EXP[600]	EXP[30]
5	[35,30,20,15]	EXP[5]	EXP[600]	EXP[25]

Table 3.2: Simulation parameters

The Mean Time Between Failures (MTBF) is assumed to follow an exponential distribution, with an average rate of 15 or 10 hours depending on the scenario. For both MTBF and tool requests, the exponential distribution was selected due to its memoryless property, as well as its ability to effectively represent how specific disturbances can influence the overall behavior of the system (Balakrishnan, 1996).

According to Proia et al. (2022a), the minimum allowable distance between a quadrotor and a human can be calculated using equation (3.3). This equation is used to determine both the minimum flight height for the drone and the necessary safety distance between the landing platforms and any nearby human presence or obstacle, in order to prevent collision risks. Solving for $v_q = 2.0$ m/s, $T_q = 0.2$ s, $a = 10$ m/s^2, and $C = 0.4$ m, the resulting Safety Separation Margin (SSM) is 1.4 meters.

The SSM method is implemented in FlexSim using the "Proximity Agent Systems" tool, which emulates a proximity sensor operating within a predefined range. This allows for a realistic simulation of the drone's onboard depth cameras, as both the sensor refresh rate and the drone's reaction time to obstacles can be configured to match the parameters of the tested platform. Since human-drone interaction is limited to the landing platforms, the SSM ensures that the drone can safely perform landing and take-off operations only when no operators are within the safety zone.

The battery level is monitored as a tracked variable within FlexSim. The energy consumption rate of the drone is modeled under three distinct conditions: empty flight, loaded flight, and idle. Following the approach proposed by Lara et al. (2024) and Dorling et al. (2016) for quadrotor energy consumption, the energy consumed per kilogram of payload can be expressed as follows:

$$e(t_{ij}) = \frac{p(m_j)(t_{ij})}{1 - (\alpha/\xi)(t_{ij})}$$

where:

$p(m_j)$ = linear regression model $\alpha m_j + \beta$
m_j = payload in kg at location j (with battery)
α = energy consumed in kW per kg of payload
β = kW spent for hovering (w/o battery)
t_{ij} = time in s spent to go from location i to j
ξ = Battery energy density in kJ/kg

This expression is used to calculate the different values for $e(t_{ij})$ depending on the item carried. Table 3.3 shows the results obtained from the linear regression analysis. Using the drone parameters $\alpha = 0.0871$ kW/kg, $\beta = 0.0397$ kW, $\xi = 594.642$ kJ/kg, and t_{ij} equal to one second. m_j ranges from 1.347 kg when performing an empty flight, up to 2.247 kg when carrying a payload of 900 grams.

Regression statistics	
Multiple correlation coefficient	0.99844
Coefficient of determination (R^2)	0.99688
Adjusted R^2 (most likely)	0.99688
Standard error of the estimate	2.10870
Observations	1500

Regression coefficients				
Variable	Coefficient[W]	Std. Error	p-value	95% confidence interval
β (constant)	39.7814	0.1088	≈ 0	[39.5679, 39.9949]
α (independent)	87.1264	0.1257	≈ 0	[86.8798, 87.3731]

Table 3.3: Linear regression analysis

3.2.5 Batch Production System

The batch production scenario is first modeled to serve as a framework for comparison. Since the system follows a pull methodology, production is scheduled according to demand requests. This behavior is simulated by generating random purchase orders, which are batched hourly and supplied with an additional one-hour delay. This delay accounts for the time required by operators to prepare the materials needed to complete the batch. Purchase order entries follow an exponential inter-arrival distribution that varies depending on the scenario. Each order may contain a random quantity between 5 and 20 items, which is represented in FlexSim as duniform(5,20). Orders that cannot be fulfilled due to capacity constraints are carried over and scheduled for the next batch.

	Throughput per Scenario					Utilization rate per Scenario				
Instance	1	2	3	4	5	1	2	3	4	5
1	7342	7342	8267	9502	9502	61.09	61.09	68.82	79.21	79.21
2	6970	6970	8008	9212	9484	58.06	58.09	66.84	77.54	79.13
3	7204	7204	8728	9362	9502	60.78	60.78	72.63	78.03	79.21
4	7396	7396	8454	9314	9418	61.73	61.73	70.55	77.71	78.49
5	7500	7500	8801	9482	9502	62.29	62.29	73.11	79.14	79.21
6	7218	7218	8292	9426	9482	60.15	60.15	69.10	78.66	79.14
7	7300	7300	8376	9410	9502	60.84	60.84	66.59	78.50	79.21
8	7674	7674	8894	9502	9502	66.90	66.90	73.90	79.21	79.21
9	6904	6904	8822	9484	9482	58.03	58.03	73.27	79.13	79.14
10	6782	6782	7800	9192	9502	56.52	56.52	65.12	76.70	79.21

Table 3.4: Throughput and Utilization Rate - Batch Production System

As shown in Table 3.4, the maximum throughput capacity of the production line is reached in scenario 5 and partially in scenario 4. As more production orders enter the system, schedulers no longer need to focus on maximizing the utilization of every production cell. Instead, the production flow can prioritize items that require immediate processing. However, during periods of fluctuating demand—such as lower inter-arrival rates—the efficiency of the production system becomes highly dependent on the intralogistics capacity. Due to current technological limitations, which do not allow for dynamic and high-speed deliveries, managers face challenges in optimizing resource utilization. This rigidity can lead to operational inefficiencies, as machines may not run at full capacity and the workforce may be underemployed.

3.2.6 Drone Delivery System

In the proposed scenario where the drone is integrated, the hourly supply of raw materials to the production line will follow a different strategy. Since schedulers cannot anticipate

variations in demand, scheduling will be based on consumption estimates. For this study, these estimates follow the same pattern as the production orders. Accordingly, the total storage capacity of the back-cover workstation will be allocated based on the back-cover distribution rate shown in Table 3.2. Whenever a custom item needs to be delivered before the arrival of the next scheduled batch, a drone will carry out the task. In this context, the key feature of the proposed supply system is its ability to ensure continuous production.

The reorder point is set at five pieces, allowing the drone to react in time whenever a delivery request is triggered. A higher reorder point would only overload the aerial platform with unnecessary tasks. Table 3.5 presents the throughput for the various simulated scenarios after the drone was integrated into the system. As shown, scenario 5 yields results comparable to those of the batch production system, indicating that the drone can respond efficiently to delivery requests without compromising overall production line performance. In contrast, scenarios 3 and 4 exhibit a significant difference.

	Throughput per Scenario					Utilization rate per Scenario				
Instance	1	2	3	4	5	1	2	3	4	5
1	7520	7588	9204	9300	9502	62.71	63.01	76.71	77.54	79.21
2	7048	7048	8906	9486	9476	58.84	58.84	74.45	79.05	79.21
3	6984	6984	8498	9244	9502	58.21	58.21	70.93	77.15	79.21
4	7398	7398	8344	9440	9502	61.62	61.62	69.55	78.76	79.21
5	7558	7558	8488	9490	9502	63.00	63.00	70.73	79.06	79.21
6	7554	7554	8450	9442	9502	62.96	62.96	70.40	78.33	79.21
7	7606	7606	8684	9504	9502	63.06	63.06	72.36	79.21	79.21
8	7876	7876	8770	9506	9502	65.66	65.66	72.99	79.21	79.21
9	7520	7520	8598	9472	9502	62.71	62.71	71.67	78.95	79.21
10	8060	8060	9204	9508	9502	67.05	67.05	76.71	79.21	79.21

Table 3.5: Throughput and Utilization Rate - Drone Delivery System Scenario

On average, the throughput for scenarios 3 and 4 under the batch production strategy was 8,444 and 9,388 pieces, respectively. In contrast, the proposed production system achieved average throughputs of 8,715 and 9,439 pieces in the same scenarios. A similar pattern was observed in scenarios 1 and 2, where the average throughput of the batch production system was 7,229 pieces, compared to 7,519 pieces for the drone-based implementation. Since both scenarios shared the same inter-arrival rate for production orders, the software simulated identical production environments for each instance, regardless of the scenario.

Tables 3.6 and 3.7 summarize the drone performance during the simulation. As shown in Table 3.6, the drones responsible for supplying raw materials from the warehouse exhibit the highest utilization rates. Although the utilization rates of the maintenance and quality

	Average usage per scenario		
Scenario	Busy	Idle	Charging
Warehouse			
1	14.23%	57.59%	28.18%
2	14.58%	56.66%	28.76%
3	16.61%	51.04%	32.35%
4	17.47%	49.35%	33.18%
5	20.35%	40.74%	38.91%
Maintenance			
1	6.75%	79.12%	14.20%
2	6.46%	79.01%	14.54%
3	7.29%	75.49%	17.21%
4	8.68%	71.92%	19.40%
5	10.47%	67.82%	21.72%
Quality			
1	5.79%	94.22%	–
2	5.85%	94.15%	–
3	5.85%	94.15%	–
4	6.84%	93.16%	–
5	10.33%	89.67%	–

Table 3.6: Drone utilization rate

drones appear relatively low, the objective of the proposed system is to provide a cost-effective tool to respond to demand fluctuations and urgent requests, while minimizing drone airtime. This approach not only helps prevent collisions and conflicts, but also ensures that a drone is always available to complete a delivery task.

Table 3.7 presents the minimum and average battery levels recorded upon landing at the charging station. Although the average battery performance is generally optimal, the minimum values observed indicate that a different strategy may be necessary when battery levels approach the safety threshold. For example, during the first scenario, the maintenance drone landed with only 4.05% of battery remaining, which is an extremely low margin that could significantly increase the risk of failure in the event of an unexpected event. This sharp decline in battery level is partly attributed to the variable weight of the tools being transported. In this context, simulations are valuable for identifying patterns in which such events may occur, thereby enabling the development of targeted protocols to mitigate risk.

Table 3.8 summarizes the performance data of the Drone Delivery System (DDS). First, Table 3.8 presents data from the warehouse drones. As shown, lower inter-arrival rates of production orders lead to a significant increase in raw material requests. Consequently, the average waiting time for an order to be delivered also increases. However, this

Scenario	Battery performance per scenario		
	Min. level	Avg. level	# of charges
Warehouse			
1	8.16%	17.36%	54
2	8.56%	17.58%	55
3	8.16%	17.93%	62
4	7.85%	17.56%	61
5	8.06%	17.68%	74
Maintenance			
1	4.05%	16.45%	26
2	8.11%	16.56%	27
3	7.86%	17.40%	30
4	8.18%	16.85%	35
5	9.01%	17.40%	41

Table 3.7: Battery level at Charging station

	Warehouse		Waiting time			Maintenance	Waiting time		
Station	Requests	Blocked	Min	Max	Avg	Requests	Min	Max	Avg
AL1									
1	126	0.14%	1.14	5.83	1.87	57	1.27	4.65	1.97
2	124	0.08%	1.14	5.89	1.96	59	1.27	5.15	2.05
3	182	0.29%	1.14	6.24	2.09	67	1.27	5.46	2.21
4	220	0.17%	1.14	6.24	2.13	79	1.27	5.59	2.29
5	222	0.16%	1.14	6.02	3.94	100	1.27	6.76	2.69
AL2									
1	140	0.14%	1.21	6.73	2.14	59	1.49	4.50	2.25
2	144	0.10%	1.21	7.04	2.23	61	1.49	4.87	2.38
3	156	0.15%	1.21	9.10	2.67	70	1.49	5.50	1.41
4	142	0.12%	1.21	9.57	2.98	82	1.49	6.75	2.68
5	205	0.14%	1.21	9.13	3.94	95	1.49	9.70	3.22
Quality									
1	136	–	1.5	12.50	2.91	55	1.22	5.97	1.82
2	142	–	1.5	12.99	3.16	58	1.22	5.32	1.95
3	141	–	1.5	19.34	3.85	64	1.22	5.12	2.01
4	143	–	1.5	17.57	4.44	81	1.22	8.50	2.41
5	163	–	1.5	16.51	6.3	92	1.22	7.88	2.61

Table 3.8: DDS Performance Comparison — Warehouse (left) and Maintenance (right)

increase in delivery requests does not significantly impact the time that stations remain blocked due to material feeding delays. The proportion of time that Assembly Line 1 (AL1) and Assembly Line 2 (AL2) are blocked remains relatively stable across all scenar-

ios. This suggests that the delivery system performs consistently, enabling it to respond effectively to demand fluctuations, even when delivery requests are doubled. Nevertheless, if both vehicles need to be recharged simultaneously due to continuous use, the lead time for non-priority areas, as well as some priority deliveries, will be affected.

In fact, the maximum waiting time for a priority request was 9.57 minutes, which occurred in the fourth scenario of Assembly Line 2 (AL2). On average, the waiting time during this simulation was 2.98 minutes, indicating that delayed deliveries were infrequent. However, measures could be implemented to prevent waiting times from exceeding the average. One potential solution would be to manually replace drone batteries, thus minimizing the impact of charging times on scheduling. Since autonomous charging takes an average of 40 minutes, and the drone's total autonomy during continuous flight ranges from 20 to 25 minutes, scenarios involving continuous drone usage would either require larger fleets or reduced charging times.

However, logistics managers should focus on this technology not as a standalone delivery platform, but as a support for existing technologies during decision-making. Increasing the drone fleet in a similar scenario as the one presented in this paper would not be an optimal solution. Since the combined autonomy of the aerial delivery system could go from 40 to 50 minutes during continuous flight, other strategies can be taken to fulfill the 10-20 minutes gap before scheduled standard deliveries arrive at the production line.

Similarly, Table 3.8 presents the supply performance of the two drones located in the maintenance area. In contrast to warehouse requests, tool demands are significantly lower, resulting in more stable delivery times. The most notable difference is in the maximum waiting time observed at the Quality station. As shown in the layout depicted in Fig. 3.3, maintenance drones can avoid conflicts at the quality area's landing platform, even when the drone interfacing with the engineering room is in operation. This also holds true when warehouse drones are supplying either assembly line 1 or 2.

As a means of comparison, the walkable distances between the warehouse and the three shopfloor areas are as follows: 88 meters to Assembly Line 1, 109 meters to Assembly Line 2, and 133 meters to the Quality area. Assuming a constant walking speed of 1.3 m/s for an operator, the round-trip walking times, excluding any additional tasks, would be approximately 2.26 minutes to Assembly Line 1, 2.79 minutes to Assembly Line 2, and 3.41 minutes to the Quality area. Even though the drone, on average, performs better—considering both loading and unloading tasks already—the total time an operator would spend walking during the accumulated delivery requests in Scenario 5 (as shown in Table 3.8) would amount to 27.16 hours over a 40-hour production period.

However, since the proposed delivery system introduces supply tasks from the warehouse that are not present in a batch production scenario, a more accurate way to assess time savings is to compare only tasks common to both production strategies. Applying

the same analysis to tool deliveries, the walkable distance from the maintenance area to Assembly Line 1 is 118 meters, while the distance to Assembly Line 2 is 133 meters. The distance between the maintenance area and the Quality area is 97.3 meters. Assuming an operator walking speed of 1.3 m/s, the corresponding round-trip times are 3.02, 3.41, and 2.49 minutes, respectively. Based on the data from Scenario 5 in Table 3.8, the total walking time required to fulfill all tool delivery requests over a 40-hour production schedule amounts to 14.27 hours.

This analysis provides valuable insights for managers aiming to improve operations, as material transport is a non-value-added activity and is classified as one of the seven wastes in Lean Manufacturing philosophy (Niemann et al., 2021). On the other hand, a fast response to tool demands also increases machine capacity. In scenarios where tooling is required before a batch change, having the right items available on time can significantly reduce machine downtime. Furthermore, using an autonomous delivery robot, such as a drone, increases operator efficiency, as they can focus on tasks essential to maintaining continuous production.

3.3 Economic benefits

The simulation results presented above demonstrate the operational gains of the drone system. However, the decision to adopt this technology also requires an economic evaluation that takes into account demand uncertainty and the value of flexibility.

In the context of this study, production flexibility can be understood as the ability to adapt the process to changes in demand. As stated before, drones are envisioned to support manufacturing systems, rather than changing the whole structure and creating a new production system.

As it can be seen from the results, batch production systems do not fail when demand is constant and utilizes the full capacity of the system. However, scenarios where demand varies from one batch to another, and machine capacity cannot be optimized, the system cannot properly handle those sudden changes, hence resources are misused.

Given this, system flexibility would then be the ability the process has to change from one mode of operation to another. In fact, economic advantage of flexibility should be evaluated as the value provided when different production modes are used to better adapt to demand uncertainty. However, studies have shown that companies tend to favor other indicators for decision-making to invest in flexible technologies, such as throughput, quality, data management, and reliability (Adler, 1985).

Typically, researchers have addressed the economic value of flexibility in terms of its “option” value (McDonald and Siegel, 1985). So flexibility can be seen as financial options, which are contracts that provide the opportunity to trade one asset for another

under predetermined conditions, even though the asset's value fluctuates unpredictably over time.

Considering a call option on a stock: it grants the holder the right, but not the obligation, to buy the stock at a set price, known as the exercise price, on or before the option's expiration date. In practice, this means the option holder can decide at any point before expiration whether to pay the exercise price in exchange for the stock, thereby securing ownership at that fixed cost regardless of the stock's market value (Mason and Merton, 2019).

Following this definition, the value of an option increases with the level of uncertainty surrounding the underlying variable. If the outcome is known with certainty and expected to follow its normal behavior, the option holds little value, as there is no potential for unexpected gain. However, when the underlying variable is uncertain and may deviate significantly from its normal level, the option gains value. The greater the uncertainty, the higher the probability that the option will yield a profit, and thus the more valuable it becomes.

Extending the analogy, it can be assumed a flexible production system that can operate in two distinct modes, A and B. The firm currently runs in mode A but has the ability to switch instantly and without cost to mode B. At each decision point, the firm compares the profitability of both modes: if mode B generates higher returns, it switches; otherwise, it remains in mode A. When switching entails costs, however, the decision becomes more complex. The firm must then weigh the immediate gain from changing modes against the long-term impact that choice will have on future production opportunities. This dynamic gives rise to a chain of interdependent, or "nested" options, whose value can be assessed using compound-option valuation techniques (Kester, 1984).

Building upon the work of Kulatilaka (2002), a flexible manufacturing system can be represented as one that supports M distinct modes of operation. Each operational mode is associated with a corresponding system *state*, which is determined by the realization of one or more stochastic variables (θ) (being the total utilization rate due to demand in this study). For a given state (θ_k), the profit (or cost) generated by mode i within a unit time interval is denoted by $\pi_i(\theta_k)$. Transitions between modes i and j are feasible but incur a switching cost δ_{ij}. These costs may arise from sources such as equipment reconfiguration, workforce retraining, production downtime, inventory adjustments, or compensatory payments.

The stochastic environment is characterized by the evolution of θ, which is modeled as a Markov process. Let θ_t take values from a finite set of N discrete states, $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$. The probability of transitioning from state θ_i at time t to state θ_j at time $t+1$ is denoted by $p_{ij} = \Pr(\theta_{t+1} = \theta_j \mid \theta_t = \theta_i)$.

The stochastic evolution of θ is influenced by the level of variability in the system. We

define three scenarios: *low variability*, *medium variability*, and *high variability*. For each scenario, the transition probabilities are constructed using a truncated discrete normal distribution centered at the current state:

$$p_{ij} \propto \exp\left(-\frac{(\theta_j - \theta_i)^2}{2\sigma^2} \right), \quad \theta_j \in [\theta_{\min}, \theta_{\max}], \tag{3.4}$$

where σ controls the degree of variability and $[\theta_{\min}, \theta_{\max}]$ denotes the scenario-specific range of θ. Low variability corresponds to small σ, restricting transitions to neighboring states, whereas high variability allows for transitions across the entire state space.

Consider a flexible system with a finite lifetime of T discrete periods. Let the system operate in mode $m \in 1, \ldots, M$ at the terminal period. Time is indexed such that period 1 begins at $t = 0$, and consequently, the final period begins at $t = T - 1$. At this terminal stage, the firm observes the realization of the random state $\theta_{T-1} = j$ and obtains a profit $\pi^m(\theta_{T-1})$. The discrete-time formulation can be viewed either as an approximation to an underlying continuous-time decision process or as an exogenous time structure imposed by contractual or organizational constraints.

With only one period remaining at $t = T - 1$, the value of the system is deterministic and equals the maximum attainable profit across all operational modes after accounting for switching costs. Formally,

$$F(T-1, m) = \max_{i \in 1,\ldots,M} \left[\pi^i(\theta_{T-1}) - d_i \right] \tag{3.5}$$

where the adjustment term d_i equals zero if $i = m$ (continuing in the current mode), and equals the switching cost δ_{mi} if $i \neq m$.

3.3.1 Calibration of the economic model

Based on the results of section 3.2.5 and 3.2.6, the impact of demand variability on production performance can be assumed to be higher when using a batch production strategy. However, further simulations are needed to validate this hypothesis.

Using similar parameters as in Scenario 3, section 3.2.4, an extended analysis aimed to tracking the production performance based on demand variability is presented in Table 3.9.

For the current analysis, system demand is tracked per hour and linked to the system throughput, replicating the same behavior for both production strategies. Total capacity per hour is assumed to be 119 units. Standard deviation is then calculated using the added demand of the four product types per hour and run during 40 hours. Figure 3.5 shows the variance analysis of the first run.

As it can bee seen from Figure 3.5, production demand varies considerably from one hour to another, making it harder to follow a fixed production batch that would not create

				Performance	
Run	Mean	Std deviation	Demand/Capacity	Batch	DDP
1	105.42	37.04	88.76%	96.06%	95.56%
2	102.50	31.13	86.30%	90.86%	94.99%
3	108.05	43.33	90.97%	98.10%	97.68%
4	103.85	34.95	87.43%	95.81%	95.20%
5	113.63	45.11	95.66%	99.68%	99.36%
6	102.78	42.78	86.53%	91.23%	95.01%
7	100.55	32.27	84.66%	90.84%	96.12%
8	101.48	33.82	85.43%	90.92%	95.70%
9	100.58	37.45	84.68%	90.39%	96.08%
10	106.95	44.06	90.04%	98.10%	97.68%

Table 3.9: System performance compared to total demand

an overstock of one item type or conversely, an underproduction of another.

Similarly, further simulations are run to compare the behavior of a more stable demand. For the comparison, a different statistical distribution is used for the inter-arrival rate of orders. The new rate is defined in FlexSim as uniform(3,12) minutes, being 3 the minimum arrival rate and 12 the maximum. Furthermore, batches are adapted in each scenario to follow the demand behavior, while the drone delivery system behaves as described in the previous section; adapting to variations in demand when needed. Table 3.10 shows the results of the first 10 runs.

				Performance	
Run	Mean	Std deviation	Demand/Capacity	Batch	DDP
1	107.58	10.70	90.57%	98.94%	96.61%
2	105.95	13.41	89.20%	97.88%	97.03%
3	115.93	11.81	97.60%	99.50%	98.91%
4	97.25	14.97	81.88%	95.41%	95.69%
5	100.07	11.99	84.26%	96.27%	97.12%
6	100.7	14.34	84.78%	96.30%	96.39%
7	96.15	9.33	80.95%	92.4%	98.5%
8	105.15	15.02	88.53%	96.44%	96.60%
9	107.25	14.09	90.30%	97.98%	97.54%
10	100.56	15.2	84.68%	96.57%	96.18%

Table 3.10: System performance with stable demand

For each scenario, stable and highly variable demand, 3 sets of 30 runs are simulated to account for a wider range of total demand. For the second and third set, the inter-arrival rate of orders is delayed, so that the impact of lower demand can also be analysed. As it

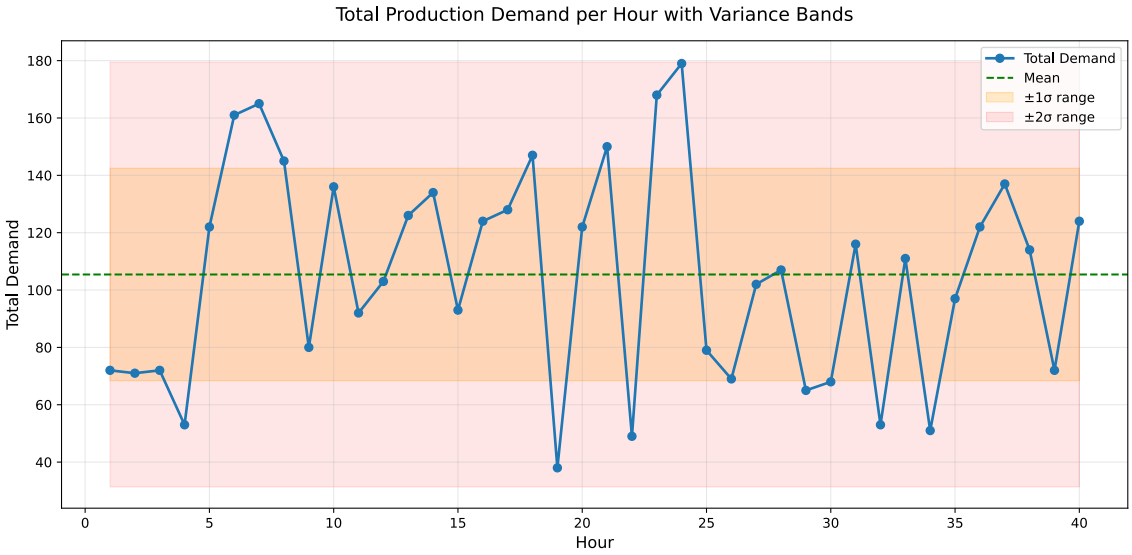


Figure 3.5: Standard deviation of demand per hour

can be seen from the results, when variability remains low, demand can be estimated with higher accuracy, hence the benefits of a flexible material handling system do not have a huge impact on overall production performance. However, scenarios where demand variability do not fit within a normal distribution, estimations over the batch sizes may lead to over production without satisfying demand, or to a decrease in the overall performance.

Performance results for the different values of total demand/capacity (θ) when variability is high (std deviation $\geq 30\%$ compared to mean) are summarized in Figure 3.6. As demand decreases and variability between orders remains high, batch production systems underperform compared to the proposed system using drones. Since total capacity is not used continuously due to demand variation

As shown in Figure 3.6, when demand decreases to at least 87% of system capacity, batch production strategies cannot adapt fast enough to follow the variability of the process to optimise utilisation. Above this threshold, and under conditions of variable but high demand, the batch production strategy appears more or equally reliable compared to the DDP, since deliveries are not based on urgent protocols. Unlike the drone system, batch deliveries are known in advance and can be planned according to the next resupply schedule. In contrast, the drone system may become overloaded, where even small delivery delays can reduce overall efficiency.

3.3.2 Operational modes and profit functions

At earlier stages ($t < T-1$), the system's value incorporates the uncertainty about future realizations of (θ). For instance, at $t = T-2$, suppose the system is operating in mode m. The value at this stage reflects the maximum net profit achievable in period $T-2$ (profits

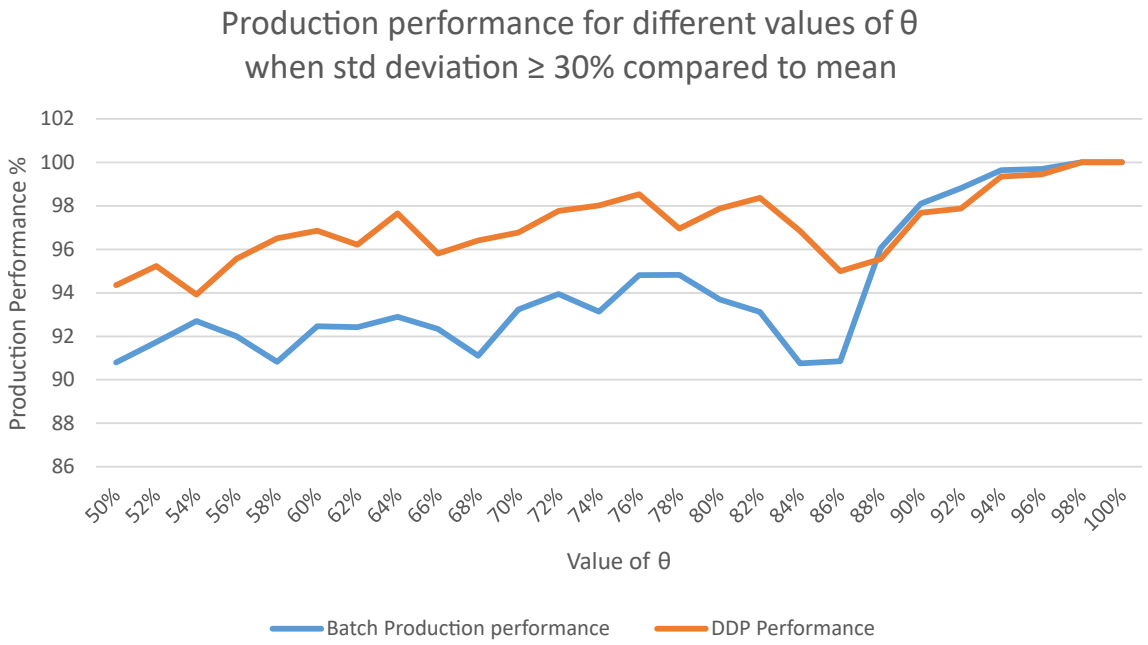


Figure 3.6: Production performance when demand/(total capacity) varies significantly and standard deviation $\geq 30\%$ of system mean

less switching costs), plus the expected continuation value at $t = T - 1$, conditional on the information available at $t = T - 2$.

Following this idea, a firm operating under a batch production strategy as in this study can be defined as static at time T unless it invests in a new technology. Prior to undertaking the initial investment, the firm is characterized as being in a "waiting" state, which we denote as mode 1. Transitioning out of this state requires the payment of an irreversible investment cost I, after which the firm gains access to two productive configurations, labeled as modes 2 and 3. While engaged in either productive mode, the system retains the flexibility to switch between modes 2 and 3. This means that the operational structure can be represented by three distinct modes:

Mode	Description	Profit
1	Waiting to invest	0
2	Batch production system	$\pi^2(\theta)$
3	FMS with drone	$\pi^3(\theta)$

For the current study, the profit functions associated with the two productive modes are assumed to be linear in the state variable (θ). Under this specification, the optimal switching decision reduces to a direct comparison of the net profits across modes, inclusive of any switching costs.

Condition	Decision
$\theta_{T-1} \leq \theta^*_{12}$	Do not invest
$\theta^*_{12} \leq \theta_{T-1} \leq \theta^*_{13}$	invest and operate in mode 2
$\theta^*_{13} \leq \theta_{T-1}$	invest and operate in mode 3

For the numerical analysis, the profit functions corresponding to the two productive modes are specified as linear functions of the state variable (θ):

$$\pi^2(\theta) = a_2 + b_2\theta = -1 + 2\theta$$

$$\pi^3(\theta) = a_3 + b_3\theta = -0.13 + \theta.$$

The parameterization is chosen so that when $\theta < 0.87$, mode 3 yields higher profits than mode 2, whereas for $\theta > 0.87$ the reverse holds. These values are based on the results of the batch production and DDP performance when variation is high, shown in Figure 3.6. Moreover, the linearization of the profit function is chosen for simplicity, since it can be concluded that lower values of θ would not yield higher profits when using a batch production strategy under 87% capacity.

3.3.3 Model validation

To validate the linearization and the parameterization of the profit functions introduced in Section 3.3.2, simulation results were used to estimate piecewise-linear approximations for the two productive configurations (batch production and FMS with drone delivery).

Using the dataset of 523 simulated observations, profits for each mode were regressed against the normalized demand/capacity ratio (θ). The resulting piecewise linear fits, shown in Figure 3.7, demonstrate that both relationships are well captured by a two-segment specification, confirming that the assumption of local linearity across demand regimes is appropriate for this system.

The estimated breakpoints for the batch production system (Mode 2) were $[0.84, 0.879, 1.02]$, while for the flexible drone-assisted system (Mode 3) were $[0.84, 0.937, 1.02]$. The respective coefficients of determination ($R^2 = 0.904$ and $R^2 = 0.853$) indicate a strong explanatory capacity, supporting the validity of the linear functional form used in the theoretical model. Results are summarized in Table 3.11.

As seen from Figure 3.7, the black dotted-intersection between the two fitted profit functions defines the empirical investment threshold (θ^*), representing the point at which both production strategies yield equivalent expected profit.

For the full sample, this threshold was estimated as $\theta^* = 0.944$, implying that when system utilization exceeds approximately 94% of capacity, both production systems be-

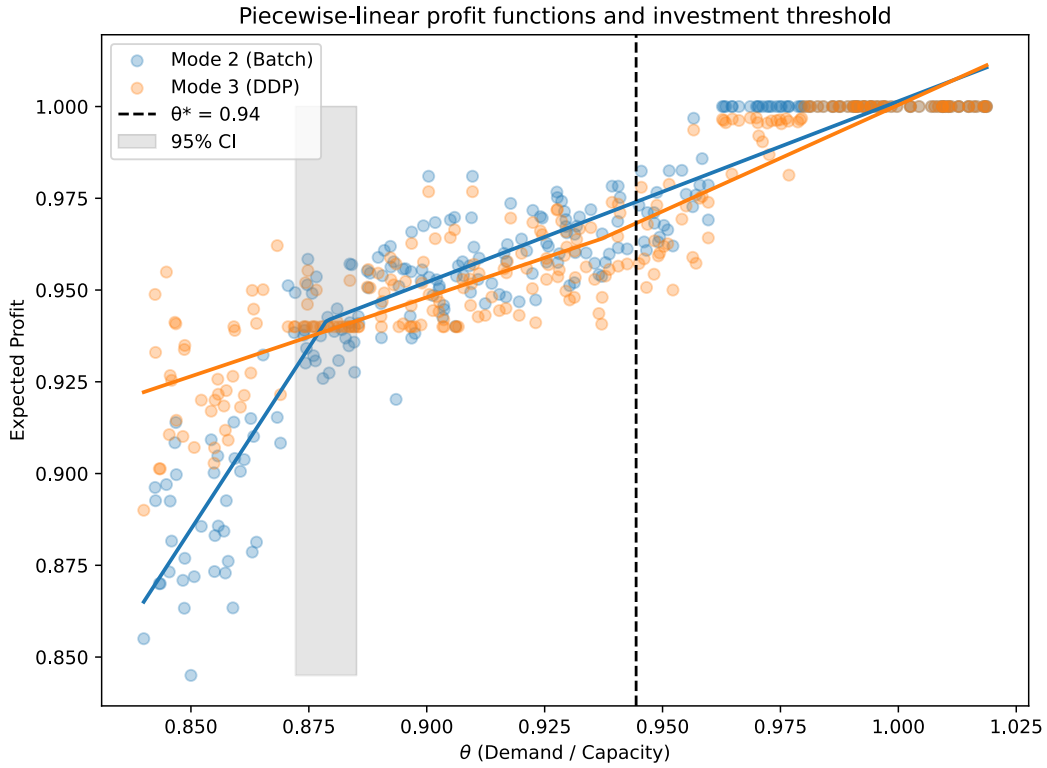


Figure 3.7: Piecewise-linear fit of profit functions

Parameter	Batch (Mode 2)	DDP (Mode 3)
Breakpoint	[0.84, 0.8787]	[0.84, 0.9370]
R^2	0.904	0.853
Estimated θ^*	0.944	0.944

Table 3.11: Piecewise-linear analysis - parameter results

have similarly and constant. Below this threshold, the additional responsiveness of the drone system begins to provide a comparative advantage in accommodating variable demand.

To assess the robustness of this result, a nonparametric bootstrap with 500 resamples was conducted. For each iteration, both mode-specific piecewise regressions were re-estimated, and the intersection point $\theta*$ recalculated.

The bootstrap distribution of $\theta*$ (Figure 3.8) was tightly concentrated around a mean of 0.878, with a 95% confidence interval of [0.872, 0.886]. The narrow dispersion indicates high stability of the estimated switching threshold, and confirms that the apparent deviation of the full-sample estimate (0.944) is attributable to sampling variability rather than structural model misspecification.

Overall, the statistical analysis corroborates the hypothesized relationship between

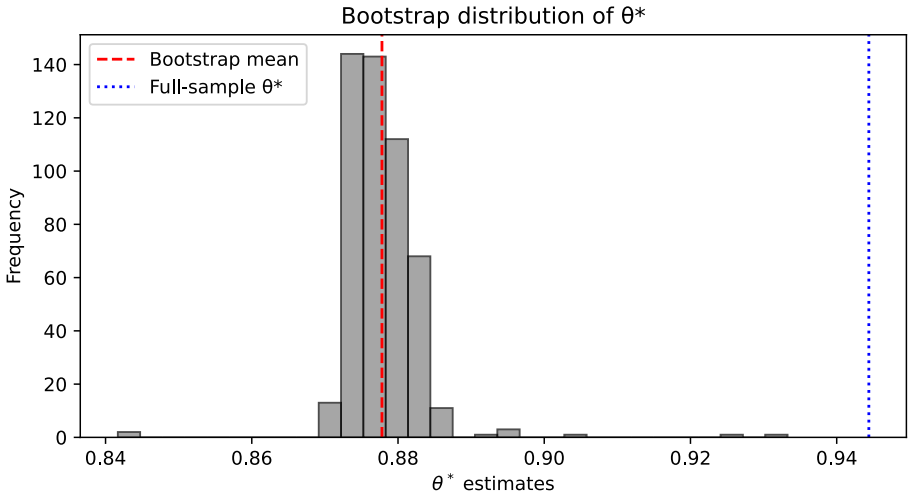


Figure 3.8: Bootstrap distribution of estimated switching thresholds

system utilization and profitability. The validated threshold range (0.87–0.89) aligns closely with the theoretical benchmark derived from the performance simulations, reinforcing the interpretation that at utilization levels below 87%, the flexibility of the drone-enabled FMS provides superior performance under high demand variability.

These findings support both the linear parameterization adopted in the analytical model and the economic rationale underlying the investment and switching conditions described earlier.

3.3.4 Dynamic Programming formulation

Let $V(t, \theta_i, m)$ denote the value of the system at time t when the state is θ_i and the system operates in mode $m \in \{1, 2, 3\}$. At the terminal period $t = T - 1$, the value function reduces to the maximum net profit achievable:

$$V(T-1, \theta_i, m) = \max_{j \in \{1,2,3\}} \left[\pi^j(\theta_i) - d_{mj}\right], \tag{3.6}$$

where $d_{mj} = 0$ if $m = j$ and $d_{mj} = \delta_{mj}$ otherwise.

For earlier periods ($t < T - 1$), the value function incorporates the expected continuation value under uncertainty:

$$V(t, \theta_i, m) = \max_{j \in \{1,2,3\}} \Big[\pi^j(\theta_i) - d_{mj} + \mathbb{E}[V(t+1, \theta_{t+1}, j) \mid \theta_t = \theta_i]\Big], \tag{3.7}$$

where the expectation is computed with respect to the transition probabilities p_{ij}. This recursive formulation allows the determination of optimal switching decisions at each time step.

Based on the results presented in Figure 3.7, the value of flexibility of the system evaluated is shown in Figure 3.9

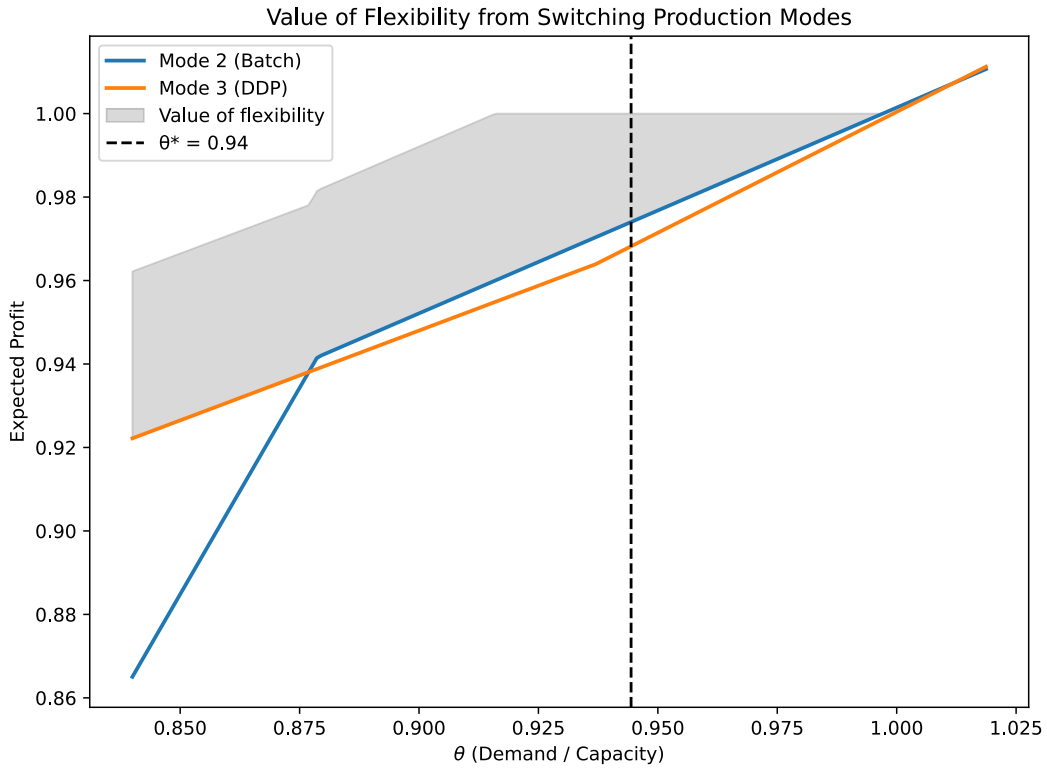


Figure 3.9: Value of flexibility

3.3.5 Numerical results

To examine the system's dynamic properties, we set the project lifetime to $T = 50$ discrete periods. The associated dynamic programming problem is then solved to determine the optimal switching thresholds and the value functions for each period.

Given an initial state θ_0, the evolution of θ under each variability scenario can be simulated using the transition matrix P. At each time step, the next state is sampled according to:

$$\theta_{t+1} \sim \Pr(\theta_{t+1} = \theta_j \mid \theta_t = \theta_i) = P[i, j]. \tag{3.8}$$

This procedure enables the analysis of stochastic trajectories and the evaluation of expected profits across different operational modes. The results obtained for the three scenarios are detailed in Table 3.12.

On average, when system capacity does not exceed 87%, the drone delivery system achieves a 7.18% performance improvement compared to the batch production system.

Conversely, batch production systems perform better when capacity is fully utilized and demand remains constant over time, yielding on average a 4% improvement relative to the drone system. Based on these values, profit can be expressed in terms of throughput increase per period (T), depending on which production mode is employed.

3.4 Conclusion

The adoption of drones in manufacturing requires a careful balance between technological capability, economic feasibility, and strategic planning. As outlined in this chapter, drones offer unique advantages compared to traditional ground-based systems, particularly in leveraging unused aerial space within facilities. However, their integration is not without challenges. High equipment costs, reliance on robust localization systems, and the need for safe human-drone interaction highlight that adoption must be deliberate, systematic, and aligned with organizational goals.

One of the most critical factors in successful drone adoption is the consideration of Human-Drone Interaction (HDI). Since drones operate in environments shared with humans, the ergonomic design of collaborative work zones and the enforcement of safety distances are essential. By accounting for variations in human posture, body measurements, and regulatory standards, drone systems can minimize risks and ensure operator comfort. Furthermore, the emphasis on Speed and Separation Monitoring (SSM) as a feasible safety mechanism demonstrates the necessity of adapting industrial safety frameworks specifically for aerial platforms.

Beyond the human factors, the chapter emphasizes the importance of simulation as a tool for evaluating drone performance under realistic operating conditions. Discrete Event Simulation models allow manufacturers to test stochastic scenarios, including variable demand, machine failures, and tool requests, without risking actual downtime. By embedding randomness into the evaluation, simulations provide more accurate insights into system resilience and efficiency. Tools such as FlexSim offer a robust platform to assess drone performance, particularly in conflict management, resource allocation, and fleet scheduling.

The envisioned scenario demonstrates how drones can significantly improve reactivity in agile manufacturing systems. By autonomously transporting urgent raw materials, tools, and faulty devices, drones reduce dependency on rigid supply systems that cannot adapt quickly to fluctuations in demand. The integration of charging platforms, path-planning algorithms, and conflict management protocols ensures that drone operations remain efficient and reliable. In doing so, drones enable more dynamic, flexible, and responsive supply chains that align with the principles of Industry 4.0.

In conclusion, the chapter highlights that drone adoption is not simply a matter of

adding new technology; it requires rethinking workflows, safety protocols, and supply chain strategies. Human factors, simulation-based testing, and conflict-free scheduling all play central roles in ensuring that drones deliver strategic value rather than operational risk. By adopting a structured and data-driven approach, manufacturers can integrate drones into intralogistics systems in a way that enhances efficiency, responsiveness, and long-term competitiveness.

Low variability		Medium variability		High variability	
θ	Mode	θ	Mode	θ	Mode
0.90	1	0.90	1	0.90	1
0.90	1	0.95	1	0.55	3
0.90	1	1.00	1	0.85	3
0.95	1	1.00	1	0.90	2
0.90	1	0.95	1	0.60	3
0.90	1	0.85	3	0.65	3
0.95	1	0.85	3	0.80	3
0.95	1	0.80	3	0.95	2
1.00	1	0.85	3	0.75	3
1.00	1	0.75	3	0.50	3
1.00	1	0.90	2	0.55	3
1.00	1	0.85	3	0.70	3
1.00	1	0.85	3	0.75	3
0.95	1	0.75	3	0.55	3
0.90	1	0.65	3	0.55	3
0.85	3	0.65	3	0.85	3
0.85	3	0.65	3	0.60	3
0.95	2	0.75	3	0.75	3
1.00	2	0.75	3	0.80	3
1.00	2	0.85	3	0.65	3
1.00	2	1.00	2	0.80	3
1.00	2	0.95	2	0.90	2
1.00	2	1.00	2	0.90	2
0.95	2	0.85	3	0.85	3
1.00	2	0.95	2	0.70	3
1.00	2	1.00	2	1.00	2
0.95	2	1.00	2	0.90	2
0.90	2	1.00	2	0.70	3
0.90	2	0.95	2	0.60	3
0.95	2	0.80	3	0.50	3
0.95	2	0.70	3	0.90	2
1.00	2	0.75	3	0.55	3
1.00	2	0.80	3	0.65	3
1.00	2	0.90	2	0.70	3
0.85	3	0.90	2	0.95	2
0.90	2	0.85	3	0.75	3
0.90	2	0.90	2	0.65	3
0.95	2	1.00	2	1.00	2
0.90	2	0.90	2	0.90	2
0.90	2	0.75	3	0.65	3
0.85	3	0.85	3	0.80	3
0.85	3	0.95	2	0.70	3
0.80	3	0.95	2	0.90	2
0.80	3	0.85	3	0.55	3
0.80	3	0.70	3	0.75	3
0.85	3	0.70	3	0.55	3
0.90	2	0.65	3	0.95	2
1.00	2	0.75	3	0.65	3
1.00	2	0.75	3	0.95	2
1.00	2	0.95	2	0.65	3

Table 3.12: Comparison of operational modes across variability levels.

Chapter 4

Early Adoption: operational and supply chain factors

As autonomous vehicles operate in shared environments, the risk of resource contention and mid-air collisions grows significantly. Because of this reason, this chapter examines the critical operational and supply chain factors that influence the early adoption of drones in intralogistics systems.

To address these challenges, the chapter presents a conflict-management algorithm designed to supervise and coordinate a fleet of drones in intralogistics scenarios. The discussion focuses on how resources, such as landing stations, charging points, and transit paths, can be allocated in a way that guarantees conflict-free operations.

The conflict-management algorithm relies on the principle of centralized supervision. Unlike decentralized approaches where each drone independently plans its route, this method introduces a supervisory entity that oversees global resource allocation and trajectory planning. By adopting this strategy, it becomes possible to ensure that once a mission has started, both the assigned paths and required resources remain free from interference. This not only prevents accidents but also minimizes delivery delays caused by resource bottlenecks.

In this chapter it is described the two main types of conflicts that arise in drone delivery: resource conflicts and collision conflicts. Resource conflicts occur when multiple drones require simultaneous access to the same landing station or charging point, while collision conflicts emerge when flight paths overlap. The supervisor addresses both categories through proactive routing and reservation mechanisms, ensuring that drones only operate in conflict-free zones. This classification of conflicts will guide the design and implementation of the algorithm introduced in this chapter.

To strengthen the conflict-management approach, the algorithm also incorporates transfer-points and position markers, which act as intermediate waypoints within the operational map. These elements enable more granular control of drone trajectories, allowing

the supervisor to issue updates on speed, direction, and path availability. Position-markers such as ArUco tags further enhance the reliability of the system by providing real-time localization feedback. Together, these tools ensure a high degree of precision in both navigation and conflict resolution.

Finally, the chapter situates the conflict-management algorithm within a broader system architecture for UAV fleet management. This architecture follows a multi-layer paradigm in which the scheduling system interacts seamlessly with drone autopilot software, pose estimation tools, and task allocation modules. By presenting both the algorithm and its integration within the overall architecture, the chapter offers a comprehensive view of how conflict-free drone intralogistics can be achieved in practice.

4.1 Conflict-Management Algorithm: Integration with a supervisor

Considering a system where resources are landing stations that may function as pick-up, drop-off, parking, and charging points, and where a set of vehicles has known locations, the implementation of a supervisor allows the management of resource assignment through routing algorithms that focus on conflict-free trajectories.

In this scenario, the role of the conflict management algorithm is to ensure that, once a vehicle begins its trajectory, no other vehicle occupies or crosses the assigned path. In coordination with the supervisor, the algorithm also ensures that all resources required for the delivery remain available throughout the mission. Once a delivery order is assigned and the vehicle begins its mission, the associated resources and paths are reserved and become unavailable to the rest of the system.

As long as an active vehicle remains within its conflict zone, other vehicles cannot be assigned to the same resource or cross its designated path. During the mission, drones report their positions, and resource status is updated when active vehicles enter a conflict-free zone.

4.1.1 Types of conflicts and how to manage them

During task assignment, the supervisor must manage various conflicts that may arise and affect drone operations. Since responsiveness is a defining feature of a drone delivery system, maintaining optimal conditions to prevent deadlocks and potential accidents is essential. In general, the supervisor must address two types of conflicts:

Resource conflict: This type of conflict may arise in two scenarios. First, when two active delivery orders share the same pick-up or drop-off location. Since drone-

dedicated landing platforms are often scarce and shared across different purposes, multiple orders may target the same resource. The second scenario occurs when a drone requires battery recharging and other vehicles also need access to the same charging location. If battery replacement is used instead of automatic charging, separate landing locations can be assigned to each vehicle to avoid resource sharing.

Collision conflict: A collision conflict arises when an active drone shares all or part of its route with another vehicle in flight. Even if an inactive vehicle only needs to cross a route already occupied by another vehicle, the supervisor must block that route and delay the inactive vehicle's departure until it is completely free of conflicts or unplanned obstacles.

To ensure optimal performance, the supervisor must route vehicles in a way that avoids resource and collision conflicts throughout the mission. If such conflicts arise, they must be resolved before drone execution begins. Consequently, delivery orders should be carried out only when the assigned area is confirmed to be conflict-free.

This means that drones rely on the supervisor for location assignments and therefore are not permitted to alter their trajectory once a mission has begun. A drone may change its route only if a safety protocol is activated; in all other cases, it must follow the assigned path.

Optimal tracking and conflict management can be achieved by defining a maximum path length between two locations within a route. These locations are not resources or physical sites where drones operate, but rather waypoints on the map that serve as transfer points with specific functions. Transfer points can be used to apply trajectory updates, such as speed adjustments, drone rotation, or route changes, and to reopen previously occupied paths once the active drone has moved a safe distance beyond the last transfer point. Some transfer points can also be associated with physical reference markers, such as ArUco tags, which the drone can use to re-evaluate its global position on the map.

4.1.2 Integration of the conflict-management algorithm

The conflict-management algorithm can be expressed as a pseudo-code that can be further implemented within the supervisor. Algorithm 1 outlines the integration steps.

As explained earlier, a collision conflict may occur when the path of an active drone overlaps with the estimated path of an inactive drone. In such cases, delivery orders are placed on hold until the system receives a position update. Figure 4.1 illustrates the concept of path reservation after a mission has been assigned. In this example, the blue path represents an assigned mission with a final destination at N-01. While the assigned drone is active, paths P-04, P-02, and N-02 remain unavailable for new delivery requests.

Algorithm 1 CONFLICT_MANAGEMENT Method

```
1: Input: order – delivery request with pickup, drop-off, and required resources
2: Output: status – SUCCESS if a conflict-free mission is assigned, FAILURE otherwise
3: candidateVehicles ← FindAvailableVehicles(order)
4: for each v ∈ candidateVehicles do
5:     path ← PlanPath(v.position, order.pickup, order.dropoff)
6:     mission ← CreateMission(v, path, order.resources)
7:     conflict ← false
8:     for each activeMission ∈ ActiveMissions do
9:         if Intersects(mission, activeMission) then
10:            conflict ← true
11:            break
12:        end if
13:    end for
14:    if conflict = true then
15:        continue                                   ▷ try next vehicle
16:    end if
17:    if Reserve(mission) = true then
18:        AssignMission(v, mission)
19:        return SUCCESS
20:    end if
21: end for
22: return FAILURE
```

When the drone lands at N-01, the supervisor updates the route status and releases the previously reserved path. If a delivery order must be delivered to N-02, the inactive drone can then begin its mission, while the drone at N-01 waits until the other vehicle reaches its destination and starts landing.

However, different strategies can be implemented to mitigate collision conflicts. One solution to the conflict illustrated in Figure 4.1 is to add an additional path between P-04 and N-02, allowing the drone to reach a higher altitude than the active drone while maintaining safety boundaries. Although the map is represented in a 2D model, alternative paths may correspond to changes along the z-axis, which are transmitted as commands to the drone regardless of the visual representation.

4.1.3 Related work on conflict management for drone scheduling

The conflict-management algorithm described above has long been an established research area within operations research and multi-robot systems. On one side, researchers propose Vehicle Routing Problems with Time Windows (VRPTW) to provide a natural framework for batch planning and temporally constrained deliveries. In this sense, VRPTW formulations determine which vehicle serves which customer and when, optimizing route cost under capacity and time constraints (Kallehauge, 2008).

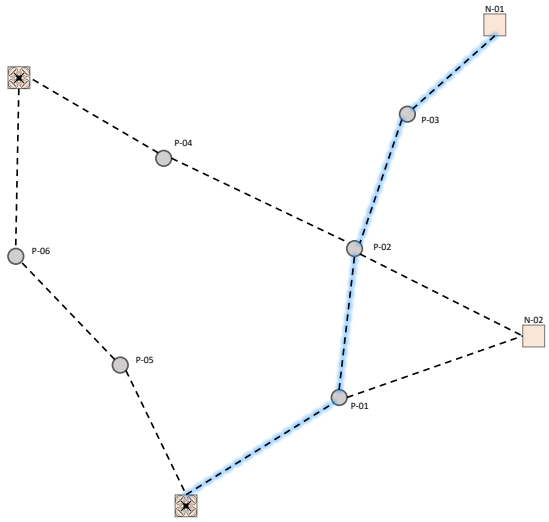


Figure 4.1: Path reservation

However, classical VRPTW models generally treat vehicles as abstract routing entities without explicitly modeling geometric path overlap or segment-level exclusivity. Consequently, they do not inherently prevent mid-route conflicts in dense shared spaces. For problems requiring temporal reservations of physical arcs or nodes, as in our approach, VRPTW techniques (including dynamic or online VRP extensions) remain useful for high-level assignment and time-feasibility checks, but they must be complemented by trajectory-level conflict detection to ensure safety in continuous space.

Similarly, the Multi-Agent Path Finding (MAPF) literature provides the closest methodological analogue to the collision-avoidance component of our algorithm. MAPF explicitly searches for conflict-free paths for multiple agents on a discretized graph and offers both optimal and suboptimal solvers (Stern et al., 2019).

The main difference lies in the modeling assumptions and constraints. MAPF typically assumes discrete time steps on a known graph with fixed start and goal pairs, optimizing either makespan or the sum of costs. This formulation is effective when the environment and agent kinematics can be accurately represented on a graph. In the context of indoor drone intralogistics, however, operations combine graph-like transfer points (which MAPF can leverage) with continuous 3D motion, energy and battery constraints, and shared resources such as charging stations or landing platforms that must be reserved for the duration of a mission.

Thus, while MAPF solvers provide a foundation for path-reservation logic, our implementation integrates MAPF-style conflict checking with resource reservation and energy models that are not typically included in standard MAPF formulations.

Furthermore, an important aspect of the system design concerns the choice between centralized and distributed/decentralized coordination. Distributed protocols, such as token-passing, market or auction mechanisms, and negotiation frameworks, scale ef-

fectively and provide robustness against communication failures or single-point breakdowns (Doriya et al., 2015). These approaches have been successfully applied across various multi-robot domains (Miele et al., 2025). Token-based schemes, for example, regulate access to shared resources by transferring permission objects among agents, while market-based and negotiation schemes allow agents to bid for tasks or time slots, distributing decision-making across the network.

The trade-off motivating a centralized supervisor is predictability and authority over safety and critical reservations. In indoor warehouse settings characterized by scarce landing platforms and strict energy constraints, centralized global knowledge ensures that once a mission is accepted, the corresponding path and physical resources remain reserved for its entire duration. This property is difficult to guarantee in decentralized schemes without additional synchronization or coordination overhead. In summary, distributed approaches enhance scalability and fault tolerance, whereas centralization simplifies the verification of safety invariants required for conflict-free operation under tight resource constraints.

Overall, the drone scheduling and intralogistics literature increasingly converges on the same hybrid requirements addressed in this work: vehicle assignment, path conflict management, energy constraints, and shared infrastructure (Boysen et al., 2021). As noted by Boysen et al. (2021), decision layers in drone scheduling typically consist of three main elements: assignment, routing, and resource infrastructure. The author also emphasizes that realistic warehouse implementations require co-design of flight corridors, charging and landing layouts, and vehicle-dependent scheduling logic—principles directly reflected in our supervisor's integration of transfer points, resource reservations, and energy-aware task assignment.

Finally, the approach presented in this work represents a pragmatic synthesis rather than a new paradigm. It combines path-conflict reasoning from MAPF, scheduling and time-window feasibility from the VRPTW and dynamic VRP literature, and rigid reservation strategies common in safety-critical airspace management.

As described earlier, the idea behind for this synthesis arises from the following operational characteristics:

- Indoor environments lacking GNSS coverage and with constrained vertical separation,
- Limited multipurpose landing and charging stations that require end-to-end reservation, and
- Tight energy budgets that make mission aborts or mid-flight replanning costly.

For these reasons, we argue that a centrally supervised, reservation-centric algorithm, supported by local autonomy for perception and avoidance (Visual odometry, ArUco-

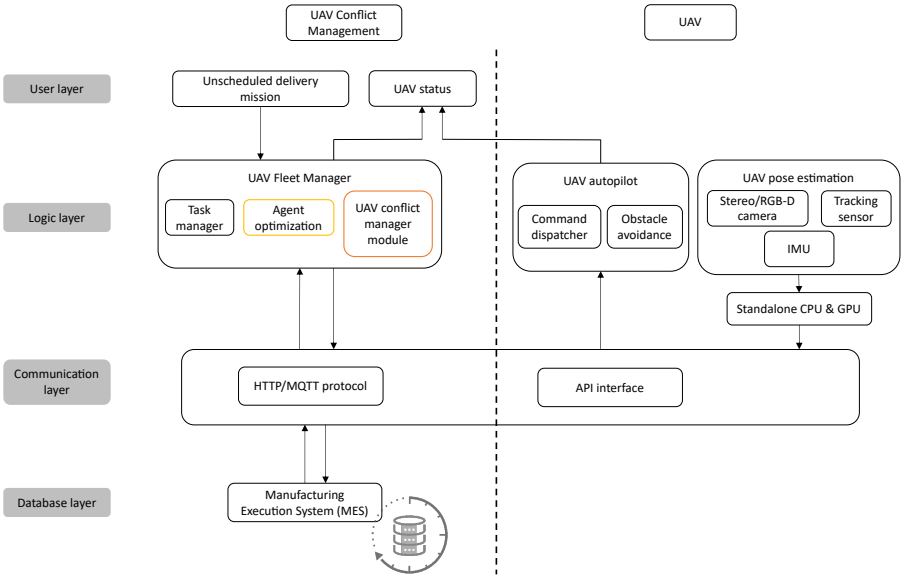


Figure 4.2: UAV System Architecture

based localization, and low-level collision prevention), best satisfies the operational requirements of industrial intralogistics while remaining compliant with formal safety verification and practical deployment considerations.

4.2 System Architecture

The UAV Conflict Management System would work as a component of the UAV Fleet Manager. According to Fowler (2012), the system can be described as a multilayer architecture. Figure 4.2 shows the different layers and components of the scheduling system, providing a broader view of the drone intralogistics system.

UAV scheduling is handled at the logic layer. This layer is divided into two sections: the online scheduler, which generates and assigns tasks to drones, and the UAV autopilot and pose estimation software, which together communicate real-time drone status to the scheduling system. The agent optimization module selects the optimal drone for each task based on energy levels and locations, while the task manager module prioritizes urgent areas. Simulations should replicate this communication behavior between the scheduler and the UAVs.

4.2.1 Drone platform

The development of autonomous aerial platforms has been accelerated by advances in onboard computing and perception technologies. Among these platforms, the Sentinel drone developed by ModalAI represents a reference architecture tailored for both research and prototyping of autonomous navigation. Designed as a modular and open system,

Sentinel integrates high-performance compute resources with a robust flight controller and multiple vision sensors, enabling researchers to investigate and validate navigation algorithms in real-world environments. This platform provides a balance between payload capacity, computational throughput, and flight endurance, making it suitable for robust applications.

A central component of Sentinel is its visual navigation capability, particularly its stereo vision system. Equipped with multiple synchronized image sensors, the platform supports stereo camera configurations that enable real-time depth perception. Stereo visual odometry algorithms exploit image disparity to reconstruct the three-dimensional structure of the environment and simultaneously estimate the drone's motion. This approach is especially advantageous in indoor environments where global navigation satellite systems (GNSS) are unavailable. Using visual odometry, Sentinel can compute its relative position and orientation with high accuracy, enabling consistent localization without reliance on external infrastructure.

The benefits of stereo visual odometry are closely related to the algorithms implemented. Algorithms such as feature-based matching and semi-dense reconstruction provide robustness against texture-poor surfaces and varying illumination conditions, while minimizing drift through continuous optimization (ModalAI, 2025). Compared to monocular approaches, stereo visual odometry reduces scale ambiguity by directly estimating the metric depth from image disparities. Moreover, when integrated with inertial measurements from Sentinel's onboard IMU, visual–inertial odometry enhances both accuracy and robustness, particularly under rapid maneuvers or temporary visual occlusions.

In addition to perception, this platform benefits from a powerful onboard processor, the Qualcomm QRB5165, which is optimized for machine vision and AI workloads. This processing capability allows complex algorithms, including simultaneous localization and mapping (SLAM), obstacle detection, and real-time trajectory planning, to run simultaneously without external computation. The ability to process high-bandwidth sensor data onboard contributes to autonomy in constrained indoor environments where latency and communication reliability limit the usefulness of remote processing. Consequently, Sentinel serves as a platform where perception and decision-making are highly important, reinforcing reliability during extended autonomous missions.

Sentinel's autonomy is reinforced by its integrated hardware and software architecture. With a flight controller based on PX4, Sentinel provides compatibility with middleware such as ROS and MAVLink, facilitating interoperability within robotics research ecosystems. This integration enables seamless deployment of autonomous capabilities, from waypoint navigation to reactive obstacle avoidance. Moreover, Sentinel's modular connectivity options, including Wi-Fi and 5G, enable remote supervision when needed while maintaining full autonomy when disconnected. The combined hardware and software

stack thus supports not only autonomy but also flexibility across diverse experimental conditions.

4.3 openTCS: Drone Fleet Manager

OpenTCS is an open-source control system developed by the Fraunhofer Institute for Material Flow and Logistics (IML). It is a vendor-independent tool for automated transportation systems that require coordinated fleet management. The software manages vehicles independent of their specific characteristics, such as navigation methods, track guidance systems, or handling equipment, and is capable of simultaneously controlling multiple vehicle types performing different tasks.

Because openTCS is developed in Java, it runs on any operating system with a Java Runtime Environment (JRE) version 1.8 or higher. The software package consists of several components that together form a complete control system. Figure 4.3 illustrates these components and their communication within the system.

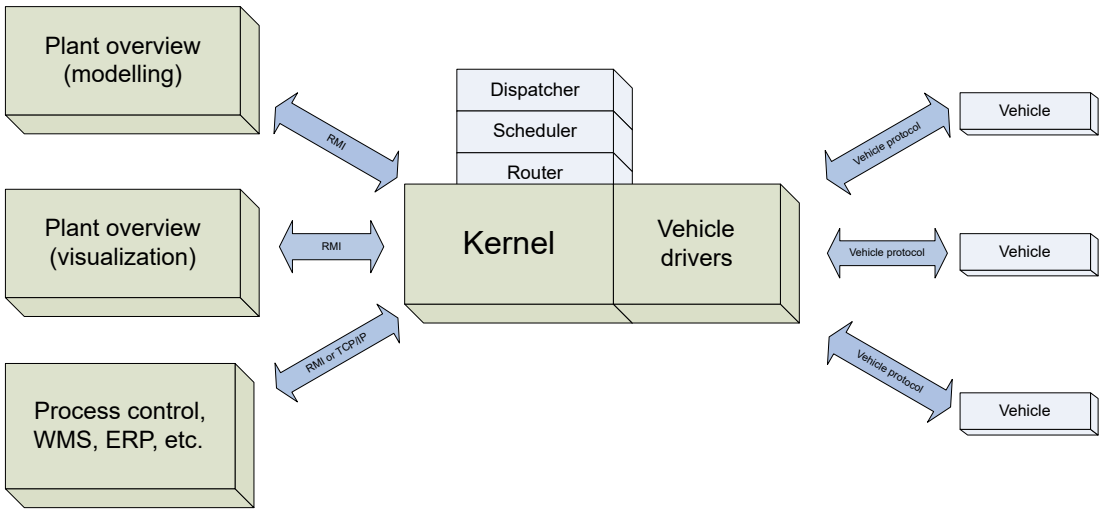


Figure 4.3: openTCS system components

The Kernel manages the overall system state, assigns transport orders to vehicles, computes their routes, and oversees track clearance. It also provides interfaces for user interaction and integration with higher-level process control systems.

In addition, the kernel provides a framework with standardized interfaces for vehicle drivers. These drivers translate the kernel's abstract instructions into vehicle-specific communication protocols and convert the vehicles' reports into a standardized state representation for the kernel. Consequently, only the vehicle driver is responsible for handling the vehicle-specific details. Overall, the kernel includes three key strategy modules for processing transport orders:

Dispatcher: Determines which vehicle should handle each transport order and decides vehicle actions in special situations, such as when no orders are pending or when a vehicle's energy is low.

Router: Calculates optimal routes for vehicles to reach their destinations.

Scheduler: Oversees resource allocation to manage traffic and prevent collisions between vehicles.

Furthermore, through the graphical interface of the plant overview, users can design and adjust driving course models to create complete transport systems. In addition to defining key aspects of the plant layout, features such as reporting and halting points, loading/unloading stations, and vehicle path–specific speed limits can also be configured.

The plant overview provides a real-time visualization of the system's state. It displays vehicle positions and planned routes, and allows operators to interact directly with the system. For example, transport orders can be manually created or canceled if a vehicle experiences a malfunction.

4.3.1 Plant model elements

In openTCS, a plant model consists of a set of elements that represent physical features of a delivery map. The attributes of these elements can denote, for example, coordinates or the length of a path. In general, plant models can be designed with the use of 6 different elements:

Point: Points are logical mappings of discrete vehicle positions in the driving course. A point carries the following attributes: type, position, vehicle orientation angle, vehicle envelopes, and a maximum vehicle bounding box.

Path: Paths are connections between points that are navigable for vehicles. A path's main attributes, next to its source and destination point, are: length, maximum velocity, locked flag, peripheral operations, and vehicle envelopes.

Location: Locations are markers for points at which vehicles may execute special operations (load or unload cargo, charge their battery, etc.). A location's attributes are: type, links to points, and locked flag.

Vehicle: Vehicles map physical vehicles for the purpose of communicating with them and visualizing their positions and other characteristics. A vehicle provides the following attributes: critical energy level, good energy level, sufficiently recharged energy level, fully recharged energy level, maximum velocity, integration level, paused flag, acceptable transport order types, route color, envelope key, and a bounding box.

Block: Blocks (or block areas) are areas for which special traffic rules may apply. They can be useful to prevent deadlock situations, e.g. at path intersections or dead ends. A block has two relevant attributes: set of members that the block is composed of, and a type which determines the rules for entering the block.

Layer: Layers are abstract elements that group points, paths, locations and links. They can be useful for modeling complex plants and dividing plant sections into logical groups (e.g. floors in a multi-floor plant). A layer has the following properties: active flag, visible flag, descriptive name, group, and a group visible flag.

Figure 4.4 illustrates these elements and their integration within a delivery system. In this model, the purple line represents the planned path for Vehicle-03 to travel from "Point-0013" to "Goods in North 01". Points 0002, 0004, 0006, 0010, and 0012 indicate the locations of various recharge stations.

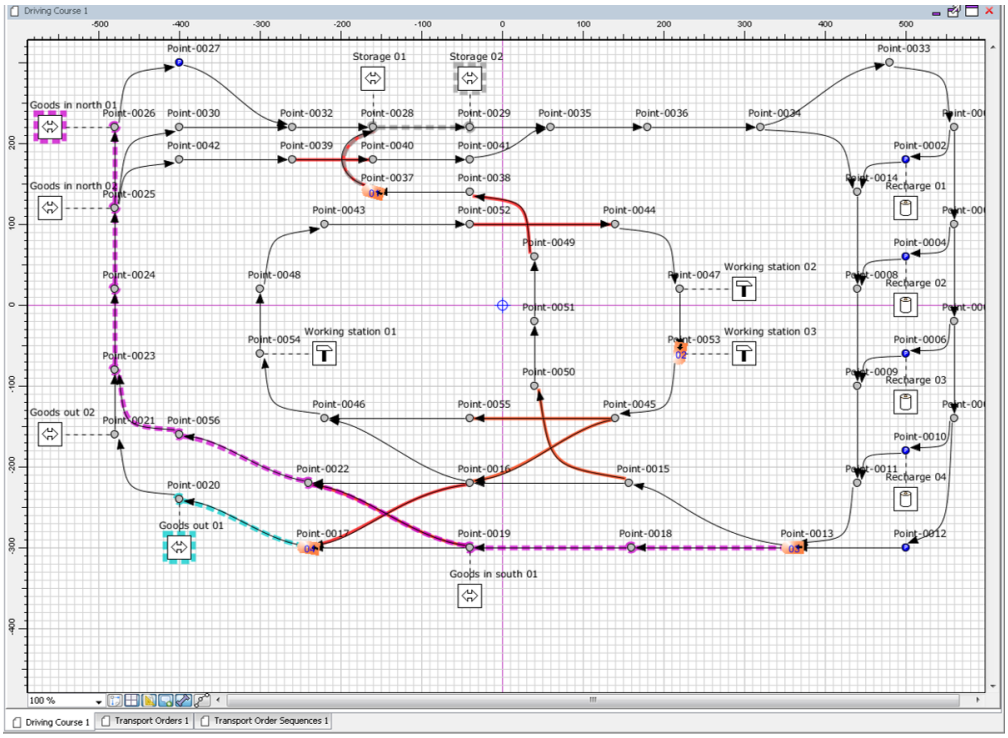


Figure 4.4: openTCS Plant overview

4.3.2 Plant operation elements

Transport orders and order sequences exist only during plant operation, with attributes defined primarily at creation. A transport order specifies a sequence of movements and operations performed by a vehicle and includes parameters that can be set when it is created. In general, transport orders are composed of the following attributes:

Destinations: A transport order consists of an ordered sequence of destinations for the vehicle to visit. Each destination includes a location to travel to and an operation to perform there.

Deadline(optional): Specifies when the transport order should be completed.

Type(optional): A string used to filter which vehicles can be assigned to the transport order. A vehicle can only receive an order if its set of acceptable order types includes the order's type. Example types might include "Transport" or "Maintenance".

Intended vehicle(optional): Directs the dispatcher to assign the order to a specific vehicle rather than selecting one automatically.

Dispensable flag: Indicates whether the order can be withdrawn automatically, typically to free the vehicle for another transport order. For example, parking orders are often marked as dispensable: if a vehicle is en route to a parking position but a new transport order becomes available, the vehicle may skip the remainder of its journey to handle the new order sooner.

Dependencies(optional): References to other transport orders that must be completed first. Dependencies are transitive: if order A depends on order B, and order B depends on order C, the processing sequence is C $\rightarrow$ B $\rightarrow$ A. This allows defining an execution order for a set of transport orders. Dependencies do not require all orders to be processed by the same vehicle unless the intended vehicle attribute is also set.

4.3.3 openTCS integration

Integrating openTCS as a supervisor for a fleet of drones requires developing custom modules to interface with the system. Both the routing algorithm proposed in Chapter 2 and the conflict manager algorithm must be incorporated into the kernel. At its core, this integration allows the dispatcher and router modules to implement the proposed methodology.

First, communication between the supervisor, the drones, and the manufacturing system is enabled using the MQTT protocol. MQTT employs a publish–subscribe architecture, decoupling publishers from subscribers. This abstraction facilitates modular and flexible system design, allowing sensors, controllers, and supervisory software to exchange information without direct coupling. Its low bandwidth usage and small packet size make it well-suited for wireless or embedded systems with constrained resources.

As a means of comparison, the Hypertext Transfer Protocol (HTTP) operates primarily on a request–response model, unlike MQTT, which is event-driven and connection-

oriented. While HTTP is ubiquitous and widely supported, it has significantly higher overhead due to verbose headers and its stateless nature, making it less suitable for continuous or low-latency robotic communication. For this reason, MQTT is better suited as a communication protocol for testing new systems.

Natively, openTCS allows external data requests and modifications through its API. This is done by sending GET, POST, or PUT requests to predefined endpoints. By default, these requests are accepted without authentication, although an access key can optionally be configured in the kernel. For transport orders, new requests can be submitted via the POST topic */transportOrders/{NAME}*.

To ensure compatibility with openTCS, the Manufacturing Execution System should convert new delivery orders into a JSON request body formatted for openTCS. The request body should have the following format:

```
{
  "incompleteName": false,
  "dispensable": false,
  "deadline": {},
  "intendedVehicle": "",
  "peripheralReservationToken": "Token-001",
  "wrappingSequence": "",
  "type": "",
  "destinations": [
    {
      "locationName": "Storage 01",
      "operation": "Load cargo",
      "properties": [
        {
          "key": "key1",
          "value": "value1"
        }
      ]
    }
  ],
  "properties": [
    {
      "key": "key1",
      "value": "value1"
    }
  ],
  "dependencies": [
    "TOrder-001"
  ]
}
```

Specifically, the MES should be able to manage itself automatically the “type”, “locationName”, “operation”, and “dependencies” if any. Furthermore, key properties under the labels “key” and “value” are essential, representing the product type (key) and its weight (value).

Once the transport order is loaded into the system, the router and dispatcher execute the algorithms proposed in this work. For the router, openTCS allows the injection of

custom methods through Guice modules that configure its components. The following steps are required to customize the application:

1. Build a JAR file for the custom injection module with the following content:
   (a) A subclass of org.opentcs.customizations.kernel. KernelInjectionModule, which can be found in the base library, must be contained. Configure the custom components or adjust the application's default wiring in this module. KernelInjectionModule provides a few supporting methods to use.
   (b) A plain text file named META-INF/services/org.opentcs.customizations.kernel.KernelInjectionModule must also be contained. This file should contain a single line of text with the fully qualified class name of your module.
2. Ensure that the JAR file(s) containing your Guice modules and the implementation of the custom component(s) are part of the class path when the kernel application is started.

For the current integration, a "customRouter" that extends the "DefaultRouter" has been developed. Extending the functions of the default router, rather than replacing them, enables differentiation between vehicles: drone-specific tasks can implement the new algorithms, while ground robots continue with the standard approaches.

The key difference between the default approach and the proposed methodology is in battery management. Standard approaches for ground vehicles do not require estimating battery usage for an entire trip. In contrast, drone implementations must ensure that the vehicle does not run out of battery before reaching its target and can safely leave the drop-off area and return to a parking zone.

A full delivery mission involves three general tasks: pick-up, drop-off, and parking/charging. The router must ensure that the drone can complete the mission before assigning the task to a vehicle. Unloading tasks should be generated as dependencies of previously assigned loading operations, either automatically within the MES or via the MQTT interface, which links requested items to their origin. If the latter applies, algorithm 2 outlines the procedure to follow.

Because delivery missions are linked via the "wrappingSequence", they can be evaluated together to compute the total energy required for the trip. As presented in Chapter 2, drone applications rely on accurate energy consumption models that reflect the nature of the delivery. Given that unloading consumes more energy, the routing algorithm should consider the weight when evaluating trip feasibility. Return-to-origin missions can be overridden if the drone reaches its battery threshold, in which case a charging mission is scheduled instead.

Algorithm 2 MES-to-openTCS Mission creation

Input: *mesOrder* – delivery order from MES with {*destinations*, *operation*, *key*}
Output: *missions* – sequence of loading, unloading, and return-to-origin tasks for openTCS
productKey ← *mesOrder.properties.key*
originLocation ← ResolveOrigin(*productKey*)
wrappingID ← GenerateWrappingSequenceID()
▷ — Create Loading Operation —
LOrder ← CreateOperation()
LOrder.name ← "LOrder-01"
LOrder.type ← "Load"
LOrder.destinations ← {*originLocation*, "Load cargo"}
LOrder.wrappingSequence ← *wrappingID*
LOrder.dependencies ← {}
▷ — Create Unloading Operation from MES data —
UOrder ← CreateOperation()
UOrder.name ← "UOrder-01"
UOrder.type ← *mesOrder.operation*
UOrder.destinations ← *mesOrder.destinations*
UOrder.wrappingSequence ← *wrappingID*
UOrder.dependencies ← {*LOrder.name*}
▷ — Create Return-to-Origin Operation —
ROrder ← CreateOperation()
ROrder.name ← "ROrder-01"
ROrder.type ← "Return"
ROrder.destinations ← {*vehicle.initialLocation*}
ROrder.wrappingSequence ← *wrappingID*
ROrder.dependencies ← {*UOrder.name*}
▷ — Publish Operations to openTCS via MQTT —
SendToOpenTCS(*LOrder*)
SendToOpenTCS(*UOrder*)
SendToOpenTCS(*ROrder*)
return { *LOrder*, *UOrder*, *ROrder* }

Regarding the conflict management algorithm, there are two implementation approaches. The first approach involves working directly with the Kernel's "standard-Dispatcher" module and updating the logic when a resource is shared during path assignment. The second approach interacts with the web API module, which provides a PUT function to lock paths and locations via the topics /paths/NAME/locked and /locations/NAME/locked, respectively. Once a path is locked, it cannot be used by any other vehicle; however, this does not prevent vehicles from starting their journeys.

Missions placed on hold due to conflicts can be managed externally using an MQTT module that retrieves and updates data directly in openTCS. First, the algorithm should call GET /transportOrders and evaluate orders with an active state. The returned body matches the format presented above. If the "locationName" of an active order matches that of the newly created delivery order, the new order cannot be set as "Dispatchable" and remains in the "RAW" state.

The module should update missions whenever an active order's state changes to "Finished". As orders are linked to an increasing "wrappingSequence" value, the module can assign new orders based on this sequence, ensuring that resources are not allocated simultaneously to multiple vehicles.

Path conflicts are handled in a subsequent step. Similar to resource conflicts, the MQTT module requests current information about active vehicles using the GET function

/vehicles. The returned body is shown in Appendix B. Active vehicles have "PROCESSING_ORDER" as the value of the "procState" property.

After this, assigned paths can be verified under the properties "allocatedResources" and "claimedResources". If no conflicts exist with the paths assigned to the new delivery order, the drone can start its journey. Each vehicle is assigned a "paused" state equal to true upon landing; without this, vehicles would continue operations regardless of conflicts. The "paused" state is set to false once the previous validations are complete. The steps required to accomplish this are described in Algorithm 3.

Algorithm 3 EXTERNAL_CONFLICT_MANAGEMENT via MQTT

Input: *newOrder* – delivery order with $\{destinations, operation, key\}$
Output: *status* – SUCCESS if mission is dispatchable, RAW otherwise
▷ — Step 1: Check Resource Conflicts —
activeOrders ← HTTPRequest(`GET /transportOrders`)
for each *order* ∈ *activeOrders* **do**
  if *order.state* = "ACTIVE" **and** *order.destinations.locationName* = *newOrder.destinations.locationName* **then**
    newOrder.state ← "RAW" ▷ Hold order until conflict clears
    return RAW
  end if
end for
▷ — Step 2: Wait for Completion of Previous Orders —
while ActiveOrder(< *newOrder.wrappingSequence*) **do**
  WaitForEvent("orderFinished")
end while
▷ — Step 3: Check Path Conflicts —
vehicles ← HTTPRequest(`GET /vehicles`)
for each *v* ∈ *vehicles* **do**
  if *v.procState* = "PROCESSING_ORDER" **then**
    if Interferes(*newOrder.path*, *v.allocatedResources* ∪ *v.claimedResources*) **then**
      vehicle.paused ← "true"
      return true
    end if
  end if
end for
▷ — Step 4: Release Vehicle if Safe —
assignedVehicle.paused ← false
newOrder.state ← "DISPATCHABLE"
return SUCCESS

Finally, battery data cannot be transmitted via MQTT through the openTCS web API. To address this, a standalone vehicle driver must be developed and integrated into the openTCS architecture. First, a folder containing the driver modules should be created in the root directory of openTCS. The folder structure should be as follows:

```
opentcs/
└── opencs-SentinelCommAdapter/
    ├── build.gradle
    └── src/
        └── main/
            ├── resources/
            │   └── META-INF/
            │       └── services/
            └── java/
                └── org/
                    └── opentcs/
                        └── sentinel/
                            ├── SentinelCommAdapter.java
                            ├── SentinelCommAdapterDescription.java
                            ├── SentinelCommFactory.java
                            ├── SentinelCommAdapterModule.java
                            └── SentinelProcessModel.java
```

The content of each file is described in the Appendix. Similar to the MES, the drone should independently send MQTT messages containing at least data related to battery level, VIO location points, speed, and status. The MQTT topic used in this development is labeled “sentinel/vio”, which transmits the following dataset:

```
{
    "px": 0.0,
    "py": 0.0,
    "pz": -1.0,
    "vx": 0.0,
    "vy": 0.0,
    "vz": 0.0,
    "ax": 0.0,
    "ay": 0.0,
    "az": 0.0,
    "jx": 0.0,
    "jy": 0.0,
    "jz": 0.0,
    "yaw": 0.0,
    "wz": 0.0
    }
```

4.4 Conclusion

This chapter presented a structured approach to the early adoption of drone-based intralogistics, focusing on the operational mechanisms required to ensure safe and efficient

fleet coordination. The main contribution lies in the development of a centralized conflict-management algorithm capable of supervising multiple UAVs operating in shared industrial spaces.

By distinguishing between resource conflicts and collision conflicts, the framework enables proactive allocation of critical assets such as landing stations, charging points, and transit paths. The integration of transfer-points and position markers further enhanced the precision of route supervision, while the proposed system architecture, built upon the openTCS fleet management platform and the Sentinel UAV, demonstrated the technical feasibility of embedding conflict-free operations within industrial settings.

Despite its promising results, the proposed approach remains subject to several limitations that must be addressed in future work. The algorithm assumes a controlled and structured environment, where the number of drones, operational zones, and available resources can be clearly defined. As such, scalability to larger fleets and adaptability to unstructured or dynamically changing layouts remain open challenges.

In addition, factors such as communication latency, sensor noise, and environmental disturbances may affect real-time decision-making and should be incorporated into future simulations and field tests. From a broader perspective, the successful deployment of such systems also depends on their compatibility with existing production workflows and on the organizational readiness to manage autonomous aerial operations.

Building on the technical foundations established in this chapter, the next stage of analysis shifts toward the organizational and behavioral dimensions of drone adoption. While centralized supervision and conflict management ensure operational safety, long-term integration requires a comprehensive understanding of human–system interaction, workplace safety, and risk perception.

The following chapter therefore introduces a risk management framework inspired by aviation safety principles, adapted to the context of indoor manufacturing. It examines how organizational structures, safety standards, and behavioral factors collectively influence the sustainable and trusted deployment of drones within Industry 4.0 environments.

Chapter 5

Adoption: Organizational and behavioral factors

Chapter 5 explores the organizational and behavioral dimensions that shape the adoption of drones in intralogistics, focusing on how internal structures, workforce readiness, change management, and stakeholder perceptions influence successful implementation.

The adoption of drones in industrial intralogistics extends far beyond technical feasibility; it also requires addressing organizational and behavioral factors that determine safe and sustainable integration. Unlike traditional ground-based systems, aerial vehicles introduce new layers of complexity, as their operations take place in shared environments that involve both humans and automated machinery. Consequently, the evaluation of drone deployment cannot be limited to performance and efficiency alone, it must also ensure that risks to operators, infrastructure, and workflow continuity remain within acceptable thresholds.

A well-established approach to aviation safety, the Equivalent Level of Safety (ELOS), provides a useful reference point for defining operational limits in unmanned aerial systems. However, because drones differ significantly from manned aircraft in size, application, and failure modes, a direct transfer of these standards is not practical. Instead, equivalent principles must be adapted to account for the specific hazards of indoor intralogistics, including unintended movements, mid-air collisions, and premature flight termination. These risks, if unmanaged, could not only result in physical damage or injury but also undermine organizational acceptance of drone operations, thereby limiting their long-term viability.

To address these challenges, this chapter introduces a structured risk management framework tailored to drone operations in manufacturing contexts. By combining hazard identification, risk evaluation, reduction, and acceptance into a systematic process, managers can establish clear safety thresholds while enabling flexible and efficient operations.

The technical backbone of this approach is provided by two complementary systems:

the Sentinel drone platform and the openTCS supervisory framework. The Sentinel, built on ModalAI's VOXL2 architecture, integrates advanced sensors, redundant localization systems, and robust failsafe mechanisms to ensure reliable autonomous navigation. Meanwhile, openTCS supervises fleet-level task management, path allocation, and conflict avoidance, ensuring that drones operate in a coordinated and predictable manner across complex production environments. Together, they create a dual-layer system in which hardware reliability and supervisory control reinforce each other.

Ultimately, the successful adoption of drones in intralogistics depends on balancing technical performance with organizational readiness and risk acceptance. This chapter examines how safety protocols, mission profiles, and supervisory systems can be aligned with broader organizational factors to foster trust, reduce hazards, and pave the way for widespread adoption. By doing so, it establishes the necessary foundations for integrating drones as reliable, safe, and strategically valuable assets in Industry 4.0 ecosystems.

5.1 Risk Management

The primary goal of evaluating drone operations is to ensure that the system does not present a risk to the environment in which it is used. When considering manned aviation, the "Equivalent Level of Safety" (ELOS) is usually employed to quantify the safety level an aircraft may have. Although manned aviation and drones are not the same, they share some aspects that make them comparable in practice.

As drone applications are not yet standardized, some agencies, such as the EASA (A-NPA, 2005), follow equivalent principles to those of manned aviation when evaluating safe operation of unmanned aerial vehicles. In this matter, for example, the Range Commanders Council (Council, 1999) states that:

> Any UAV operation or test must show a level of risk to individuals no greater than that for an operation or test of a piloted vehicle.

Logically, evaluation methods for large manned aircraft, such as ELOS, cannot be directly used for drones as they consider failure conditions that are not present in unmanned small-size applications. Applying the same requirements as those for commercial applications would only lead to unrealistic high-reliability equipment (Dalamagkidis et al., 2009). In addition, research on aircraft accidents have shown that the main cause for failure in manned aviation is pilot error (Dalamagkidis et al., 2009). Based on this, even if unrealistic high-reliability equipment is available, it would have only a minor effect on overall flight safety.

Because the failure rate standards established for commercial aircraft are not directly transferable, alternative methods must be applied when setting safety requirements for

UAVs. One common practice in safety engineering is to establish accident-specific safety limits by linking them to the acceptable probability of the most severe potential consequence (Council, 1999). From this, the upper limit for acceptable failure frequency can be established. In the case of industrial drone operations, the most severe consequence of accident is human injury, which can be comparable to the Equivalent Level of Safety (ELOS) and be framed around it. However, an ELOS defined solely by the probability of accidents leading to at least one human causality is insufficient, since it does not account for scenarios involving multiple risks.

Contrary to the ELOS standard, intralogistics drone operations are subject to various hazards that can lead to three primary accidents (Dalamagkidis et al., 2009): unintended or abnormal system mobility operation, mid-air collision, and early flight termination. Figure 5.1 provides an overview of potential accidents and their associated consequences arising from indoor drone operations.

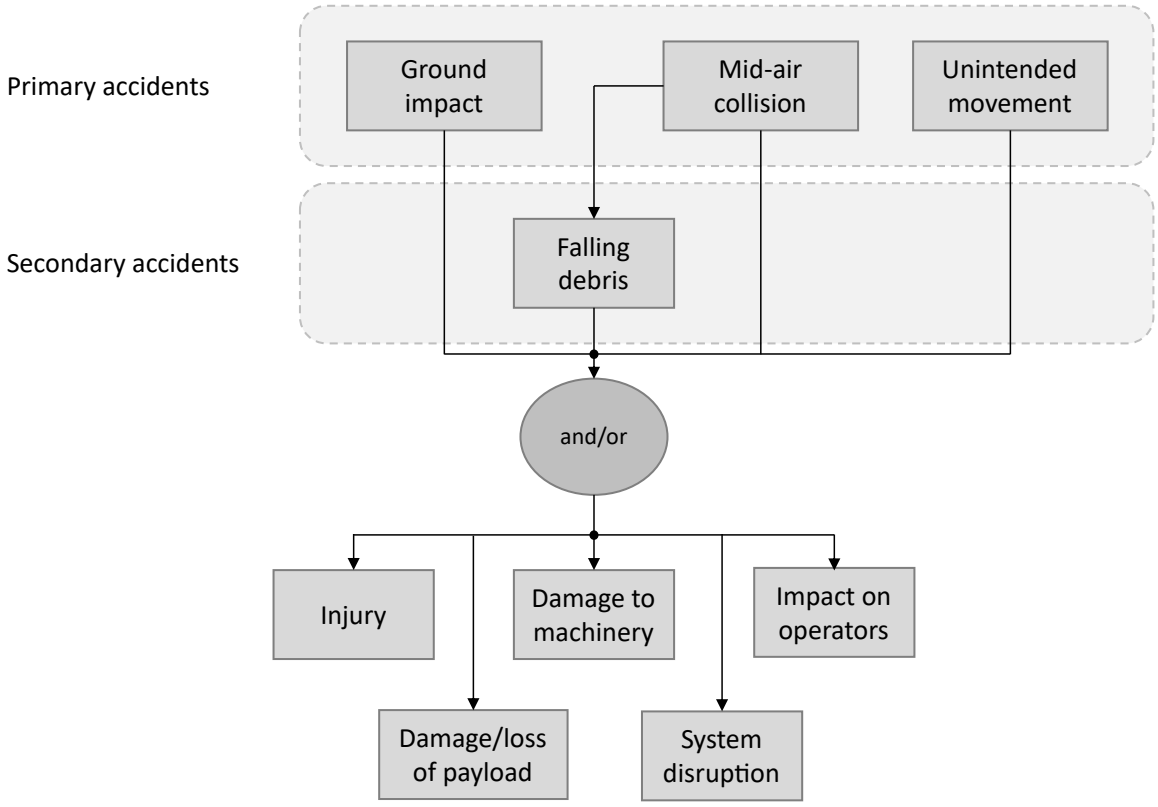


Figure 5.1: Primary and secondary accidents from indoor drone operation

Accidents involving unintended or abnormal ground movement occur when a drone begins to move while still on the surface. Such incidents can pose serious risks to ground personnel, often resulting from operator error. This may happen when the operator lacks direct visual contact with the aircraft and assumes the area is clear. These events are not examined further here, as the associated risks can be effectively minimized through stricter operational management and adherence to established procedures.

Mid-air collisions represent another class of accidents, either between two drones or between a drone and the facilities. The severity of such collisions can vary, but they often result in the loss of one or both actors involved during the crash. A common secondary hazard is the uncontrolled descent of debris, which may cause injuries to people on the ground and damage to machinery.

A further accident scenario is premature flight termination, which may occur in either a controlled or uncontrolled manner. In controlled termination, the operator may influence the impact point, as well as the speed and orientation of the aircraft, thereby reducing the likelihood of injuries and limiting damage to both property and the vehicle itself.

The consequences of such accidents may include injuries among operators on the ground, damage or total loss of the drone itself, and harm to surrounding property. Secondary impacts can also arise, such as direct system disruption caused by delay of deliveries, and indirect system issues caused by chemical leakage from the batteries, fire, or the release of hazardous payload materials. This concern is particularly significant for drones employed in applications involving toxic chemicals, such as transporting products for machinery maintenance.

Another often overlooked consequence is the loss of internal acceptance, which can hinder or even stop future drone operations. Such rejection may emerge even in the absence of casualties if accident rates are perceived as unacceptably high.

5.1.1 Ground impact

As a general rule of thumb for aerial vehicles, exposure to ground impact accidents is calculated by the product of the area of a uniformly distributed population, and its density (A-NPA, 2020), as presented in equation 5.1. The population density is typically estimated using the average population density where the drone will operate.

$$N_{exp} = A_{exp} \times \rho \tag{5.1}$$

N_{exp} represents the expected number of individuals exposed within the impact area. This metric is then used to classify operational zones into three risk categories (optimistic, average, pessimistic) and to determine appropriate mitigation strategies for each zone.

The expected impact area, A_{exp}, can be estimated in different ways depending on the crash scenario. In the case of a vertical impact, which is the most probable case in multi-rotor applications, it may be represented by the drone lower frame area, with an added margin to account for the average human body width. Depending on the value of N_{exp}, different strategies may be used to mitigate the risk of damage in the event of an accident. Zones with a higher value may need more robust passive protocols to ensure safe drone operation.

In this regard, the A-NPA 16-2005 (A-NPA, 2020) divides the population density ρ into three categories, based on the risk presented on each scenario when operating a large commercial aircraft. Table 5.1 shows the value for each scenario.

Table 5.1: Population density scenarios

Scenario	Population density [ppl/km^2]
Optimistic	50
Average	200
Pessimistic	5000

Concerning industrial implementations, damage to machinery also represents a significant risk and should therefore be considered part of the assets under exposure, since operations depend directly on machinery. For this reason, approaches that require active emergency protocols should be used only as the last line of defense. In general, ground impacts should be avoided, even if system failure cannot be totally excluded. Passive emergency protocols, such as installing nets or grids beneath the drone's area of operations, can effectively prevent damage to nearby facilities or operators.

5.1.2 Mid-air collision

To establish a proper mid-air collision avoidance protocol, it is first needed to identify two main types of accidents that may occur: incidents where a drone hits external obstacles such as machinery or walls, and collisions that occur between drones while in flight.

However, it is important to highlight that a conflict in flight paths does not always result in a collision, since one or both vehicles may maneuver to preserve the necessary separation and prevent danger. Therefore, the likelihood of a collision is influenced both by the avoidance capabilities of the drone involved and by the obstacle avoidance system in place to ensure adequate spacing.

When a threshold for maximum allowable collision probability is established, it can serve, rather than the limit on acceptable accident frequency, as the basis for defining the minimum performance standards of UAVs avoidance systems.

In this sense, openTCS plays a decisive role in mitigating mid-air collision risks by providing a centralized supervisory architecture that ensures conflict-free routing, scheduling, and resource allocation tasks for drone operations. Unlike fully decentralized flight systems, where individual vehicles may rely solely on onboard perception and local decision-making, openTCS enforces a conflict-free planning paradigm through deterministic coordination of all aerial agents.

Each mission is divided into precomputed trajectories that are assigned exclusively to one vehicle at a time, effectively preventing overlapping flight paths and shared spatial resources. This preemption of shared resources ensures that two different vehicles cannot occupy the same airspace simultaneously, thereby eliminating the most common preconditions for mid-air collisions in multi-agent environments.

Additionally, openTCS integrates real-time monitoring and dynamic task reassignment, which further reduce residual collision risk arising from unforeseen events or drone performance deviations. Through lightweight communication protocols such as MQTT and continuous data exchange with the onboard systems, the supervisor maintains awareness of each drone's position, velocity, and operational state.

When deviations are detected, such as delays, navigation errors, or energy limitations, the system recalculates trajectories or temporarily suspends new task assignments to preserve separation between vehicles. Combined with the drone's onboard obstacle avoidance and PX4 geo-fencing layers, openTCS establishes a hierarchical safety structure in which supervisory coordination operates as the first line of defense, and autonomous avoidance serves as a final contingency layer, which is then ensured by the navigation and operating system within each drone.

This dual redundancy between centralized control and local autonomy enables the near-complete suppression of mid-air collision risk, ensuring that aerial logistics operations remain both safe and predictable even under high system throughput.

5.1.3 System reliability: Risk management framework

The core principle of risk management is to keep system hazards under control or reduce them to an acceptable level. The objective is to minimize, or ideally eliminate, the occurrence of events with serious consequences during system operation. In this context, the risk for any given event is characterized by the combination of the likelihood that the event will occur and the severity of its potential outcomes (Leveson, 1995).

As the main concern is determining whether a given drone can be operated safely, it is required to identify the measures needed to reduce risks. Following this, a drone risk management framework can be defined using four distinct stages:

Mission definition: At this stage, both the mission profile and the system's characteristics are specified, and suitable safety thresholds are established. Potential hazards associated with operating the system are then identified for use in the next step.

Risk evaluation: Each identified hazard, along with possible failure scenarios, is analyzed to determine its associated risk. The evaluation can be refined by applying different models or assumptions as needed.

Risk reduction: The calculated risks are compared against the defined safety thresholds. If risks exceed these limits, they are reduced through adjustments to system design, operational procedures, safety criteria, or risk modeling approaches.

Risk acceptance: Once risks fall within acceptable bounds, drone operation can be authorized. Continuous monitoring during mission execution ensures that the level of risk does not increase.

In certain mission scenarios, risks can be reduced by managing the environment in which the drones operate. One approach is to limit flight to designated areas where population density remains within acceptable levels. Another option is to plan routes that avoid zones with high population or valuable assets. Such areas may include facilities that pose secondary hazards, such as chemical products.

Mission definition

As stated in Chapter 2, the mission profile of the drone delivery system centers on providing fast, flexible, and safe intralogistics operations within manufacturing environments. Unlike rigid ground-based systems, drones enable the rapid transportation of raw materials, tools, and faulty components across production lines, improving system reactivity in scenarios where agility is critical.

To ensure smooth operations, missions are defined in terms of pickup, transport, drop-off, and charging tasks, which are dynamically allocated by a supervisory system. Each mission includes path reservations and resource allocations that prevent interference with other vehicles in the fleet. This means that once a mission is assigned, both the route and the necessary infrastructure, such as landing and charging stations, are exclusively reserved for the active drone until task completion. The resulting workflow ensures that high-priority deliveries are executed with minimal delay, while maintaining conflict-free coordination among multiple drones operating in shared spaces.

The mission also emphasizes safety and predictability. Since drones share operational environments with humans and other autonomous systems, they must operate within strict safety margins. Speed and Separation Monitoring (SSM), geo-fencing, and altitude restrictions are employed to reduce risks of mid-air collisions or unsafe interactions. Furthermore, the integration of transfer-points and position markers along the flight path enhances navigation accuracy, enabling granular control of drone behavior and reliable monitoring of mission execution. Together, these measures establish a mission profile where reliability, efficiency, and safety are the defining characteristics.

The system's characteristics combine the physical capabilities of the Sentinel drone platform by ModalAI with the supervisory and fleet management functionalities of openTCS. The Sentinel drone offers a payload capacity of up to 1 kg and a flight time exceeding 20

minutes, which makes it well-suited for transporting tools and small parts within manufacturing plants. Its PX4-based configuration enables maximum speed, altitude, and geo-fencing limits. As shown in Figure 5.2, it is equipped with multiple image sensors, but also GPS, and IMU modules. Combining them, the drone supports precise localization and navigation, while its Qualcomm QRB5165 processor enables advanced onboard perception and autonomy.

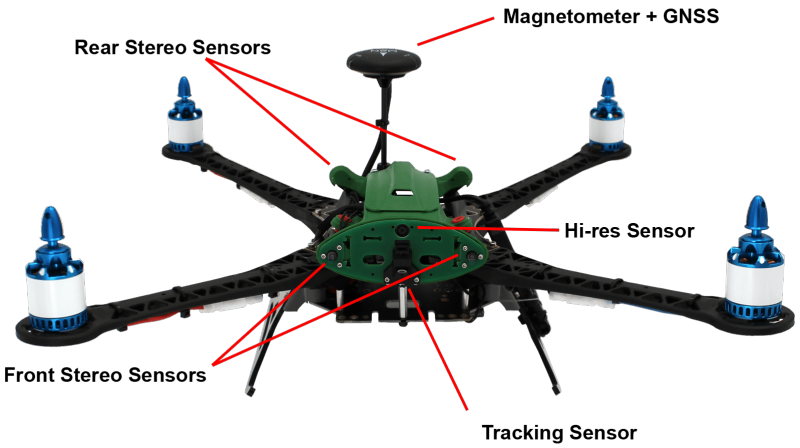


Figure 5.2: Sentinel image sensors

Autonomous navigation is achieved by combining the movement estimations of the rear and front stereo cameras, with the Visual Inertial Odometry (VIO) algorithm used through the tracking sensor. The front and rear stereo cameras are built upon the OV7251 VGA image sensor that uses a global shutter to reduce unwanted image artifacts, which are a usual problem when traditional rolling shutters are used as a result of motion during image capture (OmniVision, 2020). The resolution of the cameras is limited to 640 x 480 pixels, with a framerate up to 60 Hz. The focusing range of the cameras starts at 5cm, while they reach a 85° x 68° x 56° Field Of View (FOV).

Regarding the tracking sensor, it is also built upon the OV7251 VGA image sensor, however it features a fisheye lens to increase FOV, contrary to the stereo cameras which have regular lenses. With it, it can reach a 166° x 133° x 100° FOV, which facilitates the VIO algorithm's ability to track features while in motion. The sixth camera, or high resolution sensor, uses a Sony IMX214 image sensor at a maximum resolution of 4208 x 3120 pixels, working with a rolling shutter, contrary to the rest of the cameras. This sensor task is restricted to acquiring images or recording video.

Two separate TDK ICM42688p IMU's are used to avoid interference and overload within the system. One IMU is connected and used by the PX4, while the second one is used by the VIO algorithm. With it, data processing can be dedicated to the different

needs of each application. Similarly, two independent barometers are incorporated and connected to the PX4 to provide redundancy, enhance accuracy and stability, enable fault detection and isolation, and support environmental compensation. While not active under normal conditions, they remain available for use in extreme scenarios.

Position estimates and error correction are performed by using a state-of-the-art Machine Vision Visual Inertial Simultaneous Localization and Mapping (MVVISLAM), originally developed by Qualcomm (Singh, 2019). This algorithm relies on the images of the tracking sensor and the data from the IMU to estimate the 6DoF pose of the drone.

The main difference between a MVVISLAM approach and a traditional VISLAM is that the former incorporates machine vision techniques to extend data processing, object detection and classification, and object tacking/navigation, while simultaneously combining it with the data collected from the IMU using an extended Kalman filter to fuse both measurements (Singh, 2019).

Furthermore, a relocalization filter developed by ModalAI fuses the data measured through the MVVISLAM with that obtained from secondary sources, such as the rear and front stereo cameras, as well as from the GPS and fiducial markers if implemented. To refine the outputs provided by the secondary localization source, a moving average filter is also applied. This filter operates by collecting a predefined number of measurements from the secondary source, averaging them, and then integrating the result with the MVVISLAM estimates.

Obstacle detection and avoidance is then achieved using the stereo cameras, which can provide a depth image of the environment. Figure 5.3 shows the navigation and localization architecture of the proposed drone for indoor integration based on the previously described components.

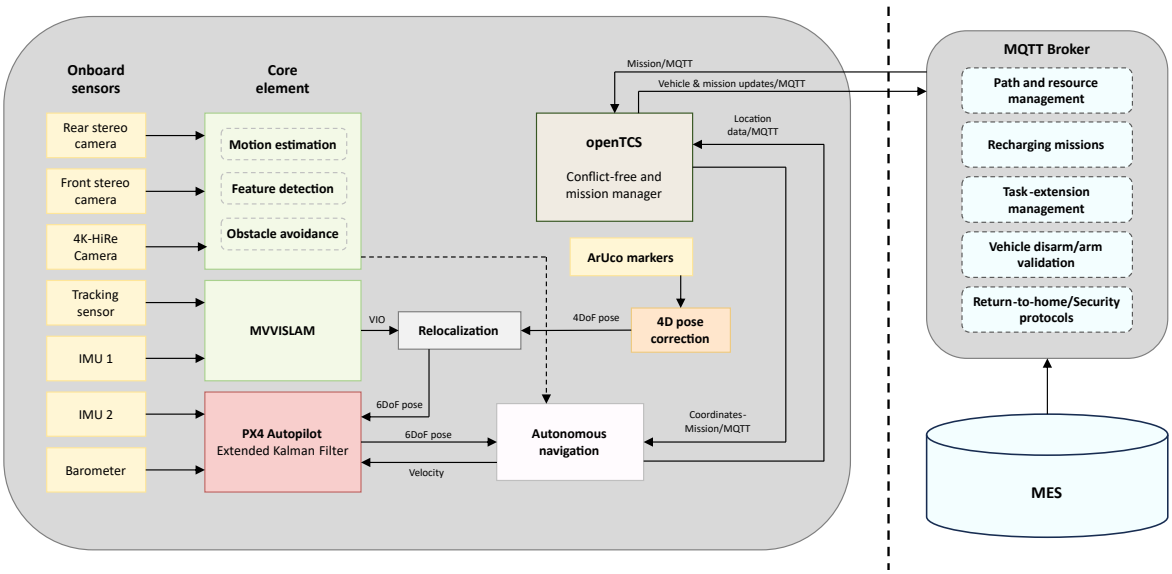


Figure 5.3: UAV navigation system architecture

As demonstrated in the work of (Dias et al., 2024), using a navigation system based on a VOXL2, such as the one integrated in the Sentinel, the pose estimation error can be as low as 7.75 cm compared to ground truth. In fact, the fusion of the different sensors to relocate the drone within the environment makes the system reliable for industrial applications. By calculating the mean and maximum pose estimation errors, it can be ensured that SSM protocols and pathing consider this drift, hence safe navigation is ensured. Table 5.2 shows the values obtained from the pose estimations of the VOXL2, compared to an OptiTrack Prime 22 motion capture system which was in charge of tracking the drone pose ground truth.

On the software side, the drone runs open-source packages such as ROS, OpenCV, and TensorFlow Lite, allowing for flexibility in mission programming and the integration of computer vision for obstacle avoidance. Communication with the supervisory system is supported via Wi-Fi, or 5G, ensuring low-latency transmission of telemetry and control data. Safety is reinforced by the PX4 safety configuration and failsafe mechanisms, while energy management is handled by monitoring the drone's 5000 mAh battery and rerouting missions when recharge thresholds are reached. This ensures that drones can complete their tasks without risking mid-mission power loss.

Complementing the drone's capabilities, openTCS acts as the supervisory and coordination layer. It manages task scheduling, path planning, and conflict resolution through its modular architecture, which includes dispatcher, router, and scheduler components. By reserving paths and resources, openTCS ensures that drones never enter into resource or collision conflicts during missions. Communication between drones and openTCS is achieved via MQTT, a lightweight publish–subscribe protocol optimized for robotics, while custom routing modules incorporate drone-specific considerations such as battery capacity and payload constraints. This dual-system design, pairing the physical platform with a robust supervisory framework, creates a mission-ready drone delivery system capable of safe, reliable, and scalable intralogistics operations.

Risk evaluation

The initial stage of risk assessment involves identifying hazards associated with the system conditions that could potentially lead to an accident. Once these hazards are recognized,

Table 5.2: VIO estimated error using Sentinel onboard navigation system (Dias et al., 2024)

	X (cm)	Y (cm)	Z (cm)	Yaw (deg)	3D (cm)
Mean error	3.67	4.40	4.38	1.2728	7.75
Maximum error	19.42	11.32	21.20	7.9450	21.40

their probability of occurrence is estimated. The overall risk of a hazard is then defined as the combination of its potential severity and the likelihood that it will occur.

Then, a direct cause-consequence relationship can be obtained by balancing the frequency with which adverse events can occur and the severity of their consequences. This balance is illustrated in the cause-consequence matrix in Table 5.4, which classifies risk according to four likelihood levels and five levels of potential severity as described by Stolzer et al. (2023).

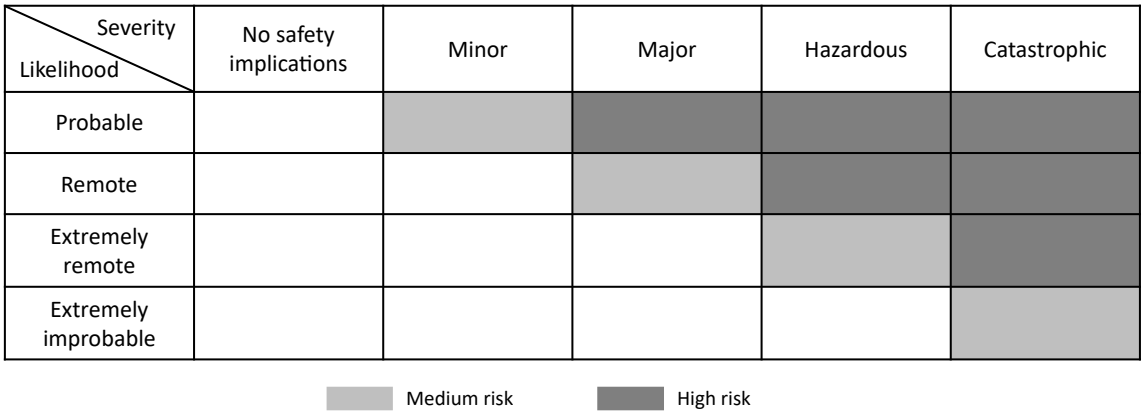


Figure 5.4: Cause-Consequence relationship

According to the Federal Aviation Administration (FAA) (FAA, 2022), the different type of severities are defined as follows:

No safety implications: Has no effect on safety

Minor: Effects may account for slight reduction in system functional capabilities, slight increase in workload, minor environmental damage, and/or minor property damage

Major: Diminishes the ability of the system or its operators to handle unfavorable operating conditions. In addition to having significant effects on the system and the environment, it includes injuries to humans.

Hazardous: It accounts for a large reduction in system functional capability, workload increase such that the system cannot perform the required tasks accurately or completely, and serious injuries to a single person involved on the accident.

Catastrophic: Results in multiple serious injuries to operators and total loss of the system.

Overall, the most serious outcomes of indoor drone operations can lead to multiple injuries to operators, which should be classified as hazardous. Accidents that primarily result in property damage or environmental harm are categorized as major. Since this

analysis focuses on critical hazards, events rated as minor or with no safety implications can be excluded from consideration.

The second element of risk assessment, as shown in Table 5.4, is the likelihood of an event occurring. This likelihood is expressed as the average probability that the event occurs per hour of operation of the system (FAA, 2022). According to the FAA (2022), four categories of likelihood can be used within the evaluation process:

Probable: Expected to occur at least once during the full operational lifetime of the system or agent.

Remote: Not expected to occur for an individual agent over its lifetime, though it may happen several times across the entire system or fleet.

Extremely remote: Not expected to occur for an individual agent over its lifetime, but may happen a few times across the overall system or fleet.

Extremely improbable: Considered so improbable that it is not expected to occur throughout the operational life of the system or fleet.

Based on this framework, it can be obtained the risk assessment shown in Figure 5.5. Since the drone delivery system works under a supervisory control managed by openTCS focusing on obstacle-free routing, mid-air crashes, as well as battery depletion risk is reduced to an extremely improbable scenario. However, there is still the possibility of drone malfunction, which may lead to ground impact in case of lost control. Problems related to external sources, such as payload drop while in-flight, or any leakage in case of transporting liquids, represent remote risks that should also be addressed. Furthermore, it is important to highlight that emergency landings protocols may not represent a hazardous risk to humans or the system, however, special landing areas should be predefined, so that the drone does not suddenly land over a populated area or machinery.

Severity / Likelihood	No safety implications	Minor	Major	Hazardous	Catastrophic
Probable	• Lost connection • Wrong mission	• Emergency landing at an unplanned location • Payload damage			
Remote		• System deadlock	• Components/payload leakage	• Payload drop • Drone unsafe landing	
Extremely remote				• Drone unsafe arm • Mid-air property crash	• Drone ground impact • Hacked system
Extremely improbable					• Drone mid-air crash • Battery depletion

Figure 5.5: UAV risk assessment

Risk reduction

Drones inherently pose risks to humans and infrastructure when operated over populated regions. A commonly adopted mitigation strategy is the restriction of operational areas to minimize the probability of ground casualties in the event of system failure. By reducing the likelihood of direct impact with individuals, such measures significantly lower the associated safety risk. Several implementation pathways for this strategy exist, each differing in the manner by which drone operations are constrained.

One approach involves restricting drone activity to regions with low population density, thereby establishing operational thresholds based on demographic distribution. This method is currently employed in contexts such as military test flights and experimental UAV trials (Weibel and Hansman, 2006), where flights are confined to restricted airspace or unpopulated test ranges. While effective, these restrictions can also limit UAV deployment in areas where their utility may be greatest.

An alternative approach emphasizes localized risk management through precise navigation and trajectory planning. Advanced guidance systems enable drones to operate over designated low-risk zones, even within urban environments. Flight paths can be constrained to areas such as empty corridors, machinery using safe-guards, or areas with structural sheltering sufficient to protect operators. This method facilitates drone operations in populated areas while simultaneously reducing public exposure to potential harm.

On the other hand, the severity of ground impact from a drone is determined by several factors, including vehicle mass, geometric configuration, impact velocity, and stored fuel energy. Each of these parameters represents a potential lever for risk reduction, as limiting UAV size, mass distribution, fuel capacity, or cruise speed constrains the overall momentum and energy at impact. Such design constraints thus serve as passive mitigation measures, reducing the potential consequences of a ground collision.

Having that the values presented in Table 5.1 are tailored to large commercial aircraft, a new population density classification is proposed following the same principle as the A-NPA 16-2005, but using the Sentinel drone as reference framework. Considering a standard A320 aircraft, the expected ground impact A_{exp} can be calculated as follows:

$$A_{exp} = Wingspan \times Lenght \tag{5.2}$$

$$A_{exp} = 34.1m \times 37.6m$$

$$A_{exp} = 2,025m^2$$

In the same manner, the average expected ground impact of a Sentinel drone can be

estimated using the lower frame area of the drone, added to the estimated error present on the location system, which adds a safety margin. Besides, the human reach C used in equation 3.2 is also accounted for a more realistic expected area of damage.

$$A_{exp} = \text{lower frame area} + \text{pose estimation error} + \text{human reach} \tag{5.3}$$

$$A_{exp} = (0.73m + 0.214m + 0.4m)^2$$

$$A_{exp} = 1.806m^2$$

With this data, Table 5.3 shows the population density values per scenario using the same relationship as in A-NPA (2020), but with the expected ground impact of the Sentinel drone. Given that the Sentinel's impact area (1.806 m^2) is approximately 1,120 times smaller than that of an A320 (2,025 m^2), population density thresholds are adjusted proportionally to maintain equivalent risk levels across scenarios. This scaling preserves the relative risk assessment framework while adapting it to the smaller dimensions of UAV operations.

Table 5.3: Population density scenarios - Aircraft & drone

Scenario	Population density aircraft [ppl/km^2]	Population density drone [ppl/$10m^2$]
Optimistic	50	0.553
Average	200	2.214
Pessimistic	5000	55.360

As stated before, in industrial settings, the population should be evaluated not only in terms of human presence but also in terms of infrastructure that may be damaged, since an accident could compromise production. Figure 5.6 illustrates the different elements proposed for this evaluation as a function of population density.

Using this evaluation framework, the risk index for each zone is then calculated as:

$$Risk_{zone} = \sum_{i}^{n} w_{zone}$$

where w_{zone} is the infrastructure/human weight shown in Figure 5.6. For each scenario described in Table 5.3, the population density values should be considered as the upper limit within a range, having that the intervals for optimistic, average and pessimistic scenarios are: [0, 0.553], (0.553, 2.214], (2.214, 55.36].

Following this, a zone with 1 operator and 2 non-critical machines (total weight = 2) within 10 m^2 yields: $Risk = 2ppl/10m^2$, classifying it as "Average". The inclusion

Weight	Description
1	• Operator • Machinery that can be damaged or contaminated • Chemical or fragile items involved
0.5	• Machinery with safeguards or flat robust surfaces over them
0	• Empty corridors

Figure 5.6: Risk evaluation framework

of moving elements, such as human presence or mobile robots should be considered as the total amount of people/robots that can cross at a given moment that specific zone, regardless if they do it at the same time or one by one at different moments.

Drawing on the layout described in Chapter 3 (Figure 3.3), the risk areas illustrated in Figure 5.7 are identified.

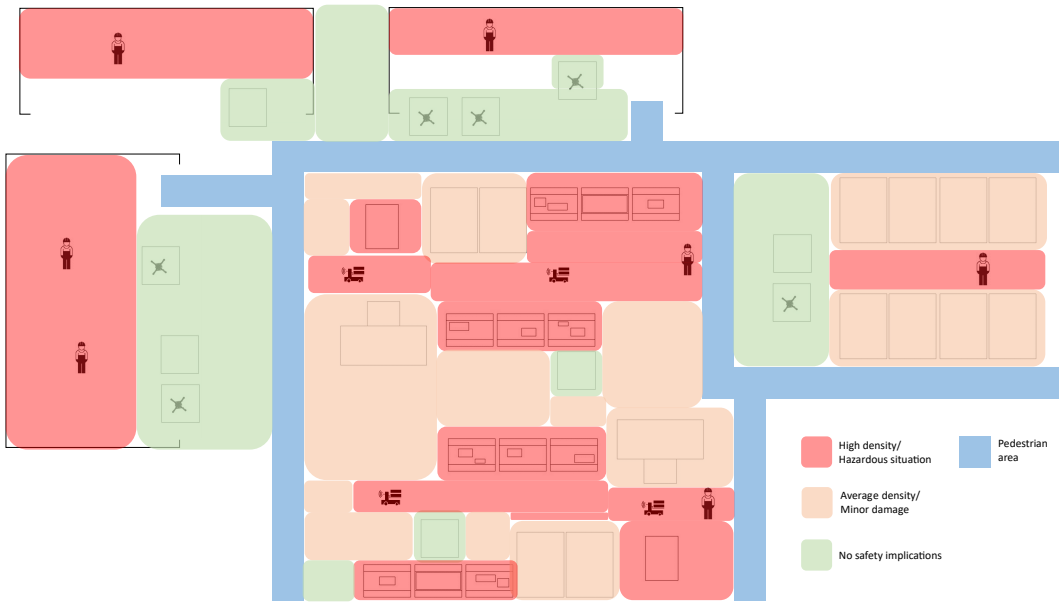


Figure 5.7: UAV risk assessment

Based on this risk map, routing can be prioritized through areas classified as average or optimistic, so that in case of an emergency the drone does not pose a risk to humans or ongoing operations.

In addition to passive strategies, active mitigation measures may be employed to attenuate impact severity. For example a parachute-like device that can be triggered in case of rotor malfunction. However, the implementation of active systems introduces several critical considerations. These measures require reliable detection and triggering mechanisms, an unintended activation may itself create an accident.

As the probability of mid-air collision is directly correlated with the density of air traffic within the operational airspace, the conflict-management algorithm presented in Chapter 4 provides a robust strategy to completely avoid path and resource sharing. With this, mid-air collision is reduced to system malfunction while initiating drone position system, or unexpected obstacle out of vision of the image sensors.

Risk acceptance

Unexpected or premature termination of flight, whether deliberate or uncontrolled, poses one of the greatest risks in drone operations due to its strong association with hazardous outcomes. Failures with the potential to trigger such an event are classified as catastrophic. These must be demonstrated to occur only with extremely low likelihood.

However, even when the highest standards are followed, the risk of malfunction will always be present. Because of this, protocols that mitigate damage to property or harm to humans are mandatory for proper drone operation. In such cases, it may be preferable to permit a drone to land in a controlled manner if that reduces potential harm to people and property. Reflecting this perspective, the Federal Aviation Administration (FAA, 2008) advises that, in any emergency, protecting individuals on the ground takes precedence over preserving the drone itself.

In industrial settings, risk can be managed by allowing drones to fly over optimistic areas without additional restrictions. However, pessimistic areas, as defined in the proposed evaluation framework, should be safeguarded if drone operations occur within those boundaries. Even if risk cannot be completely mitigated, the likelihood of harm can be minimized in areas where accidents would otherwise be highly likely to cause injury or damage.

One way to provide complete protection to operators and machinery located within pessimistic areas is to install a protective shield capable of withstanding falling debris. The shield may protect against uncontrolled landings, and it can also function as a landing platform when emergency protocols are activated. Moreover, operators maintain a clear line of sight with the drone.

Taking advantage of the inability of monocular Visual-Inertial SLAM to recognize transparent objects, the shield can be constructed from a durable material that does not interfere with altitude estimation. However, visual fiducial markers must be used to cover the entire field of view of the shield, ensuring that the drone can still localize itself within the environment without measurement inaccuracies.

A second solution to protect high-risk areas is the use of safety nets that can prevent the drone from impacting the ground, as proposed in Perez-Grau et al. (2021). However, this approach requires altitude estimates to be referenced to the ceiling rather than the floor. Visual features from the nets may cause inaccurate measurements of the drone's

distance from the floor, thereby reducing system reliability. As demonstrated in their work, using the ceiling as a plane for altitude estimation is advantageous in most cases, since ceilings are typically flat and unobstructed.

It should be noted that with this approach, emergency landings would not be possible due to the absence of flat landing areas, and the payload may still pass through the net if accidentally dropped. However, it still provides a safe protocol to protect, as much as possible, areas where operators or unshielded machinery would otherwise be at risk of injury or damage.

5.2 Working Situation Model

The proposed working situation model described in Chapter 2 focuses on the perspective of safety and potential hazards. Its purpose is to provide managers with both technical and methodological support in evaluating design choices and decisions, with a particular focus on managing risks linked to operational factors. This conceptual framework seeks to account for drone integration and production system behavior during the organizational fit evaluation.

The working situation includes not only the system and relevant actors, but also other relevant elements, such as tasks, auxiliary systems, work areas, and potential risks, needed to design reactive protocols in case of emergency. In this context, the modelling of a working situation should contain three main categories:

Physical elements: such as the system, the work team, and auxiliary resources.

Hazards and risks: including risk, dangerous zones, hazardous events, and hazardous phenomena.

Linking concepts: the task to be carried out, which creates an interaction between the system, the work team, and the intervention method; or the working situation, which establishes connections among several elements.

As shown in Appendix A, the proposed working situation model can be used to define potentially hazardous events, measure their severity, and hence be used to trigger emergency protocols when the hazardous event can be controlled.

For this, the risk assessment framework described in section 5.1.3 can be used to define the severity impact by adopting the five categories for severity described above (0 for no safety implications and 4 for a catastrophic evaluation). Furthermore, the frequency used follows the same principle, where probable events are rated as 4 within the evaluation criteria, while extremely improbable would take a value of 1. The risk index is then the product of the severity multiplied by the frequency.

The avoidance potential is measured as the addition of three factors: system, drone, and operator. If a risk can be mitigated following a well-defined protocol triggered by the system then the risk avoidance is equal to 1. If the drone can implement strategies to ensure operator and property safety, then another unit is added to the evaluation. And finally, if operators behavior influence the possibility for a risk, then changing this behavior would be translated in a safer operation, hence risks can be avoided, giving a maximum value for risk avoidance of 3.

It is important to note that, depending on the risk index and the intervention mode, operators should be aware of the tasks they are expected to follow to mitigate the hazardous situation to resume the normal flow of the process.

The working situation model would then be the backbone to describe the interaction between the different agents of the system, and how they should react in case of hazardous event. In the current work, agents involve not only human resources, but also the system needed to operate the drone. In most cases, operators should be only linked within the "intervention mode" of the working situation model. Hence, reactive protocols should be entirely manged by the autonomous vehicle or the supervisor.

5.2.1 Defining safety protocols

Having defined the working situation model, the next step on the safety management framework for risk acceptance is to define the hazards associated with each system, and link them to a potential failure condition. In conducting this analysis, multiple dimensions of system behavior must be considered. These include:

- The potential for a single malfunction within a component to impact other functions executed by that same element.
- The extent to which the failure of one system function may propagate and compromise interdependent or related functions.
- The contribution of latent, low-severity, and undetected faults to the initiation of subsequent failures.
- The mitigating influence of compensatory mechanisms, such as redundancy, backup subsystems, or alternative functional pathways.
- The amplifying influence of adverse factors, including elevated workload, unfavorable environmental conditions, or the absence of effective failure detection and notification mechanisms.

Similar to a FMEA analysis, the objective is to establish suitable reliability requirements for individual components and to identify complementary strategies for hazard

mitigation. While designing for minimal risk is always the preferred approach, certain situations may need the implementation of safety or warning devices that can give a feedback in case of system failure. Regardless of the strategy employed, it remains essential that the associated hazards are systematically identified.

As long as the drone system remains stable, and accidents stay low following the risk assessment framework, the objective in this context is not to design a new system, but rather to identify the hazards intrinsic to the operation and testing of a specific drone implementation. Accordingly, the analysis focuses on recognizing distinct hazardous conditions, tracing their potential causes, and implementing risk-reduction measures as necessary to ensure that overall risk remains within acceptable limits for risk acceptance.

Following the recommendation of the Range Commanders Council (1999) for unmanned aerial vehicles, hazardous conditions (and their causes) can be classified in 4 groups:

Uncontrolled flight

Condition	Possible causes
Propulsion loss	rotor failure, controller failure
Lift loss	altitude sensors, current draw failure
Attitude loss	attitude/navigation system failure
Connection loss with supervisor	transmitter, computer/communication protocol failure
Battery depletion	excessive load from payload, non-optimal routing

Ground impact

Condition	Possible causes
Altitude error	mission, sensor failure; fly below/over safe routing altitude
Attitude error	barometer failure, controller failure
Navigation error	visual odometry failure, measurement discrepancy
Obstacle avoidance error	feature recognition failure, depth measurement discrepancy
Battery depletion	excessive load from payload, non-optimal routing

Take-off or landing

Condition	Possible causes
Landing system error	Positioning failure, location mismatch
Drone oscillation	controller failure
Take-off/landing not triggered	system failure

Mid-air collision

Condition	Possible causes
Navigation error	measurement discrepancy
Altitude error	controller failure, sensors failure
Obstacle avoidance error	feature recognition failure, depth measurement discrepancy
Resource reservation error	supervisor failure

Based on this criteria and with the help of the working situation model, emergency protocols can be triggered in the event of an unavoidable risk. As stated before, all the agents involved in drone operations should be defined within the situation modeling so that each element can follow a certain task. Appendix I describes the main algorithms to ensure safety in case of an unplanned event based on the previous classification.

The benefit of splitting supervisory tasks for resource assignment with navigation functionalities is that vehicles do not depend on a external source to reach a safe state. Besides, drone routing is not based on computer-expensive algorithms that evaluate 3D environments, so the supervisor can react on real-time to any change in the system.

Ensuring conflict-free zones for drone flight completely diminishes one of the highest risks in drone operations, which simplifies the protocols needed for safe deployment while the drone is active. While the working situation model works as a guideline and evaluator to know how to react to different events that may lead to hazardous situations. Combining the three elements: a reliable navigation system of the drone based on a VOXL2, the custom fleet supervisor built upon openTCS, and the working situation model focused on risk management, companies can ensure safe operations and enhance the intralogistics.

5.3 Conclusion

This chapter highlighted the central role of risk management in the adoption of drone-based intralogistics, emphasizing that safe integration into industrial environments requires a structured evaluation framework. By drawing from established aviation standards, such as the Equivalent Level of Safety (ELOS), it was shown that while direct transfer of commercial aviation criteria is impractical, equivalent principles can be adapted to smaller UAV systems. Central to this process is the recognition that hazards must be managed within acceptable thresholds, ensuring that drone operations do not compromise human safety, infrastructure integrity, or organizational acceptance.

The discussion identified three main classes of potential accidents: unintended mobility, mid-air collisions, and premature flight termination. It further assessed their consequences in terms of human injury, property damage, and system disruption. Among these, mid-air collisions and uncontrolled flight termination were found to carry the most hazardous outcomes, requiring both passive and active mitigation strategies. Risk quantifica-

tion, including exposure modeling through ground impact analysis, provides a structured way to assess accident probabilities and consequences.

The chapter further emphasized that mission definition plays a critical role in ensuring safe operations. Intralogistics drone missions are built around pickup, transport, drop-off, and charging phases, with strict conflict-avoidance protocols supported by supervisory systems. Safety margins such as Speed and Separation Monitoring (SSM), geo-fencing, and altitude restrictions ensure that drones coexist safely with human operators and automated systems. By embedding safety constraints directly into the mission profile, the system strengthens predictability and operator trust.

At the system level, the ModalAI Sentinel platform provides both the physical robustness and technological sophistication necessary for industrial operations. Its payload capacity, endurance, and image sensors integrated in the VOXL2-based MVVISLAM navigation system, enable accurate, redundant, and reliable positioning within dynamic environments. These features allow compliance with SSM protocols, ensuring that localization errors remain within safe limits. Together with redundant barometers, IMUs, and failsafe configurations, the Sentinel's hardware is designed to maintain operational stability even under degraded conditions.

On the supervisory side, openTCS complements the drone platform by managing fleet-level operations. Its modular routing and scheduling framework ensures conflict-free task execution and resource allocation, while lightweight communication protocols such as MQTT enable reliable integration with onboard systems. When combined with the working situation model, which maps risks, hazards, and mitigation strategies, openTCS ensures that operational and organizational factors are addressed simultaneously. This dual-layer design—merging technical reliability with supervisory coordination—provides a holistic foundation for safe and efficient drone adoption.

Ultimately, the chapter demonstrated that successful adoption of drones in intralogistics depends not only on technical capability but also on behavioral and organizational acceptance. By integrating risk assessment, mission definition, and system-level redundancy into a coherent framework, companies can achieve safe, scalable, and reliable drone operations. The convergence of Sentinel's advanced onboard autonomy, openTCS supervision, and risk-management operational models ensures that industrial drone delivery systems can transition from experimental deployments to trusted components of Industry 4.0 ecosystems.

General conclusion

The increasing complexity and variability of modern manufacturing have redefined the role of internal logistics, requiring systems that can adapt rapidly to fluctuating production demands. Traditional ground-based solutions such as AGVs and conveyors, though efficient in stable environments, struggle to deliver the flexibility and responsiveness needed in agile production systems.

Within this evolving context, drones emerge as a promising complement, offering a new operational dimension that exploits unused aerial space to achieve fast-paced deliveries. This research has systematically examined the potential of drones in manufacturing intralogistics, addressing not only their technical feasibility but also the strategic, operational, and organizational conditions necessary for successful integration.

The first part of this study situates drone-based intralogistics within the broader transformation driven by Industry 4.0 and Smart Manufacturing. It highlights how digitalization, cyber-physical systems (CPS), and the Industrial Internet of Things (IIoT) have paved the way for interconnected, intelligent, and adaptive production systems.

Within this context, drones represent a technological next step toward manufacturing agility, serving as mobile agents capable of collecting, transporting, and sharing information and materials across networked environments. The conceptual framework developed in the state-of-the-art review establishes the theoretical foundation upon which subsequent chapters build practical models for drone integration.

From a technical standpoint, the formalization of the drone integration problem has demonstrated that aerial intralogistics require complex planning and system supervision. Drones' limited energy resources, their inability to hover for long periods, and their reliance on clear, conflict-free routes demand a two-stage optimization approach encompassing both pre-computed trajectories and dynamic, real-time supervision.

This study shows that the combination of autonomous navigation and centralized management through online supervisors and resource reservation protocols forms the cornerstone of safe and efficient operations. By introducing mathematical representations of drone delivery systems and conflict-free path planning, the research bridges theoretical optimization with practical deployment strategies suitable for agile manufacturing contexts.

The economic and strategic analyses underscore that drone adoption extends far beyond technological readiness. Decision-makers must evaluate the cost-benefit ratio of implementation, account for human-drone interaction (HDI) safety, and ensure that operational gains align with corporate objectives. Simulation-based assessments play a vital role in this pre-adoption phase, allowing stakeholders to model stochastic production environments and measure drone performance under variable demand.

Through realistic simulation tools, companies can evaluate key performance indicators, validate system resilience, and optimize workflows before full-scale implementation. The study therefore provides a practical decision-support framework that combines simulation, safety assessment, and strategic planning for drone adoption in Industry 4.0 ecosystems.

In the early adoption stage, the work introduces a conflict-management algorithm that enables coordinated and reliable drone operations. Centralized supervision and proactive conflict resolution mechanisms, supported by real-time localization and trajectory updates, ensure that drones can coexist safely within shared industrial spaces.

The integration of transfer points, position markers, and communication protocols such as MQTT enhances precision and synchronization across fleets, bridging the gap between theoretical control systems and their industrial application. This approach demonstrates that UAV fleet management is not merely a technical challenge but a systems-engineering problem that requires synchronization between hardware, software, and operational policies.

Finally, the study emphasizes that sustainable drone integration depends on the interplay between technical performance, organizational readiness, and behavioral acceptance. The adaptation of aviation safety principles, such as the Equivalent Level of Safety (ELOS) to indoor industrial settings provides a structured foundation for risk assessment and mitigation. The dual-layer system combining the Sentinel platform and the openTCS supervisory framework ensures redundancy, predictability, and safety at both the hardware and system-management levels.

By aligning risk management with mission definition and system supervision, the research establishes a comprehensive roadmap for the safe and scalable deployment of drone intralogistics. Ultimately, this work concludes that drones, when integrated through structured frameworks encompassing technical optimization, economic evaluation, and organizational adaptation, can significantly enhance manufacturing agility, responsiveness, and resilience, solidifying their role as strategic enablers of the next generation of smart factories.

Appendices

Appendix A

Working Situation Model

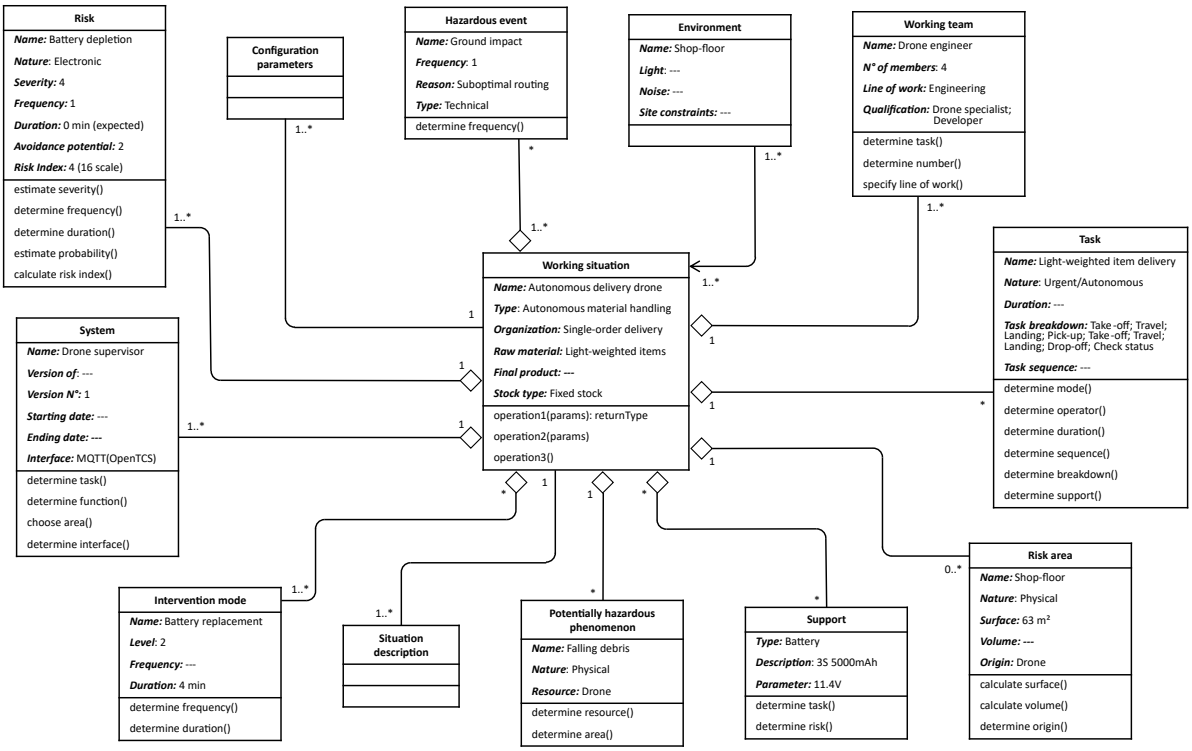


Figure A.1: Working Situation Model based on Hasan (2002)

Appendix B

/vehicle


```
{
    "name": "Vehicle-0001",
    "properties": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "boundingBox": {
      "length": 60,
      "width": 40,
      "height": 50,
      "referenceOffset": {
        "x": 1500,
        "y": 2000
      }
    },
    "energyLevelGood": 90,
    "energyLevelCritical": 30,
    "energyLevelSufficientlyRecharged": 30,
    "energyLevelFullyRecharged": 90,
    "energyLevel": 60,
    "integrationLevel": "TO_BE_IGNORED",
    "paused": false,
    "procState": "UNAVAILABLE",
    "procStateTimestamp": {},
    "transportOrder": "TOrder-01",
    "currentPosition": "Point-0001",
    "precisePosition": {
      "x": 60,
      "y": 40,
      "z": 0
    },
    "orientationAngle": 90,
    "state": "UNKNOWN",
    "stateTimestamp": {},
    "allocatedResources": [
      [
        "Path-0039--0040",
        "Point-0040"
      ],
      [
```

```
        "Path-0040--0041",
        "Point-0041"
      ]
    ],
    "claimedResources": [
      [
        "Path-0041--0042",
        "Point-0042"
      ],
      [
        "Path-0042--0043",
        "Point-0043",
        "Location-2345"
      ]
    ],
    "acceptableOrderTypes": [
      {
        "name": "Park",
        "priority": 0
      },
      {
        "name": "Load cargo",
        "priority": 0
      }
    ],
    "envelopeKey": "envelopeType-01"
  }
```

Appendix C

build.gradle

```
{
plugins {
    id 'java'
}

group = 'org.opentcs.sentinel'
version = '1.0'

java {
    toolchain {
        languageVersion = JavaLanguageVersion.of(21)
    }
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.5'
    implementation 'com.google.inject:guice:7.0.0'
    implementation 'jakarta.inject:jakarta.inject-api:2.0.1'
    implementation project(':opentcs-api-base')
    implementation project(':opentcs-common')
    implementation project(':opentcs-kernel')
}




tasks.withType(JavaCompile).configureEach {
    options.encoding = 'UTF-8'
}
```

Appendix D

SentinelCommAdapter.java

```
{
package org.opentcs.sentinel;

import org.eclipse.paho.client.mqttv3.IMqttMessageListener;
import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;

import org.opentcs.data.order.TransportOrder;
import org.opentcs.drivers.vehicle.BasicVehicleCommAdapter;
import org.opentcs.drivers.vehicle.MovementCommand;
import org.opentcs.util.ExplainedBoolean;

import javax.annotation.Nonnull;
import java.nio.charset.StandardCharsets;
import java.util.concurrent.ScheduledExecutorService;

public class SentinelCommAdapter extends BasicVehicleCommAdapter {

  private MqttClient mqttClient;
  private static final String MQTT_BROKER = "tcp://localhost:1883";
  private static final String BATTERY_LEVEL_TOPIC = "sentinel/soc";

  public SentinelCommAdapter(@Nonnull SentinelProcessModel processModel,
                             @Nonnull ScheduledExecutorService executor) {
    super(processModel, 1, "SENTINEL-Adapter", executor);
  }

  @Override
  public void connectVehicle() {
    try {
      String clientId = "SentinelClient-" + getProcessModel().getVehicle().getName();
      mqttClient = new MqttClient(MQTT_BROKER, clientId);
      mqttClient.connect();

      mqttClient.subscribe(BATTERY_LEVEL_TOPIC, new IMqttMessageListener() {
        @Override
        public void messageArrived(String topic, MqttMessage message) {
          handleBatteryLevelMessage(new String(message.getPayload(), StandardCharsets.
              UTF_8));
        }
```

```
    });

    getProcessModel().setVehicleConnected(true);
  } catch (MqttException e) {
    e.printStackTrace();
    getProcessModel().setVehicleConnected(false);
  }
}

private void handleBatteryLevelMessage(String payload) {
  try {
    double soc = Double.parseDouble(payload);
    int batteryPercent = (int) (soc * 100);

    // Update battery level inside the process model
    getProcessModel().setBatteryLevel(batteryPercent);

    System.out.println("Battery level updated to " + batteryPercent + "% for vehicle "
                       + getProcessModel().getVehicle().getName());
  } catch (NumberFormatException e) {
    System.err.println("Invalid battery level format: " + payload);
  }
}


//@Override
public boolean supportsSetVehiclePosition() {
  return true;
}




@Override
public ExplainedBoolean canProcess(@Nonnull TransportOrder order) {
  return new ExplainedBoolean(true, "Can process order");
}

@Override
public void disconnectVehicle() {
  try {
    if (mqttClient != null && mqttClient.isConnected()) {
      mqttClient.disconnect();
    }
  } catch (MqttException e) {
    e.printStackTrace();
  }
  getProcessModel().setVehicleConnected(false);
}

@Override
public boolean isVehicleConnected() {
  return getProcessModel().isVehicleConnected();
}


@Override
public void sendCommand(MovementCommand command) {
```

```
    getProcessModel().commandExecuted(command);
  }

  @Override
  public void processMessage(Object message) {
    // Optional: handle other messages if needed
  }

  @Override
  public void onVehiclePaused(boolean paused) {
    // Optional: handle pause state
  }

  @Override
  public SentinelProcessModel getProcessModel() {
    return (SentinelProcessModel) super.getProcessModel();
  }
}
```

Appendix E

SentinelCommAdapterDescription.java

```
{
package org.opentcs.sentinel;

import org.opentcs.drivers.vehicle.VehicleCommAdapterDescription;

public class SentinelCommAdapterDescription extends VehicleCommAdapterDescription {

  @Override
  public String getDescription() {
    return "Sentinel MQTT Vehicle Adapter";
  }

  @Override
  public boolean isSimVehicleCommAdapter() {
    return false;
  }
}
```

Appendix F

SentinelCommFactory.java

```
{
package org.opentcs.sentinel;

import org.opentcs.data.model.Vehicle;
import org.opentcs.drivers.vehicle.VehicleCommAdapter;
import org.opentcs.drivers.vehicle.VehicleCommAdapterDescription;
import org.opentcs.drivers.vehicle.VehicleCommAdapterFactory;

import javax.annotation.Nonnull;
import java.util.concurrent.Executors;

public class SentinelCommAdapterFactory implements VehicleCommAdapterFactory {

  private boolean initialized;

  @Override
  public boolean providesAdapterFor(@Nonnull Vehicle vehicle) {
    return true;  // Customize as needed
  }

  @Override
  public VehicleCommAdapter getAdapterFor(@Nonnull Vehicle vehicle) {
    return new SentinelCommAdapter(
      new SentinelProcessModel(vehicle),
      Executors.newSingleThreadScheduledExecutor()
    );
  }

  @Override
  public VehicleCommAdapterDescription getDescription() {
    return new SentinelCommAdapterDescription();
  }

  @Override
  public void initialize() {
    initialized = true;
  }

  @Override
  public boolean isInitialized() {
    return initialized;
```

```
  }

  @Override
  public void terminate() {
    initialized = false;
  }
}
```

Appendix G

SentinelCommAdapterModule.java

```
{
package org.opentcs.sentinel;

import org.opentcs.customizations.kernel.KernelInjectionModule;

public class SentinelCommAdapterModule extends KernelInjectionModule {
  @Override
  protected void configure() {
    vehicleCommAdaptersBinder()
        .addBinding()
        .to(SentinelCommAdapterFactory.class);
  }
}
```

Appendix H

SentinelProcessModel.java

```
{
package org.opentcs.sentinel;

import org.opentcs.data.model.Vehicle;
import org.opentcs.drivers.vehicle.VehicleProcessModel;
import org.opentcs.drivers.vehicle.MovementCommand;

public class SentinelProcessModel extends VehicleProcessModel {

  private boolean vehicleConnected;
  private final Vehicle vehicle;
  private int batteryLevel;

  public SentinelProcessModel(Vehicle vehicle) {
    super(vehicle);
    this.vehicle = vehicle;
  }

  public boolean isVehicleConnected() {
    return vehicleConnected;
  }

  public void setVehicleConnected(boolean connected) {
    this.vehicleConnected = connected;
  }

  public Vehicle getVehicle() {
    return vehicle;
  }

  public void commandExecuted(MovementCommand command) {
    super.commandExecuted(command);
  }



  public int getBatteryLevel() {
    return batteryLevel;
  }

  public void setBatteryLevel(int batteryLevel) {
```

```
    this.batteryLevel = batteryLevel;
    // Also update the vehicle model if needed, notify observers, etc.
  }

}
```

Appendix I

Emergency protocols

Algorithm 4 SUPERVISOR_BOUNDARY_CHECK Method

Input: $sentinel/VIO$ – continuous VIO coordinate updates from each active drone
Parameters: A, B – boundary segment endpoints; r – corridor radius; N – number of consecutive violations before trigger
Output: $status$ – INSIDE if drone within corridor, OUTSIDE if violation occurs
for each $drone \in ActiveDrones$ **do**
    if ReceiveUpdate($sentinel/VIO[drone]$) = **true then**
        $P \leftarrow$ GetCoordinates($sentinel/VIO[drone]$) ▷ current position (x,y)
        $d \leftarrow$ DistanceToSegment(P, A, B) ▷ projection + clamp
        if $d \leq r$ **then**
            $violationCount[drone] \leftarrow 0$
            $status[drone] \leftarrow$ INSIDE
            continue ▷ normal operation
        else
            $violationCount[drone] \leftarrow violationCount[drone] + 1$
            if $violationCount[drone] \geq N$ **then**
                $status[drone] \leftarrow$ OUTSIDE
                TriggerHover($drone$) ▷ force hover
                TriggerSafeLanding($drone$) ▷ safe landing
            end if
        end if
    else
        $status[drone] \leftarrow$ DISARM
        NotifyOperator($drone$) ▷ no updates received
    end if
end for

Algorithm 5 DRONE_MONITORING Method

Input: *signalTimer* – MQTT signal verification timer (every 5 seconds)
Output: *status* – ACTIVE if message received, LOST if disconnection occurs
while MQTT.status = ACTIVE **do**
    if VerifySignal(*signalTimer*) = **true then**
        status ← CONNECTED
        continue ▷ keep operating mission
    else
        status ← LOST
        reconnected ← TryReconnect()
        if *reconnected* = **true then**
            status ← CONNECTED
            continue ▷ resume mission
        else
            FinishMission()
            LandSafely()
            DisarmDrone()
            DisableTaskExecution()
            return LOST
        end if
    end if
end while

Algorithm 6 SUPERVISOR_MONITORING Method

Input: *sentinel/VIO* – continuous VIO coordinate updates from each active drone
Output: *status* – CONNECTED if drone updates are received, LOST if disconnection occurs
for each *drone* ∈ *ActiveDrones* **do**
    if ReceiveUpdate(*sentinel/VIO*[*drone*]) = **true then**
        status[*drone*] ← CONNECTED
        continue ▷ normal operation
    else
        status[*drone*] ← LOST
        reconnected ← TryReconnect(*drone*)
        if *reconnected* = **true then**
            status[*drone*] ← CONNECTED
            continue ▷ resume monitoring
        else
            FlagDroneLost(*drone*)
            RestrictOrderAssignments() ▷ no new missions allowed
            NotifyOperator(*drone*)
        end if
    end if
end for
if OperatorValidation() = **true then**
    EnableOrderAssignments()
end if

Bibliography

A-NPA, N. (2005). 16/2005, policy for unmanned aerial vehicle (uav) certification. *Available at: https://www.easa.europa.eu/document-library/notices-of-proposed-amendments/npa-16-2005*.

A-NPA, N. (2020). 16/2005, policy for unmanned aerial vehicle (uav) certification, european aviation safety agency–easa, koln, germany,[online]. 2005.[date of reference february 4th of 2020]. *Available at: https://www. easa. europa. eu/document-library/notices-of-proposed-amendments/npa-16-2005*.

Abrate, F., Bona, B., and Indri, M. (2007). Experimental ekf-based slam for mini-rovers with ir sensors only. In *EMCR*.

Adler, P. (1985). Managing flexibility: A selective review of the challenges of managing the new production technologies' potential for flexibility. *Report to the Organizational for Economic Co-operation and Development*, page 36.

Agárdi, A., Kovács, L., and Bányai, T. (2019). Vehicle routing in drone-based package delivery services. In *Solutions for Sustainable Development*, pages 151–159. CRC Press.

Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.

Askerbekov, D., Garza-Reyes, J. A., Ghatak, R. R., Joshi, R., Kandasamy, J., and de Mattos Nascimento, D. L. (2024). Embracing drones and the internet of drones systems in manufacturing–an exploration of obstacles. *Technology in Society*, 78:102648.

Bagheri, B., Yang, S., Kao, H.-A., and Lee, J. (2015). "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment". *IFAC-PapersOnLine*, 48(3):1622–1627.

Baheti, R. and Gill, H. (2011). "Cyber-physical systems". *The impact of control technology*, 12(1):161–166.

Balakrishnan, K. (1996). *Exponential distribution: theory, methods and applications*. CRC press.

Balamurugan, G., Valarmathi, J., and Naidu, V. (2016). Survey on uav navigation in gps denied environments. In *2016 International conference on signal processing, communication, power and embedded system (SCOPES)*, pages 198–204. IEEE.

Baldwin, C. Y. (1991). How capital budgeting deters innovation—and what to do about it. *Research-Technology Management*, 34(6):39–45.

Belingardi, G., Heydaryan, S., Chiabert, P., et al. (2017). Application of speed and separation monitoring method in human-robot collaboration: industrial case study. In *17th International Scientific Conference on Industrial Systems*. 17th International Scientific Conference on Industrial Systems.

Belkadi, F. (2006). *Contribution au pilotage des compétences dans les activités de conception : De la modélisation des situations à la caractérisation des compétences*. PhD thesis, Université de Franche-Comté.

Bouman, P., Agatz, N., and Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4):528–542.

Boysen, N., Briskorn, D., Fedtke, S., and Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, 72(4):506–527.

Boysen, N., Fedtke, S., and Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, 43(1):1–58.

Bukova, B., Brumercikova, E., Cerna, L., and Drozdziel, P. (2018). The position of industry 4.0 in the worldwide logistics chains. *LOGI–Scientific Journal on Transport and Logistics*, 9(1):18–23.

Campbell, J., Sukthankar, R., Nourbakhsh, I., and Pahwa, A. (2005). A robust visual odometry and precipice detection system using consumer-grade monocular vision. In *Proceedings of the 2005 IEEE International Conference on robotics and automation*, pages 3421–3427. IEEE.

Carlsson, J. G. and Song, S. (2018). Coordinated logistics with a truck and a drone. *Management Science*, 64(9):4052–4069.

Carter, C. (1979). Transfer processes in technical change. *The Economic Journal*, 89(356):970–972.

Cedillo-Campos, M. and Sánchez-Ramírez, C. (2013). Dynamic self-assessment of supply chains performance: an emerging market approach. *Journal of applied research and technology*, 11(3):338–347.

Chang, Y. S. and Lee, H. J. (2018). Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications*, 104:307–317.

Cheng, C., Adulyasak, Y., and Rousseau, L.-M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, 139:364–387.

Chew, W. B., Leonard-Barton, D., and Bohn, R. E. (1991). Beating murphy's law.

Chiang, W.-C., Li, Y., Shang, J., and Urban, T. L. (2019). Impact of drone delivery on sustainability and cost: Realizing the uav potential through vehicle routing optimization. *Applied energy*, 242:1164–1175.

Chiarini, A., Belvedere, V., and Grando, A. (2020). "Industry 4.0 strategies and technological developments. an exploratory research from italian manufacturing companies". *Production Planning & Control*, 31(16):1385–1398.

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Haghnazar, R., Zuidema, D., Lima, M. S., and da Costa, A. R. (2017). A multi-objective green uav routing problem. *Computers & Operations Research*, 88:306–315.

Cole, D. M. and Newman, P. M. (2006). Using laser range data for 3d slam in outdoor environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1556–1563. IEEE.

Council, R. C. (1999). Range safety criteria for unmanned air vehicles. *Series Range Safety Criteria for Unmanned Air Vehicles. White Sands Missile Range, NM: Range Safety Group of the Range Commander's Council at the White Sands Missile Range*.

Dabić-Miletić, S. and Raković, K. (2023). Ranking of autonomous alternatives for the realization of intralogistics activities in sustainable warehouse systems using the topsis method. *Spectrum of Engineering and Management Sciences*, 1(1):48–57.

Dalamagkidis, K., Valavanis, K. P., and Piegl, L. A. (2009). *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*. Springer.

Das, A. and Narasimhan, R. (2001). Process-technology fit and its implications for manufacturing performance. *Journal of Operations Management*, 19(5):521–540.

Dayarian, I., Savelsbergh, M., and Clarke, J.-P. (2020). Same-day delivery with drone resupply. *Transportation Science*, 54(1):229–249.

De Freitas, J. C. and Penna, P. H. V. (2020). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, 27(1):267–290.

de Paula Ferreira, W., Armellini, F., de Santa-Eulalia, L. A., and Thomasset-Laperrière, V. (2022). A framework for identifying and analysing industry 4.0 scenarios. *Journal of Manufacturing Systems*, 65:192–207.

Deja, M., Siemiatkowski, M. S., Vosniakos, G.-C., and Maltezos, G. (2020). Opportunities and challenges for exploiting drones in agile manufacturing systems. *Procedia Manufacturing*, 51:527–534. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

Dell'Amico, M., Montemanni, R., and Novellani, S. (2020). Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. *Annals of Operations Research*, 289(2):211–226.

Dertouzos, M. L., Lester, R. K., and Solow, R. M. (1989). Made in america. *The MIT Press*.

Dias, A., Martins, J. J., Antunes, J., Moura, A., and Almeida, J. (2024). Mantis: Uav for indoor logistic operations. In *2024 7th Iberian Robotics Conference (ROBOT)*, pages 1–7. IEEE.

Doriya, R., Mishra, S., and Gupta, S. (2015). A brief survey and analysis of multi-robot communication and coordination. In *International conference on computing, communication & automation*, pages 1014–1021. IEEE.

Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S. (2016). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85.

Dornhege, C. and Kleiner, A. (2006). Visual odometry for tracked vehicles. In *Proc. of the IEEE Int. Workshop on Safety, Security and Rescue Robotics (SSRR)*. IEEE.

Duguay, C. R., Landry, S., and Pasin, F. (1997). From mass production to flexible/agile production. *International Journal of Operations & Production Management*, 17(12):1183–1195.

Duray, R., Ward, P. T., Milligan, G. W., and Berry, W. L. (2000). Approaches to mass customization: configurations and empirical validation. *Journal of Operations Management*, 18(6):605–625. Configuration in Operations management: Taxonomies and Typologies.

Effoduh, J. O. (2016). The fourth industrial revolution by klaus schwab. *The Transnational Human Rights Review 3*.

Elkins, D. A., Huang, N., and Alden, J. M. (2004). Agile manufacturing systems in the automotive industry. *International journal of production economics*, 91(3):201–214.

FAA (2008). Small unmanned aircraft systems (suas) arc complete file. https://www.faa.gov/regulations_policies/rulemaking/committees/documents/ index.cfm/document/information?documentID=300.

FAA (2022). Risk management handbook (faa-h-8083-2a). https://www.faa.gov/regulationspolicies/handbooksmanuals/risk-management-handbook-faa-h-8083-2a.

Famili, A. and Park, J.-M. J. (2020). Rolatin: Robust localization and tracking for indoor navigation of drones. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

Fowler, M. (2012). *Patterns of enterprise application architecture*. Addison-Wesley.

Frohlich, M. T. and Dixon, J. R. (1999). Information systems adaptation and the successful implementation of advanced manufacturing technologies. *Decision Sciences*, 30(4):921–957.

Galasso, F., Mercé, C., and Grabot, B. (2009). Decision support framework for supply chain planning with flexible demand. *International Journal of Production Research*, 47(2):455–478.

Gavilán, A. (2021). Drones in the factory of the future.

Gerwin, D. (2007). Relationships between structure and technology at the organizational and job levels. *Journal of Management Studies*, 16:70 – 79.

Ghobakhloo, M. (2020). "Industry 4.0, digitization, and opportunities for sustainability". *Journal of Cleaner Production*, 252:119869.

Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., and Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies*, 114:657–680.

Goodrich, M. and Schultz, A. (2007). Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1:203–275.

Ha, Q. M., Deville, Y., Pham, Q. D., and Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86:597–621.

Ha, Q. M., Deville, Y., Pham, Q. D., and Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics*, 26(2):219–247.

Han, B., Qu, T., Tong, X., Jiang, J., Zlatanova, S., Wang, H., and Cheng, C. (2022). Grid-optimized uav indoor path planning algorithms in a complex environment. *International Journal of Applied Earth Observation and Geoinformation*, 111:102857.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hasan, R. (2002). *Contribution à l'amélioration des performances des systèmes complexes par la prise en compte des aspects socio-techniques dès la conception : proposition d'un modèle original de situation de travail pour une nouvelle approche de conception*. PhD thesis, Université Henri Poincaré de Nancy.

Hayes, R. H. and Jaikumar, R. (1991). Requirements for successful implementation of new manufacturing technologies. *Journal of Engineering and Technology Management*, 7(3-4):169–175.

Helmick, D. M., Cheng, Y., Clouse, D. S., Matthies, L. H., and Roumeliotis, S. I. (2004). Path following using visual odometry for a mars rover in high-slip environments. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720)*, volume 2, pages 772–789. IEEE.

Hutchinson, G. (1976). Production capacity: Cam vs. transfer line. *INDUSTR. ENGNG; U.S.A.; DA. 1976; VOL. 8; NO 9; PP. 30-35; BIBL. 4 REF.*

Ilin, V., Simic, D., and Saulic, N. (2019). Logistics industry 4.0: challenges and opportunities. In *Ilin, V. Simić, D., Saulić, N.(eds.) LOGIC: 4th Logistics International Conference*.

Jeong, H. Y., Song, B. D., and Lee, S. (2019). Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. *International Journal of Production Economics*, 214:220–233.

Kagermann, H., Wahlster, W., and Hellbig, J. (2013). Recommedations for implementing the strategic initiative INDUSTRIE 4.0. In *final report of the Industrie 4.0 Working Group*.

Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7):2307–2330.

Kester, W. C. (1984). Today's options for tomorrow's growth. *Harvard business review*, 62:153–160.

Khosiawan, Y., Nielsen, I., Do, N. A. D., and Yahya, B. N. (2016). Concept of indoor 3d-route uav scheduling system. In *Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology–ISAT 2015–Part I*, pages 29–40. Springer.

Khosiawan, Y., Park, Y., Moon, I., Nilakantan, J. M., and Nielsen, I. (2019). Task scheduling system for uav operations in indoor environment. *Neural Computing and Applications*, 31(9):5431–5459.

Kim, S. and Moon, I. (2018). Traveling salesman problem with a drone station. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1):42–52.

Kleeman, L. (2003). Advanced sonar and odometry error modeling for simultaneous localisation and map building. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 699–704. IEEE.

Klocke, F., Kratz, S., Auerbach, T., Gierlings, S., Wirtz, G., , and Veselovac, D. (2011). "Process monitoring and control of machining operations". *International Journal of Automation Technology*, 5(3):403–411.

Krystosik-Gromadzińska, A. (2021). The use of drones in the maritime sector - areas and benefits. *Zeszyty Naukowe Akademii Morskiej w Szczecinie*.

Kulatilaka, N. (2002). Valuing the flexibility of flexible manufacturing systems. *IEEE Transactions on engineering management*, 35(4):250–257.

Lakner, P. and Winkler, H. (2018). *Erfolgsfaktoren der Digitalisierung in Logistikprojekten*. PhD thesis, Esslingen University.

Lampel, J. and Mintzberg, H. (1996). Customizing customization. *MIT Sloan Management Review*.

Lara, O. A. P., Nouiri, M., Belkadi, F., and Chriette, A. (2024). Towards a behaviour model for drones integration in agile manufacturing systems. In *2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pages 1–6. IEEE.

Lee, J., Bagheri, B., and Kao, H.-A. (2015). "A cyber-physical systems architecture for industry 4.0-based manufacturing systems". *Manufacturing letters*, 3:18–23.

Leonard-Barton, D. (1988). Implementation as mutual adaptation of technology and organization. *Managing knowledge assets, creativity and innovation*, pages 431–447.

Leveson, N. G. (1995). *Safeware: system safety and computers*. ACM.

Li, M. and Huang, G. Q. (2021). Production-intralogistics synchronization of industry 4.0 flexible assembly lines under graduation intelligent manufacturing system. *International Journal of Production Economics*, 241:108272.

Li, Y., Zhang, G., Pang, Z., and Li, L. (2020). Continuum approximation models for joint delivery systems using trucks and drones. *Enterprise Information Systems*, 14(4):406–435.

Lin, P. T., Liao, C.-A., and Liang, S.-H. (2021). Probabilistic indoor positioning and navigation (pipn) of autonomous ground vehicle (agv) based on wireless measurements. *IEEE Access*, 9:25200–25207.

Liu, Y. (2019). An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research*, 111:1–20.

Loch, C. H. (2017). Creativity and risk taking aren't rational: Behavioral operations in mot. *Production and Operations Management*, 26(4):591–604.

Macrina, G., Pugliese, L. D. P., Guerriero, F., and Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762.

Maghazei, O., Lewis, M. A., and Netland, T. H. (2022). Emerging technologies and the use case: A multi-year study of drone adoption. *Journal of Operations Management*, 68(6-7):560–591.

Maghazei, O. and Netland, T. (2020). Drones in manufacturing: Exploring opportunities for research and practice. *Journal of Manufacturing Technology Management*, 31(6):1237–1259.

Marinelli, M., Caggiani, L., Ottomanelli, M., and Dell'Orco, M. (2018). En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems*, 12(4):253–261.

Marvel, J. A. and Norcross, R. (2017). Implementing speed and separation monitoring in collaborative robot workcells. *Robotics and computer-integrated manufacturing*, 44:144–155.

Mason, S. P. and Merton, R. C. (2019). The role of contingent claims analysis in corporate finance. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation*, pages 123–168. World Scientific.

Mathew, N., Smith, S. L., and Waslander, S. L. (2015). Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, 12(4):1298–1308.

Matthies, L. and Shafer, S. (2003). Error modeling in stereo navigation. *IEEE Journal on Robotics and Automation*, 3(3):239–248.

Mbiadou Saleu, R. G., Deroussi, L., Feillet, D., Grangeon, N., and Quilliot, A. (2018). An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4):459–474.

McCutcheon, D. M., Raturi, A. S., and Meredith, J. R. (1994). The customization-responsiveness squeeze. *MIT Sloan Management Review*, 35(2):89.

McDonald, R. L. and Siegel, D. R. (1985). Investment and the valuation of firms when there is an option to shut down. *International economic review*, pages 331–349.

Meredith, J. R. and Suresh, N. C. (1986). Justification techniques for advanced manufacturing technologies. *International journal of production research*, 24(5):1043–1057.

Miele, A., Lippi, M., and Gasparri, A. (2025). A distributed framework for integrated task allocation and safe coordination in networked multi-robot systems. *IEEE Transactions on Automation Science and Engineering*.

Mintzberg, H. (1988). Generic strategies: toward a comprehensive framework. *Advances in strategic management*, 5(1):1–67.

ModalAI (2025). https://docs.modalai.com/.

Modrák, V. and Šoltysová, Z. (2020). Development of an organizational maturity model in terms of mass customization. *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*, pages 215–250.

Moshref-Javadi, M., Hemmati, A., and Winkenbach, M. (2020a). A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Applied Mathematical Modelling*, 80:290–318.

Moshref-Javadi, M., Lee, S., and Winkenbach, M. (2020b). Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E: Logistics and Transportation Review*, 136:101887.

Mourelo Ferrandez, S., Harbison, T., Webwer, T., Sturges, R., and Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management*, 9(2):374–388.

Mourtzis, D., Angelopoulos, J., and Panopoulos, N. (2022). Unmanned aerial vehicle (uav) manipulation assisted by augmented reality (ar): The case of a drone. *IFAC-PapersOnLine*, 55(10):983–988.

Mourtzis, D., Angelopoulos, J., and Panopoulos, N. (2024). Unmanned aerial vehicle (uav) path planning and control assisted by augmented reality (ar): The case of indoor drones. *International Journal of Production Research*, 62(9):3361–3382.

Murray, C. C. and Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109.

Murray, C. C. and Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, 110:368–398.

Niemann, J., Reich, B., and Stöhr, C. (2021). Lean six sigma. In *Lean Six Sigma: Methoden zur Produktionsoptimierung*, pages 11–61. Springer.

OmniVision (2020). Ov7251vga product brief. https://www.ovt.com/wp-content/uploads/2022/01/OV7251-PB-v1.9-WEB.pdf.

Opray, M. (2016). Revealed: Rio tinto's plan to use drones to monitor workers' private lives. *The Guardian*, 8.

Pan, Y., Qu, T., Wu, N., Khalgui, M., and Huang, G. (2021). Digital twin based real-time production logistics synchronization system in a multi-level computing architecture. *Journal of Manufacturing Systems*, 58:246–260.

Pasha, J., Elmi, Z., Purkayastha, S., Fathollahi-Fard, A. M., Ge, Y.-E., Lau, Y.-Y., and Dulebenets, M. A. (2022). The drone scheduling problem: A systematic state-of-the-art review. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14224–14247.

Perez-Grau, F. J., Martinez-de Dios, J. R., Paneque, J. L., Acevedo, J. J., Torres-González, A., Viguria, A., Astorga, J. R., and Ollero, A. (2021). Introducing autonomous aerial robots in industrial manufacturing. *Journal of Manufacturing Systems*, 60:312–324.

Pine, B. J. (1999). *Mass customization: the new frontier in business competition*. Harvard business press.

Poikonen, S. and Campbell, J. F. (2021). Future directions in drone routing research. *Networks*, 77(1):116–126.

Poikonen, S. and Golden, B. (2020a). The mothership and drone routing problem. *INFORMS Journal on Computing*, 32(2):249–262.

Poikonen, S. and Golden, B. (2020b). Multi-visit drone routing problem. *Computers & Operations Research*, 113:104802.

Ponza, A. (2016). Optimization of drone-assisted parcel delivery.

Porter, M. E. (1985). Creating and sustaining superior performance. *Competitive advantage*, 167:167–206.

Proia, S., Cavone, G., Camposeo, A., Ceglie, F., Carli, R., and Dotoli, M. (2022a). Safe and ergonomic human-drone interaction in warehouses. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6681–6686. IEEE.

Proia, S., Cavone, G., Carli, R., and Dotoli, M. (2022b). A multi-objective optimization approach for trajectory planning in a safe and ergonomic human-robot collaboration. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 2068–2073. IEEE.

Radivojević, G. and Milosavljević, L. (2019). The concept of logistics 4.0. In *4th Logistics International Conference*, pages 23–25.

Rexroth, B. (2012). Ergonomics guidebook for manual production systems.

Rungtusanatham, M. J. and Salvador, F. (2008). From mass production to mass customization: Hindrance factors, structural inertia, and transition hazard. *Production and Operations Management*, 17(3):385–396.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., and Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston consulting group*, 9(1):54–89.

Salama, M. and Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, 114:620–642.

Sarder, M. B., Rogers, J., and Liles, D. (2006). *Role of Logistics in Mass Customization*. PhD thesis, University of Texas at Arlington.

Schermer, D., Moeini, M., and Wendt, O. (2019). The traveling salesman drone station location problem. In *World congress on global optimization*, pages 1129–1138. Springer.

Singh, D. (2019). How to avoid slamming into things with slam. https://www.qualcomm.com/news/onq/2019/08/how-avoid-slamming-things-slam.

Small, M. H. and Yasin, M. M. (1997). Advanced manufacturing technology: implementation policy and performance. *Journal of Operations Management*, 15(4):349–370.

StartUs-Insights (2023). Drones in the factory of the future.

Stern, R., Sturtevant, N., Felner, A., Koenig, S., Ma, H., Walker, T., Li, J., Atzmon, D., Cohen, L., Kumar, T., et al. (2019). Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, pages 151–158.

Stolzer, A. J., Sumwalt, R. L., and Goglia, J. J. (2023). *Safety management systems in aviation*. CRC Press.

Suresh, N. C. and Meredith, J. R. (1985). Justifying multimachine systems: an integrated strategic approach. *Journal of Manufacturing Systems*, 4(2):117–134.

Swamidass, P. M. (1987). Planning for manufacturing technology. *Long Range Planning*, 20(5):125–133.

Swamidass, P. M. and Waller, M. A. (1990). A classification of approaches to planning and justifying new manufacturing technologies. *Journal of Manufacturing Systems*, 9(3):181–193.

Talukder, A., Goldberg, S., Matthies, L., and Ansar, A. (2003). Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 2, pages 1308–1313. IEEE.

Tang, G., Tang, C., Claramunt, C., Hu, X., and Zhou, P. (2021). Geometric a-star algorithm: An improved a-star algorithm for agv path planning in a port environment. *IEEE access*, 9:59196–59210.

Tezza, D. and Andujar, M. (2019). The state-of-the-art of human–drone interaction: A survey. *ieee access*, 7:167438–167454.

Thoben, K.-D., Wiesner, S., and Wuest, T. (2017). "Industrie 4.0 and smart manufacturing - a review of research issues and application examples". *International journal of automation technology*, 11(1):4–16.

Trist, E. L. (1981). *The evolution of socio-technical systems*, volume 2. Ontario Quality of Working Life Centre Toronto.

Troudi, A., Addouche, S.-A., Dellagi, S., and Mhamedi, A. E. (2018). Sizing of the drone delivery fleet considering energy autonomy. *Sustainability*, 10(9):3344.

Tu, P. A., Dat, N. T., and Dung, P. Q. (2018). Traveling salesman problem with multiple drones. In *Proceedings of the 9th international symposium on information and communication technology*, pages 46–53.

Uckelmann, D. (2008). "A definition approach to smart logistics". In *International Conference on Next Generation Wired/Wireless Networking*, pages 273–284. Springer.

Ustundag, A. and Cevikcan, E. (2017). *Industry 4.0: managing the digital transformation*. Springer.

Van de Ven, A. H. (1986). Central problems in the management of innovation. *Management science*, 32(5):590–607.

Veil, A. (2019). Zf is first in germanny to fly drones over plant premises.

Veroustraete, F. (2015). The rise of the drones in agriculture. *EC agriculture*, 2(2):325–327.

Wang, X., Poikonen, S., and Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. *Optimization Letters*, 11(4):679–697.

Weibel, R. E. and Hansman, R. J. (2006). Safety considerations for operation of unmanned aerial vehicles in the national airspace system. Technical report, MIT International Center for Air Transportation.

Winkelhaus, S. and Grosse, E. H. (2020). Logistics 4.0: a systematic review towards a new logistics system. *International Journal of Production Research*, 58(1):18–43.

Winkler, H. and Zinsmeister, L. (2019). Trends in digitalization of intralogistics and the critical success factors of its implementation. *Brazilian Journal of Operations & Production Management*, 16(3):537–549.

Wu, X., Lei, Y., Tong, X., Zhang, Y., Li, H., Qiu, C., Guo, C., Sun, Y., and Lai, G. (2022). A non-rigid hierarchical discrete grid structure and its application to uavs conflict detection and path planning. *IEEE Transactions on Aerospace and Electronic Systems*, 58(6):5393–5411.

Yadav, Y. and Narasimhamurthy, A. (2017). Algorithms for solving the vehicle routing problem with drones. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*. IEEE Bangalore, India.

Yousif, K., Bab-Hadiashar, A., and Hoseinnezhad, R. (2015). An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4):289–311.

Yurek, E. E. and Ozmutlu, H. C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 91:249–262.

Zaychenko, I., Smirnova, A., and Borremans, A. (2018). Digital transformation: the case of the application of drones in construction. In *MATEC web of conferences*, volume 193, page 05066. EDP Sciences.

Zhang, T., Liu, X., Kühnlenz, K., and Buss, M. (2009). Visual odometry for the autonomous city explorer. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3513–3518. IEEE.

Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. (2017). "Intelligent manufacturing in the context of industry 4.0: A review". *Engineering*, 3(5):616–630.

Titre : Intégration des drones pour supporter l'agilité des systèmes de production dans le contexte d'industrie 4.0

Mots clés : Industry 4.0, Intralogistics, Drones, Agilité, Analyse de faisabilité

Résumé : Cette thèse explore l'intégration des drones dans les systèmes de production agiles, en mettant l'accent sur leur rôle comme outils de soutien pour l'intralogistique. Alors que les robots au sol sont largement utilisés comme systèmes flexibles de manutention, leur efficacité reste limitée dans des environnements dynamiques nécessitant des livraisons unitaires. Les drones, initialement conçus pour la surveillance, présentent un potentiel pour assurer le transport rapide d'objets légers et accroître la réactivité dans des contextes de production et d'entreposage hautement personnalisés. Leur adoption implique toutefois de relever plusieurs défis d'ordre opérationnel, organisationnel, économique et législatif. Pour accompagner la prise de décision, ce travail propose un cadre d'adoption des Technologies Avancées de Production permettant aux responsables d'évaluer faisabilité et rentabilité selon trois axes : économique, organisationnel et opérationnel. À travers des outils de simulation, l'étude évalue la performance des drones dans des conditions réalistes et développe une architecture système intégrant communication, planification, routage et gestion des conflits. Ainsi, cette recherche positionne les drones comme compléments intralogistiques capables de renforcer agilité, efficacité et compétitivité dans les environnements de l'Industrie 4.0.

Title : Integration of drones to support the agility of production systems in the context of industry 4.0

Keywords : Industry 4.0, Intralogistics, Drones, Agility, Feasibility analysis

Abstract : This thesis investigates the integration of drone technology into agile manufacturing systems, with a focus on their role as support tools for intralogistics. While ground robots have been widely adopted as flexible material handling systems, their efficiency is limited in dynamic environments requiring one-piece flow deliveries. Drones, originally developed for surveillance, offer potential as fast-paced transport agents for lightweight items, enabling responsiveness in highly customized production and warehousing settings. However, their adoption requires addressing operational, organizational, economic, and legislative challenges. To support decision-making, this work proposes an Advanced Manufacturing Technology adoption framework that helps managers assess feasibility and profitability across three axes: economic, organizational, and operational. Using simulation tools, the study evaluates drone-based delivery systems under realistic production conditions, incorporating uncertainties and performance metrics. Furthermore, a system architecture is developed to address communication, scheduling, routing, and conflict management. Overall, this research bridges technical feasibility with strategic considerations, positioning drones as complementary intralogistics enablers that enhance agility, efficiency, and competitiveness in Industry 4.0 environments.