POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

An Integrated Model for the Performance Evaluation of Manufacturing Systems with Operational Learning of AI Quality Control Algorithms during Ramp-Up

Tesi di Laurea Magistrale in
Management Engineering - Ingegneria Gestionale

Author: **Alessandro Bordone**

Student ID: 224762
Advisor: Prof. Maria Chiara Magnanini
Academic Year: 2024-25

Abstract

The growing integration of artificial intelligence (AI) and adaptive decision systems within industrial operations raises not only questions of model accuracy, but also of governance: in particular, when adaptive learning should be activated in production environments characterized by uncertainty and risk sensitivity. While predictive models can operate continuously, the decision to enable learning-driven adaptation represents a structural transition that may affect both operational stability and cumulative risk exposure.

This thesis investigates the impact of activation timing in a simulated industrial inspection system. A discrete-event simulation framework is developed to model a two-regime architecture: an initial conservative deterministic phase followed by an adaptive regime in which error probabilities evolve parametrically with accumulated production volume. The transition between regimes is governed by a statistically defined activation threshold.

The study compares alternative activation policies under identical learning dynamics and evaluates their effects on transient error behavior, cumulative risk exposure and throughput performance. Results show that activation timing does not modify steady-state capacity or structural learning behavior. However, transient differences in error dynamics generate persistent cumulative effects when integrated over production volume. Under the baseline risk structure, delayed activation reduces cumulative exposure without penalizing long-run throughput.

A sensitivity analysis demonstrates that the optimal activation threshold depends on the relative weighting of false positive and false negative costs. Activation timing should therefore be interpreted primarily as a risk-governance decision rather than a productivity optimization problem. The methodological framework proposed in this work provides a transferable approach for managing the introduction of adaptive decision systems in industrial contexts.

Keywords: Production Ramp-up, Manufacturing Systems, Learning Curves, Production Quality, Risk-based Inspection, Discrete-Event Simulation

Abstract in lingua italiana

La crescente integrazione di modelli di intelligenza artificiale (AI) e di architetture decisionali abilitate all'apprendimento nei contesti industriali solleva non solo questioni legate all'accuratezza predittiva, ma anche problematiche di governance: in particolare, quando attivare il perfezionamento basato sull'apprendimento in ambienti produttivi caratterizzati da incertezza e sensibilità al rischio. Sebbene i moduli predittivi possano operare in modo continuo, la decisione di attivare un comportamento guidato dall'apprendimento rappresenta una transizione strutturale tra un regime deterministico conservativo e un regime adattivo le cui prestazioni evolvono con l'esperienza accumulata. Tale transizione può influenzare sia la stabilità operativa sia l'esposizione cumulata al rischio.

Questa tesi analizza l'impatto del timing di attivazione in un sistema di ispezione industriale simulato. Viene sviluppato un framework di simulazione ad eventi discreti per modellare un'architettura a due regimi: una fase iniziale deterministica e conservativa seguita da un regime adattivo in cui le probabilità di errore evolvono in modo parametrico al crescere del volume produttivo cumulato. Il passaggio tra i due regimi è governato da una soglia di attivazione definita su base statistica.

Lo studio confronta politiche di attivazione alternative, a parità di dinamiche di apprendimento, valutandone gli effetti sul comportamento transitorio degli errori, sull'esposizione cumulata al rischio e sulle prestazioni di throughput. I risultati mostrano che il timing di attivazione non modifica la capacità a regime né la traiettoria di apprendimento sottostante. Tuttavia, differenze transitorie nelle dinamiche di errore generano effetti cumulativi persistenti quando integrate rispetto al volume produttivo. Nella configurazione di rischio di riferimento, un'attivazione ritardata riduce l'esposizione cumulata senza penalizzare le prestazioni di lungo periodo.

Un'analisi di sensitività dimostra inoltre che la soglia ottimale di attivazione dipende dalla ponderazione relativa dei costi associati ai falsi positivi e ai falsi negativi. Il timing di attivazione deve pertanto essere interpretato principalmente come una decisione di governance del rischio piuttosto che come un problema di ottimizzazione della produttività. Il framework metodologico proposto fornisce un approccio trasferibile per la gestione

dell'introduzione di architetture decisionali guidate dall'apprendimento in contesti industriali.

Parole chiave: Fase di avviamento produttivo, Sistema di produzione, Curve di apprendimento, Qualità della produzione, Ispezione basata sul rischio, Simulazione a eventi discreti

Contents

Introduction

The ongoing digital transformation of manufacturing systems, often framed within the paradigms of Industry 4.0 and smart production environments, has accelerated the integration of predictive analytics and adaptive decision systems into operational processes. Artificial intelligence is increasingly deployed in quality inspection, process control and supply chain management to enhance responsiveness and efficiency. However, beyond predictive accuracy, insufficient attention has been devoted to the governance of activation timing for learning-driven decision logic in risk-sensitive production environments.

In industrial settings, predictive systems are frequently introduced alongside existing deterministic control logic. While prediction modules may operate continuously, the activation of learning-driven adaptation, where model outputs begin influencing decisions, represents a governance decision with direct implications for risk exposure and operational stability. Activating adaptive logic too early may introduce transient instability, whereas delaying activation may postpone efficiency improvements. The timing of activation thus becomes a structured decision variable rather than a purely technical parameter.

This thesis addresses the problem at two complementary levels. The primary objective is to develop a quantitative representation of reliability evolution in AI-based inspection systems during ramp-up, as a function of cumulative production experience. A discrete-event simulation framework is developed to model a two-regime architecture governed by a statistically defined activation threshold. This modeling perspective formalizes how error probabilities evolve and how regime transition can be embedded within a structured analytical environment.

Building on this framework, the thesis then addresses a governance-oriented research question: *"Does earlier or later activation of the learning-enabled regime improve system performance when cumulative risk exposure is explicitly considered?"*.

To answer this question, the study integrates transient error analysis, cumulative risk evaluation, throughput assessment and sensitivity analysis with respect to risk weighting. The objective is not to improve algorithmic design, but to formalize activation governance within industrial contexts characterized by uncertainty and asymmetric risk tradeoffs.

The thesis is organized as follows:

- *Chapter 1* presents the industrial and theoretical background of production ramp-up, learning-enabled inspection systems and risk-based quality control.
- *Chapter 2* introduces the analysis of the industrial problem that forms the basis of the research, with particular reference to the ramp-up phase in manufacturing systems.
- *Chapter 3* outlines the methodological framework adopted for predictive modeling of the learning behavior of an AI-based automatic inspection system during the production ramp-up.
- *Chapter 4* defines the industrial case and describes the manufacturing system under study.
- *Chapter 5* presents and discusses the results, including baseline analysis, activation timing comparison, cumulative risk evaluation, sensitivity analysis and industrial implications.
- *Chapter 6* summarizes the main findings, discusses limitations and outlines future research directions.

Finally, a set of appendices provides supplementary technical material supporting the main analysis. These include the detailed logical architecture of the implemented simulation model, the MATLAB automation and post-processing procedures and the extended scenario exploration matrix.

1 State of the Art

1.1. Ramp-up in Industrial Production

Globalization has profoundly changed the production models of manufacturing companies. To remain competitive, companies need to be able to respond quickly to fluctuations in demand. They also need to be cost-effective and able to adapt to an increasingly diverse customer base in highly globalized markets. The competitive environment is no longer driven primarily by supply, as was the case in the last century, but by customer needs, which have a decisive influence on corporate strategies. Added to this scenario is technological evolution, which pushes companies to continuously improve their products and processes in order to defend and increase their market share. As a result, the industrialization of new products has now become a structural element of business activity, bringing with it significant challenges in managing product release and production ramp-up ([1]).

In this context, an effective integration between the product life cycle, the process and the production system becomes crucial ([2]). This integration gives the ramp-up phase a central role, as it allows for better management of the transition from the start of production to the achievement of stable and high-quality volumes. Industrial literature has devoted increasing attention to this concept, developing models and methodological approaches aimed at managing and reducing the time duration from the initial production of the first part to the consistent output of quality parts at the desired effective throughput level. This time span is also referred to as *ramp-up time* or *time to volume* and it occurs subsequent to a greenfield implementation or significant reconfiguration (brownfield) within the manufacturing system in question ([3]). A zero ramp-up time is preferable, as the desired effective production rate would be achieved without any production loss. Nonetheless, in practical systems, this optimal condition is not attained due to several factors contributing to output losses. Therefore, the increase in productivity can be analysed by graphing the throughput against time, as illustrated in Figure 1.1.

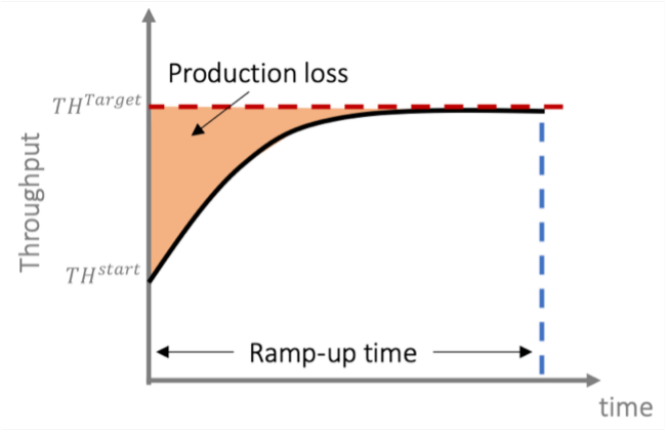

Figure 1.1: Representation of productivity ramp-up in manufacturing systems, [2].

In particular, Figure 1.1 shows the production loss (P^{Loss}), which represents the cumulative loss of output compared to the target throughput during the ramp-up phase, and it can be expressed as:

$$P^{Loss} = \int_{t=0}^{t=t_{ramp}} \left(TH^{Target} - TH^{Eff}(t)\right) dt \tag{1.1}$$

where t_{ramp} is the length of the ramp-up phase, TH^{Target} is the target throughput and TH^{Eff} is the actual throughput observed over time during the ramp-up phase.

Moreover, several authors have highlighted that modern production systems should be considered as complex cyber-physical systems, characterized by significant dynamism and the necessity to swiftly adjust to demand variations and ongoing technological advancements ([4, 5]). [4] emphasize that manufacturing companies today operate in a very dynamic environment, where constant improvement is not an option but a must if they want to stay competitive. Digital innovations and Industrial Internet of Things (IIoT) make an extensive amount of system data available, which helps advanced decision-support tools work better. With the evolution of production systems and the advent of Industry 4.0, research has progressively shifted towards the development of digital and simulation tools, as well as advanced monitoring systems, aimed at supporting decision-making and improving the overall efficiency of manufacturing processes. At the same time, research by [6] and [7] shows that these new technologies have additionally generated new integration problems, which make it even harder to manage ramp-up.

Finally, [5] point out that the time interval between two significant changes in manufacturing systems is gradually decreasing, due to the growing need for reconfigurations, the adoption of new technologies and the introduction of innovative digital solutions for opti-

mizing operations. In this perspective, the ramp-up phase should not be regarded merely as the initial start of production for a new product, but rather as both a system adaptation process and a testing ground for assessing the resilience of manufacturing systems, which are required to balance often contrasting objectives: speed and quality, flexibility and stability, innovation and sustainability ([1, 8]). In light of the above, two crucial aspects to consider during the ramp-up phase today are managing the workforce and learning, and making sure operations are strong and long-lasting. In the first case, it is essential to develop systems that facilitate organizational learning and knowledge management processes, providing decision support tools for operators as well ([9, 10]). Instead, in the second case, it is important to make sure that the target capacity is reached quickly and that costs and resources are used wisely to avoid waste and inefficiency ([4, 11]).

The concept of production ramp-up stems from the new product development (NPD) literature ([1]). Indeed, according to the authors [12], the NPD process can be break down into six steps (Figure 1.2), defining the last one as the *Production ramp-up*.

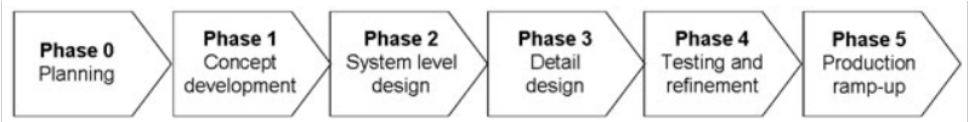

Figure 1.2: Break down of the NPD process, [12].

Unfortunately, the NPD literature does not study in detail the ramp-up issue. Nevertheless, in the last decade, a ramp-up literature emerges independently.

Generally speaking, the ramp-up phase occurs when a new product is introduced in a factory but also when a new process or a new plant starts up ([13, 14]). To facilitate a more profound comprehension of the ramp-up phase in production systems, it is first necessary to provide a comprehensive overview of the existing literature concerning the definition of ramp-up. Its concept has evolved over time, moving from a narrow view linked solely to product development to a broader perspective that considers the entire production system and its performance objectives.

[13] defined the production ramp-up as the period between the completion of development and the full capacity utilization of the system, thus giving the first formal definition of this concept. In this early view, ramp-up was considered the final stage of the product development process. At the same time, [14] introduced some time indicators related to ramp-up, distinguishing between *time-to-market*, which ends with the start of commercial production and *time-to-volume*, which explicitly includes the production ramp-up period. Added to these is *time-to-payback*, which represents the time interval necessary to recover

the initial financial costs. Then, the ramp-up concept was then extended to the achievement of full production or the satisfaction of initial targets in terms of volume, yield and costs ([15, 16]).

Subsequently, [17] proposed dividing the ramp-up into sub-phases, distinguishing between *preparation* phase (with pre-series and pre-production run) and the *run-up* phase, thus giving a more detailed definition (Figure 1.3). Later studies, such as [18], referring back to the ramp-up definition of [13], showed that product development rarely ends at a specific moment and that, as a result, the ramp-up phase should be considered to start more correctly with the Start of Production (SOP). Finally, [1] arrived at defining the ramp-up phase as the period of time, which concerns not only production, but also product development and the supply chain, introducing a systemic and comprehensive view of the phenomenon.

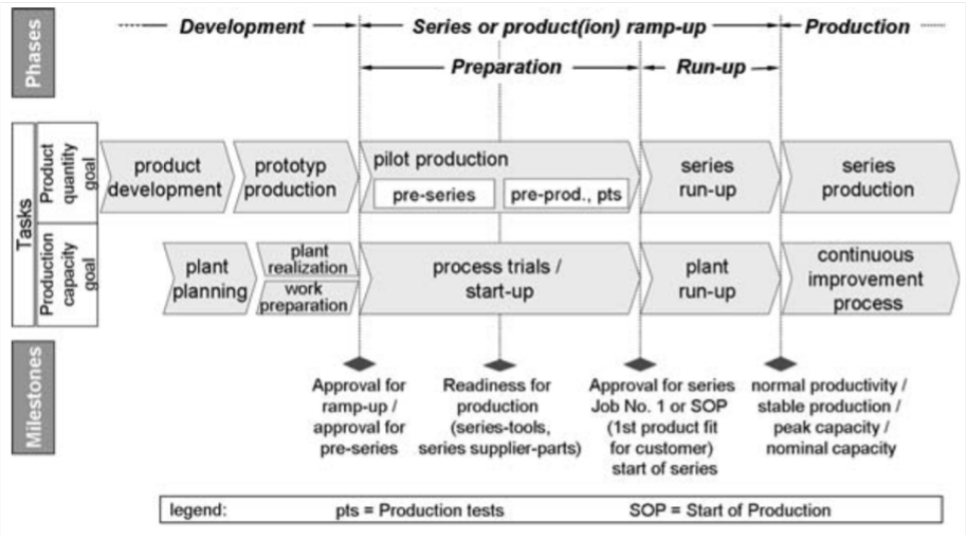

Figure 1.3: Phases and tasks in production ramp-up, [17].

More recently, within the context of Industry 4.0, [11] defined ramp-up as the phase that begins after product development, during which production is progressively increased until a predefined maximum throughput is achieved. This definition illustrates the significance of digitization and sophisticated decision-support technologies in the necessity of optimizing and adapting systems within current cyber-physical production environments.

According to the above definitions, the meaning of ramp-up has changed from a purely technical term related to production volume to a broader one that sees it as a time for the production system to adapt and learn, marked by uncertainty, initial inefficiencies and gradual improvement in performance until stabilization is reached.

To fully understand the challenges that emerge during the ramp-up period, it is essential to understand its nature. The most common characteristics of the ramp-up phase, which distinguish it from full-scale production, are explained below.

One of the main features is the low initial level of knowledge about the product and the process, which results in a gradual and often difficult learning process ([13, 15, 19, 20]). Firstly, a lack of knowledge frequently leads to low production output during ramp-up ([14, 15]), and companies often experience higher cycle times than planned in parallel ([13–15]). These characteristics bring to low production capacities, which prevent firms from meeting the targeted volumes in the early phases of industrialization ([13–15]). Another recurring property is the high demand for new products, which places significant pressure on production systems that are still unstable and not fully optimized ([13–15]). At the same time, disturbances in processes, supply chains, or product quality are frequent, further complicating the achievement of stable performance ([14, 18]).

To provide a more systematic view of the topic, the common category of problems encountered during ramp-up are listed below, according to [1]:

- Product – incomplete specifications, late engineering changes, lack of technical maturity.
- Technical processes – long set-ups, bottlenecks, low robustness.
- Logistics – supplier delays, misaligned supply chain.
- Quality – high scrap rates and rework.
- Methods and tools – planning instruments unsuitable for unstable conditions.
- Personnel – insufficient training, unclear responsibilities.
- Cooperation and communication – poor communication between R&D, production and logistics.

The literature confirms that the ramp-up phase is a critical and unstable stage, typically marked by inefficiencies, delays and organizational complexities. Recent studies, however, increasingly interpret ramp-up as a systemic adaptation process, in which resilience, flexibility and sustainability play a central role.

Even though great progress has been made, ramp-up is still a period in which resources and processes do not achieve their intended performance levels immediately. The key issues discovered include integrating the supply chain, developing tools for making operational decisions and comprehending the human dimension. Recent research has empha-

sized digital and data-driven techniques for multi-objective decision-making, which have been verified in complicated industrial situations. The goal is to save time, losses and inefficiencies, making ramp-up easier to manage and more competitive ([1, 2]).

1.2. Quality Ramp-up

In the current manufacturing environment, production systems are experiencing ramp-up phases with increasing frequency throughout their life cycle. This trend is driven by the growing variety of products, higher levels of customization and accelerating innovation cycles. However, traditional quality management methodologies are primarily designed for high-volume, steady-state production and tend to lose effectiveness during the early stages of ramp-up, when process instability and the presence of disturbances have a significant impact on overall performance ([3]). Indeed, the concept of quality in the ramp-up phase is a fundamental requirement. During this stage, companies need to keep up with both the rise in production levels and the gradual increase in product quality until they reach the standards that were originally established ([1, 19]).

[3] point out that traditional quality improvement methods (such as Six Sigma or Just-in-Time) are not particularly effective during the ramp-up phase, since they depend on large volumes of data collected in stable conditions. To overcome this limitation, the authors suggest a framework for quality management in ramp-up, built on two complementary strategies. The first one is to anticipate potential problems already during the design phase, by modeling possible failures, component variability and interactions between resources, also through multi-level and multi-physical digital simulations capable of representing complex scenarios. This makes it possible to reduce the gap between expected and actual performance once production starts ([21–23]). The second one is to continuously improve during the ramp-up phase, making use of real production data to spot bottlenecks, track how defects spread and uncover their root causes. In this regard, methods such as Stream of Variation Analysis (SOVA) have shown to be particularly effective and have been widely applied and validated in assembly lines and multi-stage production systems ([21, 24]). This second strategy is embedded in an iterative cycle of continuous improvement, in which the real system is constantly monitored, data are collected and analysed, and the resulting information is used to update digital models of resources and of the production system. System performance can then be evaluated under alternative configurations. Such analyses make it possible to identify the most critical bottlenecks and to conduct sensitivity studies that support targeted intervention decisions. In this perspective, the approach can be further strengthened by integrating

advanced bottleneck detection methodologies and the adoption of digital technologies such as digital twins and virtual commissioning, which allow to reduce ramp-up time and costs by testing and validating alternative solutions in a virtual environment ([25–27]). As illustrated in Figure 1.4, this generates a closed feedback loop between the real system and its models, which guides the implementation of improvement actions throughout the ramp-up phase ([3]).

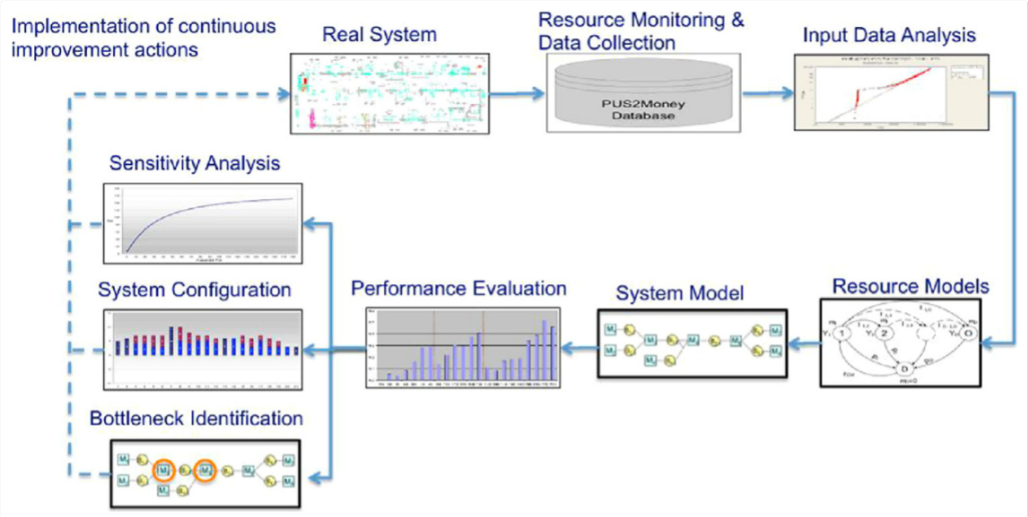

Figure 1.4: Continuous improvements loop during the ramp-up phase, [3].

This introduces the concept of effective throughput (TH_{Eff}), which is the most significant means to measure overall performance during ramp-up as it simultaneously reflects the combined effects of quality, logistics and maintenance ([3, 28]). It is defined as the product of the total system throughput and its quality yield. The corresponding effective throughput losses (TH_{Loss} - previously defined in Equation (1.1) as P^{Loss}) represent the hidden cost of ramp-up, which can be particularly significant, especially in the automotive sector, where the ramp-up phase of a new model may last 20–30 weeks, generating significant economic impacts and temporarily reducing the overall competitiveness of the production system ([3, 29, 30]).

Furthermore, [3] provide a detailed analysis of the causes of throughput losses during ramp-up, distinguishing between internal factors (such as unexpected machine behavior, integration defects between resources, component variability, inadequate system and control design, human errors and slow learning curves) and external factors, including input material variability, plant service instability and cultural or organizational issues. To address these challenges and in addition to the two strategies mentioned above, the literature additionally discusses about a more recent contribution by [31], who suggest a model-based

quality assurance framework for managing ramp-up in multi-stage systems. The authors argue that classic tools such as Failure Mode and Effects Analysis (FMEA), Statistical Process Control (SPC) or Control Plans are of limited effectiveness in this phase, since they rely on steady-state conditions and large datasets. Their approach, instead, leverages Bayesian Networks to combine expert knowledge with progressively collected data, becoming possible to represent cause-effect relationships between processes and quality, to predict the propagation of defects and to identify the critical paths to stabilization.

Finally, several authors in the literature - such as [3], [8] and [32] - highlight the central role of Key Enabling Technologies (KETs) and Industry 4.0 digital architectures in making ramp-up both faster and more sustainable. The integration of advanced sensor technologies, big data analytics and cyber-physical systems is regarded as the most effective approach to reducing throughput variability and facilitating knowledge transfer across successive ramp-up phases. As shown in Figure 1.5, a KETs-based approach reduces cumulative throughput losses during ramp-up, bringing the system closer to target performance levels more quickly.

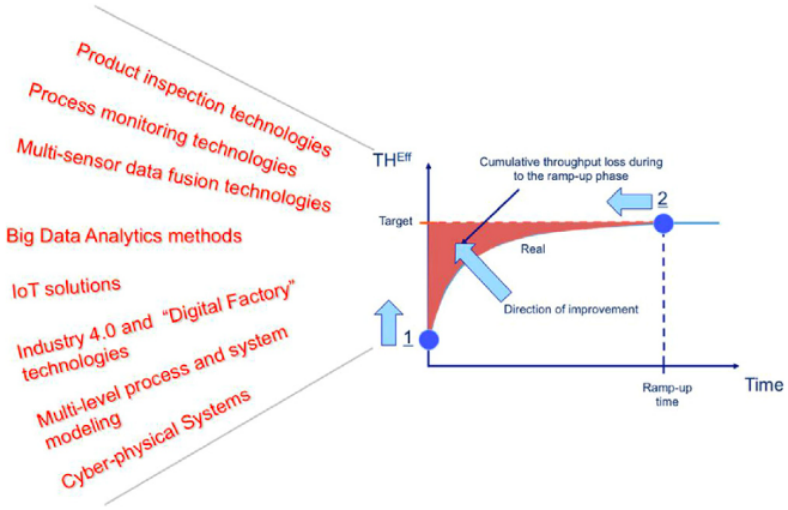

Figure 1.5: Cross-KETs approach for production quality ramp-up reduction, [3].

1.3. Performance Evaluation of Manufacturing Systems

The evaluation of manufacturing systems is a key concern for both researchers and practitioners, as it helps to monitor efficiency, guide managerial choices and drive continuous improvement. This task becomes even more important during the ramp-up phase, when indicators such as throughput, Overall Equipment Effectiveness (OEE), defect rates, cy-

cle times and unit costs are typically unstable and subject to strong variability, reflecting the uncertainties of early production stages, as discussed in the previous sections. As a result, traditional performance measurement systems may offer only a partial or unreliable picture, which in turn limits a company's ability to make timely and well-informed decisions.

As noted by [33], performance measurement has gone through a clear evolution over time. In the 1960s, the focus was mainly on costs and productivity. By the 1980s, with the spread of approaches such as Total Quality Management (TQM) and ISO9000 certifications, the spotlight shifted towards quality. From the 1990s onward, the idea of measuring performance became more multidimensional, combining both financial and non-financial aspects. Thanks to many tools and frameworks (e.g., the Performance Measurement Matrix, the SMART Pyramid, the Balanced Scorecard and the Performance Prism), performance assessment moved beyond a purely economic–productive view, embracing a wider perspective that also considers customers, internal processes and organizational learning. This evolution is further confirmed by several literature reviews. [34] stress that performance measurement systems must be designed in line with corporate strategy and remain flexible enough to adapt to changes in the competitive environment. Indeed, the authors emphasize that an effective evaluation system should go beyond the mere collection of indicators. The measurements feed into benchmarking, gap identification, the implementation of corrective actions and continuous monitoring, thus creating a closed performance management cycle. As illustrated in Figure 1.6, this approach highlights the iterative nature of performance management: results - whether related to costs, quality, timing, customer satisfaction, or market share - are constantly compared against objectives and best practices, generating a continuous process of learning and improvement. In the ramp-up context, this perspective is particularly valuable, as adopting a closed-loop logic transforms performance evaluation from a simple descriptive tool into a strategic lever for decision-making and for accelerating system stabilization.

Along the same lines, [35] describe the gradual shift from traditional evaluation tools to comprehensive performance measurement systems, which are characterized by an integrated perspective, designed not only to monitor performance but also to support decision-making and promote continuous improvement.

[33] also introduces a hierarchical 5-level approach. The 5-levels cover *machine*, *cell*, *line*, *factory* and *network*. Within each level, five major metrics are used for performance measurement, i.e., time, cost, quality, flexibility and productivity. The matrix framework thus covers all the essential elements for manufacturing from a systems point of view. Naturally, there are specific measures designed for different levels of manufacturing systems

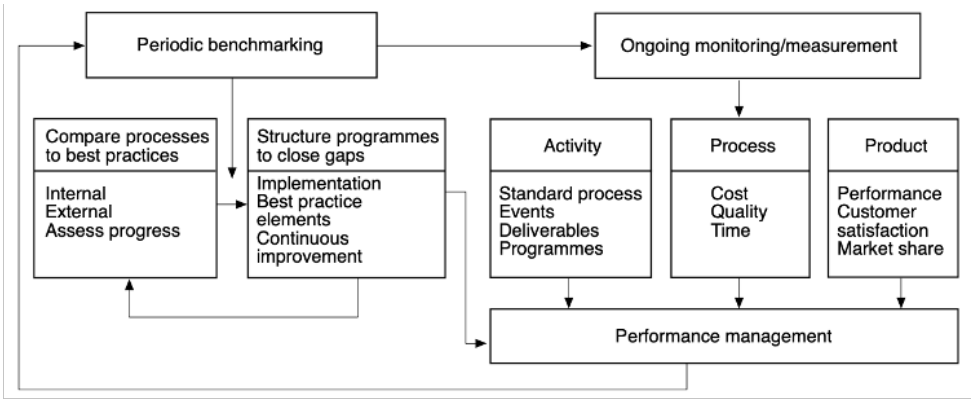

Figure 1.6: Closed loop performance management, [34].

([36–38]). This framework aims to avoid focusing on local optimizations that undermine the overall performance of the system. It instead encourages a more comprehensive, systemic perspective that is in line with the company's strategic objectives ([36, 39]). In his analysis of the aerospace sector, [33] observed that companies usually give priority to lead time, delivery reliability, operating costs and defect rates. By contrast, flexibility is often neglected - even though it is becoming increasingly important in today's industrial contexts, where demand is highly variable and customization is the norm.

This perspective is further expanded by the introduction of additional evaluation dimensions: environmental sustainability and eco-efficiency, highlighted by [40] and by various contributions on eco-manufacturing; corporate social responsibility and the creation of shared value, which have progressively redefined the role of performance from a strategic perspective ([41]); the agility and resilience of supply chains ([42]); and finally, e-manufacturing and the adoption of integrated digital architectures ([43]). In this sense, the transition from static measurement systems to dynamic, data-driven systems based on advanced sensor technology, big data analytics and digital twins, has been accelerated by the advent of Industry 4.0 ([32, 44]).

The way production system performance is evaluated has shifted from a focus on costs and productivity to more multidimensional, integrated and dynamic approaches. However, in ramp-up phases, traditional indicators often prove unstable and incomplete, making it necessary to adopt specific metrics and predictive tools that can support fast and complex decision-making.

1.3.1. Performance Measurement during Ramp-up

The ramp-up phase has unique characteristics that make traditional performance measurement systems, which are designed mainly for stable and high-volume production, largely inadequate. In industrial ramp-up stages, [45] noticed that delays, inefficiencies and costly iteration cycles - which undermine the reliability of conventional performance measurement methods - are mainly caused by an incomplete knowledge of the product and process, unstable operating conditions and decisions are often driven by trial-and-error approach.

As is evident in the extant literature, a number of authors, including [18], highlighted that tools such as Overall Equipment Effectiveness (OEE) or other traditional indicators fail to capture the dynamic nature of ramp-up, as they mostly provide ex-post and static measures. In this phase, performance targets cannot be treated as fixed but rather evolve progressively as the production system learns. For this reason, there is a need for a framework capable not only of measuring final outcomes, but also of assessing the trajectory through which the system converges towards capacity and quality targets.

A critical issue highlighted in numerous studies is knowledge management. [13] and [19] show that one of the main reasons for inefficiency is that there are no structured feedback loops or mechanisms to capture knowledge during ramp-up. Similarly, [46] and [47] underline that clearly defined intermediate objectives and continuous monitoring of progress are essential to effective decision-making. Indeed, decisions risk being based on fragmentary information, leading to delays, inefficiencies, and additional costs if there is not a proper system of metrics.

To tackle these issues, [45] propose a structured framework for assessing ramp-up performances based on three fundamental dimensions, illustrating the evolution of each dimension over time until a steady state is achieved, as illustrated in Figure 1.7. They are articulated through distinct quantitative metrics, facilitating an impartial assessment of the system's progression over time. In this formulation, the indices i and j are used consistently: j represents the individual elementary metrics monitored (e.g., disruption, quality or efficiency parameters), while i indicates the different temporal or state instances in which these measures are detected. Instead, the parameters m and n define the number of instances and the total number of metrics considered, respectively. Lastly, s_i acts as a time weighting factor, providing more weight to more recent cycles and less weight to earlier ones. This makes the ramp-up index a dynamic measure that responds to improvements over time.

This common structure allows the calculation of the three performance dimensions to be

unified within the same mathematical framework. The following are the three components, with their analytical formula:

1. **Functionality**: measures the ability of resources and equipment to perform the planned operations correctly, integrating data from monitoring systems and qualitative feedback from operators. Analytically:

$$f_f(i,j) = -\sum_{i=1}^{m} \frac{1}{s_i} \left(\sum_{j=1}^{n} \gamma_j D_j \right)_i \tag{1.2}$$

   where:

   - D_j: Boolean signal indicating a disruption (malfunction/non-functional condition) detected by sensors, alarms or operator observations.
   - γ_j: weight assigned to disruption j (relative importance).

2. **Quality**: assesses the level of compliance of the output with the expected specifications and unlike traditional indicators, in this context, it is considered a dynamic variable. Analytically:

$$f_q(i,j) = -\sum_{i=1}^{m} \frac{1}{s_i} \left(\sum_{j=1}^{n} \lambda_j Q_j \right)_i \tag{1.3}$$

   where:

   - Q_j: quality target expressed as a Boolean variable (1 = target achieved; 0 = not achieved).
   - λ_j: weight of quality objective j.

3. **Performance**: refers to the system's ability to converge towards efficient operating conditions, progressively reducing cycle times, scrap and rework to levels compatible with full-scale production. Analytically:

$$f_o(i,j) = -\sum_{i=1}^{m} \frac{1}{s_i} \left(\sum_{j=1}^{n} \beta_j T_j \right)_i \tag{1.4}$$

   where:

   - T_j: magnitude of deviation (continuous value) by which cycle time-related parameters (overall cycle time, phases, task durations) exceed the target.
   - β_j: weight assigned to deviation T_j.

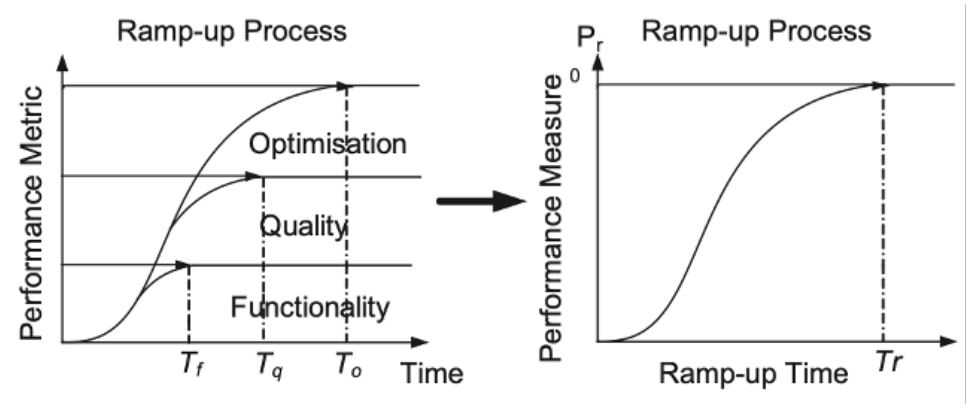

Figure 1.7: Manufacturing ramp-up index, [45].

Consequently, it becomes possible to monitor in real time the gap between the current status and the desired target, integrating the three dimensions over mentioned in an overall performance measurement index:

$$R_{PM} = w \cdot f_{RU} = \sum_{i=1}^{m} \frac{1}{s_i} \left(-w_f \left(\sum_{j=1}^{n} \gamma_j D_j \right) - w_q \left(\sum_{j=1}^{n} \lambda_j Q_j \right) - w_o \left(\sum_{j=1}^{n} \beta_j T_j \right) \right)_i \tag{1.5}$$

Where:

- $w = [w_f, w_q, w_o]$: weight vector for functionality, quality and optimization.
- $f_{RU} = [f_f, f_q, f_o]$: vector of the three sub-indices.

In summary, the index provides a summary measure of the maturity level of the production system and its degree of operational readiness.

Equations (1.2), (1.3), (1.4) and (1.5) are reproduced faithfully from [45]; any normalization/scaling is applied as indicated in the paper to allow comparisons between case studies.

In short, measuring performance during ramp-up must reflect the dynamic and progressive nature of this phase and cannot be reduced to a mere ex-post collection of indicators. The literature shows that convergence towards capacity and quality objectives depends heavily on organizational and technical learning mechanisms that develop over time. This link between measurement and learning forms the transition to the next section, which is dedicated to the analysis of learning curves applied to ramp-up.

1.4. Ramp-up and Learning Curves

A learning curve (LC) is a mathematical description of workers' performance in repetitive tasks ([48–50]). As repetitions occur, workers typically require less time to complete tasks due to their familiarity with the operations and tools, as well as the discovery of shortcuts in task execution ([48, 51]). The classic formulation, introduced by [48] in the aeronautical industry, demonstrates that the production time per unit systematically decreases as the cumulative number of produced units increases. Subsequently, learning curves have taken on a central role in the study of industrial competitiveness, so much so that several studies have extended this paradigm, adapting to different application contexts several variants of learning curves, such as log-linear, exponential, hyperbolic and logistic models ([52, 53]). In particular, several studies propose the use of power law and exponential functions to represent response time in repetitive tasks, highlighting their effectiveness in modeling human learning ([54]). Furthermore, empirical industrial models such as the *Wright Curve* are often compatible with logistic or sigmoid structures in production contexts ([55]).

During ramp-up, learning curves play a crucial role because they describe the empirical trend of progressive improvement: in the early stages, production systems typically experience high defect rates, long cycle times and unstable capacity, which are gradually reduced thanks to accumulated experience ([13, 18]). However, this improvement is less the result of structured predictive models and more the effect of organizational and human learning. As [13] point out, performance gains during ramp-up are driven by learning by doing and targeted experiments (engineering trials), which reduce technical and organizational uncertainty, rather than by the simple accumulation of production volumes.

[56] also point out that traditional learning curves are basically empirical tools that show measurable improvements in cycle times and product quality, but they assume that things are always changing and getting better. In industrial practice, the nature of the ramp-up breaks up the linearity of the process and makes it hard to use classical models directly.

Further developments have sought to extend learning curve models beyond productivity alone. [57] showed that learning is closely linked to quality stabilization and loss reduction across multiple production stages. They further elaborated the composite learning curve (i.e. Wright's learning curve, which consider the learning process in the time to produce and the time to rework a given lot), originally developed by [58], integrating the effects of learning in both production and rework processes, aiming to maximize a joint performance measure that combines average processing time and process yield. Similarly, [59] a method for modeling both the enhancement of quality and productivity. He showed that

process variability goes down as production experience builds up, using a Cobb–Douglas multiplicative power function. These contributions highlight a crucial evolution of learning theory, from purely temporal and cost-based formulations to multidimensional models that also account for quality and process reliability.

In a nutshell, during ramp-up, learning curves effectively describe the effect of accumulated experience and operational learning, but they are still just descriptive tools after the fact, since they are grounded on historical data and retrospective observations. They explain "*how*" performance improves, but they do not give a quantitative tool to predict "*how much*" and "*when*" these improvements will happen. This limitation motivates the need for ex-ante predictive models capable of integrating multiple dimensions (quality, productivity and costs) and representing the dynamics of ramp-up systematically, as will be discussed in the next section.

1.5. Predictive Algorithms for Ramp-up Performance and AI-based Quality Assessment

Unlike learning curves, which represent an ex-post view of learning, predictive models aim to anticipate the evolution of production system performance during the ramp-up phase. Recent literature has explored different directions. For example, [23] proposed an analytical and simulation-based model aimed at predicting the ramp-up behavior of manufacturing technologies and supporting the selection of the most suitable technology for a given production scenario; [16] propose an optimal control model for the joint analysis of investments in design and production capacity, with the aim of simultaneously determining the optimal time to introduce the product to the market and the duration of the ramp-up phase.

At the same time, simulation and digital twin methodologies have made it possible to estimate alternative scenarios and evaluate improvement policies under conditions of high uncertainty, supporting data-driven decisions during ramp-up ([3, 8]). In addition, data-driven and machine learning approaches have been applied to predict quality defects and system performance, leveraging complex industrial datasets ([60]).

However, as [56] observe, most predictive models suffer from significant limitations. The main ones concern poor ability to simultaneously integrate multiple variables (time, quality and costs), limited robustness in the presence of interruptions and forgetting and above all, a lack of validation on real industrial cases. These critical issues are the same as the research gaps that [8] and [3] also found. They stress the need for real operational and

decision-support tools that can combine quantitative and qualitative aspects.

1.5.1. AI-based Quality Predictive Algorithms in Manufacturing Ramp-up

In recent years, quality management in manufacturing systems has been profoundly transformed by the development of Artificial Intelligence (AI) and Machine Learning (ML) techniques, in particular during the ramp-up phase. This gave rise to a new generation of data-driven predictive models capable of automatically learning the relationships between process parameters and product quality, early detection of anomalies and estimation of process reliability ([61, 62]). In industry, supervised learning algorithms, such as Random Forest, Support Vector Machine (SVM), and Artificial Neural Networks (ANNs), are among the most widely used for defect detection and process monitoring ([61, 63]). By training these algorithms, they are able to provide real-time predictions, supporting decision-making in testing or inspection workstations and reducing overall validation times.

However, one of the main challenges in ramp-up concerns the training phase, which often takes place in conditions of data scarcity and imbalance ([64]). To mitigate this problem, several studies propose incremental learning and transfer learning techniques, in which models are progressively updated as new production data becomes available ([63, 65]). This approach mirrors the behavior of the production system itself, which gradually improves its performance thanks to accumulated experience.

In general, the evaluation and validation of a classification model's performance is typically based on the confusion matrix, which summarizes the number of correct and incorrect predictions with respect to the actual conditions of the samples and forms the basis for calculating the main indicators of the model's accuracy and reliability ([66, 67]). A generic confusion matrix for a binary classification problem is shown in Figure 1.8.

Real condition \ Model prediction	**Positive class**	**Negative class**
Positive class	True Positives (TP)	False Positives (FP)
Negative class	False Negatives (FN)	True Negatives (TN)

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{TP + FN}$$

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{F1} - \text{score} = \frac{2}{1/\text{precision} + 1/\text{recall}}$$

Figure 1.8: Confusion matrix and common performance metrics, adapted from [66].

Given the industrial nature of this work, and in particular the focus on quality inspection, the confusion matrix can be directly mapped to the decision process of classifying system outputs as *conforming* or *non-conforming* ([68]), also referred to as OK and NOK respectively. In this context, the positive class corresponds to defective parts (NOK) and the negative class corresponds to conforming parts (OK). According to the classical Statistical Quality Control (SQC) theory ([69]), this representation allows the identification of two main sources of classification error:

- **Type I error (α)**, i.e., the probability of rejecting a conforming unit, also known as *producer's risk*.
- **Type II error (β)**, i.e., the probability of accepting a defective unit, also known as *consumer's risk*.

This terminology is also reflected in international quality standards such as ISO 2859-1:1999 ([70]), which establishes standard procedures for acceptance sampling by attributes and defines the statistical principles underlying sampling plans indexed by the Acceptable Quality Limit (AQL). Thus, it implicitly incorporates the notions of producer's risk (α) and consumer's risk (β) as the two fundamental probabilities governing acceptance and rejection decisions. Moreover, this interpretation has been explicitly discussed and critically analyzed by [71].

Model prediction / Real condition	NOK (Defective)	OK (Conforming)
NOK (Defective)	True Positives (TP)	False Negatives (FN)
OK (Conforming)	False Positives (FP)	True Negatives (TN)

Figure 1.9: Confusion matrix in industrial quality control, adapted from [68].

As mentioned before, the indicators in Figure 1.8 provide a quantitative measure of the model's reliability. However, in an industrial context, they can also be directly linked to the economic and qualitative consequences of classification decisions. In fact, the recall and precision indicators can represent, respectively, the risk of defective products reaching the customer and unnecessary scrap or rework, resulting in a loss of productivity.

Although the framework presented in Figure 1.9 is well established in industrial applications, it is often used in a static form, limited to post-process evaluation of model

performance. In contrast, the approach developed in this thesis reformulates the confusion matrix within a dynamic and decision-based framework, in which classification outcomes are associated with system activation choices. This generalization enables the probabilistic modeling of α and β as evolving parameters within the ramp-up phase, whose interpretation depends on the adopted decision framework, as detailed in Chapter 3.

1.6. Thesis Positioning

This thesis aims to introduce a predictive model based on a generalized logistic curve, which aims to formalize the relationship between operational learning of an AI-based algorithm and the maturation of the production system, through the progressive reduction in Type I and Type II error probabilities (α and β), interpreted as experience-dependent error probabilities of the AI-based inspection system, as the cumulative production volume increases. In other terms, the model developed aims to quantitatively represent the evolution of quality during the ramp-up phase.

The proposed approach allows:

- Linking learning curve theory with an explicitly predictive formulation, transforming an ex-post empirical model into an ex-ante tool capable of estimating future performance trends.
- Integrating productivity and quality into a single analytical framework, overcoming the traditional separation between efficiency metrics and defect indicators.
- Simulate alternative scenarios and support decision-making during ramp-up, providing a quantitative basis for capacity planning and managing trade-offs between time, cost and quality.

In this way, this thesis contributes to addressing one of the main gaps identified in the literature ([3, 8, 45]), namely the lack of predictive tools capable of jointly representing learning dynamics and error reduction during industrial ramp-up. The model developed provides a generalizable, data-driven method that can be calibrated using real industrial data, allowing for more accurate estimates of the trajectory of quality and productivity improvement during the ramp-up phases. In this sense, the proposed approach represents a significant evolution compared to traditional descriptive models, offering a quantitative tool to support operational decisions during the most critical phases of the production system life cycle.

The existing literature on learning and ramp-up modeling can be divided into two main

areas of research. The first one concerns empirical and descriptive models, which focus on quantifying improvements resulting from operational experience and learning by doing. Whereas, the second one regards predictive and simulation-based models, which aim to anticipate performance evolution by integrating analytical formulations or data-driven approaches.

While the former works ([48, 52]) provide a static ex-post representation of learning, subsequent studies such as [58] introduce analytical formulations with predictive purposes, capable of estimating the evolution of productivity and rework, but still limited to a deterministic description of the phenomenon. At the same time, [13] and [19] explicitly applied the concept of learning curves to the ramp-up phase, developing analytical models based on dynamic programming to optimize the trade-off between production efficiency and operational learning. Although these models mathematically formalize the learning process as a decision-making dynamic, they remain confined to a deterministic framework and do not provide an explicit representation of the learning function or a predictive quantification of system performance. More recent contributions ([3, 8, 23]) mark the transition to dynamic and data-driven paradigms, characterized by adaptive forecasting capabilities applicable to complex industrial contexts. While several studies have progressively incorporated learning curves within predictive frameworks (e.g., [13], [57] and [58]), none of them explicitly integrates a mathematically defined learning function within a predictive model describing the behavior of artificial intelligence-based inspection systems during the ramp-up phase. The proposed thesis addresses this gap by introducing a generalized logistic formulation that quantitatively models the reduction of Type I and Type II errors (α and β) as a function of production experience.

This evolution in the literature is summarized in Table 1.1, which classifies the main contributions on the topic of learning curves and predictive models applied to production ramp-up. The table highlights the gradual transition from empirical and descriptive models to analytical formulations and, more recently, to dynamic and data-driven approaches, up to the integration of learning curves and predictive algorithms, as proposed in this thesis.

To facilitate the interpretation of the table, a brief explanation of its structure is provided below. The column "Model Formulation (Learning/Predictive Type)" describes the mathematical or conceptual structure of each model, specifying whether it is an empirical, analytical or simulation-based approach. The "LC" column indicates whether the work explicitly includes a learning curve. A check mark indicates that the learning curve is an integral part of the mathematical model, while the absence of a check mark indicates that learning is not formalized. The "Ramp-up" column indicates whether the model explicitly

addresses the ramp-up phase. The column "Predictive Modeling Approach" distinguishes the nature of the predictive approach. Indeed, the presence of a check mark indicates that the model has predictive purposes, either through analytical formulations or mathematical simulations, or through data-driven or machine learning approaches. Finally, the last column of the table does not refer exclusively to AI applications, but more generally to the use of the learning curve as a functional component of the predictive model or whether the learning curve is only descriptive (check mark missing).

Reference	Model Formulation (Learning/Predictive Type)	Limitations/Notes	LC	Ramp-up	Predictive Modeling Approach	LC integrated in Predictive Modeling
Wright [48]	Power-law learning curve	Descriptive; assumes constant learning rate.	✓			
Yelle [52]	Log-linear learning curve variants; comparative review of exponential, power-law, and S-shaped formulations	Identifies external factors (organization, technology, labor) affecting learning; purely descriptive.	✓			
Terwiesch & Bohn [13]	Analytical, dynamic programming	Deterministic model; absence of stochastic uncertainty and empirical validation.	✓	✓	✓	✓
Terwiesch & Xu [19]	Analytical, decision-theoretic	The approach is theoretical and qualitative; it does not include an explicit quantitative formulation of the learning curve.	✓	✓	✓	
Jaber & Guiffrida [58]	Composite learning curve (production+rework)	Quality evolution not explicit; limited validation.	✓		✓	✓
Jaber & Khan [57]	Composite learning with yield/rework/scrap	Focus on yield and lot splitting; predictive scope limited.	✓		✓	✓
Anzanello & Fogliatto [56]	Univariate, multivariate and forgetting learning curve	Review integrating learning, forgetting, and relearning; models mainly descriptive.	✓			
Klocke et al. [23]	Analytical and hybrid simulation model for predicting ramp-up performance	Predictive analytical-simulation framework estimating throughput and process disturbances during ramp-up.		✓	✓	
Colledani et al. [3]	Analytical framework for ramp-up quality improvement integrating production, logistics and maintenance	Strong for improvement loop; lacks explicit learning curve link.		✓	✓	
Schmitt et al. [8]	Conceptual and mixed-method framework (digital twin/cyber-physical system)	Emphasizes data-driven, real-time learning and decision support.		✓	✓	
This Thesis (2025)	**Logistic learning-based predictive model for ramp-up quality performance**	**Introduces a data-calibrated predictive model linking learning curves to quality performance; formalizes reduction of error probabilities as a function of cumulative production experience.**	✓	✓	✓	✓

Table 1.1: Evolution of Learning Curve and Predictive Models in Ramp-up Research.

2 | Industrial Problem and Objectives

This chapter undertakes an analysis of the industrial problem that forms the basis of the research, with particular reference to the ramp-up phase in manufacturing systems. The first section focuses on outlining the industrial context of the problem, highlighting the main operational and organizational challenges of the ramp-up. Instead, the second section formalizes the research question and scientific objectives of the thesis. This part introduces a causal map that describes the relationships between the different factors that characterize the ramp-up phase.

2.1. Industrial Context and Problem Relevance

As outlined in Section 1.1, the production ramp-up phase, during which production capacity gradually increases until steady state is reached, is one of the most critical stages in the life cycle of a manufacturing system. This is a period in which process instability, high quality variability, inspection errors and operational inefficiencies occur. All these significantly affect costs, industrialization times and the overall profitability of the production system.

In the automotive sector, as stated by [3] for example, the average duration of the ramp-up phase can vary between 20 and 30 weeks, with a significant impact on costs related to scrap, rework, and quality losses, which represent a large portion of the production value during the initial weeks of the ramp-up phase ([29, 30]). Similar phenomena can also be observed in other highly complex technological sectors, such as electronics, aerospace and mechatronics, where ramp-up is particularly critical due to the interaction between new processes, innovative components and global supply chains ([72–74]).

From an organizational point of view, ramp-up results in significant pressure on human resources and the supply chain. Furthermore, a lack of coordination and communication between the design, production and logistics departments is one of the main causes

of organizational inefficiencies during production ramp-up ([1]). At the same time, the growing complexity of production systems, the introduction of digital and cyber-physical technologies and the increase in product customization amplify the difficulty of predictively managing process behavior during ramp-up. In this scenario, the availability of reliable predictive models capable of quantitatively describing the evolution of performance and quality becomes a key element in accelerating system stabilization, reducing non-quality costs and improving the overall competitiveness of the firm ([3, 8, 32]).

Finally, as highlighted by [13], ramp-up presents a real learning paradox: companies must balance the need to meet market demand with the need to devote resources to learning and improving the process. The moment of maximum learning potential coincides with maximum instability and scarcity of operational information, making this phase critical from both a technical and managerial perspective.

In particular, the main industrial challenges can be summarized as follows:

- **High inspection effort & costs**: currently, 100% of products undergo thorough inspection, which slows productivity.
- **Data shortages & Changing line conditions**: during the ramp-up phase, limited data makes it difficult to develop accurate predictive models.
- **Process bottlenecks & Line reconfiguration**: as new digital models are introduced, system bottlenecks shift, requiring continuous rebalancing of production lines.
- **Ensuring model accuracy**: to guide adjustments without increasing rework or reducing yield, process control models must be precise enough.

To address these challenges, a number of lines of action have been identified:

- Develop data-driven models to optimize the inspection process and minimize manual effort.
- Implement real-time process control using machine learning models for anomaly detection and root-cause analysis.
- Enhance production flexibility through simulation models, allowing for rapid reconfiguration during ramp-up.
- Improve overall system performance by balancing inspection, adjustment time and yield rates.

2.2. Research Problem Definition and Objectives

Among the courses of action proposed at the end of the previous section, this thesis focuses primarily on the development of data-driven models for the optimization of inspection processes, with the aim of supporting operational and managerial decisions in the early stages of production. In particular, the proposed approach emulates the learning behavior of artificial intelligence-based inspection systems in conditions of limited data availability, providing a predictive framework capable of estimating how decision error probabilities evolve with cumulative production experience.

The main problem of the thesis can therefore be formulated at two complementary levels.

From a modeling perspective:

"How can the evolution of the decision reliability of an automatic inspection system based on artificial intelligence (AI) be represented quantitatively and predictively during the ramp-up phase, as a function of cumulative production experience?"

From a governance perspective:

"Given such reliability evolution, how should the activation timing of learning be determined when cumulative risk exposure is explicitly considered?"

This dual formulation reflects the two-step contribution of the thesis: first, the formalization of learning-dependent reliability dynamics; second, the evaluation of activation policies under asymmetric risk structures.

To answer these research questions, a series of scientific and application objectives have been set, which can be summarized in the following main points:

1. Develop a generalized mathematical model capable of quantitatively representing the evolution of the performance of an automatic inspection system during the ramp-up phase, describing the progressive reduction in the probabilities of Type I and II errors (α and β), interpreted as learning-dependent error probabilities of the AI-based inspection algorithm, as a function of cumulative first-pass inspection volume and operational experience.

2. Implement the model within a simulation environment to analyze alternative activation scenarios, estimate the evolution of quality and cumulative risk exposure and support governance decisions related to inspection planning and system stabilization.

3. This thesis provides a methodological contribution by integrating a formally de-

fined learning function into a predictive decision framework and by establishing a structured approach for evaluating activation timing as a risk-sensitive governance variable during production ramp-up. In doing so, it advances the existing literature by explicitly modeling the evolution of decision error probabilities within an AI-based inspection architecture and by linking learning dynamics to cumulative risk exposure.

The achievement of these objectives requires the definition of a coherent methodological framework capable of integrating the analytical formulation of the model with its application in different production contexts. Therefore, the following chapter describes the methodological architecture adopted, illustrating the structure of the proposed model, the key variables that characterize it and the process of validation and use for the purposes of predictive analysis of the system's behavior during ramp-up.

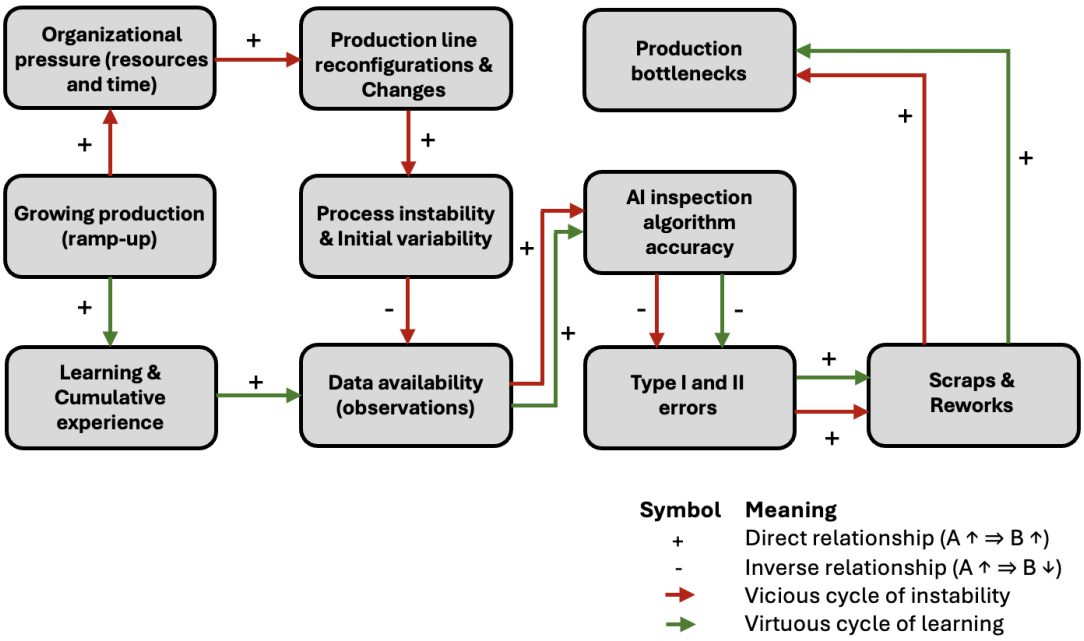

Figure 2.1: Causal map of the industrial problem during the ramp-up phase.

Figure 2.1 illustrates the causal map that summarizes the main relationships between the variables involved in the production ramp-up phase and the behavior of artificial intelligence-based inspection systems, representing:

- a *vicious cycle of instability*, which amplifies the negative effects in the initial ramp-up phase;
- a *virtuous cycle of learning*, which over time allows the system to stabilize and progressively improve quality and productivity.

Vicious cycle of instability

In the early stages of production, increased volumes (growing production) are combined with increased organizational pressure in terms of resources and time, which in turn increases the number of reconfigurations and line changes. These factors generate process instability and initial variability, reducing the availability of accurate and sufficient data for training AI inspection models. The low accuracy of the algorithm leads to an high value in terms of Type I and II errors, which result in an high level of scrap and rework. These, in turn, reduce effective productivity and increase bottlenecks in the production flow. The result is a negative self-reinforcing loop, in which high variability and a lack of reliable data prevent effective learning and slow convergence towards operational stability.

Virtuous cycle of learning

However, as production progresses, the accumulation of operational experience and observations triggers a gradual positive feedback mechanism. The increase in available data improves the accuracy of AI-based inspection models, reducing the likelihood of Type I and II errors and, consequently, scrap and rework, freeing up production resources and reduces congestion in processes. This virtuous cycle of learning gradually leads the system towards a stable operating regime, characterized by increasing performance in terms of quality, efficiency and reliability.

The model proposed in this thesis aims to quantify this transition, mathematically formalizing the relationship between production volume, algorithm learning and reduction in inspection errors, and explicitly analyzing how the timing of adaptive activation affects cumulative system performance.

3 | Methodological Framework and Model Formulation

This chapter outlines the methodological framework adopted for predictive modeling of the learning behavior of an AI-based automatic inspection system during the production ramp-up phase. The objective is to formalize a quantitative approach capable of representing the relationship between cumulative first-pass inspection volume, used as a proxy for production experience, and the progressive reduction of inspection errors driven by the learning dynamics of the AI-based inspection algorithm. In this way, the evolution of predictive performance during ramp-up is analytically represented.

Rather than reproducing a specific data-driven classifier trained on proprietary historical datasets, the proposed framework abstracts the learning effect through a parametric statistical representation ([75]). The objective is not to replicate the internal training process of a particular ML model, but to model the structural evolution of decision reliability as production experience accumulates ([76]).

Accordingly, the concept of learning rate specific to machine learning is replaced by a prescribed predictive reliability curve that evolves as a function of cumulative first-pass inspection volume. This formulation does not "learn" in the conventional algorithmic sense; instead, it emulates the operational effect of learning by imposing a mathematically defined improvement trajectory. The guiding principle is therefore to represent the system's operational maturity in a generalizable and analytically tractable manner, independently of any specific dataset or classifier implementation.

3.1. Predictive Model Formulation and Functional Architecture

In order to address the need to quantitatively and predictively describe the evolution of the accuracy of AI inspection systems in the early stages of production, a parametric predictive model based on a generalized logistic curve is proposed. The model describes the

progressive reduction in the probabilities of Type I and II errors, interpreted as learning-dependent error probabilities of the AI-based inspection algorithm, as the cumulative first-pass inspection volume increases. In this way, the logistic curve acts as a simulated accuracy curve, which reproduces the learning behavior of the algorithm even in the absence of real training data.

This approach allows the learning speed of the inspection system to be represented quantitatively and convergence towards a stable regime to be predicted, thus providing a support tool for optimizing operating parameters and planning quality improvement strategies. Moreover, it is intentionally replicable, not tied to a specific machine learning algorithm, and can be applied to any predictive model that improves its performance as the amount of observed data increases. This makes it particularly suitable for integration into industrial simulation or digital twin environments during ramp-up phases.

The challenges and objectives defined in Chapter 2 led to the design of an integrated functional architecture, conceived to integrate analysis, forecasting and control modules, predictive models and process data, providing a coherent methodological framework. The objective of the architecture is to dynamically represent the interactions between the actual production system, the predictive inspection model and the performance evaluation system in order to monitor and optimize quality evolution during the ramp-up phase.

The proposed functional architecture has been developed to integrate two dimensions of the ramp-up phase that are usually treated as separate domains within manufacturing systems:

1. The technological and operational layer, focused on process reconfigurations, bottleneck analysis and efficiency improvement.
2. The data-driven quality layer, dedicated to the learning and validation of AI-based inspection models.

The overall architecture can be interpreted through two configurations:

- The **AS-IS configuration**, in which process analysis and AI training operate separately and do not exchange information.
- The **TO-BE Decision Support System (DSS) configuration**, in which both layers are integrated through a unified feedback system enabling joint optimization.

Figure 3.1 outlines the functional architecture of the AS-IS configuration, which is organized into four main functional modules:

1. **Production System**: represents the real manufacturing environment, consisting

of physical resources, operators and process flows. It produces the actual parts and generates operational data (such as cycle times, throughput and defect occurrences), which feed both the process-analysis and the AI-learning pipelines. In the AS-IS industrial condition, this system provides two parallel data outputs, i.e., process data for system efficiency analysis and product data for AI training.

2. **Data Collection**: acts as the interface between the shop floor and the analytical layer. It acquires, filters, and stores process and product data in a consistent and traceable format (through MES or data lake systems), making it available for further analysis and model training.

3. **System Analysis**: identifies the key process parameters (e.g., cycle time, resource utilization, defect rate) and in the AS-IS configuration, this module works independently from the AI training process, focusing solely on the identification of inefficiencies and bottlenecks within the production flow.

4. **Training for Quality Prediction Model**: implements the machine learning algorithm that based on the data collected, is trained to estimate product quality or the probability of defects based on process variables. The model outputs the predicted process variable and, by comparing its decision with the actual quality outcome of each part, it allows calculating empirical classification error rates (Type I and II), which represent the observed predictive performance of the AI algorithm at system level.

In the current (AS-IS) industrial setup, Process Analysis and Training for Quality Prediction Model modules act as two independent feedback loops that interact separately with the Production System. The process branch (Data Collection $\rightarrow$ System Analysis) extracts performance indicators and sends a direct feedback to the Production System, typically aimed at resolving bottlenecks or adjusting cycle times. The AI branch (Data Collection $\rightarrow$ Training for Quality Prediction Model) trains and validates the algorithm, generating α and β empirical errors and sending its own feedback to the Production System, such as activating or tuning the inspection model. However, these two levels do not communicate with each other. As a result, while the process optimization loop may seek to reduce inspection time to improve throughput and resolve bottlenecks, the AI loop might require more time for inspections to learn effectively, producing conflicting objectives. This configuration corresponds to the AS-IS situation: data are extracted from the ramp-up system but flow into two distinct analysis streams (process and product), without an integrated learning logic.

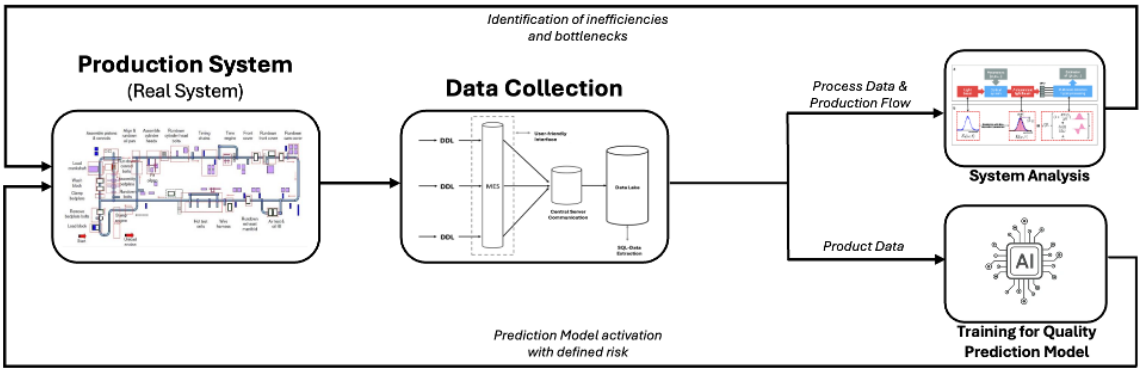

Figure 3.1: AS-IS configuration - Functional Architecture.

This thesis proposes an evolutionary configuration of the AS-IS setup of the functional architecture, aimed at representing and integrating the interaction between the two traditionally separate dimensions of ramp-up. The goal is to overcome the separation typical of the AS-IS model, in which process analysis and AI model training operate independently, towards a TO-BE configuration based on a continuous and coordinated feedback system. While in the AS-IS configuration the error rates α and β are empirically estimated from observed classification outcomes, in the proposed TO-BE configuration they are no longer inferred from the data, but prescribed as learning-dependent parameters to emulate different stages of AI maturity.

3.1.1. Proposed TO-BE Functional Architecture Configuration

Figure 3.2 presents the functional architecture of the proposed TO-BE configuration, designed to support the analysis and management of production ramp-up phases in the presence of AI-based quality inspection systems. The framework integrates production, analytical and decision-making layers into a unified and coherent structure, explicitly accounting for the evolution of AI performance as production experience accumulates. Unlike conventional approaches, the proposed configuration does not focus on the explicit execution or optimization of AI training procedures. Instead, it adopts a model-based representation of AI learning, enabling the evaluation of system-level performance at different stages of AI maturity during ramp-up.

Unchanged Production and Data Acquisition Layers

The Production System and Data Collection modules remain conceptually and operationally unchanged with respect to the AS-IS configuration, indeed it executes the manufacturing process, generating physical parts and operational data. Instead, the data collection layer acts as an interface between the shop floor and the analytical modules,

aggregating production information and updating the cumulative production experience index v, defined as the cumulative number of first-pass inspected units. This variable represents the system's operational experience and constitutes the key driver for the subsequent analytical evaluation of the learning model.

Training for Quality Prediction Model (Learning Emulation)

Within the proposed framework, the Training for Quality Prediction Model does not perform an explicit data-driven training or retraining of an AI algorithm. Instead, it defines the assumed learning behavior of the AI-based quality inspection system through parametric accuracy functions. Specifically, the evolution of the inspection performance is modeled by predefined functions α and β, representing the learning-dependent Type I and Type II error probabilities of the AI-based inspection algorithm as functions of the cumulative first-pass inspection volume v. These functions describe how the accuracy of the AI system is expected to improve as production experience increases and therefore constitute a theoretical representation of the learning process. The learning dynamics are defined a priori and are not modified during the simulation. As a result, the framework does not aim at validating or updating the AI model itself, but at analyzing the implications of different AI maturity levels on production performance. In the TO-BE configuration, the Training for Quality Prediction Model does not perform data-driven learning, but acts as a surrogate of the AI training process, instantiating predefined accuracy parameters that represent different stages of algorithm maturity as a function of cumulative production experience v. The Training for Quality Prediction Model defines the AI learning behavior as continuous accuracy functions of the experience variable v. The Evaluation Kernel then "reads" the curves defined in the Training module and instantiates these functions at the current cumulative first-pass inspection volume v, yielding the operational parameters $\hat{\alpha}(v)$ and $\hat{\beta}(v)$. The Training for Quality Prediction Model is not part of the operational loop. It only defines the assumed learning behavior during the system setup phase and provides static reference functions that are instantiated by the Evaluation Kernel at each batch checkpoint during runtime.

Evaluation Kernel

The Evaluation Kernel represents the analytical core of the proposed TO-BE configuration. Its role is to instantiate the assumed learning model at a specific point of the ramp-up phase. At the beginning of each cycle, the Evaluation Kernel receives the current cumulative first-pass inspection volume $v(t)$ and evaluates the predefined learning curves, obtaining the expected accuracy parameters $\hat{\alpha}(v(t))$ and $\hat{\beta}(v(t))$. These values represent the expected operational accuracy of the AI inspection system at the given level of production experience. By operating in this way, the framework does not evaluate the

AI model in the abstract, but emulates its behavior as if it had reached the corresponding level of maturity defined by the learning curves.

Performance Evaluation Module

At each iteration, the accuracy parameters correspond to the learning curve evaluated at the cumulative first-pass inspection volume reached at the end of the previous cycle. The progression of learning is therefore driven by the evolution of production experience, rather than by direct updates of the learning model. The Performance Evaluation module represents the operational verification layer of the framework. It receives the expected accuracy parameters ($\hat{\alpha}$ and $\hat{\beta}$) from the Evaluation Kernel and uses them as reference conditions to simulate the behavior of the production system. The module can be implemented either as a digital simulation model or as a monitored physical system. During execution, it reproduces the effects of the AI-based inspection performance on the production process, evaluating key metrics such as productivity, quality, throughput and operational stability. As production progresses, the cumulative volume increases from $v(t)$ to $v(t+1)$ and at the end of each cycle, the Performance Evaluation module returns the updated production volume and observed system performance metrics to the Evaluation Kernel. This feedback is not used to update the learning model, which remains fixed by assumption, but to verify that the simulated system behavior remains coherent with the imposed AI learning hypothesis.

Decision Support System (DSS) for Ramp-up

The Decision Support System for Ramp-up, which constitutes the decision-making layer at the top of the framework, receives performance indicators, the cumulative production experience index v and the associated accuracy parameters from the analytical modules, and compares them with predefined acceptance thresholds. Based on this comparison, the DSS determines whether the current ramp-up phase can be validated and the system can progress towards stable production, or whether a new iteration is required. In this case, the DSS may regulate the progression of the ramp-up by maintaining the simulation active and updating the target production volume, defined as a decision variable governing the progression of the ramp-up phase, without modifying the assumed learning dynamics. In the current structure, the DSS operates as a supervised decision-making process. However, the framework is designed to support future automated extensions, enabling real-time decision-making on ramp-up progression while maintaining a balance between AI maturity, production stability and efficiency objectives. Therefore, the DSS for ramp-up evaluates whether the system performance achieved at the current cumulative first-pass inspection volume satisfies predefined acceptance criteria. If the criteria are met, the current ramp-up phase is validated and the system is allowed to progress.

Otherwise, the learning cycle is iterated by extending production, without modifying the assumed AI learning dynamics. In the proposed framework, the Decision Support System evaluates ramp-up progression primarily based on throughput thresholds, using observed performance metrics to determine whether the system has reached a sufficiently stable operational regime.

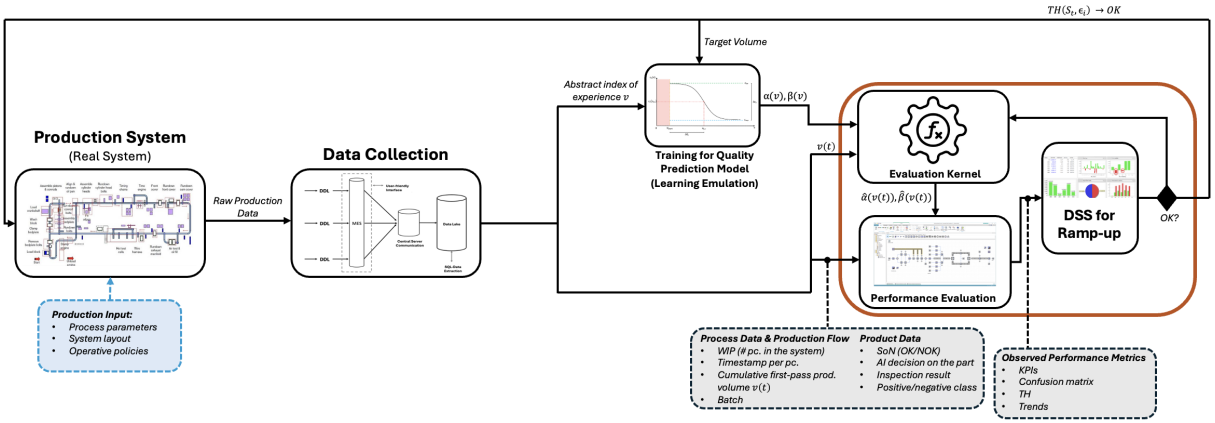

Figure 3.2: TO-BE configuration - Functional Architecture.

The proposed functional architecture integrates the productive, analytical and decision-making dimensions into a single continuous feedback loop, allowing for a consistent representation of the predictive system's learning evolution during ramp-up. Starting from this structure, the following section introduces the mathematical formalization of the AI model's learning dynamics.

3.1.2. System Modeling of the AI-based Inspection System

In order to analyze the impact of artificial intelligence maturity on production ramp-up performance, it is first necessary to formalize how AI-based inspection decisions are represented at the system level. Rather than focusing on the internal structure of a specific machine learning algorithm, the proposed model abstracts the behavior of the AI inspection system through a simplified and interpretable decision mechanism.

The objective of this section is therefore to introduce a conceptual system-level representation of AI-driven inspection decisions, explicitly separating deterministic operational policies from learning-related uncertainty. This abstraction enables the analysis of AI maturity effects on production performance without introducing assumptions on explicit training procedures or algorithmic updates.

```
DecisionAI = baseDecision + AI-noise
```

With:

- `DecisionAI`: deterministic policy (tolerances and confidence intervals).
- `AI-noise`: stochastic flip with probability $\alpha(v)$, $\beta(v)$.

In the proposed model, the AI-based inspection system is represented as a two-layer decision mechanism:

- **1st layer**: consists of a deterministic decision rule based on fixed tolerance thresholds and confidence bounds, which reflects the operational inspection policy adopted in industrial environments. This policy translates continuous quality predictions into discrete production decisions (e.g., activation of a full inspection cycle).
- **2nd layer**: models AI maturity as a stochastic decision noise component, whose intensity is governed by the volume dependent error probabilities $\alpha(v)$ and $\beta(v)$. These parameters represent the likelihood that the AI introduces misclassifications beyond the deterministic policy, due to instability, immaturity or unreliable behavior typically observed during early ramp-up phases.

As cumulative production experience increases, $\alpha(v)$ and $\beta(v)$ decrease, leading to a progressively more stable and reliable execution of the same decision policy. In other terms, the AI maturity is modeled exclusively as a reduction in decision uncertainty, while the continuous prediction mechanism itself is assumed stationary: learning affects decision reliability[1] rather than prediction accuracy, consistently reflecting industrial scenarios where model usage becomes more robust over time without immediate retraining.

This modeling approach is particularly robust as it explicitly decouples decision policy from learning-related uncertainty, allowing the isolated analysis of AI maturity effects on system performance. By avoiding the simulation of explicit retraining or parameter updates, the framework prevents the introduction of artificial learning dynamics and focuses on operational reliability. As a result, the model provides a transparent and realistic representation of AI integration without overstating learning capabilities that are not explicitly implemented.

Finally, in order to ensure a physically consistent representation of the inspection signal, the prediction variable used by the deterministic decision rule is statistically conditioned

[1]Decision reliability is defined as the probability that the AI executes the intended decision rule without introducing additional stochastic misclassifications.

on the underlying ground-truth state of the part (SoN). Non-conforming parts are assumed to generate prediction values that are, on average, closer to tolerance boundaries or outside specification limits, reflecting the fact that defective units typically exhibit measurable deviations from nominal behavior. This modeling assumption does not modify the decision architecture, it simply ensures that the deterministic policy operates on an informative signal rather than on a purely random variable, preserving the interpretability and structural consistency of the framework.

3.1.3. Modeling Assumption

The proposed framework adopts a system-level modeling approach to analyze the interaction between AI-based quality inspection and production ramp-up dynamics. In order to maintain analytical tractability and focus on production performance implications, a number of modeling assumptions are introduced.

First, it is assumed that an AI-based quality inspection model exists and operates in the real TO-BE production system. In practice, such a model would be trained and updated using product-level data, including features, inspection outcomes, and ground-truth quality labels. However, the internal structure, training procedures and retraining mechanisms of the AI model are not explicitly represented within the proposed framework. Instead, the learning behavior of the AI system is abstracted through parametric accuracy functions, namely $\alpha(v)$ and $\beta(v)$, representing the Type I and Type II error rates as functions of the cumulative first-pass inspection volume v. These functions provide a surrogate representation of the AI learning dynamics and describe how inspection performance is expected to evolve as production experience accumulates. The learning curves are defined a priori and remain fixed.

Second, product-level data are not used to update or calibrate the learning curves and since they include the actual quality state of each part and the corresponding inspection outcome, are employed exclusively within the Performance Evaluation module to assess the operational impact of the assumed AI accuracy on system performance. This includes the computation of confusion matrices, quality indicators, their effects on throughput, stability and productivity.

Third, the cumulative production experience index v is treated as a system-level process variable derived from the production flow. It represents an index of system experience and AI maturity. This variable is used to instantiate the learning curves at discrete evaluation points during the ramp-up phase. The numerical values $\hat{\alpha}$ and $\hat{\beta}$ denote the operational accuracy parameters applied at a given production volume and do not correspond to

empirical estimations from observed data.

Finally, the Decision Support System (DSS) operates at a supervisory level and does not influence the assumed AI learning dynamics. In fact, its role is limited to evaluating whether the observed system performance, including throughput and quality-related indicators, satisfies predefined acceptance criteria, thus determining whether the ramp-up phase can progress or additional production experience is required.

Overall, these assumptions allow the proposed framework to focus on the system-level effects of AI maturity during production ramp-up, without loss of generality with respect to the actual implementation of AI models in real industrial environments.

3.2. Mathematical Modeling of AI Learning Dynamics via Generalized Logistic Function

This section defines the mathematical model developed to represent the evolution of the performance of an AI inspection system during the ramp-up phase of a production system. The objective is to identify a function capable of coherently describing the relationship between production experience and the progressive reduction of Type I and Type II inspection errors. For this purpose, a generalized logistic curve is introduced. It has been chosen for its ability to model nonlinear and saturating learning processes. The following paragraphs formalize the mathematical aspects of the function, discuss its main properties and present the analytical procedure for estimating the characteristic parameters necessary for calibrating the model.

3.2.1. Modeling Objectives

In the current thesis, the objective of the modeling is to represent the trend of an error quantity $\varepsilon(v)$, especially the Type I (α) and Type II (β) errors, as a function of the volume of data available v. This approach is applied to the ramp-up phase of a production assembly line. In such phase, the increase in the number of units produced is also interpreted as the accumulation of experience or data that can be used to improve system performance. Therefore, the aim is to model a function that can consistently represent the learning process of an AI-based algorithm, during the ramp-up phase, and which satisfies the following properties:

- **P1** – For a very low volume of data ($v \to -\infty$), the error is maximized, thus $\varepsilon(v) \to \varepsilon_0$.

- **P2** – For a very high volume of data ($v \to +\infty$), the error tends to an asymptotic minimum, thus $\varepsilon(v) \to \varepsilon_\infty$.
- **P3** – The error decreases monotonically, continuously and with a controllable slope.
- **P4** – The transition between ε_0 and ε_∞ is concentrated around a critical volume of data v_f, which can be interpreted as the inflection point of the curve, corresponding to the most sensitive portion of the ramp-up.

Among the families of functions that satisfy these requirements, the Generalized Logistic Function (GLF) has been selected, as it provides a flexible and analytically tractable representation of the error reduction during ramp-up phase.

Moreover, in the proposed framework, the learning behavior of the AI-based quality inspection system is not modeled as an explicit function of time. It is expressed as a function of production experience. Formally, the AI accuracy is described by the parametric functions $\alpha(v)$ and $\beta(v)$, which depend exclusively on the experience variable v and not on the simulation time or cycle index. This modeling choice reflects the assumption that improvements in AI performance are primarily driven by exposure to production data rather than by the mere passage of time. Time enters the framework only indirectly through the evolution of the production process. As production progresses, the cumulative volume increases from $v(t)$ to $v(t+1)$, and the learning curves are instantiated at the corresponding experience level. Consequently, the operational accuracy parameters applied during a given cycle are obtained as $\hat{\alpha}(v(t))$ and $\hat{\beta}(v(t))$, without introducing any explicit temporal dependency in the learning model. This distinction allows for the separation of learning dynamics from production timing effects. Changes in AI performance are therefore associated with accumulated experience, while temporal aspects such as throughput, cycle time and production rate are treated independently within the Performance Evaluation layer. This approach is consistent with system-level modeling practices and avoids conflating learning effects with time-driven dynamics.

3.2.2. Logistic Function as Modeling Framework for Predictive Error Dynamics

To satisfy the above constraints, it has been adopted the Logistic Function as basic structure, defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

which has a sigmoid trend, limited in the interval (0, 1), monotone increasing and with

inflection at $x = 0$.

In Equation (3.1), the generic variable x is replaced by v, indicating the function's dependence on the cumulative first-pass inspection volume of a generic quality control station. This choice allows the logistic curve to be formally linked to the performance evaluation context in which the experience of the model grows over time.

To obtain a decreasing behavior and move the inflection point from $v = 0$ to an arbitrary point v_f, a variant of the Logistic Curve is considered:

$$\sigma(v) = \frac{1}{1 + e^{k(v - v_f)}} \tag{3.2}$$

where:

- k: parameter controlling the slope of the curve. For $k > 0$, the curve is monotonous decreasing.
- v_f: inflection point.

It is worth noting that, although the standard logistic function is expressed with a negative exponent, i.e. e^{-x}, the current formulation adopts a positive exponent $e^{k\left(v - v_f\right)}$ in order to produce a monotonically decreasing behavior. As a result, the learning rate parameter k corresponds to $-B$ in the canonical form of the Generalized Logistic Function (GLF), as will be shown later [77]. Moreover, the term $(v - v_f)$ to the exponent allows to horizontally translate the inflection point of the curve, so as to position the maximum error transition at the point $v = v_f$. This allows the error transition to be modeled at the data volume where the most relevant jump in model performance occurs.

Since Equation (3.2) returns values in the range (0, 1), to model an error that transitions from an initial value ε_0 to an asymptotic one ε_∞, an affine transformation with the following form is performed:

$$\varepsilon(v) = \varepsilon_\infty + (\varepsilon_0 - \varepsilon_\infty) \cdot \sigma(v) \tag{3.3}$$

Substituting the expression of $\sigma\,(v)$, it is possible to obtain the following:

$$\varepsilon(v) = \varepsilon_\infty + \frac{\varepsilon_0 - \varepsilon_\infty}{1 + e^{k\,(v - v_f)}} \tag{3.4}$$

where:

- $\varepsilon\,(v) \in \alpha(v),\ \beta(v)$: decision error probability (depending on experience).

- ε_0: maximum asymptotic error.
- ε_∞: minimum asymptotic error.
- k: slope coefficient that governs the speed of improvement.
- $v_f = V_{start} + \Delta V$: data volume corresponding to the inflection point of the curve.
  - V_{start}: threshold (in terms of production volume) that activates the learning-based decision refinement.
  - ΔV: parameter in terms of production volume that determines the distance of the curve's inflection from V_{start}.

Table 3.1 lists the properties of Equation (3.4):

Property	**Motivation**
Realism	It reflects the progressive learning of an AI system in a production environment (ramp-up phase).
Controllability	It is possible to easily set the initial error, target value and learning speed.
Modularity	It can be integrated as a direct function within programming languages (e.g. Matlab) and simulation software (e.g. Tecnomatix Plant Simulation).
Interpretability	Each parameter has its own engineering interpretation.
Flexibility	It can be adapted for different scenarios (shifts, operators, buffer capacity, etc.).

Table 3.1: Properties of the logistic equation.

It can be notice that Equation (3.4) has the same structure as the GLF:

$$f(x) = A + \frac{K - A}{(C + Qe^{-Bx})^{1/\nu}} \tag{3.5}$$

with parameters $x = v - v_f$, $A = \varepsilon_0$, $K = \varepsilon_\infty$, $C = 1$, Q =1, $-B = k$ and $v = 1$.

Equation (3.4), constructed in this way, satisfies all the required properties:

- **Asymptotic behavior**:
  - For $v \to -\infty$, this results in: $\varepsilon(v) \to \varepsilon_0$.
  - For $v \to +\infty$, this results in: $\varepsilon(v) \to \varepsilon_\infty$.

- **Monotonicity**:
  - The function is monotonically decreasing for every $k > 0$:

$$\frac{d\varepsilon}{dv} = -\frac{(\varepsilon_0 - \varepsilon_\infty) \cdot k \cdot e^{k(v-v_f)}}{\left(1 + e^{k(v-v_f)}\right)^2} < 0 \tag{3.6}$$

    This is consistent with the assumption that decision reliability improves with increasing production experience.

- **Point of inflection**:
  - Found at $v = v_f$, where the second derivative changes sign.

Finally, from a formal point of view, the final formulation of the learning curve is as follows:

$$\varepsilon(v) = \begin{cases} \text{undefined}, & \text{for } v < V_{start} \\ \varepsilon_\infty + \dfrac{\varepsilon_0 - \varepsilon_\infty}{1 + e^{k(v-v_f)}}, & \text{for } v \geq V_{start} \end{cases} \tag{3.7}$$

The function is defined only for $v \geq V_{start}$, consistently with the assumption that the learning-enabled decision refinement becomes operative only after a minimum production experience has been accumulated. Prior to this threshold, the system operates under a conservative deterministic regime. This formulation reflects a realistic ramp-up behavior: in the early stages, decision reliability cannot yet benefit from learning effects due to insufficient production experience. In line with properties P1–P4 defined in Section 3.2.1, the parameter V_{start} identifies the minimum data volume beyond which a measurable reduction in decision error probabilities begins to emerge. From a physical point of view, V_{start} corresponds to the starting point of the actual integration of the predictive system into the production process. From a physical standpoint, V_{start} represents the production volume threshold beyond which the learning-enabled decision refinement becomes operative within the production process. Prior to this threshold, the system operates under a conservative deterministic regime. From an analytical perspective, V_{start} introduces a horizontal shift in the logistic function, ensuring that the onset of the decreasing error region aligns with a sufficient data-accumulation regime and precedes the inflection point v_f, where the most significant variation in performance occurs. This formulation reflects realistic ramp-up dynamics and ensures compliance with monotonicity (P3) and asymptotic consistency (P2). The resulting GLF provides a compact analytical representation of the progressive reduction in decision error probabilities (α and β) as a function of accumulated production experience, consistently satisfying properties P1–P4.

3.2.3. Statistical Reliability Conditions for Error Estimation

In order to define the V_{start} parameter, it was necessary to establish a minimum statistical reliability condition for estimating the rate of false positives (α) and false negatives (β). Error rates α and β are not deterministic values, but represent population proportions, i.e., unknown probabilities of classification error of the control system (e.g. an algorithm, a statistical control, an operator, etc.) called upon to make a binary decision. Such situations can arise in numerous industrial contexts, for example when the actual condition of a part determines whether additional action is objectively necessary or not, as in the case of reworking, a complete rather than partial inspection, or the execution of an extended machining cycle. The decision whether or not to activate such action is then left to the control system, which may be subject to classification errors.

As highlighted in sampling theory ([78]), such proportions are unknown parameters that can only be estimated through sample proportions. In the statistical literature, these error rates are modeled as binomial proportions ([79]) and this approach is widely adopted in Statistical Quality Control (SQC), where defect rates are treated as binomial parameters estimated on production samples ([69]). In the present work, the same statistical treatment is extended to the classification errors α and β, interpreted as binomial portions.

In practice, however, only a finite sample of parts can be observed, therefore α and β are estimated as sample proportions: $\tilde{\alpha} = i_\alpha/n_\alpha$ and $\tilde{\beta} = i_\beta/n_\beta$, where i is the number of observed errors and n is the total number of relevant cases, i.e. the subset of situations in which each error could actually occur. More precisely:

- for α, the denominator includes all cases in which the action was not required,
- for β, the denominator includes all cases in which the action was required.

The general definition of classification errors can be introduced from the following formulation:

$$\alpha = P(D = 1 \mid C = 0) = \frac{FP}{FP + TN}, \quad \beta = P(D = 0 \mid C = 1) = \frac{FN}{FN + TP} \tag{3.8}$$

Where C denotes the ground truth condition (with $C = 1$ for positive cases, i.e. situations in which the additional action was truly required, and $C = 0$ for negative cases, i.e. situations in which the action was not required), and D represents the system decision (with $D = 1$ when the action is activated and $D = 0$ when it is not). The probability of positive cases is denoted by:

$$r = P(C = 1) \tag{3.9}$$

so that, in a sample of N items, approximately $(1-r)N$ negative cases and rN positive cases are expected to be available for the estimation of α and β, respectively.

This is equivalent to the representation of classification errors in terms of the confusion matrix:

Ground truth	**System decision: *Activate action***	**System decision: *Do not activate***
Action required	True Positive (TP)	False Negative (FN) Action not activated when required
Action not required	False Positive (FP) Action activated unnecessarily	True Negative (TN)

Table 3.2: Confusion matrix of system decisions vs. ground truth.

This follows the standard statistical treatment of binomial proportions, where the sample proportion $\tilde{p} = i/n$ is the maximum likelihood estimator of the true population parameter p ([80]). Since sample proportions are subject to sampling variability and converge to the true population values as the sample size increases ([78]), it is necessary to introduce a fixed margin of error and a statistical confidence level in order to define when estimates of α and β can be considered sufficiently reliable ([78, 81]).

In the proposed framework, these classical sample proportion estimators of α and β are recalled exclusively to define statistical reliability conditions and minimum sample size requirements, which are used to justify the introduction of the V_{start} parameter. The volume-dependent error rates $\alpha(v)$ and $\beta(v)$ employed in the model are not inferred from observed confusion matrices and are not updated during execution. Instead, they are predefined, experience-dependent parameters representing the assumed learning behavior of the AI-based inspection system. Empirical error ratios computed from observed data are therefore used solely for system-level performance assessment and consistency analysis and do not influence the learning curves themselves.

3.2.4. Sample Size Determination for α and β

The minimum sample size required to obtain a statistically reliable observation of a proportion with a given margin of error d and confidence level $1-\delta$, assuming a theoretical

infinite population N, is calculated as follows ([78]):

$$n_0 = \left\lceil \frac{t^2 pq}{d^2} \right\rceil \quad \text{with } q = 1 - p \tag{3.10}$$

where p represents the expected value of the proportion, d the desired margin of error and t the quantile of the standard normal distribution corresponding to the confidence level.

In the case of α, the calculation refers only to the population of negative cases (i.e. situations where the additional action was not required in the ground truth), whereas for β it refers only to the population of positive cases (i.e. situations where the additional action was actually required). The number of items that must be processed in order to obtain the necessary number of observations is adjusted according to the production flow, where the expected proportion of negative cases is $1 - r$ and that of positive cases is r. Here N_{pos} denotes the number of items for which the action is required and N_p the total number of produced items:

$$N_{tot}^{\alpha} = \left\lceil \frac{n_0}{1-r} \right\rceil, \quad N_{tot}^{\beta} = \left\lceil \frac{n_0}{r} \right\rceil \quad \text{with } r = \frac{N_{pos}}{N_p} \tag{3.11}$$

In other words, N_{tot}^{α} represents the total number of items that must be processed in order to obtain a sufficient sample of negative cases and estimate α with the desired margin of error, while N_{tot}^{β} represents the corresponding number of positive cases required for the estimation of β.

Subsequently, $N_{min,tot}$ is defined as the minimum information volume required for the learning curves to be considered operationally meaningful, i.e., the minimum number of produced and observed items necessary to ensure statistical reliability of the error observation process. This quantity directly defines the V_{start} parameter introduced in the predictive error model. Formally, in order to ensure both estimates:

$$N_{min,tot} = \max\left(N_{tot}^{\alpha},\, N_{tot}^{\beta}\right) \tag{3.12}$$

This logic is in line with how proportions are handled in SQC, where sample size requirements are based on both the relative frequency of the events of interest and the desired precision ([69, 82]).

3.2.5. Definition of V_{start} Parameter

Once the minimum information volume required to obtain statistically reliable estimates of classification error probabilities has been determined, the parameter V_{start} is defined as the minimum cumulative first-pass inspection volume required for the activation of the learning surrogate. In particular, V_{start} is set equal to the minimum number of first-pass inspected items necessary to ensure sufficient statistical reliability for both Type I and Type II error estimation, as defined in Section 3.2.4.

$$V_{start} = N_{\text{min,tot}} \tag{3.13}$$

This definition reflects the assumption that no meaningful learning behavior can be represented before a sufficient amount of production experience has been accumulated. Prior to V_{start}, the system operates under a conservative deterministic regime in which learning-based stochastic refinement is not applied. It is worth to note that V_{start} does not correspond to the onset of steady-state operating conditions of the production process. Rather, it represents a statistical threshold ensuring that the learning surrogate operates on a sufficiently informative data regime. Process steady-state conditions and the validity of system-level performance indicators are addressed separately in Section 3.2.6 ([82]).

3.2.6. Operational Validity of System-Level Performance Indicators

In addition to the statistical sufficiency required for the activation of the learning surrogate, it is necessary to ensure that system-level performance indicators are evaluated under stable operating conditions. In discrete-event simulation, performance measures such as throughput, cycle time and work-in- process are known to be affected by transient effects during the initial warm-up phase ([83]). If not properly accounted for, these transient dynamics may bias the estimation of system-level Key Performance Indicators (KPIs) leading to misleading conclusions on production performance. To address this issue, classical warm-up removal techniques are adopted to identify the point at which the production process reaches sufficiently stable operating conditions. In particular, variants of Welch's method are employed, in which the cumulative average of the observed time series is smoothed using a moving average and the stabilization point is identified graphically ([84]). More robust quantitative approaches, such as the Marginal Standard Error Rule (MSER) and its extension MSER-5, are also considered. These methods define the warm-up length as the point that minimizes the marginal standard error of the sample

mean thus the observed KPI has become sufficiently stable to reliably represent long-run system behavior ([85, 86]).

It is important to emphasize that this operational threshold is independent of the learning activation threshold defined by V_{start}. While V_{start} ensures the statistical validity of the assumed learning curves, the present criterion guaranties the reliability of system-level performance indicators used for evaluation and comparison purposes.

The approach used in this work applies a percentage tolerance criterion to the final estimated value and is structured as follows. Let $\{X(t_i)\}_{i=1}^{N}$ be the time series generated by a simulation or taken from real data when available, where:

- t_i: the i-th instant of time (discrete).
- $X(t_i)$: the observed performance indicator at t_i.
- N: number of total observations.

The application of a moving average to this series, with a window of length ω_{MA}, produces the following result:

$$\{X(t_i)\}_{i=1}^{N} = \text{Moving Average}\big(X(t_i),\, \omega_{\text{MA}}\big) \tag{3.14}$$

The procedure is the following:

1. Compute the moving average series $X_{MA}(t_i)$ from the original data $X(t_i)$, using a window of length ω_{MA}.
2. Estimate the reference steady-state value μ_{ref} as the average of the ω last values of the moving average series:

$$\mu_{\text{ref}} = \frac{1}{\omega} \sum_{i=N-\omega+1}^{N} X_{\text{MA}}(t_i) \tag{3.15}$$

   Where $\omega \in \mathbb{N}$ is the check window for stability.

3. Starting from the beginning of the series, search for the smallest index $s \in \{1, ..., N-\omega+1\}$ such that the moving average remains within the tolerance band for the next ω values:

$$X_{\text{MA}}(t_j) \in \big[(1-\tau)\mu_{\text{ref}},\, (1+\tau)\mu_{\text{ref}}\big], \quad \forall j \in \{s, s+1, \ldots, s+\omega-1\} \tag{3.16}$$

   Where $\tau \in (0, 1)$ is the fixed tolerance.

4. The time instant t_s is considered the beginning of the steady-state period.

This procedure can be interpreted as a simplified operational variant of the MSER method and is referred to as the Tolerance Band Variant (TBV). The TBV method is general and can be applied to any performance indicator observed either in a simulation study or obtained from real production data, since its primary objective is to identify the onset of stable operating conditions, denoted by t_s. The TBV approach is particularly suitable for assessing the reliability of system-level performance indicators, as it allows the exclusion of transient effects associated with the initial warm-up phase. When the indicator under analysis is a rate-type metric (e.g., throughput, processing rate, transactions per unit time), the stabilization time t_s can be interpreted as the point from which the observed rate provides a reliable representation of long-run system behavior. In such cases, cumulative measures derived from the stabilized rate can be meaningfully interpreted as indicators of accumulated production or system performance. For non rate-type indicators (e.g., quality ratios, average cycle time), the TBV procedure can still be applied to determine the stabilization time t_s. However, in these cases, the method is used exclusively to ensure the statistical validity of the observed performance indicators, rather than to define a corresponding cumulative volume threshold. Accordingly, the role of the TBV method in the proposed framework is limited to the identification of a reliable observation window for system-level KPIs, independent of the learning activation threshold defined by V_{start}.

Below a brief explanation of how it has been possible to move from the MSER Rule to this simplified approach. In practice, a standard technique for reducing initialisation bias is the use of truncation rules. Stated formally, given the sample $\{Y_i : i = 1, 2, \ldots, n\}$, which in the context of this work corresponds to the discrete time series of throughput observations, truncation removes the first $d \ll n$ observations from the series and compute the truncated mean as:

$$\overline{Y}_{n,d} = \frac{1}{n-d} \sum_{i=d+1}^{n} Y_i \tag{3.17}$$

Subsequently, the aim consists in identifying the optimal truncation d^*, such as to maintain the greatest possible number of original observations while ensuring a sufficiently accurate and unbiased estimate of the steady state mean.

For this purpose, the MSER defines d^* as the truncation that minimises the marginal SE of the sample mean:

$$d^* = \arg\min_{d} \left(\frac{1}{n-d} \sum_{i=d+1}^{n} \left(Y_i - \overline{Y}_{n,d} \right)^2 \right)^{1/2} \cdot \frac{z_{1-\delta/2}}{\sqrt{n-d}} \tag{3.18}$$

Where Y_i are the observations of the time series, $\overline{Y}_{n,d}$ is the truncated mean, the term in brackets represents the standard deviation s_d of the residual sequence and $\frac{z_{1-\delta/2}}{\sqrt{n-d}}$ defines the half-width of the confidence interval at level $(1-\alpha)$ ([85, 86].

This thesis proposes a simplification consistent with the logic of the MSER: instead of explicitly searching for the value of d^* that minimises the marginal SE, it is required that the truncated mean $\overline{Y}_{n,d}$ remains within the confidence interval of the mean at steady state μ_{ref}. This is equivalent to imposing the following condition:

$$\left|\overline{Y}_{n,d} - \mu_{\text{ref}}\right| \leq z_{1-\delta/2}\, SE_d, \quad \text{with } SE_d = \frac{s_d}{\sqrt{n-d}} \tag{3.19}$$

The relative form is obtained dividing by μ_{ref}:

$$\frac{\left|\overline{Y}_{n,d} - \mu_{\text{ref}}\right|}{\mu_{\text{ref}}} \leq \tau_d, \quad \text{with } \tau_d = \frac{z_{1-\delta/2}\, SE_d}{\mu_{\text{ref}}} \tag{3.20}$$

Equation (3.20) emphasizes the percentage deviation of the truncated average from the steady-state value μ_{ref}.

In this work, μ_{ref} does not represent a parameter known a priori, but rather a value estimated a posteriori. Specifically, it is calculated as the tail average of the last ω values of the series of moving averages $X_{MA}(t)$. In this way μ_{ref} acts as a proxy for the steady-state value of the performance indicator, providing an external and stable reference with which to compare the fluctuations of the current moving average. This choice is consistent with the logic of MSER: while in the original approach stability is assessed only with respect to the truncated mean $\overline{Y}_{n,d}$, here an explicit reference is introduced that allows the stability condition to be reformulated in a more intuitive and relative way.

Moreover, to arrive to the TBV formulation, the truncated mean $\overline{Y}_{n,d}$ has been replaced with the moving average $X_{MA}(t)$ which is more suitable in simulation studies to represent the local trend of the chosen indicator. Finally, instead of dynamically calculating τ_d, a fixed threshold τ is introduced, chosen a priori as a percentage margin of stability. The final condition therefore becomes:

$$\frac{|X_{\text{MA}}(t) - \mu_{\text{ref}}|}{\mu_{\text{ref}}} \leq \tau \quad \text{for at least } W \text{ consecutive observations} \tag{3.21}$$

The TBV method, instead of calculating and minimising the marginal SE of the sample mean using statistical procedures, imposes a simpler and more intuitive condition of relative stability. Namely, a percentage tolerance band is defined around the reference value

μ_{ref} and the system is considered to be stable when the moving average ($X_{MA}(t)$) remains within this interval for at least W consecutive observations.

This formulation offers some practical advantages:

- Same underlying philosophy applies: the goal remains to identify when the average becomes stable, i.e. when the contribution of the transient is negligible.
- Engineering interpretability: the parameters τ and W are easily understandable (e.g., +2% for at least 100 observations), while marginal SE and batch means require greater statistical expertise.

3.2.7. Specification of the Learning-Based Decision Error Curve

To describe the dynamics of the predictive error during the ramp-up phase, the error curve is characterized by a set of parameters that govern its trend. These parameters are consistent with the required properties of the curve (P1–P4):

- **Asymptotic error ε_0**: theoretical maximum decision error probability at the onset of the learning-enabled regime.
- **Asymptotic error ε_∞**: represents the theoretical minimum value that can be reached by the error, assumed to be positive and different from zero to reflect the structural limits of the system.
- **Inflection point volume v_f**: identifies the point of maximum variation of the curve, corresponding to $\varepsilon(v_f) = (\varepsilon_0 + \varepsilon_\infty)/2$. It is related to V_{start} and ΔV by the relationship: $v_f = V_{start} + \Delta V$.
- **Translation parameter ΔV**: indicates the horizontal distance from the statistically defined activation threshold V_{start} (minimum information volume) to the inflection point of the curve. Therefore, it governs "*when*," after the start, the model enters the phase of most rapid improvement.
- **Slope coefficient k**: governs the speed of the transition. Higher values make the descent faster and more concentrated, while lower values make it more gradual.

3.2.8. Operational implications of α and β & Definition of ε_0 and ε_∞

In the context of predictive systems, Type I errors (FP) and Type II errors (FN) do not have the same operational impact and, consequently, must be treated with differentiated parameters within Equation (3.4).

From a general perspective:

- **Type I errors**: correspond to false positives, i.e. the system activates the additional action when it was not required, leading mainly to additional costs and reduced efficiency.
- **Type II errors**: correspond to false negatives, i.e. the system does not activate the additional action when it was actually required, which may lead to more critical risks depending on the application domain (e.g. product quality, process safety or service reliability).

For this reason, in many applications it is correct to assign greater weight to Type II errors. This is justified by both a statistical and an operational argument:

- **Statistical aspect**: Type II errors are associated with the subset of cases in which an action is truly required. In many industrial contexts, and particularly in quality control, these events represent a minority compared to non-required cases, which implies that estimating β often requires larger samples than estimating α ([69, 78]).This affects, as discussed above, the definition of V_{start}.
- **Operational aspect**: reducing β rapidly is generally a priority, as false negatives in early phases may undermine the system's perceived reliability and increase overall risk.

Consequently, the two logistic curves must be parameterized separately:

$$\varepsilon_\alpha(v) = \varepsilon_{\infty,\alpha} + \frac{\varepsilon_{0,\alpha} - \varepsilon_{\infty,\alpha}}{1 + e^{k_\alpha(v-(V_{start}+\Delta V_\alpha))}}, \quad v \geq V_{start} \tag{3.22}$$

$$\varepsilon_\beta(v) = \varepsilon_{\infty,\beta} + \frac{\varepsilon_{0,\beta} - \varepsilon_{\infty,\beta}}{1 + e^{k_\beta\left(v-(V_{start}+\Delta V_\beta)\right)}}, \quad v \geq V_{start} \tag{3.23}$$

This ensures that Equations (3.22) and (3.23) reflect not only the assumed learning dynamics, but also the different industrial criticality of the two types of error. This approach is consistent with well-established practices in quality and reliability engineering ([69]).

Therefore, for both α and β it is necessary to define the asymptotes (ε_0 and ε_∞) of the error function. For each error $x \in \{\alpha, \beta\}$, the following are formally defined:

- **Upper asymptote ε_0**:

$$\varepsilon_{0,x} \triangleq \lim_{v \to -\infty} \varepsilon_x(v) \tag{3.24}$$

  It represents the initial maximum theoretical error level and due to physical constraints, $0 < \varepsilon_{\infty,x} < \varepsilon_{0,x} \leq 1$ must apply. In the adopted model, the function is defined only for $v \geq V_{start}$, therefore $\varepsilon_{0,x}$ does not coincide with the value observed at the start, which is instead given by $\varepsilon_x(V_{\text{start}}) = \varepsilon_{\infty,x} + \frac{\varepsilon_{0,x}\varepsilon_{\infty,x}}{1+e^{-k_x \Delta V_x}}$.

- **Lower asymptote ε_∞**:

$$\varepsilon_{0,x} \triangleq \lim_{v \to -\infty} \varepsilon_x(v) \tag{3.25}$$

  In this case, the limit is included in the domain and has an operational meaning, as it represents the residual error at steady state and, therefore, the structural limits of the system, which should be defined in accordance with quality industrial requirements.

These are general definitions of the quantities ε_0 and ε_∞, valid in any industrial context. However, in practice, their values must be specified separately for the two error types (α and β).

3.2.9. Definition of ΔV for α and β

To parameterize the horizontal translation of the curve, it is possible to start from Equation (3.4), from which we obtain the normalized residual error (i.e. how much error remains above the lower asymptote $\varepsilon_{\infty,x}$):

$$\eta_x(v) = \frac{\varepsilon_x(v) - \varepsilon_{\infty,x}}{\varepsilon_{0,x} - \varepsilon_{\infty,x}} = \frac{1}{1 + e^{k_x(v - v_{f,x})}}, \quad x \in \{\alpha, \beta\} \tag{3.26}$$

Subsequently, a design constraint is imposed, i.e., it is required that at volume $v_{t,x}$ (with $v_{t,x} > V_{start}$), there remains a fraction $\rho_x \in (0, 1)$ of the initial gap $\Delta\varepsilon_x = \varepsilon_{0,x} - \varepsilon_{\infty,x}$:

$$\eta_x(v_{t,x}) = \rho_x \tag{3.27}$$

Then, from Equation (3.26), it is possible to derive:

$$v_{f,x} = v_{t,x} - \frac{1}{k_x} \ln\left(\frac{1-\rho_x}{\rho_x}\right) \tag{3.28}$$

Since $v_{f,x} = V_{start} + \Delta V_x$, the final formula is obtained:

$$\Delta V_x = v_{t,x} - V_{\text{start}} - \frac{1}{k_x} \ln\left(\frac{1-\rho_x}{\rho_x}\right) \tag{3.29}$$

In the model adopted in this work, by definition, $v_{f,x}$ is always located at 50% of the improvement (from Figure 3.3, it is worth $\varepsilon_x(v_{f,x}) = (\varepsilon_{0,x} + \varepsilon_{\infty,x})/2 = \varepsilon_{\infty,x} + 1/2\ \Delta\varepsilon_x$). Therefore:

$$\rho_x = 0.5 = \frac{1}{1 + e^{k_x(v_{t,x} - v_{f,x})}} \quad \Rightarrow \quad v_{t,x} = v_{f,x} \tag{3.30}$$

From this equality, with $\rho_x = 0.5$:

$$\Delta V_x = v_{f,x} - V_{\text{start}} \tag{3.31}$$

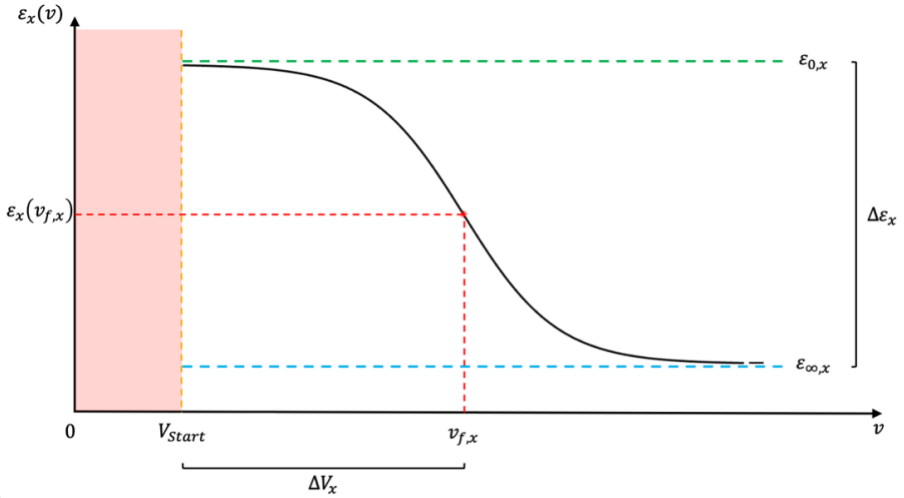

Figure 3.3: Graphical representation of the learning curve of error function $\varepsilon_x(v)$.

3.2.10. Definition of k for α and β

The parameter k $(k > 0)$ is calculated by determining the transition width W, namely the number of parts needed to reduce the error between two residual values ρ_1 and ρ_2 of

the improvement gap $\Delta\varepsilon_x$. Formally:

$$W = v(\rho_1) - v(\rho_2) \tag{3.32}$$

Starting from the definition of normalized residual error $\eta(v)$, the aim is to find the relationship between volume v and residual error $\rho = \eta(v)$. Consequently, starting from Equation (3.26), v is expressed explicitly:

$$v(\rho) = v_f + \frac{1}{k} \ln\left(\frac{1-\rho}{\rho}\right) \tag{3.33}$$

Then two residual fractions ρ_1 and ρ_2 are considered, corresponding to two volumes $v\,(\rho_1)$ and $v\,(\rho_2)$, hence the following will apply:

$$v(\rho_1) = v_f + \frac{1}{k} \ln\left(\frac{1-\rho_1}{\rho_1}\right) \tag{3.34}$$

$$v(\rho_2) = v_f + \frac{1}{k} \ln\left(\frac{1-\rho_2}{\rho_2}\right) \tag{3.35}$$

As defined by W, replacing Equation (3.34) and (3.35) in Equation (3.32), the result is:

$$k = \frac{\ln\left(\frac{1-\rho_1}{\rho_1}\right) - \ln\left(\frac{1-\rho_2}{\rho_2}\right)}{W} \tag{3.36}$$

In the proposed framework, k is assumed to be common to both curves, i.e. $k_\alpha = k_\beta = k$ (this also justifies the fact that the subscript x is not present in the previous formulas). This choice meets two main requirements:

- **Direct comparability**: by imposing the same transition speed for both curves, the differentiation between false positives and false negatives emerges solely from the translation parameter ΔV and from the initial and final levels (ε_0 and ε_∞). In this way, the curves remain homogeneous in shape and differ only in terms of "*when*" and "*how much*" the improvement occurs.
- **Interpretability**: the parameter k provides a global measure of learning speed, which can be consistently applied across error types. Smaller values of W (larger k) correspond to steeper and faster reductions, while larger W (smaller k) correspond to slower and more gradual reductions.

Therefore, in order to compute k, the transition width W should be set as design con-

straint.

3.2.11. Composite error function

To obtain a synthetic predictive performance indicator during ramp-up, it is possible to define a composite function as a weighted linear combination of the two curves defined in Equation (3.22) and (3.23). At each experience level v, the inspection system is assumed to behave as if it were characterized by error probabilities $\alpha(v)$ and $\beta(v)$. On this basis, the expected operational risk associated with false positives and false negatives can be formally defined and quantified as follows:

$$\varepsilon_{\text{comp}}(v) = c_\alpha \cdot \varepsilon_\alpha(v) + c_\beta \cdot \varepsilon_\beta(v) \tag{3.37}$$

where the coefficients c_α and c_β reflect the relative importance of the two types of error. To obtain the previous formulation, it is necessary to refer to Bayesian Decision Theory, which states that the expected risk is expressed as a weighted combination of error probabilities, with weights corresponding to misclassifications costs ([87]). The steps to obtain Equation (3.37) are listed below:

1. The expected loss or conditional risk is by definition:

$$R(\gamma_i \mid \mathbf{x}) = \sum_{j=1}^{c} L(\gamma_i, w_j)\, P(w_j \mid \mathbf{x}) \tag{3.38}$$

   Where the vector feature is denoted by $\mathbf{x}$, the set of classes by $\{w_1, \ldots, w_c\}$, the set of possible actions by $\{\gamma_1, \ldots, \gamma_a\}$, the loss function (i.e. the loss/cost incurred for taking action γ_i when the class is w_j) by $L(\gamma_i, w_j)$ and the posterior probability by $P(w_j \mid \mathbf{x})$.

2. In a simplified binary case with two classes (w_1, w_2) and two actions (γ_1, γ_2), the cost matrix can be written as:

Class / Action	w_1	w_2
Accept (γ_1)	0	λ_{FN}
Reject (γ_2)	λ_{FP}	0

Table 3.3: Loss matrix highlighting the costs of FP and FN.

   Here λ_{FP} and λ_{FN} represent the costs associated with FP and FN, respectively. Furthermore, it is assumed that the quality of the predictor depends exclusively

on the experience accumulated, thus the vector $\mathbf{x}$ is replaced by 1D variable, i.e. cumulative volume of observed data.

3. From Equation (3.38), it is obtained:

$$R(\gamma_1 \mid v) = 0 \cdot P(w_1 \mid v) + \lambda_{FN} P(w_2 \mid v) = \lambda_{FN} P(w_2 \mid v) \tag{3.39}$$

$$R(\gamma_2 \mid v) = \lambda_{FP} P(w_1 \mid v) + 0 \cdot P(w_2 \mid v) = \lambda_{FP} P(w_1 \mid v) \tag{3.40}$$

4. Using $P(w_1 \mid v)$ as α and $P(w_2 \mid v)$ as β, the composite expected loss (or cost) function can be computed as:

$$R(v) = \lambda_{FP}\, \varepsilon_\alpha(v) + \lambda_{FN}\, \varepsilon_\beta(v) \tag{3.41}$$

If the weights $\lambda_{FP} = c_\alpha$ and $\lambda_{FN} = c_\beta$ are normalized, i.e. $c_\alpha + c_\beta = 1$, the composite expected loss (or cost) can be interpreted as the weighted average error. This formulation is in line with standard performance aggregation measures in machine learning ([87, 88]).

3.3. Generality and Applicability of the Framework

The approach developed is designed to be generalizable and reusable in any context in which one wishes to represent the experience-driven evolution of the predictive capacity of an intelligent system in an abstract and controlled manner. Since the logistic function adopted does not depend on a specific machine learning algorithm but simply describes the decreasing trend of an error probability as experience increases, it can be used to emulate the expected output of any predictor, in environments where explicit training is not feasible. In this way, the proposed model serves as a modular and parametric component that can be easily integrated into decision flows or complex systems, providing a realistic and calibratable representation of the predictive maturation of a system during critical initial phases such as ramp-up.

The proposed function is compatible with the behavior observed empirically in learning curves ([89, 90]), although its explicit adoption for modeling the evolution of Type I and Type II error probabilities, $\alpha(v)$ and $\beta(v)$, within a ramp-up predictive framework has received limited formal treatment in the literature. In this sense, the present work extends existing formulations by providing a parametric framework for predictive error dynamics, bridging statistical learning theory and practical applications in industrial ramp-up

contexts.

3.4. Mapping of TO-BE Architecture in Digital Environemt

This section describes how the proposed TO-BE functional architecture could be implemented in a digital environment (e.g. simulation software). The implementation is designed to emulate an AI-based inspection system during ramp-up by combining: a deterministic decision rule (policy layer) based on acceptance thresholds and a stochastic learning component whose misclassifications probabilities evolve with cumulative production experience. The Decision Support System (DSS) is not explicitly implemented in the simulation model; instead, the simulation generates the performance indicators that a DSS would use to validate ramp-up progression.

3.4.1. Overall logic and correspondence with the functional blocks

In a possible digital instantiation of the proposed architecture, production progresses over time, while performance evaluation is performed at predefined checkpoints (e.g., batches or cumulative volume intervals). The cumulative first-pass inspection volume is represented by an experience counter v, operationally implemented as `v_eff`, which increases only when a new unit undergoes its first inspection at the quality control stage. Retesting loops do not contribute to this counter, ensuring consistency between the conceptual definition of cumulative production experience and its operational realization within the digital model.

Conceptually, the implementation follows this sequence:

1. The production system generates parts and process events.
2. For each part, a ground truth state of nature (SoN: OK/NOK) is generated.
3. If the ramp-up data sufficiency condition is met (Volume $\geq V_{start}^{op}$), an AI decision is produced:
    - a deterministic base decision (`baseDecision`) derived from tolerance thresholds,
    - plus a stochastic error injection controlled by $\alpha(v)$ and $\beta(v)$.
4. For each batch, system-level observed metrics are computed (confusion matrix and flow/performance indicators).

5. These outputs are stored as observed performance metrics that would feed supervisory DSS layer.

The following subsections illustrate the correspondence between the conceptual TO-BE modules and their operational implementation within the digital modeling environment.

3.4.2. Ground truth generation

Within the proposed architecture, each unit is associated with a ground-truth quality state, referred to as State of Nature (SoN), which represents the true condition of the part (conforming or non-conforming). This state is required to evaluate classification outcomes and to compute Type I and Type II errors. The occurrence of non-conforming parts is governed by an exogenous defect prevalence parameter `pDef`, representing the underlying process quality level and assumed independent of the learning dynamics of the inspection system.

In a possible digital instantiation of the framework, the ground truth state can be generated through a stochastic mechanism based on `pDef`, as illustrated below:

```
var pred : real := z_uniform(4, 0, 100)
if pred < pDef
    mu.SoN := "NOK"
else
    mu.SoN := "OK"
end
```

Here, `pDef` represents the defect prevalence of the process and is independent of the Al learning curve. This separation is intentional: the learning component governs decision reliability (classification errors), while the occurrence of NOK parts represents the underlying production quality level.

3.4.3. Deterministic decision rule: base decision from tolerance thresholds

Within the proposed TO-BE architecture, the policy layer generates a deterministic base decision by comparing a predicted quality indicator against predefined tolerance thresholds. Each unit is associated with a continuous prediction value, representing the output of the inspection model prior to any learning-based refinement. In a possible digital instantiation of the framework, this logic can be operationalized as follows:

```
var SimPred : real := z_normal(10, PredMean, PredStd)
@.PredictionValue := SimPred
```

Subsequently, a conservative uncertainty band is constructed around this prediction using quantiles of an assumed error distribution. The resulting interval is then evaluated against the acceptance tolerances:

```
var UpperBoundCI : real := ErrorMean + 2.576 * ErrorStd
var LowerBoundCI : real := ErrorMean - 2.576 * ErrorStd

@.UpperBoundError := SimPred + UpperBoundCI
@.LowerBoundError := SimPred + LowerBoundCI
```

If the uncertainty bounds violate the tolerance window, the additional action (Full Cycle) is activated; otherwise, the part proceeds without intervention. This rule defines the deterministic base decision (`baseDecision`):

```
if @.UpperBoundError > UpperTolerance or @.LowerBoundError < LowerTolerance
    baseDecision := true
else
    baseDecision := false
end
```

In the industrial interpretation adopted in this thesis, this threshold-based rule represents a fixed acceptance policy: if the predicted value (with uncertainty bounds) violates the tolerance window, the system activates the additional action (*Full Cycle*). Importantly, this rule is deterministic and does not learn over time. As discussed in the methodological section, residual misclassifications may persist even at high AI maturity due to the intrinsic discretizations induced by fixed thresholds on a continuous quality space.

3.4.4. AI learning dynamics: definition of α and β via GLF

The learning behavior is modeled through predefined, volume-dependent error functions $\alpha(v)$ and $\beta(v)$, implemented using generalized logistic curves. These functions represent the evolution of Type I and Type II error propensities as a function of cumulative experience v. In a possible digital instantiation of the framework, the theoretical values of $\alpha(v)$ and $\beta(v)$ can be computed at predefined experience levels and stored for subsequent use within the decision logic:

```
alpha_v := epsilon_inf_alpha + (epsilon_0_alpha - epsilon_inf_alpha)/(1 + exp(k*(v_eff - v_f_alpha)))
beta_v  := epsilon_inf_beta  + (epsilon_0_beta  - epsilon_inf_beta )/(1 + exp(k*(v_eff - v_f_beta )))

.Model.tblPerformance["Theoretical Alpha", row_tbl] := alpha_v
.Model.tblPerformance["Theoretical Beta", row_tbl] := beta_v
```

The meaning of α and β does not change; the model treats them as the same Type I/II error quantities viewed dynamically, i.e., as functions of experience rather than fixed

constants. However, in this framework $\alpha(v)$ and $\beta(v)$ represent maturity-driven stochastic error components of the AI-based inspection decision, prescribed exogenously through logistic functions. They do not coincide point-by-point with empirical system-level ratios FP/(FP+TN) and FN/(FN+TP) computed from a finite batch.

3.4.5. Instantiation at checkpoints: operational parameters $\hat{\alpha}(v(t))$ and $\hat{\beta}(v(t))$

Within the proposed architecture, the learning module defines continuous theoretical error functions $\alpha(v)$ and $\beta(v)$. During execution, these functions are instantiated at discrete evaluation checkpoints corresponding to the current cumulative first-pass inspection volume $v(t)$. The resulting quantities $\hat{\alpha}(v(t))$ and $\hat{\beta}(v(t))$ represent operational error parameters used by the stochastic refinement layer of the decision process. In a possible digital realization of the framework, the current operational values can be retrieved by querying the precomputed theoretical curves at the experience level corresponding to $v(t)$, as illustrated below:

```
r := .Model.tblPerformance.yDim
alpha_v := .Model.tblPerformance["Theoretical Alpha", r]
beta_v  := .Model.tblPerformance["Theoretical Beta",  r]
```

3.4.6. Stochastic AI error injection (AI-noise) and maturation effect

Within the proposed architecture, the stochastic refinement layer modifies the deterministic base decision through a conditional flip mechanism governed by the operational parameters $\hat{\alpha}(v(t))$ and $\hat{\beta}(v(t))$. The final AI decision is initialized as the deterministic base decision and may be probabilistically perturbed depending on the ground-truth state and the corresponding maturity-dependent error parameter.

In a possible digital instantiation, this mechanism can be operationalized as follows:

```
@.DecisionAI := baseDecision

if mu.SoN = "OK" then
    if (not baseDecision) and (z_uniform(7,0,1) < alpha_v) then
        @.DecisionAI := true
    end
else  -- NOK
    if baseDecision and (z_uniform(7,0,1) < beta_v) then
        @.DecisionAI := false
    end
end
```

This mechanism models AI maturation as follows:

- $\alpha(v)$ is implemented as the probability that the AI introduces a Type I error when the baseline decision would otherwise be correct for an OK part.
- $\beta(v)$ is implemented as the probability that the AI introduces a Type II error when the baseline decision would otherwise be correct for a NOK part.

As v increases, $\alpha(v)$ and $\beta(v)$ decrease, reducing the frequency of stochastic flips. This captures a learning effect in terms of reduced decision uncertainty, not improved continuous prediction accuracy: the distribution of `SimPred` is stationary in the current model, while learning affects only the reliability of the final decision outcome.

3.4.7. Warm-up and data sufficiency

Before enabling the learning-driven refinement of the inspection decision, the framework enforces a data sufficiency condition through the threshold V_{start}, derived from statistical reliability requirements. In operational terms, an activation threshold V_{start}^{op} regulates when the AI-based decision mechanism becomes eligible for deployment. During the initial ramp-up phase ($v < V_{start}^{op}$), the system adopts a conservative inspection regime in which the full inspection cycle is always activated. This reflects industrial practice, where automated decision support is not trusted until sufficient production evidence has been accumulated.

Once the activation threshold is reached ($v \geq V_{start}^{op}$), the deterministic tolerance-based base decision becomes operative. The stochastic refinement layer governed by $\alpha(v)$ and $\beta(v)$ is applied only if the logical switch `LearningEnabled` is set to true.

```
-- Warm-up and activation logic

if Volume < VStart_used then
    IntegratePredictionModel.Value := false
    @.DecisionAI := true   -- conservative full-cycle during warm-up

else
    -- Deterministic base decision already computed
    @.DecisionAI := baseDecision

    if LearningEnabled then
        IntegratePredictionModel.Value := true
        -- stochastic refinement governed by α(v), β(v)
        ...
    else
        IntegratePredictionModel.Value := false
        -- learning disabled: static inspection policy
    end
end
```

This separation ensures that: data sufficiency (statistical validity), operational activation

policy and maturity-driven stochastic refinement are conceptually distinct mechanisms within the model.

3.4.8. System-level observed performance: confusion matrix and batch evaluation

For each produced unit entering the quality control station for the first time, the operational layer updates the confusion matrix counters by comparing ground truth (`mu.SoN`) and decision outcome (`mu.DecisionAI`). To avoid double-counting during retesting loops, each unit contributes to the confusion matrix only once, through a dedicated first-pass flag. Under the adopted decision convention, `DecisionAI = true` corresponds to activating the action (*Full Cycle*) and is treated as the positive decision:

```
if mu.DecisionAI and (mu.SoN = "OK")
    .Model.FP += 1
elseif (not mu.DecisionAI) and (mu.SoN = "NOK")
    .Model.FN += 1
elseif mu.DecisionAI and (mu.SoN = "NOK")
    .Model.TP += 1
elseif (not mu.DecisionAI) and (mu.SoN = "OK")
    .Model.TN += 1
end
```

At batch checkpoints (i.e., every `batch_size` first-pass units), the operational layer records the confusion matrix values and updates the cumulative first-pass inspection volume v, which serves as the experience index driving the learning curves. The updated value of v is used to instantiate the operational parameters $\hat{\alpha}(v)$ and $\hat{\beta}(v)$ at the next decision stage.

In this way, system-level metrics such as FP/(FP+TN) and FN/(FN+TP) can be computed per batch as observed indicators. Due to stochastic variability, these observed ratios are expected to fluctuate around the underlying prescribed $\alpha(v)$ and $\beta(v)$ behavior and converge only in expectation (e.g., over larger batches or multiple replications with different seeds).

3.4.9. DSS layer: conceptual role in the simulation

The DSS block shown in the TO-BE architecture would not be explicitly implemented as a control module within a possible digital instantiation of the framework. A possible digital instantiation would generate performance metrics (throughput, confusion matrix and flow indicators) that a DSS would use to decide whether the ramp-up phase can be validated or must continue. The loopback arrow in the architecture diagram represents

a supervisory iteration at discrete checkpoints: if ramp-up is not validated, production continues, $v(t)$ increases and the learning curves are instantiated again at the next checkpoint. This arrow is therefore a control-flow concept, not a data feedback that updates the prescribed learning curves. Consistently with the statistical reliability condition introduced earlier, $N_{min,tot}$ is defined as the minimum information volume required for the learning curves to be considered operationally meaningful, i.e., the minimum number of produced and observed items necessary to ensure statistical reliability of the error observation process. This quantity directly defines the V_{start} parameter introduced in the predictive error model. The proposed case study provides a concrete industrial setting for the application of the methodological framework introduced in Chapter 3.

3.5. Parametric Modeling of Operator Learning Dynamics

As discussed in previous sections, there is not only progressive learning by the supporting AI-based inspection system in production ramp-up phases, but also by the human operators involved in production. In particular, in semi-manual or manual assembly processes, operators progressively acquire skills, reduce execution variability and human-induced errors and these situations are typically reflected in a gradual reduction in cycle processing times ([13]).

Extending the same formalism used for predictive error probabilities, a logistic-type curve can be formulated to describe the execution time of manual tasks during ramp-up. The adoption of a single mathematical formalism, appropriately parameterized, to describe both automated and human system learning ensures modeling consistency, parameter comparability and integration, in particular within complex digital environments. Consequently, as with the predictive error curve $\varepsilon(v)$, which is defined only for $v \geq V_{start}$, a corresponding initial threshold N_{start} may be conceptually defined for the operator's learning representation. It is assumed that, in this formulation, no observable reduction in execution time occurs prior to the conclusion of an initial training phase. During this phase, the operator undergoes training, coaching, or familiarization tests, which do not yet result in a structured reduction in cycle time.

The logistic curve can therefore be redefined as follows:

$$T(n) = \begin{cases} \text{undefined}, & \text{for } n < N_{start} \\ T_f + \dfrac{T_0 - T_f}{1 + e^{k(n-n_0)}}, & \text{for } n \geq N_{start} \end{cases} \tag{3.42}$$

Where:

- $T(n)$: average time to run operation cycle n.
- T_0: initial operator processing time (first execution).
- T_f: final steady state processing time (after learning).
- k: learning coefficient (speed of improvement).
- $n_0 = N_{start} + \Delta N$: inflection point (cycle at maximum rate of change).
    - N_{start}: threshold (in terms of number of operation cycles) that indicates the entrance in the observable learning phase.
    - ΔN: parameter in terms of number of operation cycles that determines the distance of the curve's inflection from N_{start}.

In this context, N_{start} represents the minimum number of executions required for the process to enter the observable learning phase. From a physical and engineering point of view, it can be interpreted as the end of the training phase and the beginning of the operator's actual integration into the standard production flow.

In contrast to the $\varepsilon(v)$ curve, where parameters are typically determined based on modeling considerations or data availability, the $T(n)$ curve requires parameter calibration in accordance with the actual operating context. In particular, quantities such as T_0, T_f, k, n_0 and the threshold N_{start} itself are closely linked to the empirical data of the production process in question. This suggests that, while maintaining a consistent formal structure, the model must be adapted to different application cases through historical analysis, field observations, or specific measurement campaigns. In this sense, the function $T(n)$ can be considered as a situated representation of human learning, sensitive to the type of operation, the complexity of the task and the interindividual variability of the operators.

The function $T(n)$ satisfies general properties consistent with empirical learning:

- Monotonically decreasing behavior for $T_f < T_0$.
- Continuity and derivability on $\mathbb{R}$.
- Asymptotic behavior for $n \to +\infty$.

The proposed function is particularly suitable to describe the time evolution of human learning in the ramp-up phase for the following reasons:

- It represents a process of progressive improvement, which is typical of the early operational phases.

- It allows a controlled transition between initial and ramp-up time to be modeled.
- It is monotonically decreasing, continuous and derivable, in line with empirically observed phenomena.
- The inflection point n_0 allows the determination of the phase of maximum variation, useful for training policies and benchmarking

Moreover, it is explicitly assumed that, in the case of human learning, error can never be eliminated completely, due to physiological, cognitive and environmental limitations. Therefore, the value T_f represents a realistic asymptote, which can be very low but never zero, in agreement with what has been observed in the literature on ergonomics and human error analysis ([91]) and aligned with resilience engineering perspectives, where human performance variability is seen as inevitable in real-world settings ([92]). This also reflects a fundamental principle of empirical learning curves, which exhibit convergence toward a minimum threshold of error or operating time, without ever reaching theoretical perfection ([54]). In addition, the joint use of the two explicit asymptotes T_0 and T_f, a slope parameter k and an inflection point n_0 makes this formula particularly suitable for realistic simulation modeling: it allows the transition of the operator to the state of productive maturity to be accurately described, while maintaining consistency with the formalism already adopted for algorithmic learning.

In industrial manufacturing and ergonomic modeling, logistic-type parametric structures are frequently adopted to represent the declining evolution of processing time or human error as experience increases. The literature documents the widespread use of parametrized structures with explicit asymptotes, including hyperbolic or sigmoidal forms suitable for simulation-based modeling of human performance ([93]. This provides theoretical support for adopting a generalized logistic framework for operator learning, ensuring structural consistency with established human factors modeling approaches.

The operator learning component presented above has been formalize for conceptual completeness, but is not implemented in the current study. The analysis focuses exclusively on the AI-driven learning dynamics.

3.5.1. Engineering Reformulation of the Logistics Function

The logistic curve, expressed in Equation (3.42), provides a mathematical model to describe the evolution of cycle time as a function of the operator's accumulated experience over time, during production ramp-up. However, in order to facilitate its interpretation in an industrial context, it is useful to reformulate it in more operational terms. This sec-

tion presents several engineering variants of the function, geared towards practical needs such as estimating operational efficiency, economic analysis of improvement, predicting operational maturity time and integration into discrete simulation models. These reformulations make the model more amenable to digital representation and consistent with engineering-oriented parameterization. Based on Equation (3.42), it is possible to derive alternative expressions adapted to specific modeling contexts:

1. **Normalized Operational Efficiency**
   To provide a direct representation of the operator's learning evolution, it is beneficial to reformulate the logistic function in terms of normalized operational efficiency $\eta(n)$. This is defined as an index ranging from 0 to 1, which provides a quantitative measurement of the degree of maturity/learning (in percentage form) achieved by the operator over time, depending on the number of executions n:

$$\eta(n) = \frac{T_0 - T(n)}{T_0 - T_f} = \frac{1}{1 + e^{-k(n-n_0)}} \tag{3.43}$$

   The negative sign of the exponent $-k(n - n_0)$ reflects the increasing nature of the function, as opposed to the decreasing cycle time in the original formulation. The parameter $k > 0$ retains the same physical meaning, i.e. the larger it is, the faster the learning process.

2. **Productivity Index**
   A useful alternative to represent cycle time is the use of the operational productivity index, defined as the inverse of the average execution time of the operator at cycle n:

$$PI(n) = \frac{1}{T(n)} = \left(T_f + \frac{T_0 - T_f}{1 + e^{k(n-n_0)}} \right)^{-1} \tag{3.44}$$

   This index represents a direct measure of the operator's instantaneous productivity, expressed as the number of cycles completed per unit of time, i.e., the frequency of operation execution.

3. **Computation of the number of cycles for a target time**
   A usable reformulation of the logistic function consists of the inversion of the model, in order to estimate the number of cycles n needed for an operator to reach a given target cycle time T_{target}:

$$n = n_0 + \frac{1}{k} \ln \left(\frac{T_0 - T_{\text{target}}}{T_{\text{target}} - T_f} \right) \tag{3.45}$$

This expression allows you to calculate, analytically, how many operational repetitions are necessary for the operator's average cycle time to fall below a desired value. The validity of the formula is contingent upon the value of T_{target} being within the limits of the logistic curve, i.e.: $T_f < T_{target} < T_0$. Values outside this range are not physically consistent with the model.

4. **Residual Effort or Cognitive Load**
   In addition to improvements in cycle time, operational learning can also be interpreted from an ergonomic and cognitive point of view, modeling the residual effort sustained by the operator during the execution phases. From this perspective, it is possible to derive a complementary function from the logistic efficiency curve that represents the residual fatigue or mental/physical load associated with the operation:

$$F(n) = F_0 \cdot (1 - \eta(n)) = F_0 \cdot \left(\frac{e^{-k(n-n_0)}}{1 + e^{-k(n-n_0)}} \right) \tag{3.46}$$

   Where $F(n)$ is the residual operating load at cycle n and F_0 is the initial perceived effort when the operator performs the task for the first time. Since $\eta(n)$ is defined as a normalized index, the formulation will adopt a value of F_0 within the range $[0, 1]$, depending on the perceived initial effort associated with the task. This formulation is consistent with industrial ergonomics studies showing that learning reduces the mental and motor load required to perform repetitive or complex tasks ([94]).

5. **Cumulative Time Savings**
   Another useful reformulation concerns the cumulative time savings ΔT_{cum} achieved through learning over a number of cycles. Instead of focusing on instantaneous performance or residual effort, this formulation quantifies the total gain achieved by the operator in terms of time, as an effect of increased efficiency. The function can be defined as:

$$\Delta T_{\text{cum}}(n) = n \cdot T_0 - \sum_{i=1}^{n} T(i) \tag{3.47}$$

   Where $T(i)$ represents the actual cycle time at cycle i, which decreases over time. This expression represents the fact that, if the operator had kept the initial time T_0 constant, the execution of n cycles would have required a total time equal to nT_0. However, thanks to learning, the actual times $T(i)$ are shorter, and the difference represents the total time saved.

The engineering reformulations presented in this section highlight the versatility of the logistics learning model when reinterpreted in operational and ergonomic terms. They

allow the model to adapt to different industrial contexts, bridging the gap between abstract mathematical modeling and the concrete needs of the manufacturing world, where interpretability, measurability and adaptability are key factors in supporting continuous improvement strategies.

4 | Industrial Case

This chapter introduces the Robert Bosch GmbH and the manufacturing system under study. Next, the concrete implementation of the TO-BE architecture introduced in Chapter 3 will be presented, within a simulation-based environment tailored to the Bosch HGI Assembly Line.

4.1. Industrial Context: Automotive Sector

Robert Bosch GmbH, a global leader in technology and engineering solutions and a key player in the automotive industry, is actively engaged in the research and development of hydrogen-based power systems for future mobility and zero-emission vehicles. A major focus of this effort is the development of the Hydrogen Gas Injector (HGI), a component of the Fuel Cell Electric Vehicle (FCEV) system.

This thesis examines the ramp-up phase of the Bosch HGI Assembly Line, which is characterized by initially low production volumes and high levels of process uncertainty. The proposed case study provides a concrete industrial setting for the application of the methodological framework introduced in Chapter 3. Consequently, this phase requires continuous validation and optimization procedures to ensure efficiency, accuracy, and high-quality output. In the early stages of ramp-up, knowledge about both the product and the manufacturing process is limited. This results in restricted data availability, 100% inspection requirements and reduced productivity. As production matures, however, data-driven models and digital solutions are progressively implemented to improve productivity, reduce inspection needs and refine quality control strategies. Accordingly, Bosch aims to integrate data-driven monitoring, predictive quality control and simulation-based decision-making into the HGI Assembly Line. These efforts contribute to building a more flexible, efficient and zero-defect manufacturing process.

This case study belongs to the automotive sector, which is subject to extremely stringent quality requirements due to the safety-critical nature of its products. Undetected defects can generate severe consequences, not only in economic terms (e.g., plant shut-

downs, recall campaigns, contractual penalties), but also in terms of customer safety and brand reputation. For this reason, international reference standards (such as IATF 16949, APQP, PPAP) prescribe not only compliance with specific statistical indicators, but also promote a risk-proportionate approach, which translates into conservative strategies during the early stages of production. In addition, the automotive sector is undergoing a profound transformation driven by electrification, hydrogen-based propulsion, digitalization and sustainability requirements. These dynamics impose additional challenges on manufacturers, who must cope with shorter ramp-up times, highly automated production environments and increasingly complex supply chains. This context highlights the strategic importance of adopting data-driven, conservative, and risk-proportionate approaches to ensure both operational excellence and long-term competitiveness.

This prudential logic is widely documented in industrial literature. Empirical studies in the automotive sector ([95, 96]) show that inspection systems in the ramp-up phase operate under strong asymmetry between two risks: the rate of false positives (α) is significantly higher, while the rate of false negatives (β) is minimized with the highest priority. The prudent selection of parameters in the logistic functions adopted in this work to model α and β is therefore based on this logic: a rapid and significant reduction in β compared to a more gradual reduction in α.

4.2. System Under Study

The system under analysis is the Bosch HGI Assembly Line, considered during ramp-up phase. It is a discrete production line characterized by low initial volumes, partially stabilized processes and a strong influence of both technological and human factors.

From a technological perspective, the line integrates automatic control and inspection systems designed to detect anomalies and process defects. In this initial phase, however, the limited availability of historical data and the intrinsic variability of the process lead to a high degree of uncertainty in the ability of the controls to correctly discriminate between compliant and defective cases. As a result, it is common to observe high rates of both false positives (FP) and false negatives (FN) at the system decision level, which must be monitored and managed throughout the ramp-up.

At the same time, human operators play a significant role by performing manual or semi-manual activities. For these operators, a progressive learning curve can also be observed: the time required to perform operations tends to decrease gradually as the number of completed cycles increases, until it stabilizes. This phenomenon can be represented, in principle, through logistic curves similar to those used for α and β, allowing for a

conceptually coherent integration of automated and human learning dynamics within the overall framework.

4.2.1. The product: Hydrogen Gas Injector (HGI)

The Hydrogen Gas Injector (HGI) is the main component in the hydrogen supply path of a fuel cell electric vehicle (FCEV). It is a compact and silent proportional valve that ensures both constant recirculation and continuous supply within the hydrogen circuit, according to the demand from the tank, as shown in Figure 4.1. Control is performed by the Fuel Cell Management Unit. In addition, the valve is equipped with a shutdown function that allows the medium-pressure stage to be separated from the stack. The stack, a set of fuel cells and the core of the fuel cell system, requires precise hydrogen injection to enable efficient energy conversion and overall system performance. In simple terms, the HGI is responsible for regulating the flow of hydrogen into the stack, which is fundamental for the reliability and effectiveness of the entire powertrain.

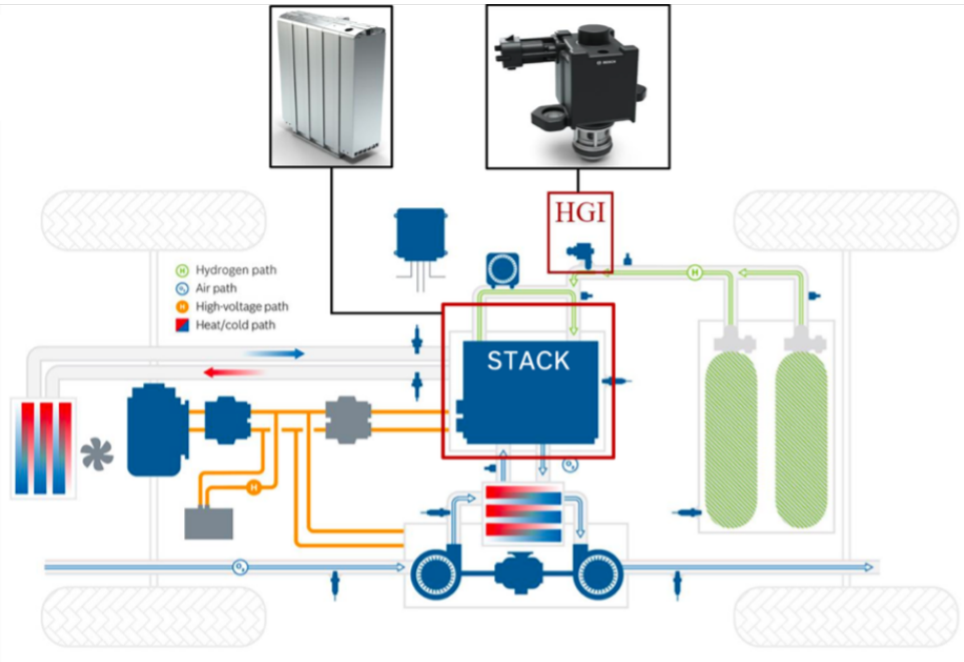

Figure 4.1: Overview of the FCEV System and HGI.

Figure 4.1 illustrates the overall architecture of a Fuel Cell Electric Vehicle. Hydrogen, stored in high-pressure tanks, is conveyed through the HGI, which regulates the amount fed into the stack. Inside the stack, hydrogen reacts with oxygen from the air to generate electricity, which is transferred to the high-voltage system to power the electric motor and on-board components.

The HGI essentially consists of internally manufactured components, ad depicted in Figure 4.2:

- **Valve Body**: the main structural component of the injector, housing the internal elements that regulate hydrogen flow. It ensures precise regulation and sealing to prevent leaks.
- **Solenoid Group**: an electromagnetic actuator that manages the opening and closing of the valve. When activated by an electrical signal, the solenoid moves a plunger, thereby allowing hydrogen to flow through the injector.
- **Nozzle**: the outlet component responsible for directing hydrogen gas into the fuel cell system.

Figure 4.2: HGI Type A components.

The valve body and nozzle are manufactured at the Homburg West plant, while the solenoid group is produced in-house at the Nuremberg plant. Collectively, these three parts are referred to as *Type A* components.

Type A components are produced internally because they represent the highest value-added elements of the product. In contrast, the remaining parts, known as *Type B* components, are sourced from external suppliers. Subsequently, all components are then assembled on the 6301 Assembly Line at Homburg West plant, resulting in the final HGI product. After assembly, the product is packaged and can be delivered directly to the customer.

The distinction between *Type A* and *Type B* components is illustrated in the technical

exploded view in Figure 4.3.

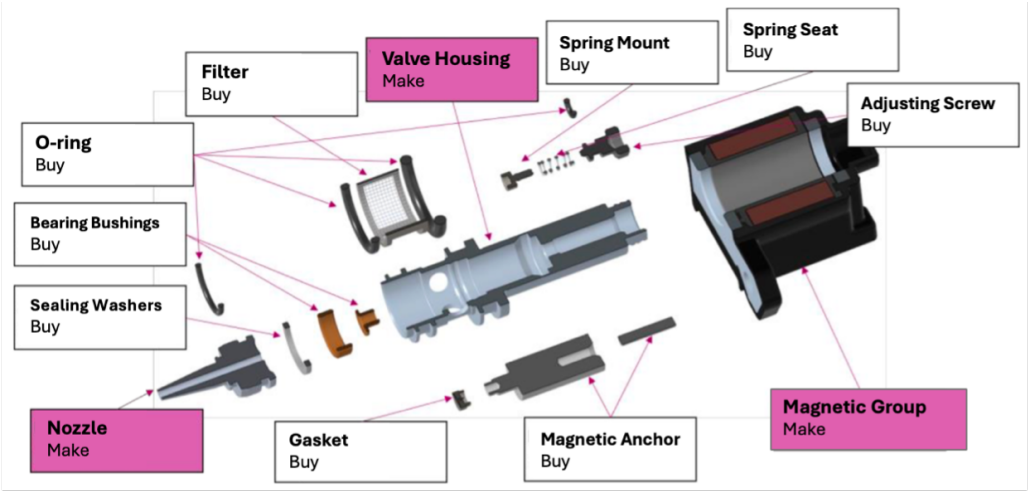

Figure 4.3: Product components of an HGI.

The HGI functions as an electrovalve actuated by a solenoid. When the solenoid coil is powered by the control unit, the magnetic field generated attracts the plunger (i.e. the movable element of the magnetic group that regulates hydrogen flow through the valve) inside the valve body. This movement opens the passage and allows hydrogen to flow through the nozzle. When the current is interrupted, a return spring returns the plunger to the closed position, interrupting the flow.

The injector is integrated into the hydrogen powertrain fuel and represents a critical component for both efficiency and sustainability. Proper injector operation not only maximizes vehicle performance in terms of power and energy efficiency but also ensures compliance with safety requirements and emissions regulations.

From a performance requirements perspective, must meet extremely stringent criteria, including:

- High accuracy and repeatability of hydrogen dosing.
- Fast response times, compatible with high engine speeds.
- Resistance to harsh operating conditions (pressure, temperature, vibrations).
- Sealing and reliability to prevent hydrogen leaks, which could pose serious safety risks.

From a manufacturing perspective, the production of HGIs involves several critical challenges:

- Tight dimensional tolerances, necessary to ensure injection accuracy.
- Use of materials and surface treatments capable of withstanding high-pressure hydrogen.
- Complex assembly processes, combining a high degree of automation with precision manual tasks.
- Advanced quality control procedures to detect micro-defects that could compromise safety or performance.

4.2.2. HGI Assembly Line Architecture and Functioning

The HGI Assembly Line is composed of several workstations, each contributing to the overall production process. In particular, the line consists of multiple stations (ST) dedicated to different phases of assembly, testing and quality validation. Figure 4.4. illustrates the architecture of the assembly line, highlighting the stations objects of analysis:

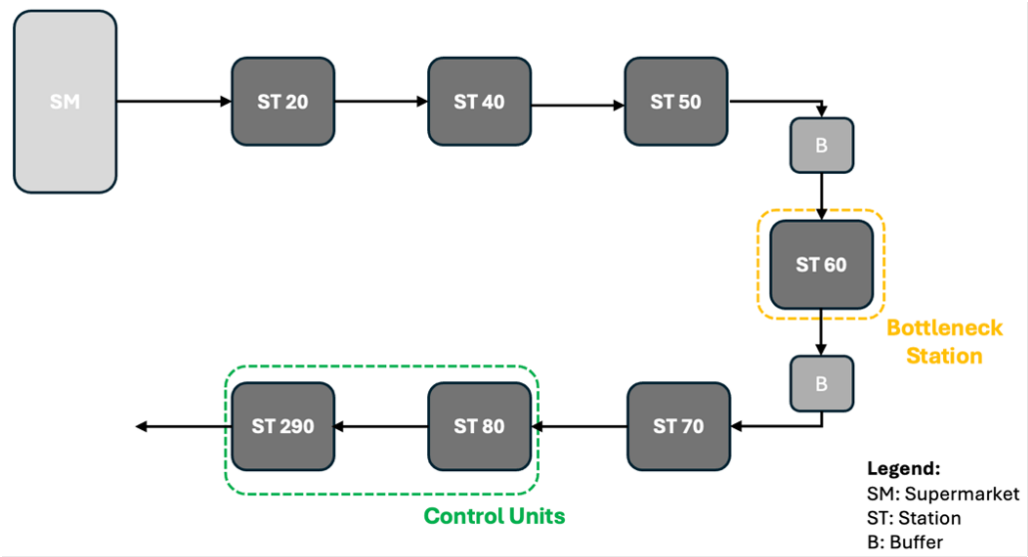

Figure 4.4: HGI 6301 Assembly Line architecture at Bosch plant in Homburg.

After introducing the product and its main performance requirements, it is necessary to describe in detail the manufacturing process through which the HGI is assembled. The process description is useful for two reasons:

- From a technical standpoint, it clarifies the steps (mechanical assembly, calibration and functional checks) that transform the original sub-components into a fully functional injector.

- From a methodological standpoint, it identifies the line's critical points, which may act as bottlenecks and are crucial in determining both quality outcomes and the learning dynamics examined in this thesis.

The bottleneck of the process is station 60 (ST60), which performs the end-of-line test (EOL test). Whereas station 290 (ST290) is dedicated exclusively to the reworking of non-conforming (NOK) parts and for this reason is not included in Table 4.1. The analysis of process times refers to series parts of type 057 and 059 assembled on the 6301 Assembly Line at Homburg West plant.

ST	**20**	**40**	**50**	**60**	**70**	**80**
$t_{0.5}[s]$	47.26	78.73	ND	246.4	ND	ND

Table 4.1: Process times (median) $t_{0.5}$ at production line stations.

The evaluation is based on the medians of the $t_{0.5}$ process times of the stations, measured on actual production data. Table 4.1 highlights that ST20 and ST40 have times of less than one and a half minutes, while ST60 requires over 4 minutes, confirming its criticality. The other stations (ST50, ST70, ST80) do not have measured times because they relate to operations that are not relevant or not applicable.

In the following, the main stations of the Bosch HGI Assembly Line are presented, with a focus on their function, the type of operations carried out and the role they play in ensuring compliance with the requirements of the automotive sector:

Station 20
Pressing and assembling of components, through which the initial mechanical body is obtained. At this stage, bearing bushings and sealing washers are inserted, laying the foundation of the injector structure.

Station 40
Installation of the nozzle and additional components. The outcome of this station is the formation of the hydraulic body (valve body + nozzle).

Station 50
Mounting of the plunger inside the valve body, in combination with its permanent magnet. This step enables the injector to function as an electrovalve actuated by the solenoid.

Station 60
The mass flow measurement carried out in this station is a crucial step for part qualifica-

tion on the production line (EOL functional test). Here, the flow rate of each injector is adjusted and validated, making ST60 a critical bottleneck in the system. The complexity of this station derives from the need for advanced process control, data-driven optimization, and anomaly detection solutions to ensure compliance with quality standards.

Station 70

At this stage, the injector is pressurized and checked for possible leaks in its various compartments (internal chambers, gaskets, joints). This ensures that the component is perfectly airtight, a condition that is essential both for safety and for maintaining high energy efficiency in the system.

Station 80

This station carries out the final quality control. The part is inspected to ensure compliance not only from a functional standpoint, but also in terms of aesthetics and adherence to regulatory requirements. This represents the last validation step before packaging and delivery to the customer.

Station 290

This station is dedicated exclusively to the reworking of non-conforming (NOK) parts. Components that fail the qualification tests in earlier stations are routed here for corrective operations. For this reason, ST290 is not included in the process time analysis but plays a crucial role in maintaining overall line efficiency and reducing scrap rates.

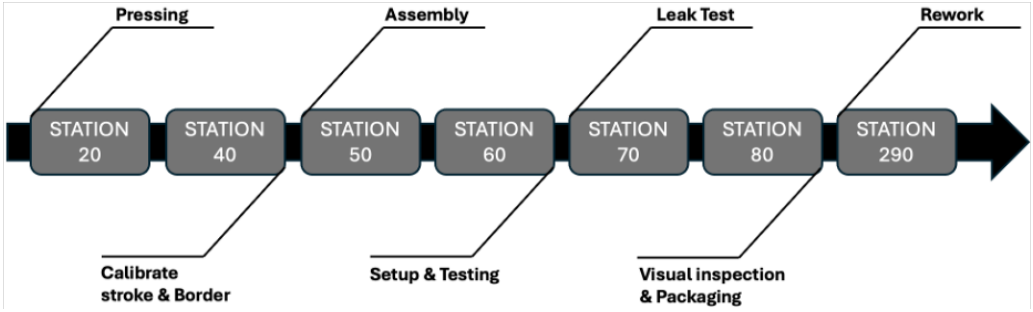

Figure 4.5: HGI assembly sequence view for stations.

4.2.3. End-of-line (EOL) Quality Test

The end-of-line (EOL) test at station 60 is composed by five process steps:

1. **Step 60-1 - Mass flow regulation**
   The initial flow calibration is performed when the injector is still incomplete: the hydraulic body (valve body + nozzle) is already assembled, but the final magnetic group has not yet been mounted. Therefore, to simulate magnetic operation, a

Master Magnetic Unit (MMU) is used (i.e., a reference coil, with magnetic core, used only for calibration, installed directly on the line, so the machine supplies power directly to the reference coil without the need for an external connector). Next, the adjusting screw is set.

2. **Step 60-2 - Adjusting screw fastening**
   The assembly is removed from the valve housing and nozzle, and the adjusting screw is locked.

3. **Step 60-3 - Serial magnetic group pressing**
   The serial magnetic group (coil + connector + magnetic core) is mechanically pressed onto the injector body.

4. **Step 60-4 - DMC code printing**
   The injector is marked with a unique identification code (DMC – Data Matrix Code), enabling full traceability of the product.

5. **Step 60-5 - Mass flow checking**
   At this stage, the final solenoid group is energized, and the injector is tested under real operating conditions (no longer using the MMU, but the serial solenoid group). Thus, the mass flow check is performed to ensure the functionality and quality of the product. To guarantee this control, eight test points have been defined, described in detail later on.

The EOL test at station 60 plays a central role in the production process. The adjustment and testing sequence is almost entirely automated, ensuring high precision and repeatability. Only a limited number of tasks are still performed manually, namely: the switch between the MMU and the standard magnetic assembly, the installation of the O-rings and the application of the DMC code. At station 60, all measurements are carried out using nitrogen (N_2) instead of hydrogen, primarily for safety and cost-efficiency. Safety is the overriding priority in this phase, since the full functionality of the product must be verified and validated before operating with hydrogen under real conditions.

For an overall view, Figure 4.6 illustrates station 60, where the end-of-line (EOL) functional test is performed. At this stage, the mass flow rate is measured using a Coriolis mass flow sensor, which provides high accuracy and reliability in flow measurement.

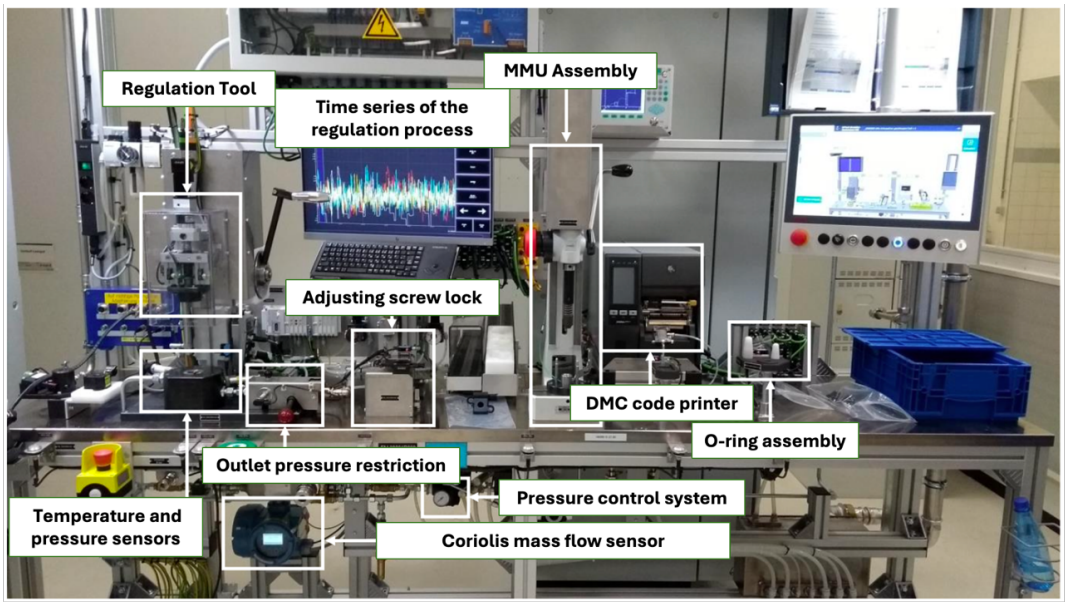

Figure 4.6: Station 60 – EOL functional test.

Station 60 includes two distinct mass flow control steps: step 60-1, performed with the MMU, and step 60-5, carried out with the serial solenoid group (Table 4.2). In the first case, it is an active adjustment: the injector is not yet equipped with its final coil and is instead calibrated using a standardized reference coil (MMU). This approach guarantees repeatable conditions and, in the event of non-compliance, enables a recalibration loop until the tolerance limits are achieved. In practice, this ensures that the hydraulic body and plunger are already properly adjusted before final closure of the injector. In the second case, the operation is a passive validation: once the injector is fully assembled, the serial magnetic group is energized, and the flow rate is measured again. The value is then compared with the re-measurement data to confirm that the settings obtained with the MMU translate into correct performance under real operating conditions.

This two-step strategy is necessary because direct adjustment with the serial coil would be unreliable: each coil is subject to small manufacturing tolerances, which could compromise repeatability. By contrast, the MMU provides standardized conditions for calibration, while the serial solenoid group serves exclusively for the final verification, accounting for the intrinsic variability of the actual component.

	Step 60-1	Step 60-5
State of the part	Hydraulic sub-assembly (Valve Body + Nozzle + Plunger)	Full injector (with serial solenoid group)
Coil type	Master Magnetic Unit	Serial solenoid group
Goal	Active mass flow rate control	Final validation of mass flow rate
Possible Actions	If out of tolerance → recalibration loop (return to 60-1-3 or 60-1-5 step)	If out of tolerance → scrap
Reference Data	Calibration data	Re-measurement data
Role in the process	Ensure that the hydraulic sub-assembly is correctly calibrated	Confirm that the complete injector meets specifications under real conditions.

Table 4.2: Step 60-1 and 60-5 comparison.

Finally, at station 70, a helium leak test is performed, in which two HGIs are tested in parallel within two separate test chambers. This procedure ensures maximum efficiency while verifying that each injector is completely airtight. Whereas station 80 represents the final stage for all conforming (OK) components. Here, a camera performs an automatic visual inspection, followed by a manual quality check, since not all characteristics can be verified automatically. Components that exhibit defects (NOK status) at any stage from station 20 to station 80 are directed to the rework station (ST290). Here, expert operators analyze the non-conforming parts and decide whether the part can be reworked or must be permanently discarded.

Below is a more detailed overview of EOL test. Figure 4.7 summarizes the sequence of processes and provides an overview of the functional testing of an HGI.

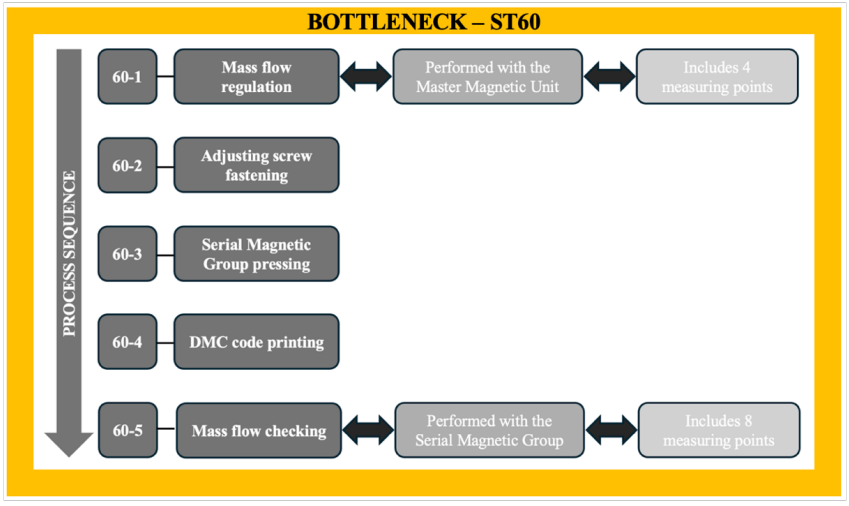

Figure 4.7: Overview of processes and data at the EOL testing station.

The mass flow rate adjustment (step 60-1) is performed in seven steps, as shown in Figure 4.8. It is worth underling that each part that reaches station 60 necessarily goes through the entire sequence of sub-steps from 60-1-1 to 60-1-7 at least once.

The adjustment process begins with magnetization, during which a constant current of $I = 1.07A$ flows throughout step 60-1, while a constant inlet pressure of $p_{in} = 12\,bar$ or $14\,bar$ (depending on the series type) is applied to the HGI. The inlet pressure p_{in} is regulated by a PID controller to ensure stability. Subsequently, for the initial angular position φ_1 of the adjustment screw, the corresponding mass flow Q_{MP1} is measured at measuring point 1 (MP1). The adjusting screw is then tightened further until reaching a new angular position φ_2, at which the corresponding mass flow Q_{MP2} is also measured.

Subsequently, it is recommended to pulse the magnetic anchor after each rotation of the adjusting screw, allowing the mass–spring system to stabilize and the internal components to align correctly. The excitation pulse is applied at a frequency of $1000\,Hz$. After the first pulse, the fine adjustment is carried out (step 60-1-5): the adjustment screw is slightly turned back, and the angular position φ_3 is fixed, corresponding to the position reached during the previous step. A second pulse is then applied at this angular position, now denoted as φ_4.

Finally, the adjustment point is verified: since the angular position after the second pulse no longer changes, the condition $\varphi_3 = \varphi_4$ is satisfied, confirming the stability of the adjustment. Finally, the mass flow rate is measured and compared against the specified tolerance limits. Two possible corrective actions are foreseen:

- If the measured flow falls below the lower limit Q_{LTL} counterclockwise correction (loosening) of the adjusting screw is permitted.
- If the measured flow exceeds the upper limit Q_{UTL}, the adjustment process must be restarted from step 60-1-3, further tightening the adjusting screw in order to reduce the mass flow rate.

The mass flow Q_{MP4} must fall within the tolerance limits specified in Table 4.3.

SERIES PART TYPE	NOMINAL VALUE [kg/h]	LTL [kg/h]	UTL [kg/h]
057	9.5	8.5	10.5
059	6.4	5.4	7.4

Table 4.3: Component specifications – Q_{MP4} mass flow in the regulation process.

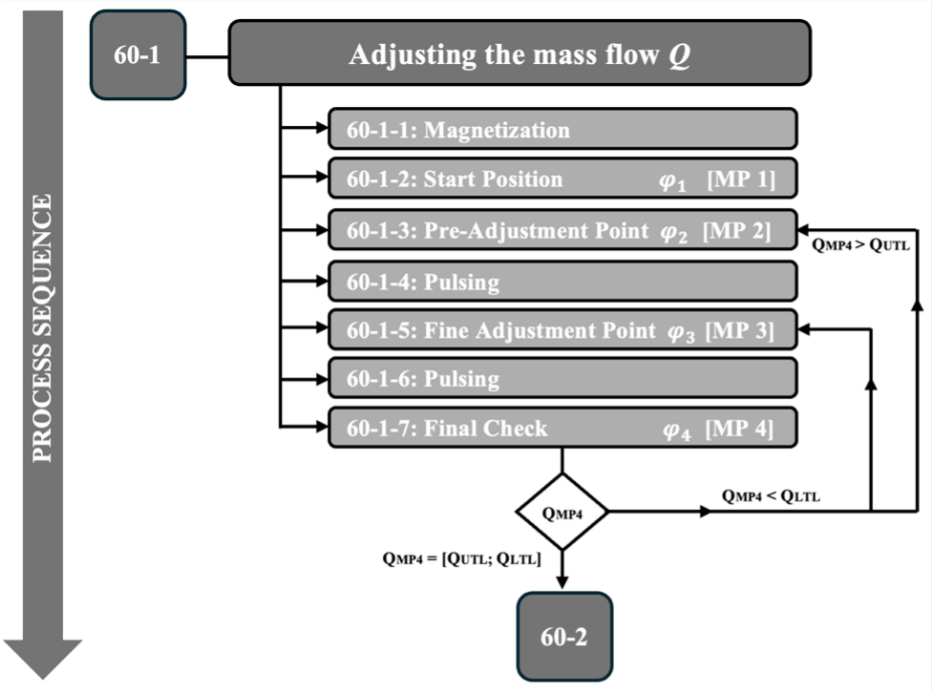

Figure 4.8: Sequence of adjustment steps (60-1).

After completion of process steps 60-2, 60-3, and 60-4, the injector is fully assembled and provided with a unique DMC code.

In the final step, 60-5 (mass flow checking), each HGI is tested at predefined measurement points to ensure that all product requirements are met, and functionality is guaranteed. At this stage, the injector is first pulsed and magnetized, after which the current is gradually increased from $0.8\,A$ to $1.8\,A$. The corresponding mass flow signal Q is recorded using a Coriolis mass flow sensor. Since mass flow is a critical parameter for product performance, Figure 4.9 illustrates its dependence on the current I, represented as a hysteresis curve.

The hysteresis curve qualitatively describes the injector's behavior during both opening and closing phases. Series types 057 and 059 exhibit different flow rates at identical current values, which is attributable to their different nozzle diameters. To normalize the results, the relative mass flow Q_{rel} is calculated according to Equation (4.1):

$$Q_{rel} = \frac{Q}{Q_{max}} = \frac{Q}{Q(I = 1.8\,A)} \tag{4.1}$$

Where Q_{rel} is the relative mass flow [%], Q is the absolute mass flow [kg/h] and Q_{max} is the maximum mass flow [kg/h].

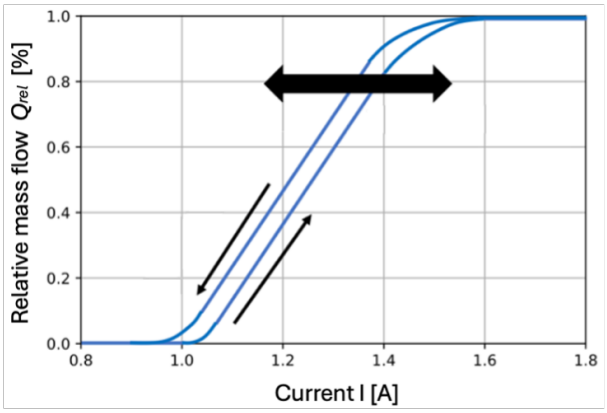

Figure 4.9: Qualitative representation of the HGI characteristics curve.

Tuning the adjusting screw does not change the shape of the curve but shifts the characteristic field horizontally.

Four main measurement variables can be distinguished (see Figure 4.10 and Table 4.4):

1. Current measurement points (PP1 and PP5)
2. Flow measurement points (PP2, PP6, and PP8)
3. Slope between current measurement points (PP3)
4. Plateau parameters (PP4 and PP7)

At PP1, with a coil current of $I = 0.9\,A$, the corresponding mass flow Q_{PP1} is measured. This value is primarily used to detect macroscopic leaks. In addition, the maximum mass flow Q_{PP5}, measured at PP5 with a current of $I = 1.8,\ A$, is considered by experts to be of particular relevance for the customer. For this reason, the regression model developed on this target variable will be briefly presented later for the sake of completeness. However, the primary focus of this work is not on the development of the AI model itself, but rather on its integration into the HGI Assembly Line simulation model.

The measurement points PP1 and PP5 cover the essential requirements of the product. For the prediction of the target value Q_{PP5}, the process parameters measured at control points PP1, PP2 and PP3 are employed as input variables (features) in the modeling. The remaining control points are excluded, since their information becomes available only after the mass flow has been measured at PP5.

From this point onward, the focus will be exclusively on series parts type 057, which will be used as the reference case for both the analysis and the simulation model of the ramp-up phase.

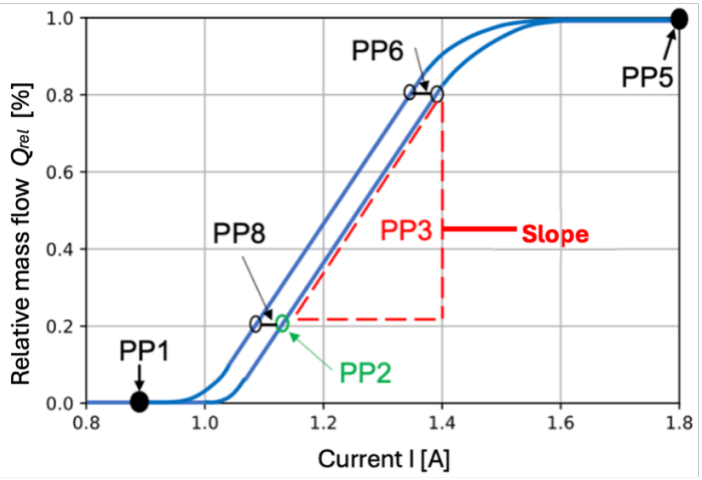

Figure 4.10: Measurement points in process step 60-5 Mass flow checking.

PP	NAMING	MEASURED VALUE	NOMINAL VALUE	TOLERANCE	FEATURE
1	Macroscopic leaks	Current measurement point 1 ($I = 0.9\,A$)	0.2 [kg/h]	0–0.4 [kg/h]	✓
2	Curve alignment on I axis	Flow measurement point 1 ($Q = 6\,\mathrm{kg/h}$)	1.08 [A]	0.98–1.18 [A]	✓
3	Slope	Between two current measurement points	79 [kg/h*A]	57–101 [kg/h*A]	✓
4	Plateau → up	Plateau parameter 1	7.5 [mA]	0–15 [mA]	*just completeness*
5	**Maximum flow**	**Current measurement point 2** ($I = 1.8\,A$)	**30.1 [kg/h]**	**28–32.2 [kg/h]**	**TARGET**
6	Hysteresis 1	Flow measurement point 1 ($Q = 9.2\,\mathrm{kg/h}$)	Hysteresis < 100 [mA]	100 [mA]	*just completeness*
7	Plateau → down	Plateau parameter 2	7.5 [mA]	0–15 [mA]	*just completeness*
8	Hysteresis 2	Flow measurement point 3 ($Q = 6\,\mathrm{kg/h}$)	Hysteresis < 100 [mA]	100 [mA]	*just completeness*

Table 4.4: Definition of measurement points and specifications (Series 057).

In summary, the main target variable for the modeling is the maximum flow measured at PP5. The prediction is based on the process parameters obtained at PP1, PP2, and PP3, which are used as input features. In parallel, safety and rejection checks are performed at

PP4 and PP7: in particular, if the plateau current exceeds 15 mA, the part is classified as non-conforming (NOK). Finally, PP6 and PP8 are included only to verify the completeness of the hysteresis behavior. Although these points are not critical for the customer, they provide useful information for internal validation and process understanding.

4.2.4. Management of Non-Conforming Parts

During each process step, the status of the part is recorded as either OK or NOK, depending on whether all measurement and test parameters fall within the specified tolerance limits. In summary, all the causes that led to the NOK status are shown in Figure 4.11.

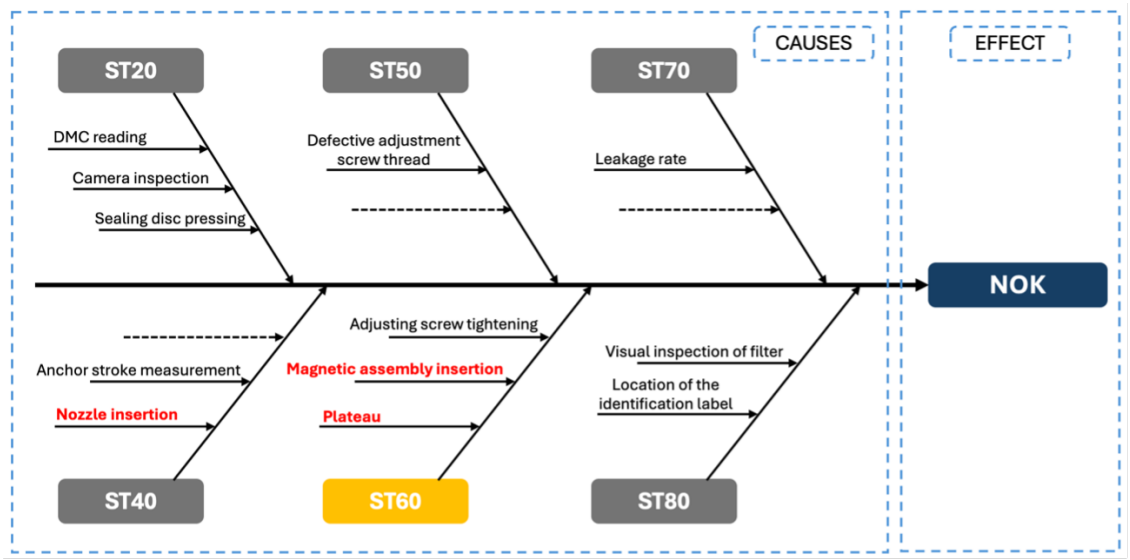

Figure 4.11: Cause-effect Analysis NOK parts (Series 057).

It is important to highlight a distinction between the different stations along the line, as illustrates in Figure 4.12. Before station 60 (ST20, ST40, ST50), NOK parts are typically sent to station 290 (ST290), where mechanical corrections may be possible. If successful, the part can re-enter the production flow and continue towards EOL functional testing. If the rework is not feasible, the part is discarded.

At station 60, the two main control tests must be taken into consideration. Step 60-1 allows recalibration loops, indeed if the measured value Q_{MP4} is out of tolerance, the system triggers re-adjustment cycles. If, after several iterations, compliance is not reached, the part is classified as NOK and routed to station 290 for further analysis. At this point, two outcomes are possible: if the rework is successful, the part is reintroduced into the production flow and retested; if the rework fails or the part remains non-compliant after repeated EOL testing, it is permanently scrapped. Instead, at step 60-5, no further

adjustment is possible, since the injector is fully assembled. If the final flow measurement Q_{PP5} or other test points are out of tolerance, the part is immediately classified as NOK. The only recovery option is retesting (up to three times). If the part still fails, it is permanently scrapped.

At station 70, the injector is pressurized and checked for leaks in different compartments (internal chambers, gaskets, joints). If the measured leakage rate exceeds the specified tolerance, the part is marked as NOK. No adjustment can be performed here; the only option is rework at Station 290, and if not recoverable, scrapping. Finally, at station 80, parts that do not meet functionality, as well as aesthetic and regulatory standards are classified as NOK and sent to station 290. If rework is not possible, the parts are definitively scrapped. After station 70 or station 80, if a part is sent to station 290 and successfully reworked, it does not re-enter the standard production flow. Instead, it is validated and released as OK directly from the output of station 290.

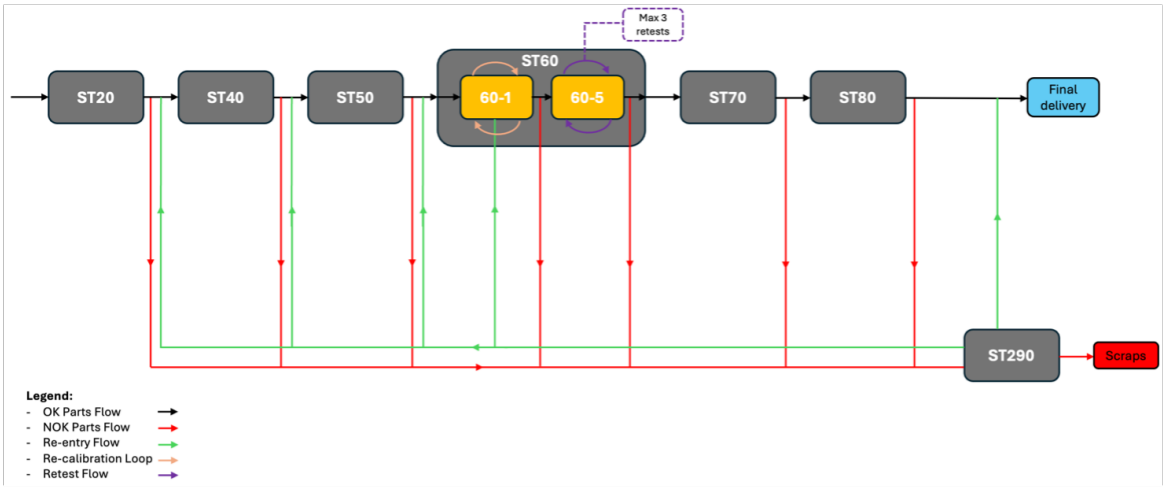

Figure 4.12: Flow of OK and NOK parts through the HGI Assembly Line.

4.2.5. Regression Model Development for Series Part Types 057

The machine learning (ML) model applied to the HGI Assembly Line and outlined below was developed by Robert Bosch GmbH. Its inclusion just provides the reader with a comprehensive overview and to support the understanding of the principles behind the *full cycle* and *reduced cycle* logic applied to the assembly line.

For the 057 series, a total of 693 injectors were analyzed with the objective of predicting the maximum mass flow rate Q_{PP5}, measured at station 60 – step 60-5. This parameter represents the key functional characteristic for the customer, with a nominal value of 30.1 kg/h and a tolerance range between 28.00 and 32.2 kg/h (Figure 4.13). Among

the analyzed samples, 671 injectors were compliant, whereas 22 were classified as non-compliant. The distribution of the compliant samples followed an approximately normal pattern, while the distribution of the NOK samples appeared irregular due to their limited number. This class imbalance required the adoption of specific data balancing techniques.

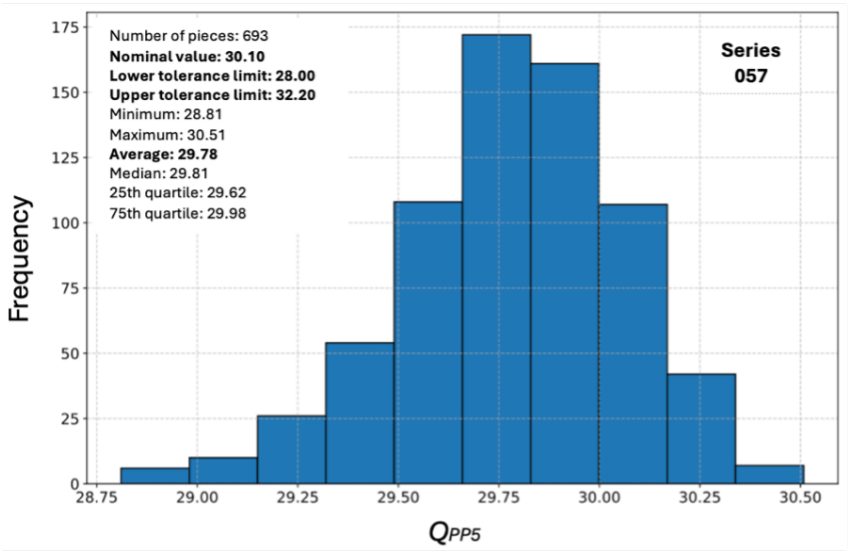

Figure 4.13: QPP5 distribution (Series 057).

The dataset was split into a training set, comprising 80% of the injectors (the oldest samples in chronological order) and a test set, comprising the remaining 20% (the most recent samples), which was used as a holdout set for final validation. The training set was further divided into five folds for cross-validation, thus reducing the risk of overfitting and enabling a more robust assessment of model performance across different data splits. To address the imbalance between OK and NOK classes, the SMOTE (Synthetic Minority Oversampling Technique) method was applied during the training phase.

The regression model was implemented using the XGBoost (Extreme Gradient Boosting) algorithm, selected for its robustness in handling complex industrial datasets with non-linear relationships. Overall, the model developed demonstrated a solid ability to predict the value of Q_{PP5} prior to its actual measurement at station 60 – step 60-5. This capability of early prediction provides the conceptual foundation for the *full cycle* and *reduced cycle* logic, enabling decisions on whether a part should undergo the complete adjustment and validation or whether it can be processed in a reduced mode. In perspective, with increasing production volumes and the accumulation of larger datasets, the predictive accuracy of the model is expected to improve further, thereby reducing the reliance on complete cycles and contributing to the mitigation of bottlenecks along the assembly line.

4.2.6. Application of the Regression Model to the HGI Assembly Line

The objective of the regression model is to anticipate the value of the target variable Q_{PP5}, namely the maximum mass flow rate measured at station 60 – step 60-5. By enabling such early prediction, the model helps to reduce cycle times at station 60, while maintaining product quality. In order to fully understand how the ML model operates, it is worth noting that the recalibration loops at station 60 – step 60-1 are independent of the *full cycle* and *reduced cycle* logic. The decision between full cycle and reduced cycle affects only the validation stage 60-5.

Therefore, if the prediction model is not active, the assembly line operates in a conventional mode in which every part follows the fullcycle configuration. At station 60 – step 60-1, each part undergoes at least one complete sequence of sub-steps (60-1-1 to 60-1-7), with additional recalibration loops triggered whenever the measured value of Q_{MP4} falls outside the specified tolerance. At step 60-5, all parts are subjected to full validation, including the eight measurement points. In this configuration, no cycle reduction is applied: while product quality is safeguarded and the risk of false negative (FN) is eliminated, the overall cycle time remains at its maximum, reinforcing the bottleneck effect at station 60.

Whereas when the prediction model is active, station 60 operates under the *full cycle* and *reduced cycle* logic. Parts are already classified at the station entrance as "*probably good*" or "*probably critical*". The regression model relies on a set of parameters measured during the early stages of the process, which are used as input features. These include:

- The results of the initial test points PP1, PP2 and PP3.
- The values of pressure, current and screw position measured at Station 60-1.
- The riveting force recorded in phase 60-2.
- The force–stroke characteristics observed during the pressing of the serial magnetic group in phase 60-3.

Also in this case, each part still passes through the complete sequence of sub-steps (60-1-1 to 60-1-7) at least once, with recalibration loops being activated whenever the measured value of Q_{MP4} falls outside the tolerance limits. But, if the prediction indicates a high probability of conformity, the part can follow the *reduced cycle* configuration, meaning that the validation may be shortened or, in some cases, skipped altogether, significantly reducing the cycle time at station 60. Conversely, if the prediction is uncertain or suggests a risk of non-conformity, the part undergoes the *full cycle*, which includes complete

validation at step 60-5 with all eight measurement points.

It should be noted, however, that the use of the prediction model introduces the possibility of misclassification. False negatives (FN) occur when a non-compliant part is incorrectly predicted as compliant, potentially allowing a defective injector to bypass full validation at step 60-5 and reach the customer. False positives (FP), on the other hand, arise when a compliant part is incorrectly predicted as non-compliant. In this case, the part is unnecessarily subjected to the full cycle, which increases the cycle time and reduces efficiency. The use of the predictive model offers advantages, in particular, it enables a significant reduction in the cycle time of station 60, thereby increasing the overall productivity of the assembly line; but also entails certain risks, namely it introduces the possibility of FP and FN. Looking ahead, with higher production volumes and the availability of larger datasets, the model can be further trained and refined, reducing the reliance on complete cycles, providing a tangible benefit in terms of alleviating bottlenecks and stabilizing the ramp-up process.

4.3. AS-IS Scenario: Critical aspects and Challenges

The analysis of the Bosch HGI Assembly Line must be framed within a critical assessment of the current scenario (AS-IS), which is characterized by the specific objectives and issues typical of the ramp-up phase. In order to fully understand the Bosch case, it is essential to begin with a description of the AS-IS state of the production system. This perspective makes it possible to highlight the baseline conditions, identify existing criticalities and define the improvement objectives to which the models developed in this thesis will be applied. This AS-IS configuration highlights a structural trade-off typical of ramp-up phases: conservative inspection strategies are required to ensure quality compliance, but they inevitably amplify false positives and operational inefficiencies when predictive models are still immature.

4.3.1. The DAT4.ZERO Project

The Bosch Case analyzed in this thesis forms part of the European DAT4.ZERO project, which involves twenty industrial and academic partners. The aim of the project is to develop innovative technologies and methodologies for zero-defect manufacturing, fostering the systematic collection and advanced analysis of process data to enhance the quality, efficiency and resilience of European production systems. Within this framework, Bosch contributes through its Hydrogen Gas Injector (HGI) Assembly Line, which serves as a representative case study for the development and validation of digital tools aimed at

monitoring and reducing defects during the ramp-up phase.

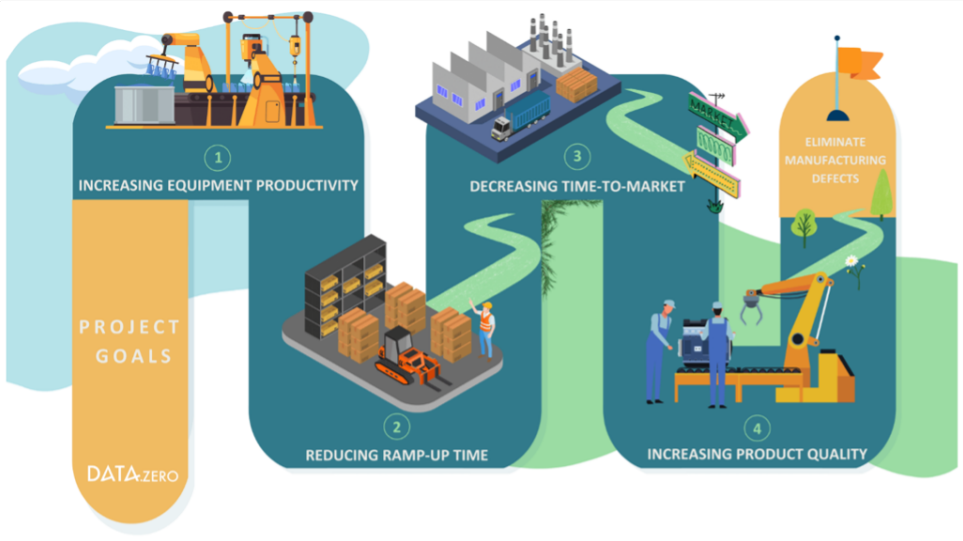

Figure 4.14: DAT4.ZERO Project Goals.

As part of the DAT4.ZERO project, Work Package 5 (WP5) is specifically focused on the development and validation of advanced tools for quality monitoring and decision support during the ramp-up phase of the HGI Assembly Line. The activities of WP5 are structured around three main pillars, which serve both as a representation of the current state of the art and as the foundation for the methodological contributions of this thesis:

1. **Anomaly Detection and Root Cause Analysis (RCA)**. This pillar aims to automatically detect anomalies in production parameters and identify the root causes of defective components.
    - Experimental Outcomes: a model has been developed that can identify whether a process was statistically abnormal during the part's production, but it requires improvement in terms of false positives and negatives.
2. **Offline prediction and visualization of results**. This module aims to create an interactive system for visualizing anomalies and root causes.
    - Experimental Outcomes: a dashboard has been implemented for the visualization of anomaly detection and root cause analysis data, providing a practical tool for supporting operators and engineers in decision-making.
3. **Simulation model for rapid line qualification**. The aim here is to develop a simulation model to represent realistic production line scenarios and support strategic decisions.

- Experimental Outcomes: a discrete-event simulation model was implemented to quantify the impact of digital systems on production. The model was parameterized using real line data and a set of ramp-up scenarios (such as different levels of process automation and worker allocation strategies) were defined to be tested and quantified in the simulation. Some uncertainties remain around rework cycles and the interpretation of the results of some scenarios. Moreover, the simulation was run both with and without the influence of the decisions made by the prediction model. The validation results show that incorporating the prediction model increases the production of good parts within the same simulation period, and therefore further enhancing productivity.

The AS-IS scenario highlights the tension between the strategic ambition of "zero defects" and the operational difficulties typical of ramp-up phase. It is within this context that the model developed in this thesis is positioned: the logistical parameterization of errors α and β, provides a tool that aligns with Bosch's trajectory, linking concrete project initiatives (WP5) with the company's long-term vision of defect-free production.

4.3.2. Industrial challenges in the Ramp-up Phase

During the ramp-up phase, a number of typical critical issues emerge, which can be summarized - with reference to Section 1.1 - as follows:

- Lack of consolidated historical data and limited initial knowledge about the product and the process.
- Process instability and high variability, due to machine running-in, suboptimal calibration and the gradual learning process of operators.
- Initially high scrap rates, resulting in additional costs and reduced productivity.

These challenges are also evident in the Bosch context, where the initial phase of the HGI Assembly Line is characterized by similar phenomena. In addition to these aspects, widely documented in the industrial literature on ramp-up, there is also a marked asymmetry between α and β risks: the reduction of β is considered the highest priority, even at the cost of a temporary increase in α.

To address this scenario, a prudent approach has been adopted, consistent with the practices of the automotive sector, articulated across three main levels:

1. **Conservative configuration of control systems**: acceptance thresholds are initially set in a restrictive manner to guarantee maximum customer protection,

thereby minimizing the probability of undetected defects.

2. **Progressive calibration and validation**: as illustrated in the HGI line stations, specific calibration steps (e.g., flow adjustment, solenoid validation) are implemented to gradually reduce the initial uncertainty.

3. **Support through predictive and simulation models**: the modeling of logistic curves for α and β enables the estimation of error trends and the prediction of the production volumes required to reach acceptable levels of statistical reliability. This constitutes a fundamental tool for guiding operational decisions during ramp-up, striking a balance between the zero-defect objective and economic as well as production sustainability.

In order to address these challenges and implement the proposed strategy, mere observation of the physical system is not sufficient. For this reason, a dedicated simulation-based environment was developed, enabling the systematic analysis of ramp-up dynamics and the evaluation of alternative scenarios.

4.4. Simulation-Based Environment for Ramp-up Performance Assessment

To analyze the ramp-up phase, a digital model was adopted, capable of reproducing the main process dynamics in a controlled and systematic manner. This approach makes it possible to overcome the limitations associated with the scarcity of data in the early start-up stages and to explore alternative scenarios. The model was implemented using Tecnomatix Plant Simulation, a discrete event simulation (DES) software widely adopted in industry. This tool enables the representation of process variability and the assessment of system performance through key performance indicators (KPIs).

4.4.1. HGI Assembly Line Simulation Model

The HGI Assembly Line Simulation Model was implemented in Tecnomatix Plant Simulation as a discrete-event system reproducing both the physical structure and the decision logic of the real production line. The model represents the main assembly and inspection stations, material flows, worker allocation and rework mechanisms, while embedding the analytical components introduced in Chapter 3.

Figure 4.15 provides an overview of the simulation environment, highlighting how the physical layout of the line is coupled with control logic, inspection policies and perfor-

mance monitoring elements. In particular, the simulation integrates the predictive inspection model at station *S60*, where continuous quality predictions, confidence intervals and decision thresholds are evaluated to determine whether additional actions are required.

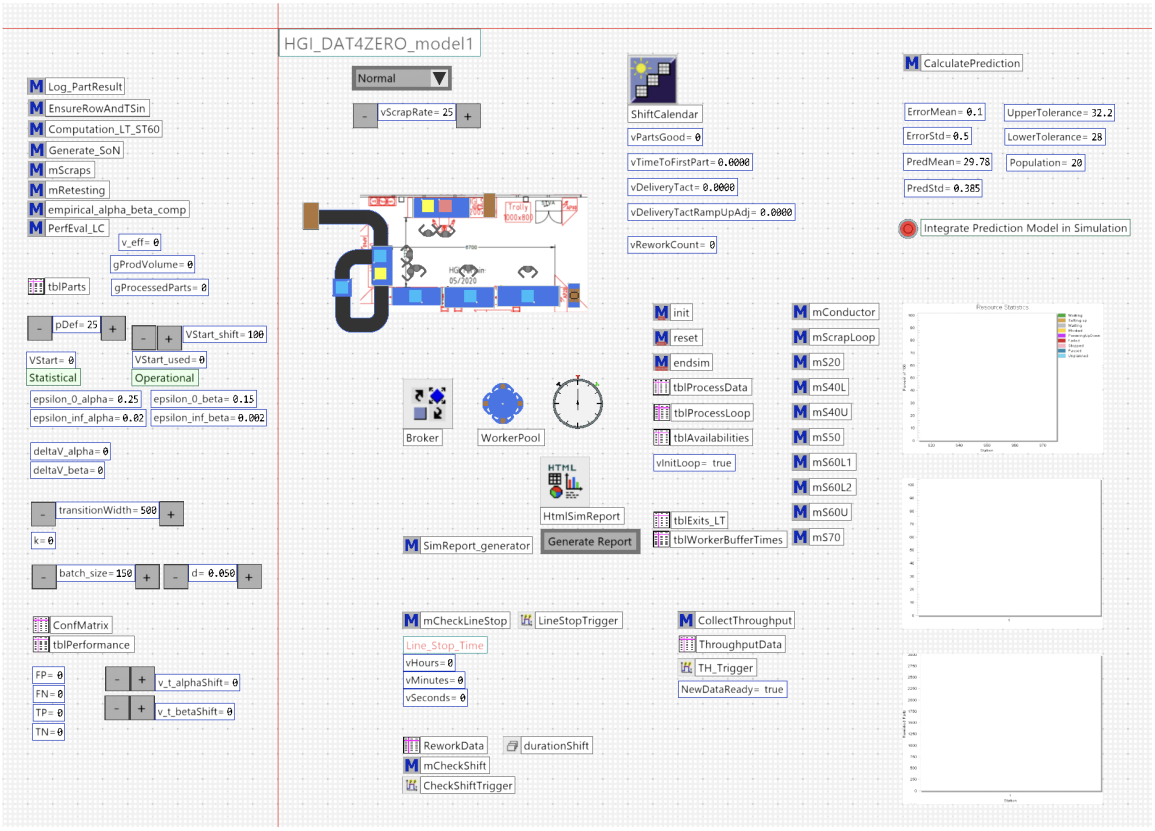

Figure 4.15: Overview of the Tecnomatix model of the HGI Assembly Line.

The use of Tecnomatix *methods* allows the separation between the structural elements of the line and the logical rules governing its behavior. This design choice enables the direct implementation of the proposed learning curves for $\alpha(v)$and $\beta(v)$, while preserving the realism of industrial process dynamics such as rework loops, resource constraints and stochastic processing times. $\varepsilon_{\infty,\alpha}$ represents the minimum conditional probability that a conforming part is incorrectly classified as non-conforming once the system has reached maturity. Specifically, for β the objective is to determine how much production experience is required for the system to reach a sufficiently reliable detection behavior, i.e., a substantial reduction of false negatives, which operationally corresponds to a high recall level.

In order to reproduce the behavioral aspects of the line, the model makes extensive use of the *methods*. These act as programmable routines that define how each object present in the model frame operates during the simulation. For instance, they specify the sequence

of operations at a station, instruct workers on when and where to move and manage events such as waiting times or resource allocation. In addition, specific methods were implemented to encode the mathematical logic of the generalized logistic function, allowing the simulation to track the decreasing trend of α and β errors as the data volume increases. In this way, methods serve as the link between the physical structure of the simulated line (stations, workers, material flow) and the logical/mathematical rules that govern its behavior.

With regard to part routing, the model shows that parts follow a structured flow but may also encounter inefficiencies. Each part undergoes multiple assembly and verification steps across different stations. The number of entries and exits recorded at each station suggests that, while the process is sequential, discrepancies between entries and exits (as computed by the simulation software) reveal the presence of rework loops. These are triggered when parts fail quality control, requiring reprocessing. This logic reflects the task performed by ST290 on the actual assembly line. For the sake of modeling simplicity, ST80 is likewise not explicitly represented in the simulation model.

In the simulation model, the so-called *Prediction Model* is implemented, which plays a crucial role in optimizing the production process. On the real assembly line, this corresponds to the ML model used for the prediction of mass flow rate Q_{PP5} at station 60 – step 60-5. Since any measurement process is subject to error, the model quantifies uncertainty through the computation of a confidence interval (CI). A higher CI (e.g., 99%) produces a wider error range, making the system more conservative, namely adopting a prudent strategy that prioritizes quality over control costs. In this case, the risk of Type II Error (FN) decreases, reducing the probability of passing defective products, but the likelihood of Type I Error (FP) increases, meaning more conforming parts are incorrectly subjected to further checks. Conversely, a lower CI (e.g., 95%) narrows the uncertainty range, thereby reducing false positives but at the expense of increasing the risk of false negatives, which raises the probability of defective products moving on to the next step.

The final predicted flow range is then calculated by adding and subtracting the upper and lower error bounds from the predicted value, resulting in an interval that represents the range within which the actual flow value is expected to lie. If this interval exceeds the predefined tolerance limits, the system flags the product for further testing.

To further expand the analytical potential of the model, it was integrated with MATLAB, enabling the automation of simulation experiments, the statistical processing of results and the efficient generation of alternative scenarios. Specifically, MATLAB is used to iteratively control the simulation model, collect the KPIs produced and analyze them

according to a predefined experimental design.

The digital environment thus created represents the central tool of this thesis. It enables the testing of different line configuration strategies, the evaluation of the impact of the progressive reduction of inspection errors (α and β) and the analysis of performance evolution during the ramp-up phase.

4.5. Application of the Methodological Model to the Bosch Case

A general framework for modeling learning curves during the ramp-up phase was introduced through the methodology discussed in Chapter 3. The proposed approach satisfies the fundamental constraints of monotonicity, asymptotic behavior and engineering interpretability, while at the same time allowing for flexible modeling of the progressive reduction of Type I (α) and Type II (β) errors as a function of the cumulative volume of observed data. This theoretical model offers the advantage of being replicable and context independent. The curves are not tied to any specific machine learning algorithm; rather, they provide a parametric description of the expected improvement of a predictive system or production process as accumulated experience increases.

Building on this abstract framework, the model is applied to the Bosch HGI Assembly Line case. In this industrial context, the primary requirement is not only to simulate the statistical evolution of errors, but also to calibrate the parameters of the logistic model in accordance with:

- The operational constraints of the automotive sector, which demand a prudent approach aimed at minimizing β.
- The quantitative benchmarks reported in the literature and established industrial practices (e.g., initial scrap rates, PPM targets, etc.).
- The specific ramp-up management logic, in which α and β entail different economic and risk implications.

The transition from the general methodology to the Bosch Case is achieved through the specific parameterization of the curves. In particular, the asymptotes, learning speed and inflection points are defined on the basis of realistic and documented values from the automotive sector and are interpreted in light of the operational priorities of the Bosch Assembly Line.

4.5.1. Simulated Accuracy Curve and Composite Error Function Parameters Definition

The α and β error logistic curves were parameterized in order to represent the evolution of the classification system's decision-making process during the ramp-up phase of the Bosch Assembly Line. The choice of a logistic model stems from its ability to describe gradual learning and saturation processes, in which performance progressively improves until reaching a steady state. This formulation is particularly suitable for modeling the reduction of operational errors in industrial systems characterized by asymptotic evolution of quality.

The following key parameters were defined for each of the two error classes (α and β). Initially, the limit values, i.e., the asymptotes of the curve, were set. In this work, they are exemplified with reference to typical values and quality metrics in the automotive sector. Predictive performances can be interpreted in terms of producer's risk (α) and consumer's risk (β), following the classical distinction introduced in SQC ([69]). In the automotive industry this terminology is particularly relevant, since supplier quality metrics are explicitly framed in terms of these two risks. Accordingly, the parameters used in this work are defined separately for each error type, as follows. For α:

- $\varepsilon_{0,\alpha}$: in the absence of historical data, its value must be estimated based on theoretical and operational considerations. In the automotive industry, during the ramp-up phase, it is common practice to adopt a prudential approach, configuring control systems conservatively to minimize false negatives. This strategy inevitably leads to a relatively high level of false positives. Recent studies confirm this approach: [95], analyzing an automotive brake cylinder production process in the automotive sector, shows how the adoption of conservative limits in the initial stages actually leads to a marked increase in false positives, justified by the need to reduce false negatives. While [95] does not provide explicit percentages, further evidence comes from empirical studies in automotive diagnostics. In particular a recent work on engine fault detection reports a false positive rate of 4.33% and a false negative rate of 24.83%, under mature system conditions ([96]). Although these figures refer to diagnostic systems rather than production controls, they highlight the asymmetry between α and β and provide a concrete benchmark. This supports the assumption that, in early ramp-up when the system is immature, considerably higher values (e.g. 20–30%) for α are plausible. Based on this evidence, and adopting a conservative profile for the automotive context, this study assumes a reference value equal to:

$$\varepsilon_{0,\alpha} = 25\%$$

- $\varepsilon_{\infty,\alpha}$: represents the minimum conditional probability that a conforming part is incorrectly classified as non-conforming once the system has reached maturity. Even in mature automotive processes, a residual fraction of false positives is unavoidable due to process variability and measurement uncertainty; in fact, quality planning frameworks (e.g., APQP, FMEA) and the SQC literature ([69]) emphasize that the producer's risk (α) cannot be reduced to zero. Based on empirical evidence from a Lean Six Sigma case study in automotive component manufacturing, where scrap generation was reduced to an average of approximately 4.9% after process stabilization ([97]), it can be inferred that in typical automotive contexts, where conforming parts account for the vast majority of production, the residual false positive rate must be significantly lower than the overall scrap rate, which also includes defective parts correctly identified and rejected. This suggests that $\varepsilon_{\infty,\alpha}$ should be modeled as a fraction of the total scrap rate and therefore set lower than the observed KPI values at regime. Based on these insights, this study conservatively assumes:

$$\varepsilon_{\infty,\alpha} = 2\%$$

For β:

- $\varepsilon_{0,\beta}$: consistently with the discussion provided for $\varepsilon_{0,\alpha}$, the definition of $\varepsilon_{0,\beta}$ in the absence of historical data also relies on theoretical reasoning and practical insights. According to the literature, β is generally regarded as more critical than α, as it directly impacts product safety and perceived quality ([69]). Consequently, β is often treated as a key parameter in quality control. As already highlighted by [95] in the context of $\varepsilon_{0,\alpha}$, the use of conservative acceptance limits tends to increase false positives but is justified by the need to reduce false negatives, which are more severe from an operational point of view. To support this view, additional empirical evidence from automotive diagnostics ([96]), quantitatively demonstrates the asymmetric impact of β compared to α, particularly in early-stage systems. Based on these considerations, the present study assumes a initial reference value of:

$$\varepsilon_{0,\beta} = 15\%$$

- $\varepsilon_{\infty,\beta}$: represents the residual probability that a defective part escapes detection under mature operating conditions. Operationally, this is the escape rate, which is tightly constrained by Original Equipment Manufacturer (OEM) contracts and expressed in terms of Parts Per Million (PPM) delivered to the customer. Op-

erationally, the mature-condition escape rate is constrained by customer/supplier quality programs. Automotive supplier manuals and agreements commonly frame the objective as "zero defects/zero PPM" and adopt escalation/monitoring thresholds in the order of a few tens of PPM ([98]). Accordingly, this study adopts a conservative asymptotic escape-rate assumption of 20 PPM[1], i.e.:

$$\varepsilon_{\infty,\beta} = 20\,PPM \approx 0.002\%$$

Subsequently, to compute $N_{min,tot}$, so the minimum information volume required for the learning curve to have operational validity, i.e. to ensure that the assumed error dynamics are applied only beyond a statistically meaningful observation volume, the following parameters of Equation (3.12) were chosen based on industrial and statistical criteria:

- $t = 1.96$
  The choice of a confidence level of 95% reflects established practice in statistics applied to quality control. Although the reference standards for the automotive sector (PPAP, IATF 16949) do not specify a numerical value, they require that processes be statistically proven to be stable and capable, with a level of rigor proportionate to the risk. In SQC literature, 95% is the most commonly adopted value as a compromise between rigor and sampling sustainability ([69]).

- $d = 0.05$
  The margin of error was set at 5% because this level represents a good compromise between statistical reliability and manageable volumes. Specialist literature notes that values above the ±5% threshold significantly reduce confidence in estimates, while lower margins entail high sampling costs ([99]).

- $p = 0.5 \;\Rightarrow\; q = 0.5$
  In theory, the calculation of the required sample size should be based on preliminary estimates of the error proportion $\hat{p}$. However, in the absence of reliable initial information, the conservative value p=0.5 is adopted, as such value maximises the variance $p\,(1-p)$ and therefore leads to the largest required sample size. This ensures that the number of observations obtained will be sufficient to estimate the error proportions with the desired precision even in the worst case ([78, 80]).

Given the above parameters, the minimum information volume required to ensure the operational validity of the curves is: $V_{start} = 1540$ parts.

[1] In the present modeling framework, β represents a normalized conditional error probability rather than the unconditional outgoing defect rate. Therefore, the value is interpreted as a scale-consistent reference level aligned with typical automotive quality benchmarks.

Moreover, also the defect rate parameter r has to be set. In the absence of real data, the parameter should be assumed. Typically, in the automotive sector, initial scrap rates fall within the range of 20% and 25%, in the early stages of ramp-up ([13]). In accordance with this evidence, $r = 25\%$ has been adopted as a prudent scenario in this thesis. This value allows the minimum volumes required for ensuring operational validity with respect to β to be calculated in a more sustainable manner than more conservative assumptions (e.g. $r = 20\%$), while remaining consistent with the principle of risk-proportionate rigor promoted by automotive quality standards (PPAP/IATF). To verify its robustness, the analysis is accompanied by a sensitivity assessment on $r \in [20\%, 25\%]$ (Table 4.5).

r	N^{α}_{tot} [pc.]	N^{β}_{tot} [pc.]	$N_{min,tot}$ [pc.]
20%	482	1925	1925
21%	488	1834	1834
22%	494	1750	1750
23%	500	1674	1674
24%	507	1605	1605
25%	514	1540	1540

Table 4.5: Sensitivity table on r.

In order to parameterize the horizontal translation of the curve ΔV for α end β, it is worth noting that from an operational perspective, as previously discussed, quickly reducing β is a priority. This is consistent with the prudential approach adopted in the automotive industry, where β should be reduced earlier than α, because escapes are much more critical than internal scrap. This principle directly influences the definition of the ΔV parameter and is mathematically expressed as:

$$\Delta V_{\beta} < \Delta V_{\alpha}$$

In accordance with what has been stated in section Section 3.2.9, to numerically compute this parameter, a design constraint is set: $\eta_x(v_{t,x}) = \rho_x \in (0, 1)$, wit $x \in \{\alpha, \beta\}$. This means that at volume v_t, a fraction rho of the initial improvement gap $\Delta\varepsilon$ remains. In the model proposed in this thesis, $\rho = 0.5$ is imposed, therefore v_t is the volume at half improvement (50% reduction in error compared to the initial value) and it is also the inflection point of the curve ($v_{t,x} = v_{f,x}$). Depending on whether ΔV is calculated for α or β, two different objectives are pursued. Specifically, for β the objective is to determine how much production experience is required for the system to reach a sufficiently reliable detection behavior, i.e., a substantial reduction of false negatives, which operationally

corresponds to a high recall level; while for α, the algorithm makes a mistake by classifying a piece that is actually good as defective. Therefore, the objective is to characterize how much production experience is required for the frequency of false positives to decrease to an operationally acceptable level. From the calculations, it is possible to derive the following results: for α, the volume at the inflection point is $v_{t,\alpha} = 2790$ parts, with a deviation $\Delta V_\alpha = 1250$ parts from V_{start}; instead for β, $v_{t,\beta} = 2232$ parts and $\Delta V_\beta = 692$ parts. These values are derived from statistical considerations on the minimum information volume required for the corresponding error rates to reach operationally reliable levels.

The last parameter of the curve to be defined is k, i.e., the slope coefficient that governs the speed of improvement. The learning speed of the production system as a whole can reasonably be considered a global property, which depends on different factors (e.g. process stability, level of automation, etc.). It is therefore realistic to assume that this learning capacity acts uniformly on both Type I and Type II error reduction, while the operational priority of ramp-up is modeled separately through ΔV, imposing $\Delta V_\beta < \Delta V_\alpha$. To numerically compute this parameter, a design constraint should be set: $W = v(\rho_1) - v(\rho_2)$. It represents the transition width, i.e., the number of parts needed to reduce the error between two residual values ρ_1 and ρ_2 of the improvement gap $\Delta\varepsilon$. The values chosen for ρ_1 and ρ_2 are 20% and 80%, because the section of the curve between 20% and 80% is the steepest and most informative; this avoids both the initial tail (transient) and the final plateau (saturation), which would bias the definition of the slope k. Another justification is the asymmetry around the inflection point v_f, i.e., it is centered on $\rho = 0.5$ (half improvement), so the transition width W is balanced and not very sensitive to local asymmetries. For $W = 500$ pieces, $k = 0.005545$ is obtained, a value that guarantees a realistic slope, consistent with the typical stabilization times of a production shift.

Overall, the defined parameters allow two realistic logistic curves to be obtained that are consistent with industrial observations:

- The β curve decreases more rapidly and earlier than the α curve, reflecting the operational priority of minimizing cases of failed full cycles already in the early stages of ramp-up.
- The α curve, on the other hand, shows a more gradual improvement, consistent with the progressive optimization of the decision threshold and the reduction of unnecessary interventions.

Moreover, in order to apply the Composite Error function, its weights must be estimated. They represent the unit costs per event, specifically:

- λ_{FP} = average FP cost (inefficiencies in terms of time and resources).
- λ_{FN} = average FN cost (reworks and possible rejects).

The following table summarizes the operational production parameters (cycle times) and the assumed cost implications associated with potential AI model classification errors:

Error Type	Event	Cost [€/min]	Cycle time [min]	Extra cycle time [min]
α	Machine used unnecessarily	0.7	1.45	0.147
β	Reworking of the piece	0.7	1.87	1.45

Table 4.6: Cost and cycle time parameters associated with α and β.

The cycle times reported in Table 4.6 derived from the actual Bosch production configuration. In contrast, the marginal production cost of 0.7 €/min is introduced as a representative average value to enable economic comparison between error types. The same unit time cost is assumed for both α and β events, as both consume the same marginal production resources within the production system. Consequently, the asymmetry in total error cost is driven by the different additional cycle times. Since the composite error function is normalized, the absolute value of the unit cost acts as a scaling factor and does not affect the relative severity weighting.

The absolute costs (i.e., how much an error is worth in euros) C_α and C_β were calculated from the data in the table above. The calculations give the following results:

$$C_\alpha = 0.417\,\text{min} \times 0.7\,€/\text{min} = 0.30€$$

$$C_\beta = 1.45\,\text{min} \times 0.7\,€/\text{min} = 1.02€$$

Next, a severity ratio R_S is computed:

$$R_S = 1.02€/0.3€ = 3.48$$

By normalizing the weights of the composite error functions, the following is obtained:

$$c_\alpha = \frac{C_\alpha}{C_\alpha + C_\beta} = \frac{1}{1 + 3.48} = 0.22$$

$$c_\beta = \frac{C_\beta}{C_\alpha + C_\beta} = 1 - c_\alpha = 0.78$$

With the definition of this composite function, it is possible to compute the expected operating cost of a system which behaves as if it had error probabilities $\alpha(v)$ and $\beta(v)$.

Finally, a parameter that is not present in the definition of the logistic curve, but which is essential for representing the evaluation cadence of the AI system during ramp-up, is the batch size B, i.e. the data aggregation interval used to compute performance indicators. Initially, this parameter was set to 350 parts, reflecting a value coherent with the parameterization of the adopted logistic curves.

In order to verify the robustness and stability of the defined model, a sensitivity analysis was conducted on the key parameters of the learning curves and operational configuration. This analysis allows the assessment of how structural assumptions regarding AI maturity and operational policies influence system-level performance, particularly in terms of throughput, error propagation and composite operational risk $R(v)$. The results of this analysis are presented and discussed in Chapter 5, where the interaction between AI maturity and production performance is quantitatively evaluated.

5 | Results and Discussion

This chapter provides and discusses the results obtained using the simulation model developed in previous chapters; therefore the main objective is to analyze the effect of the evolution of the maturity of an AI-based inspection system modeled through the error curves $\alpha(v)$ and $\beta(v)$ on the overall performance of the production system during the ramp-up phase. A series of simulation experiments and Monte Carlo replicates are used to evaluate the effects of progressively reducing classification errors on key performance indicators such as throughput, operational stability and overall risk. The chapter also introduces a comparative analysis of various AI activation strategies, emphasizing the trade-off between early and late system integration and demonstrating how this balance is influenced by the operational parameters of the process. Finally, a sensitivity analysis is conducted to discuss the robustness of the results with respect to the main modeling assumptions, providing a critical and industrially relevant interpretation of the emerging evidence.

5.1. Simulation Experiment Design

The experimental setup adopted for analyzing the results aims to ensure the reproducibility, methodological transparency and scientific soundness of the entire study by defining the elements that make up the experimental framework within which the results will be analyzed.

The simulation experiments were designed to assess the impact of the evolution of the maturity of the AI-based inspection system (modeled through the error curves $\alpha(v)$ and $\beta(v)$) on the overall performance of the production system during the ramp-up phase. In particular, the analysis aimed to quantify the following:

1. the effect of progressively reducing classification errors on operational performance;
2. the trade-off between early and late activation of the AI system;
3. the stability of the system as the main modeling parameters vary.

Each experiment covers a production simulation horizon that is long enough to represent the entire ramp-up phase and the achievement of steady state conditions. The analysis is conducted based on cumulative first-pass inspection volume (v), consistent with the methodological assumption that AI maturity depends on production experience rather than calendar time. Furthermore, given the stochastic nature of the model (i.e.the generation of the state of nature, the variability of predictions and the probabilistic error mechanism), each experimental configuration was replicated several times using different realizations of random variables. These replications enabled us to estimate the system's average behavior and characterize its variability. The simulation setup, used for the case study, is as follows:

- Simulation time: 1344 [h]
- Operating simulation time [1]: 960 [h]
- Number of operating working shifts: 120
- Number of runs per scenario: 20
- Process times: modeled using a normal distribution, $T \sim \mathcal{N}(\mu, \sigma^2)$

The performance of the production system is evaluated using a set of structured key performance indicators, which are calculated at batch level and/or cumulatively:

- **Throughput (TH)**, as the primary measure of system productivity.
- **Classification errors (FP and FN)**, derived from the confusion matrix.
- **The weighted composite error function** ($R(v)$), that summarizes the operational risk associated with inspection errors.
- **The Area Under the Curve of composite risk** (AUC_R), defined as:

$$AUC_R = \int_{v_{start}}^{v_{end}} R(v)dv \tag{5.1}$$

  Unlike the asymptotic value R_∞ alone, the AUC_R indicator measures the overall risk accumulated during the entire ramp-up phase and therefore also takes into account the speed of improvement of the system. In industry, it can be interpreted as a measure of the cost of learning, i.e. the overall inefficiency or defects generated while the system matures. A lower value of AUC_R indicates faster and less costly learning, whereas a high value reflects a longer, more unstable or costly ramp-up.

[1] Operating time has been used in the computation of results for the throughput and the other KPIs, by filtering out time windows where the manufacturing system is not supposed to work.

These KPIs enable the simultaneous analysis of production efficiency, decision-making quality and overall risk.

The experimental design distinguishes between fixed parameters, defining the structural configuration of the system (line layout, operational logic, logistics parameters, baseline ε_0 and ε_∞) and variable parameters, explored within the experimental scenarios (V_{start}, activation policies and learning curve parameters). This separation allows isolation of decision-related effects from structural system characteristics.

ID	Section	Scenario	Purpose	Parameters varied
S00_St	5.2	Static policy	No learning benchmark	Stochastic maturity disabled
S00	5.2	Baseline	Reference benchmark	none (nominal set)
S08	5.3	Wfast	Faster learning	$W \downarrow \Rightarrow k \uparrow$
S09	5.3	Wslow	Slower learning	$W \uparrow \Rightarrow k \downarrow$
S10	5.3	vtA−	Earlier α improvement	$v_{t,\alpha}$ shift −
S11	5.3	vtA+	Later α improvement	$v_{t,\alpha}$ shift +
S12	5.3	vtB−	Earlier β improvement	$v_{t,\beta}$ shift −
S13	5.3	vtB+	Later β improvement	$v_{t,\beta}$ shift +
S01	5.4	ACT_Early	Early AI deployment	$VStart_{op} \downarrow$ (negative shift)
S03	5.4	ACT_Late	Late AI deployment	$VStart_{op} \uparrow$ (positive shift)
S00	5.4 (ref)	ACT_Nominal	Reference activation	$VStart_{op} = VStart_{stat}$

Table 5.1: Simulated scenarios.

In order to systematically evaluate the impact of AI maturity and activation policies on system performance, a structured set of simulation scenarios has been defined. Each scenario modifies one key parameter while maintaining the remaining configuration constant, allowing for controlled comparative analysis. Table 5.1 summarizes the scenarios considered in Chapter 5 and the corresponding parameter variations.

Finally, in this study, the DSS is modeled at a conceptual level rather than being implemented as an autonomous control algorithm within the simulation. The model progresses

through the entire defined production horizon while decision points are evaluated retrospectively through the analysis of system KPIs. Therefore, the logic of the DSS is analytically represented but not operationally integrated into the simulation cycle.

5.2. Baseline Scenario

The reference configuration serves as the starting point for all subsequent comparisons. This scenario represents the fundamental structure of the production system and AI integration, as defined by the parameters outlined in Chapter 4. All comparative analyses developed in subsequent sections will be evaluated in terms of improvement or deterioration with respect to this configuration. Before analyzing the dynamic effects of AI maturity under nominal conditions, a structural benchmark without AI activation is introduced to isolate the intrinsic contribution of learning dynamics. This benchmark configuration, referred to as the static inspection policy, allows the system to operate exclusively under the deterministic tolerance-based rule, without stochastic modulation through $\alpha(v)$and $\beta(v)$. By contrasting this static regime with the nominal AI-enabled scenario, the impact of maturity-driven stochastic refinement can be clearly identified.

The baseline scenario adopts the nominal parameters defined in Subsection 4.5.1. Therefore, the curves $\alpha(v)$ and $\beta(v)$ represent the expected maturity profile of the AI inspection system under nominal conditions.

As a preliminary methodological observation, the trend in cumulative first-pass inspection volume ($v(t)$) can be interpreted as the progressive accumulation of production experience during the ramp-up phase. As a function of this volume, the learning curves demonstrate a gradual decrease in error rates, exhibiting sigmoid behavior characterized by an initial phase of high variability, a transition region and a subsequent saturation towards the asymptotic value.

All subsequent results will be interpreted in relation to this reference scenario. This will allow for the objective establishment of whether an alternative configuration represents an improvement or deterioration compared to the nominal base.

5.2.1. Static Inspection Policy (No-AI Learning)

To isolate the structural behavior of the production system independently of AI maturity dynamics, a benchmark configuration with learning disabled is first analyzed. In this scenario, the stochastic learning component remains inactive throughout the simulation horizon and so no stochastic perturbation governed by $\alpha(v)$ or $\beta(v)$ is applied. The final

inspection decision therefore coincides exclusively with the deterministic tolerance-based rule introduced in Subsection 3.4.3.

Importantly, the predictive signal itself is not removed. The continuous prediction model and tolerance thresholds remain fully operational; only the maturity-driven stochastic refinement is deactivated. As a consequence, decision reliability is entirely determined by the structural separability of the predictive signal and the predefined acceptance criteria. The objective is not to compare a production system with and without predictive inspection, but rather to analyze how the maturation of an AI-based inspection model affects system-level performance during ramp-up.

This configuration represents a static inspection regime and serves as a structural benchmark. The comparison with the nominal AI-enabled scenario will therefore isolate the effect of learning dynamics, rather than the mere presence of a predictive inspection architecture.

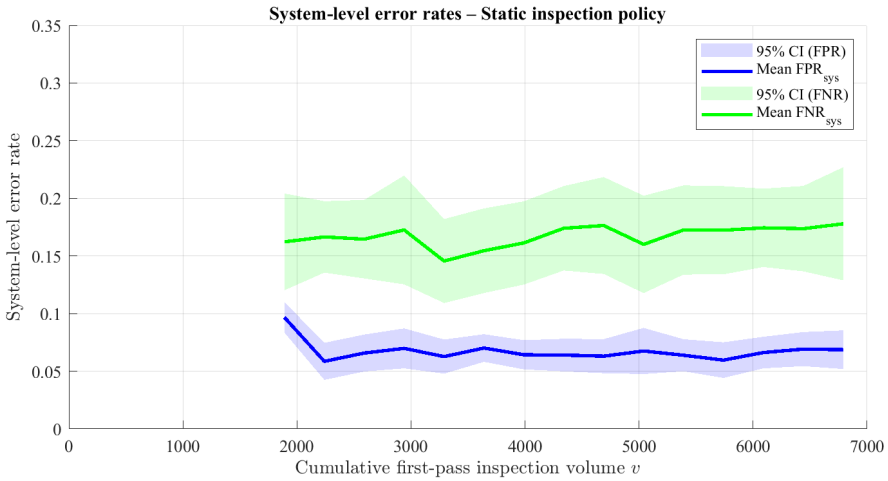

Figure 5.1: FPR and FNR mean - Static inspection policy.

Figure 5.1 reports the evolution of the system-level error rates $FPR_{sys}(v)$ and $FNR_{sys}(v)$ under this static regime. As expected, both indicators fluctuate around stable mean levels without exhibiting any monotonic trend with respect to cumulative first-pass inspection volume v. Across the simulation horizon, FPR_{sys} stabilizes around approximately 7%, while FNR_{sys} remains around 16–18%, with limited dispersion across Monte Carlo replications. These plateau values represent the structural error floor imposed by the deterministic decision rule and the statistical properties of the predictive signal. The absence of systematic convergence confirms that no maturity effects are present in this configuration and that residual fluctuations are solely attributable to finite sampling and stochastic variability across runs.

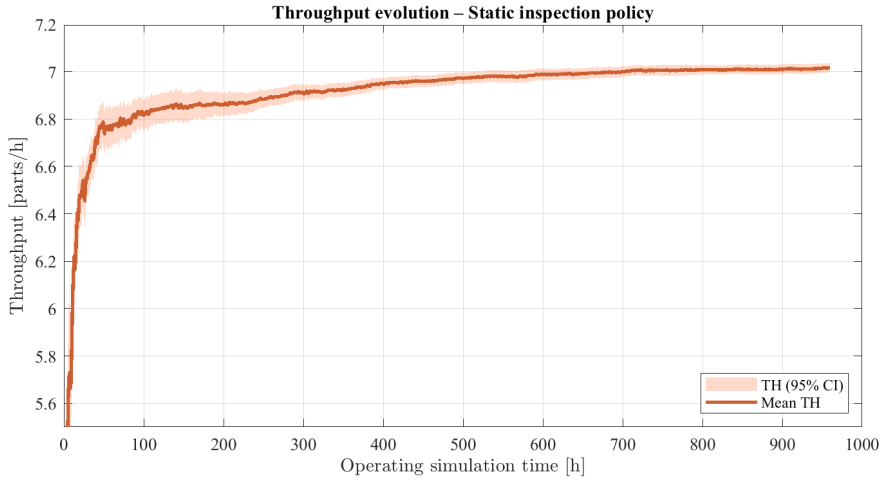

Figure 5.2: Throughput mean - Static inspection policy.

Finally, Figure 5.2 illustrates the corresponding evolution of system throughput. After the initial production transient associated with line filling, throughput stabilizes around approximately 7 parts/hour. This behavior indicates that, in the absence of learning dynamics, system productivity is entirely determined by structural process characteristics and fixed inspection rules.

This stable productivity level therefore serves as a structural benchmark against which the potential throughput gains enabled by AI maturity will be assessed in the following sections.

5.2.2. AI Maturity Profile

Having defined the static structural benchmark, we now introduce the theoretical AI maturity profile that drives the dynamic evolution of decision reliability. Figure 5.3 illustrates the theoretical maturity profile of the AI-based inspection system under baseline conditions. The curves $\alpha(v)$ and $\beta(v)$ describe the evolution of the classification error probabilities as a function of cumulative first-pass inspection volume v, which in this study represents the abstract index of production experience acquired during ramp-up.

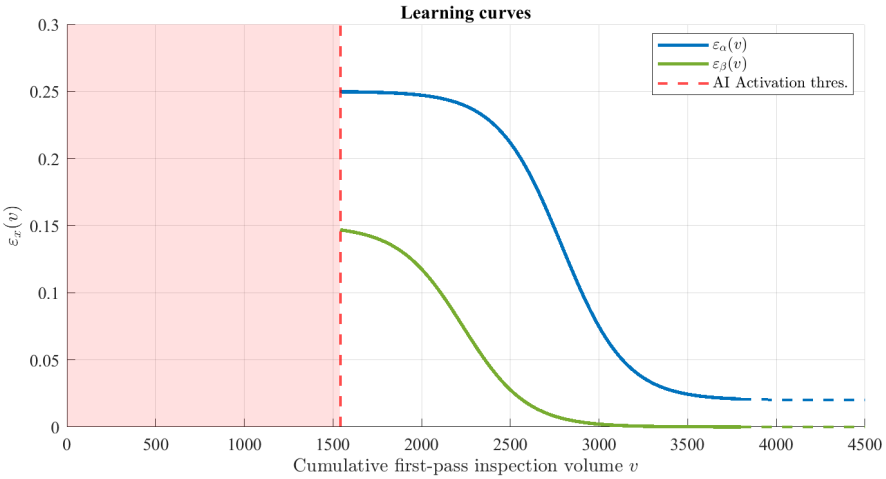

Figure 5.3: Evolution of the classification error probabilities.

The sigmoid shape confirms the assumed gradual learning mechanism, characterized by three distinct phases:

(i) an initial high-error region corresponding to the early ramp-up stage, where limited production experience leads to reduced classification reliability;

(ii) a transition phase around the inflection points $v_{t,\alpha}$ and $v_{t,\beta}$, where learning accelerates and error probabilities decrease more rapidly;

(iii) a saturation phase approaching the asymptotic values $\varepsilon_{\infty,\alpha}$ and $\varepsilon_{\infty,\beta}$, representing the stabilized performance of a mature inspection system.

Intentionally, a structural asymmetry is introduced between the two curves, since the reduction of $\beta(v)$ precedes that of $\alpha(v)$, as reflected by $\Delta V_\beta < \Delta V_\alpha$. This modeling assumption implies that the system first improves its capability to correctly identify defective parts (reducing FN) before progressively minimizing unnecessary full-cycle inspections caused by FP. From an operational standpoint, this ordering implies that the system first improves its reliability in handling defective parts during the initial inspection stage, while the reduction of unnecessary full-cycle inspections on conforming parts occurs more gradually.

It is important to emphasize that these curves represent exogenous scenario inputs rather than simulation outputs. The subsequent analysis will therefore assess how this imposed learning mechanism causally propagates to system-level decision reliability, operational workload, throughput performance and cumulative risk during ramp-up.

5.2.3. Decision Reliability at System Level

The system-level impact of the imposed learning curves is evaluated through the evolution of the observed false positive and false negative rates derived from the confusion matrix at the exit of the quality control station (identified as S60 in Tecnomatix simulation model). In order to ensure methodological consistency with the definition of cumulative first-pass inspection volume v, the confusion matrix is computed using a first-pass-only logic. Each produced unit contributes at most once to the counts of FP, FN, TP and TN, through a dedicated flag that prevents double-counting during possible retesting loops. Retesting operations therefore increase inspection workload but do not alter the measurement of decision reliability, since they do not correspond to new production evidence.

For each evaluation batch k, the empirical system-level indicators are computed as:

$$\begin{aligned} FPR_{sys,k} &= \frac{FP_k}{FP_k + TN_k} \\ FNR_{sys,k} &= \frac{FN_k}{FN_k + TP_k} \end{aligned} \tag{5.2}$$

where FP_k, TN_k, FN_k, TP_k are the first-pass confusion matrix counts accumulated over the k-th batch of inspected units.

Given the stochastic nature of the simulation model, each experimental configuration is replicated multiple times. For each batch, the indicators are averaged across Monte Carlo replications in order to distinguish structural learning trends from random variability:

$$\begin{aligned} \overline{FPR}_k &= \text{mean}_r \left(FPR_{sys}^{(r,k)} \right) \\ \overline{FNR}_k &= \text{mean}_r \left(FNR_{sys}^{(r,k)} \right) \end{aligned} \tag{5.3}$$

Figure 5.4 illustrates the evolution of $FPR_{sys}(v)$ and $FNR_{sys}(v)$ under baseline conditions. At the beginning of the ramp-up phase, both indicators assume relatively high values, reflecting the elevated theoretical error probabilities $\alpha(v)$ and $\beta(v)$ and the limited production experience available to the inspection system. As cumulative first-pass inspection volume increases, a clear decreasing trend emerges. The earlier decline of $\beta(v)$, as imposed in the theoretical maturity profile, induces a preliminary reduction in the observed FNR. Subsequently, the more gradual decrease of $\alpha(v)$ drives the sustained decline in FPR throughout the transition phase.

After the transition region, both indicators converge toward stable asymptotic levels. Notably, while $\beta(v)$ approaches zero in the theoretical model, $FNR_{sys}(v)$ stabilizes at a strictly positive value. This residual level reflects the interaction between the deterministic

tolerance-based rule and the inherent variability of the predictive signal.

It is important to emphasize that $\alpha(v)$ and $\beta(v)$ do not coincide directly with the empirical system-level rates $FPR_{sys}(v)$ and $FNR_{sys}(v)$, as they regulate only the stochastic refinement component of the decision mechanism.

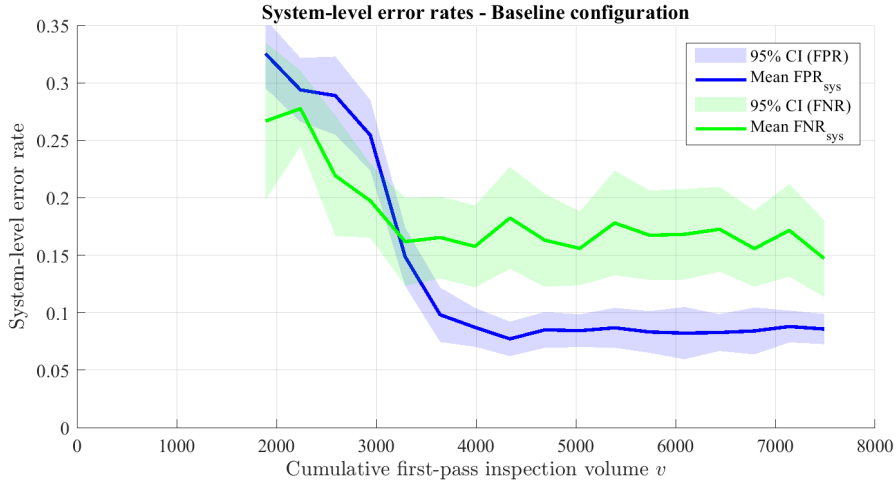

Figure 5.4: FPR and FNR - Baseline configuration.

A comparison with the static inspection regime highlights the operational impact of maturity-driven stochastic refinement: while the static configuration exhibits stationary error rates determined solely by structural characteristics, the baseline scenario demonstrates progressive improvements during ramp-up, isolating the effect of learning dynamics from structural constraints.

5.2.4. Throughput Stabilization

System-level productivity effects are evaluated through the evolution of hourly throughput (TH), computed as the total number of conforming units exiting the production system per hour. Unlike the cumulative first-pass inspection volume used for reliability analysis, throughput reflects overall system performance and integrates the effects of inspection decisions, rework flows and operational workload. Under baseline conditions, the activation of maturity-driven stochastic refinement progressively modifies inspection decisions, influencing the proportion of full-cycle operations and consequently the effective system load.

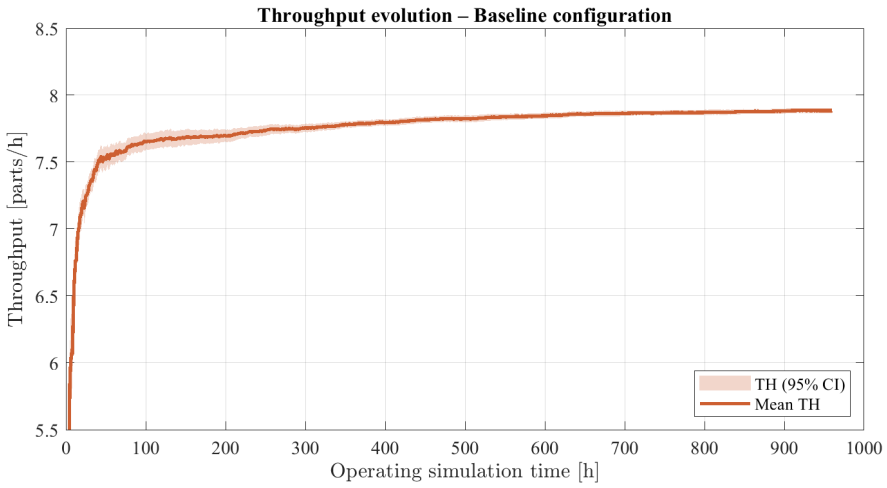

Figure 5.5: Throughput mean - Baseline configuration.

Figure 5.5 shows the evolution of mean system throughput across 20 Monte Carlo replications, with 95% confidence intervals. Baseline scenario stabilizes at a substantially higher throughput level (approximately 7.9 parts/hour) compared to the static regime (approximately 7.0 parts/hour). The observed increase in throughput during ramp-up is not driven by changes in structural production capacity, but by the reduction in unnecessary full-cycle inspections induced by decreasing $\alpha(v)$ and $\beta(v)$. Since stochastic misclassification events become less frequent, the system experiences a lower inspection workload and therefore an higher effective productivity.

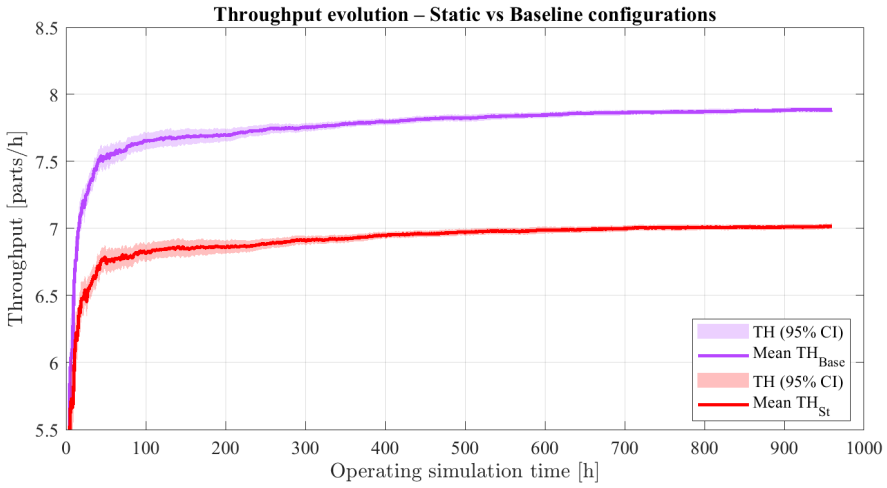

Figure 5.6: Throughput mean comparison - Static vs Baseline.

To assess whether improvements in decision reliability translate into measurable operational gains, system throughput is analyzed under both the static inspection regime and

the baseline configuration with learning enabled. Figure 5.6 compares the evolution of mean throughput.

To formally quantify the performance gap observed, a steady-state comparison over the final 200 operating simulation hours has been made. The outcomes are reported in Table 5.2:

Scenario	Mean TH (parts/h)	ΔTH vs Static (parts/h)	Improvement (%)
Static	7.01	–	–
Baseline	7.88	+0.87	+12.4

Table 5.2: Steady-state throughput comparison

This quantitative comparison confirms that AI maturity does not merely improve classification accuracy, but generates structurally significant operational gains at system level.

5.2.5. Composite Risk and AUC_R

To provide a cumulative measure of operational exposure during ramp-up, the composite system-level risk is computed by replacing the theoretical error probabilities with the observed system-level rates. At each experience level v, the instantaneous composite risk is defined as:

$$R_{sys}(v) = c_\alpha FPR_{sys}(v) + c_\beta FNR_{sys}(v) \tag{5.4}$$

where $c_\alpha = 0.22$ and $c_\beta = 0.78$ reflect the normalized misclassification cost weights defined in Section 3.2.11.

Figure 5.7 shows the evolution of the instantaneous composite risk during ramp-up, where the AI-enabled baseline configuration exhibits a pronounced high-risk phase in the early production stage, reflecting elevated misclassification rates while the learning mechanism is still immature. The composite risk decreases and progressively converges toward the static regime, as cumulative inspection volume increases. In the steady-state region, the baseline configuration stabilizes at comparable or slightly lower risk levels, confirming that learning improves long-term decision reliability.

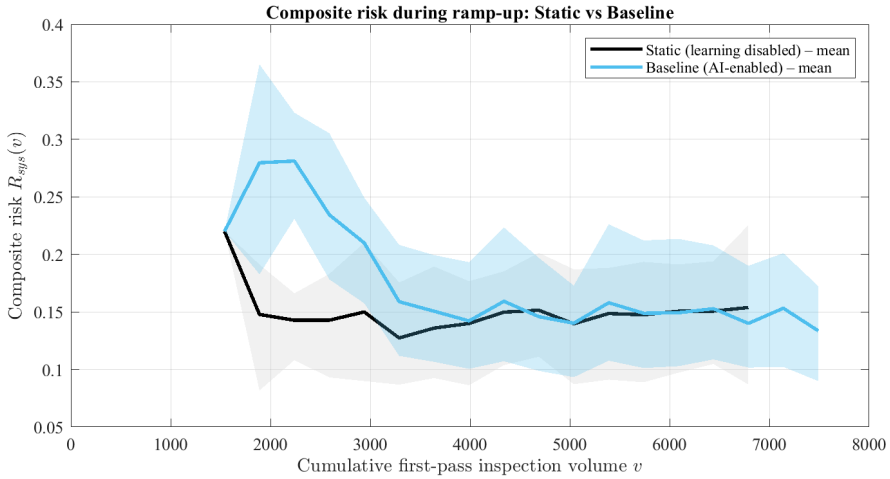

Figure 5.7: Instantaneous composite risk – Static vs Baseline.

Moreover, to capture the overall exposure accumulated over the ramp-up horizon, the Area Under the Composite Risk curve (AUC_R) is computed, which is numerically approximated using the trapezoidal rule over the discrete evaluation batches:

$$AUC_R \approx \sum_{k=1}^{n-1} \frac{R_k + R_{k+1}}{2} (v_{k+1} - v_k) \tag{5.5}$$

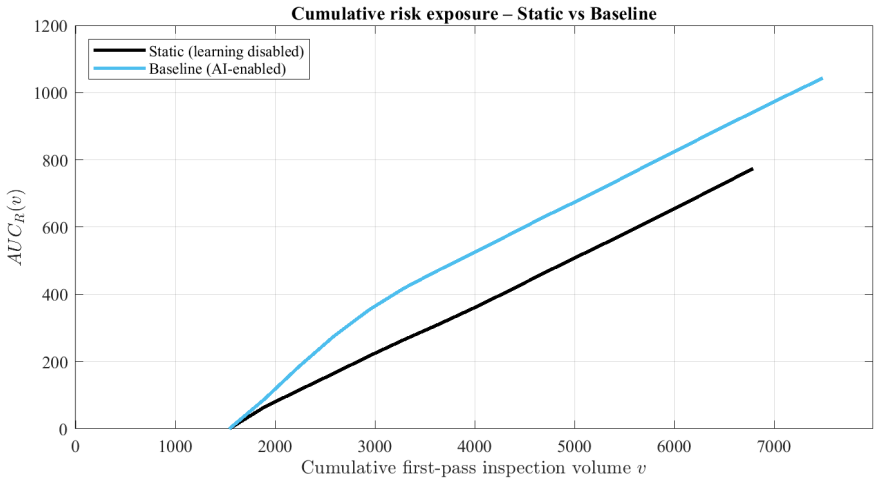

Figure 5.8: Cumulative risk exposure – Static vs Baseline

Figure 5.8 illustrates the cumulative composite risk exposure as a function of experience. Due to the elevated risk in the initial learning phase, the AUC_R curve of the baseline configuration grows more rapidly and remains above that of the static regime throughout

the observed horizon.

The final AUC_R values are summarized in Table 5.3:

Scenario	AUC_R	Δ vs Static	Δ (%)
Static	774.14	–	–
Baseline	1043.38	+269.25	+34.8%

Table 5.3: Composite risk (AUC_R) comparison across scenarios.

The baseline configuration accumulates 34.8% more composite operational risk during ramp-up compared to the static regime. This result indicates that, although AI maturity improves steady-state reliability and throughput performance, it introduces a transient high-risk phase during early adoption. The AI-enabled system therefore entails a structural trade-off: improved long-term efficiency, but increased short-term exposure to misclassification costs. This dynamic reflects a system mechanism, in which the learning phase requires an initial operational investment before performance gains fully materialize.

The analysis conducted under the baseline configuration shows that while AI maturity improves steady-state decision reliability and throughput performance, it simultaneously introduces a transient phase of elevated operational risk during early ramp-up. This outcome suggests that system performance is not only determined by the presence of an AI component, but by the specific maturity dynamics governing its evolution. The following section examines the impact of AI maturity parameters on system-level performance indicators to investigate this dependency.

5.3. Impact of AI Maturity on System Performance

The baseline configuration analyzed in Section 5.2 established the reference behavior of the AI-enabled inspection policy, highlighting two fundamental properties: improved steady-state throughput compared to the static regime and increased cumulative exposure during ramp-up due to initial learning dynamics.

However, the baseline maturity trajectory represents only one nominal parameterization of the learning process. The evolution of classification performance may vary in terms of overall learning speed and relative timing of false positive and false negative reduction. To evaluate the sensitivity of the system to such variations, this section analyzes alternative maturity configurations defined in Table 5.1, so the objective is not to modify the

structural logic of the inspection policy, but to investigate how different experience-driven learning trajectories influence the composite risk and its cumulative counterpart AUC_R.

Importantly, learning dynamics are modeled as functions of cumulative inspection volume v and not of time. Therefore, maturity variations correspond to displacements and scaling effects along the experience axis; this means that the system does not learn over time, but rather through accumulated production exposure.

5.3.1. Effect of learning speed W

To assess the impact of learning speed on system-level performance, the nominal scenario ($W = 500$) is compared with a faster learning configuration ($W = 400$) and a slower one ($W = 600$). The static policy is retained as a no learning benchmark.

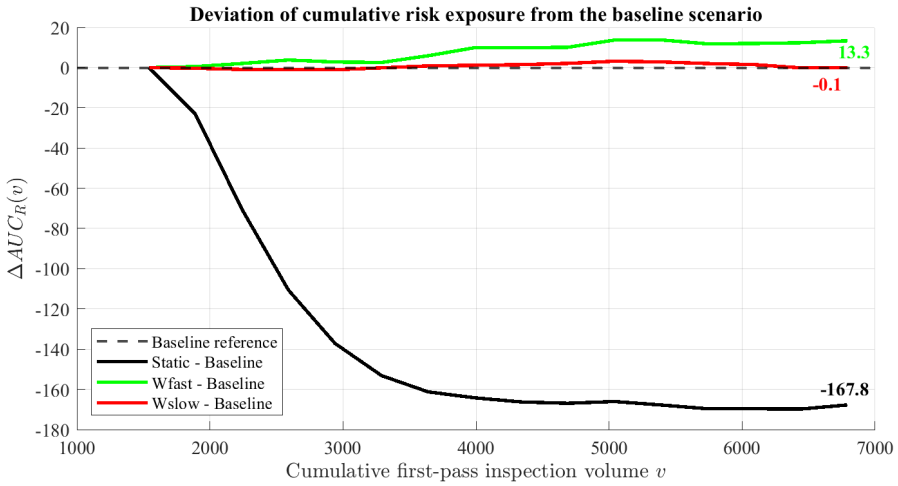

Figure 5.9: Learning speed comparison - Deviation from baseline configuration.

Figure 5.9 reports the cumulative exposure expressed as the relative deviation:

$$\Delta AUCR(v) = AUCR(v) - AUCR_{baseline}(v) \tag{5.6}$$

As already highlighted in Section 5.2, the static policy exhibits the lowest cumulative exposure. Although it does not improve over time, it avoids the initial high-error phase associated with adaptive learning, so this confirms that learning introduces a ramp-up cost.

Comparing the three learning configurations, the difference between Baseline and Wslow is negligible (-0.01%), indicating that slightly slower learning does not materially change cumulative exposure over the considered horizon. In contrast, Wfast produces a higher cu-

mulative exposure than the nominal case (+13.3 units, approximately +1.4%). Although faster learning reduces errors earlier in the trajectory, it also modifies the transient profile of $R_{sys}(v)$, increasing exposure during the early ramp-up region.

Overall, learning speed affects cumulative risk only marginally within the tested range, suggesting that maturity timing (investigated in Section 5.4) may play a more structurally relevant role than global learning acceleration.

5.3.2. Effect of asymmetric improvement

To isolate the impact of asymmetric maturity timing, the nominal baseline configuration is compared with four scenarios in which the improvement trajectory of either α (FPR related component) or β (FNR related component) is shifted along the experience axis v. Unlike Section 5.3.1, where learning speed was modified globally, here the overall convergence profile remains unchanged, but the onset of improvement for a single component is displaced earlier or later in terms of cumulative experience.

Figure 5.10 reports the deviation in cumulative exposure:

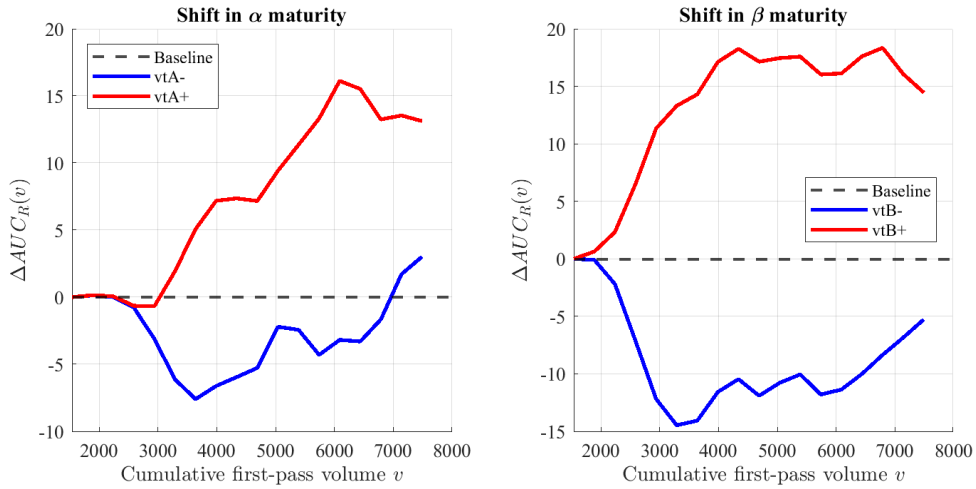

Figure 5.10: Relative cumulative risk deviation from baseline.

The table below shows the final AUC_R values:

Scenario	AUC_R	Δ vs Baseline	Δ (%)
Baseline	1043.38	–	–
vtA−	1046.41	+3.02	+0.29%
vtA+	1056.51	+13.12	+1.26%
vtB−	1038.12	-5.27	-0.50%
vtB+	1057.84	+14.45	+1.38%

Table 5.4: Composite risk comparison under asymmetric improvement scenarios.

Anticipating the α improvement (vtA−) produces only a marginal change in cumulative exposure (+0.29%), while delaying it (vtA+) increases AUC_R by +1.26%. This indicates that shifts in FPR reduction timing have only a modest impact on overall system-level exposure.

The β shift scenarios exhibit a stronger effect, because anticipating β improvement (vtB−) is the only configuration that reduces cumulative exposure (−0.50%), whereas delaying it (vtB+) yields the largest increase (+1.38%).

This asymmetry follows directly from the composite risk structure $R_{sys}(v)$, where FNR carries substantially greater weight. As a result, sequencing the reduction of β dominates cumulative risk dynamics, while α timing plays a secondary role.

It is possible to conclude that the cumulative exposure is not primarily driven by how fast the system learns, but by which error component improves first. The distribution of risk reduction along the ramp-up trajectory matters more than the global convergence rate. These findings suggest that maturity sequencing and activation timing are more critical design levers than learning acceleration alone. Finally, it is worth noting that varying the global learning speed (W) uniformly scales the convergence dynamics without altering the structure of error reduction. In contrast, shifting maturity timing (vtA/vtB) changes which error component improves first, thereby modifying the distribution of risk along the ramp-up trajectory.

5.4. Scenario Analysis: Early vs Late AI Activation

This section evaluates the operational implications of alternative AI activation policies, comparing *early* and *late* activation strategies. The objective is to determine whether activating the AI component (i.e., the adaptive learning layer) earlier or later along the cumulative production trajectory leads to measurable differences in cumulative system-level risk exposure (AUC_R), transient error dynamics and steady-state operational performance (throughput).

Importantly, this is not a comparison of model structures or learning configurations, but strictly a comparison of activation timing policies under identical system and learning parameters. Therefore, any observed difference can be directly attributed to the temporal positioning of the AI regime along the production ramp-up.

The guiding decision focuses on assessing whether earlier or later activation of the AI results in more favorable outcomes in terms of cumulative risk exposure and operational performance.

5.4.1. Scenario definition and activation thresholds

Two alternative activation policies are analyzed: an early activation policy (S01) and a late activation policy (S03). Both scenarios share identical system configuration, stochastic parameters, learning curves $\alpha(v)$ and $\beta(v)$, and simulation horizon, so no structural modification of the decision logic or learning dynamics is introduced. The only difference concerns the operational activation threshold V_{start}^{op}.

The difference between the two policies is therefore purely temporal: early activation anticipates the transition to the AI-enabled regime, while Late activation delays it. The operational thresholds adopted in the two scenarios are reported in Table 5.5:

Scenario	Description	V_{start}^{op} [parts]	Activation basis
S01	Early activation	840	First-pass volume
S03	Late activation	2240	First-pass volume

Table 5.5: AI activation policy scenarios.

5.4.2. Effect on error rates and instantaneous risk

The impact of the activation policy is first analyzed at the level of system-level error rates. In this analysis, any deviation between early and late activation must originate exclusively from the temporal shift in AI enablement.

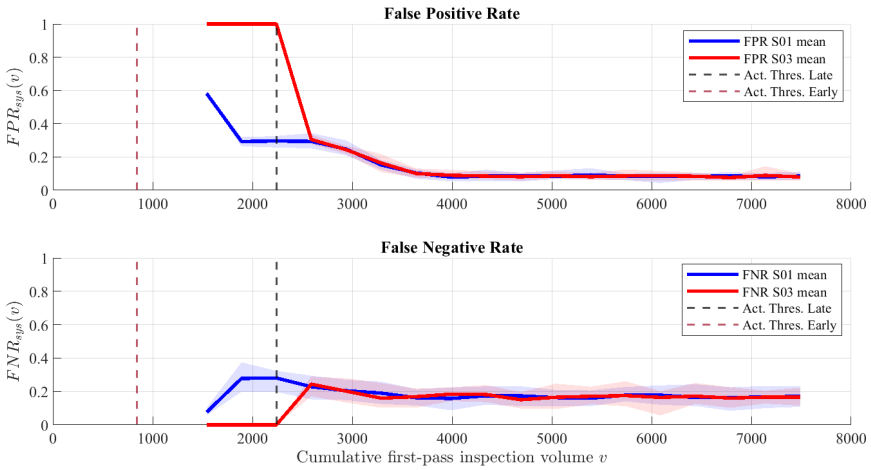

Figure 5.11: Early vs Late activation policy - System-level error rates.

Figure 5.11 reports the evolution of $FPR(v)$ and $FNR(v)$ for the two policies, as functions of cumulative first-pass inspection volume v. Shaded areas represent the 5th–95th percentile range across the 20 Monte Carlo replications, while the vertical dashed lines indicate the activation thresholds of the early ($v = 840$) and late ($v = 2240$) configurations. It should be noted that $FPR(v)$ and $FNR(v)$ begin to be plotted at $v = 1540$. This is a consequence of the batch-based aggregation adopted in the analysis: each plotted point corresponds to a volume interval rather than individual observations. Therefore, the conservative regime before $v = 840$ is not explicitly visible in the graph. However, in that region both scenarios follow the same deterministic logic and no divergence can occur.

The separation between the two policies emerges in the transitional window between the activation thresholds. In the graphical representation, the divergence becomes visible in the first batch following $v = 840$, and progressively vanishes after $v = 2240$, confirming that the effect is confined to the interval between the two activation volumes.

The shape of the curves reflects the mechanics of the learning process. The early configuration shows a monotonic decrease in FPR, as the learning process progressively reduces Type I errors over increasing experience. In contrast, FNR exhibits a temporary increase immediately after activation. This occurs because the conservative regime prior

to activation strongly limits Type II errors; once AI is enabled, stochastic refinement is introduced and some false negatives initially emerge. Finally, as learning stabilizes, FNR progressively decreases toward its steady-state level.

Overall, the activation policy modifies the timing of exposure to adaptive decision-making, but not the asymptotic behavior of the system. The observed differences are localized in the transition phase and vanish once both configurations operate under the same learning regime.

5.4.3. Cumulative Risk Exposure

While the previous subsection analyzed the local dynamics of FPR and FNR, the activation decision must ultimately be assessed in cumulative terms. For this reason, the AUC_R KPI is evaluated.

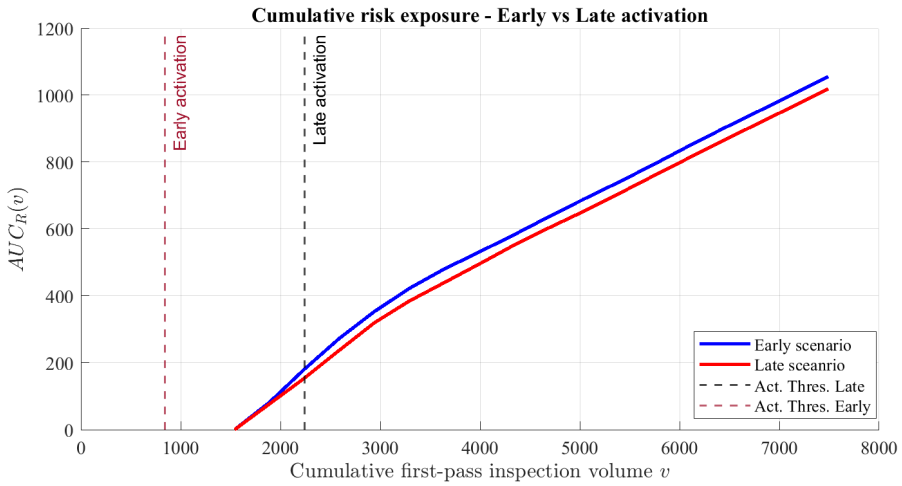

Figure 5.12: Cumulative risk exposure - Early vs Late activation.

Figure 5.12 reports the cumulative exposure for early and late activation. As discussed in Section 5.4.2, early activation induces a temporary increase in FNR during the transition to the adaptive regime, leading to a short-lived rise in instantaneous composite risk. This effect results in a systematically higher cumulative exposure for early activation. The late configuration, by remaining longer in the conservative regime, avoids the initial FNR overshoot and accumulates less risk during the transitional phase. Although the two policies converge after the late activation threshold and exhibit nearly identical steady-state dynamics, the cumulative effect of the early overshoot persists.

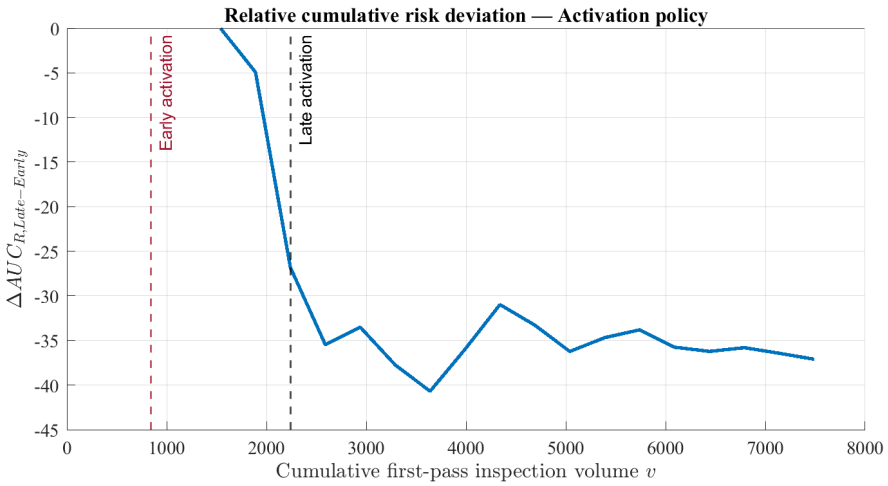

Figure 5.13: Relative cumulative risk deviation - Early vs Late activation.

The final cumulative values confirm this behavior:

Scenario	AUC_R	**Δ vs Early**
Early activation	1056	–
Late activation	1018.9	−37.10

Table 5.6: Final composite risk comparison - Early vs Late activation policies.

This corresponds to an approximate 3.5% reduction in cumulative exposure for the late activation policy. After the late activation threshold, the cumulative curves evolve with nearly parallel slopes, indicating that steady-state dynamic is identical. The observed difference is therefore entirely attributable to the transitional phase and remains as a persistent offset.

The activation policy does not alter the long-run behavior of the system, in fact both configurations converge to the same steady-state error dynamics. However, it does affect how much cumulative risk is accumulated during the transition toward adaptive operation. In the analyzed configuration, where Type II errors (FNR) are heavily weighted in the risk function, activating earlier exposes the system sooner to the transient instability of the learning phase, producing a higher cumulative exposure that is not recovered later. Therefore, in terms of cumulative risk exposure, delaying activation is preferable under the adopted risk structure.

This outcome directly reflects the dominance of FNR in the composite risk function and the transient FNR overshoot observed after early activation. Different risk weightings

or alternative transient dynamics would modify this balance. Hence, activation timing should be interpreted as a function of the underlying risk structure and learning dynamics, rather than as an intrinsically optimal early or late choice.

5.4.4. Impact on operational performance

After analyzing the effect of activation timing on error rates and cumulative risk, the operational implications are evaluated in terms of throughput dynamics and steady-state performance.

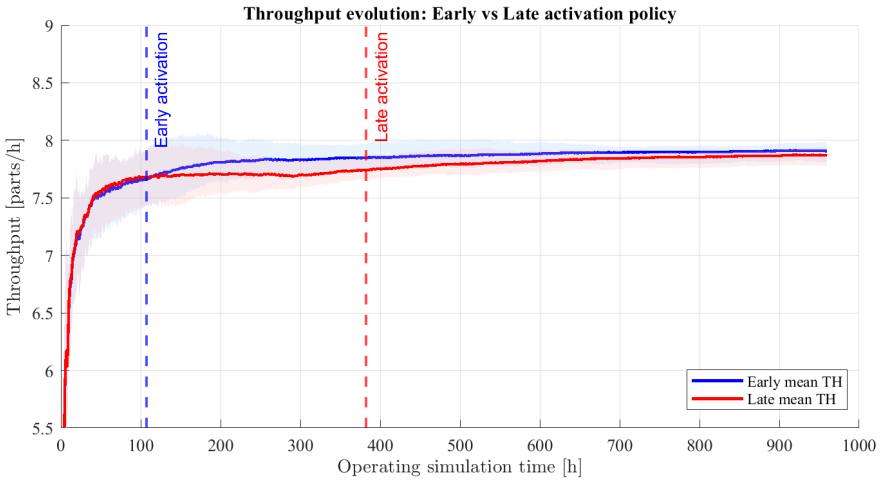

Figure 5.14: Throughput evolution - Early vs Late activation policy.

Figure 5.14 shows the evolution of the system throughput over time for the early and late activation policies (including smoothing and p05–p95 variability bands). The two configurations exhibit nearly identical trajectories during the initial production ramp-up. Differences emerge only after the early learning activation threshold, where the early policy begins to benefit from adaptive refinement sooner. From that point onward, the early trajectory remains slightly above the late one for most of the horizon. In fact, it is possible to note that between the early and late activation thresholds, the early configuration exhibits a slightly higher throughput. This effect reflects the earlier introduction of adaptive learning, which progressively reduces conservative full-cycle inspections and marginally improves operational efficiency. However, this marginal productivity advantage does not compensate for the cumulative risk dynamics discussed in Section 5.4.3.

However, the magnitude of this difference is limited, indeed the steady-state throughput is 7.88 parts/h and 7.83 parts/h, respectively, for early and late activation policies, as shown in the table below. The gap is approximately 0.05 parts/h, corresponding to less

than 1% of the steady-state level.

Policy	Activation time [h]	Steady-state onset t_{ss} [h]	TH_{ss} [parts/h]
Early	106.5	197.4	7.88
Late	382.6	400	7.83

Table 5.7: Activation timing and steady-state throughput comparison

Table 5.7 also reports the onset time of steady-state throughput for the two activation policies. In both configurations, steady-state is reached after the activation threshold, confirming the presence of a transient adjustment phase induced by the introduction of adaptive learning.

Moreover, stepwise representation and the box-plot of steady-state throughput per run provide further insight (Figure 5.15). It is worth noting that horizontal segments (left plot) denote pre-learning and steady-state throughput, while vertical transitions indicate activation timing for the early and late policies. The pre-activation throughput is computed at an operating time equal to 100 hours, corresponding to a time window immediately preceding the earliest learning activation, because at this point, all configurations operate under the same deterministic regime ensuring structural comparability across policies.

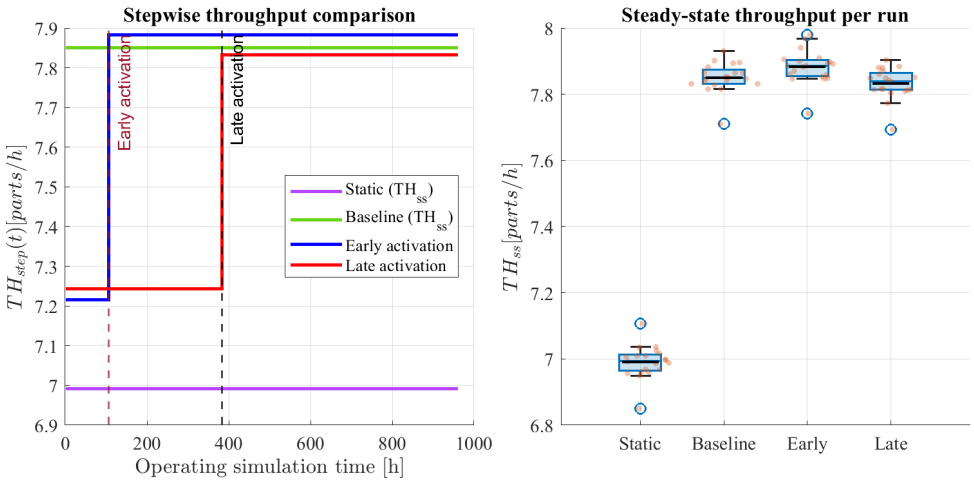

Figure 5.15: Throughput comparison under different activation policies.

The stepwise representation highlights the regime-driven nature of the throughput dynamics. A separation emerges between the pre-learning level and the post-activation

steady-state plateau, indicating that performance shifts occur when the decision regime changes rather than evolving gradually over time. Early activation produces this transition sooner, while late activation delays it; however, both configurations converge to nearly the same steady-state level. This confirms that the difference between policies is temporal rather than structural. The comparison with the static and baseline scenarios further reinforces that the system architecture remains unchanged and only the timing of adaptive refinement differs.

The ranking observed in the stepwise representation (early > baseline > late) reflects the timing of adaptive learning introduction. Earlier activation allows operational efficiency gains to materialize sooner, resulting in a slightly higher steady-state throughput. Conversely, delayed activation postpones these gains, leading to a marginally lower plateau. However, the magnitude of these differences remains limited (<1%), confirming that activation timing influences performance primarily in temporal rather than structural terms.

The box-plot of steady-state throughput per Monte Carlo run confirms that differences between early and late activation remain limited. Although the early configuration exhibits a slightly higher median throughput, the interquartile ranges largely overlap, indicating that the observed gap is small relative to simulation variability. No structural separation between the two activation policies emerges at steady state. This further supports the conclusion that activation timing does not materially alter long-run productivity.

5.4.5. Integrated interpretation of activation timing

The objective of this section was to determine whether activating the learning mechanism earlier or later is preferable in terms of cumulative risk exposure and operational performance. Taken together, the results provide a coherent and internally consistent picture. The activation policy does not alter the structural learning dynamics of the system nor its steady-state throughput; rather it shifts the timing of transition to the AI-enabled regime.

From an error dynamics perspective (Section 5.4.2), the effect of activation timing is strictly confined to the transitional window between the two activation thresholds. Before the early threshold, both configurations operate under the same conservative regime and behave identically. After the late threshold, both operate under the same adaptive regime and converge again. Early activation anticipates adaptive refinement, but also anticipates the transient instability associated with regime switching. In particular, the temporary increase in FNR immediately after activation amplifies the composite risk due to its higher weighting.

When evaluated cumulatively (Section 5.4.3), this transient divergence becomes structurally embedded in the integral of the composite risk. Although instantaneous differences are limited to the activation window, their cumulative impact persists throughout the analysis horizon. Under the adopted risk weighting, the late activation policy results in lower cumulative risk exposure. Importantly, this difference does not originate from steady-state behavior, but from the interaction between transient error dynamics and the relative importance assigned to FP and FN.

Putting these elements together, the trade-off is asymmetrical: the productivity gain associated with earlier activation is marginal, whereas the cumulative risk effect is structurally embedded in the transition phase. In the analyzed configuration, delaying activation reduces cumulative risk exposure without penalizing steady-state throughput. However, this conclusion cannot be generalized in a simplistic way. Activation timing cannot be evaluated independently of the risk structure of the system. Early activation anticipates adaptive learning, but also anticipates exposure to transient variability. If the transient phase amplifies the dominant risk component, cumulative exposure can increase; conversely, if learning rapidly suppresses the dominant error mode, earlier activation can reduce cumulative risk.

The optimal activation threshold therefore emerges from the interaction between:

- the transient behavior of error rates,
- the relative weighting of type-I and type-II risk components,
- operational constraints of the system.

Thus, in simplified terms:

- Activating earlier means learning earlier, but also being exposed earlier to transitional instability.
- Activating later means remaining longer in a stable deterministic regime, while postponing adaptive refinement.
- The preferable strategy depends on which error component dominates the risk structure and how it evolves during learning.

The activation policy thus represents fundamentally a risk-timing decision, rather than a productivity decision.

5.5. Sensitivity analysis of risk weighting

The results of Section 5.4 indicate that delayed activation leads to lower cumulative risk exposure under the adopted composite risk weighting. However, this conclusion depends on the relative importance assigned to false positives and false negatives in the risk function.

To assess the robustness of the activation decision, this section performs a sensitivity analysis with respect to the weighting coefficients of the composite risk. The objective is to determine whether the preference for Early or Late activation remains stable or changes when the risk structure is modified.

5.5.1. Methodology

To evaluate the robustness of the activation decision, the composite risk function is reformulated as a parametric combination of the two error components:

$$R_\lambda(v) = \lambda\, FPR(v) + (1-\lambda)\, FNR(v), \quad \lambda \in [0,1] \tag{5.7}$$

where λ represents the relative weight assigned to false positives, while $(1-\lambda)$ corresponds to the weight of false negatives.

For each value of λ, the cumulative risk exposure is computed as:

$$AUC_{R,\lambda}(v) = \int_{v_{start}}^{v_{end}} R_\lambda(u)\, \mathrm{d}u \tag{5.8}$$

And the decision indicator at the end of the analysis horizon is defined as:

$$\Delta AUC_{R,\mathrm{end}}(\lambda) = AUC_{R,\mathrm{Late,end}}(\lambda) - AUC_{R,\mathrm{Early,end}}(\lambda) \tag{5.9}$$

The sign of this quantity determines the preferable activation policy:

$$\Delta AUC_{R,\mathrm{end}}(\lambda) < 0 \;\Rightarrow\; \text{Late activation yields lower cumulative risk} \tag{5.10}$$

$$\Delta AUC_{R,\mathrm{end}}(\lambda) > 0 \;\Rightarrow\; \text{Early activation is preferable} \tag{5.11}$$

The sensitivity analysis is performed by recomputing the composite risk and its cumulative integral using the previously obtained FPR and FNR trajectories for both configurations.

5.5.2. Results and break-even weighting

The sensitivity analysis evaluates the decision indicator defined in Equation (5.9) over the full range $\lambda \in [0,1]$.

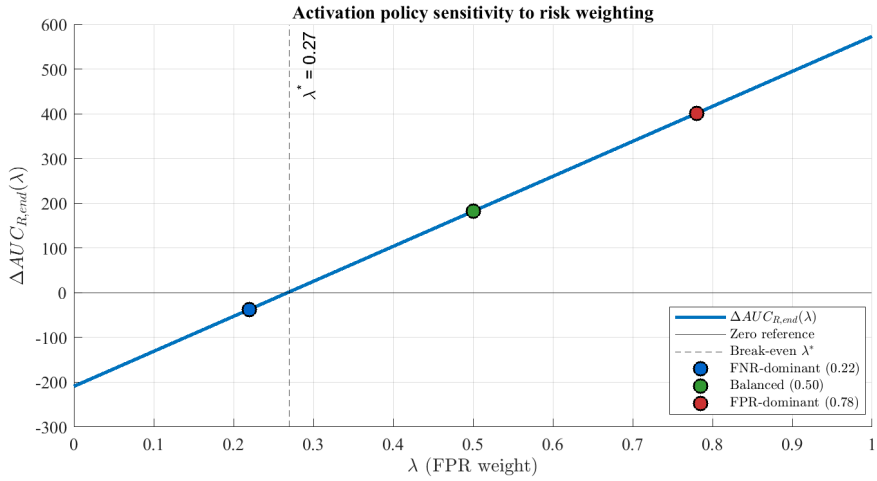

Figure 5.16: Sensitivity of cumulative risk difference to risk weighting.

Figure 5.16 shows a linear and monotonic increase of $\Delta AUC_{R,\text{end}}(\lambda)$, consistent with the linear structure of the composite risk function. For low values of λ, where the composite risk is dominated by FNR, the cumulative exposure of the late configuration is lower, yielding negative values of $\Delta AUC_{R,\text{end}}(\lambda)$. As the weight assigned to FPR increases, the advantage progressively shifts toward early activation.

The curve crosses the zero axis at $\lambda^* = 0.27$, which represents the break-even weighting between the two policies. For:

- $\lambda < 0.27$: late activation yields lower cumulative risk.
- $\lambda > 0.27$: early activation becomes preferable.

In the baseline configuration adopted in Section 5.4 ($\lambda = 0.22$), the system lies within the FNR-dominated region, explaining the previously observed preference for delayed activation.

To complement the analysis, three representative weighting structures are considered:

The discrete sensitivity cases confirm the inversion identified in the continuous analysis. Under FNR-dominant weighting, delayed activation reduces cumulative exposure by approximately 3.5%. However, when the importance of FPR increases, the advantage shifts toward Early activation, reaching differences above 40% in the FPR-dominant case.

Case	c_α	c_β	$AUC_{R_{\text{end}}}^{\text{Early}}$	$AUC_{R_{\text{end}}}^{\text{Late}}$	$\Delta AUC_{R_{\text{end}}}$	$\Delta AUC_{R_{\text{end}}}$ (%)
A	0.22	0.78	1056	1018.9	−37.10	−3.514
B	0.50	0.50	999.06	1181	+181.92	+18.21
C	0.78	0.22	942.12	1343.1	+400.94	+42.55

Table 5.8: Final AUC_R sensitivity analysis.

This behaviour is structurally consistent with the underlying error dynamics. Early activation anticipates the transient increase in FNR, which becomes particularly penalizing when false negatives dominate the composite risk. Conversely, late activation maintains higher FPR levels for a longer portion of the horizon, which leads to a substantial cumulative disadvantage when false positives are assigned greater importance.

Overall, the results highlight the asymmetric interaction between transient error dynamics and risk prioritization, reinforcing that activation timing cannot be evaluated independently of the adopted risk structure.

5.6. Discussion and Implications

The results of this chapter can be summarized into three main claims regarding activation timing and industrial deployment.

Claim 1 - Activation timing affects when the system changes regime, not how it behaves in steady state.

The analysis shows that the impact of activation timing is strictly confined to the transitional window between the early and late thresholds. Before activation, both configurations operate under the same conservative logic and therefore behave identically. After the late threshold, both operate under the same adaptive regime and converge to comparable steady-state throughput levels.

The difference is therefore temporal rather than structural: activation timing determines when the system switches regime, but it does not alter long-run capacity or the fundamental learning dynamics.

Claim 2 - Transient error dynamics can create long-term cumulative effects.

Although differences in FPR and FNR are localized in the activation window, their integration over volume produces persistent effects in cumulative risk exposure.

Under the adopted risk weighting ($c_\alpha = 0.22$, $c_\beta = 0.78$), where false negatives carry greater weight, the temporary increase in FNR observed under early activation outweighs the simultaneous reduction in FPR. As a result, early activation leads to higher cumulative exposure. However, the sensitivity analysis shows that this conclusion depends on risk prioritization. A break-even weighting ($\lambda^* = 0.27$) exists, in fact below this value, late activation is preferable and above it, early activation becomes advantageous.

Activation timing is therefore not intrinsically better or worse, but conditional on how risk components are weighted.

Claim 3 – Activation timing is a governance decision, not a productivity optimization.

Throughput analysis confirms that activation timing does not materially affect steady-state performance. The observed differences between early and late configurations remain below 1% and fall within Monte Carlo variability.

Consequently activation policy should be interpreted primarily as a risk governance choice rather than a productivity lever. Therefore, the threshold defines when adaptive learning is allowed to influence operational decisions and thus reflects the organization's tolerance toward different types of error.

5.6.1. Industrial Feasibility of the TO-BE Architecture

From an industrial standpoint, the proposed TO-BE architecture can be implemented without requiring structural modifications to the production system. The predictive module operates continuously in the background, generating classification signals based on incoming inspection data. Moreover, the activation threshold does not introduce new hardware or alter process times; rather, it governs when learning-based refinement is allowed to influence operational decisions.

In practice, this architectural separation between prediction and decision integration is very important. For instance, the system might start out in "shadow mode," where predictions are gathered and watched without changing the logic of the inspection and during this phase, the organization can check the model's operational authority by looking at its predictive consistency, error patterns, and data integrity.

From a technical infrastructure perspective, implementation would primarily require: Reliable real-time tracking of first-pass production volume at the critical inspection station, as this variable governs activation. Integration between inspection systems and the existing production data infrastructure (e.g., MES or equivalent), ensuring traceability and

synchronization. A formal definition of risk priorities (establishing the relative importance of false positives and false negatives). A clearly defined governance rule specifying who authorizes activation (automatic threshold-based logic versus managerial validation).

In practical terms, deployment could go through three stages:

1. **Monitoring phase**: the predictive model runs all the time, but decisions stay very safe.
2. **Controlled activation phase**: learning-based refinement can happen under supervision, and there may be extra monitoring or escalation systems in place.
3. **Full operational integration**: activation governance is built into standard operating procedures, and the model is an important part of the inspection strategy.

The main problem is not how hard the algorithms are to understand, but how well the organization works together. In this way, the timing of activation is less of a technical limitation and more of a design choice made by management that shows how much risk the organization is willing to take on while learning integration.

6 | Conclusions

The aim of this research was to examine the impact of the implementation and activation timing of an AI-driven decision module on system performance and risk exposure within a stochastic production setting. The study focused on two main areas of research:

1. *What impact does AI maturity have on the dynamics of system errors and operational performance?*
2. *In terms of cumulative risk and throughput, is it better to turn on the learning mechanism sooner or later?*

The results can be summarized in three ways. The first thing the AI maturity analysis showed was that learning dynamics change the balance between Type-I and Type-II errors over time. The system does not merely improve over time; instead, temporary instabilities may appear before convergence. Second, the time of activation was demonstrated to only alter how the system worked during the shift from conservative to adaptive regimes. Steady-state throughput remains substantially constant; nevertheless, transitory error dynamics can produce enduring cumulative impacts when aggregated over volume. Third, the sensitivity analysis showed that there is a break-even risk weighting, which is the point at which the preferred activation policy changes. This substantiates that activation timing is mostly a risk-structuring decision rather than one influenced by production.

This thesis contributes methodologically by presenting a unified framework that combines: analysis of temporary errors, integration of cumulative risk (AUC_R), steady-state throughput assessment and an examination of the sensitivity of risk weights.

The results demonstrate that localized transient anomalies can become structurally significant when combined. This highlights the importance of combining dynamic learning analysis with cumulative performance measurements. The study also indicates that activation timing can't be looked at separately from risk prioritization. This gives a formal way to choose a threshold when there are different prices for making mistakes.

The final outcomes show that the AI learning layer should not be considered merely a technical feature; rather, it requires clear governance and alignment with the organiza-

tion's risk priorities. In practice, stakeholders from engineering, quality and operations should agree on the activation threshold.

The conclusions of this work therefore depend on the modeling assumptions adopted throughout the investigation, including the parametric specification of learning curves, error dynamics and risk weights. While these assumptions were grounded in industrial benchmarks and methodological consistency, they may vary across different production contexts.

Future research could further extend the proposed framework by investigating adaptive activation thresholds capable of dynamically responding to observed performance, integrating multi-objective optimization strategies that explicitly balance cost, downtime and risk exposure and analyzing the robustness of the model under alternative stochastic distributions. In addition, empirical validation using real industrial datasets would represent a natural and valuable next step toward strengthening the practical applicability of the proposed architecture.

Bibliography

[1] L. Surbier, G. Alpan, and E. Blanco, "A comparative study on production ramp-up: state-of-the-art and new challenges," *Production Planning & Control*, vol. 25, no. 15, pp. 1264–1286, 2014.

[2] M. C. Magnanini, K. Medini, and B. I. Epureanu, "Decision making for fast productivity ramp-up of manufacturing systems," in *CIRP Novel Topics in Production Engineering: Volume 1*. Springer, 2024, pp. 235–266.

[3] M. Colledani, T. Tolio, and A. Yemane, "Production quality improvement during manufacturing systems ramp-up," *CIRP Journal of Manufacturing Science and Technology*, vol. 23, pp. 197–206, 2018.

[4] M. C. Magnanini and T. A. Tolio, "Robust improvement planning of automated multi-stage manufacturing systems," in *Selected Topics in Manufacturing: AITeM Young Researcher Award 2021*. Springer, 2021, pp. 61–75.

[5] A. Cerqueus and X. Delorme, "Evaluating the scalability of reconfigurable manufacturing systems at the design phase," *International Journal of Production Research*, vol. 61, no. 23, pp. 8080–8093, 2023.

[6] A. Kampker, S. Wessel, N. Lutz, S. Heine, A. Mayr, and A. Kuhn, "Model improvement through real data connection for virtual commissioning in ramp-up management of scalable production systems," *Procedia CIRP*, vol. 99, pp. 645–649, 2021.

[7] M. Ugarte Querejeta, M. Illarramendi Rezabal, G. Unamuno, J. L. Bellanco, E. Ugalde, and A. Valor Valor, "Implementation of a holistic digital twin solution for design prototyping and virtual commissioning," *IET Collaborative Intelligent Manufacturing*, vol. 4, no. 4, pp. 326–335, 2022.

[8] R. Schmitt, I. Heine, R. Jiang, F. Giedziella, F. Basse, H. Voet, and S. Lu, "On the future of ramp-up management," *CIRP Journal of Manufacturing Science and Technology*, vol. 23, pp. 217–225, 2018.

[9] W. P. Neumann and P. Medbo, "Simulating operator learning during production

ramp-up in parallel vs. serial flow production," *International Journal of Production Research*, vol. 55, no. 3, pp. 845–857, 2017.

[10] S. Di Luozzo, G. R. Pop, and M. M. Schiraldi, "The human performance impact on oee in the adoption of new production technologies," *Applied Sciences*, vol. 11, no. 18, p. 8620, 2021.

[11] S. Doltsinis, P. Ferreira, M. M. Mabkhot, and N. Lohse, "A decision support system for rapid ramp-up of industry 4.0 enabled production systems," *Computers in industry*, vol. 116, p. 103190, 2020.

[12] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*. McGraw Hill/Irwin, 2012.

[13] C. Terwiesch and R. E. Bohn, "Learning and process improvement during production ramp-up," *International journal of production economics*, vol. 70, no. 1, pp. 1–19, 2001.

[14] C. Terwiesch, R. Bohn, and K. Chea, "International product transfer and production ramp-up: a case study from the data storage industry," *R&d Management*, vol. 31, no. 4, pp. 435–451, 2001.

[15] M. Haller, A. Peikert, and J. Thoma, "Cycle time management during production ramp-up," *Robotics and Computer-Integrated Manufacturing*, vol. 19, no. 1-2, pp. 183–188, 2003.

[16] J. E. Carrillo and R. M. Franza, "Investing in product development and production capabilities: The crucial linkage between time-to-market and ramp-up time," *European Journal of Operational Research*, vol. 171, no. 2, pp. 536–556, 2006.

[17] H. Winkler, M. Heins, and P. Nyhuis, "A controlling system based on cause–effect relationships for the ramp-up of production systems," *Production Engineering*, vol. 1, no. 1, pp. 103–111, 2007.

[18] S. Fjällström, K. Säfsten, U. Harlin, and J. Stahre, "Information enabling production ramp-up," *Journal of manufacturing technology management*, vol. 20, no. 2, pp. 178–196, 2009.

[19] C. Terwiesch and Y. Xu, "The copy-exactly ramp-up strategy: Trading-off learning with process change," *IEEE Transactions on Engineering Management*, vol. 51, no. 1, pp. 70–84, 2004.

[20] J. Juerging and P. M. Milling, "Interdependencies of product development decisions

and the production ramp-up," in *The 23rd International Conference of the System Dynamics Society, Boston, MA, USA*, 2005.

[21] S. Du, L. Xi, J. Ni, P. Ershun, and C. R. Liu, "Product lifecycle-oriented quality and productivity improvement based on stream of variation methodology," *Computers in Industry*, vol. 59, no. 2-3, pp. 180–192, 2008.

[22] P. D. Ball, S. Roberts, A. Natalicchio, and C. Scorzafave, "Modelling production ramp-up of engineering products," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 225, no. 6, pp. 959–971, 2011.

[23] F. Klocke, J. Stauder, P. Mattfeld, and J. Müller, "Modeling of manufacturing technologies during ramp-up," *Procedia Cirp*, vol. 51, pp. 122–127, 2016.

[24] D. Ceglarek, W. Huang, S. Zhou, Y. Ding, R. Kumar, and Y. Zhou, "Time-based competition in multistage manufacturing: Stream-of-variation analysis (sova) methodology," *International Journal of Flexible Manufacturing Systems*, vol. 16, no. 1, pp. 11–44, 2004.

[25] M. Colledani, A. Matta, and T. Tolio, "Analysis of the production variability in multi-stage manufacturing systems," *CIRP annals*, vol. 59, no. 1, pp. 449–452, 2010.

[26] M. Subramaniyan, A. Skoogh, M. Gopalakrishnan, H. Salomonsson, A. Hanna, and D. Lämkull, "An algorithm for data-driven shifting bottleneck detection. cogent eng. 3, 1239516 (2016)."

[27] C. G. Lee and S. C. Park, "Survey on the virtual commissioning of manufacturing systems," *Journal of Computational Design and Engineering*, vol. 1, no. 3, pp. 213–222, 2014.

[28] M. Colledani and T. Tolio, "Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous manufacturing systems with degrading machines," *CIRP annals*, vol. 61, no. 1, pp. 455–458, 2012.

[29] H. Almgren, "Towards a framework for analyzing efficiency during start-up:: An empirical investigation of a swedish auto manufacturer," *International journal of production economics*, vol. 60, pp. 79–86, 1999.

[30] S. Wochner, M. Grunow, T. Staeblein, and R. Stolletz, "Planning for ramp-ups and new product introductions in the automotive industry: Extending sales and operations planning," *International Journal of Production Economics*, vol. 182, pp. 372–383, 2016.

[31] M. P. Hinrichs, F. Sohnius, and R. H. Schmitt, "Production ramp-up in multi-stage discrete manufacturing systems: A conceptual framework for model-based quality maturity assurance in the automotive industry based on a systematic literature review," *Procedia CIRP*, vol. 135, pp. 855–862, 2025.

[32] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *Cirp Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[33] K. Hon, "Performance and evaluation of manufacturing systems," *CIRP annals*, vol. 54, no. 2, pp. 139–154, 2005.

[34] A. Neely, M. Gregory, and K. Platts, "Performance measurement system design: A literature review andresearch agenda," *International journal of operations & production management*, vol. 15, no. 4, pp. 80–116, 1995.

[35] M. Franco-Santos, M. Kennerley, P. Micheli, V. Martinez, S. Mason, B. Marr, D. Gray, and A. Neely, "Towards a definition of a business performance measurement system," *International journal of operations & production management*, vol. 27, no. 8, pp. 784–801, 2007.

[36] T. Tolio, A. Matta, and F. Jovane, "A method for performance evaluation of automated flow lines," *CIRP annals*, vol. 47, no. 1, pp. 373–376, 1998.

[37] A.-Y. Chang, D. J. Whitehouse, S.-L. Chang, and Y.-C. Hsieh, "An approach to the measurement of single-machine flexibility," *International Journal of Production Research*, vol. 39, no. 8, pp. 1589–1601, 2001.

[38] K. Hon and F. Lopez-Jaquez, "Configuration of manufacturing cells for dynamic manufacturing," *CIRP Annals*, vol. 51, no. 1, pp. 391–394, 2002.

[39] H.-P. Wiendahl and S. Lutz, "Production in networks," *CIRP Annals*, vol. 51, no. 2, pp. 573–586, 2002.

[40] M. E. Porter and C. v. d. Linde, "Toward a new conception of the environment-competitiveness relationship," *Journal of economic perspectives*, vol. 9, no. 4, pp. 97–118, 1995.

[41] M. T. Bosch-Badia, J. Montllor-Serrats, and M. A. Tarrazon-Rodon, "Corporate social responsibility from the viewpoint of social risk," *Theoretical Economics Letters*, vol. 4, no. 8, pp. 639–648, 2014.

[42] D. Gerwin, "Manufacturing flexibility: a strategic perspective," *Management science*, vol. 39, no. 4, pp. 395–410, 1993.

[43] M. Koc, J. Ni, J. Lee, and P. Bandyopadhyay, "Introduction to e-manufacturing." 2005.

[44] R. F. Babiceanu and R. Seker, "Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Computers in industry*, vol. 81, pp. 128–137, 2016.

[45] S. C. Doltsinis, S. Ratchev, and N. Lohse, "A framework for performance measurement during production ramp-up of assembly stations," *European journal of operational research*, vol. 229, no. 1, pp. 85–94, 2013.

[46] E. A. Locke and G. P. Latham, "Building a practically useful theory of goal setting and task motivation: A 35-year odyssey." *American psychologist*, vol. 57, no. 9, p. 705, 2002.

[47] T. C. Stansfield and C. O. Longenecker, "The effects of goal setting and feedback on manufacturing productivity: a field experiment," *International Journal of Productivity and Performance Management*, vol. 55, no. 3/4, pp. 346–358, 2006.

[48] T. P. Wright, "Factors affecting the cost of airplanes," *Journal of the aeronautical sciences*, vol. 3, no. 4, pp. 122–128, 1936.

[49] A. B. Badiru, "Computational survey of univariate and multivariate learning curve models," *IEEE transactions on Engineering Management*, vol. 39, no. 2, pp. 176–188, 2002.

[50] G. Fioretti, "The organizational learning curve," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1375–1384, 2007.

[51] C. J. Teplitz, "The learning curve deskbook: A reference guide to theory, calculations, and applications," *(No Title)*, 1991.

[52] L. E. Yelle, "The learning curve: Historical review and comprehensive survey," *Decision sciences*, vol. 10, no. 2, pp. 302–328, 1979.

[53] M. Y. Jaber and M. Bonney, "Lot sizing with learning and forgetting in set-ups and in product quality," *International Journal of Production Economics*, vol. 83, no. 1, pp. 95–111, 2003.

[54] A. Heathcote, S. Brown, and D. J. Mewhort, "The power law repealed: The case for

an exponential law of practice," *Psychonomic bulletin & review*, vol. 7, no. 2, pp. 185–207, 2000.

[55] C. H. Glock, E. H. Grosse, M. Y. Jaber, and T. L. Smunt, "Applications of learning curves in production and operations management: A systematic literature review," *Computers & Industrial Engineering*, vol. 131, pp. 422–441, 2019.

[56] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[57] M. Y. Jaber and M. Khan, "Managing yield by lot splitting in a serial production line with learning, rework and scrap," *International Journal of Production Economics*, vol. 124, no. 1, pp. 32–39, 2010.

[58] M. Y. Jaber and A. L. Guiffrida, "Learning curves for processes generating defects requiring reworks," *European Journal of Operational Research*, vol. 159, no. 3, pp. 663–672, 2004.

[59] A. Jeang, "Learning curves for quality and productivity," *International Journal of Quality & Reliability Management*, vol. 32, no. 8, pp. 815–829, 2015.

[60] T. Disselhoff and R. J. Martin, "Data-driven early quality prediction in high pressure die casting," *International Journal of Metalcasting*, pp. 1–11, 2025.

[61] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of manufacturing systems*, vol. 48, pp. 144–156, 2018.

[62] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata, "Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook," *IEEE access*, vol. 8, pp. 220 121–220 139, 2020.

[63] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & manufacturing research*, vol. 4, no. 1, pp. 23–45, 2016.

[64] A. Chatterjee, B. S. Ahmed, E. Hallin, and A. Engman, "Testing of machine learning models with limited samples: an industrial vacuum pumping application," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 1280–1290.

[65] Z. Liu, X. He, B. Huang, and D. Zhou, "A review on incremental learning-based fault diagnosis of dynamic systems," *Authorea Preprints*, 2024.

[66] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[67] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[68] S. Sankhye and G. Hu, "Machine learning methods for quality prediction in production," *Logistics*, vol. 4, no. 4, p. 35, 2020.

[69] D. C. Montgomery, *Introduction to statistical quality control*. John wiley & sons, 2020.

[70] International Organization for Standardization, "Iso 2859-1:1999 – sampling procedures for inspection by attributes — part 1: Sampling schemes indexed by acceptance quality limit (aql) for lot-by-lot inspection," ISO Standard, Geneva, Switzerland, 1999.

[71] E. von Collani, "A problem called iso 2859-1 sampling procedures for inspection by attributes-part 1," 2004.

[72] C. Brecher, S. Storms, C. Ecker, and M. Obdenbusch, "An approach to reduce commissioning and ramp-up time for multi-variant production in automated production facilities," *Procedia CIRP*, vol. 51, pp. 128–133, 2016.

[73] R. Bultó, E. Viles, and R. Mateo, "Overview of ramp-up curves: A literature review and new challenges," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 232, no. 5, pp. 755–765, 2018.

[74] P. Svensson and J. Blom, "Critical factors for production ramp-up in high technology companies: A case study at an aerospace company," 2019.

[75] S. Chakraborty, S. Adhikari, and R. Ganguli, "The role of surrogate models in the development of digital twins of dynamic systems," *arXiv preprint arXiv:2001.09292*, 2020.

[76] T. Viering and M. Loog, "The shape of learning curves: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7799–7819, 2022.

[77] N. Patwardhan, M. Ingalhalikar, and R. Walambe, "Aria: utilizing richard's curve

for controlling the non-monotonicity of the activation function in deep neural nets," *arXiv preprint arXiv:1805.08878*, 2018.

[78] W. G. Cochran, *Sampling techniques*. john wiley & sons, 1977.

[79] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934.

[80] A. Agresti and M. Kateri, "Categorical data analysis," in *International encyclopedia of statistical science*. Springer, 2011, pp. 206–208.

[81] A. Agresti and B. A. Coull, "Approximate is better than "exact" for interval estimation of binomial proportions," *The American Statistician*, vol. 52, no. 2, pp. 119–126, 1998.

[82] T. P. Ryan, *Statistical methods for quality improvement*. John Wiley & Sons, 2011.

[83] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*. Mcgraw-hill New York, 2007, vol. 3.

[84] P. D. Welch, "The statistical analysis of simulation results," *Computer performance modeling handbook*, 1983.

[85] K. P. White, M. J. Cobb, and S. C. Spratt, "A comparison of five steady-state truncation heuristics for simulation," in *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*, vol. 1. IEEE, 2000, pp. 755–760.

[86] P. S. Mahajan and R. G. Ingalls, "Evaluation of methods used to detect warm-up period in steady state simulation," in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 1. IEEE, 2004.

[87] P. E. Hart, D. G. Stork, and R. Duda, *Pattern classification*. Wiley Hoboken, 2001.

[88] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine learning*, vol. 77, no. 1, pp. 103–123, 2009.

[89] D. Hoiem, T. Gupta, Z. Li, and M. Shlapentokh-Rothman, "Learning curves for analysis of deep networks," in *International conference on machine learning*. PMLR, 2021, pp. 4287–4296.

[90] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," *arXiv preprint arXiv:1712.00409*, 2017.

[91] J. Reason, *Human error*. Cambridge university press, 1990.

[92] H. ERIK, "Human reliability assessment in context," *Nuclear engineering and technology*, vol. 37, no. 2, pp. 159–166, 2005.

[93] D. A. Nembhard and M. V. Uzumeri, "Experiential learning and forgetting for manual and cognitive tasks," *International journal of industrial ergonomics*, vol. 25, no. 4, pp. 315–326, 2000.

[94] M. Y. Jaber and C. H. Glock, "A learning curve for tasks with cognitive and motor elements," *Computers & Industrial Engineering*, vol. 64, no. 3, pp. 866–871, 2013.

[95] A. Katona, "Validation of risk-based quality control techniques: a case study from the automotive industry," *Journal of Applied Statistics*, vol. 49, no. 12, pp. 3236–3255, 2022.

[96] M. Salehian, A. Haghani, and T. Jeinsch, "Application of hidden markov models for fault detection in automotive engines," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 767–771, 2022.

[97] J. Pereira, F. J. Silva, J. C. Sá, and J. Bastos, "Reducing the scrap generation by continuous improvement: A case study in the manufacture of components for the automotive industry," in *European Lean Educator Conference*. Springer, 2019, pp. 231–240.

[98] Fujikura Automotive Europe SAU, "Supplier quality agreement," 2022, edition August 2019, Revision October 2022.

[99] P. Pereira, J. Seghatchian, B. Caldeira, P. Santos, R. Castro, T. Fernandes, S. Xavier, G. de Sousa, and J. P. d. A. e Sousa, "Sampling methods to the statistical control of the production of blood components," *Transfusion and Apheresis Science*, vol. 56, no. 6, pp. 914–919, 2017.

A | Tecnomatix Model Architecture

A.1. Logical Architecture of the AI Decision Module

The simulation model has been developed in Siemens Tecnomatix Plant Simulation and represents a discrete-event inspection system integrating a learning-enabled decision module.

The architecture follows a two-regime structure:

1. Conservative deterministic regime (pre-activation phase)
2. Learning-enabled adaptive regime (post-activation phase)

Figure A.1 illustrates the internal logic of the AI inspection module implemented at the quality inspection station. The prediction signal is always generated for every inspected part. However, before the activation threshold $VStart_{used}$ is reached, the decision is forced into a conservative full-cycle regime and learning-based stochastic refinement is disabled. After activation, the deterministic base decision is always computed. If learning is enabled (UseAI = true), stochastic flips are applied according to the volume-dependent error probabilities $\alpha(v)$ and $\beta(v)$. If learning is disabled, the system operates using the deterministic rule only.

The activation threshold governs when learning-based refinement is allowed to influence operational decisions.

A.2. Activation Logic

The operational activation threshold is defined as:

- VStart: statistically derived minimum volume ensuring target reliability.
- $VStart_{used} = VStart + VStart_{shift}$: operational activation threshold used for sce-

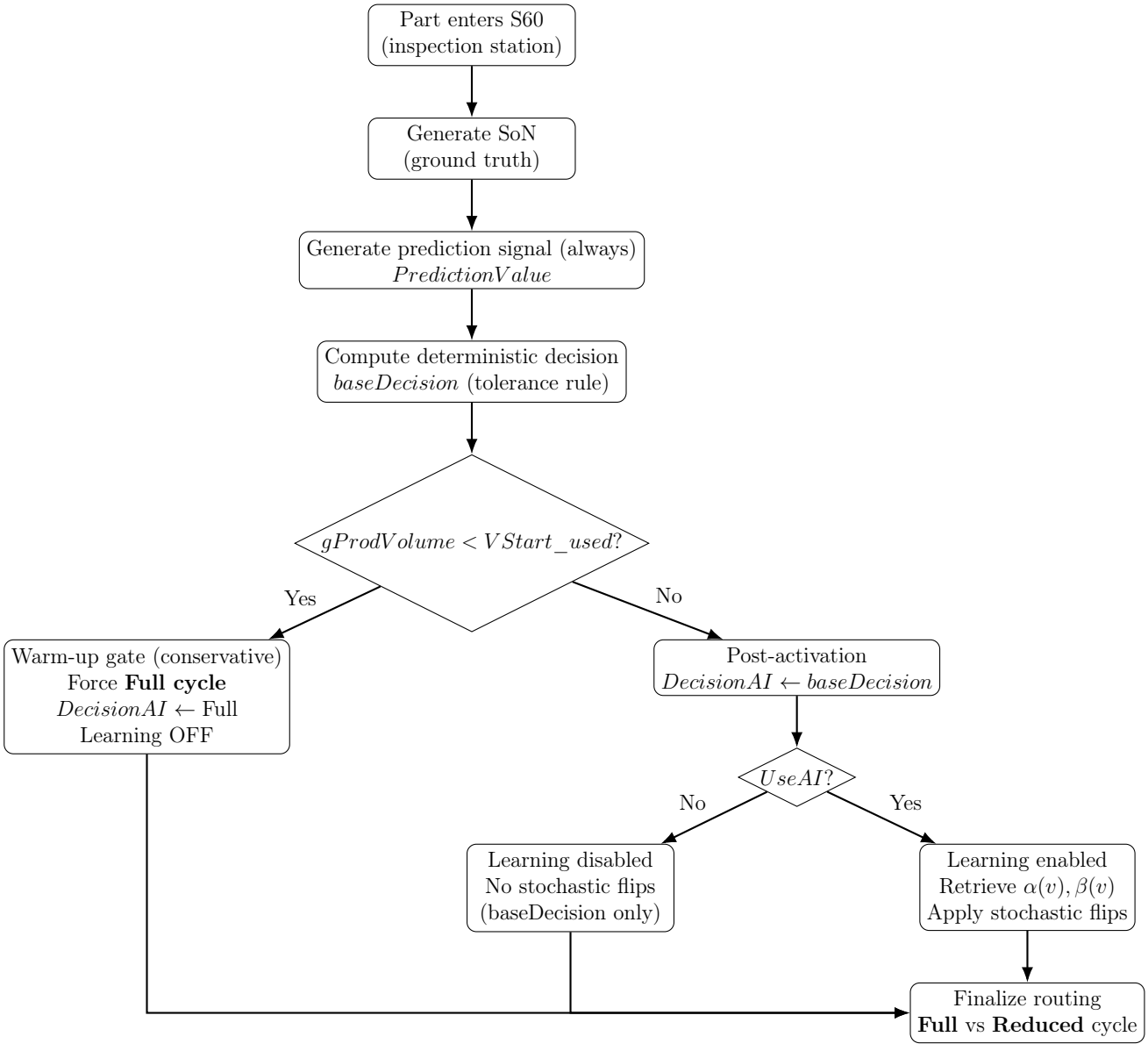

Figure A.1: Logical architecture of the AI decision module.

nario comparison.

The model therefore distinguishes between:

- Statistical reliability threshold
- Operational activation timing

This separation allows comparison of early and late activation policies without modifying structural learning dynamics

A.3. Stochastic Refinement Mechanism

After activation, error probabilities evolve parametrically with cumulative production volume.

The refinement mechanism operates as follows:

- For OK parts (Type I error), a Reduced → Full flip occurs with probability $\alpha(v)$.
- For NOK parts (Type II error), a Full → Reduced flip occurs with probability $\beta(v)$.

The volume index v is discretized into batches of size B, ensuring consistency with the batch-based aggregation used in the simulation analysis. Learning-dependent error probabilities are retrieved from the performance table and bounded within valid index ranges to ensure numerical stability.

A.4. SimTalk Implementation

The following snippet shows the core post-activation refinement logic:

Listing A.1: Post-activation stochastic refinement logic (SimTalk)

```
@.DecisionAI := baseDecision

if UseAI
   IntegratePredictionModel.Value := true

   var B : integer := 350
   var v0 : integer := VStart

   var r : integer := 1 + floor((Volume - v0) / B)
   r := max(1, min(r, .Model.tblPerformance.yDim))
```

```
   var alpha_v : real := .Model.tblPerformance["Theoretical Alpha", r]
   var beta_v : real := .Model.tblPerformance["Theoretical Beta", r]

   if mu.SoN = "OK"
      if (not baseDecision) and (z_uniform(0,1) < alpha_v)
         @.DecisionAI := true
      end
   else
      if baseDecision and (z_uniform(0,1) < beta_v)
         @.DecisionAI := false
      end
   end
else
   IntegratePredictionModel.Value := false
end
```

B | MATLAB Implementation

B.1. Simulation controller

A MATLAB App was developed as a simulation controller to ensure repeatable execution of Monte Carlo runs and consistent data export. The app connects to Tecnomatix Plant Simulation via COM API, sets scenario parameters (e.g., activation thresholds and learning-curve parameters), launches batch simulations and collects outputs (e.g., throughput traces, FPR/FNR summaries, cumulative production, etc.) into structured tables for analysis.

Figure B.1 shows the user interface used to configure runs and learning parameters and to monitor execution status.

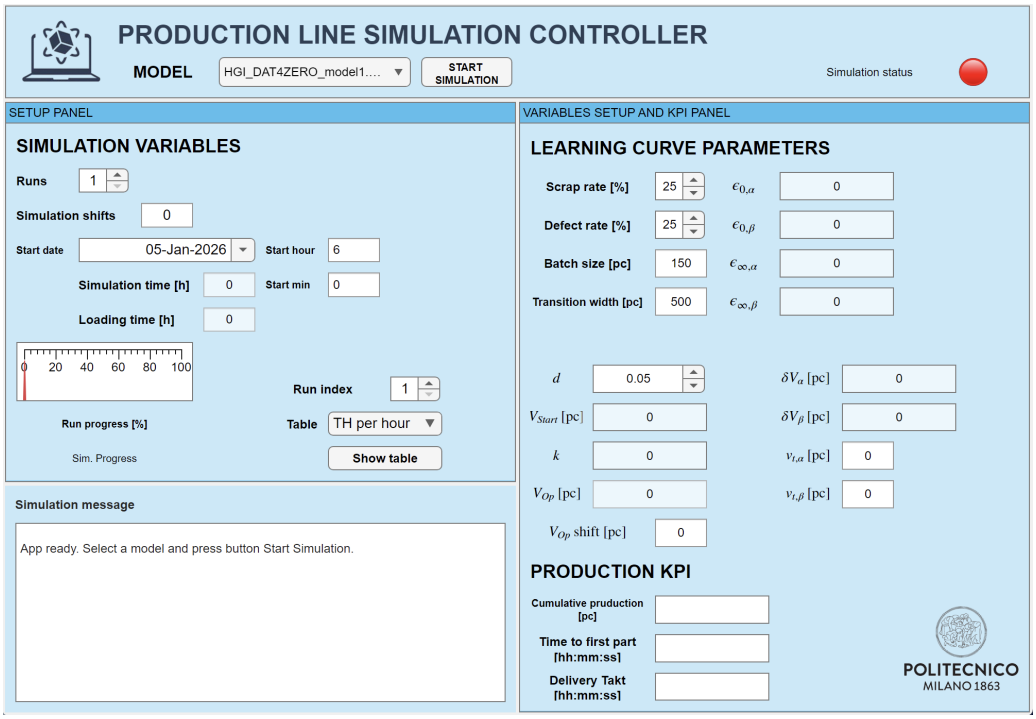

Figure B.1: MATLAB simulation controller app used to run Tecnomatix Plant Simulation.

B.2. Extended Scenario Exploration Table

Table B.1 reports the full scenario matrix explored during the simulation campaign. In total, multiple combinations of structural and operational parameters were evaluated before selecting representative scenarios for detailed analysis.

Scenario ID	**Analysis Type**	**Defect Rate**	V_{start} [pc]	B	**Notes**
S00	Baseline	25%	1540	350	Reference configuration
S01–S03	OFAT (d)	25%	varied	350	Activation shift study
S04–S07	OFAT (r)	20–25%	varied	350	Defect rate sensitivity
S08–S09	OFAT (W)	25%	1540	350	Risk weighting variations
S10–S13	OFAT (vt)	25%	1540	350	Transition window shifts
S14–S17	2x2 (rxd)	20–25%	varied	350	Combined rate variations
S18–S26	2x2 (d×B)	25%	varied	100–600	Batch size sensitivity
S27–S30	2x2 (B×W)	25%	1540	100–600	Joint batch and risk tests

Table B.1: Summary of extended scenario exploration space.

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
$\boldsymbol{v}$	Cumulative first-pass inspection volume	parts
$\boldsymbol{V_{start}^{op}}$	Operational AI learning-layer activation threshold	parts
$\boldsymbol{V_{start}}$	Statistical activation threshold	parts
$\boldsymbol{FPR_{sys}(v)}$	System-level false positive rate	–
$\boldsymbol{FNR_{sys}(v)}$	System-level false negative rate	–
$\boldsymbol{\alpha(v)}$	False positive learning component	–
$\boldsymbol{\beta(v)}$	False negative learning component	–
$\boldsymbol{\varepsilon_{0,\alpha}}$	Initial false positive error level	–
$\boldsymbol{\varepsilon_{\infty,\alpha}}$	Asymptotic false positive error level	–
$\boldsymbol{\varepsilon_{0,\beta}}$	Initial false negative error level	–
$\boldsymbol{\varepsilon_{\infty,\beta}}$	Asymptotic false negative error level	–
$\boldsymbol{k_\alpha}$	Logistic slope for false positive dynamics	–
$\boldsymbol{k_\beta}$	Logistic slope for false negative dynamics	–
$\boldsymbol{v_f}$	Logistic inflection point	parts
$\boldsymbol{\Delta V_\alpha}$	Horizontal shift for false positive curve	parts
$\boldsymbol{\Delta V_\beta}$	Horizontal shift for false negative curve	parts
$\boldsymbol{W}$	Logistic transition width	parts
$\boldsymbol{B}$	Batch size	parts
$\boldsymbol{c_\alpha}$	Weight for false positives	–
$\boldsymbol{c_\beta}$	Weight for false negatives	–
$\boldsymbol{\lambda}$	Risk weighting coefficient	–
$\boldsymbol{R_\lambda(v)}$	Composite risk function	–
$\boldsymbol{AUC_R(v)}$	Area under risk curve	risk·parts
$\boldsymbol{AUC_{R,\mathrm{end}}}$	Final cumulative risk	risk·parts

Variable	Description	SI unit
$\boldsymbol{\Delta AUC_{R,\mathrm{end}}}$	Cumulative risk difference	risk·parts
$\boldsymbol{TH(t)}$	Throughput at time t	parts/h
$\boldsymbol{TH_{ss}}$	Steady-state throughput	parts/h
$\boldsymbol{t_{ss}}$	Steady-state onset time	h

Acknowledgements

I would like to express my gratitude to all the people who have been by my side throughout this journey.

The path that led me here was neither straightforward nor free of difficulties. In particular, during the first year of my master's degree, I went through a very challenging period. It was a time when I questioned my own resilience and when the road suddenly seemed steeper than I had imagined. It was during that phase that I realized how important it is not to face difficulties alone.

I owe my family more than I can express in words. I am deeply grateful to them for never stopping believing in me, even when I struggled to believe in myself. In particular, thank you to Mami and Papo: your constant support, your trust and your ability to always be there for me were the anchor that allowed me to rise again and rediscover my direction. Without you, this achievement would not have been possible and would not have had the same meaning.

A small but very important thank you also goes to my little cousin Cele. During the most intense periods of study, her spontaneity and ability to make me smile provided a precious balance. Even the shortest breaks, when shared with someone who knows how to bring happiness effortlessly, made all the difference.

A special thought also goes to those who, at an important stage of this journey, stood by me, sharing my struggles, my goals and a fundamental part of my life. Thank you, Gloria. Even though our paths have now diverged, the support and love I received during those moments were real and meaningful and I will always be grateful for it.

I would also like to thank my thesis advisor, Professor Maria Chiara Magnanini, for her availability and the attention she dedicated to me throughout this project. Her guidance and support have been fundamental to the success of this journey.

Finally, I would like to thank myself as well. For never giving up on this journey, even in the most difficult moments. For finding the strength to get back on my feet when it seemed easier to stop and for continuing to believe in what I wanted to become.