POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Development of a framework for a controller of an injection molding machine based on AI

TESI MAGISTRALE IN AUTOMATION AND CONTROL ENGINEERING – ING-INF/04 - AUTOMATICA

AUTHOR: DAVID PEREZ PERALTA

ADVISOR: MATTEO CORNO

ACADEMIC YEAR: 2022-2023

1. Introduction

Plastic is widely used worldwide due to its low-cost production. Many techniques exist, but some are more widely used. Injection molding (IM), accounts for over 30% of global plastic output [1]. As every other process, IM can make faulty parts, which increase costs. To tackle this and meet changing market needs, digitalization and automation are essential, which at the same time generate lots of data from plant sensors. To process them, many industries currently use machine learning (ML) methods to manage quality and improve performance, and IM machines are no different [2].

To efficiently manage ML models and adapt them to changing data trends is a difficult task and automated pipelines are used to solve the problem. They automate various steps, from data gathering to model deployment, saving time and making the process more efficient.

Regarding controllers for IM machines, open and closed loop options have been available for more than 50 years [3], but do not usually use ML for part quality assessment. As ML gains popularity, companies are looking at solutions combining them with IM machines to control part quality. However, this is a complex task that needs more research.

In summary, this work has three main goals: creating a ML model to predict IM part quality, setting up an automated pipeline to manage the model, and to design a real-time controller for better part quality. Achieving these three tasks in a real manufacturing system will be a big accomplishment in injection molding.

2. Literature review

2.1. Plastic IM machines

Injection molding involves injecting molten material into a mold and once it has cooled and solidified, the final product is ejected. The process is very versatile and used to produce pieces like glasses, elastomers, thermoplastics, and thermosetting polymers.
Machinery used in injection molding typically comprises two units: an injection unit for material

introduction and a clamping unit to hold the mold closed during injection.

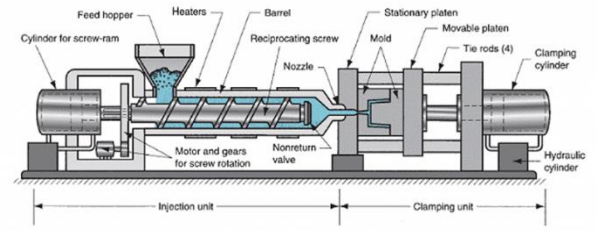

Figure 2-1. Injection molding machine diagram.

To produce a piece with these machines, a specific mold needs to be created first. Then, plastic pellets are fed continuously to the barrel, which heats them up until they are melted. They are then injected into the mold cavity with pressure and ejected after it solidifies. Many factors affect a pieces' result, from mold design to IM machine properties, and this work will be focused on predicting and improving the final quality.

2.2. IM classical control approaches

Injection molding undergoes complex changes in polymer properties due to rapid pressure and temperature variations, making quality prediction challenging without advanced software. Continuous monitoring is crucial to maintain quality standards, given the impact of slight machine parameter variations (e.g., hydraulic power, temperature, humidity) on part quality. Extensive research aims for adaptive and automatic quality control. Variables in injection molding fall into three groups: machine, process, and quality. Table 2-1 shoes some examples.

Machine variables	Process variables	Quality variables
Barrel temperature	Melt temperature	Part weight
Maximum injection pressure	Melt pressure	Part thickness
Injection speed	Maximum shear stress	Sink marks
Clamp opening/ closing time	Heat and cooling dissipation rate	Other aesthetic defects

Table 2-1. Example variables of the three levels of IM.

Machine variables are well-controlled with built-in controllers like PID and PLC. Process variables depend on conditions, materials, and mold configurations, influencing quality variables. Finally, quality variable control is the primary focus, but understanding their intricate relationships remains a challenge [4].

Direct and online part quality control is challenging due to IM's rapid nature, and difficulty to measure part quality quantitatively based on qualitative features. Classic approaches rely on observers to mimic closed loop controllers. Some studies proposed thickness control using simulation programs or empirical models [5]. Conditional logic, like fuzzy logic, is also used as a controller [6].

More research is needed for closed-loop quality control, particularly addressing the rapid process changes that current sensor technologies struggle to capture [4].

2.3. ML algorithms and controllers for IM machines

Ensuring high-quality parts in injection molding is challenging due to the complex interconnected variables involved in the process. Machine learning and neural networks (NN) excel in uncovering intricate patterns in such scenarios.

ML algorithms can be divided into three categories: reinforcement, supervised, and unsupervised learning. Supervised learning uses labeled data to establish input-output relations, while unsupervised learning extracts patterns from unlabeled datasets.

Quality assessment in IM centers on three main indicators: surface properties (sink marks, roughness), dimensions and weight, and physical properties (mechanical, optical, electrical). Weight is particularly important as it inversely correlates with part quality [7].

A 2022 study used ML, specifically k-nearest neighbor (kNN), naïve classifier, decision trees, and linear discriminant analysis, to predict part quality based on weight. Results showed high average performance, even with minimal data [8].

Oversampling is a commonly used technique also to address data scarcity, as the number of defective parts is much lower than correct parts.

Autoencoders, an unsupervised neural network architecture, learn lower-dimensional

representations of high-dimensional data, thus can be efficient also in predicting part quality.
Comparing ML models, autoencoders outperformed regression and tree-based models in predicting part quality. F1-scores for autoencoders were notably higher (0.97), while regression and support vector machines (SVM) scored lower (0.15-0.17), and tree-based models fell in the range of 0.70.
Another study divided pieces into four groups (3 OK and 1 NOT OK) and used ML to alert users when quality deviated from optimal. F1-scores ranged from 0.89 to 0.98 with different algorithms [23].
In summary, implementing automatic control techniques for IM machines is challenging but feasible with additional variables like weight, surface inspection, and part dimensions.

2.4. Automated pipeline

In the production phase of an ML model, maintenance remains essential. To enhance efficiency and scalability while reducing time and effort, automated ML pipelines come into play.
Automated pipelines automate various sequential steps in machine learning, from data ingestion to model training and deployment. They are indispensable for automating repetitive tasks, ensuring reproducibility, and streamlining model development and maintenance. In data science projects of all scopes, having a comprehensive end-to-end pipeline is vital for efficiency and organization.
Contrasting manual pipelines, where everything is typically in a few files, automated pipelines break down the process into manageable components like data extraction, model training, validation, and re-training. This engineering approach facilitates tracking changes, repeating processes, and updates as complexity grows over time.
Machine learning pipelines offer flexibility and transform the workflow into a reproducible, organized, and easily manageable process.
The main steps of a ML pipeline are the following:

1. Data ingestion.
2. Data preprocessing.
3. Data validation.
4. Feature engineering.
5. Hyperparameter tuning.
6. Model training.
7. Model evaluation.
8. Model deployment.
9. Model monitoring.

Orchestrators are the tools or frameworks that help manage and organize the various steps involved in the creation and implementation of a ML pipeline. In this work *Apache Airflow* will be the orchestrator used, for its simple usage and scalability properties.

3. Overview of the proposed approach

The approach of this work involves creating a framework for an IM machine controller with three main components: the ML model, the pipeline, and the controller.

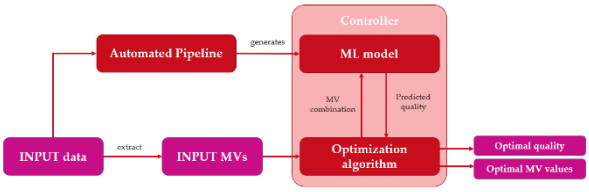

Figure 3-1. Complete proposed approach.

As seen in Figure 3-1, the proposed approach can be divided into two parts. First, the ML model is developed by feeding input data into the pipeline, which outputs the best-fit ML model.
The second part is the controller, which receives manipulated input variables. It iteratively provides MV combinations to the ML model to predict part quality and outputs optimized MV values. Although the two parts are separate, the pipeline development follows ML model architecture definition.
Regarding the data used, it comes from two datasets with similar structures but varying variables. The first dataset has 84 variables and about 12,000 samples from 2020 to 2021. The second dataset includes 37 variables and over 330,000 samples from 2021 to 2023, with some overlapping variables.
Both datasets have a binary controlled variable: 0 for defect-free (OK) parts and 1 for defective (NOK) parts. Due to the rarity of NOK samples (about 2%), designing a ML model for this class imbalance is a challenge, known as One Class Classification (OCC).
The technical implementation predominantly uses Python due to its popularity and library support, especially for neural networks. Tensorflow is chosen for ML aspects and Apache Airflow as the

orchestrator. SQL databases store metadata, and Docker containerization ensures consistency and portability. Visual Studio Code is the development environment, integrated with GIT for version tracking (GitHub), and the controller is entirely implemented in Python for code and algorithmic simplicity.

4. Initial dataset analysis

The dataset's objective is to perform a preliminary data analysis and create an initial process model. A larger dataset will be generated in the future for the final ML model integrated with the control algorithm.

The controlled variable is unique and binary (OK or NOK piece). The process parameters include temperatures, pressures, times, and velocities. Some variables are categorized by machine parts and zones. Information Gain (IG) algorithm, which calculates how much information the output obtains from each input, was selected as a preprocessing step. The 12 variables containing most information were selected to train the model. Less than 4% of the samples were NOK, which poses challenges for model training. The problem is known as One Class Classification and is used in other situations like scam filtering. An extensive OCC bibliographic review was conducted, including neural network approaches like weighting differently each class or oversampling. Other algorithms like Support Vector Machines (SVM), isolation forest and Local Outlier Factor (LOF) were also investigated.

The results of different algorithms are displayed in Table 4-1, where best threshold refers to the value separating the classes in the model prediction.

Model	Test			
	F1	PRC avg	ROC AUC	Best threshold
SVM	0	0.04	0.523	-
Iso. Forest	0.007	0.05	0.597	-
LOF	0.109	0.04	0.592	-
Normal	0.288	0.19	0.737	0.05
Weighted	0.316	0.17	0.8	0.56
Resampled	0.317	0.26	0.872	0.76

Table 4-1. Different models' results for the test set.

In summary, the analysis of results indicates that NN-based models outperform other algorithms in addressing imbalanced datasets. NNs consistently yield better results across all metrics, with the most significant improvement seen in the F1 score. While other algorithms can have a minimum F1 score of zero, NNs achieve a minimum of 0.213 and a maximum of 0.317. This suggests that further algorithm development is unnecessary, and the focus will shift to enhancing models and exploring alternative machine learning techniques.

5. Final dataset analysis

A different approach was taken with this new dataset compared to the initial one. Meetings were held with the data provider to understand it better. The new dataset introduced a *decMold* column indicating the mold type used, which could lead to variations in other variables. Some previously selected variables were missing, requiring a fresh analysis. However, insights from the initial analysis remained valuable, and techniques/results still applied.

In this dataset, the focus was on the impact of the different molds used to collect data, because it led to variations in operating points for several variables.

Regarding control variables, after meetings and information gain analysis, it was determined that only four variables were sufficient for a meaningful process description: *tmpBarrel1Zone1*, *tmpMoldZone1*, *prsInjectionHyd1*, and *spdInjection1*. This reduced the previous set of 12 variables to just these four.

A different model was trained for each mold type in two steps. First, it was trained with all molds except the desired one, and then it was retrained only with the desired data samples. This proved better results, shown in Table 4-1 for mold type 2535.

	Train	Validation	Test
F1-Score	0.970	0.976	0.974

Table 5-1. F1-scores for final model for mold 2535.

The metric values reached are impressive and reaffirm that the chosen approach is the way to go. Similar values were achieved for the rest of models for the other molds.

6. Automated pipeline

With the ML model's architecture defined, the automated pipeline to manage it can be built. The steps used are the following:

1. Get and store data
2. Generate statistics and schema
3. Validate data
4. Preprocess data
5. Tune model parameters
6. Train best model
7. Evaluate models
8. Push model

Apache Airflow was the selected orchestrator, where each pipeline execution is a "run."

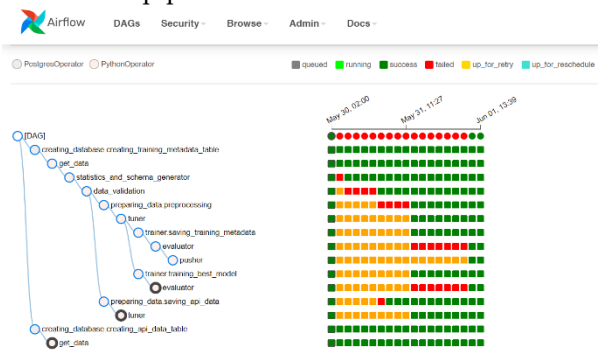

Figure 6-1. Tree diagram of several pipeline runs.

Figure 6-1 shows multiple runs of the pipeline, with rows as tasks and columns as runs. Colors highlight success or issues. In this case, the first and last two runs succeeded, and the rest of the runs mark the pipeline's evolution from its initial working state through updates.

7. Controller design and tuning

After reviewing techniques and available data, an open-loop controller was chosen due to real-time constraints. Open-loop controllers are less robust than closed-loop ones but were the only viable option because real-time feedback on part quality was unavailable.

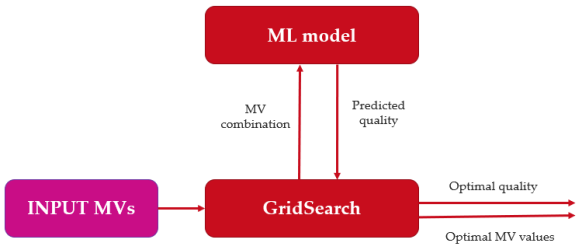

Figure 7-1. Diagram of open loop controller.

The open-loop controller uses the ML model to predict part quality based on machine parameters, and employs the *GridSearch* technique for hyperparameter tuning, evaluating all parameter combinations to find the best quality.

Here's the data flow: machine input parameters are inputted into the ML model, which predicts quality. *GridSearch* uses this prediction as an initial guess for optimal quality and searches for better combinations, considering variable changes per step and search depth. The final output is the optimal quality and corresponding machine parameters suggested to the operator.

The controller was implemented in Python, but it didn't perform as expected. The machine parameter values changed, but the predicted quality remained constant instead of approaching zero (indicating an OK piece). Despite numerous trials and thorough inspection, the root cause of the issue remained unidentified.

8. Conclusions

Injection molding is a complex, interconnected process for producing quality plastic parts. This work has addressed part quality control challenges by creating an automated pipeline that provides ML model replicating the IM process, which is then used for a part quality controller.

Dealing with an imbalanced dataset (less than 2% defective parts) posed challenges, however, the ML model successfully predicted quality, achieving an F1-score of 0.97 after dividing the data by mold types and training separate models.

Initial dataset results were lower due to data quality and using a single model for all mold types. Various techniques and algorithms were tested, with resampled and weighted ML models yielding the best F1-scores (0.317 and 0.316, respectively).

The automated pipeline, comprising eight steps, worked correctly, but real-world testing with a deployed ML model remains as future work.

Initially, a closed-loop controller was planned, but due to dataset constraints and quality inspection delays, an open-loop controller was implemented with unexpected results.

In summary, the framework of a ML model, AI pipeline, and controller is promising but requires additional time and effort. This solution can be extended to other machines and industries, providing efficiency, feedback, robustness, and

easier management at the expense of an initial data collection investment.

8.1. Limitations of the study

Limitations include an imbalanced dataset, with data augmentation offering imperfect solutions due to the difficulty of obtaining enough real defective samples.
Additionally, AI-driven production model development demands incredible data amounts, and any changes in machines, products, or the environment can render previous data obsolete, impacting model performance.
The dataset's binary output (OK-NOK) and delayed quality inspection constrained controller techniques that could be used. Implementing a closed-loop controller became increasingly complex and time-restricted, resulting in modifications to the approach.

8.2. Future work

The ML model was integrated successfully with the automated pipeline, but the controller couldn't be added to the equation, nor tested on a real machine, making all presented results theoretical. Therefore, real-world validation is crucial.
Moreover, continuous pipeline operation over a prolonged period is necessary to ensure proper adaptation to new data trends. Achieving this could be challenging as machine conditions ideally should change minimally to maintain part quality. Relocating the machine or introducing neighboring systems might pose environmental and variable changes.
Once a complete system, including the ML model, pipeline, and controller, is validated on a shop floor, it could be tested on machines in other industries with a new model. While demanding, such an effort could pave the way for future industry applications.

References

[1] M. Schubert, S. Perfetto, A. Dafnis, D. Mayer, H. Atzrodt and K.-U. Schröder, "Multifunctional load carrying lightweight structures for space design," in *Deutscher Luft und Raumfahrtkongress*, Bonn, Germany, 2017.

[2] M. Bertolini, D. Mezzogori, M. Neroni and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," *Expert. Syst. Appl.*, vol. 175, no. 114820, 2021.

[3] Andy, Routsis Training, 28 9 2009. [Online]. Available: https://routsis.blog/2009/09/open_loop_vs_closed_loop_controls/.

[4] "A Review of Current Developments in Process and Quality Control for Injection Molding," *Advances in Polymer Technology*, vol. 24, no. 3, pp. 165-182, 2005.

[5] P. J. Wang and K. K. Wang, in *ANTEC'01*, 2001.

[6] A. R. Agrawal, I. O. Pandelidis and Pecht, *M. Polym Eng Sci*, vol. 27, no. 18, pp. 1345-1357, 1987.

[7] Xcentric Mold & Engineering, "Xcentric Mold & Engineering," Xcentric Mold & Engineering, [Online]. Available: https://xcentricmold.com/injection-molding-process/.

List of figures

List of tables