POLITECNICO DI TORINO

Master's degree in Mechanical Engineering

Master's degree Thesis

Modelling and simulation of a glass production system with Machine Learning algorithms

Supervisor:

Prof. Massimo Sorli

Candidate:

Stefano Pappadopolo

December 2020

ABSTRACT

Modelling and simulation of a glass production system

with Machine Learning algorithms

The present study aims to model and simulate the behaviour of a flat glass annealing lehr, by means of data analysis algorithms.
In the framework of "Industry 4.0", the availability of more production data and the need to find correlations between process variables, enhances the use of Machine Learning to generate models that are able to predict a process or equipment outputs, given certain inputs.
The target of this work is to create a regression model that predicts the permanent stress of a glass sheet (output) generated in the annealing phase of a controlled cooling. The inputs considered are process variables, such as the machine's setting and other working variables as production parameters, initial and boundary conditions. The pipeline of this analysis includes a first preprocessing phase: the data are evaluated and treated in terms of features engineering, distribution, outliers and scaling. Once the dataset is prepared, it can be used for the model's training: in this phase, the Support Vector Machine (SVM) algorithm operates on a subset of samples to generate the regression model, then it is tested on another subset to validate the results. In order to improve the model accuracy, a tuning of the parameters and its Cross-validation are performed.
The purpose of the model in real production is oriented to the process control and consists of the simulation of the system's output to manage its variation and detect an eventual deviation in the quality output before the quality check is effectively done; furthermore, it can be used to simulate different equipment settings and preliminary give a higher level of confidence to the tests performed on-line.

ACKNOWLEDGEMENTS

Prima di procedere vorrei dedicare qualche parola di riconoscimento alle persone che sono state presenti in questo percorso.

Ai miei genitori ed alle mie sorelle,
che mi sono stati vicini durante questi anni molto impegnativi,
e che mi hanno dato un supporto cruciale lungo questo cammino.

Al relatore di questa Tesi, il Prof. Massimo Sorli,
che mi ha seguito nello sviluppo di questo progetto
con interesse e professionalità.

INDEX

INTRODUCTION

The present work is developed within a business context and aims to find an engineering and practical solution for issues concerning the production of flat glass. Within the framework of the "Industry 4.0", the availability of data and the non-linearity of the phenomena that characterize these processes, lead the companies operating in this field to look for systems that make easier to find correlations between process variables, in a perspective of "black box" analysis. For engineering and deeper analysis of this kind of information, it is necessary to create prediction models able to foresee the output of a certain system, under several conditions and process settings. In this study, to get this objective a Machine Learning method is used to generate a regression model and tested in real production conditions.

Case study

With the company *AGC Glass Europe,* operating in the field of flat glass for building applications, has been studied the possibility to apply the above-mentioned method in a Float glass production line.

This production process is composed of several phases, the main ones are *melting*, *forming* and *annealing*; in this phase the glass is slowly cooled down for afterwards manipulations. The purpose of this thermal treatment is to relieve the stresses applied during manufacturing and bring them to acceptable values. During this stage the glass, that flows uninterruptedly in a continuous *ribbon* along the line, changes its properties and characteristics from visco-elastic to a solid-state.

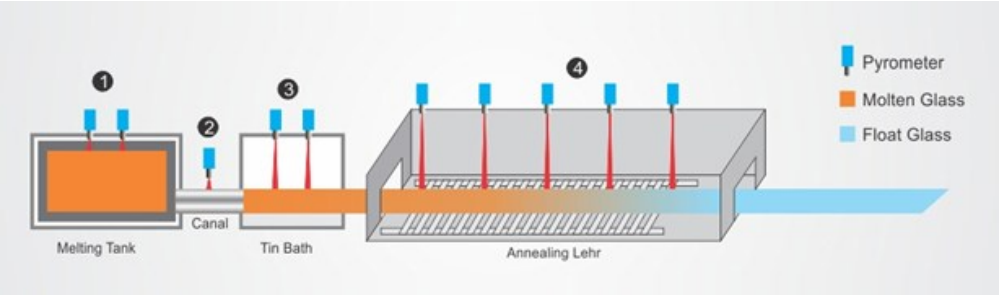

Figure 0.1: Layout of a Float glass production line. The main steps of the process are: melting(1), forming(3) and annealing(4); colours represent the physical state of the glass: liquid phase (orange) and solid phase (light blue).

Here the thermal treatment provides to the glass the strains that, after the annealing is completed, will become permanent stress; the distribution of

these tensions is crucial for quality acceptability, and they are continuously measured to ensure they stay within certain thresholds.

Working at high temperatures, the heat transfer is mainly exchanged with the glass by radiation. The heating and cooling thermal power needed to perform the desired spatial and time gradients of temperature is provided by two main devices: electrical resistances and air heat exchangers; their regulation allows to have a desired distribution of temperature and cooling rate across the glass ribbon.

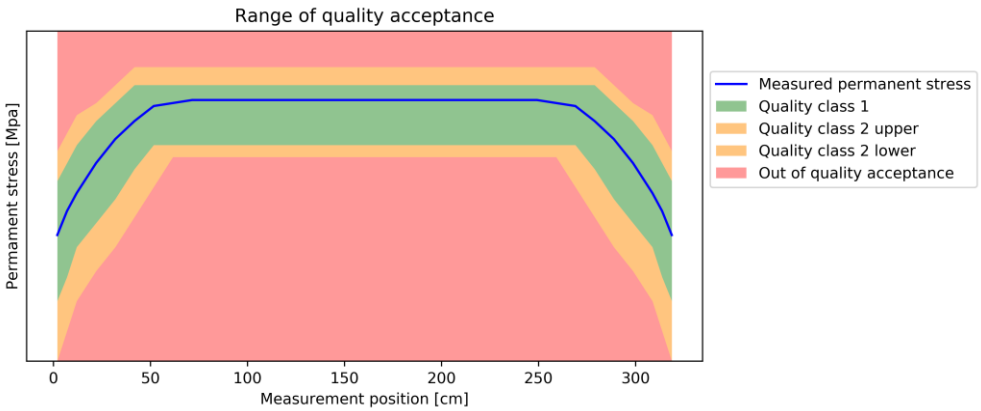

Figure 0.2: Quality acceptance levels and example of permanent stress measurement across the sheet width.

Development's framework and target

From an engineering point of view, what explained in the previous paragraph can be described globally as a system where all the phenomena and equipment are connected each other and the output of one block represents the input of the block afterwards. Very simplistically, the system considered can be represented in the following way: the system receives as input the initial condition of the glass from the forming process, the settings of the annealing lehr and other external conditions; the actuation of these input parameters and the behaviour of the glass, produce as output the permanent stress measured on the glass sheet.

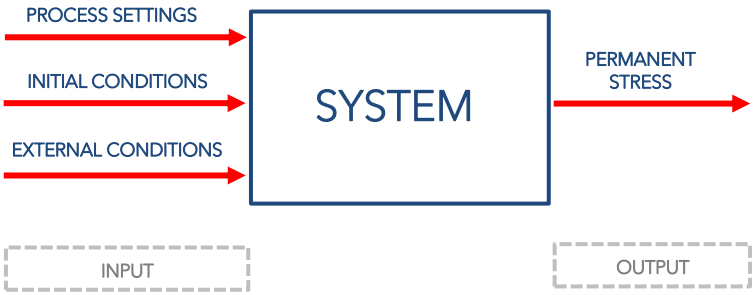

Figure 0.3: Functional block diagram of the system analyzed.

Part of the input variables depends on the operators, that continuously modify the settings to get a constant and acceptable permanent stress, other variables depend from the behaviour of the equipment and external conditions. Since this kind of system is quite difficult to be simulated analytically, due to non-linear phenomena and the complexity of its parts, it can be explained as a *black box,* where a variation of some specific parameters, generate a variation of the output.

As mentioned before, this is one of the best practice to analyze and simulate this kind of system, in order to find the correlation between input and output variables, that will be furtherly use for a deeper knowledge of the process and to improve the quality output. The methods used in this thesis to point out these correlations are based upon regression algorithms; given historical production data, the regression model can highlight the dependency of the output from each input. Thereby the Machine Learning model is provided by historical data where it can be trained and generate the regression, then is tested on new data to check and validate the accuracy of the model.

The target of this model is dual: in a real production condition, it can continuously simulate the output of the process, to spot eventual deviation in the permanent stress that would mean out-of-spec product; this action is done

by the quality check but the higher frequency of the simulation can give almost "live" results and reduce the quantity of product out of acceptance interval. Considering the production continuity and the impact that a test failure can produce, the second purpose of this model is to study the effect of new settings and conditions without testing on the production line.

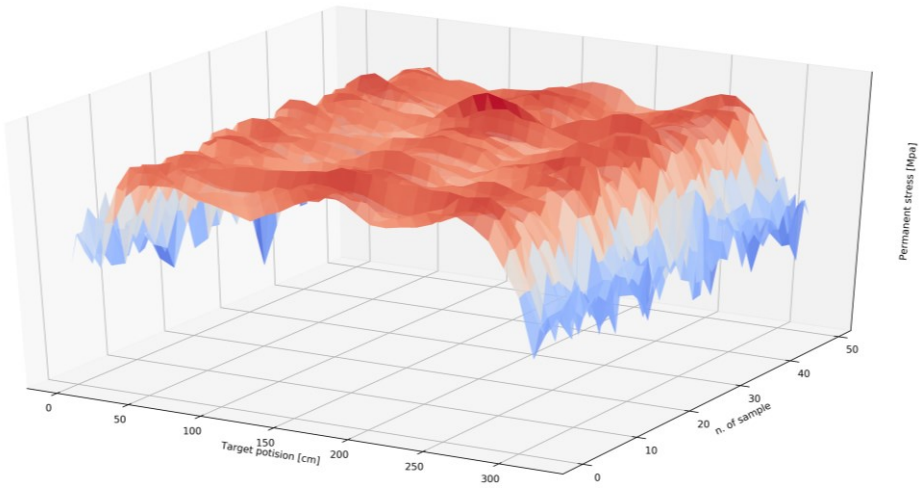

Figure 0.4: Time progression of the measured output. On the x-axis is represented the position of the output across the glass sheet, on the y-axis the sampling of the output (it can be figured on a time axis) and on the z-axis the value of the measurement.

The innovative contribution that the present work provides to this production line, fit into a policy of data analysis for improvement goals and paves the way to other studies based on similar tools; furthermore, it offers indication for the installation of more sensor for the data collection and modification of some devices, in order to enhance the model accuracy and completeness.

Research methodology

As mentioned above the solution to this problem requires tools of data analysis able to handle a large number of data samples, mathematically

manipulate them, and manage complex regression tasks to create correlations between variables. The approach used regards the family of Supervised Machine Learning algorithms for regression task. These algorithms map patterns and relationships by inferring a function from a labelled training dataset – that consists of an input object and the measured output values. Since machine learning algorithm is provided with the correct "answers" during training, the algorithm is able to "learn" how the rest of the features relate to the target, enabling to make predictions about future outcomes based on historical data, in an optimal scenario the algorithm correctly determines the output values for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in the most accurate way.

Organization of the dissertation

This report aims to explain the steps that have characterized the study of this system and the application of Machine Learning algorithms for data analysis purpose.

The following chapters will show a first part of theoretical research on the Machine Learning algorithms, about the theory of glass annealing and the technology that enact the production process. The second part will focus on the experimental analysis and the model building, through the guidelines of data analysis, with a focus on the functions used for the data pre-processing, the model selection and the parameter's tuning.

CHAPTER 1

THEORY OF MACHINE LEARNING

1.1. Introduction

In 1959, IBM published a paper in the IBM Journal of Research and Development authored by IBM's Arthur Samuel; the paper involved the use of machine learning in the game of checkers "to verify the fact that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program". Even if it was not the first publication to use the term "machine learning", Arthur Samuel is widely considered as the first person to use and define machine learning in the form we now know today. In this paper, he introduces machine learning as a sub-field of computer science that gives computers the ability to learn without being explicitly programmed. Almost sixty later, this definition remains widely accepted. Even if not directly mentioned in this definition, a key feature of machine learning is the concept of self-learning. This refers to the application of statistical modelling to detect patterns and improve performances based on experimental data and empirical information; all without direct programming commands. This is what A.S. described as the ability to learn without being explicitly programmed even if he doesn't say that machines formulate decisions with no previous programming. On the contrary, Machine Learning is based on computer programming. Instead, A.S. observed that Machine Learning performs a task when provided by data rather than direct commands. A simple example of an input command is typing "2+2" into a programming language and hitting "Enter.". This represents a direct command with a direct output. Input data, however, is different: data is given to the machine, an algorithm is selected, hyperparameters (settings) are configured and tuned, and the machine is learned to conduct its analysis. The machine proceeds to detect patterns found in the data through the process of trial and error. The mathematical model, formed from analyzing data patterns, can then be used to predict future values. This is because the machine is formulating decisions based on experience and emulating the process of human-based decision-making. "A

simple example is creating a model that detects undesired email messages. The model is trained to block emails with suspicious subject lines and body text containing three or more flagged keywords. At this stage, though, it is not yet performing machine learning. If we recall the visual representation of input command vs input data, we can see that this process consists of only two steps: Command > Action. Machine learning entails a three-step process: Data > Model > Action. Thus, to incorporate machine learning into our spam detection system, we need to switch out "command" for "data" and add "model" to produce an action (output). In this example, the data comprises sample emails and the model consists of statistical-based rules. The parameters of the model include the same keywords from our original negative list. The model is then trained and tested against the data. Once the data is fed into the model, there is a strong chance that assumptions contained in the model will lead to some inaccurate predictions. Traditional programming is highly susceptible to such cases because there is no built-in mechanism to test assumptions and modify the rules of the model. Machine learning, on the other hand, can adapt and modify assumptions through its three-step process and by reacting to errors."[1]

1.2. Machine Learning categories

Machine learning incorporates several statistical-based algorithms and choosing the right algorithm or combination of algorithms for the task is a challenge. Before going deeper into the model selection, it is important to understand the three general categories of machine learning. These three categories are supervised, unsupervised, and reinforcement.

[1] (Theobald, 2017)

1.2.1. Supervised Learning

As the first branch of machine learning, Supervised learning focuses on detect patterns by creating the relationships between variables and known outcomes and working with labelled datasets. Supervised learning works by providing the machine sample data with features (represented as "x") and the correct output value (represented as "y"). The fact that the output and feature values are known qualifies the dataset as "labelled." The algorithm then detects patterns that exist in the data and creates a model that can reproduce the same rules with new data.

After the machine deciphers the rules and patterns of the data, it generates a model: a function for producing an outcome with new data based on the rules derived from the training data. Once the model is built, it can simulate new outcomes and tested for accuracy. After the model has completed both the training and test data stages, it is ready to be applied and used in the real conditions. Examples of supervised learning algorithms include regression analysis, decision trees, k-nearest neighbours, neural networks, and support vector machines.

1.2.2. Unsupervised Learning

In the case of unsupervised learning, variables and data patterns are not classified. Instead, the machine must detect hidden patterns and create labels through the use of unsupervised learning algorithms. The k-means clustering algorithm is a typical example of unsupervised learning. This simple algorithm groups data points that are found to have similar features.

The advantage of unsupervised learning is it permits to discover patterns in the data that you were unaware existed. Clustering methods such as k-means clustering can also lead to further analysis after discrete groups have been discovered.

1.2.3. Reinforcement Learning

Reinforcement learning is the third and most advanced algorithm category in machine learning. Unlike supervised and unsupervised learning, reinforcement learning continuously improves its model by using the feedback from previous iterations. This is different from supervised and unsupervised learning, which both reach an indefinite endpoint after a model is formulated from the training and test data phases. Reinforcement learning can be complex and is typically explained through an analogy to a video game. As a player progresses through the virtual space of a game, he learns the value of various actions under different conditions and become more familiar with the play. Those learned values then inform and influence a player's subsequent behaviour and their performance immediately improves based on its learning and experience. Reinforcement learning is similar, where algorithms are set to train the model with continuous learning. In the case of self-driving vehicles, avoiding a crash will assign a positive score and in the case of chess, avoiding defeat will likewise receive a positive score.

1.3. Regression

Regression analysis is a simple supervised learning technique used to find the best trendline to describe a dataset. The goal of a regression task is to predict the value of continuous target variables **t** given the value of a multi-dimensional vector **x** of input variables. The polynomial is one example of a large class of functions called linear regression models, which have the property of being linear functions of the adjustable parameters. The simplest form of linear regression models is also linear functions of the input variables. However, it can be obtained a much more useful class of functions by taking linear combinations of a fixed set of nonlinear functions of the input variables, known as basis functions. These models are linear functions of the parameters, which gives them simple analytical properties and yet can be non-linear regarding the input variables. Given a training dataset comprising N observations **{xn},** where **n =1,..,N**, together with corresponding target values **{tn}**, the goal is to

predict the value of **t** for a new value of **x**. In the simplest approach, this can be done by directly constructing an appropriate function **y(x)** whose values for new inputs **x** constitute the predictions for the corresponding values of **t**. More generally, from a probabilistic perspective, we aim to model the predictive distribution **p(t|x)** because this expresses our uncertainty about the value of t for each value of **x**. From this conditional distribution, we can make predictions of **t**, for any new value of **x**, in such a way as to minimize the expected value of a suitably chosen loss function. Although linear models have significant limitations as practical techniques for pattern recognition, particularly for problems involving input spaces of high dimensionality, they have nice analytical properties and form the foundation for more sophisticated.

1.3.1. Linear basis function model[2]

The simplest linear model for regression is one that involves a linear combination of the input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

Where $\mathbf{x} = (x_1, \ldots, x_D)^T$. This is often simply known as linear regression. The key property of this model is that it is a linear function of the parameters $w_0, \ldots, w_D$. It is also, however, a linear function of the input variables x_i, and this imposes significant limitations on the model. We, therefore, extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

[2] (Bishop, 2006)

where $\varphi_j(\mathbf{x})$ are known as basis functions. By denoting the maximum value of the index **j** by $\mathbf{M-1}$, the total number of parameters in this model will be **M**. The parameter $\mathbf{w_0}$ allows for any fixed offset in the data and is sometimes called a bias parameter (not to be confused with 'bias' in a statistical sense). It is often convenient to define an additional dummy 'basis function' $\varphi_0(\mathbf{x})=1$ so that

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

where $\mathbf{w}=(\mathbf{w0},\ldots,\mathbf{w}_{M-1})^{\mathrm{T}}$ and $\varphi=(\varphi_0,\ldots,\varphi_{M-1})^{\mathrm{T}}$. In many practical applications of pattern recognition, we will apply some form of fixed pre-processing, or feature extraction, to the original data variables. If the original variables comprise the vector **x**, then the features can be expressed in terms of the basis functions $\{\varphi_j(\mathbf{x})\}$. By using nonlinear basis functions, we allow the function $\mathbf{y(x,w)}$ to be a nonlinear function of the input vector **x**. Functions of the previous form are called linear models, however, because this function is linear in **w**. It is this linearity in the parameters that will greatly simplify the analysis of this class of models. However, it also leads to some significant limitations. The example of polynomial regression […] is a particular example of this model in which there is a single input variable x, and the basis functions take the form of powers of **x** so that $\varphi_j(\mathbf{x})=\mathbf{x}_j$. One limitation of polynomial basis functions is that they are global functions of the input variable so that changes in one region of input space affect all other regions. This can be resolved by dividing the input space up into regions and fit a different polynomial in each region, leading to spline functions (Hastie et al., 2001). There are many other possible choices for the basis functions, for example

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

where the μ_j govern the locations of the basis functions in input space, and the parameter **s** governs their spatial scale. These are usually referred to as 'Gaussian' basis functions, although it should be noted that they are not required to have a probabilistic interpretation, and in particular the normalization coefficient is unimportant because these basis functions will be multiplied by adaptive parameters w_j. Another possibility is the sigmoidal basis function of the form

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where **σ(a)** is the logistic sigmoid function defined by

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$

Equivalently, we can use the 'tanh' function because this is related to the logistic sigmoid by **tanh(a)=2 σ(a)−1**, and so a general linear combination of logistic sigmoid functions is equivalent to a general linear combination of 'tanh' functions. These various choices of basis function are illustrated in Figure 1.1. Yet another possible choice of basis function is the Fourier basis, which leads to an expansion in sinusoidal functions. Each basis function represents a specific frequency and has an infinite spatial extent. By contrast, basis functions that are localized to finite regions of input space necessarily comprise a spectrum of different spatial frequencies. In many signal processing applications, it is of interest to consider basis functions that are localized in both space and frequency, leading to a class of functions known as wavelets. These are also defined to be mutually orthogonal, to simplify their application. Wavelets are most applicable when the input values live on a regular lattice, such as the successive time points in a temporal sequence, or the pixels in an image. Useful texts on wavelets include Ogden (1997), Mallat (1999), and Vidakovic (1999). Most of the discussion in this chapter, however, is independent of the particular choice of basis function set, and so for most

of our discussion, we shall not specify the particular form of the basis functions, except for numerical illustration. Indeed, much of our discussion will be equally applicable to the situation in which the vector φ(x) of basis functions is simply the identity **φ(x)=x**. Furthermore, to keep the notation simple, we shall focus on the case of a single target variable **t**.

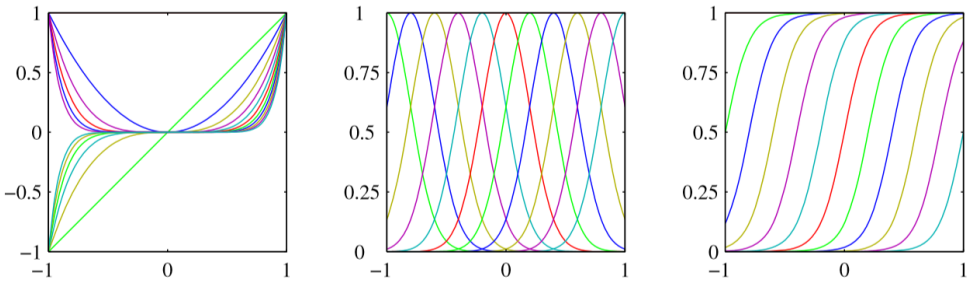

Figure 1.1: Examples of basis functions, showing polynomials on the left, Gaussians in the centre, and sigmoidal on the right.

1.3.2. Maximum likelihood and least squares

[…] consider the least-squares approach, and its relation to maximum likelihood, in more detail. As before, we assume that the target variable **t** is given by a deterministic function **y(x,w)** with additive Gaussian noise so that

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

where **ε** is a zero-mean Gaussian random variable with precision (inverse variance) **β**. Thus we can write

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

Recall that, if we assume a squared loss function, then the optimal prediction, for a new value of **x**, will be given by the conditional mean of the target variable. In the case of a Gaussian conditional distribution, the conditional mean will be simply

$$\mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t = y(\mathbf{x}, \mathbf{w}).$$

Note that the Gaussian noise assumption implies that the conditional distribution of t given **x** is unimodal, which may be inappropriate for some applications. Now consider a data set of inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with corresponding target values $t_1, \ldots, t_N$. We group the target variables $\{t_n\}$ into a column vector that we denote by **t** where the typeface is chosen to distinguish it from a single observation of a multivariate target, which would be denoted **t**. We obtain the following expression for the likelihood function, which is a function of the adjustable parameters **w** and β, in the form

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})$$

Note that in supervised learning problems such as regression (and classification), we are not seeking to model the distribution of the input variables. Thus **x** will always appear in the set of conditioning variables, and so from now on, we will drop the explicit **x** from expressions such as **p(t|x,w,β)** to keep the notation uncluttered. Taking the logarithm of the likelihood function, and making use of the standard form for the univariate Gaussian, we have

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w}) \end{aligned}$$

where the sum-of-squares error function is defined by

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2.$$

Having written down the likelihood function, we can use maximum likelihood to determine **w** and β. Consider first the maximization with respect to **w**. As observed, we see that maximization of the likelihood function under a conditional Gaussian noise distribution for a linear model is equivalent to minimizing a sum-of-squares error function given by **ED(w)**. The gradient of the log-likelihood function takes the form

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \sum_{n=1}^{N}\left\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}.$$

Setting this gradient to zero gives

$$0 = \sum_{n=1}^{N} t_n\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}}\left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\right).$$

Solving for **w** we obtain

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

Which are known as the normal equations for the least-squares problem. Here Φ is an N×M matrix, called the design matrix, whose elements are given by $\boldsymbol{\Phi}_{nj} = \varphi_j(\mathbf{x}_n)$, so that

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

The quantity

$$\boldsymbol{\Phi}^{\dagger} \equiv \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$$

is known as the Moore-Penrose pseudo-inverse of the matrix Φ (Rao and Mitra, 1971; Golub and Van Loan, 1996). It can be regarded as a generalization of the notion of matrix inverse to nonsquare matrices. Indeed, if **Φ** is square and invertible, then using the property $\mathbf{(AB)^{-1} = B^{-1}A^{-1}}$ we see that $\mathbf{\Phi^{\dagger} \equiv \Phi^{-1}}$. At this point, we can gain some insight into the role of the bias parameter $\mathbf{w_0}$. If we make the bias parameter explicit, then the error function becomes

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_n)\}^2.$$

Setting the derivative with respect to $\mathbf{w_0}$ equal to zero, and solving for $\mathbf{w_0}$, we obtain

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\overline{\phi_j}$$

where we have defined

$$\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n, \qquad \overline{\phi_j} = \frac{1}{N}\sum_{n=1}^{N}\phi_j(\mathbf{x}_n).$$

Thus the bias $\mathbf{w_0}$ compensates for the difference between the averages (over the training set) of the target values and the weighted sum of the averages of the basis function values. We can also maximize the log-likelihood function concerning the noise precision parameter β, giving

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

and so we see that the inverse of the noise precision is given by the residual variance of the target values around the regression function.

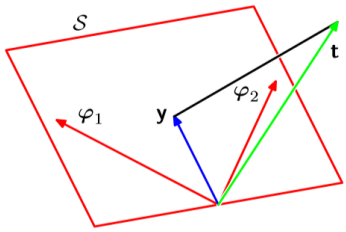

Figure 1.2: Geometrical interpretation of the least-squares solution, in an N-dimensional space whose axes are the values of $t_1,...,t_N$. The least-squares regression function is obtained by finding the orthogonal projection of the data vector t onto the subspace spanned by the basis functions $\varphi_j(x)$ in which each basis function is viewed as a vector $\boldsymbol{\phi}_j$ of length N with elements $\varphi_j(x_n)$.

1.4. Classification[3]

In the previous chapter, we explored a class of regression models having particularly simple analytical and computational properties. We now discuss an analogous class of models for solving classification problems. The goal in classification is to take an input vector x and to assign it to one of K discrete classes **Ck** where **k =1,..., K**. In the most common scenario, the classes are taken to be disjoint so that each input is assigned to one and only one class. The input space is thereby divided into decision regions whose boundaries are called decision boundaries or decision surfaces. In this chapter, we consider linear models for classification, by which we mean that the decision surfaces are linear functions of the input vector **x** and hence are defined by (D −1)-dimensional hyperplanes within the D-dimensional input space. Data

[3] (Bishop, 2006)

sets whose classes can be separated exactly by linear decision surfaces are said to be linearly separable. For regression problems, the target variable **t** was simply the vector of real numbers whose values we wish to predict. In the case of classification, there are various ways of using target values to represent class labels. For probabilistic models, the most convenient, in the case of two-class problems, is the binary representation in which there is a single target variable $\mathbf{t} \in \{0,1\}$ such that $\mathbf{t} = 1$ represents class C1 and $\mathbf{t} = 0$ represents class C2. We can interpret the value of **t** as the probability that the class is C1, with the values of probability taking only the extreme values of 0 and 1. For K>2 classes, it is convenient to use a 1-of-K coding scheme in which t is a vector of length K such that if the class is C_j, then all elements $\mathbf{t}_k$ of **t** are zero except element $\mathbf{t}_j$, which takes the value 1. For instance, if we have K =5 classes, then a pattern from class 2 would be given the target vector

$$\mathbf{t} = (0, 1, 0, 0, 0)^{\mathrm{T}}.$$

Again, we can interpret the value of $\mathbf{t}_k$ as the probability that the class is C_k. For non-probabilistic models, alternative choices of target variable representation will sometimes prove convenient.

It can be identified three distinct approaches to the classification problem. The simplest involves constructing a discriminant function that directly assigns each vector **x** to a specific class. A more powerful approach, however, models the conditional probability distribution $p(C_k|x)$ in an inference stage, and then subsequently uses this distribution to make optimal decisions. By separating inference and decision, we gain numerous benefits. There are two different approaches to determining the conditional probabilities $p(C_k|x)$. One technique is to model them directly, for example by representing them as parametric models and then optimizing the parameters using a training set. Alternatively, we can adopt a generative approach in which we model the class-conditional densities given by $p(x|C_k)$, together with the prior

probabilities p(Ck)for the classes, and then we compute the required posterior probabilities using Bayes' theorem.

In the linear regression models, the model prediction y(x,w) is given by a linear function of the parameters **w**. In the simplest case, the model is also linear in the input variables and therefore takes the form $y(x)= w^Tx+w0$, so that **y** is a real number. For classification problems, however, we wish to predict discrete class labels or more generally posterior probabilities that lie in the range(0,1). To achieve this, we consider a generalization of this model in which we transform the linear function of w using a nonlinear function f(·) so that

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right).$$

In the machine learning literature f(·) is known as an activation function, whereas its inverse is called a link function in the statistics literature. The decision surfaces correspond to y(x) = constant, so that $w^Tx+ w0$ = constant and hence the decision surfaces are linear functions of **x**, even if the function f(·) is nonlinear. For this reason, the class of models described are called generalized linear models. Note, however, that in contrast to the models used for regression, they are no longer linear in the parameters due to the presence of the nonlinear function f(·). This will lead to more complex analytical and computational properties than for linear regression models. Nevertheless, these models are still relatively simple compared to the more general nonlinear.

1.5. Bias and variance: model optimization

Algorithm selection is an important step in building an accurate prediction model, deploying an algorithm with a high rate of accuracy can be a complex balancing act. The fact that each algorithm can produce different models based on the hyperparameters provided can lead to different results.

A constant challenge in machine learning is balancing underfitting and overfitting, which describe how closely the model follows the actual datapoints of the dataset. To understand underfitting and overfitting, it must be first understood bias and variance. Bias refers to the gap between the predicted value and the actual value. In the case of high bias, predictions are likely to be skewed in a certain direction away from the actual values. Variance describes how scattered the predicted values are. Bias and variance can be best understood by analyzing the following visual representation.

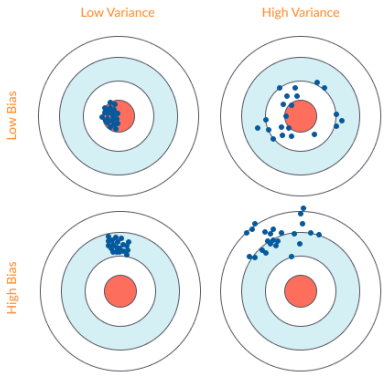

Figure 1.3: Visual representation of Bias and Variance effect on the prediction results

The Shooting targets in Figure 1.3 help to explain bias and variance. In this representation, the centre of the target perfectly predicts the correct value of your model; the dots marked on the target represent an individual simulation based on the training data. If the dots are densely positioned close to the bulls-eye, the predictions made by the model are close to the actual data. In other cases, the training data will be scattered across the target. The more the dots deviate from the bulls-eye, the higher the bias and the less accurate the model will be in its overall predictive task. In the first target, it can be seen as an example of low bias and low variance. Bias is low because the hits are close to the centre and there is low variance because the hits are densely positioned in one location. The second target (located on the right of the first row) shows a

case of low bias and high variance. Although the hits are not as close to the bulls-eye as the previous example, they are still near to the centre and bias is therefore relatively low. However, there is high variance this time because the hits are spread out from each other. The third target (located on the left of the second row) represents high bias and low variance and the fourth target (located on the right of the second row) shows high bias and high variance. Ideally, is desired a situation where there are low variance and low bias. In reality, though, there is more often a trade-off between optimal bias and variance. Bias and variance both contribute to error, but the goal is to minimize the prediction error, not bias or variance specifically.

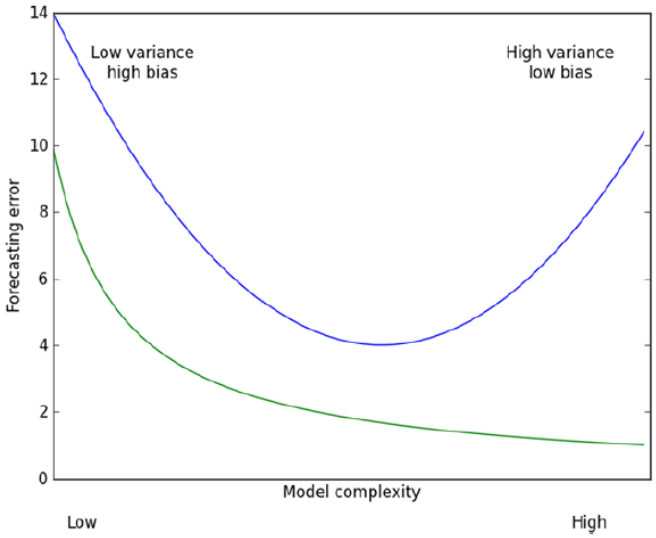

Figure 1.4: Variation of the error of the training and test set respect to the complexity of the model.

In Figure 1.4, it can be seen two curves moving from left to right. The line above represents the test data and the line below represents the training data. From the left, both curves start at a point of high prediction error due to low variance and high bias. As they move from left to right they change to the opposite: high variance and low bias. This leads to low prediction error in the case of the training data and high prediction error for the test data. In the middle of the chart is an optimal balance of prediction error between the

training and test data. This is commonly known as a case of bias-variance trade-off.

Mismanaging the bias-variance trade-off can lead to poor results. As seen in Figure 1.4, this can result in the model becoming overly simple and inflexible (underfitting) or overly complex and flexible (overfitting). Underfitting (low variance, high bias) on the left and overfitting (high variance, low bias) on the right are both to be avoided to get a good accuracy of the model. Adding complexity to the model (as shown on the right) to improve accuracy, could turn into overfitting. An overfitted model will yield accurate predictions from the training data but prove less accurate at formulating predictions from the test data. Overfitting can also occur if the training and test data aren't randomized before they are split and patterns in the data aren't distributed across the two segments of data. Underfitting is when the model is overly simple, and again, has not deeply learned the patterns in the dataset. Underfitting can lead to inaccurate predictions for both the training data and test data. Common causes of underfitting include insufficient training data to adequately cover all possible combinations and situations where the training and test data were not properly randomized. To eradicate both underfitting and overfitting is necessary to modify the model's hyperparameters to ensure that they fit patterns in both the training and test data and not just in part of the data. A suitable fit should acknowledge major trends in the data and play down or even omit minor variations. This may also mean re-randomizing the training and test data or adding new data points to better detect underlying patterns.

1.6. Steps in the development of a Machine Learning analysis

In order to set up a data analysis problem, some steps have to be taken into account. The following procedure will be explained more in details in Chapter 3, but by a general point of view the following steps can bring the problem from the raw data to the complete prediction algorithm:

- **Problem definition:** understand and clearly define the problem as it will be solved. It is firstly necessary to describe the problem and make assumptions, know the constraints imposed by the data and define the attribute of the problem by establishing all the variables that take part in the system studied. The motivation of the job should be determined to have a clear vision of the final purpose of the model and the expectations to its results.

- **Data collection**: all the task is based upon data; this phase foresees the extraction of the samples from a source as a database or collection system. It should be paid attention to issues as sampling frequencies, different type of data (e.g.: manually taken, automatic sampling etc..). In some cases, this requires the installation of field sensors to monitor certain conditions. It is necessary to structurize data to be treated later on, and have a correct correspondence between features and targets of the model. To reinforce the understanding of the problem, a first description of the data is useful, especially in terms of distribution and correlations.

- **Data preparation:** raw data coming from the field and the collection systems must be treated and "cleaned" to be processed by the regression algorithms. It is crucial to select all the input variables collected and even create new ones with analytical methods. The data Pre-processing is important to build a dataset that contains all the variables that will be subsequently correlated to each other.

- **Data analysis and algorithm training:** This is where Machine Learning takes place. Once the dataset is prepared and ready, the regression method has to be chosen according to the characteristic of the data. Many of them are suitable for this task but they are all evaluated in term of prediction accuracy and scoring of the model.

Afterwards, the regression function is built and the model evaluated by selected scoring.

- **Algorithm testing:** the information learned in the previous step is put to use. When evaluating an algorithm, it will be tested to see how well it performs. In the case of supervised learning, there are some known values that can be used to evaluate the algorithm. Here is fundamental to find out some scores that will be used later to monitor the performance improvement.

- **Model's improvement:** when using complex regression algorithms, they need to be tuned with specific parameters able to increase the accuracy of the model; they are called Hyperparameters and are characteristic of each method. From a statistical point of view, the correlation between input and output variable and between input variables themselves should be considered to simplify the problem and enhance the regression task.

CHAPTER 2

PRODUCTION PROCESS AND EQUIPMENT: A FOCUS ON THE GLASS ANNEALING

2.1. Glass production process

According to the American Society of Testing and Materials, the glass is "an inorganic product of fusion that has cooled to a rigid condition without crystallizing"[4]. In particular, the Float Glass is the product of a multi-step continuous process, that bring sands and other material to be melted from a raw form to a definite shape. The manufacturing process of the Float glass includes the following stages:

2.1.1. Melting and Refining

Selected materials are mixed to make a batch, which flows into a furnace heated up to 1500°C. The ingredients of the batch are commonly:

- Silica sand: the main component of the glass
- Limestone: contributes strength properties to glass
- Soda ash: helps the melting interval and act as a flux
- Dolomite: enhance glass resistance to melting
- Glass cullet: known as "broken glass"

Before this stage these components are weighted and mixed and ready to be sent to the melting; this phase takes place in a furnace tank heated up to 1500°C by the combustion of natural gas. Melting and refining take place simultaneously in a whole tank of about 2'000 tonnes of molten glass. They

[4] Formulation according to ASTM-C162

occur in separate zones in a complex glass flow controlled in temperature. It adds up to a continuous melting process, lasting as long as 50 hours, that delivers glass at 1100°C continuously to the float bath. The melting process is key to glass quality and compositions can be modified to change the properties of the finished product.

2.1.2. Forming (Float bath)

The glass flows from the refining tank through a refractory spout on to the mirror-like surface of molten tin, starting at 1100°C and leaving the float bath as a solid ribbon at 600°C. The properties of the tin make it suitable for the forming process: its density, melting temperature range and superficial characteristics make the melted glass floating on it. To prevent tin oxidation the tin bath is filled with a protective atmosphere of nitrogen and hydrogen.

The glass flows onto the tin surface forming a floating ribbon with perfectly smooth surfaces and uniform thickness. As the glass flows along, the temperature gradually decreases from 1100 °C until 600 °C; the control of the flow speed and the settings of top rollers enables the ribbon to vary both in thickness and wideness. Afterwards, the ribbon can be lifted from the tin onto the next section's rollers at a controlled speed.

2.1.3. Annealing

During the cooling of the glass, considerable stresses are impressed in the ribbon. A too high level of stress will break the glass during the other steps of cooling and cutting. To relieve these stresses the ribbon receives a thermal treatment in an annealing lehr where temperatures are controlled both along and across the ribbon. During this stage the glass pass through an interval of temperature where its viscosity changes radically; it therefore goes from a visco-elastic state to a solid-state and the strains acquired here do not have the time to completely relax, producing on the glass the permanent stress above mentioned.

2.1.4. Quality inspection

To ensure the highest quality of the glass sheets, measurement are performed with automatic vision and detection systems to check the quality parameters such as the absence of melting defect and bubbles, optical properties, thickness and stress state. This phase reveals process evolution upstream that can be corrected to stay at an acceptable level of these parameters.

2.1.5. Cutting

High hardness wheels trim off the sheet's edges and cut the ribbon to standard formats. The glass will be subsequently picked up on stillages and transferred.

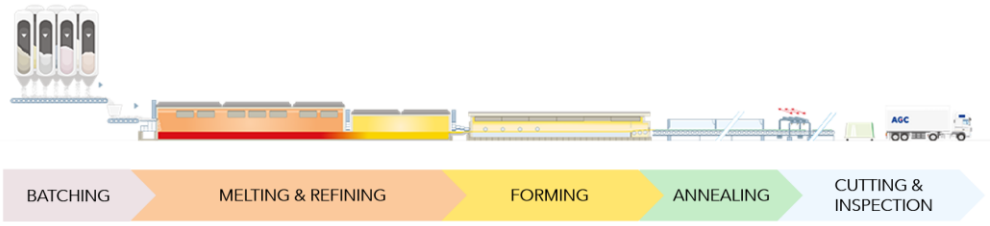

Figure 2.1: Steps of Float glass manufacturing process.

2.2. Glass properties

Glasses do not exhibit the ordered crystalline structure of most other ceramics but instead have a highly disordered amorphous structure; the atoms are not organized according to a definite structure as in a crystalline matter. When cooling, the atoms do not have enough time to arrange in fixed locations, but they take a disordered position very different from the close-packed structure typical of the crystalline materials.

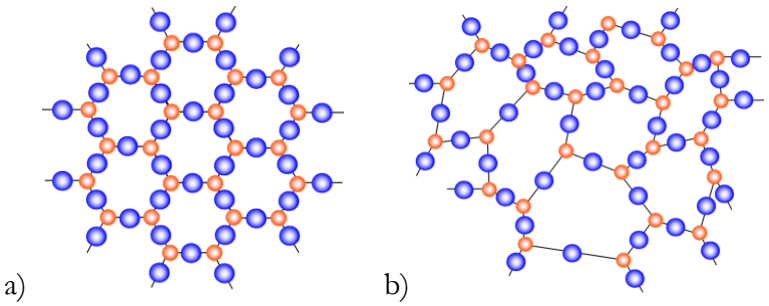

Figure 2.2: Crystalline structure (a) and amorphous structure (b) typical of the silica glass.

2.2.1. Viscosity

Due to its amorphous structure, a glass at room temperature is essentially a very viscous liquid; it was called in the past "frozen undercooled liquid". The viscosity of glass changes with temperature; four standard temperature points are used in glass working to define the viscosity. These points are strain, annealing, softening, and working point.

The strain point is the highest temperature at which a glass can be used for structural purpose without creeping. Below this temperature it is not possible to relieve the internal stresses, e.g at the strain point the stress relieve may last about 15 hours. Between the annealing and the strain point, glass products should be cooled down gradually to avoid the formation of internal stresses due to temperature gradients.

The annealing point is the temperature to which glass may be heated after working to relieve any internal stresses that arose as a result of the forming process; at this temperature, the internal thermal stresses are relieved completely by viscous relaxation within about 15 minutes. The glass undergoes a thermal contraction when it is cooled and at high temperatures the glass decrease rapidly in specific volume. At a certain point, the glass transforms to a hard brittle texture and the glass decrease in specific volume at a slower rate. The temperature at which this transformation happens is

known as Transition Temperature (Tg). The transformation temperature g is the temperature, where the transition takes place from the solid-state into the liquid phase. The value of g is dependent on the rate of cooling as shown in Figure 2.2.

At the softening point, the glass will yield with a small amount of force while at the working point the glass has the viscosity comparable to the honey. The majority of the forming operations take place between the softening point and the working point. Actually there is a working range because the glass cools down during forming. Whether the forming process takes place at a higher or a lower viscosity is determined also by the safe deformation forces (without the formation of cracks or disrupture) and cooling rate of the glass product.

Glass does not have a melting point because of the gradual transition from the solid into the liquid state. However, the melting temperature is defined as the temperature at which the glass melt has become a viscosity low enough to allow for well mixing. This low viscosity is required to achieve a proper melting process. Here, the melting process comprises both the homogenizing (mixing) and the fining (removing gas bubbles) of the melt. So here it has a completely different meaning than the melting point of crystalline matter (transition solid phase into liquid phase).

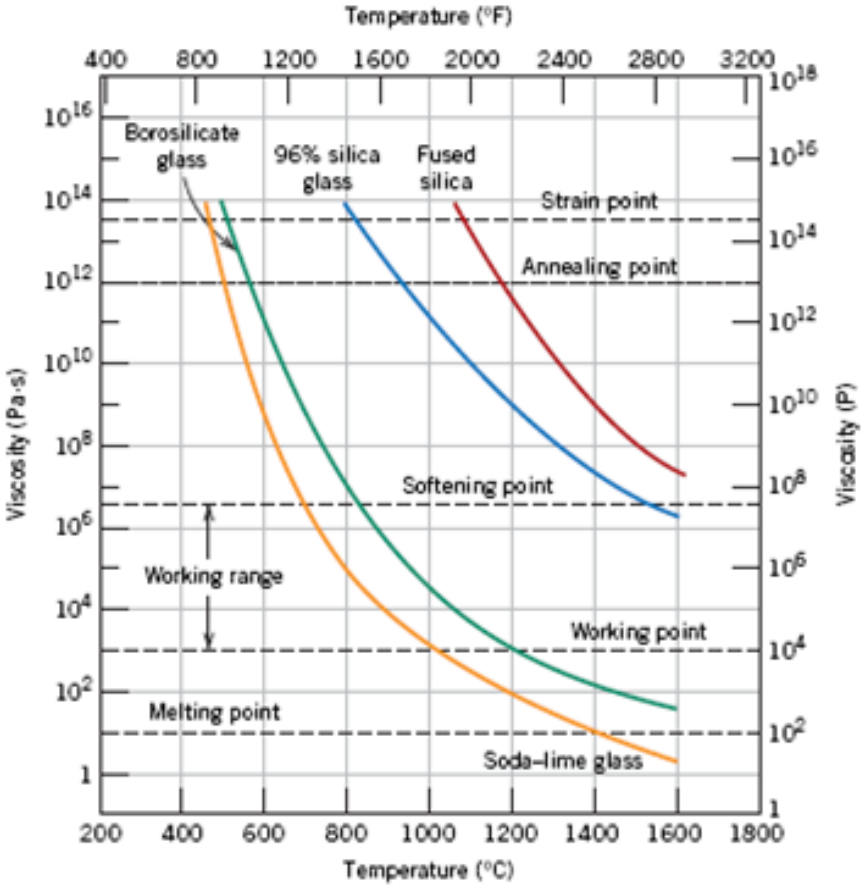

Figure 2.3: Viscosity curves for different commercial glasses.

2.2.2. Specific volume

This is shown also in Figure 2.4, in which the specific volume is plotted against temperature. The instantaneous transition from a melt phase into a solid crystal phase at point l and the corresponding change of volume does not occur in the glass when cooled fast enough. The "liquid" behaviour will still continue during further cooling until g, the transformation temperature is reached. Below g glass behaves itself like solid matter. The slope of the curve in Figure 2.4, below T_g is almost parallel to the one for the solid crystalline phase. The level of the curve below g depends on the rate of cooling and consequently on its previous history.

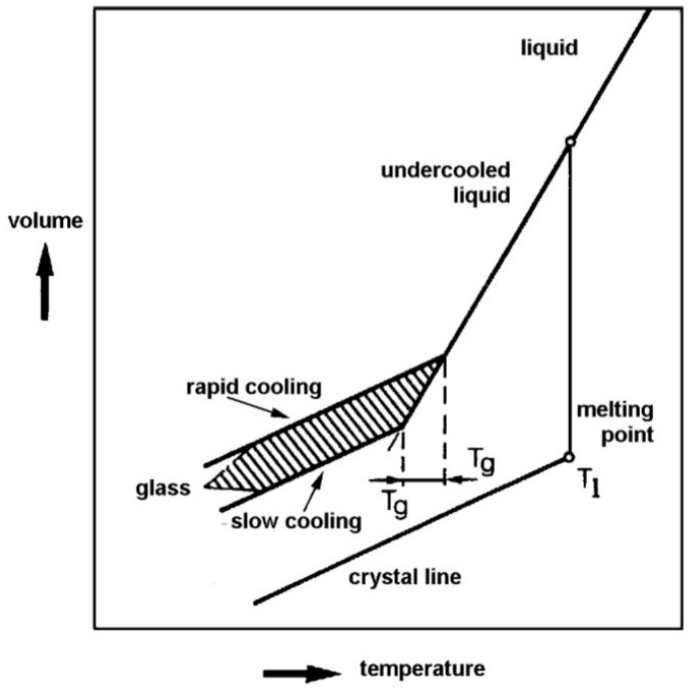

Figure 2.4: Behavior during cooling of a melt for crystalline matter and glass. For glass, the condition below about g depends on the rate of cooling. The transformation temperature g itself, the temperature where the slope changes, also depend on the cooling.

2.2.3. Density

The density of liquid glass is important for melting technology. The temperature differences present in the melt causes local density changes, which lead to free convection flows. These flows stimulate the (convective) heat transfer in the melt and improve the homogenization process by the mixing of the glass. In the temperature range of 1000 - 1500 °C the temperature dependence of the density may be described with a volumetric expansion coefficient. Mostly the expansion coefficient is assumed to have a constant value at this temperature range. Because of the difficulty and complexity of the experimental procedure, the density and the expansion coefficient are not measured frequently in the temperature range above g. For

temperatures above g, no models are available to describe the density and the expansion coefficient based on the composition.

2.2.4. Coefficient of thermal expansion

This coefficient has a strong dependency against the temperature; the usual range within it is specified is from room temperature to the glass transition range, where volume changes become greater. The expansion of the glass as the temperature is increased is an important measure of its resistance to thermal shock. Nonuniform volume changes, resulting from temperature gradients in heating or cooling, cause stresses that are larger the greater the volume changes; also for many applications, it is essential to know the expansion characteristics accurately, in view of construction requirements. The expansion coefficient increases slightly with increasing temperature up to T_g. Around T_g the glass structure changes which results in a much stronger dependency on temperature. Because the expansion coefficient depends on the glass structure, it also depends slightly on the history of the glass. The cooling rate after forming co-determines the immobilized structure and therefore also the value of the expansion coefficient.

To provide an example about the effect of this coefficient, a vitreous silica beaker heated to 1000°C can be dashed into the water without breaking it because it has a low expansion coefficient of about 5×10^{-7} °C^{-1}; on the other hand, a plate of 6mm of thickness, with a coefficient of 90×10^{-7} °C^{-1}, can be cracked with a difference of temperature between the two surfaces of about 50°C.

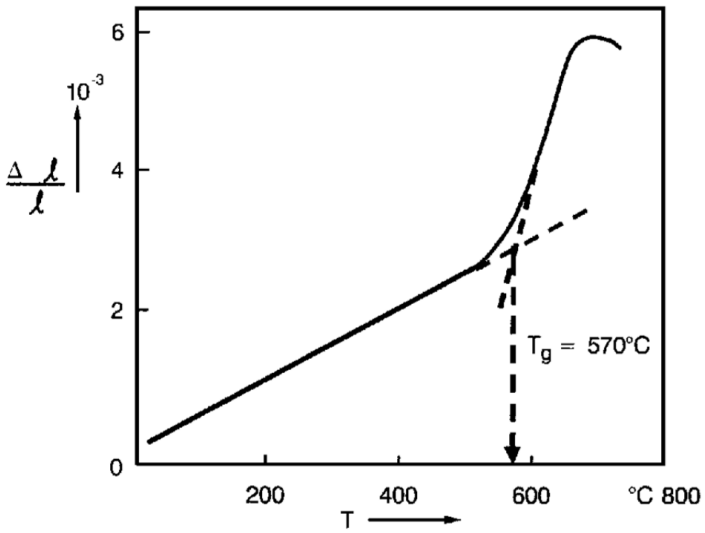

Figure 2.5: Thermal expansion as a function of temperature.

2.2.5. Birefringence

Isotropic solids do not exhibit birefringence, however, when are applied mechanical stress the birefringence results. The glass becomes anisotropic through mechanical and thermal-induced stress and therefore also the refractive index inside the glass changes locally and becomes anisotropic. Plane polarized waves will be travelling at different velocities through the stressed parts of the glass depending on their polarization direction. For small mechanical stresses, the refractive index changes are proportional to the mechanical stress itself. The permanent stresses resulting from the annealing process and the inhomogeneities of the refractive index are generally very small. The stress birefringence can be expressed as the difference in optical path length between two incident plane waves (oscillation planes oriented parallel and perpendicular to the main axis of stress) transmitting the sample of a certain length at the measurement position. The stress birefringence is proportional to the effective principal stress difference.

As shown in Figure 2.6, the way the stresses can be measured on the glass sheet is based on this principle; a beam of polarized light propagates through

the glass sample and the light vector splits into two polarized beams oscillating in the planes **x** and **y,** that correspond to the principal direction of strains. Considering the speed of light constant, and two different levels of strains on the two principal directions, the time necessary to cross the plate for each of them will be different; it is so possible to measure relative retardation between the two polarized light beams. This relative retardation, considered as a space difference between the two beams, is detected by an analyzer positioned on the other side of the sample.

The measuring method of the stresses will be deeper treated in the next paragraphs.

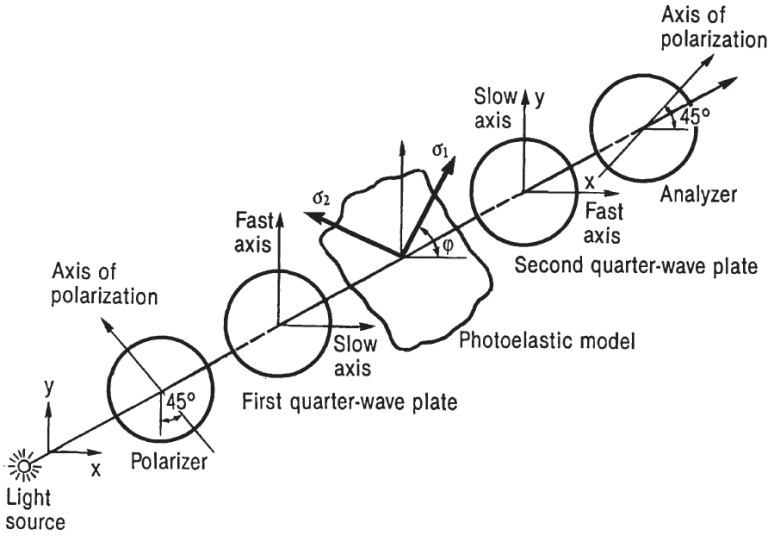

Figure 2.6: Measurement principle of a polariscope, based on the birefringence property of the glass.

2.3. Glass annealing[5]

Annealing is one of the major processes in the manufacture of glass. It is a controlled slow-cool heat treatment of glass. In the float glass manufacturing process, annealing immediately follows forming. Glass is annealed for a variety of reasons:

- The most common one is the reduction of residual stresses in glassware. Poorly annealed glass with large residual stresses break easily during handling or sometimes even spontaneously.

- Some glass products are annealed to obtain a uniform residual stress distribution. This ensures that no part of the glass is under tension and a small residual compressive stress is obtained at the glass surface. Such a stress distribution makes glass resistant to thermal shocks. Television and radio tubes exposed to cyclic heating and cooling and bottles used in food processing are examples of glassware that must resist thermal shock in service.

- In the float glass process, optimum residual stress facilitates orderly cutting of glass ribbon into rectangular brackets. High residual stresses make it difficult to cut glass ribbon into desired shapes and low stresses often cause loss of control of the cutting sequence, i.e., cracks propagate in "incorrect" directions.

2.3.1. Physical mechanisms governing the process

In this section, it will be examined three physical mechanisms that are important in annealing: structural relaxation, stress relaxation and heat transfer. Only the last two will be explained in the next sections.

[5] (Narayanaswamy)

2.3.2. Stress relaxation

Stress relaxation in hot glass has been studied extensively by glass scientists and technologists. Even though a systematic study of this subject began a century ago (Schott, 1875), only much more recently a clear, simple, and accurate formulation was made. Of the earlier studies, the classical work of Adams and Williamson (1920) is the most widely known. Their formulation of stress relaxation in the glass is still widely quoted in the technical literature. For this reason, their work will be reviewed critically before taking up modern theories of stress relaxation. Adams and Williamson attempted to use Maxwell's relaxation model as the framework to fit their data on the glass. Figure 2.7 shows a representation of Maxwell body consisting of a spring and dashpot in series. If this system is subject to a step increase in strain ε, its instantaneous response would be elastic. As viscous flow proceeds, this stress gradually relaxes to zero.

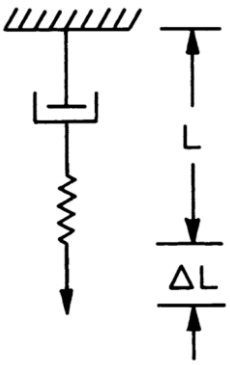

Figure 2.7: Spring- dashpot mechanism proposed by Maxwell in its stress relaxation model.

Thus, in a Maxwell body stress decay follows a simple exponential law. Adams and Williamson found that their data did not agree with their equation, where the rate of stress relaxation is proportional to the square of the stress. After many years of studies and experimental validations, it has been established that stress relaxation of glass is not a simple exponential decay but contains a spectrum of relaxation mechanisms.

Figure 2.8 shows normalized relaxation curves for a typical soda-lime glass stabilized at different temperatures.

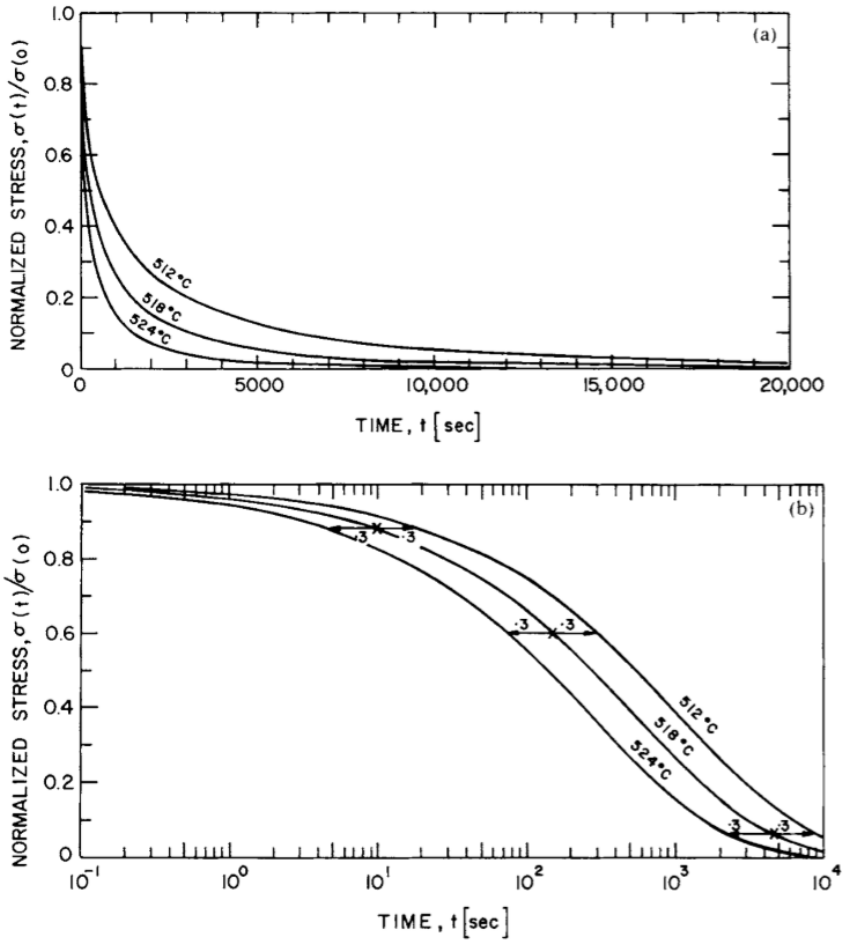

Figure 2.8: Normalized stress relaxation curves for stabilized soda-lime glass.

2.3.3. Heat transfer

Considering the temperature of the glass, three mechanisms can take part in the heat transfer phenomena: conduction, convection and radiation.

Conduction

Heat transfer by conduction is caused by the thermal vibration energy of the ions being transferred to their neighbour ions. This mechanism is also called phonon conduction. The amount of heat transferred depends on the thermal conductivity and the temperature gradient according to Fourier's law. At room temperature the thermal conductivity of a soda-lime glass amounts to 1.0 W/(m K), near T_g it is near 1.5 W/(m K) and around 1200 °C one can take a value for of about 2.0 W/(m K).

Radiation[6]

At temperatures above 300 °C heat transfer by radiation becomes more important and above 500 to 800 °C it is eventually predominant and becomes more and more important as temperatures increase further. The mechanism of radiation exchange is based on the emission of radiant heat, which is partly absorbed again by glass zones nearby. Heat radiation is composed of light quanta or photons. The transfer of radiation is also called photon conduction because the photons are the heat carriers. The photon conduction increases with the transparency of the glass for the photons. With increasing temperature, the heat spectrum of a body moves from infra-red (IR) via the near infra-red into the range of visible light wavelengths. It moves to wavelengths, for which the glass becomes more transparent for this radiation, resulting in strongly increased radiation transfer (photon conduction). See the curves in Figure 2.9.

[6] (NCNG, 1997)

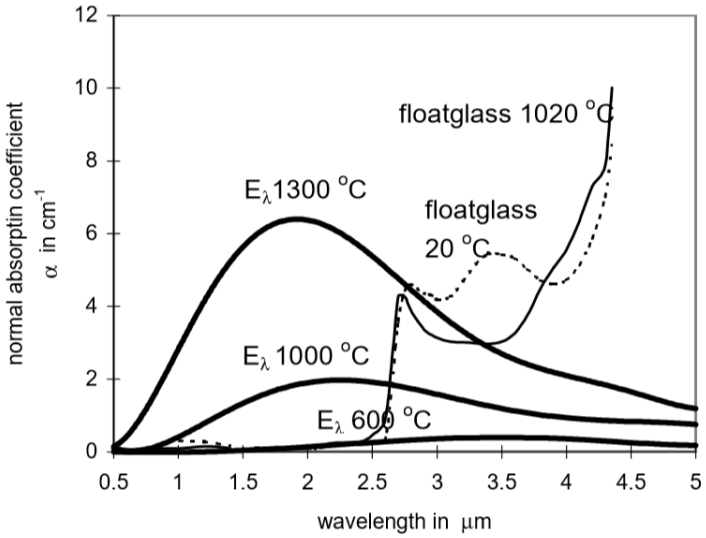

Figure 2.9: Spectrum of heat radiation of a black body at 3 temperatures and spectral absorption of float-glass.

In Figure 2.9, the spectral distribution of the heat radiation intensity of a black body is shown for three temperatures together with the absorptive spectrum of a float-glass. From Figure 2.9 it is evident, that above λ = 2.5 m, the transparency of float-glass is becoming very poor and there is no transparency at all above λ = 5 m. Also, we see that a (black) body at 600 °C practically radiates only in a range of λ > 2.5 mm. Therefore the glass will not transmit this radiation below about 600 °C and the heat can only be transferred through conduction. The glass is already very rigid at T < 600 °C and under normal conditions, no convection can take place within the glass. From the spectral curve of a body at 1000 °C it is apparent, that now a substantial part of the energy is radiated at wavelengths below 2.5 m, where this glass is reasonably transparent. Therefore this radiation will be well transmitted. Although it is difficult to see in Figure 2.9, there is still some absorption at λ < 2.5 m; the absorption coefficient being in the order of 0.1 - 1 cm^{-1}. Without any absorption, the radiation would be able to reach directly and undisturbed the opposite wall and heat it without heating the glass. This is called passive (no direct heating of the glass) radiative heat conduction. However, in this case, at a value for the normal absorption coefficient of 0.1-1 cm^{-1}, the glass will partly

pick up the heat through absorption. Each heated volume element will start radiating itself (having a certain emissivity depending on the wave-length) and by doing so also be transferring heat. Such a material which absorbs radiation, but where the free path for radiation is of the same order as the dimension of the object (in this case the depth of the furnace) is called diathermic. Radiation can be absorbed by a volume element of the glass heating the material and radiation is re-emitted by this volume-element depending on the composition and local temperature. So, the glass (melt) conducts radiation but also absorbs and re-emits radiation, being heated and being a conductor for radiation. In this process, the presence of various metal ions in the glass plays an important role, because these ions might determine the absorption to a large extent.

Convection

The convection takes part majorly from a temperature above 800°C of the glass; it is caused by the flowing of the glass as in the melting tank and during forming at viscosity's low enough to allow the flowing of the melt. In this study will not be taken into account the convection between air and glass during the annealing due to the low contribution it gives to the heat balance.

2.3.4. Theories of annealing[7]

Classical views on annealing glass were divided between two schools of thought. One, originated by Adams and Williamson (1920), was a thermomechanical view that explained annealing purely in terms of stress relaxation. The other, advanced by Lebedev (1921), was a structural view, which emphasized structural changes in glass, sometimes to the exclusion of mechanical considerations. While structuralists argued, justifiably, that structural changes must affect annealing stresses, they had no quantitative model to evaluate its importance. On the other hand, Adams and Williamson presented a quantitative model that was apparently successful in predicting

[7] (Narayanaswamy)

residual stresses in annealed glass. Thus, the mechanical view of annealing gained wide acceptance among those whose interest in annealing was control of residual stresses in the glass. Annealing of optical glass, of course, required knowledge of structural relaxation and its effect on the refractive index of glass. The interaction of the two physical phenomena was not fully explored. In the last couple of decades, considerable progress has been made in developing quantitative thermomechanical models of stress relaxation in ideal viscoelastic materials. A landmark paper of this era was published by Lee et al. (1965) who presented a sound mathematical framework for viscoelastic stress analysis in the glass. Narayanaswamy (1971) applied the mathematical formalisms of viscoelasticity to develop a quantitative model of structural relaxation in the glass. This model was first applied by Gardon and Narayanaswamy (1970) to present what might be called the modern theory of annealing, which is a unified view of stress and structural relaxation in the glass. We shall first review classical theories and their shortcomings before discussing the modern theory of annealing.

2.3.5. The modern theory of annealing

In 1965 Lee et al. published a paper on stress generation in glass in which they presented for the first time a solid mathematical framework to describe stress generation in a glass slab. This paper was the basis for much further theoretical work on both tempering and annealing and the development of "structural models" of both these processes, even though they made no reference to structural relaxation. Narayanaswamy (1971) adapted their mathematical tools to develop a comprehensive model of structural relaxation that predicted with sufficient accuracy the structural response of glass to an arbitrary temperature history. Gardon and Narayanaswamy (1970) applied this new model to analyze the annealing of soda-lime glass. Since their analysis of annealing contained all current concepts of stress and structural relaxation, their work may conveniently be referred to as the modern theory of annealing. Let us now briefly review the salient features of this theory. Any

comprehensive analysis of annealing must necessarily begin with the calculation of the thermal history of glass. Let us simplify this task by considering a glass slab cooled at a constant rate. After the decay of initial transients, a parabolic temperature gradient develops across the thickness of the slab. As long as the cooling rate is kept constant there will be no change in thermal gradient. Thus we obtain the simple expression for thermal history in the slab given by the equation:

$$T(x,t) = To - R(t) + KL2R\left(\frac{1}{4} - \frac{x2}{L2}\right)$$

If the initial temperature T_0 is chosen to be sufficiently greater than the upper limit of the glass transition range, the initial thermal transients neither introduce stresses in glass nor cause any deviation of glass structure from its equilibrium state. Thus, in subsequent analysis, these transients will be ignored and the above equation will be taken as valid from time zero. The next step in the analysis is the calculation of fictive temperature history. […] the thermal history of material at a given point P determines a unique Active temperature for that point. For the sake of simplicity, we are using only one fictive temperature. Thermal strains introduced in different layers of the glass slab are then given by:

$$\varepsilon_{th}(x,t) = \beta_g [T(x,t) - T_f(x,t)] + \beta_l [T_f(x,t) - T_0]$$

We must observe here that theories of annealing that ignored structural relaxation in glass apparently succeeded in "predicting" anneal stresses because they used an effective coefficient of thermal expansion. Thus, their expression for strain would be

$$\varepsilon_{th}(x,t) = \beta_{eff}[T(x,t) - T_0]$$

We see now that such a success was merely a matter of selecting a value for β_{eff} to produce agreement between calculated and measured anneal stresses.

Effective expansivity is not a physical property and may have different values for different processes (Narayanaswamy, 1978). On the other hand, β_g and β_l, are physical properties and hence are independent of process conditions. Haggerty and Cooper (1965) measured thermal expansivity in the liquid and glassy regions and found that in soda-lime glass structural expansivity β_s is twice the thermal expansivity β_g.

The third and final stage of the analysis is the calculation of stresses induced by the strain history. This part of the analysis is purely classical thermo-viscoelasticity except for the calculation of reduced time. Unlike structural relaxation, stress relaxations in different material points interact. This is so because stress distribution across the thickness of the slab must satisfy the following equilibrium condition at all times:

$$\int_0^{L/2} \sigma\,(x,t)dx = 0$$

Stresses and strains in different layers are related by the familiar Boltzmann's integral

$$\sigma(x,t) = \int_0^t \bar{R}[\xi(x,t) - \xi'(x,t')]\frac{\partial \sigma_g}{\partial t'}\,dt'$$

where R(ξ) is the normalized auxiliary relaxation modulus, σ_g is the generated stress, ε(t) is the total strain, which equals the viscoelastic strain plus the thermal strain, and σ(x,0) = 0 is the assumed initial condition.

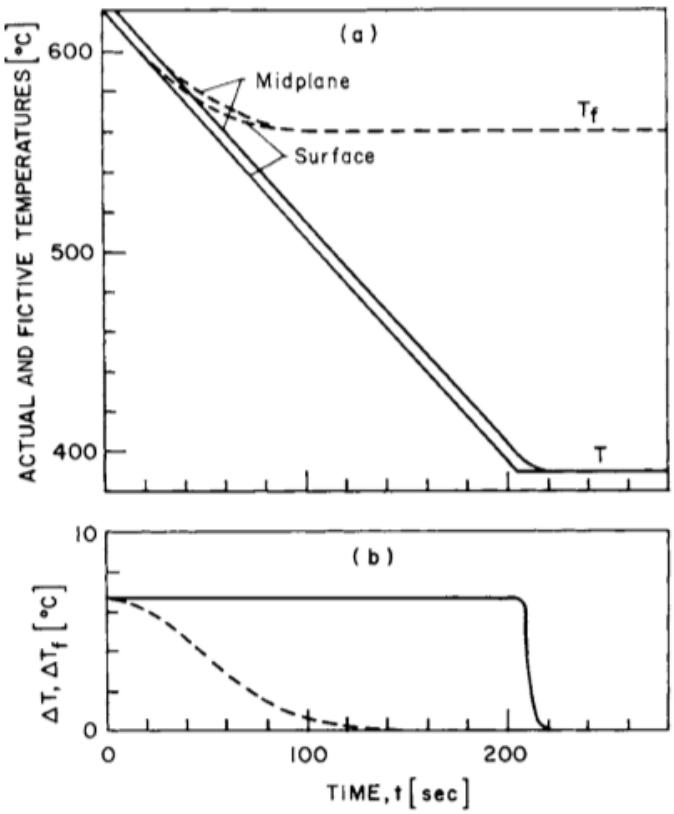

Figure 2.10: Development of actual and fictive temperatures (a) and temperature gradients (b) in glass (0,57 cm) cooled at constant rate of 1,12 °C/sec.

We have the two previous equations to solve for two unknowns σ(x,t) and ε(t). Details of deriving and solving these equations can be found in the works of Lee et al. (1965) and Narayanaswamy and Gardon (1969). Figure 2.10a shows how actual and fictive temperatures change with time in a glass slab 0.57 cm thick cooled at l.l°C/sec. Solid lines represent temperatures of the surface and midplane. They are shown by dotted lines in Figure 2.10a. Note that since both surface and midplane cool through the transition range at the same cooling rate, they experience thermal histories that are identical but shifted in time. The same is true for fictive temperatures of the surface and midplane. Fictive temperatures for other layers of the slab are obtained by a simple shift in the time scale. Figure 2.10b compares gradients ΔT and ΔT_f of actual and fictive temperatures. They are equal in magnitude but

decay to zero at different times. The fictive gradient decays first when the transition range is crossed, and ΔT decays later when the cooling is stopped. Since β_s is twice β_g, the former event generates stresses in the slab that are twice as large as those generated by the latter event. However, since the decay of fictive gradients occurs in hot glass, only a small fraction of stresses generated by this decay survive stress relaxation. On the other hand, gradients of actual temperatures decay in cold glass, so that all of the stresses generated by this remain as residual stresses. Figure 2.11 compares the numerical solution (circles) with measured stress generation (solid line) in the midplane of a glass slab cooled at l.l°C/sec. The evolution of stress is characterized by two plateaus, the first one induced by the decay of structural heterogeneity and the second by the decay of thermal gradient. Figure 2.11 answers clearly the long-debated question, "What fraction of the total anneal stress is attributable to structural relaxation in the glass?" We see that it is 40% of the total, that is, 31 nm/cm of the total, 76 nm/cm. The remaining 60% is attributed to thermo-mechanical changes, i.e., to the relaxation of thermal gradient in the glass. In some glass literature, stress and structural relaxations are treated separately. Stresses induced by residual structural heterogeneity in cold glass are added to the classical residual stress. This procedure is erroneous because stress and structural relaxations are simultaneous and coupled phenomena. The constant rate cooling schedule reviewed here is an excellent counterexample. It produces no permanent (residual) heterogeneity in structure. But, transient heterogeneity in structure does occur (ΔT_f in Figure 2.10b) and that is responsible for 40% of the total anneal stress. Before ending this section, contributions to annealing by Tackels and Crochet (1973) and Crochet et al. (1974) should be mentioned. Like Gardon and Narayanaswamy, they also take a unified view of stress and structural relaxation in annealing. However, their model of structural relaxation still retains the classical one-parameter approach of Tool. They model structural relaxation by the differential equation

$$\frac{dT_f}{dt} = \exp\left[\frac{T - T_0}{g} + \frac{T_f - T_0}{h}\right] F(T - T_f)$$

As we stated before, this formulation neglects the "memory effect," which may become important in thermal processes in which temperature variation is nonmonotonic. The author recommends the integral formulation or its equivalent (Markovsky and Soules, 1984), which permits multiple relaxation mechanisms and thus allows for observed memory effects in glass. Numerical solution of the more general formulation is quite straightforward and is a simple task for modern digital computers.

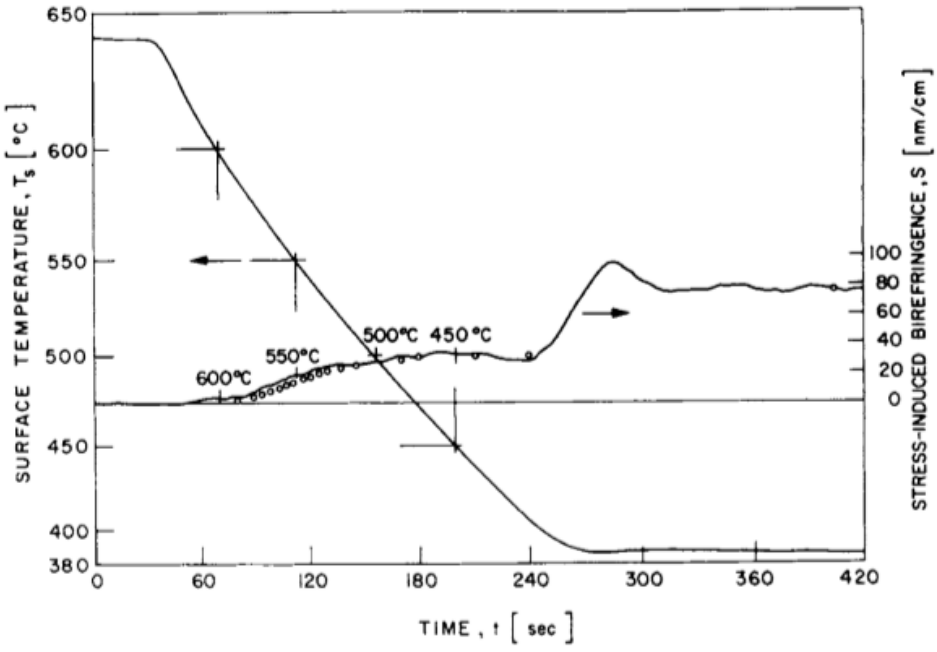

Figure 2.11: Temperature and stress as a function of time during the annealing of sheet glass. Solid curves are experimental recordings; circles represent calculated stresses. Glass thickness=0,57 cm; cooling rate= 1,12°C/sec.

2.4. Permanent stress on flat glass

During the annealing, the glass ribbon is provided with a certain distribution of spatial and time temperature gradients. The thermal effect applied to the glass ribbon generates a three dimensional status of strain that causes mechanical stresses. The stresses generated can be divided into 2 categories: the temporary stresses, that are due to the elastic properties of the glass and are majorly considered below the transition temperature of the glass, and they are present only if the thermal gradient is maintained. The second typology is the permanent stresses, that are characteristic of the visco-elasticity of the glass and are generated over the Tg=480°C, and are completely developed when the temperature in each point of the sheet is homogeneous. To sum up, a glass ribbon that is cooled from a $T>T_g$, where its internal structure can reorganize, to a $T<T_g$ where it is "frozen", acquires distribution of permanent stress caused by the different specific volume across the ribbon width. These stresses will be considered for the creation of the model and are the output of the system to be modelled and simulated. When the glass is cooled, the top and bottom surfaces of the plate are cooler than the centre; this distribution causes the so-called *laminar stresses*, that are equal if measured along and across the glass sheet; it is so possible to understand that it is due to time gradients of the temperature – so the cooling rate of the glass ribbon. Due to a natural effect and the technology constraints, the edges of the glass ribbon are cooler than its centre; when a temperature distribution is given across the width of the ribbon, this will generate a distribution of *transversal stress* with the direction along the glass ribbon. Effectively these are the stresses that want to be simulated by the present work.

To better explain the effect that distribution of temperature along the ribbon width and the temperature's transition from a temperature higher than T_g, to a lower temperature, consider the glass ribbon as divided into longitudinal slices. If the centre of the ribbon is warmer than the edges, each slice across the width has a different specific volume but above 480°C its viscosity is

enough low to let the glass release the stresses progressively by a reorganization of the internal structure; when the ribbon is cooled below 480°C, its viscosity is too low to let the structure adapt and it is fixed; if the slices were ideally able to move independently, at ambient and homogeneous temperature the would move and not generate any permanent stress. In real condition below 480°C, the structure of the glass is fixed and the slices cannot move, so it results a condition of permanent transversal stress that is proportional to the temperature difference between each point of the glass sheet.

2.4.1. Thickness Stresses[8]

Residual stresses in a glass sheet occur due to the temperature gradient during cooling across the transformation range. It is customary to consider that these stresses are the result of the superposition of two kinds of stresses: thickness stresses and membrane stresses. The former is generated by the temperature gradient which is perpendicular to the surfaces and the latter is generated by the gradient parallel to the surfaces. When a glass sheet gets cold during tempering or annealing, the heat transfer occurs principally through its surfaces. Consequently, the highest temperature gradient is directed across the sheet thickness and the main residual stresses are the thickness stresses. They are planar stresses.

More often than not, one may consider the cooling conditions to be the same on both surfaces of the glass sheet. The median plane is thus a plane of symmetry for residual stresses. When the sheet is cooled asymmetrically and it is prevented from bending, the residual stress distribution remains parabolic but the top of the parabola is shifted nearer to the surface which is cooled more slowly. Such ease can arise when, in a glass sheet, adjacent areas have been subjected to different asymmetric cooling. If these areas of glass were

[8] (Aben & Guillemet, 1993)

separate entities they would have different curvatures at room temperature. However, since they are attached to each other, they have to take a mean curvature. The result of this is a bending of each part and thus an asymmetry of their thickness stress profiles. In practice, this ease occurs after annealing of the glass ribbon when, for example, the central zone tends to get a concave bow shape and the two lateral zones a convex bow shape. If a glass sheet has been subjected to asymmetric cooling throughout the transformation range and if it is free to bend, it obtains a curvature whose concavity is directed towards the surface with the slowest cooling. This is due to the fact that the residual stress distribution has to satisfy the bending moment is equal to zero. The result of this is a perfectly symmetric residual stress profile. However, if bending is restricted then bending stresses appear, and the resultant stress distribution is asymmetric. This situation appears in the ease of horizontal tempering. When the glass sheet to be tempered rests on a horizontal frame or on horizontal rolls, the blast of air is a little more intense on the upper surface than on the lower one; this is necessary for the glass to remain in contact with its support. As a consequence of this, tempering induces a slight curvature whose concavity turns towards the lower surface. Asymmetric cooling also occurs at the end of the bending process of the two glass plates which compose a windscreen. These two sheets stacked together are put on a frame, heated up to 600°C to bend them under the action of gravity and then cooled slowly. If this cooling is carried out too quickly, the two sheets will not get the same curvature at room temperature; the lower one will be more bent than the upper one. The laminating with the polyvinyl butyral film, to be carried out later, will induce a bending moment of opposite sign in each sheet and these stresses will be superposed to the stresses due to annealing. Let us mention that the residual stress profile is no longer parabolic if cooling of the glass sheet through the transformation range has not been monotonous.

2.4.2. Transversal stresses [9]

The thickness of a glass sheet is very small in comparison with its other dimensions. Thus, generally, the temperature gradient parallel to the surface, which is a function of time and of the point of the sheet considered, can be assumed to be constant through the thickness in each point. This temperature gradient induces membrane residual stresses in the glass sheet. These stresses are constant through the thickness and are superposed to the thickness stresses. One can measure these membrane stresses regardless of thickness stresses by adapting the classical methods of two-dimensional photoelasticity.

From a practical point of view, the most important are the residual stresses which arise due to the special cooling conditions which generally prevail on the edges. As most of the time, the edges are cooled more quickly than the centre of the sheet, the edge stresses are generally compressive. Edge stresses affect the stress intensity factor of the flaws which occur during cutting or grinding. These defects are generally the ones which weaken the strength of the sheet most. Due to this, the edge stresses have a strong effect on the resistance of the sheet during bending or twisting. In laminated windscreens, it is the edge stresses which determine the probability of fracture during assembly to the car body. They are systematically measured during production. Since these edge stresses are unidirectional, they are easy to measure with photoelastic methods; the stress is proportional to the measured path difference δ.

In long glass ribbons, the longitudinal temperature gradient during the cooling is usually very slight, therefore the membrane stresses are affected only by the temperature gradient in the direction perpendicular to the ribbon axis. Thus, only the stresses parallel to the axis of the ribbon are not equal to zero. These stresses are to be controlled on-line to avoid the breaking of the

[9] (Aben & Guillemet, 1993)

ribbon and to enable easy cutting of it at the lehr exit. Formerly the stresses were measured with visual photoelastic devices supplied with a compensator.

The stress level is proportional to the expansion coefficient. The region which has become rigid first during cooling, come under compressive stress. The inner part of the product or the surfaces that have been cooled more slowly, come under tensile stress. Usually one attempts to avoid this to happen because it will weaken the product and it makes further treatment, like cutting and sawing, impossible. Therefore cooling as much as possible without creating stresses is essential.

2.5. Annealing lehr technology

The equipment that makes implementable the annealing process described in the previous parts, is the *annealing lehr*. A glass annealing lehr oven is a long, temperature-controlled, kiln. Lehrs are typically 6m wide and 120m long, with an adjustable temperature gradient through which the glass passes. In the case of the flat glass, the product is transported on a roller conveyor. Adjustable electric heating elements and air heat exchangers are located in the lehr to maintain a consistent temperature profile across the width of the glass ribbon and the desired cooling rate along the glass flow direction. If the glass ribbon were to cool in ambient temperature air, the surfaces would cool much more rapidly than the internal body of the glass. This temperature gradient would cause the surface to be in severe compression, which will cause the glass to break spontaneously as the stresses exceed the strength of the glass.

From a layout point of view, the lehr is composed of different section, and each of them has different purposes for the thermal treatment of the glass. The first part is thermally insulated because the temperatures are high and the control of the temperature distribution is crucial; this part is made by the zones A, B and C. These three zones have similar geometries and devices and control:

- Electrical resistances that provide heating power, transferred to the glass by radiation. The electrical power regulation is done by SCR (Silicon Controlled Rectifier) and the setpoint is provided by production operators. In each zone electrical resistances have a dedicated control on the top and bottom surfaces and in five sections across the ribbon width (the central one is twice wide than the others); along the glass ribbon the regulation is constant.

- Air heat exchanger that provides cooling power, not transferred by forced convection but by radiation. The airflow regulation is provided through electro-mechanical regulation of valves, there is a unique fan for each zone (top-bottom) that generates the airflow. The particular design of the heat exchanger increases the efficiency thanks to a sequence of rectangular tubes in close position, this geometry creates a kind of radiation trap that increase the exchange surface and consequently the thermal power absorbed. As the electrical resistances, heat exchanger have a dedicated control on the top and bottom surfaces and in five sections across the ribbon width (the central one is twice wide than the others). The tubes are developed along the ribbon and this shape causes a different level of thermal power that the glass exchange with the heat exchanger along this direction because at the inlet the air temperature will be lower than the outlet; it also depends by the direction of the airflow and so by the position of the fan as shown in the figure below. The opening of the valves is regulated in a close loop with a PID control upon the temperature level.

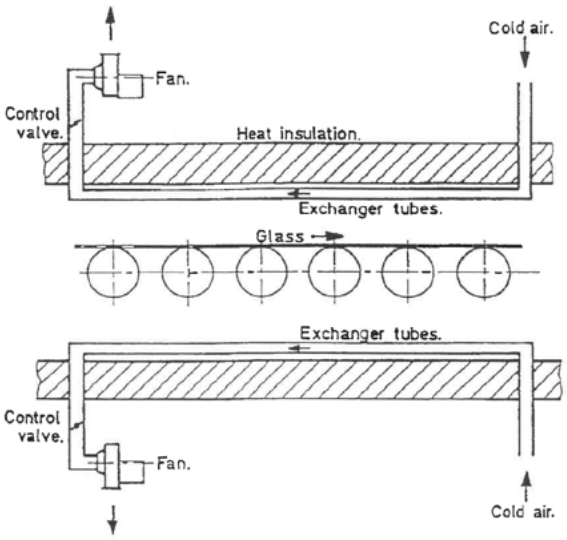

Figure 2.12: longitudinal section of the lehr. It is shown that the tubes of the heat exchanger are developed along the glass ribbon, this generates a different value of thermal power exchanged depending on the position of the fan.[10]

- Burners provide supplemental thermal power exchanged both by radiation and forced convection because the flame hit directly the glass surface. The two burners are in the edge position of the ribbon and at only in zone A; their regulation is made manually and the transversal position can be changed.

- Thermocouples: measure the temperature of the air close to the glass surface and act as feedback for the control

[10] (Squilbin, J. M.; Moulart, M.; CNUD, 1995)

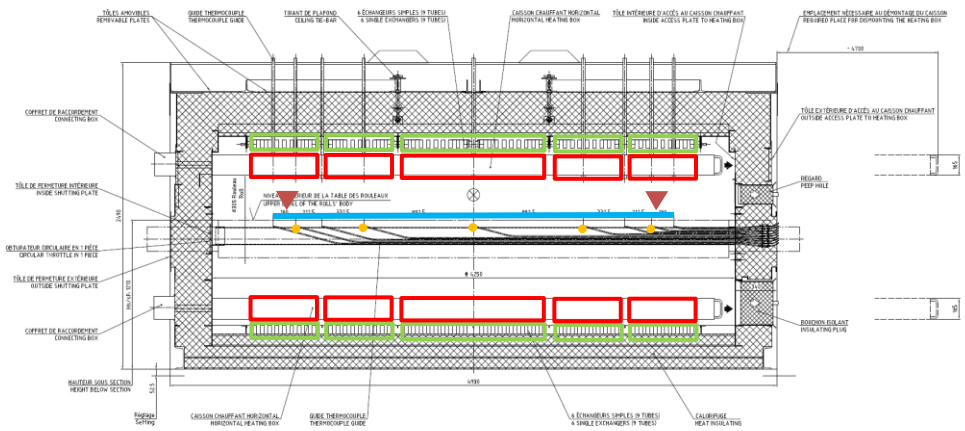

Figure 2.13: Transversal section of the annealing lehr. Electrical resistance (red), air heat exchanger (green), thermocouple position (orange), burners (brown), glass (light blue). The square pattern indicates the insulated structure of the lehr.[11]

The non-insulated part is a closed chamber divided into zones where the glass ribbon is cooled down with a controlled temperature and there is only air heat exchanger; in this part the glass is solid and the stress applied is temporary.

After the complete cooling of the glass ribbon and the cutting phase, that transform the glass ribbon into single sheets, the permanent stress measurement are performed. The instrumentation used for this operation is a circular polariscope moved in a specific position of the sheet width.

[12]A polariscope is an instrument that enables one to analyze the character of birefringence in photoelastic specimens. The simplest arrangement consists of a model placed between two linear polarizers, called the polarizer and analyzer, respectively, with their axes crossed as shown in Figure 2.14. This is the arrangement for a plane, dark-field polariscope.

[11] (Squilbin, J. M.; Moulart, M.; CNUD, 1995)

[12] (Aben & Guillemet, 1993)

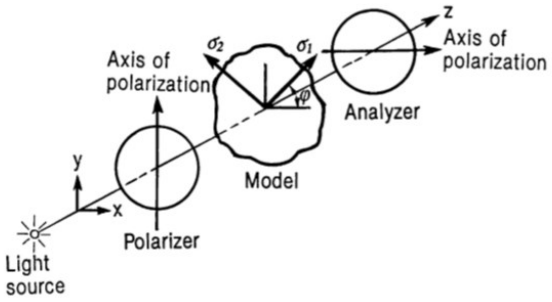

Figure 2.14: Photoelastic model in a plane polariscope.[13]

There are two separate conditions under which the extinction of the light will be obtained. One condition is that $\psi = 0$ or 90°. This is satisfied by all points on the plate where the directions of the principal stresses are parallel to the axes of the polaroids and such points appear to be dark. In general, these points lie on continuous curves forming a system of dark bands named isoclines. For any given setting of the crossed polaroids, a corresponding isoclinic pattern can be observed. The isoclinic pattern is independent of the wavelength and the phase difference and hence of the magnitude of the stresses. The second condition which extinction is obtained is that the phase is equal to $2\pi n$ radians where n is zero or any integer. This is equivalent to relative retardation of n wavelengths. Thus, all points on the plate at which the difference $\sigma_1 - \sigma_2$ in the principal stresses is such that the relative retardation produced is equal to a whole number of wavelengths, will appear dark. In general, the difference in the principal stresses varies continuously within the plate so that the loci of such points are smooth curves. These are known as isochromatics. Since the phase difference depends on the wavelength, it follows that the points of extinction for the light of different colours lie on different curves. Thus, when white light

[13] (Aben & Guillemet, 1993)

is used, all wavelengths of which the relative retardation at a given point is an integral number will be extinguished.

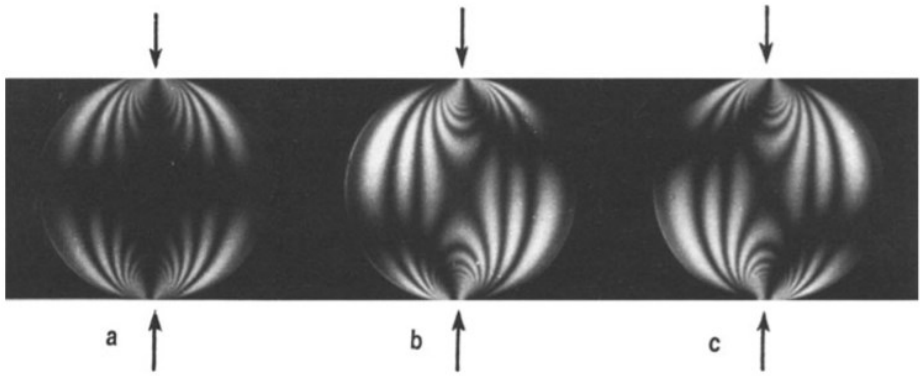

Figure 2.15: Dark-field images of a disk under diametral compression with different angles ψ between polarizer axis and direction of loading: (a) ψ = 0°; (b) ψ = _20°; (c) ψ = +20°.[14]

From the above analysis we see that when a stressed plate is placed in the field of a plane polariscope, two different systems of fringes appear simultaneously, namely, the isoclines and the isochromatics. If white light is used, these can easily be distinguished from one another since the isoclines are black while the isochromatics, except for fringes of zero-order, are coloured. However, with monochromatic light, both sets of fringes are black and somehow disturb each other. Besides, all the required information can frequently be obtained from the isochromatic pattern alone. The isoclines are then undesirable since they tend to obscure the stress pattern. In such cases, the isoclines can be eliminated by using circularly polarized light. The directional characteristic of the light incident on the model on which the formation of isoclines depends is thus removed. Figure 2.16 shows the arrangement of a dark-field circular polariscope with two quarter-wave plates placed between the polarizer and analyzer. when the circular polariscope is set to give a dark background, the intensity transmitted is zero when the phase is equal to 2π, corresponding to relative retardation equal to a whole number of wavelengths. This is the same

[14] (Aben & Guillemet, 1993)

as the second of the conditions for extinction obtained in the plane polariscope so that the isochromatic pattern is the same in each. This, however, is the only condition for extinction in the circular.

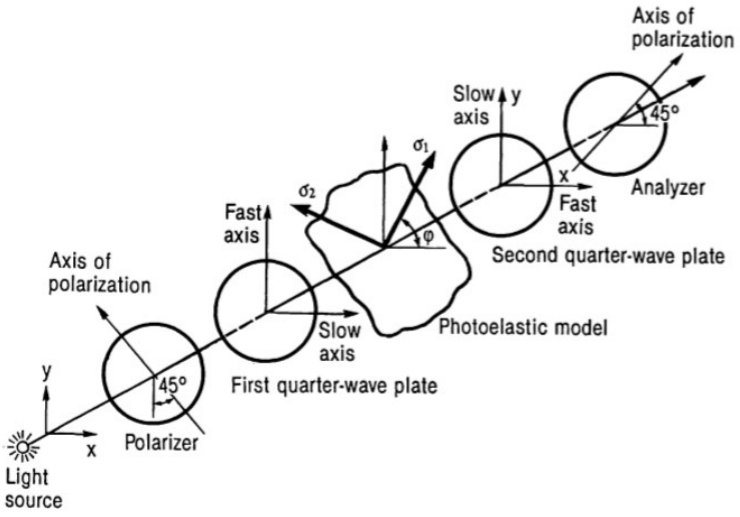

Figure 2.16: Photoelastic model in a dark-field circular polariscope.[15]

With the light-field circular polariscope, extinction occurs when the phase is equal to (2n+1) π where n is zero or integer. This corresponds to relative retardation equal to an odd number of half wavelengths. Thus, in a circular polariscope, the number of points at which the relative retardation can be determined directly from the iso-chromatics is doubled by viewing the model with the analyzer set to give dark and bright background in turn. As an illustration, in Figure 2.17 dark-field and light-field isochromatic images of a disk in diametral compression are shown.

[15] (Aben & Guillemet, 1993)

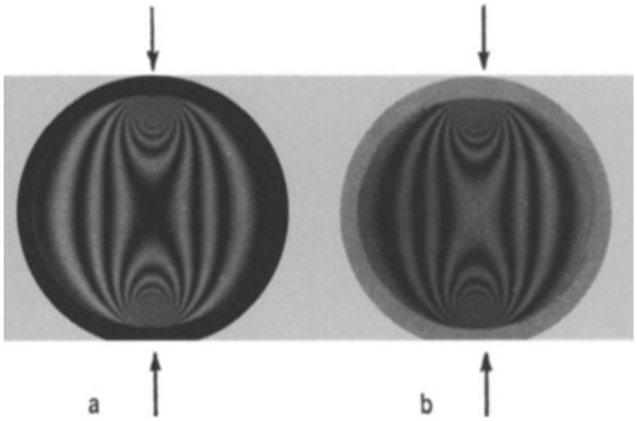

Figure 2.17: Dark-field (a) and light-field (b) isochromatic images of a disk under diametral compression.[16]

2.6. System

From a schematic point of view and for a modelling purpose, all the process and the devices that work in the described production system can be seen in terms of functional blocks. This method simplifies the way that the system is studied and highlights the interaction of the parts of the system with each other.

[16] (Aben & Guillemet, 1993)

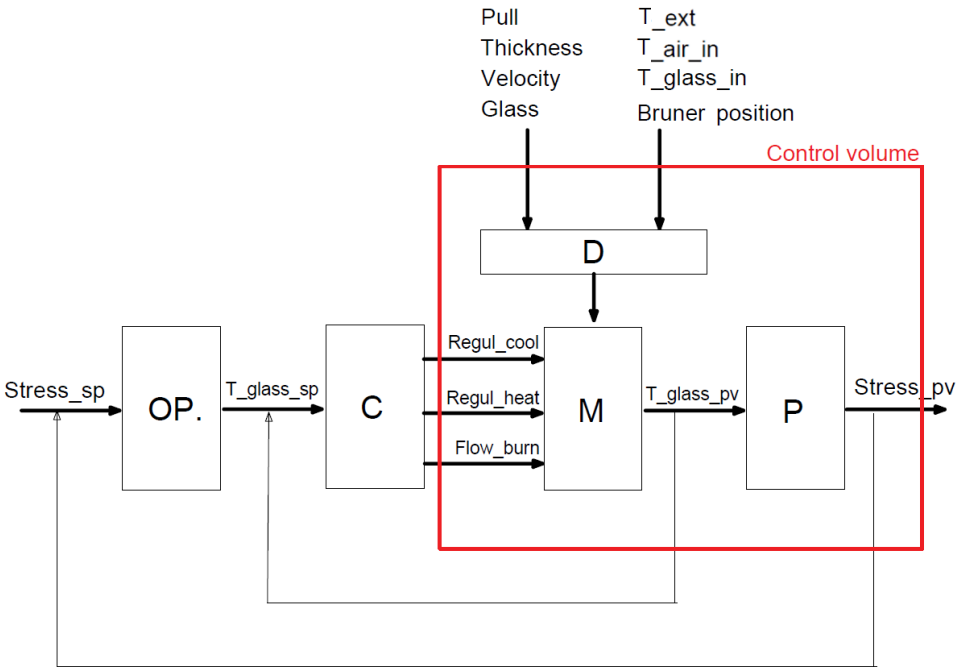

Figure 2.18: Functional block diagram of the studied system. The blocks are Operator function (OP), control (C), machine (M), glass behaviour (P), external conditions (D).

In a simplified way, the system can be described as shown in Figure 2.18; as mentioned in the previous sections, all the production is based on the control of the permanent stress impressed on the glass sheet, so it can be defined the following blocks and variable:

- **Stress_sp**: is the ideal distribution of permanent transversal stress that want to be obtained on the glass sheet

- **OP:** the operator, in order to obtain the desired distribution, provides a distribution of temperatures **T_glass_sp** to input into the system control

- **C:** is the temperature control of the system; the control receives a setpoint of temperature in 5 points across the ribbon width and by a

PID control regulate the 5 sections of cooling by acting on the valves opening **Regul_cool**

- **Regul_heat** and **flow_burn** are controlled manually and act on the heating electrical resistances and the edge burners

- **M:** once the setpoint of regulation of the machine is given, it generates a setting that causes a level of thermal power per each of the 5 sections across the ribbon width; since these sections are adjacent they affect the glass of the close section. The thermal power – whether in cooling or heating – causes the glass to get a certain temperature **T_glass_pv,** this temperature distribution measured by 5 thermocouples at each section provide to the control a feedback of the glass state.

- The temperature distribution on the glass **T_glass_pv** – that is very close to the temperature measured by the thermocouples – and the visco-elastic behaviour of the glass during the annealing phase, generate a continuous distribution of stress. The edge of the ribbon will be trimmed next; now the glass has a shape of a sheet of a precise rectangular shape, and the distribution of stress changes accordingly to the removal of the edges. The permanent transversal stresses are then measured with a correction factor in case the temperatures of the glass are not perfectly in isothermal and standard conditions to get the **Stress_pv**, that is the general output or feedback of the system. The feedback is generated by the Quality Control, and according to a specific interval of acceptance of the stress variation, evaluates if the product is compliant to the quality level required as shown in Figure 2.19. This information is sent to the operator controlling the process, who compensate for an eventual deviation in the output level.

- Some other conditions and settings affect the generation of the system's output. Some of them are considered part of the production setting, such as the **Pull** – meant as the productivity of the line -, the **Thickness** of the glass sheet produced, the **Velocity** of the line and the kind of **Glass** produced. Other important variables are the temperature of the glass outcoming from the forming phase **T_glass_in**, the **Burner position**, the environmental temperature **T_ext** and the temperature of the air incoming in the heat exchanger **T_air_in**.

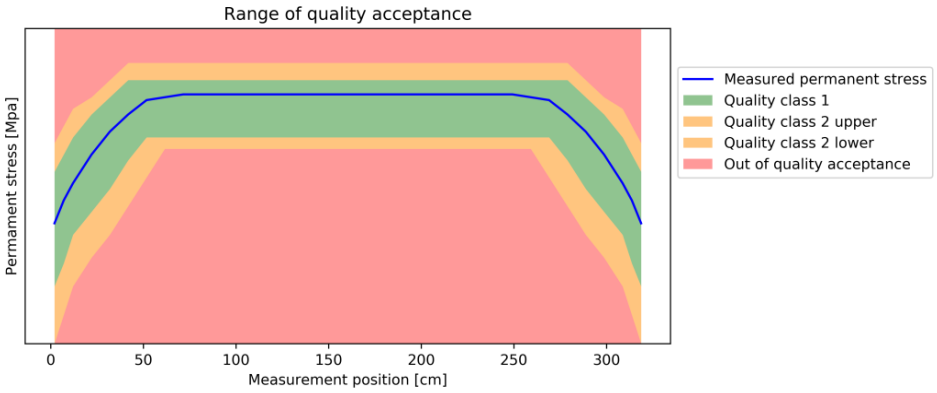

Figure 2.19: Quality acceptance levels and example of permanent stress measurement across the sheet width.

The system that will be analyzed and simulated is inside the red line in Figure 2.18. In every moment all the above-mentioned variables assume a state or a value that characterize the operation of the system; the goal of this study is to detect patterns and correlations between the input variables of the system analyzed, and the output variables.

In the next chapter, it will be explained the characterization of the model and all the steps that lead from the raw data collected on-line to a working and

validated model, that is able to simulate the system's output considering all the conditions and settings that it assumes.

CHAPTER 3

EXPERIMENTAL MODEL DEVELOPMENT: A DATA ANALYSIS APPROACH

3.1. Introduction

The model developed fits into an industrial and production context where the study of systems and the correlation of process variables is essential to increase the level of in-depth knowledge of the operation of the machines and the individual process sections. This in-depth study has as its objective, a better knowledge of the production process in terms of the correlation between the production variables and characterization of the functioning of equipment, machines and physical phenomena related to them; as a direct consequence, it is possible to obtain operational results in terms of optimization of yield and production efficiency and therefore of cost reduction, energy-saving and control over the stability of quality output.

In developing this model, the starting point is the context of automation and orientation to issues related to the *Industry 4.0*; from this point of view, the presence of an integrated data infrastructure that connects different departments of the production line, from automation management - and therefore from the field instrumentation and controls - through process control to the Data Analysis that allows, as in the case of this work, to provide predictive models that improve production conditions. To this purpose, the company *AGC Glass Europe*, where this Master's degree thesis was carried out, has prepared a series of actions that allow the development of this policy.

As anticipated in Chapter 2, the development of this model has as its topic the study of a production system - meant as the union of machines and devices, and the reaction of the product under the action of some physical phenomena - in order to build a predictive model that is able to predict the process output as a function of different input parameters, such as the

regulation of the actuation devices, the initial conditions of the product and some external conditions. The outcome of the model is to provide a simulated result that is as close as possible to the production result actually measured (the accuracy of the predictive model is calculated in that way); after having built the forecast model and validated its precision, the model can be used to simulate the quality output in totally new production conditions, based on what "learned" in the training phase. Having a simulated process output available can be used in different ways at the operational level: firstly it can be provided to the process control operator to frequently obtain the result of the process even before the quality control and here identify any deviations that cause the qualitative output to vary outside a predetermined range; subsequently, the operator will make changes to the system regulation to balance this deviation. Another interesting point of use of this model is an engineering application; as explained in the previous chapter, the float glass production process is continuous and any action performed on the line, if not properly designed and programmed, can have a critical impact on the stability of the process and on the product itself. This said, the model acts as a support for the simulation of new operating conditions and machine settings, in order to test the effect they could have; once the conditions have been tested and the outcome of the test validated, these new adjustments can be applied and verified on the production line under standard production conditions.

In order to build this predictive model, a black box study method was used. In systems theory, a black-box model is a system that, similar to a black box, can be described substantially in its external behaviour, i.e. only for how it reacts in output to certain input solicitation, but whose internal behaviour is not fully mathematically defined. This definition arises from the consideration that in the analysis of the system what is important at a macroscopic level or for practical purposes is the external behaviour, especially in a context of the interconnection of several systems, rather than the internal functioning whose result is precisely the external behaviour. This type of approach gives good results and is particularly useful in the case of very complex processes, where

for the characterization of each component and phenomenon, the identification of transfer functions is impractical or subject to very limiting simplification hypotheses.

Ultimately, for the characterization of this black box, a Supervised Machine Learning method was used, which correlates several process variables each other using a regression algorithm (SVM, Support Vector Machine). To move from raw data collected by field instrumentation to a predictive model that can use this data to provide simulated output, the following steps must be achieved:

- Data collection
- Data pre-processing
- Model selection
- Model training
- Parameters tuning
- Model validation and testing
- Prediction and results

Figure 3.1: Steps of the Machine Learning model development.

3.2. Data collection

In the production system under consideration, the availability of data derives from a series of data sources, these devices consist of sensors that generate a signal that is a function of the measured parameter. These data have a completely different nature from each other, based on the phenomenon being measured, the type of sampling, and the nature of the variable. At the field level, a classification can be as follows:

- **On-line**: this instrumentation is permanently installed on a machine and generates a signal read by the control system, in addition to being used for process regulation, this data is collected and stored through a data infrastructure on a cloud database. The value of this variable is automatically sampled according to a certain frequency and some archiving optimization functions. To this group belong all the sensors that measure the input variables of the system under consideration, for example temperature, position, size, speed, flow rate and power.

- **Off-line:** this instrumentation is installed outside the production line and produces data (the data is sampled) in a non-continuous manner and according to a variable sampling frequency. Given the characteristic of the measuring principle, this instrument is installed on an automatic handling system for the sampling of different points of the glass sheet. The measured data are analyzed and corrected according to the temperature of the glass; the final value is passed directly through an acquisition system and is stored on a cloud database.

As mentioned some variables have a different sampling period, depending on the use in terms of process control: for example, the temperature control of the glass ribbon in the annealing tunnel requires a period of a few seconds, as the value provides feedback to the PID control for the management of several actuators. Another relative subdivision sees an automatic sampling, where the

control system automatically collects a certain value which is then stored, and a manual one, where the reading of the data is provided by a person who reports the value directly to the collecting system. At the control level, the nature of the variable can be divided into:

- **Setpoint**: it is a value processed by the control and it is the state that the controlled variable must assume, this value is passed to an actuator that realizes this state.

- **Feedback or PV**: it is the actual value that the variable assumes in a certain instant. The deviation from the setpoint defines an error and therefore a reaction by the control. In certain cases, it can be similar to the setpoint, if the dynamic behaviour of the actuator has a short period and therefore the error in the transient phase is close to zero. A case in which feedback and setpoints are considered equal is that of the actuators of the air control valves of the heat exchangers.

3.3. Data preprocessing

Once the data acquisition system has been created, which in this case had already been present and guaranteed the presence of more than 2 years of historical measurements, it is time to define which variables need to be used for the creation of the model. In this case, the variables represented in the following table have been extracted, that is:

Variable	Phenomenon measured	UOM	Kind of variable	Sampling
Electrical power heating resistances	Ribbon heating	kW	FB	AUTO
Valve opening of heat exchanger's air	Ribbon cooling	%	SP	AUTO
Burner gas flow	Edge ribbon heating	Nm3/h	FB	MAN
Glass ribbon initial conditions	Glass temperature	°C	FB	AUTO
Stress on glass sheet	Transversal permanent stress	Mpa	FB	AUTO
Ambient condition	Air temperature near lehr	°C	FB	MAN
Pull	Daily productivity of the line	tons	SP	AUTO
Thickness	Thickness of the glass ribbon	mm	FB	AUTO
Velocity	Ribbon speed	m/min	FB	AUTO
Glass	Kind of glass produced	-	SP	AUTO

FB: Feedback
SP: Setpoint
AUTO: Automatic
MAN: Manual

Table 3.2: Variables present in the dataset, each variable represent the data coming from a different point of the process.

Some of the variables present in Table 3.2, collect data from a different point of the process, meaning that each of them has more sensors that collect data. E.g., as explained in Chapter 2 about the lehr technology, the cooling heat exchangers are present in the zone A and B, on top and bottom surface of the ribbon and they have a regulation in 5 points across the ribbon width; this means that is passed to the model building 20 features that represent the cooling in a different point of the process.

For clarity purpose, it is reported the system diagram that shows where the variables are working.

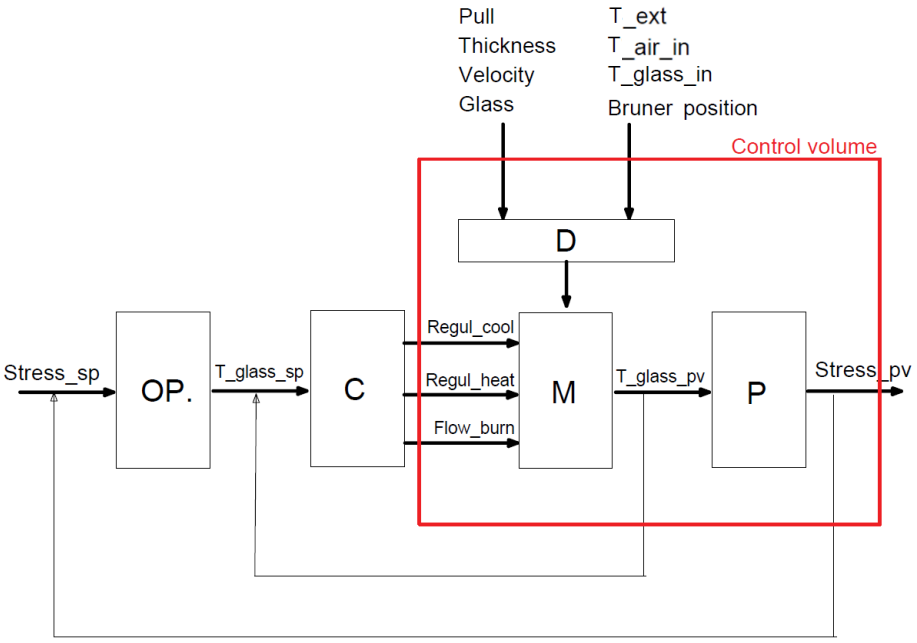

Figure 3.3: Functional block diagram of the studied system. The blocks are Operator function (OP), control (C), machine (M), glass behaviour (P), external conditions (D).

3.3.1. Choose of the sampling time

From a first analysis of the present database, some substantial differences can be noticed on the sampling of the data; the first really important point is the definition of the sampling period. As seen, each variable has a different nature and is sampled according to a different frequency by the data collection system. Looking at the entire system, we note that the sampling of the output is the least dense (it requires longer intervention times since it is an operation performed manually) and therefore the historical data present are less than the other variables. This means that the output sampling period is also 2 orders of magnitude higher than the process variables and occurs at evenly spaced intervals; this characteristic is normal since the last ones are necessary for the process control (which therefore has very short characteristic periods), while the output value is a control of the final sheet and it is needed to guarantee the quality of the product sold. Also for this reason it has not been considered appropriate to interpolate the output values to obtain more intermediate values.

Ultimately, the acquisition period of the output variable was chosen as the sampling time for all the variables of the system; therefore, in correspondence with the sampling of the output, all the input variables are correlated to it. However, the input variables are not taken only at that particular moment, but their value is averaged over a wider evaluation interval, to have a better description of the value itself. An example is the one in Figure 3.4, where a series of signals that vary over time are represented over a period of 16 hours, their sampling time are equal to 10 s and therefore graphically a continuous and representative trend of the signal shape is obtained; the moment in which the output is sampled is shown in red: as mentioned, the sampling intervals of this variable are not constant and are much more spaced from each other than the other signals. The interval before the sampling of the output lasting t_s=30 minutes is shown in yellow, within it each input variable is evaluated; the average value in this range is taken as the value that contributed to the generation of the output and is therefore related to it. Each of these samples (output and average of the input in the interval t_s) generates a row of the dataset, which will be used afterwards by the regression algorithm to build the model.

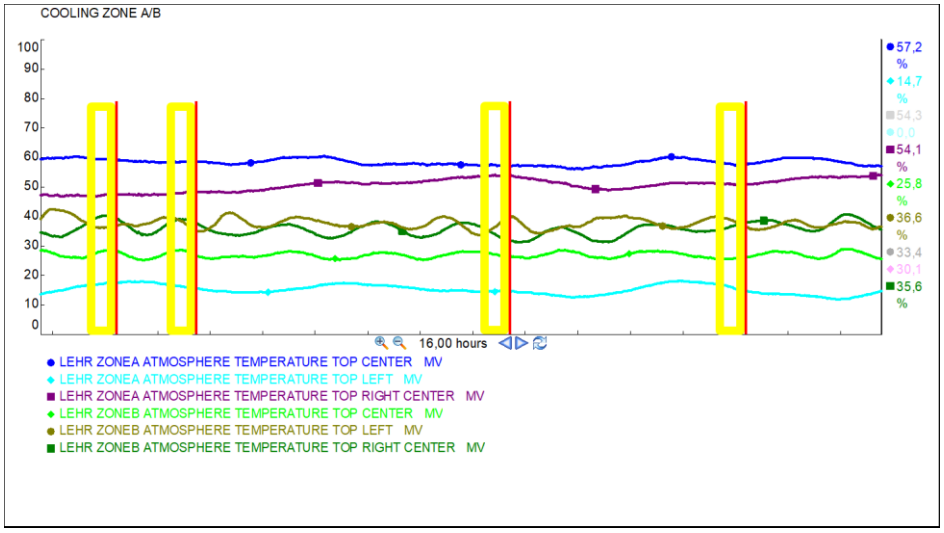

Figure 3.4: Time evolution of the value of some variables (continuous coloured curves), sampling of the output (red straight line) and interval of calculation of the input variables (yellow).

For the evaluation of the size of the calculation interval t_s of the input variables, some fundamental indicators were considered to describe the variation of the input data in the interval itself. These indicators are provided directly by the data extraction software and are:

- The standard deviation of the data samples
- Range of variation of the data samples
- Number of data samples
- Percentage of goodness in collecting the value

For each variable, these parameters were monitored to not exceed a certain threshold, established based on the physical phenomenon of reference, the tractability of the data and the experience of the production technicians. Different lengths of the chosen interval were compared, between ts = 5, 30, 120 minutes. Here are some examples of evaluations carried out on the most critical variables, to compare the three proposed duration classes:

1	**Tag**	C1_LR_ZA_ATM_TEMP_TOP_C.MV			**Tag attributes:**		
	start sorting	01/01/2018			scan time [class / s]	2	10
	end sorting	01/03/2020			compdev	0	
	time interval	**120 m**			excdev	0	
	n° of values	9480					
	n° of values usable (numbers)	100,0%					
			value	**good**	**dev.st**	**value interval count**	**value interval range**
	min		3,5	14	0,000	0	0,000
	max		100,0	100	38,972	723	100,000
	mean		65,4	100	**1,157**	**203,8**	**3,585**
	sorting condition (higher than)			99	0,500	5	0,500
	n° of values sorted			100%	**37%**	**37%**	**43%**
2	**Tag**	C1_LR_ZA_ATM_TEMP_TOP_C.MV			**Tag attributes:**		
	start sorting	01/01/2018			scan time [class / s]	2	10
	end sorting	01/03/2020			compdev	0	
	time interval	**30 m**			excdev	0	
	n° of values	37920					
	n° of values usable (numbers)	100,0%					
			value	**good**	**dev.st**	**value interval count**	**value interval range**
	min		0,0	13	0,000	0	0,000
	max		100,0	100	46,649	184	100,000
	mean		65,4	100	**0,324**	**50,9**	**1,077**
	sorting condition (higher than)			99	0,500	5	0,500
	n° of values sorted			100%	**19%**	**31%**	**32%**
3	**Tag**	C1_LR_ZA_ATM_TEMP_TOP_C.MV			**Tag attributes:**		
	start sorting	01/01/2018			scan time [class / s]	2	10
	end sorting	01/03/2020			compdev	0	
	time interval	**5 m**			excdev	0	
	n° of values	227520					
	n° of values usable (numbers)	100,0%					
			value	**good**	**dev.st**	**value interval count**	**value interval range**
	min		0,0	33	0,000	0	0,000
	max		100,0	100	25,733	32	74,000
	mean		65,4	100	**0,056**	**8,5**	**0,192**
	sorting condition (higher than)			99	0,500	5	0,500
	n° of values sorted			100%	**2%**	**29%**	**12%**

Table 3.5: Evaluation table for the data extraction, the parameters reported are needed to evaluate the interval time t_s according to the standard deviation, number of datapoints, value range and sampling quality.

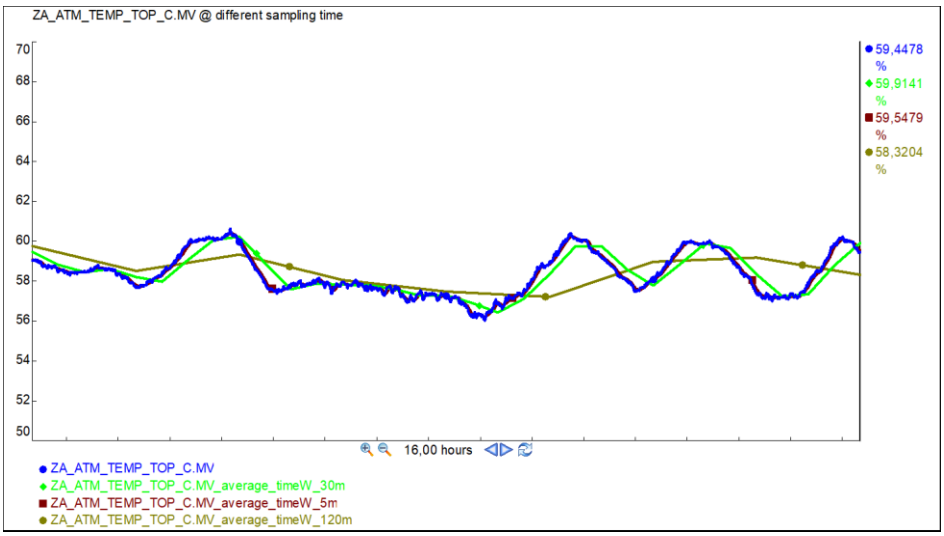

Figure 3.6: Trend of a cooling valve regulation (blue line) and comparison with a different sampling time: 5' (dark red), 30' (green) and 120' (dark green).

As can be seen from this graph, the interval of t_s = 5 min better approximates the value of the variable (in fact the two curves are almost overlapped), on the contrary, the case of t_s = 120 min seems to be not very representative of the signal trend over the time, while the interval of t_s = 30 min is the most suitable for the representation of the signal considered.

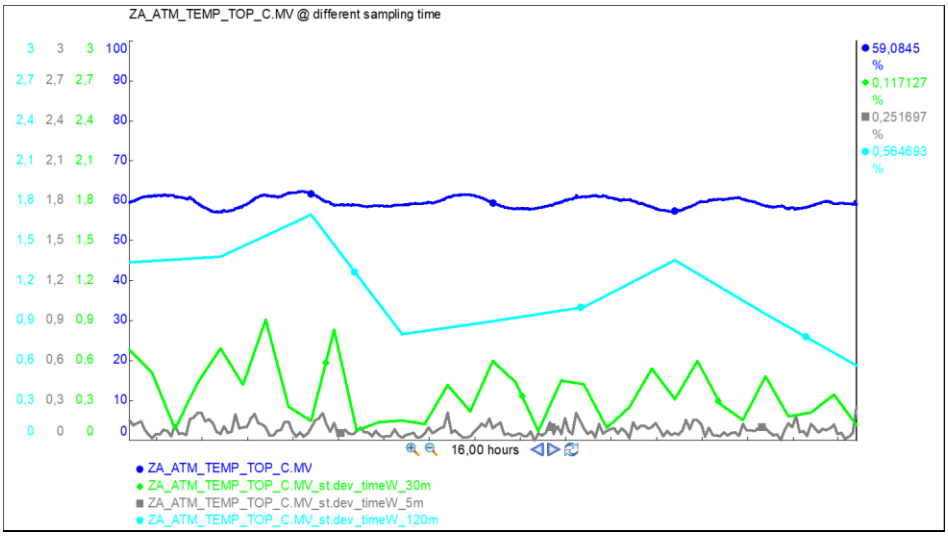

Figure 3.7: Trend of a cooling valve regulation (blue line) and comparison with standard deviation at different sampling time: 5' (grey), 30' (green) and 120' (light blue).

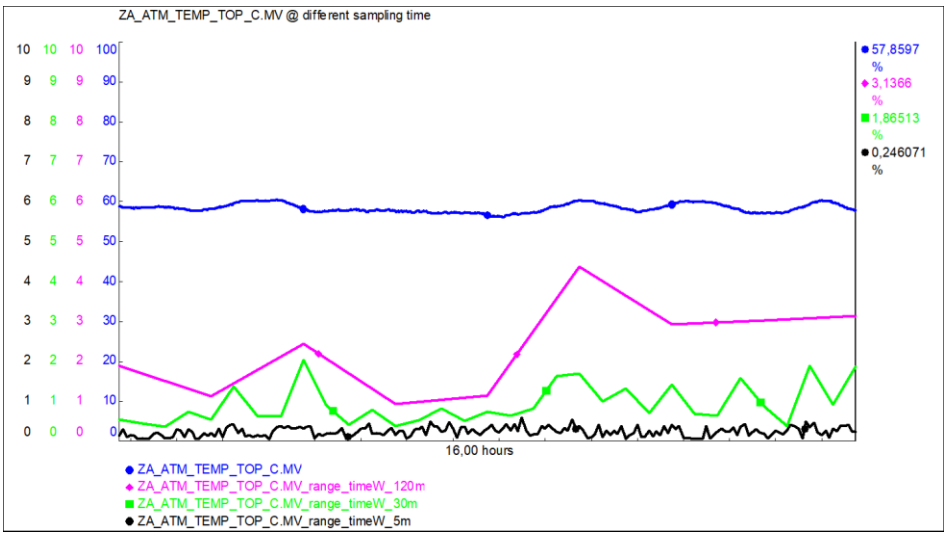

Figure 3.8: Trend of a cooling valve regulation (blue line) and comparison with value's range at different sampling time: 5' (black), 30' (green) and 120' (purple).

For the standard deviation too, as it is reasonable to expect, in the case t_s = 120 min the standard deviation is much higher than in the case t_s = 5 min since has been considered a process condition that occurs for longer and therefore it has greater variability. Precisely for this reason, it is necessary to choose a not too high t_s, otherwise the average data is no longer representative of a specific condition. The range parameter, on the other hand, indicates the variation between the minimum and maximum values in the interval t_s, and therefore less representative compared to the standard deviation.

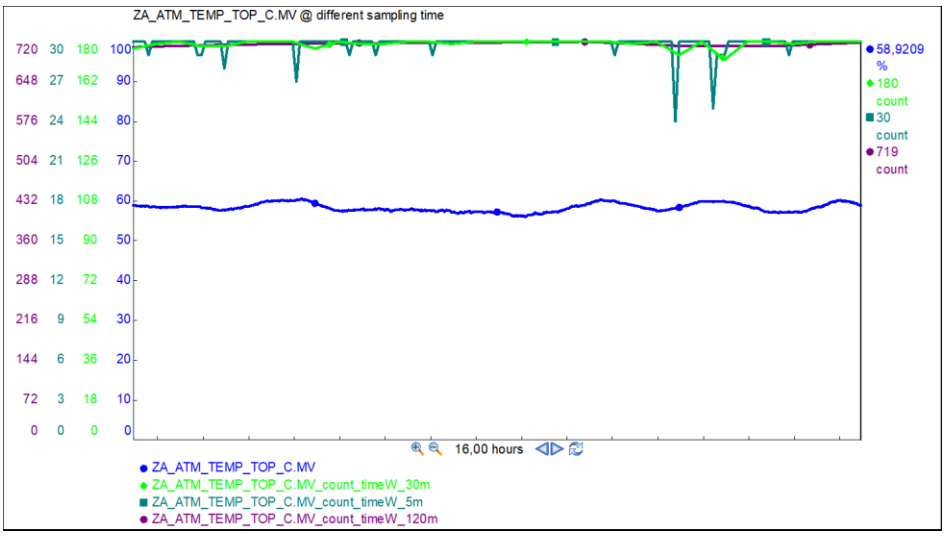

Figure 3.9: Trend of a cooling valve regulation (blue line) and comparison with datapoints counter at different sampling time: 5' (dark cyan), 30' (green) and 120' (violet).

As for the number of values sampled at the field level, these remain at a very high level, therefore it has been not necessary to exclude any interval due to the quality of the sampling itself. From the graph shown, it can be seen that the case t_s = 5 min obviously shows a greater variability of the sampled datapoints, precisely because in the case of non-sampling, they are present in a shorter interval, and therefore with a smaller number of total datapoints.

3.3.2. General description of the data

In addition to the consideration done in the previous paragraph, a graphical presentation is useful to understand the data. One can be plotting the box and whisker plot to graphically depict groups of numerical data through their quartiles. Box plots may also have whiskers, lines extending from the boxes that indicate variability outside the upper and lower quartiles.

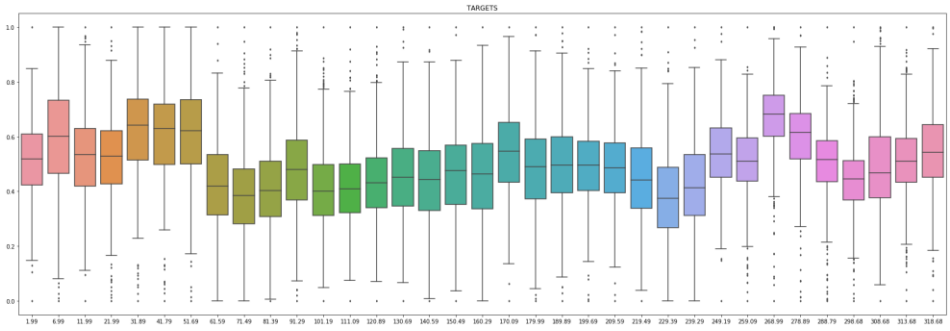

Figure 3.10: Box and whiskers plot of the output variables, values are normalized into a 0-1 interval.

Another plot used to highlights the distribution of a certain vector of value is the distribution plot, that is useful for later evaluation on the preprocessing phase and model selection.

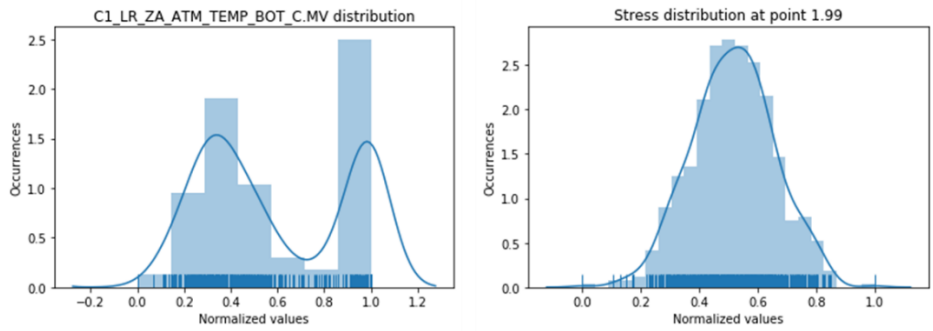

Figure 3.11: Distribution plot of one of the input and output variables; values are normalized into a 0-1 interval.

3.3.3. Outliers

Many applications need to be able to decide if a new observation belongs to the same distribution as existing observations (inlier), or should be considered differently (outlier). The dataset of samples contains outliers that are defined as observations that are "far" from the others. Outlier detection functions try to fit the areas where the training data is the most concentrated, ignoring the deviating observations. Outlier detection is used for anomaly detection, where one is interested in detecting abnormal or unusual observations; is then also

known as unsupervised anomaly detection. In the context of outlier detection, the anomalies cannot form a high-density cluster and the estimator assumes that they are located in low-density regions.

The Scikit-learn library provides a set of machine learning tools that can be used for this detection. This strategy is implemented with objects learning in an unsupervised way and the predict method makes use of a threshold on the raw scoring function computed by the estimator.

An efficient way to detect outliers is to use the Local Outlier Factor (LOF) algorithm. The `neighbors.LocalOutlierFactor` (LOF) algorithm calculates a score reflecting the degree of the anomaly of the observations, reflecting its status of outlier. It is done measuring the local density deviation of a given data point with respect to its neighbors. The purpose is to detect the observations that have a substantially lower density than their neighbors.

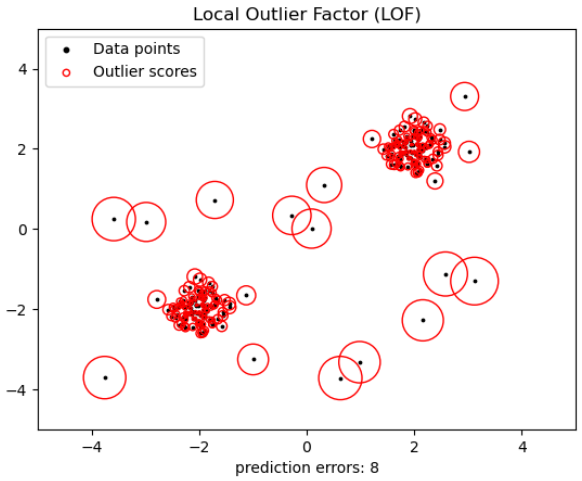

Figure 3.10: LOF method applied to detect outliers, circles around the black points are the score given to the single datapoint to evaluate the presence of an outlier.[17]

[17] (Scikit-Learn, s.d.)

Practically, the local density is obtained from the k-nearest neighbors approach. The LOF score of a sample is equal to the ratio between the average local density of his k-nearest neighbors, and its own local density; a normal observation is expected to have a local density similar to its neighbors, while abnormal data are expected to have a smaller local density. The number k of neighbors considered, must be is chosen as follow:

1) greater than the minimum number of samples that a cluster has to contain so that other samples can be local outliers relative to this cluster

2) smaller than the maximum number of near samples that can potentially be outliers

Generally, these informations are not available and taking n_neighbors=30 has given the best result in term of model performances. What makes the LOF algorithm efficient is that it takes local and global properties of datasets in consideration and it can perform well even in datasets where abnormal samples have different underlying densities. From a general point of view, the question is not how isolated the sample is, but how isolated it is with respect to the surrounding neighbourhood.

This strategy is illustrated in Figure 3.10 and is applied at this model with the following script:

```
#FIND OUTLIERS WITH METHOD OF LOCAL OUTLIER FACTOR

lof = LocalOutlierFactor( n_neighbors=30)
y_pred_lof = lof.fit_predict(data_scal)
lofs_index = np.transpose(np.array(np.where(y_pred_lof!=-1)))[:,0]
data_scal_outl = data_scal[lofs_index,:]
target_scal_outl = target_scal[lofs_index,:]

plt.figure(figsize=(10,30))
sns.boxplot( data=pd.DataFrame(data_scal_outl), orient = "h", linewidth=1.5, fliersize=2.5 ).set_title("INPUT WITHOUT OUTLIERS")
plt.show()

plt.figure(figsize=(30,10))
sns.boxplot( data=pd.DataFrame(target_scal_outl,columns=target_pos), linewidth=1.5, fliersize=2.5 ).set_title("OUPUT WITHOUT OUTLIERS")
plt.show()
```

Figure 3.11: Python script of the implementation of the LOF method.

There is also another method that has been tested for this purpose, but with lower results respect to the LOF method; is the case of the InterQuartile Range (IQR). Not all data distributions are normal enough to be treated as a Gaussian distribution. The IQR function is calculated as the difference between the 75th and the 25th percentiles of the data and defines the box in a box and whisker plot.

The IQR method can be used to detect outliers by defining limits on the sample values outer the interval of k*IQR below the 25th percentile or above the 75th percentile. A common value for the factor k is the value 1.5. A factor k of 3 or more can be used to identify values that are extreme anomalies when described in the context of box and whisker plots. On a box and whisker plot, these limits are drawn as fences on the whiskers that are drawn from the box. Values that fall outside of this interval are drawn as dots.

As mentioned this method has been implemented in the following way:

```
#FIND OUTLIERS WITH METHOD OF IQR

k_IQR = 1.5
q25 = np.zeros(len(data_scal[0,:]))
q75 = np.zeros(len(data_scal[0,:]))

for i in range (len(data_scal[0,:])):
    q25[i] = np.percentile(data_scal[:,i], 25)
    q75[i] = np.percentile(data_scal[:,i], 75)

IQR = q75 - q25
cut_IQR = k_IQR*IQR
low_IQR = q25 - cut_IQR
up_IQR = q75 + cut_IQR

iqr_index = np.zeros(len(data_scal))
iqr_cut = np.zeros(len(data_scal))

for i in range (len(data_scal)):
    for j in range (len(data_scal[0,:])):
        if data_scal[i,j]>up_IQR[j] or data_scal[i,j]<low_IQR[j]:
            iqr_cut[i] = -1

iqr_index = np.transpose(np.array(np.where(iqr_cut!=-1)))[:,0]

data_scal_outl = data_scal[iqr_index,:]
target_scal_outl = target_scal[iqr_index,:]

plt.figure(figsize=(10,30))
sns.boxplot( data=pd.DataFrame(data_scal_outl), orient = "h", linewidth=1.5, fliersize=2.5 ).set_title("INPUT WITHOUT OUTLIERS")
plt.show()

plt.figure(figsize=(30,10))
sns.boxplot( data=pd.DataFrame(target_scal_outl,columns=target_pos), linewidth=1.5, fliersize=2.5 ).set_title("OUPUT WITHOUT OUTLIERS")
plt.show()
```

Figure 3.12: Python script of the implementation of the IQR method.

3.3.4. Features scaling

Feature scaling is a method used to normalize the range of independent variables or features of the dataset; since the range of values of raw data varies into different interval, some machine learning algorithms will not be able to learn properly without normalization. E.g., many classifiers are based on the calculation of the Euclidean distance between two points; if one of the features has a wider range of values, the distance will be affected by this particular feature. Therefore, the range of all features should be at the same variation range so that each feature contributes proportionately to the final distance; it is then necessary to transform all the variables into the same scale. There are several ways of scaling the data, two of them are Normalization and Standardization.

The first rescale the variable into a range calculated with the minimum and maximum value that the variable itself can assume, with the following formula:

$$x' = \frac{x - min\ (x)}{\max(x) - \min\ (x)}$$

Where x' is the transformed value and x is the original value (or original array). The transformed array will therefore vary within a range of 0 and 1.

The second method is the standardization rescale the variable into a range calculated with a mean value equal to zero and a standard deviation equal to 1, with the following formula:

$$x' = \frac{x - \mu}{\sigma}$$

Where σ is the standard deviation and μ is the mean value.

Both methods have been applied to this model and tested according to the performances score, following this script:

```
#VALUES RESCALING WITH NORMALIZATION

min_max_scaler = preprocessing.MinMaxScaler()
data_scal = min_max_scaler.fit_transform(data)
target_scal = min_max_scaler.fit_transform(target_pull)

#VALUES RESCALING WITH STANDARDIZATION

std_scaler = preprocessing.StandScaler()
data_scal = std_scaler.fit_transform(data)
target_scal = std_scaler.fit_transform(target_pull)
```

Figure 3.13: Python script of the implementation of the Standardization and Normalization scaling methods.

3.3.5. Training / Test dataset split

The train-test split is a method needed to provide a consistent dataset to the training phase and another for evaluating the performance of a machine learning algorithm. The procedure consists of taking a dataset and dividing it into two subsets: the first subset is used to fit the model; the second subset is not used to train the model but instead, the input features of the dataset are provided to the model, then predictions are done and compared to the expected values.

- **Training dataset**: fitting of the approximate function by regression task

- **Testing dataset**: evaluating the fitting of the model

The goal is to estimate the performance of the machine learning model on new data, so that was not used to train the model, that will be then the real working conditions of the model. I.g., to fit it on available data with known inputs and outputs, then make predictions on new examples in the future conditions where there are not the expected output or target values. These subsets must be enough big to be a suitable representation of the problem domain, indeed the original dataset must also be a suitable representation of the problem domain, meaning that there are enough records to cover all common cases and most uncommon cases in the domain, and even combinations of input variables observed in practice.

The procedure has one principal configuration parameter, which is the size of the train and test sets, for example, it has been chosen training set with the size of 0.80 means that the remaining percentage 0.20 is assigned to the test set. That split percentage should be chosen according to the project's objectives and considerations that include:

- Computational cost in training the model
- Computational cost in evaluating the model
- Training set representativeness
- Test set representativeness

Furthermore, to maintain the chronological distribution of the data, the shuffle of the data samples into the two datasets has not been done; these operations can be simply implemented in Python with the following code:

```
213  #SPLIT TRAINING AND TEST SET
214
215  x_train, x_test, y_train, y_test = train_test_split(data_scal_outl, target_scal_outl, test_size=0.2, shuffle = None)
```

Figure 3.14: Python script of the implementation of the train and test dataset split.

3.4. Model selection

Model selection is the process of selecting one final machine learning model from among a collection of machine learning models given from the available libraries. Model selection is a procedure used to evaluate both upon different types of models and the same model configured with different hyperparameters.

In the specific case, different algorithms were tried for the realization of the model and subsequently tested to choose the final one that guarantees the best results in terms of accuracy; the tested models are the following:

- Linear regression
- Polynomial regression
- Support Vector Machine (SVM) for Regression task

The one that guaranteed the best results was the last one: the Support Vector Machine, on which the next description will focus.

3.4.1. Support Vector Machine (SVM) for Regression

Support vector algorithms are commonly used for classifications of data in SVMs but it can be used in regression models too; in regression models, it is used as Support Vector Regression.

If explained in the most layman terms support vector regression works on the principles of SVM but with a few alterations. Here the variables on which model is applied are numerical and not categorical. SVR does not depend upon the distribution of underlying variables rather it uses kernel functions. It is useful to create a nonlinear model without changing the explanatory variables, hence giving a better interpretation. SVR uses the principle of maximal margin in which allows SVR to view as a convex optimization problem.

Unlike other models, SVR tries the best to suit the best line within a predefined or a threshold error value. It classifies the data based upon two types one which is above the error line and one which is below the error line. For the lines that don't pass the error is estimated as the difference between the predicted value and the actual value and is denoted using the epsilon ε, visually explained in Figure 3.15.

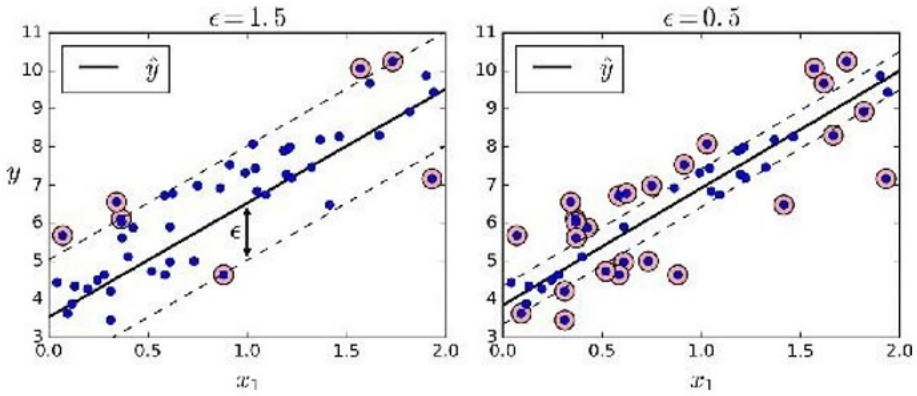

Figure 3.15: Representation of the regression Hyperplane (black solid line), the Boundary lines (dashed black lines) and datapoints (blue and red in case of margin violation).[18]

The main characteristics of this algorithm are:

- **Hyperplane**: A hyperplane is a subspace that has one less dimension than that of its surrounding space. In SVR hyperplane is used to predict continuous values.

- **Kernel**: A kernel is a set of mathematical functions. This kernel function takes data as input and transforms it into the form required by the output. Helps in mapping lower-dimensional data points to higher dimensional data points. There are many types of kernels like polynomial kernel, Gaussian kernel and radial basis function.

- **Boundary Line:** Parallel lines are drawn along two sides of the support vector used to define the threshold and have a value of epsilon.

- **Support Vector:** Line from where the distance is minimum between two boundary points.

[18] (Géron, 2017)

3.5. Model training

Once a dataset of samples has been generated, consisting of features and targets, the chosen algorithm is now able to search for patterns and correlations between output and each input. For this purpose, an iterative cycle was created to be able to simulate each measurement point of the output on the glass plate. The algorithm then repeats the same function for each output measurement point, thus generating 35 models with different characteristics, and each of which a certain number of specific hyperparameters have been set and more suited to the fitting of particular data. This method has been implemented in Python as visible in the following script:

```
#MODEL TRAINING - SVM REGRESSION WITH TUNED HYPERPARAMETERS
filename3 = 'Best_C_epsilon_rbf_scoreR2.csv'
best_C_epsilon_rbf = np.genfromtxt(filename3, dtype=str, delimiter= ';')

#mse_rbf= np.zeros(len(target_pos))
#rmse_rbf= np.zeros(len(target_pos))
#r2_rbf= np.zeros(len(target_pos))
y_pred_svr= np.zeros((len(y_test), len(target_pos)))

#optimized params
for i in range (len(target_pos)):
    if best_C_epsilon_rbf[i,0] == 'rbf':
        svr_reg = SVR(C=float(best_C_epsilon_rbf[i,1]), epsilon= float(best_C_epsilon_rbf[i,2]), gamma= float(best_C_epsilon_rbf[i,3]), kernel='rbf')
        svr_reg.fit(x_train,y_train[:,i])
        y_pred_svr[:,i] = svr_reg.predict(x_test)
    if best_C_epsilon_rbf[i,0] == 'poly':
        svr_reg = SVR(C= float(best_C_epsilon_rbf[i,1]), epsilon= float(best_C_epsilon_rbf[i,2]), gamma= float(best_C_epsilon_rbf[i,3]), degree= float(best_C_epsilon_rbf[i,4]), kernel='poly')
        svr_reg.fit(x_train,y_train[:,i])
        y_pred_svr[:,i] = svr_reg.predict(x_test)
```

Figure 3.16: Script of the implementation of the SVR training function.

3.6. Parameters tuning

Hyper-parameters are parameters that are not directly learnt by the estimator during the model training; in the Scikit-learn algorithms, they are passed as arguments. It is possible to search the hyper-parameter space for the best cross-validation score, that will be seen in the next paragraphs.

An optimal parameters search consists of:

- an estimator (kind of method used in the training phase)
- a parameter space
- a method for searching or sampling candidates

- a cross-validation scheme
- a score function

Some models allow for specialized, efficient parameter search strategies, such as the function **GridSearchCV** that exhaustively considers all parameter combinations and try them to the model. Note that it is common that a small subset of those parameters can have a large impact on the predictive or computation performance of the model while others can be left to their default values. This process consists of performing hyperparameter tuning to determine the optimal values for a given model; in fact, the performance of a model deeply depends on the value of hyperparameters. It is not possible to know in *a priori* the best values for hyperparameters so ideally, we need to try all possible values to know the optimal ones. Doing this manually could take a considerable amount of time and resources and thus it has been used the function **GridSearchCV** to make the tuning of hyperparameters automatic.

The **GridSearchCV** instance implements an estimator: when "fitting" it on a dataset all the possible combinations of parameter values are evaluated and the best combination is retained. To test these configurations a set of hyperparameters are passed to the function to be evaluated, then the function gives back a list of the configuration tested with a score. The configuration with the best value of the kind of score chosen is kept into a table and passed to the model training, to use the optimized parameters themselves. Here below it can be found the implementation of this function:

```
#FIND BEST HYPERPARAMETERS FOR ALL THE TARGETS

C_rbf= np.linspace(0.001, 3000, num=100)
epsilon_rbf= np.linspace(0.0000001, 1, num=100)
gamma_rbf = np.linspace(0.1, 10, num=100)

C_poly = np.logspace(-4,7,num=20, base=3) #from about 0.01 t0 about 2000 in a log scale
epsilon_poly = np.linspace(0.0001, 0.1, num=100)
degree_poly= [2,3,4]

param_grid_rbf = [
  {'kernel': ['rbf'], 'C': C_rbf, 'epsilon': epsilon_rbf, 'gamma': gamma_rbf},
  #{'kernel': ['poly'],'C': C_poly, 'epsilon': epsilon_poly, 'degree': degree_poly}
 ]

best_params_rbf = np.zeros((35,3), dtype= 'U70')

for i in range (35):
    svr_rbf = GridSearchCV(SVR(cache_size=1000), param_grid_rbf, cv=3, scoring= 'r2')
    svr_rbf.fit(x_train,y_train[:,i])
    svr_rbf_results = pd.DataFrame(svr_rbf.cv_results_)
    best_params_rbf[i,1]= svr_rbf.best_params_
    #best_estimator_poly = svr_poly.best_estimator_
    best_params_rbf[i,2] = svr_rbf.best_score_
```

Figure 3.17: Implementation of the Hyperparameters tuning with the GridSearchCV function.

The hyperparameters characteristic of the SVR are:

- **C**: Manage the curve fitting to the datapoints; it gives a penalty term to each datapoint proportional to their distance to the regression curve. A higher value of C means higher overfitting

- **ϒ**: define the influence of each datapoint to the regression curve, in other terms at which distance the datapoint has influence. A higher value means that only close datapoints influence the regression curve and far datapoints are not practically considered. A low value of gamma means less overfitting and also a more "straight" curve of regression

- **ε**: define the thickness of the interval between the 2 margins

Learning the parameters of a prediction function and testing it on the same data is methodologically wrong: if the model just repeats the labels of the samples already seen, will have a high score but it will fail to predict the output

on unseen data; this condition is called overfitting. Here is a flowchart of a usual cross-validation workflow in model training.

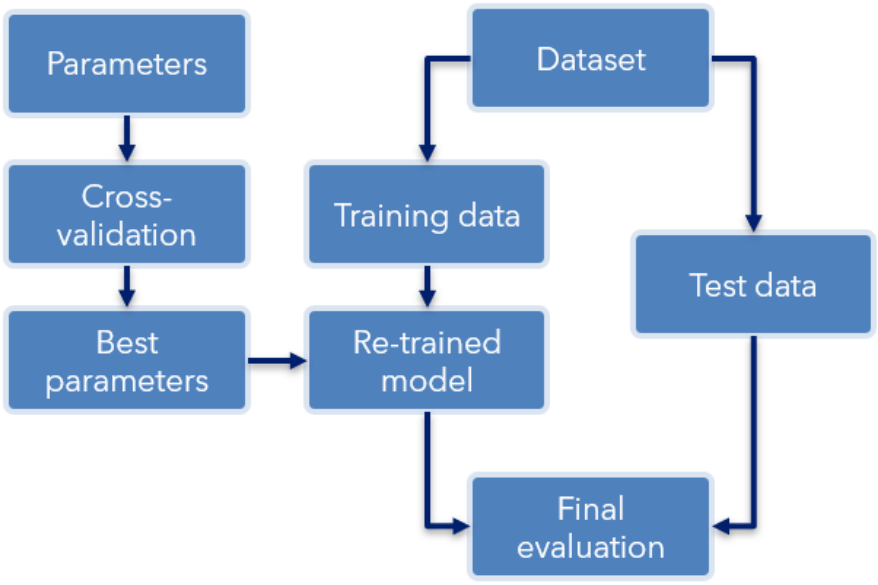

Figure 3.18: Flowchart of the application of cross-validation methods in the model training.

To implement the cross-validation has been used a k-fold CV approach; here the training set is split into k smaller sets and the following procedure is followed for each of the k "folds":

- the model is trained using $k-1$ of the folds as training data
- the resulting model is validated on the remaining part of the data

The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop. The figure below shows this kind of partitioning.

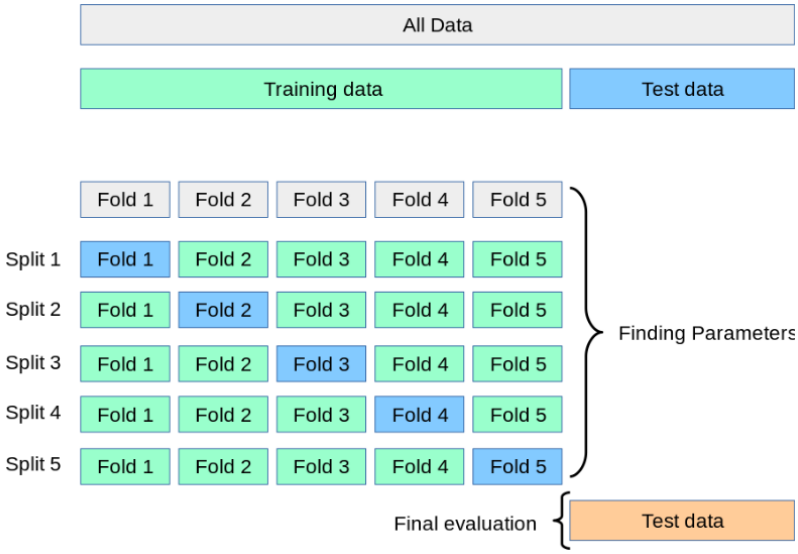

Figure 3.19: Partitioning of the training dataset to perform the cross-validation with the k-fold approach.

As mentioned the hyperparameters chosen differs from all the target variables, indeed every model has its own specific hyperparameters, in the following table:

Model n.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Kernel	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf
C	1.58	0.31	0.47	0.70	0.47	0.70	1.58	1.58	8.01	1.20
ε	9,39E-04	3,44E-04	9,80E-04	5,46E-04	9,70E-04	1,00E-03	9,80E-04	9,70E-04	9,39E-04	8,89E-04
Y	0.62	3.22	2.70	0.62	1.14	1.66	2.18	2.18	0.1	0.1

Model n.	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Kernel	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf
C	8.01	5.34	3.56	1.58	0.70	1.58	2.38	0.47	0.47	0.31
ε	1.00E-07	9,39E-04	8,28E-04	1,72E-04	1,00E-03	9,90E-04	9,80E-04	7,17E-04	1.00E-07	8,59E-04
Y	0.1	0.1	0.1	0.1	1.14	0.62	0.1	2.70	1.66	2.18

Model n.	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Kernel	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf
C	0.31	0.20	2.38	2.38	8.01	8.01	8.01	8.01	5.34	0.31
ε	8,69E-04	3,84E-04	3,84E-04	7.08E-05	4.05E-05	9,49E-04	9,29E-04	9,90E-04	1.02E-05	4,65E-04
Y	1.66	2.18	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.62

Model n.	**31**	**32**	**33**	**34**	**35**
Kernel	rbf	rbf	rbf	rbf	rbf
C	1.20	5.34	0.47	0.47	1.05
ε	9,19E-04	8.09E-05	9,60E-04	9,60E-04	8,38E-04
Y	0.1	0.1	3.22	2.70	0.62

Table 3.20: Optimal hyperparameters used for the prediction model.

3.7. Model testing

The model is now applied to the test dataset, as mentioned in the previous paragraphs, new input data are given to the model and simulated the output, then the output will be compared to the expected results; as can be seen in the Figure 3.21 the measured values (historical data) and the predicted values are represented in the same graph to be confronted, the more the two curves are close the more the accuracy is high. The values resulting from this phase are still normalized, so to go back to physical unity of measures it is necessary to transform the results.

Figure 3.21: Simulated output (red) and measured output (blue) put in comparison to visually evaluate the model accuracy.

3.8. Prediction and model results

At this point, the model is completed and ready to be brought into standard production conditions; again it will work with new data and conditions, and here it will perform with the accuracy expected in the model testing phase.

Taking a step back to the purpose for which this model was born, we can understand how the accuracy of the model itself must be as high as possible for obvious reasons, and that in proportion to the size of the product

acceptance intervals, this should be as high as possible.

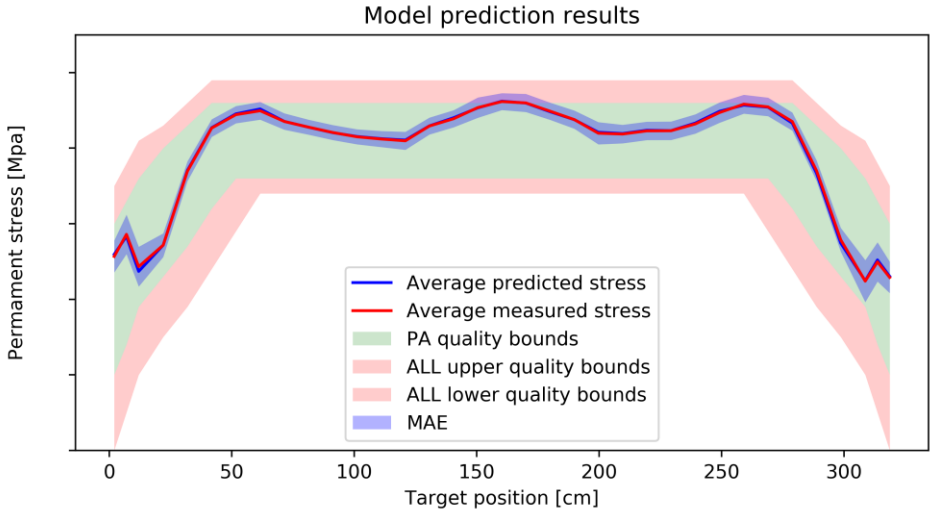

Figure 3.22: Results of the model; the measured (red) and predicted stress (blue) are averaged on the samples of the test dataset. The interval upon the predicted curve stands for the Mean Square Error of the model, that is compared to the interval of acceptance of the product quality (green and red).

As shown in Figure 3.22, it is possible to notice an important result of the model. Are represented in this graph the curve of the measured values of the real system and the simulated values of the model, these values do not refer to a single sampling but to an average over all the samples of the test-set; as it can be seen the two curves are almost overlapping, therefore the residuals of the model are not relevant and the precision of the model seems to be high. Going deeper into the analysis, it is possible to note how the Mean Absolute Error (MAE) of the model is limited and in proportion to the quality acceptance intervals it is very low, this highlights a good overall accuracy of the predictive model. Going even more into the details of the results, it is noted that the Mean Absolute Error in addition to being contained below acceptable values, shows that the accuracy of the model, calculated on different bases, is at a very good level.

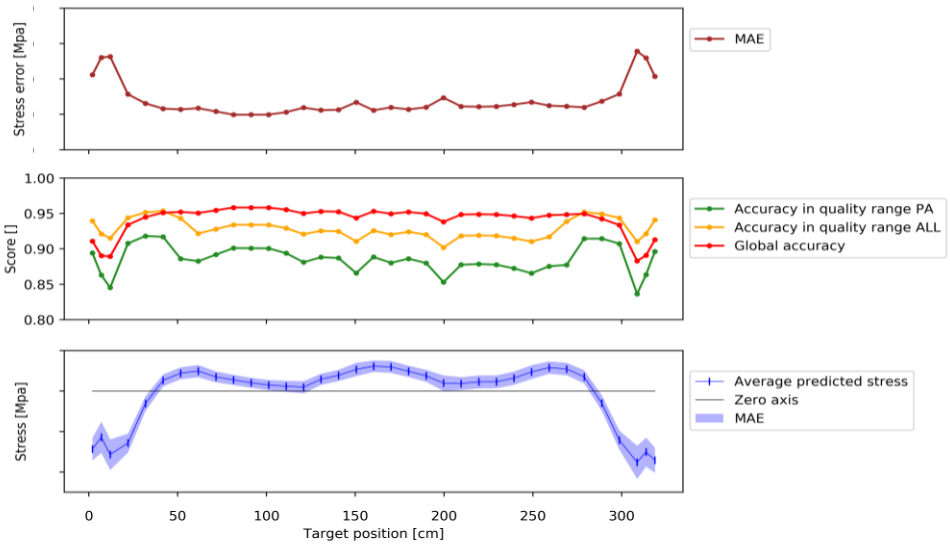

Figure 3.23: Mean Absolute Error of the model and Accuracy score, calculated on a different basis. The Global Accuracy (red) is based on the maximum variance of the mean measured output, and the other 2 Accuracy score are calculated upon the range of the quality acceptance in that point.

The accuracy of the model was calculated using the MAE, based on three parameters: the first is the maximum difference between the average measured values assumed by the system output (this range is kept constant in the calculation of the global accuracy at each point); the other parameters are the maximum variation range allowed by the two product quality acceptance classes, these ranges are variable on the sheet's width according to a production standard.

Another interesting parameter of the model is evaluated with the calculation of the residuals; they are the difference between the measured and simulated output of the system. As mentioned in the previous chapters Bias and Variance take an important role into the model's performances; to evaluate this point in the Figure 3.24 is shown how the average of the residuals of the model is almost equal to zero, this means that the presence of Bias is excluded and the presence of a systematic error too. The standard deviation of the residuals have a similar trend to the MAE and it is also at an acceptable level.

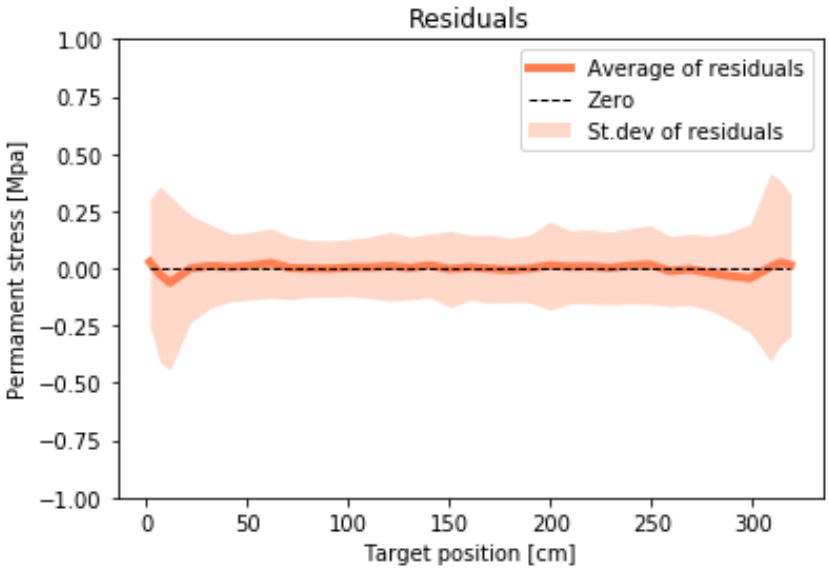

Figure 3.24: Residuals analysis of the model.

3.9. Consideration and further improvement

The model has proved good general performances in the several stages of testing and validation, specifically from the point of view of accuracy; in this way, it reflects very well the objectives set at the beginning of the study. The use of this model in real production conditions can therefore lead to tangible benefits both in terms of process and energy saving, thus allowing an increasingly advanced management of the production process.

By an improvement point of view, the potential of the algorithm and the model itself, it will be possible to insert additional variables into the model, deriving from the previous sections such as the Tin Bath to deeper integrate this model into the process. Considering the trend of the model scores, these are better in the central positions of the glass sheet with respect to the edges, where both the accuracy and the simulation residuals increase with a similar trend, leading to formulate some hypotheses:

- near the edges of the ribbon, some variables were not taken into consideration that could have better explained the behaviour of the

system at that point. This is caused by the impossibility to have historical data, crucial for the characterization of the physical phenomenon, such as the combustion of the burners and the position of the burners themselves. Since these data were not available from the start, this was an expected effect.

- near the edges, the heat exchange is higher than in the centre both in the A / B section and in the previous section of the process, and even the initial temperature conditions of the sheet are not characterized with precision, due to the lack of data also in this point. Again this was an expected effect.

In this regard, it will be necessary to integrate the model of additional variables deriving from this section, in order to increase predictive performance. These variables are as follows:

- Characterization of the burner combustion phenomenon, considering some fundamental variables such as the airflow rate, the calorific value of the gas used, the flame temperature, the speed of the jet and their position.

- Initial conditions of the glass ribbon, entering section A of the annealing lehr

- Air temperature at the inlet of the heat exchangers

- Air pressure at the inlet of the heat exchangers

- Ambient air temperature near the zones A / B

The most important subsequent development in terms of use of this model is its implementation in a process supervision system, available to operators, and in order to increase production yield and reliability of the production itself.

BIBLIOGRAPHY

Aben, H., & Guillemet, C. (1993). *Photoelasticity of glass.* Springer - Verlag.

Alcobaca, E., Mastelini, S. M., Botari, T., & others. (2019). *Explainable Machine Learning Algorithms for predicting glass transition temperatures.* Elsevier.

Benbahouche, S., Roumili, F., & Sangleboeuf, J.-C. (2011). *Modelling of internal stresses in sheet glass during the cooling.* World Journal of nanoscience and engineering.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Bowles, M. (2015). *Machine Learning in Python, Essential techniques for predictive analysis.* Indianapolis: Wiley.

Edmonds-Alt, J.; Tassignon, E.; AGC Glass Europe. (2014). *Glass annealing and lehr management.*

Géron, A. (2017). *Machine Learning with Scikit-Learn & TensorFlow.* O'Reilly.

Harrington, P. (2012). *Machine Learning in action.* Manning.

Liu, H., Fu, Z., Yang, K., Xu, X., & Bauchy, M. (2019). *Machine Learning for glass science and engineering: a review.* Elsevier.

Makarov, R. I., & Lukashin, S. A. (2001). *MAthematical models for statistical analysis and control of sheet glass annealing.*

Margolis, B. I. (2003). *An optimum regime for glass annealing with minimum energy consumption.*

Narayanaswamy, O. S. (s.d.). *Glass: science and technology.*

NCNG. (1997). Handbook for glass technologist. Course on glass manufacture.

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning.* Packt.

Scikit-Learn. (s.d.). *Novelty and Outliers detection.* Récupéré sur Sklearn.

Squilbin, J. M.; Moulart, M.; CNUD. (1995). *Annealing. Annealing lehrs.*

Theobald, O. (2017). *Machine learning for beginners.*

VanderPlas, J. (2016). *Python data science handbook.* O'Reilly.

Yin, H., Su, Y., He, X., & Zhou, Y. (2013). *Influence of annealing on the structure of silica glass.* Laboratory of nanophotonic materials and devices.

Zheng, A. (2015). *Evaluating Machine Learning models.* O'Reilly.