Chair of Waste Processing Technology and Waste Management

Master‘s Thesis

Continuous feeding of subsequent waste treatment machines by optimizing the output flow of a primary shredder

Jason Imhof, BSc.

September 2023

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 22.09.2023

Jason Imhof

Unterschrift Verfasser/in
Jason Imhof

Written by:

Jason Imhof

01535269

Mentors:

Univ.-Prof. Dipl.-Ing. Dr.mont. Roland Pomberger

Dipl.-Ing. Dr.mont. Karim Khodier

Dipl.-Ing. Tatjana Lasch

Ass.-Prof. Dipl.-Ing. Dr.mont. Renato Sarc

Acknowledgments

I am grateful for all the input and assistance from my Mentors, specifically Dr.mont Karim Khodier and DI Tatjana Lasch. Also, my many conversations with friends and colleagues helped me talk through different aspects of my thesis, which helped me specify my thoughts and rethink my work to become more coherent.

I also want to thank the chair of waste processing and the partners of the ReWaste F Project for the required funding of the experiments and the sharing of data.

Recycling and Recovery of Waste for Future is a COMET Project within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMAW, and the federal state of Styria. COMET is managed by FFG.

I could not have finished this work if it wasn´t for the help of my family and friends, that gave me the needed motivational push to keep going. So, while it is my name on the thesis, it needed everyone who contributed to complete it.

Abstract

Steady feeding of subsequent waste treatment machines by optimizing the output flow of a primary shredder

This thesis is concerned with the problem of fluctuating mass and volume streams in waste treatment plants. The heterogeneous material from the mixed municipal waste creates a fluctuating output despite the continuous feeding of the hopper. The suggested solution is to utilize a control loop on the shredder that minimizes output fluctuations and optimizes the use of the conveyor belt and following aggregates in the waste treatment plant. The experimental control loop showed no significant improvement. However, using time series analysis, a prediction model was developed. A different version of equally spaced data points is used for this model: the mass series. This prediction model is up to 75 % better than the defined benchmark of using the current value to predict the next values. The data was taken from the large-scale industrial experiments performed with mixed municipal waste as a part of the ReWaste F Project.

Kurzfassung

Gleichmäßige Beschickung nachgeschalteter Aggregate durch die Vergleichmäßigung des Durchsatzes eines Vorzerkleinerers

In dieser Arbeit werden die Maschinendaten eines Zerkleinerers analysiert um den Outputstrom regeln zu können. Durch die gegebene stark heterogene Materialzusammensetzung der Gewerbeabfälle kommt es trotz kontinuierlicher Beschickung des Zerkleinerers zu großen Schwankungen im Outputstrom. Die Herausforderung des schwankenden Volumen- und Massestromes in Abfallaufbereitungsanlagen soll durch eine geeignete Regelung geglättet werden. Die erste Regelung konnte keine signifikante Verbesserung erzielen. Daher wird in dieser Arbeit der Zerkleinerer charakterisiert um weitere Regelungen zu testen sowie ein Vorhersagemodell für das Verhalten des Volumenstromes entwickelt auf Basis der Zeitreihenanalyse. Hierfür wird eine alternative Art der gleichmäßig verteilten Datenpunkte benutzt, die Massenreihe. Es konnte eine bis zu 75% bessere Vorhersage der nächsten Datenpunkte gegenüber dem Vergleichsmaßstab, einer Vorhersage mit nur dem letzten Messwert, erreicht werden. Die Versuchsdaten sind den großtechnischen Versuchen des ReWaste F Projektes entnommen.

Contents

Page

1 Introduction

Municipal waste treatment has become an essential part of the EU circular economy package as it requires the preparation for re-use and recycling of a minimum of 55% by weight until 2025 and 65% by 2035(EU 2018, p. 21). In 2020, there were 14.5 million metric tons of mixed solid municipal and commercial waste in Germany (174 kg/inhabitant) (Umweltbundesamt, 2023) and in Austria, 2.1 million metric tons of mixed solid waste (236 kg/inhabitant), about 32% of the total municipal waste. The other 68% are made up of separately collected secondary raw materials (e.g., glass, paper, metal), organic waste, and electrical and hazardous waste (BAWP, 2023). To achieve these goals in Austria, the recycling of solid municipal waste must be increased from the current 62.3 % within the coming years (BAWP, 2023). Most of the increase will need to come from improving the recycling of the mixed solid municipal waste, as the other waste fractions are either only a small portion of the whole (electrical and hazardous waste) or already have 83% of the collected materials going towards recycling since recycling of separately collected packaging has been enforced since 1999 (Weißenbach et al., 2020). Therefore, more development for the recycling of municipal solid waste is required.

In waste management and processing, the first step of most treatment plants is to comminute the large mixed solid waste into a grain fraction that is easier to process and transport with conveyor belts. Also, for producing solid recovered fuels (SRF), the first step is comminution, and depending on the desired quality of the product, several more steps can follow (Pomberger, 2008). There are many comminution aggregates, but the most common for mixed commercial waste is a shredder. This is an energy-intensive step in the waste treatment process; however, the benefits of improved grain size distribution and, most importantly, the liberation of the individual materials for further processing are more significant. The steadiness at which this first step operates influences every following step's performance (Curtis et al., 2021). Therefore, optimizing the first step has compounding benefits for the waste treatment plant.

1.1 Problem Presentation

As municipal waste is a very heterogeneous material, it is difficult to predict what sort of comminution will occur within the shredder at any given moment, as both ductile and brittle materials will enter the shredder and engage with the cutting tool. This interaction of the cutting tool with the material influences the comminution performance and the shredder shaft rotation speed. As a result, there are significant fluctuations in the output stream that can cause issues with the following aggregates in the process chain, such as sieving and sorting (Curtis et al., 2021). As the heterogeneous material creates a challenge for the shredder to perform at a constant output level, the output fluctuates significantly. The high fluctuation in the shredder output creates inefficiencies because of an over and under-loading of the subsequent aggregates rather than a constant optimal material volume to be processed by the machine. This issue created the desire to find a way to regulate the output stream of the shredder to minimize the fluctuations.

Among the many factors that will influence the output of the shredder are the input material, the shaft rotation speed, the radial cutting gap width, the shape of the blades, and possibly more (Khodier et al., 2021). This thesis will focus on correlations between the current and previous volume output data points to investigate the hypothesis that a prediction of the next data point can be made. This has only been inferred by previous studies.

1.2 Objective

The idea of a control loop for the shredder was born out of a previous study from the ReWaste F project (Khodier et al., 2021). The hypothesis was that the material from within one load of a wheel loader is more similar than the rest of the heap, and this was supported by some correlations within the output flow. If it were possible to control the output of the shredder by understanding and controlling the shredder itself, no other machines or facilities would be required to improve the waste plant processing performance. Therefore, this thesis will show the data analysis of several real-scale experiments to discover and quantify the possible potential of the idea to change the parameters of the shredder on the fly to control the output.

The first goal is to develop a control loop for the shredder output stream. This requires a better description of the output mass and volume streams of the primary shredder and the development of a model that can be used to forecast the output a few steps ahead and use that model to adjust the parameters of the shredder. Ideally, this model can also be applied to other shredder types since the model is based on a material correlation rather than just a machine parameter. Determining the potential of an improved model of the output behavior from the shredder is the second goal. To use this model, the machine parameters that influence the control loop (e.g., shaft rotation speed, volume and mass output, the input material) must be well known as the control path's reaction time and calibration curve determine whether any forecast of the mass and volume stream can be exploited.

2 Foundations

The statistical and technical basics for understanding the analysis methodology are reviewed in this chapter.

2.1 Comminution Basics: Shredder

Comminution is an important first step in any mechanical treatment facility as it serves several purposes. First, it is necessary for the liberation of material phases to enable further sorting processes; secondly, it creates a maximum grain size to which other process steps can be adjusted; and lastly, it increases the specific surface area of the material; this is mostly important for chemical or biological reactions. The comminution process is complex in most cases, and the choice of a comminution machine is dependent on the material properties. As this is an energy-intensive step, the proper choice of stressing will significantly improve the performance. In waste treatment plants, a wide range of materials will need to be comminuted, e.g., from brittle and hard to ductile and soft. There are many companies that produce a wide variety of comminution machines for customer specifications. A common machine for solid waste comminution is a single shaft shredder, which at its core consists of a single rotor with cutting tools attached, an adjustable gap between the cutting tools, and operating at low rotational speeds (Kranert, 2017; Schönert, 2002). In Figure 1, an example of what some cutting tools could look like is given.

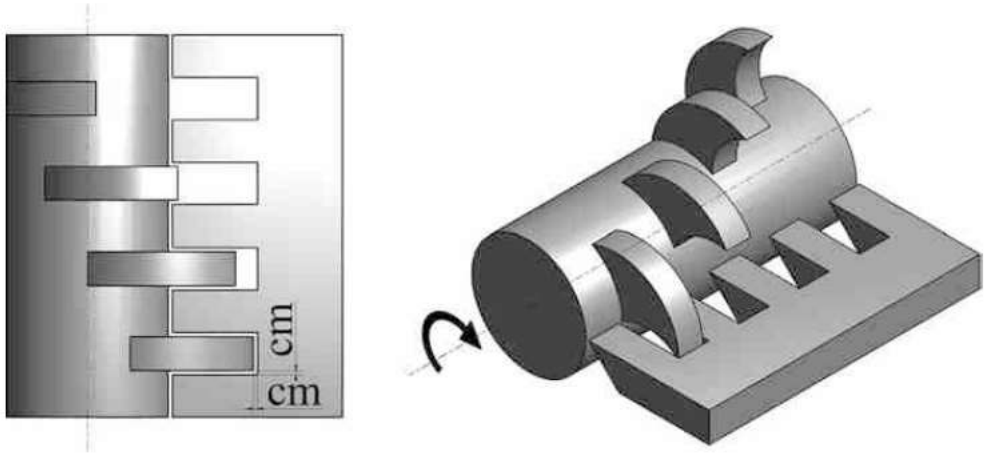


Figure 1: A cutting system with an adjustable main cutting bar (Kranert, 2017)

The type of stressing is a mix of tearing and cutting, as this has proven effective for mixed solid waste. Since mixed solid waste is very heterogeneous, the shredder shaft can be overloaded during a jam caused by a too-hard particle. In such a case and also at regular intervals to assist in unblocking the feed opening, the shaft will counter-rotate a bit, and then the particle may unblock the shaft, and comminution can continue. Also, due to the maximum power supplied by the motor (electric or diesel), the higher the shaft rotation speed the lower the torque

becomes, as illustrated in Figure 2. The lower torque is due to the fact that power (P) is the product of torque (M) and shaft rotation speed (n) (Käser, 2021).

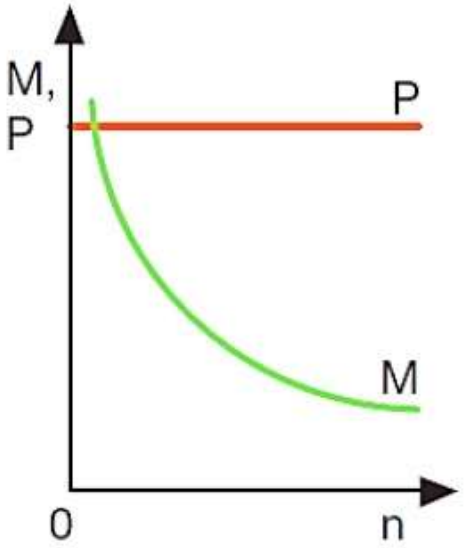


Figure 2: Motor characteristic curve; Power (P), Torque (M), Shaft rotation speed (n); (Käser, 2021)

The reverse rotation procedure contributes to a discontinuous output, which leads to further mass and volume fluctuations further down the process chain (Coskun et al., 2018; Kranert, 2017). These fluctuations are further studied by Curtis et al. (2021). The fluctuation in the time series of the volume and mass flow can be categorized into three groups: the short-term (<15s), the mid-term (15-600s), and the long-term (>600s); all of these have more distinct influences (Curtis et al., 2021). For the sake of this thesis, the same terms will be used. The short and mid-term fluctuations are based more on the material and shredder loading than the long-term fluctuations.

The shape of the cutting tools, gap width, and shaft rotation speed influence the cutting performance and result in a change of energy input, volume and mass output, while particle size distribution is unaffected by the gap width and shaft rotation speed (Khodier et al., 2021; Khodier and Sarc, 2021).

2.2 Technical Basics: Control Loops

A control loop is the fundamental building block of control systems, particularly industrial ones. It consists of three components: the process sensor, the controller function, and the final control element (FCE). Together, they automatically adjust the value of the controller output in order to change a measured process variable (PV) to the value of a desired set-point (SP). In Figure 3, the output response of a controller to a disturbance is illustrated; this can be pictured like a cruise control on a car; the more headwind or, the higher the incline of the road is, the more the fuel needs to be burned to provide the required power to maintain the speed of the vehicle.

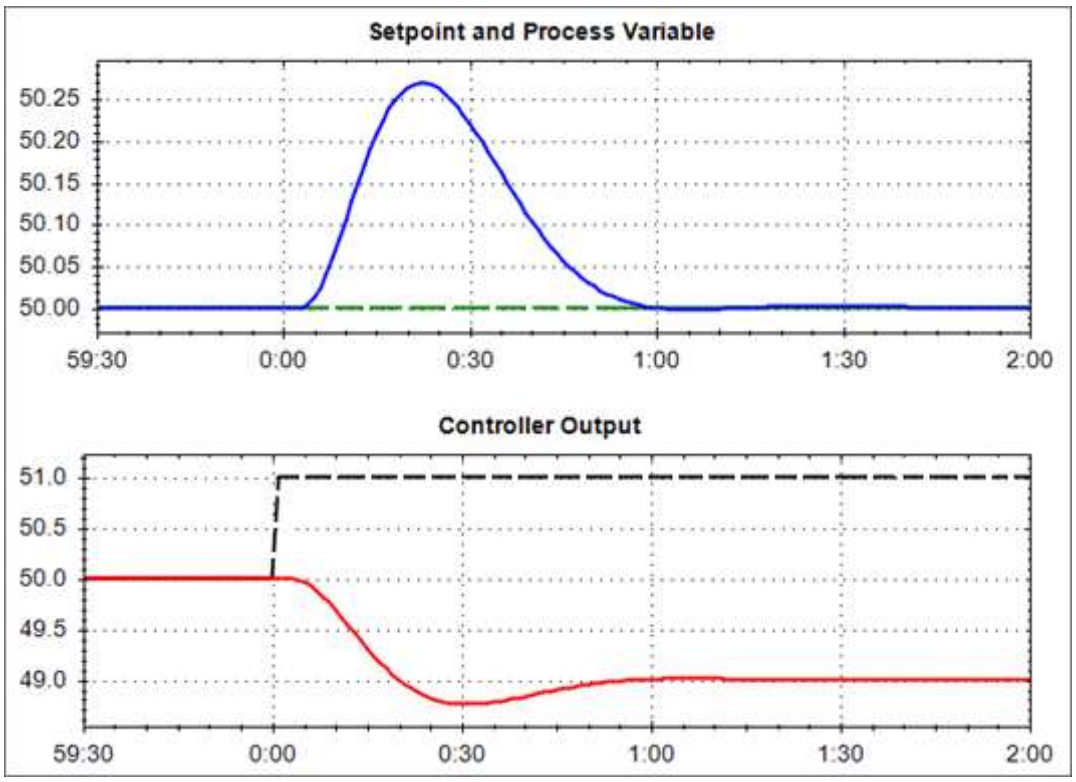


Figure 3: A disturbance (bottom - black, dashed line) shifts the process variable (PV, top – solid line) from the set point (SP, top – dashed line), so the Controller Output (bottom – solid line) responds; (Smuts, 2012)

Figure 4 illustrates that the classical control loop can be shown as a flow chart. The measured error can be as simple as a difference between the set point and the current value but can also be a proportional band, i.e., a range of values within the PV should remain. This can be particularly helpful for slow-responding control loops. The proportional band is similar to the output response, but the PV is compared to an upper and lower limit rather than just a single SP. This method helps to limit the controller output response to reduce the risk of an unstable control loop (Smuts, 2012).

In Figure 4, the SP is the desired output value, which is compared with the current process variable by a mathematical function (e.g., a simple difference), generating a measured error. The controller is a combination of proportional (P), integral (I), and derivative (D) building blocks, which create a P, PI, or PID controller. Using the measured error, the controller creates an output as an electrical signal to the shaft motor to change the rotation speed. The rotation speed takes effect on the comminution process, and thus the process variable changes. The new process variable is measured by a sensor, and so the new value is compared with the set point, generating a new measured error, i.e., the difference between the set point and the current process variable. This completes the loop, and with good controller tuning, the process variable will ideally remain at the set point.

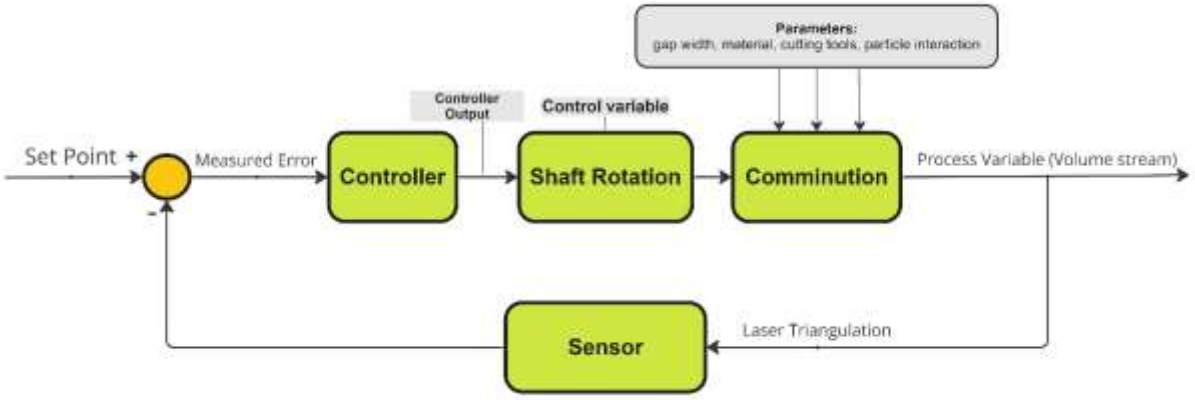


Figure 4: Shredder control loop Flowchart (closed loop)

There are two common control loop classes: open and closed. In an open-loop control system, the control action from the controller is independent of the process variable. For example, a shredder operating with a set shaft rotation speed only attempts to keep the shaft rotation speed independent of the volume stream output. This is how most shredders are operated. The Flowchart from Figure 4 shows that the control system becomes an open loop if the sensor does not connect to the sum point. In contrast, in a closed-loop control system, the control action from the controller depends on the desired and actual process variable. Using the shredder analogy, the goal is to change the shaft rotation speed based on the shredder output. A closed-loop controller has a feedback loop, ensuring the controller exerts a control action to maintain the process variable at the same value as the SP. For this reason, closed-loop controllers are also called feedback controllers (Smuts, 2012).

Output feedback control, also known as controlling a system by its output, involves measuring the system's output variables, such as position, speed, rpm, or pressure, to adjust the control input and regulate the system’s behavior. While output feedback control is widely used in many practical applications and was used during the experiments performed for this thesis, it can also present several challenges and limitations. Here are some of the main problems associated with controlling a system by its output:

1. Observability: To implement output feedback control, the system must be observable, which means that the system's state can be accurately estimated using only measurements of its outputs. However, in some systems, not all of the state variables can be measured, and therefore, the observability of the system is limited. This can lead to difficulties in designing an effective output feedback controller.
2. Stability: Output feedback control can be less stable than other control methods that use full-state information. This is because the control action is based solely on the measured outputs, which may not be able to capture all the dynamics of the system. As a result, the controller may respond slower or less accurately to disturbances or changes in the system.

3. Noise and disturbances: Output measurements may be affected by noise and disturbances, which can cause the controller to respond erroneously or unpredictably. This can lead to instability or reduced performance of the control system.
4. Controller design: Designing an effective output feedback controller requires knowledge of the system's transfer function, which relates the system's output to its input. However, accurately determining the transfer function from output measurements alone can be challenging, particularly for complex systems.

Despite these challenges, output feedback control remains a popular and useful method for controlling many different types of systems. With careful design and tuning, output feedback controllers can achieve stable and accurate regulation of system behavior, even when full-state information is unavailable.

The tuning of a control loop and, thus the degree of success depends on the process characterization. Regarding controlling a process variable, the most important factors are the dead time and the lag time. The dead time is the time between the change of the controller output and the beginning of the PV change, and the lag time is the duration from 0% to 63% of the total PV change, i.e., the new stable PV value after an adaptation to the controller output. In Figure 5, these two factors are determined by step testing and determining the time at which the greatest rate of change in the response curve occurs (Smuts, 2012). The ratio between dead and lag time is important for the tuning of the controller as it influences the stability and response time of the control loop.

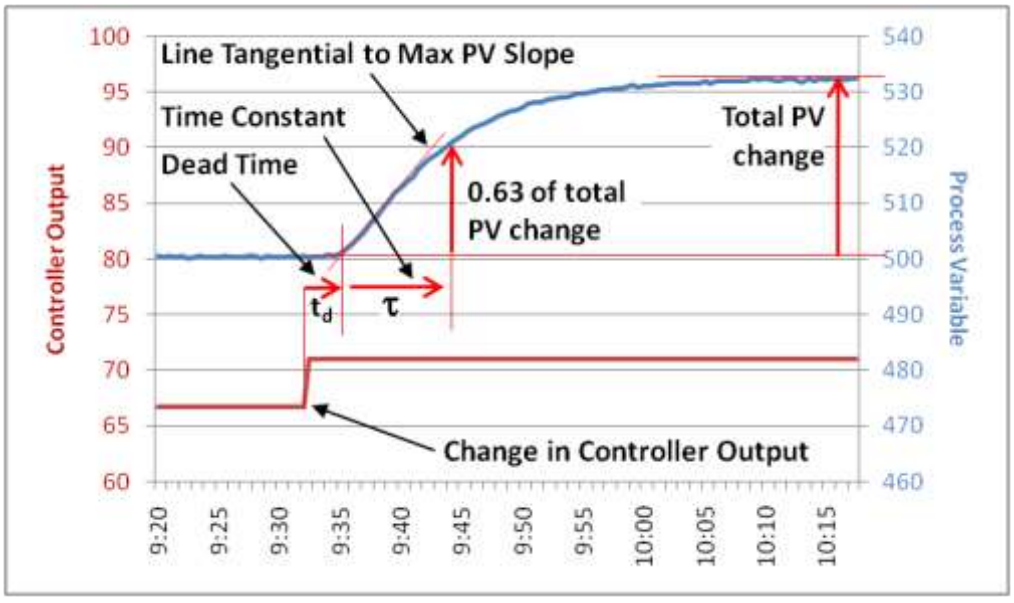


Figure 5: Measuring Dead time and Lag time (Time Constant); (Smuts, 2012)

There are several approaches for tuning a control loop to minimize the loop settling time and optimize the controller output response. Some of these tuning rules have a narrow range of applications and others are more universal. The minimum settling time is primarily influenced by the dead time, while the stability of the control loop is better with lag time dominant control

loops. Figure 6 shows an overview of the effect that dead and lag time have on the tuning process. The lag time and dead time influence the control loop differently, so a proper determination is necessary for the tuning and optimization of the control loop. To better understand the difference between dead and lag time, some extreme examples are useful. Pure dead time would define a process that takes a long time to begin changing, but when it begins, it reacts quickly, e.g., the mass flow along a conveyor belt. Pure Lag time defines a process that begins changing immediately, but the change of the PV is sluggish, e.g., the charging of a capacitor. Another way of thinking about it is an example from everyday life. When a file is transferred on a computer, there are two extreme cases after the command to move the file is given. Assuming that the transfer requires 5s to complete, either the computer immediately opens a progress bar that is then continuously filled over the 5s, or there is no reaction for 5s and then the progress bar appears and is filled immediately. For the user, the first case would be pure lag time, while the second case would be pure dead time. This also gives a more intuitive sense of why pure dead time is more difficult to control. If there is no feedback on whether the given command has had any effect, it is easy to give too many commands and thus overload the system. If the progress bar gives some feedback on how far along the system is in responding to a command, it is easier to give further commands that are appropriate. This is what Figure 6 shows: which tuning rules can be applied to achieve a minimum settling time for the PV while maintaining control loop stability. Also, unless lag time dominates the response time, only a PI controller rather than a PID controller can be used. This is a result of how the differential component interacts with the PV. For the sake of this thesis, it is enough to understand that depending on the ratio of dead and lag time, the control loop must be tuned differently and that tuning a control loop is an important step that requires a deeper understanding of control loops and their properties.

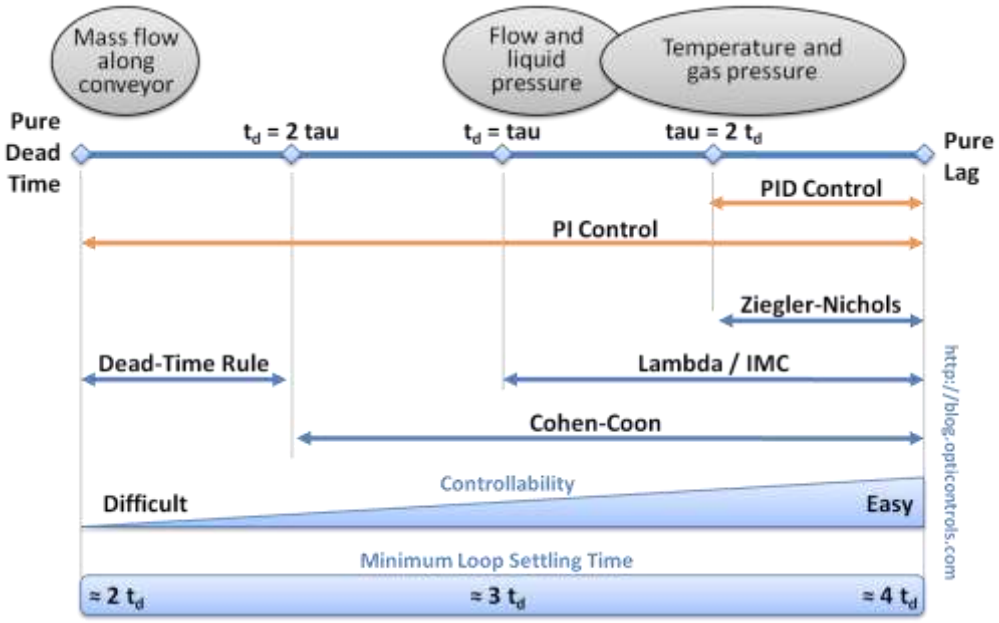


Figure 6: Pure dead time to lag time on a continuum with the applicable tuning rules (t_d: dead time, τ: lag time (Smuts, 2012)

2.3 Basics of Time Series

Analyzing experimental data over time presents unique challenges in statistical modeling and inference. The correlation introduced by sampling adjacent time points can limit the applicability of traditional statistical methods that depend on the assumption of independent and identically distributed observations. Time series analysis is the systematic approach to answering mathematical and statistical questions posed by these time correlations.

Correlation is a statistical concept that measures the strength and direction of the linear relationship between two variables. It provides insights into how changes in one variable are associated with changes in another variable, i.e., is there a systematic change between the two values, or is the change only random and can be described by a stochastic variable with a certain variance (Shumway and Stoffer, 2017).

There are ARIMA models to describe a time series. ARIMA is a compound acronym for Auto Regressive Integrated Moving Average. The Auto-Regressive (AR) part is a series x_t that is defined by p past values $x_{t-1},\ldots,x_{t-p}$, and after each value is multiplied with a corresponding coefficient, a stochastic variable (w_t) defined as white noise is added to the sum. A formal definition can be seen in Equation 1 (Shumway and Stoffer, 2017).

Equation 1: AR(p) Model

$$x_t = w_t + \sum_{t=1}^{p} \varphi_t * x_{t-p} = w_t + \varphi_1 * x_{t-1} + \varphi_2 * x_{t-2} + \cdots + \varphi_p * x_{t-p}$$

The integrated part can be thought of as the trend of the series, i.e., a drift in a particular direction. This could be as simple as a linear function or a higher-order polynomial. Instead of using the autoregressive representation where the left-hand side of the equation assumes a linear combination of x_t, the moving average model, referred to as MA(q), assumes a linear combination of the white noise w_t to form the observed data. In Equation 2, the MA(q) Model is defined formally. The idea is to define the current value as the sum of noise q steps back (Shumway and Stoffer, 2017).

Equation 2: MA(q) Model

$$x_t = w_t + \sum_{t=1}^{q} \theta_t * w_{t-q} = w_t + \theta_1 * w_{t-1} + \theta_2 * w_{t-2} + \cdots + \theta_q * w_{t-q}$$

Using these ARIMA models as linear combinations, they can describe a wide range of data sets. The next section will look at tools to better fit an ARIMA Model to an observed data set. The combined model can be seen in Equation 3.

Equation 3: Auto Regressive Integrated Moving Avarage (ARIMA) Model of Order p, d, q

$$x_t = AR(p) + I + MA(q) = w_t + \sum_{t=1}^{p} \varphi_t * x_{t-p} + I(t) + w_t + \sum_{t=1}^{q} \theta_t * w_{t-q}$$

The effect of the integrated part can be reduced by differencing the data set. This can help to achieve stationarity, which means that the data set has an average around which it fluctuates, i.e., no trend. The differencing order is denoted with d, so an ARIMA model would be described by the variables (p,d,q), which give the order of AR, I, and MA parts. Looking at the differenced data can help to understand what next steps could be useful. If it appears to be randomly distributed, i.e. white noise, then there is likely no more trend in the data.

Auto-Correlation

The concept of correlation involves measuring how closely two variables are related. In the context of time series, the two variables are usually part of the same data set but separated by a time difference expressed as lag k. Therefore, the correlation is between two data points of "itself"; hence it is called Auto-Correlation. This is usually expressed as a correlation coefficient that ranges from -1 to 1 and the equation is expressed as r_k and can be seen in Equation 4.

A positive correlation coefficient signifies a direct or positive relationship, indicating that as one variable increases, the other variable also tends to increase. Conversely, a negative correlation coefficient indicates an inverse or negative relationship, where one variable increases while the other variable decreases. A correlation coefficient of zero suggests that there is no linear relationship between the variables. The denominator describes the variance of the data point at lag 0, while the numerator describes the covariance of the value at lag 0 with the value at lag k. The covariance is a measure of how strongly two variables will vary together and thus the correlation coefficient can be thought of as a measure of the strength of the linear relationship between two variables, normalized by the standard deviations of the variables (Fahrmeir et al., 2016)

Equation 4: Correlation Coefficient r_k (Chatfield, 1975)

$$r_k = \frac{\frac{1}{N-k}\sum_{t=1}^{N-k}(x_t - \overline{x})(x_{t+k} - \overline{x})}{\frac{1}{N}\sum_{t=1}^{N}(x_t - \overline{x})^2}$$

Taking this correlation coefficient, applying it to various lags, and gathering the results in a single diagram leads to the autocorrelation function (ACF), which is displayed with the partial auto-correlation function (PACF) in Figure 7. As mentioned previously, the correlation coefficient r_k will result in a specific value between -1 and 1 for every value of k, which is the lag or distance between the two considered data points, i.e., if only r_1 and r_2 is 1 and every other r_k value is 0 then the correlation between with the current value and the next two values of a given dataset is linear while every value beyond the next two is stochastic. So, for the example given in Figure 7, the correlation approximates a dampened oscillation, i.e., there is

a recurring feature in the data that can be called seasonality. Seasonality (S) is a useful tool for data sets that have periodic trends (e.g., sin(x)). Data sets where one would expect such a seasonality would be any set where a time frame is cycled through, such as yearly, monthly, or weekly data sets (e.g., weather data for rainfall or temperatures (summers vs. winters)). The ARIMA Model is extended with S to the Seasonal Auto-Regressive Integrated Moving Average (SARIMA) and can now accommodate an AR(P) and MA(Q) as a periodic function with a defined period or season, i.e., a function that recurs every S lags. Figure 7 shows an ACF where such a seasonality can easily be seen as a harmonic oscillation with a period S=12. Since after 12 lags, the value of the ACF has the same prefix and is only dampened by some factor. SARIMA now can have up to 7 variables (p,d,q, P, D, Q, S), each with their calculated coefficients within the model.

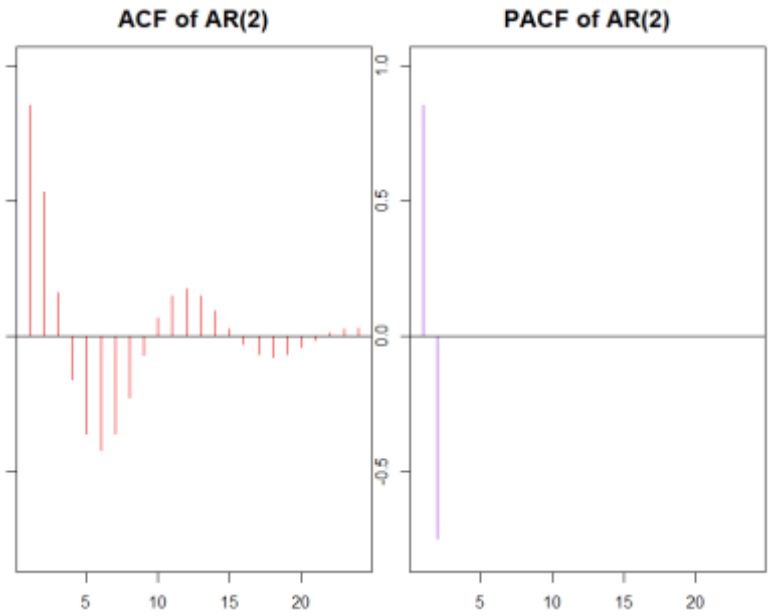


Figure 7: Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) for second-order autoregressive Model (AR(2)), Autocorrelation coefficient has a periodic fluctuation visible in the ACF, while the PACF cut off after two peaks.

Partial Autocorrelation Function (PACF) is used to measure the correlation between two variables, X and Y, while controlling for the effects of one or more other variables, Z, within a time series. This is where the concept of partial correlation ($\rho_{XY|Z}$) comes into play. The idea behind $\rho_{XY|Z}$ is to measure the correlation between X and Y with the linear effect of Z removed or partialled out. For example, if the three variables that are considered are the waste generated per capita, electric vehicles per capita, and income per capita, it would be expected to see a correlation between all three, but the correlation between waste generation and electric vehicles would need to be controlled for using the third variable which will influence the other two and therefore must be removed to properly reflect the correlation between the first two. For time series, this is the same effect; the PACF is similar to the ACF, but it only shows the relationship between two observations that are not explained by the shorter time lags between them. The PACF can show the relationship between data points that are some time

units apart. For example, with lag 3, the PACF considers only the unique correlation between those two data points after removing the influence of the correlations at lags 1 and 2. The PACF is, therefore, useful to better visualize the Autoregressive contribution to a model. This can be applied to the AR model since it trails off in the ACF because x_t is correlated with x_{t-2}, as defined in Equation 5. x_{t-1} was created using x_{t-2} and x_{t-2} is defined by x_{t-3} and so on. The stochastic variables w_{t-n} start to add up, and thus the correlation begins to become less significant as more random variables are introduced into the definition. Therefore, the linear causality between lag 0 and lag k needs to be removed, i.e., the values between x_t and x_{t-k}.

Equation 5: AR(1) Model correlation of x_t values

$$x_{t-1} = \varphi_2 * x_{t-2} + w_{t-1}; x_t = \varphi_1 * x_{t-1} + w_t = \varphi_1 * \varphi_2 * x_{t-2} + w_{t-1} + w_t$$

The formal definition for the partial correlation coefficient is given in Equation 6, from which the PACF can be created for k lags in the same manner as the ACF.

Equation 6: PACF Definition, $\hat{x}_{t+k} = \hat{x}_t = \sum_{i=2}^{k} x_{t+i-1}$; k -number of lags, ϕ_k – partial correlation coefficient; (Shumway and Stoffer, 2017)

$$k = 1: \phi_1 = \mathrm{corr}(x_{t+1}, x_t) = r_k(1); k \geq 2: \phi_k = \mathrm{corr}(x_{t+k} - \hat{x}_{t+k}, x_t - \hat{x}_t)$$

The ACF and PACF can be used to better model data sets because of the distinct behavior from the ARMA models described in Table 1. The ARIMA Model is the same as the ARMA Model, given that it was sufficiently differenced to achieve stationarity. The ARMA Model is analog to ARIMA except for the integrated part. This can be used to estimate the order of the best-fit model.

Table 1: ACF and PACF behavior for ARMA models; (Shumway and Stoffer, 2017)

	AR(p)	MA(q)	ARMA(p,q)
ACF	Trails off	Cuts off after lag q	Trails off
PACF	Cuts off after lag p	Trails off	Trails off

Table 1 shows the ideal model behavior. However, real data sets have a less clear cut-off, and therefore, the ACF and PACF are only used to test for potential cut-offs; if none are obvious, then there are other more advanced methods available that can be studied in the appropriate literature. The estimation of p and q can be used to create a smaller number of models that can be compared with the observed data. This reduction in computing power was particularly useful when computational power was low, but this method can still optimize some calculations. Since, for this thesis the brute force method is applicable, other methods will not be further studied here.

After several different p and q are modeled and using the sum of squared errors (SSE) or Akaike´s Information Criterion (AIC), the best fit is chosen. The AIC penalizes the number of parameters used for the model, as seen in Equation 7. This is done so that simple models, which are more flexible, are given an advantage over significantly more complex models with an insignificant better fit. The lower the SSE or AIC value is, the better the model fit is (Shumway and Stoffer, 2017). Basically, the parsimony principle is in effect: Take the simplest mathematical model with the best fit to describe the correlation between two variables, e.g., the mean volume output and the shaft rotation speed. Whenever possible, i.e. it doesn't sacrifice too much accuracy, a linear model is preferred for its simpler calculations over a higher-order equation.

Equation 7: AIC Definition; Standarddeviation: σ_k = SSE/n, Number of datapoints(n), Number of parameters (k)

$$AIC = \log(\hat{\sigma}_k^2) + \frac{n + 2k}{n}$$

Another Method for comparing models is the F-test. If there are a total of n data points available to estimate parameters for both models, you can compute the F-statistic using Equation 8, where RSSi represents the residual sum of squares for model i. In cases where the regression model has been computed with weighted data, you can replace RSSi with χ2, representing the weighted sum of squared residuals.

Under the null hypothesis, which posits that Model 2 does not offer a significantly improved fit compared to Model 1, the F-statistic follows an F-distribution with (p2 - p1, n - p2) degrees of freedom. The null hypothesis is rejected if the calculated F-statistic derived from the data surpasses the critical value associated with the F-distribution, typically chosen to control the desired false-rejection probability, such as 0.05.

It is important to note that since F is a monotone function of the likelihood ratio statistic, the F-test effectively operates as a likelihood ratio test in this context.

Equation 8: F-Statistic: RSS_i (Residual sum of squares model i), p_i (degrees of freedom i), n (Total sample size); (Fahrmeir et al., 2016)

$$F = \frac{\frac{RSS_1 - RSS_2}{p_2 - p_2}}{\frac{RSS_2}{n - p_2}}$$

Time Series Smoothing

Analyzing data through time series smoothing is a crucial technique. By minimizing short-term fluctuations or noise in the data, the underlying trend or pattern becomes more visible, enabling better identification of seasonal patterns, long-term trends, or other structural components of the time series. Furthermore, time series smoothing can aid in detecting anomalies or irregularities that may be concealed in the raw data. It is critical to choose an appropriate

smoothing method based on the characteristics of the data, such as the duration of the time series, the level of noise, and the desired level of flexibility in the trend estimation. Common smoothing methods include moving averages, exponential smoothing, and linear decay.

In Figure 8, the time series is smoothed using a moving average of several time frames, including an overall average, which is basically the traditional arithmetic mean of the entire data set. Ultimately, time series smoothing aims to gain an in-depth understanding of the data and extract meaningful insights that can aid in decision-making. Ballini and Yager (2014) describe the advantages of the linear decay smoothing method and advocate for its superior flexibility over Moving Average and exponential smoothing. Exponential smoothing and linear decay use a weighted average, giving more recent values greater importance. This is good for the average age, but the expected variance is greater. The best balance is found for weights that decay following a linear function rather than an exponential one. The most flexible method was chosen for the smoothing in this thesis. However, there may be a better option if further studied.

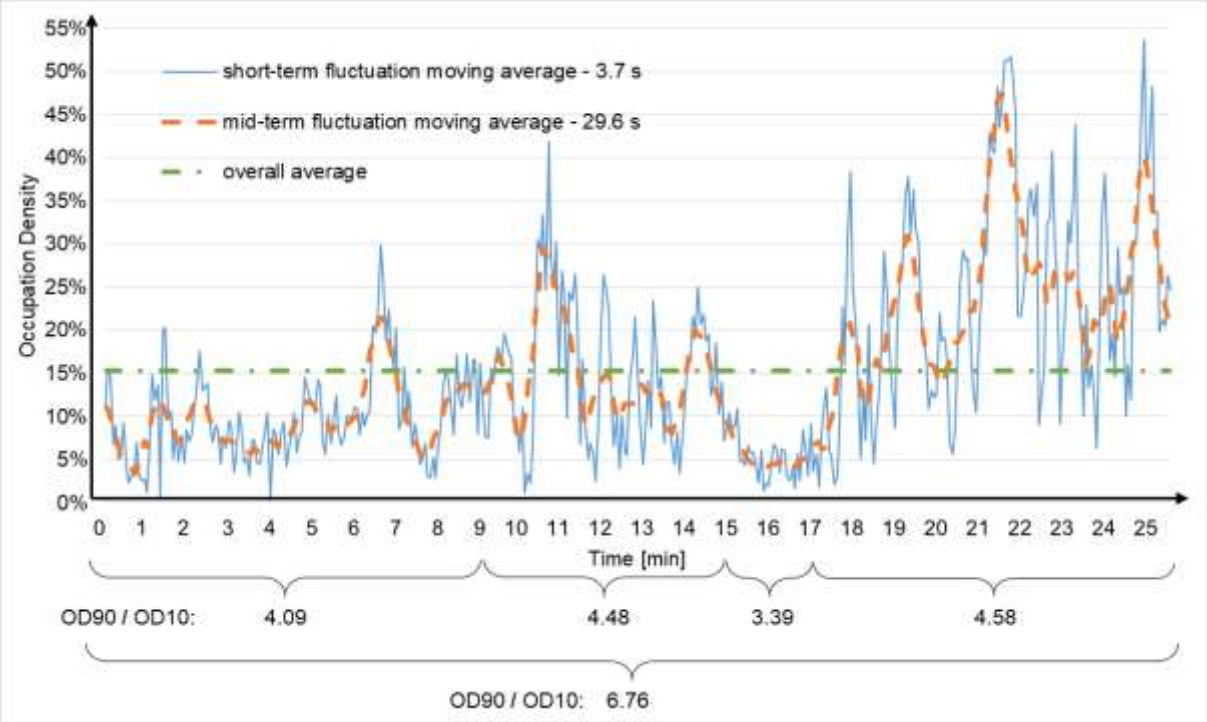


Figure 8: Moving Average time frames used to reveal layers of information (Curtis et al., 2021)

Quantile-Quantile Plot (QQ- Plot)

A QQ-Plot is a visual test to help identify departures from a normal distribution.

A Normal Q-Q (quantile-quantile) plot, also known as a Normal probability plot, is a graphical tool used to assess whether a dataset follows a normal distribution. It helps to visually compare the distribution of a given data set to a theoretical normal distribution by plotting the quantiles of the given data against the quantiles of a normal distribution.

- Quantiles: A quantile is a value that divides a dataset into equal parts. For instance, the median is a quantile that divides the data into two halves, with half the values below it and half above it. Other quantiles include quartiles (dividing the data into four parts) and percentiles (dividing the data into 100 parts).
- Creating the Plot: In a Normal Q-Q plot, the data is sorted in ascending order, and the corresponding quantiles are calculated. Next, these quantiles are plotted against the quantiles of a normal distribution with the same number of data points. If the data closely follows a normal distribution, the points on the plot should fall approximately along a straight line.

Interpreting the Plot:

- Straight Line: If the points on the plot follow a straight line, it suggests that the data is approximately normally distributed. Deviations from the straight line might indicate departures from normality.
- Curvature: If the points curve upwards at the ends of the plot, it suggests heavy tails compared to a normal distribution. If the points curve downwards at the ends, it suggests light tails compared to a normal distribution.
- S-Shaped Curve: An S-shaped curve indicates that the data is bimodal or has multiple modes, which means it's not well-described by a normal distribution.

Given a dataset, it can be checked if it follows a normal distribution using a Normal Q-Q plot. First, sort the dataset in ascending order and calculate the quantiles. Then, generate the corresponding quantiles for a normal distribution using the mean and standard deviation of the data. Next, plot the calculated quantiles of the data against the theoretical quantiles of the normal distribution. If the points fall along a straight line, the data is approximately normally distributed (Chambers et al., 2018). This is modeled in R and is seen in Figure 9.

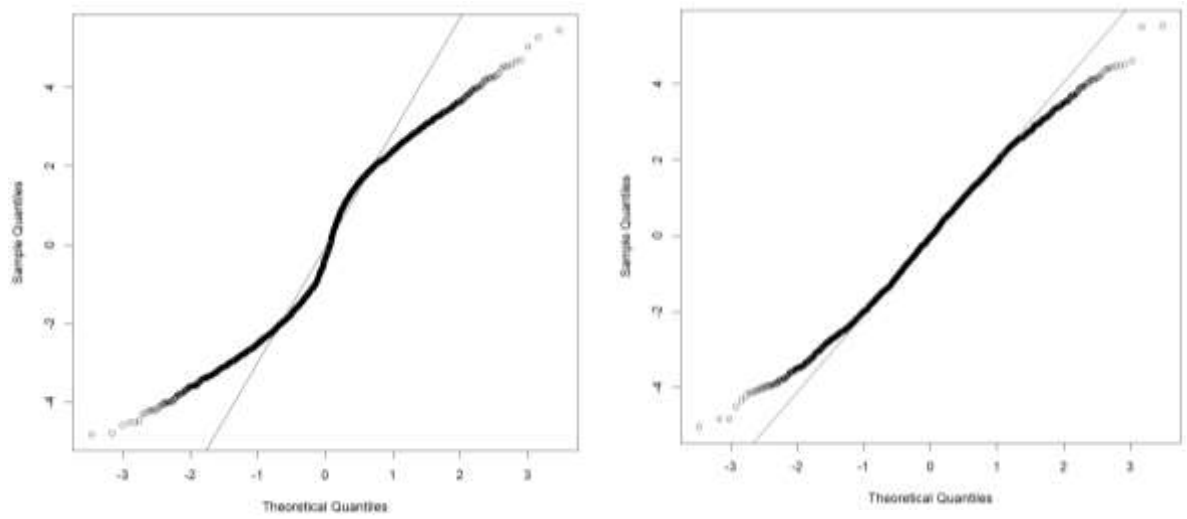


Figure 9: QQ-Plot of Bimodal(left) distribution and an approximately normal distribution with heavy tails (right); Line is a reference for normal distribution.

Normal Q-Q plots are valuable tools for quickly assessing the distribution of your data and identifying departures from normality. However, they are not definitive tests for normality and should be used in conjunction with other methods, such as statistical tests and histograms, to get a comprehensive understanding of your data's distribution.

3 Methodology

Based on the hypothesis that there is a material and time correlation between the output volume and mass stream data points, a mathematical model should be developed using empirical data and the model needs to be tested using real world data. This hypothesis was a result of the gathered data from ReWaste 4.0 experiments by Khodier et al. (2021) and Curtis et al. (2021). To this end, the control loop parameters were determined and an experimental control loop tested. The resulting data was further studied to determine the correlation as well as a potential model with a time series analysis.

3.1 Data Acquisition

To acquire the desired data some preparations were made and some planning of the test runs was done. This will be the topic of this section.

3.1.1 Experimental Setup

The following passage should give a brief summary of the required resources and setup of the experiments. It is necessary to have a shredder and material that can be shredded as well as a suitable measuring device for the mass and volume streams. The other requirements to run this experiment are a place to set up the machines and store the material with the necessary legal permissions, a crane or wheel loader (vehicle and operator), all the different extension cables for the electronics, and many other miscellaneous tools. The location in St. Margarethen (Austria) was provided by Müllex GmbH, one of the project partners.

The data was collected from several experiments using a Digital Material Flow Monitoring System (DMFMS), a mobile measuring unit which was placed directly after the mobile shredder the “Terminator 5000 SD” which were both from the company Komptech GmbH (Khodier et al., 2021). The DMFMS is a Komptech prototype and consists of a conveyor belt, a mass flow measuring unit and a volume flow measuring unit. The mass flow unit consists of an integrated weighing belt scale and functions at a speed of 0.5 m/s, has a throughput rate of 5 – 100 t/h with a tolerance of +/- 2% within the 25 – 100% operating range. The volume flow measurement uses an optical sensor above the belt and with laser triangulation it determines the contour of the material on the belt and the belt speed (Curtis et al., 2021). The shredder was continuously fed by a crane with a cactus grab to ensure that the volume and mass streams would not drop to zero because there was no material available to be shredded. The material used was mixed solid commercial waste from Styria, Austria. In the Figure 10 the setup of the experiment can be seen. Using this set up all the test runs were conducted.

Figure 10: Komptech Terminator and DMFMS in St. Margarethen at the Müllex GmbH facilities

The mass and volume streams are recorded using the DMFMS, the Terminator also has a volume stream sensor which can record data independently and the data is stored as a Dewesoft file. These files can be rather large (~4 GB for 1h) since the software is configured to record a lot of machine data that is not relevant for this experiment. After selecting the relevant Data and exporting it as a CSV-File, the further analysis is done using the statistical programming language R. Within Dewesoft the files can be viewed and edited, an example of what these time series can look like is in Figure 11. The bottom shows the high rpm fluctuations of the shaft, while the top shows the volume output fluctuations over the length of the experiment. It is clear that the long term (>600s) time frame needs to be examined further to extract any information.

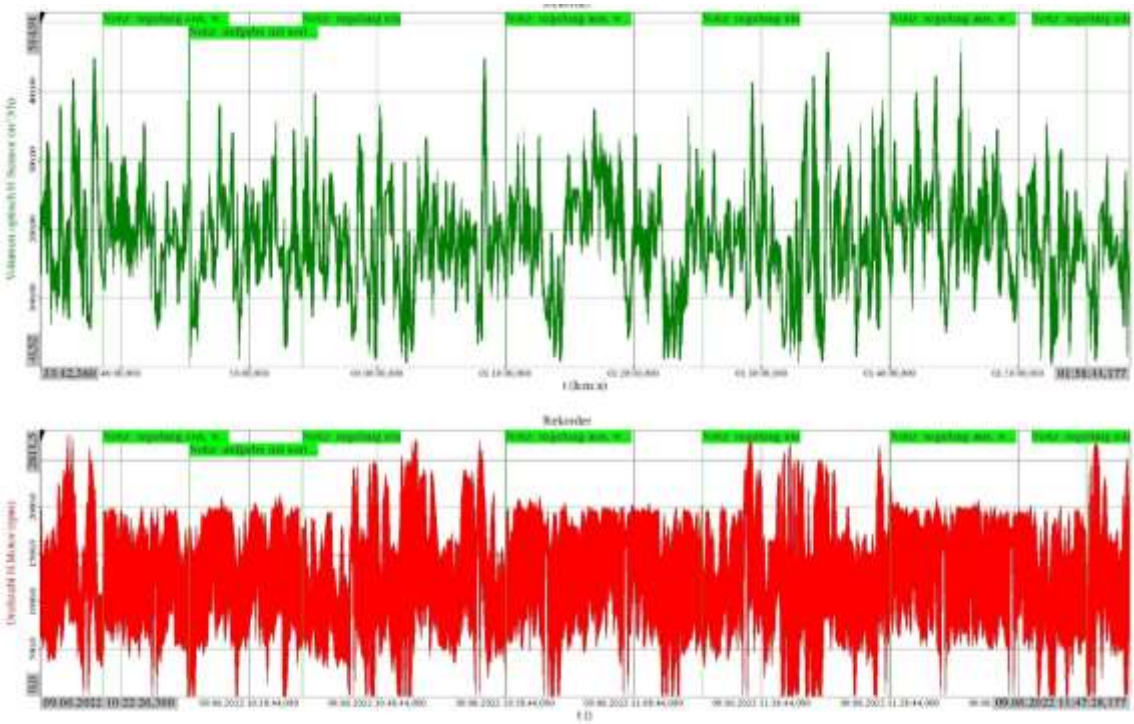


Figure 11: Dewesoft File of shredder data with Volume output in m^3/h (top) and shaft rotation speed in rpm (bottom)

3.1.2 Test Runs

Some experiments were conducted to set up the tuning for the control loop of the shredder to limit its short-term fluctuations. The test runs were done in the described order and the first two tests set up the third and primary experimental goal. The following is a overview of the tuning process:

1. Calibration curve: Long term (>600s) mean volume output at different set shaft rotation speeds to create a calibration curve and gain a better understanding of the correlations between the mean volume output and the shaft rotation speed.
2. Step testing: A change of the set shaft rotation speed of 40% and recording the volume output, with the goal of gaining a better understanding of the response time (dead and lag time) of the shredder output to PV changes.
3. Application of a control loop to the shredders shaft rotation speed to minimize output fluctuations. This control loop is a rudimentary test to see the potential of the material correlation hypothesis.

The first experiment was focused on gathering some data to gain a better understanding of the shaft rotation speed and the volume and mass throughput that was indicated by previous experiments (Khodier et al., 2021). The test runs for the calibration curve were done at 50%, 60%, 70%, 80%, 90%, and 100%. Each setting was done twice and lasted for 15 minutes, with the fixed shaft rotation speeds selected in random order, and the same commercial waste heap was utilized as feed for all runs. The 10% steps from 50% up to 100% were chosen because this was the recommended operating range and since a higher average mass and volume

output was expected at higher shaft rotation speeds (>60% maximum rpm) from previous experiments by Khodier et al. (2021). By starting at 50% the limits of the recommended operating range are included and the results from Khodier et al. (2021) can be verified. The 10% steps were the result of the minimum step distance the shredder shaft rotation speed could be set at and the 15 min were chosen to ensure that the data gathered was dominated by the shredder operating at a steady state i.e. the initial oscillations have passed. The data from these runs were incorporated into the control loop experiments.

The second test runs were step testing and were not able to be directly applied for the experimental control loop. The test runs consisted of the usual set up and the shredder changing its set shaft rotation speed every 3 min by 40% i.e. from 50% to 90% and back again. The 40% jump was done to maximize the output change of the shredder with the goal of identifying the response time of the shredder more easily. This jump was repeated 8 times.

The third and last of the test runs done at the Müllex facilities in July were the control loop experiments. The test runs were 15 min long, set at 70% of the maximum shaft rotation speed and to alternate between the control loop activated and deactivated. The alternation was done to minimize the chance of any waste material change to disproportionately affect either the active or inactive runs. The 15 min were chosen to balance the interest of long run times with many repetitions. The control loop was programmed by Komptech GmbH, and was set up to keep the current volume output flow within the 10% boundaries from 180 m^3/h determined by the linear model from the calibration curve test runs. The control loop was also limited to only change the PV once every 30s, to reduce the chance of over stimulating the control loop, since it was not possible to determine the response time before the experiment. To keep all the data sets organized the experimental data was exported into a file with each run as an extra set. The results from the experiments in July were saved as VMJ_oR(1-9) and VMJ_mR(1-9) which stands for “Versuche Muellex Juli ohne/mit Regelung” (Experiments Muellex July with/without control loop). This nomenclature is referred to when addressing specific data sets. Other data sets from different experiments from within the ReWaste 4.0 and F projects are also used and are individually named.

The exported CSV-File has datapoints every few thousandths of seconds which is why the files can have well over 10000 Datapoints at the beginning. Since these files are just columns of numbers the size is significantly smaller and also rather uninformative by themselves. The head of an example file can be seen in Figure 12.

```
Time (s),Volumen optisch H Sensor (m^3/h),Bandwaage (t/h)
24.830,0.0228450894355774,0.0130171123892069
24.896,0.0241386331617832,0.0125608732923865
24.963,0.0184345711022615,0.0129984905943274
25.030,0.0179202705621719,0.0130574610084295
25.096,0.0217229779809713,0.0130388382822275
```

Figure 12: Head of a CSV-File containing Volume and Mass Stream Data.

3.2 Data Analysis

Time series analysis is a statistical analysis method that is well supported in the programming language R. Therefore the decision was made to develop the tools for the data analysis in R. In order to analyze the data the files need to be prepared so that the statistical tools of R can be utilized, then the statistical analysis is done.

3.2.1 File Preparation

The first few steps of the analysis are simply cleaning up the data and preparing it for further calculations. The entire Code will not be shared here so only the essentials will be mentioned and commented. However the uncommented code can be seen in the appendix.

First of all the CSV-Files are loaded and only every 60th value is saved into a data frame as this results in a 4s time difference between each datapoint. This time spacing was chosen because of how the sensor stored its data. As the measuring unit for both the volume and mass streams generated datapoints every 0.06s but a new value was updated far less at just under 4s, i.e. the data points form a step function. So a the timeframe of 4s was chosen in order to avoid any duplicate datapoints later on in the analysis. In Figure 13 the 4s interval is marked and the two graphs are the volume and mass streams. This illustrates how the value for mass and volume only updates every 3,9s and therefore all the exported data points in between would interfere with the time series analysis in the following calculations, as this would distort the results of the analysis. It is better to air on the side of sampling the data a bit to less and miss a data point than to get data points double since this would suggest a correlation that does not exist while the other error would only slightly hide a correlation that might exist, thus making any discovered correlation more robust.

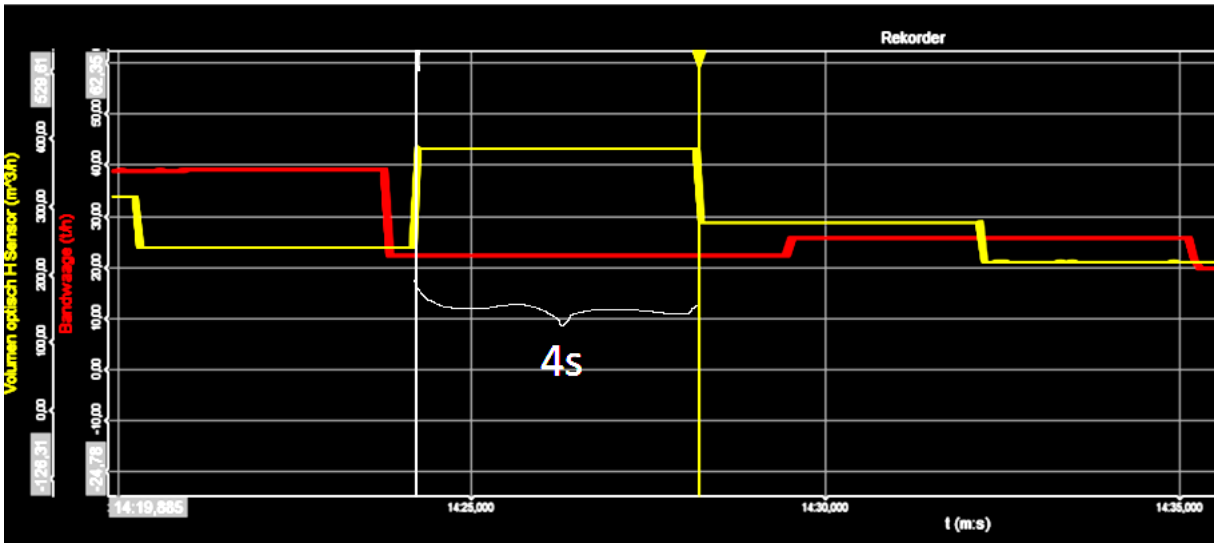


Figure 13: The raw datapoints with constant values for every ~4s

The goal to use a form of a control loop to limit the short and mid-term fluctuations requires a better understanding of the shredder parameters and their influence on the output fluctuations. As Curtis et al. (2021) points out there are fluctuations caused because of the loading of the

hooper with a wheel loader and others are caused by material inhomogeneity. To this end the calibration curve was recorded by running the shredder for a set amount of time at a defined shaft rotation speed in a random sequence.

3.2.2 Mass Series Analysis

While the results of the regular time series analysis are discussed in section 4.4, the analysis of the data prompted a slightly different approach. The Hypothesis that the correlation between the data points is actually better described as mass-related rather than time-related is further analyzed with the next developed method: Mass series.

Since the theory behind the correlation of the volume stream is that the material is shredded differently and therefore different types of material will produce a different volume and mass output. The lower the plastic contribution to the overall stream is the higher the mass output becomes and is also more dominated by pulp-based materials (Curtis et al., 2021). This material correlation should be reflected in the data series analysis. As the time series is also a fluctuating mass stream the two are obviously proportional, i.e. the more time passes the more mass passes by. This connection is exploited to transform the time series into a mass series. Conceptually a time series is just a list of datapoints that are spaced at equal distances in time. Applying this concept the equal distance in time for a time series is what an equal distance in mass is for a mass series, i.e. the same amount of mass that passes by per data point. It could also be thought of as a conveyer belt of buckets with equal mass within them but differing volumes.

To transform the timeseries data to a mass series data the current mass flow is multiplied with the time until the next data point. This mass increment is then added up until a target value is met and the whole process repeats till the end of the data set. This process is illustrated in Figure 14. The target value is basically the chosen separation between two consecutive data points in the mass series. The separation is now a unit of mass rather than time. A few other variables (n, i, c_m) are necessary for the code to function, however these are only indexes and control variables to keep the associated while-loops running.

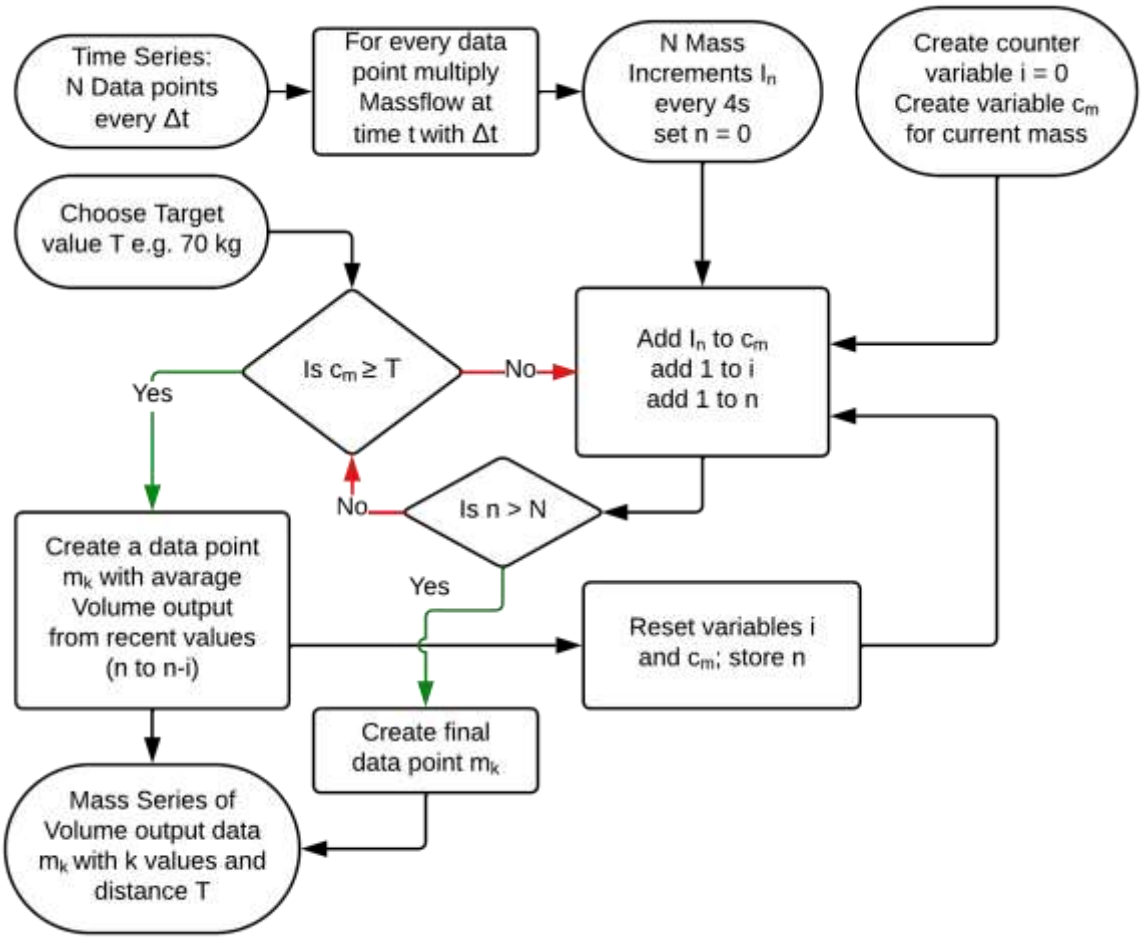


Figure 14: Flowchart for mass series algorithm

The transformation of a time series will result in a shorter mass series, for a target value greater than the mass flow between two data points of the time series, because it requires several data points from the time series for one value of the mass series. This means that a longer data set is required than the 15 minutes from the control loop experiments to ensure that the resulting mass series has a few hundred data points. The longer the data set the better the prediction model can be trained. Therefore the data set used is taken from a ReWaste F experiment done under similar conditions at the same place using the same mixed municipal waste, but the data set runs for more than an hour and will be referred to as VRW data.

As the data is recorded discretely at 4s intervals the mass increments are also discrete, i.e. the addition of these elements result in steps depending on the current mass flow for the last 4s. To keep the code simple a minimum target value is chosen since the mass increment steps are not equal and have a spread that will be transferred to the separation between the final data points of the mass series. In other words, the mass increments created are not equal but have a distribution described by a 90/10-quantile ratio. The 97th-Percentile is 50 kg and the median is 26 kg which limits the variation of around the target value. This is illustrated by the reduction of the 90/10-quantile ratio from 4.8 at the start to 1.4 when the mass separation for

a given target value is calculated, i.e. the mass series data point separation becomes more homogeneous than the mass increments.

The actual variation for the target value of 70 kg is shown in Figure 15 and has a variance of $1.39*10^{-4}$ which has a minimum for the data set VRW. The variance is a function of both the spread of the data and the number of data points, since with a larger target mass increment the number of data points will become less and the absolute spread of the data is relatively constant the variance is consistently in the order of 10^{-4}. How significant the variance of the distance between individual data points was not further studied for this thesis, however a further analysis of the behavior of the time series could be done with a strictly statistical interest.

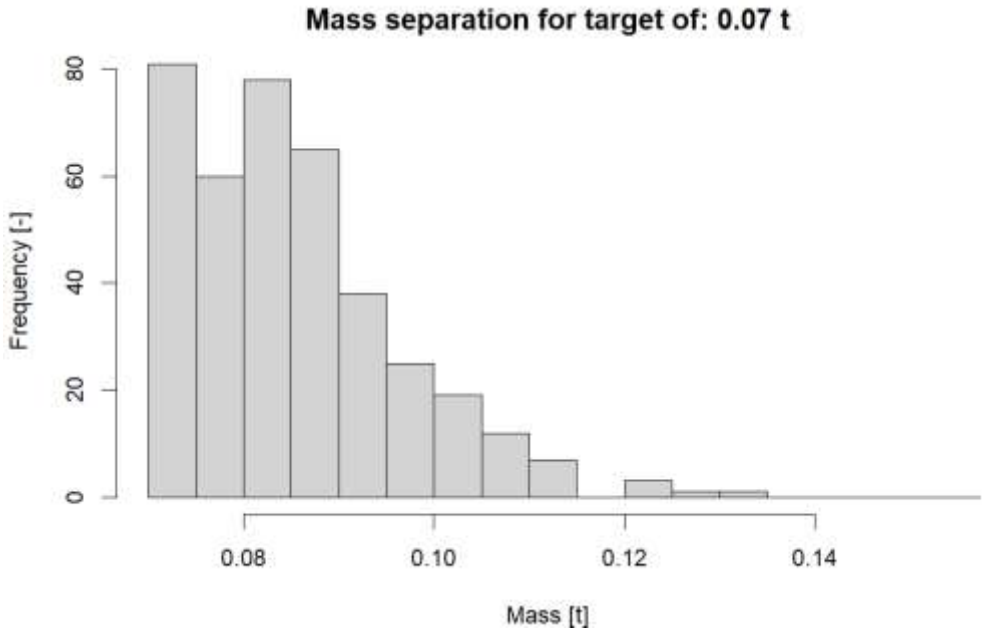


Figure 15: Histogram for value of mass between each data point of mass series

As Figure 15 shows the histogram and the 97th Quantile is at 0.11 which illustrates that the code for a simple transformation of the time series to a mass series can be optimized, but for this thesis the deviation is accepted and the actual median of 0.083 t is noted. For the purpose of the mass series analysis the actual target value is less significant than the spread which is relatively small and can still be used to compare the utility of mass series over time series.

Using these Mass series the best ARIMA Model is fitted and the AR(p), level of differencing (d) and MA(q) terms are chosen for the lowest AIC value. This (p,d,q)ARIMA model can be forecasted for the next n steps which for this analysis n was set at 20 and can be seen in Figure 16. This translates to 20*target value so e.g. for every 50 kg the forecast is for 1000 kg material into the future. 1000 kg is equivalent to 153 s using the 26 kg per 4s data point.

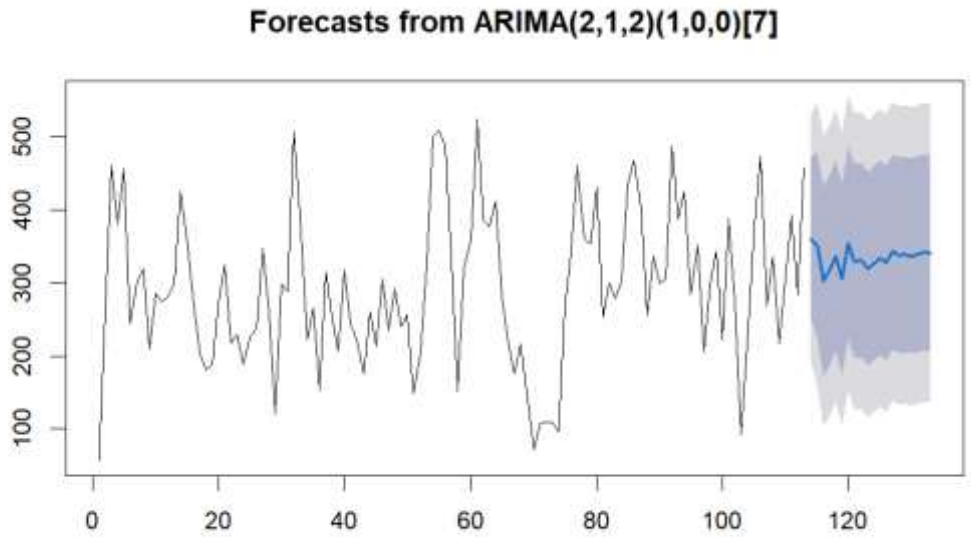


Figure 16: ARIMA (p,d,q)(P,D,Q)[S] Model forecast for 50 kg increments with 95% and 80% confidence interval

The results of the models for different target mass increments are shown in Table 5.

4 Results and Discussion

4.1 Calibration Curve for Control Loop

Experimental runs were conducted to determine the connection between the speed of the shaft rotation and the output of volume and mass from the shredder. The goal was calibrate the control loop by establishing a clear connection between the shaft rotation speed and the mean volume output. Khodier et al. (2021) proposed this idea as an observation from those experiments.

As a protection mechanism the shaft will stop its rotation and reverse for half a turn and then start up again if anything blocks the cutting tools and also at regular intervals every few minutes to clear the shredder cutting compartment. Also as a result of the power coupling the higher the rotation speed the lower the torque becomes. This leads to the effect that when the shaft load rises above what the motor can provide at that speed, the shredder will automatically reduce the shaft rotation speed in order to increase the torque. Therefore for the higher target rotation speeds it was observable that the actual rotation speed would vary more than for lower target values.

In Figure 17 the gathered data for the calibration curve, i.e. the long-term fixed rotation speed are plotted as the mean and median volume flow over the entire time at the set rpm. The data points indicate a trend of a directly proportional function i.e. the higher the rpm of the shaft the higher the volume output, however it is important to remember that there is variance and that this data is a 15 min average, which limits its practical application.

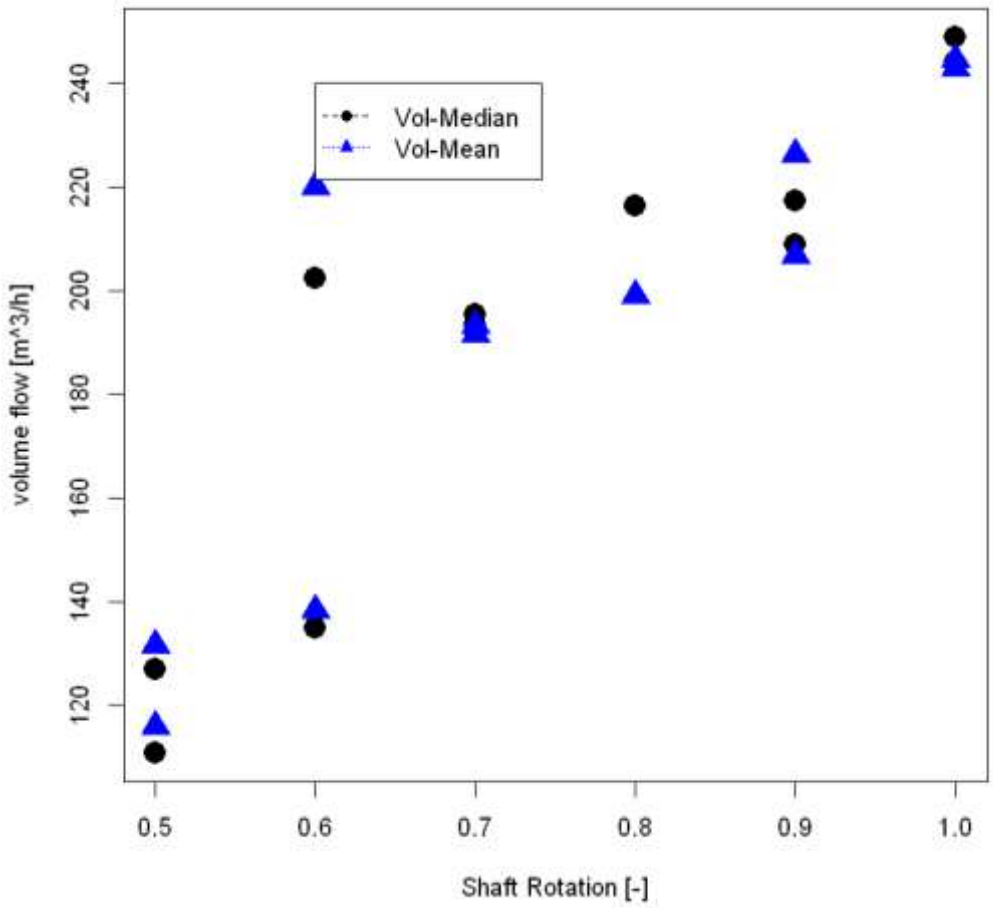


Figure 17: Volume output data for varying shaft rotation speeds (described as share of maximum rpm)

A similar yet less clear trend can be seen when plotting the mass flow data from the calibration curve experiments for the respective shaft rotation speeds in Figure 18. For the mass flow the trend seems to be less linear than with the volume flow, however both trends will be further described in Figure 19.

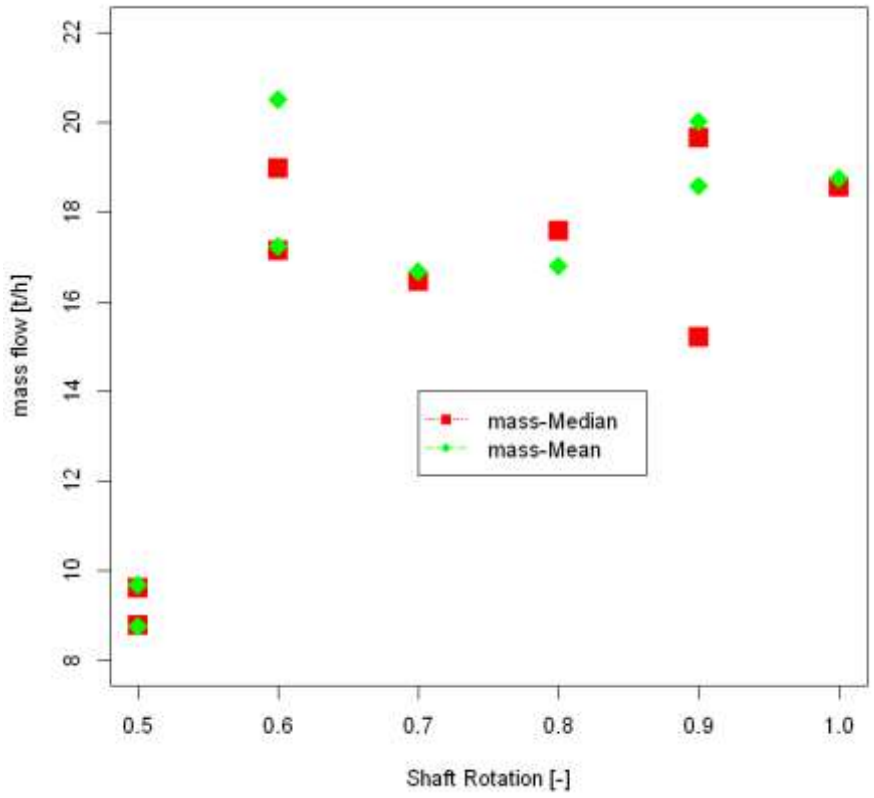


Figure 18: Mass output data for varying shaft rotation speeds (described as share of maximum rpm)

Using this experimental data a linear model can be calibrated for the datapoints and a confidence interval calculated. This is done using R and in Figure 19 the median data and the mean data is used to calculate the linear model. This shows that despite the large confidence interval, meaning the data distribution is wide or in this case there are not sufficient data points to lower the width of the interval without lowering the degree of confidence. However, there is a clear trend indicating a proportional increase of volume output with an increase in shaft rotation speed. A more precise function, i.e. lower the width of the interval, would need more data points to generate a higher degree of statistical confidence.

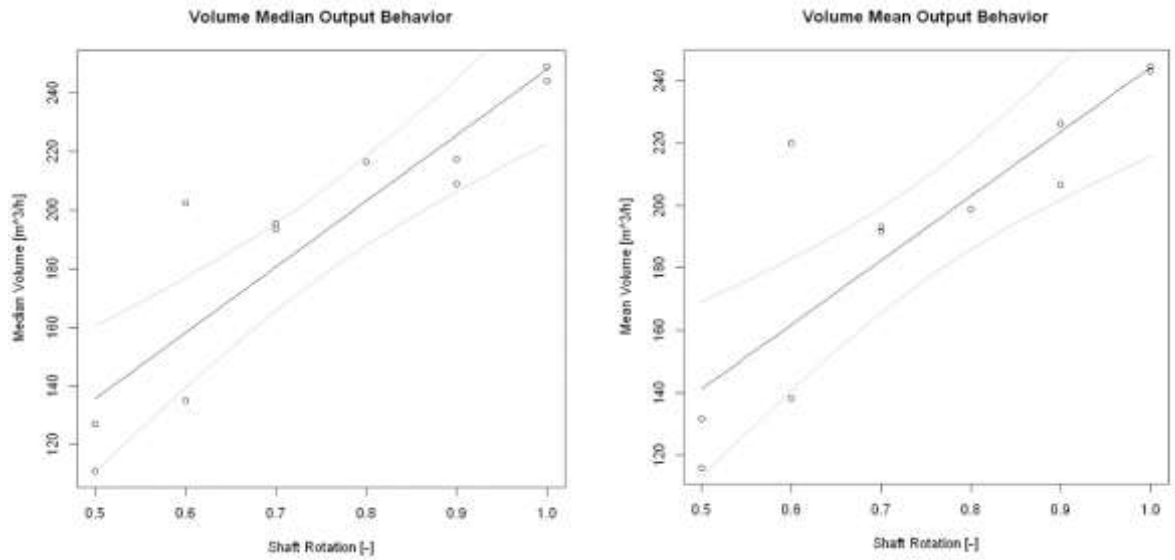


Figure 19: Comparison of Linear model for output behavior with confidence interval based on median(left) and mean(right) output, Shaft rotation described as share of maximum rpm

The values of the modelling for the median are listed below and are calculated with R:

1. Quadradic model for median: $y = -342.1 * x^2 + 739.5 * x - 159.5$
2. Linear model for median $y = 224.93 * x + 23.23$:

The comparison of these two models is done by looking at how good of a fit each model is for the data and then testing if all the terms of the model are necessary. This is done in R with the drop() function.

The results of the modelling for the mean are also calculated:

1. Quadradic model for Mean: $y = -262 * x^2 + 600 * x - 101.5$
2. Linear model for Mean: $y = 205.88 * x + 38.41$

The purpose of the drop1() function in R is to aid in variable selection and model comparison by examining the impact of dropping individual terms on the overall model fit. By comparing the AIC, the F-values and p-values, you can assess the importance and significance of each term in the model. Since the p-Values for the F-Test of the linear model for dropping the x² term are above any standard significance value (α = 0.05) the null hypothesis is accepted that the term is insignificant, in addition the AIC is lower for the model with only x. Therefore the linear model with just the x term is selected. The values for the F-Test and AIC can be seen in Table 2.

Table 2: Results of F-Test for models of Mean Volume Output;
Model: a*(Shaft rotation)^2 + b*Shaft rotation + c = Mean Volume Output

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F) (p-Value)
<none>	NA	NA	4726.505	72.69351	NA	NA
Shaft roation	1	1054.6911	5781.196	72.90918	1.785152	0.2182738
(Shaft roation)^2	1	458.9477	5185.452	71.71289	0.776807	0.4038212
Shaft rotation (Linear Model)	1	14720	19905.455	84.50939	25.5484	0.0006861009

Using the linear model calculated for the mean volume output, the range for the control loop was chosen to be centered at 0.7 since then a 20% increase or decrease in Shaft Rotation speed leads to an approximate 20% proportional change in mean volume output. By centering the target volume output at the approximate 0.7 value for the shaft rotation speed it was ensured that the shredder would be able to both increase and decrease the shaft rotation speed without encountering the boundaries of the shredders chosen operating range. Since the 95% confidence intervals allow for many possible other linear models of which for some the change of 20% for the shaft rotation speed can lead to as much as 35% output change or as little as 10% output change compared to the 180 m^3/h at the chosen center of 0.7 for the shaft rotation speed. The output value of 180 m^3/h was chosen since it is the value the linear model for mean volume output calculates for the chosen set point. Accepting a directly proportional linear model for the calibration, this means the more dominant the shaft rotation speed change is the more likely the output will actually change in any significant way. With this calibration model in place, an important part of the tuning process was completed and the control loop was set up.

4.2 Step Testing for Control Loop Tuning

The previously described calibration curve experiments of the response only accounted for the overall 15 min response, not the short term reaction behavior. The next experiment attempted to determine the dead time (t_d) and lag time (τ) in the control loop. Determining either of these proved to be a particularly difficult challenge, because this involves doing step testing, and moreover since the fluctuations of the process variable are greater than the change introduced by the control variable. The long term calculated change in mean volume output by the control variable is 80 m^3/h, as seen in Figure 19, and the fluctuations can regularly exceed 100 m^3/h, as seen in Figure 21 with the first derivative. These fluctuations make it difficult to determine the times at which the shredder output actually responds and how fast. As can be seen in Figure 20 despite a large change in shaft rotation speed it is not obvious at what time the volume output changes to match it. Taking a rough estimate, the total response time of t_d and τ is 11,6s. The time difference between the control variable changing the target rpm of the

shaft and the shredder shaft rpm responding is approximately t_1=1s. This time delay is both more accurate and shorter than the process variable changing, however it is worth remembering that t_1 does add time to the total control loop settling time by effectively being dead time as far as the volume output is concerned.

The step testing was done by switching the target shaft rotation between 50% and 90% speed every 3 min. This time frame for the settling of the step was chosen to optimize the experimental time and generate some more step changes. If the step change takes longer than 3 min to settle then the response time is too slow for any effective control loop looking at smoothing short term to mid-term fluctuations between 4s and 200s. Therefore a longer settling time is irrelevant for the development of the control loop and 3 min is a compromise between an infinite settling time and cutting off the settling prematurely.

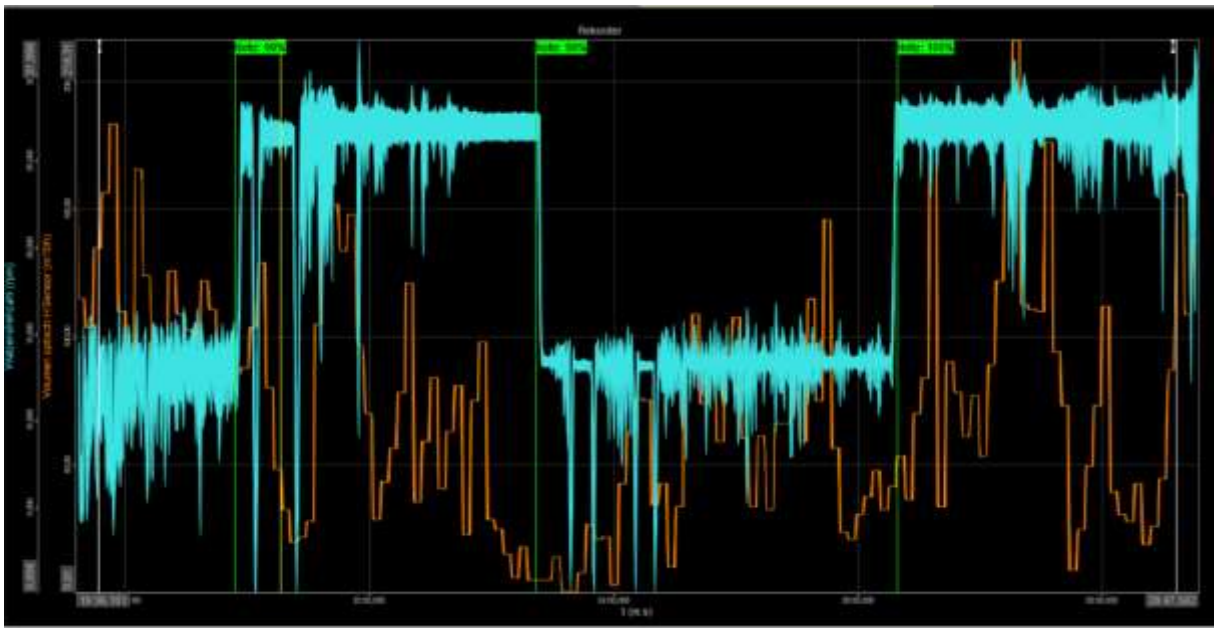


Figure 20: Results from step-testing between 50% and 90% shaft rotation speed (blue), Volume output (orange) is process variable (PV) without clear response time to the step change.

A more robust analytical tool must be developed to properly determine any response time and have some sort of standard to compare results to. The discussion to what possible methods could be plausible will be in the section: 5.

4.3 Control Loop Test

The prepared CSV list can be stored and visualized in a diagram. Before any fancy mathematical tools are applied to the data it is useful to get a first impression of how the time series behaves since this is the starting point for any proper analysis. So a better understanding is gained, an example of such a raw file is displayed in Figure 21. The short term fluctuations (one datapoint every 4s – solid line) vary significantly while the function with linear decay smoothing for the 15 previous values (dashed red line) varies less. The difference function in the diagram below simply calculates the difference between two neighboring data

points, i.e. the first derivative. As the bottom of Figure 21 shows the differenced data resembles white noise with no medium or long term trends. As discussed in section 2.3 this means the differenced data can be used for further analysis.

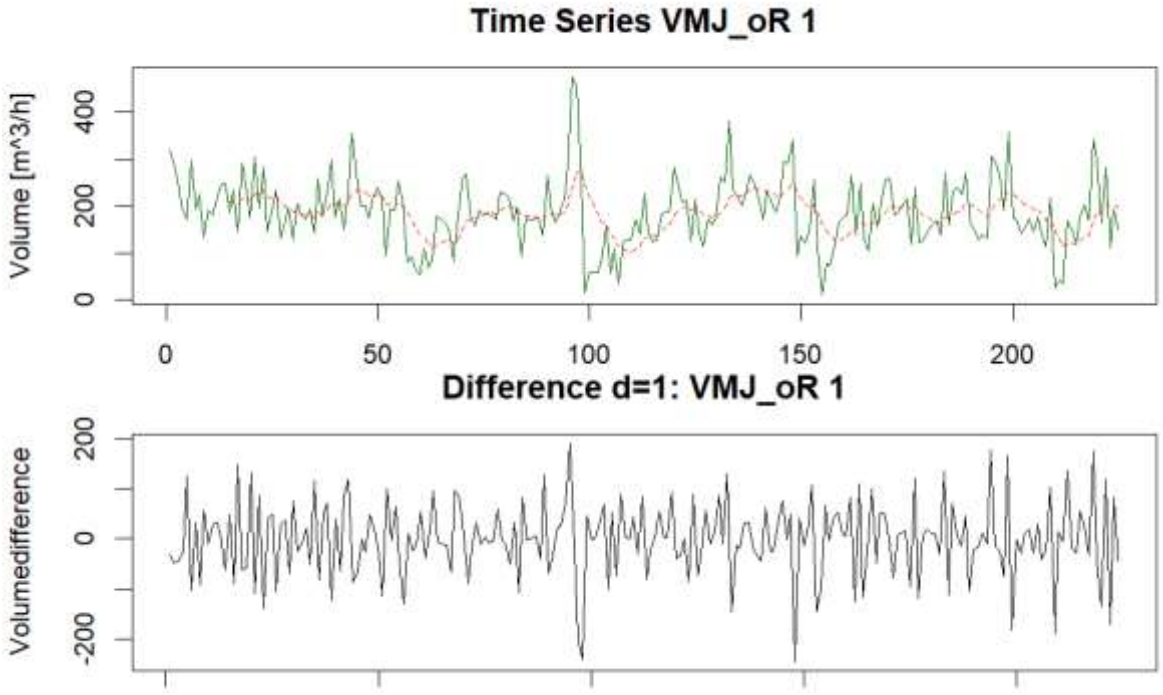


Figure 21: Time series of the first runtime without a control loop: top – data point every 4s (solid green line), 60s fluctuations (dashed red line); bottom – first derivative of the 4s time series

The possibility of a control loop was tested by using the available information from the calibration curve and having a simple control loop change the shaft rotation speed if the volume output went out of bounds for a specified amount of time, as described in Section 3.1.2. The next step is to determine whether a statistically significant improvement in the fluctuations was achieved in the test runs with versus without the control loop.

The 90/10 Percentile ratio has been used before and is chosen in this thesis as a metric of how strongly a waste stream fluctuates (Curtis et al., 2021). An average of all the 90/10 Quantiles was made for both the runs without and with a control loop. These 90/10 Quantiles were tested for normal distribution using a Quantile-Quantile Plot, as well as the Shapiro-Wilk test that can be seen in Figure 22. If the p-Value is below the significance level, then the null hypothesis (H_0: Data is normally distributed) must be rejected. Since the Q-Q plot is mostly insignificant, considering that there are only 9 values, the indication of a divergence from the normal distribution can be neglected and the p-Value accepts the null hypothesis, the data can be assumed to be normally distributed. Therefore with only 9 data points the histogram and QQ plot are not reliable, so despite no confirmation of normality from these plots, for the sake of simplicity and based on the Shapiro-Wilk Test the normal distribution is accepted. Further experimental data should be used to better verify future findings.

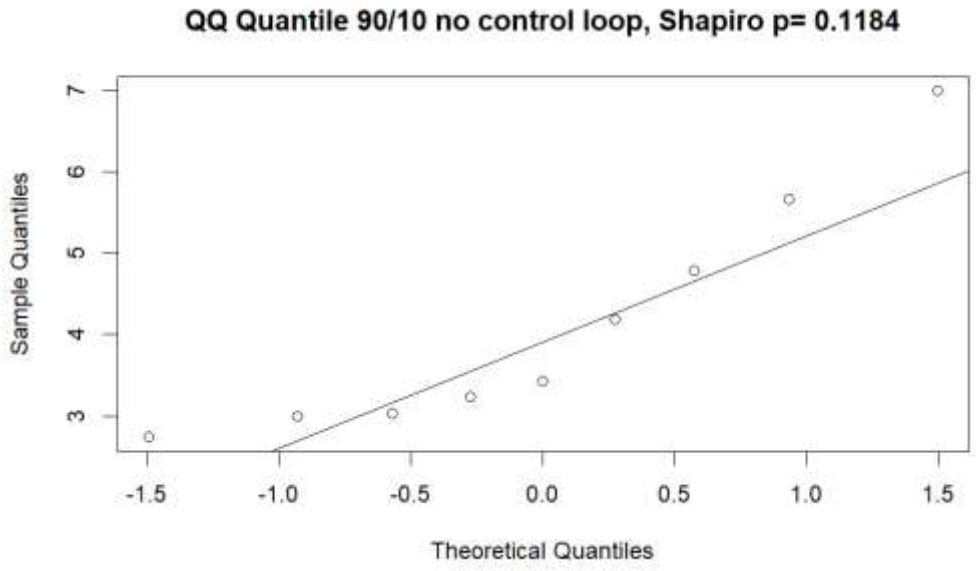


Figure 22: 90/10 Control loop quantiles

Since the 90/10 Quantiles are assumed to be normally distributed a two sample t-Test can be done to determine whether there is a significant difference between the 90/10 Quantiles without and with a control loop.

The result from the control loop tests are displayed as boxplots of the volume data from each test run and can be seen in Figure 23. Looking at the width of each boxplot there does not seem to be any significant difference between the two settings, with and without the control loop active. The medians are comparable and the 75th and 25th percentile are also comparatively distributed.

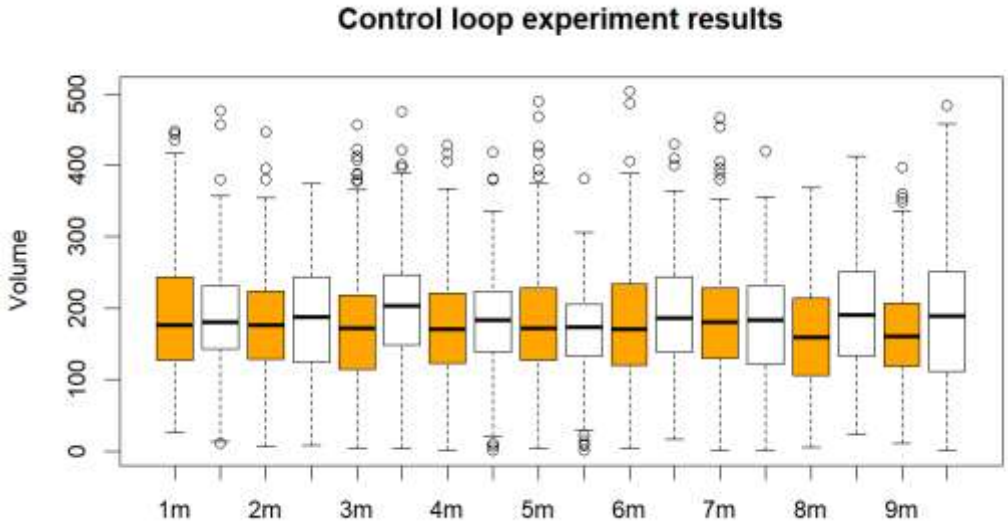


Figure 23: Test runs with(orange) and without(white) control loop

In Figure 24 it can be seen that the confidence interval for the difference of the mean volume output goes below zero and the confidence interval overlap completely. This means that the control loop made no significant difference to the 90/10 Quantile metric. So while the standard deviation of the 90/10 Quantiles for the group with the control loop active is smaller ($\sigma = 0.8$) than without the control loop ($\sigma = 1.4$), the actual Quantiles are not significantly better because of the control loop. In other words, the control loop made no meaningful improvement.

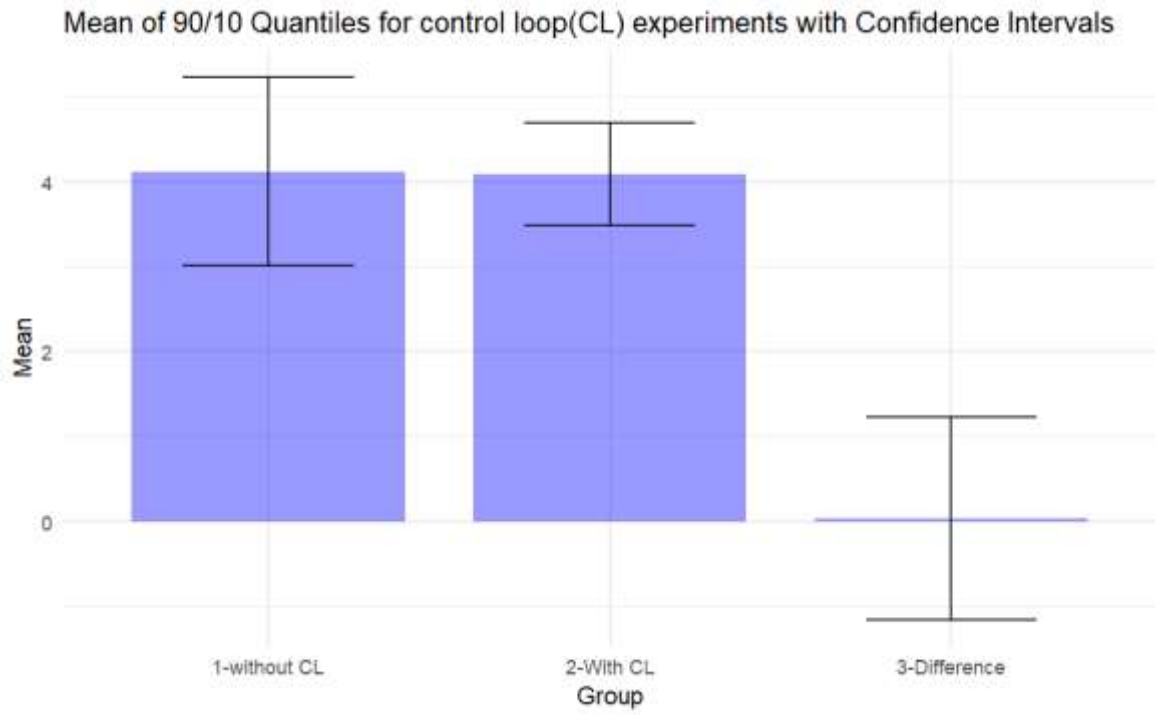


Figure 24: Mean of 90/10 Quantiles using a confidence interval of the mean for unknown variance; 1- without control loop; 2 - with control loop (CL); 3 – Difference

4.4 Time Series Forecast

The result of the control loop experiments from section 4.3 begged the question of why the control loop did not make a meaningful improvement. So, the next analysis will focus on the time series analysis and material correlations of the shredder output.

Going through the basics of time series analysis the first step is to plot the data and see if there is any obvious trend, seasonality or other significant anomalies in Figure 25. Using linear decay smoothing the dashed line is added to better estimate the medium-term fluctuations with a 60s window, i.e. the most recent 15 data points (15*4s=60s) are used to calculate the current average. Linear decay smoothing is used to create the dashed line in Figure 25 by calculating linearly decaying values for the most recent 15 data points. This is only relevant for the best way to calculate the smoothed time series, which can help to visualize the relevant fluctuations better.

Some things to take notice of in Figure 25 are the significant reduction in fluctuations achieved by a 60s smoothing. This should come as no surprise, as it indicates that the greatest potential

to control the output lies in the short-term fluctuations rather than the long term. Also looking at Figure 25 there still remain mid-term fluctuations in the range of 120-400 s that are likely to be outside the control loops ability to influence.

The goal of any control loop would be to reduce the short to medium-term fluctuation as those between 10 and 120s, since the smoothed values show the effect of the fluctuations in this time frame. The fluctuations in these data sets cannot be attributed to discontinuous feeding of the shredder, since a crane operator was tasked to ensure the feeding hopper of the shredder was always full. Some issues described by Curtis et al. (2021) such as bridging or the material and shape of the objects were not addressed in this thesis.

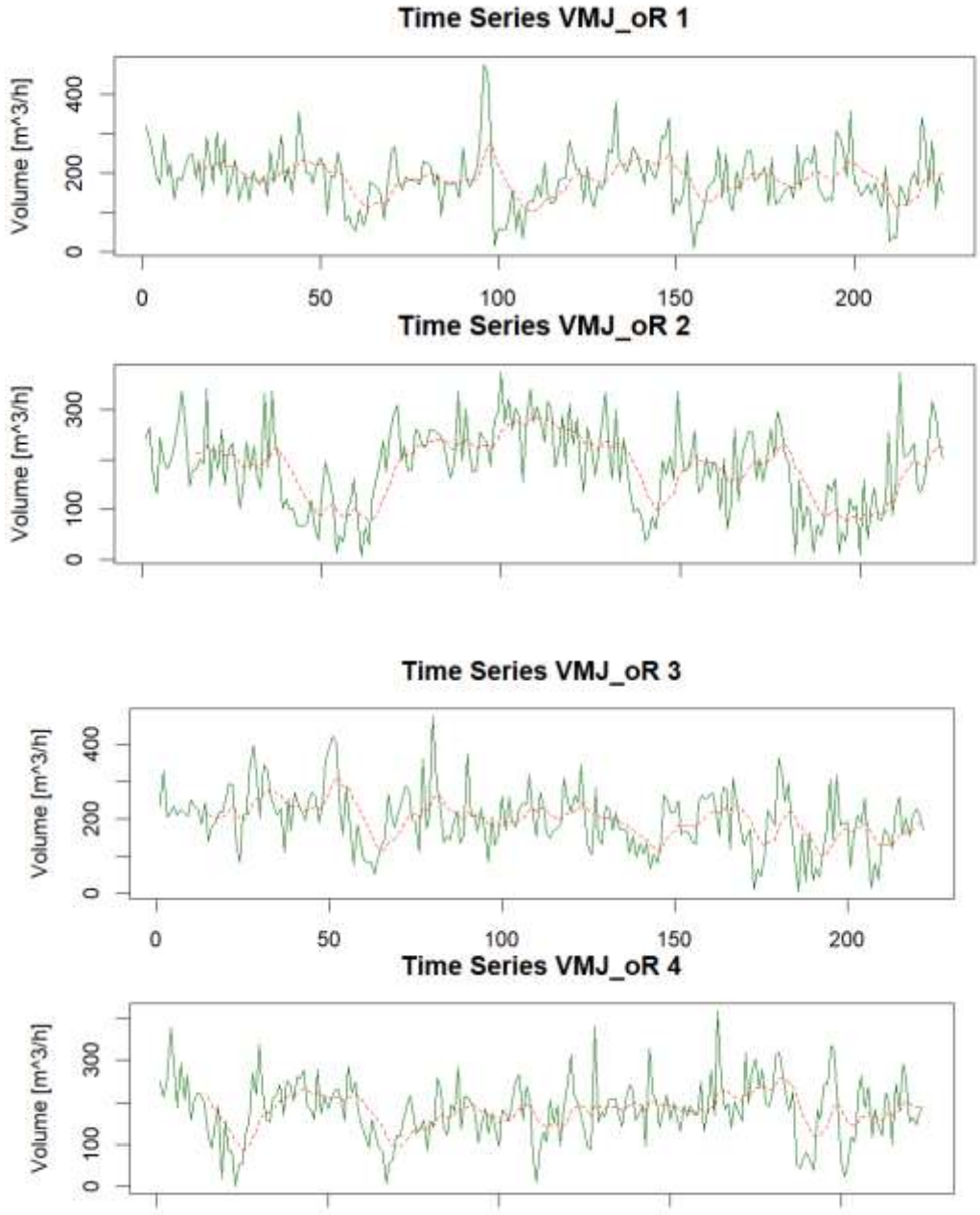


Figure 25: Time Series of volume output data from July experiments at Müllex with no control loop (VMJ_oR): solid line – 4s data point intervals; dashed line – 60s mid-term fluctuations with linear decay smoothing

There is no obvious trend i.e. an integrated part for all the data sets. However, for longer sections(300s) there can be an overall increase or decrease in the output average, which is better seen with the 60 s smoothed values (e.g. in Figure 25 for VMJ_oR 4 for the smoothed values from approximately lag 70 to lag 180 there is an output increase from 120 to 220 m^3/h) and therefore the differenced data i.e. 1st Derivative is also displayed in Figure 26.

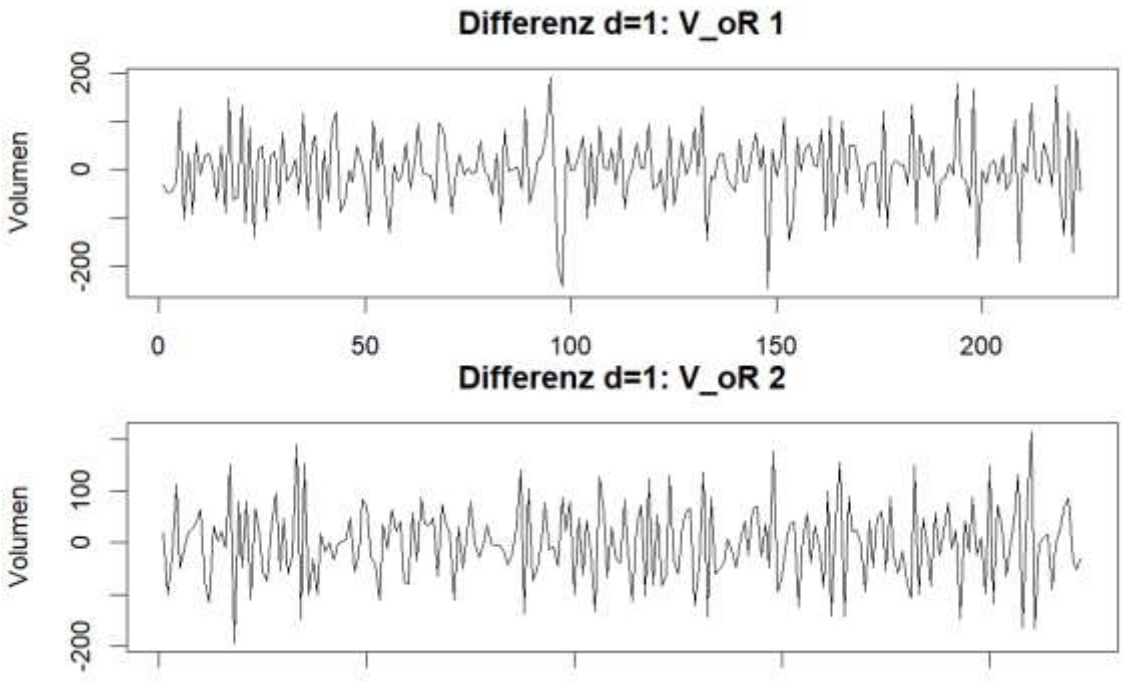


Figure 26: Differenced volume output data

The differenced data looks similar to white noise i.e. no recurring patterns for all the data sets and there is no trend or seasonality visible in any of them and the ACF and PACF in Figure 27 also have no slow trailing off but a rather sharp cut off. The Box-Pierce test used for determining correlation between data points is applied and as the p-Values for all the data sets are well below the significance level of 0.05 the null hypothesis (H_0: The datapoints are independent) is rejected.

As a correlation between the data points is given, the next tool to get a more accurate understanding of the time series is to use the ACF and PACF as displayed in Figure 27. These diagrams help to estimate the order of the best ARIMA model, as described in the chapter: 2.3.

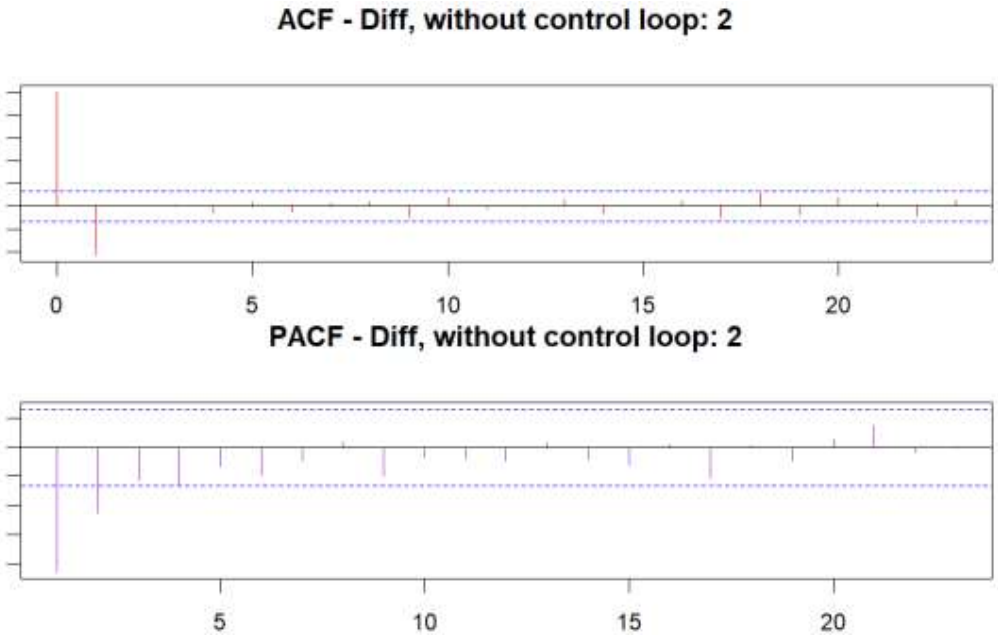


Figure 27: ACF and PACF of differenced data (d=1) with no control loop

All the 9 data sets have a similar appearance, as seen in the Appendix. They all indicate an MA(q≤2) and AR(p≤3) with little to no seasonality or recurring pattern. Therefore a few models were tested with the parameters varied around these values and using the AIC result the best fit was chosen. The Box-Pierce test should give a p-Value high enough to accept the null hypothesis, as the residuals of the model and observed data should be independently distributed. Some of the models are displayed in Table 3 where for the lower AIC values the p-Value is consistently high enough to not reject the null hypothesis. SARIMA models i.e. models that use P,D,Q variables are used to account for any kind of seasonality.

Table 3: SARIMA Models compared to observed data from July 2022 “Versuche Müllex Juli ohne Regelung (VMJ_oR)” for lowest AIC values

VMJ	p	d	q	P	D	Q	AIC	SSE	p-Value
oR1	1	1	1	1	1	1	2492.720	907271.0	0.1790
oR1	1	1	1	0	1	2	2492.739	906884.5	0.1743
oR1	1	1	1	0	1	1	2493.075	921020.2	0.0627
oR1	0	1	1	0	1	2	2500.235	980806.1	0.3296
oR2	0	1	1	0	1	1	2459.198	931960.3	0.9402
oR3	1	1	1	0	1	1	2493.896	1066366	0.2990
oR4	1	1	1	0	1	1	2474.001	892150.5	0.0900
oR5	0	1	1	0	1	1	2367.654	777565.1	0.9116
oR6	1	1	1	0	1	1	2411.778	1000909	0.3109
oR7	1	1	1	0	1	1	2386.919	824207.7	0.7346
oR8	0	1	1	0	1	1	2424.830	855878.3	0.9815
oR9	0	1	1	0	1	1	3489.705	1676321	0.9810

While this method will generate the possibility to create a forecast, meaning predicting future values based on the generated model, the model must be better than the bench mark of forecasting, which simply takes the last empirical data point and extrapolates that as a stationary process for the next 20 values i.e. the next 80s.

The time series model for the VRW data from the November experiments calculated the forecast in Figure 28. The model and the benchmark (using the last measured value for the next 20) is compared with the measured datapoints to check their accuracy. The numerical result for the sum of squared erros (SSE) is calculated and a ratio is created to better see how much smaller the Forecast is compared to the benchmark:

- SSE of Benchmark: 366299
- SSE of Forecast: 247776
- Forecast/Benchmark: 68 %

This is done 5 times to reduce the chance that the benchmark was only particularly unfortunate. Also another benchmark is compared where the last 5 values are used with a weighted average, i.e. linear decay smoothing, to smooth out the most recent value with 4 other data points. The results are presented in Table 4.

Table 4: Time series forecast improvements compared to benchmark models, last value (the most recent value) or a linear decaying weighted average of the last 5 values (LD 5); The Index is for identifing the individual forecasts

	Benchmark type	
Index	Last value	Last 5 values (LD)
1	77,1	55,7
2	-2,5	-1,7
3	-9,1	4,3
4	32,4	68,0
5	67,2	48,8
Mean [%]	33,0	35,0

Comparing the average across all 5 improvement ratios, it becomes clear that the linear decay smoothing over 5 steps (LD 5) is actually slightly worse than taking just the last value. The takeaway is that for both benchmarks the time series prediction improvement averages around 35%.

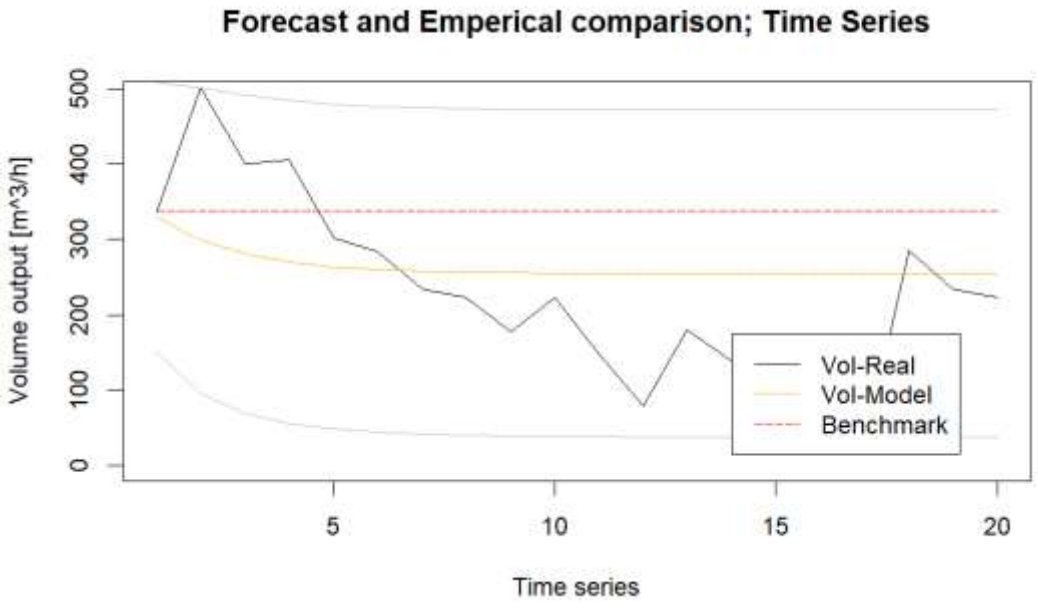


Figure 28: Forecast model for Time Series Data from Experiments done in November 2022 (File name: “VRW - Nov alte Zähne”); Vol-Real – is the actual Volume output from empirical data points, Vol-Model – is the predicted volume output from the generated model with a 95% confidence interval (grey), Benchmark – is the prediction using just the last real value to forecast the next 20 datapoints

4.5 Mass Series Forecast

The developed method of the mass series was applied to another ReWaste F data set which had a longer run time than the control loop experiments and provided another real world alternative for the application of the forecasting method. The method must function on all data

sets that are produced in a similar fashion otherwise it would only produce a model that is useful for that particular set up and day which would not be a very useful model. The different target values and their calculated results compared to the actual results are shown in Table 5. The target value is the mass amount before the counter resets or using the metaphor of the bucket conveyor belt, the target value is the size of the bucket. The SSE (Benchmark and Forecast) is the numerical value to compare the two forecasting methods while the fourth column states whether the SSE of the Forecast is smaller and therefore better. The last column shows the ratio of the two SSE values which indicates how much smaller the SSE value is. Looking at the ratio it becomes clear that the mass series forecast is significantly (70-80%) better than just using the benchmark value available to forecast.

Table 5: Comparison of forecast and benchmark models using SSE at set target values for the mass series

Target Value [kg]	SSE Benchmark [-]	SSE Forecast [-]	Forecast is better:	$\frac{SSE\ Forecast}{SSE\ Bench}$ [%]
50	572760	130056	TRUE	22.707
60	585560	109661	TRUE	18.728
69	623434	110609	TRUE	17.742
70	573250	123016	TRUE	21.459
80	598962	139369	TRUE	23.268
90	962229	296843	TRUE	30.849

The results of the modelling are graphed in Figure 29 where it shows the last 20 values of the VRW November data set compared to the forecasted values of the model with the 95% confidence interval. This illustrates how the model behaves. The images for all the other results from Table 5 can be seen in the appendix.

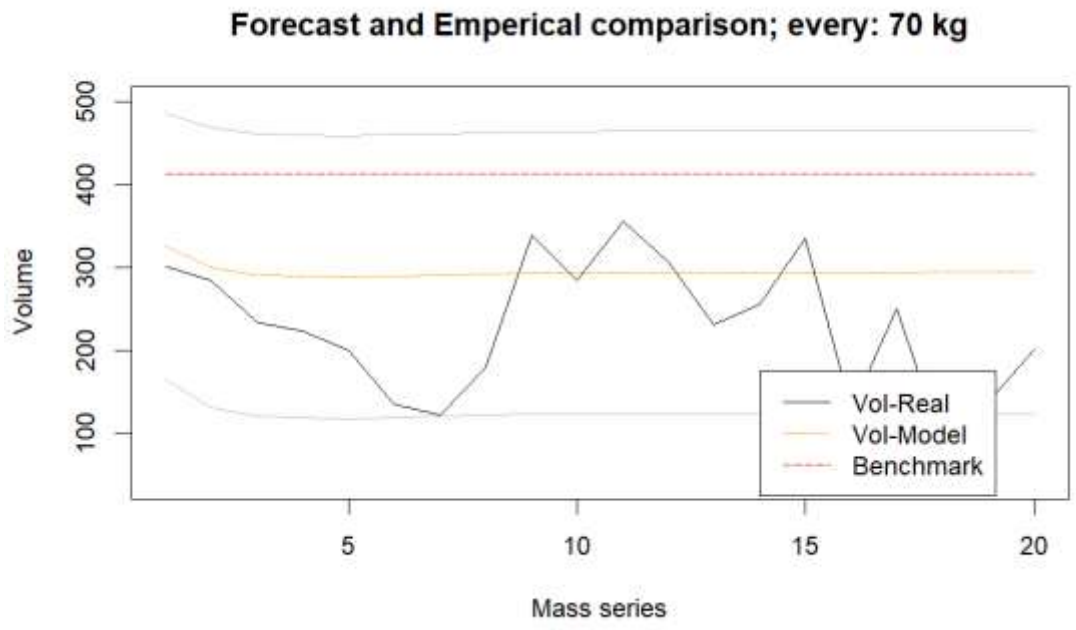


Figure 29: Forecast model for Mass Series Data from VRW - Nov alte Zähne

Table 5 means that with the mass series prediction model the output of the shredder can be predicted better than the simplest alternative of using the last value to predict the next value, which was used as the benchmark comparison. As the comparison between the size of the mass series increments (every 50 kg up to every 90 kg) shows is that the SSE difference is lower for lower mass increments i.e. the model prediction fits better for smaller mass increments. This tracks well with the fact shown in Figure 30, that the ACF and PACF peaks become lower for larger mass increments, meaning the correlation becomes less significant and therefore the basis for the model becomes weaker. The other mass series increment ACF and PACF diagrams are in the Appendix. This is particularly apparent with the target value of 200 kg where the PACF peaks disappear.

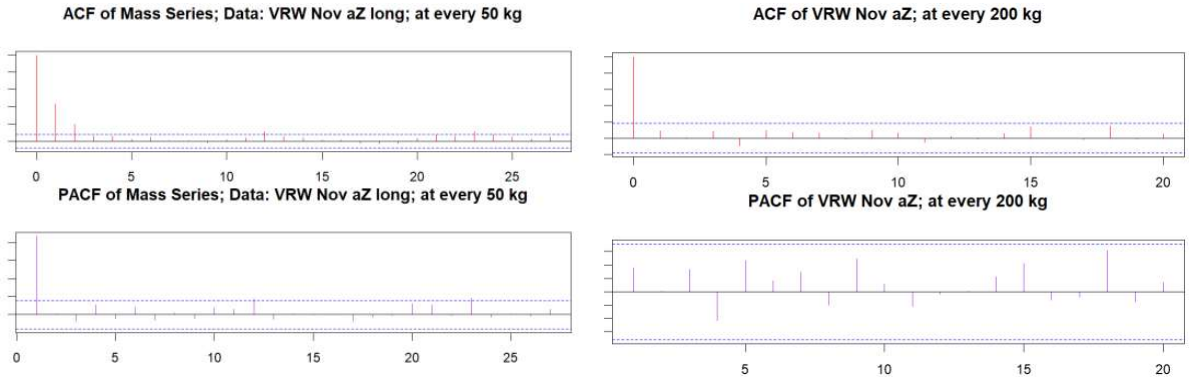


Figure 30: ACF and PACF of Mass Series for small target increment (50kg - left side) and for large target increment (200kg - right side); Notice the loss of significant peaks for the 200 kg increments compared to the 50 kg increments

The advantage of using mass series rather than time increments is not statistically analyzed with a significant amount of data sets. However, the mass increments appear to function more reliably and with better improvements from the models than using the time series since the ACF and PACF show a stronger correlation. This is collaborated by the fact that the mass series has an average improvement of 77% over the benchmark compared with an average improvement of 35% using the time series.

As mentioned in section 3.2.2, the transformation from a time series to a mass series results in a shorter data set because of the summing of several data points into one. This happens both for the mass flow and volume flow. Therefore, the transformation process could be improved to ensure an equivalence transformation between the two methods. In addition to the static transformation of the data sets, there is the issue of forecasting. The model creates a forecast for either the mass or volume flow. Therefore, if both are required to transform the data, once a forecast is made, this modeled data only has one of the two flows, and this is also only an estimation, so transforming it into the other method would require a proper error propagation analysis to estimate its feasibility. In short, a direct comparison between the two forecasting methods is not as straightforward.

The method used in this thesis to compare the forecasts was to take the same data set to do a forecast with both methods and use the same benchmark. This shows the effectiveness of the models in their own right.

5 Summary and Outlook

A Summary of the results will be discussed now and further research suggested:

- The relationship between shaft rotation speed and volume output was described for the tuning of the control loop with a best-fit model, which indicated a linear dependence over a 15 minute output average in Figure 19. The linear model that was calculated to describe the correlation between the long-term median volume output (y) and the shaft rotation speed (x) is: $y = 224.93 * x + 23.23$
- The step testing resulted in the necessity of a more robust determination of dead and lag time, since the fluctuations, even for constant settings, are greater than the average volume output change for the set shaft rotation speed.
- The initial control loop experiment resulted in no significant improvement for the 90/10-quantiles compared to the experiment with no control loop active.
- The time series analysis resulted in a mass series model that can generate forecasted values that are better than just taking the last available value. The forecasted values are up to 75% more accurate in terms of SSE for a 120 s forecast.

The shaft speed and volume output have a proportional dependence on each other for longer time frames. However, this does not translate well to time frames below 5 min. The strong fluctuations in the short-term are what the control loop would aim to reduce, which is heavily dependent on the ability of the shaft rotation speed to rapidly change the output. This is best characterized by the dead time and lag time of the shredder. Since the output varies by a large amount in the short-term, it is difficult to define terms at which the step answer is achieved and, even more so, at what point the dead time ends and the lag time begins. This is where another analysis, using the more accurate forecasting models, could aid this endeavor.

As the forecast predicts what the next 80s of the output would be without any change to the system, this result could be compared to the actual data after the step change. Assuming the model prediction is accurate enough, the comparison between the actual data with the change in shaft rotation speed and the predicted values of the unchanged system can be analyzed. The diagrams would require significant smoothing to achieve any sort of visual distinction between dead and lag time, but this would be a significant move towards creating a functional control loop that could be tuned and tested in another experiment. However, the available data from the experiments for this thesis should be enough to test this method of determining the response time of the shredder.

With an improved forecasting model over the method used at the beginning of the first control loop experiment, a different result may be expected for the output fluctuations of an improved control loop with a 75% better forecast and better tuning because of a more accurate measurement of the dead and lag time parameters. This could be the topic of further research.

6 References and Appendices

6.1 References

Ballini, R., Yager, R.R., 2014. Linear Decaying Weights for Time Series Smoothing: An Analysis. Int. J. Unc. Fuzz. Knowl. Based Syst. 22 (01), 23–40.

BAWP, 2023. Bundes-Abfallwirtschaftsplan. https://www.bmk.gv.at/themen/klima_umwelt/abfall/aws/bundes_awp/bawp2023.html. Accessed 28 April 2023.

Chambers, J.M., Cleveland, W.S., Kleiner, B., Tukey, P.A., 2018. Graphical Methods for Data Analysis. Chapman and Hall/CRC.

Chatfield, C., 1975. The analysis of time series: Theory and practice, 1. publ ed. Monographs on applied probability and statistics. Chapman and Hall, London, 263 pp.

Coskun, E., Pretz, T., Joost, C., 2018. Energieeffiziente Abluftbehandlung in MBAs.

Curtis, A., Küppers, B., Möllnitz, S., Khodier, K., Sarc, R., 2021. Real time material flow monitoring in mechanical waste processing and the relevance of fluctuations. Waste management (New York, N.Y.) 120, 687–697.

Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I., Tutz, G., 2016. Statistik : der Weg zur Datenanalyse, 8., überarbeitete und ergänzte Auflage ed. Springer-Lehrbuch. Berlin : Heidelberg : Springer Spektrum, xvi, 581 Seiten, Diagramme, 26 cm.

Käser, E., 2021. Fachlexikon der Mechatronik. http://www.fachlexika.de/technik/mechatronik/elektromotor.html.

Khodier, K., Feyerer, C., Möllnitz, S., Curtis, A., Sarc, R., 2021. Efficient derivation of significant results from mechanical processing experiments with mixed solid waste: Coarse-shredding of commercial waste. Waste management (New York, N.Y.) 121, 164–174.

Khodier, K., Sarc, R., 2021. Distribution-Independent Empirical Modeling of Particle Size Distributions—Coarse-Shredding of Mixed Commercial Waste. Processes 9 (3), 414.

Kranert, M. (Ed.), 2017. Einführung in die Kreislaufwirtschaft: Planung - Recht - Verfahren, 5. Auflage ed. Springer Vieweg, Wiesbaden, Heidelberg, 832 pp.

Pomberger, R., 2008. Entwicklung von Ersatzbrennstoff für das HOTDISC-Verfahren und Analyse der abfallwirtschaftlichen Relevanz: Development of substitute fuel for the HOTDISC technology and analysis of its relevance regarding waste management, Getr. Zählg., Ill., graph. Darst., Kt.

Schönert, K., 2002. Zerkleinerungstechnik für nicht-spröde Abfälle und Schrotte. https://docplayer.org/28040907-Zerkleinerungstechnik-fuer-nicht-sproedeabfaelle-und-schrotte.html. Accessed 2 May 2023.

Shumway, R.H., Stoffer, D.S., 2017. Time series analysis and its applications: With R examples, Fourth edition ed. Springer texts in statistics. Springer, Cham, 562 pp.

Smuts, J.F., 2012. Process control for practioners: How to tune PID controllers and optimize control loops, Second printing, with corrections ed. Opticontrols, League City, Texas, 315 pp.

Umweltbundesamt, 2023. Abfallaufkommen. https://www.umweltbundesamt.de/daten/ressourcen-abfall/abfallaufkommen#deutschlands-abfall. Accessed 28 April 2023.

Weißenbach, T., Graf, J., Pomberger, R., Sarc, R., 2020. Calculation of the additional recycling potential in the European Union by implementing the circular economy package. Journal of Environmental Waste Management and Recycling 3.2020 (2).

6.2 Abbreviations

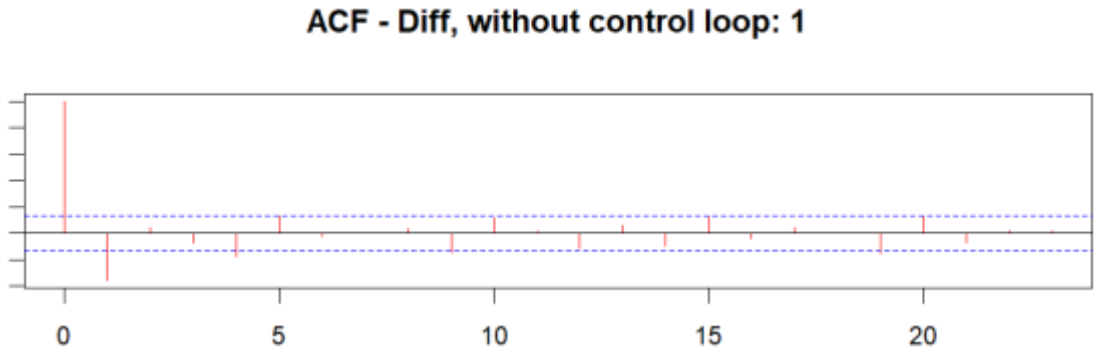
AIC	Akaike´s Information Criterion
AR(p)	Auto-Regressive Model of order p
DMFMS	Digital Material Flow Monitoring System
e.g.	Exempli gratia / for example
Etc.	et cetera
FCE	Final Control Element
i.e.	id est / that is
M	torque
MA(q)	Moving Average Model of order q
NIR	Near-Infrared
P	Power
PACF	Partial Autocorrelation Function
p_i	Degrees of freedom i
PID - Controller	Proportional Integral Derivative - Controller
PV	Process Variable
r_k	Correlation Coefficient for lag k
rpm	Rotations per minute
RSS_i	Residual sum of squares model i
s	seconds

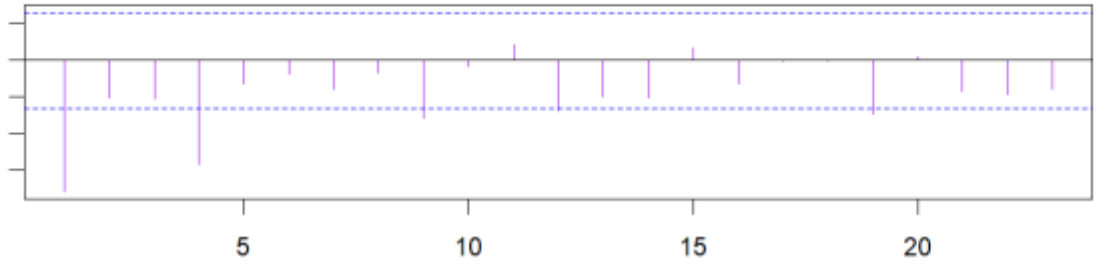
SARIMA	Seasonal Auto Regressive Integrated Moving Average Model
SP	Set Point
SSE	Sum of squared Errors
t_d	Dead Time
w_t	Stochastic variable
x_t	Time Series Data Point at Index t
$\bar{x}$	Average of all Data Points x
θ_t	Moving Average Coefficients
τ	Lag Time
ϕ_k	Partial Correlation Coefficient for variables at lag k
φ_t	Auto-Regressive Coefficients
$\hat{\sigma}_k$	Standard deviation

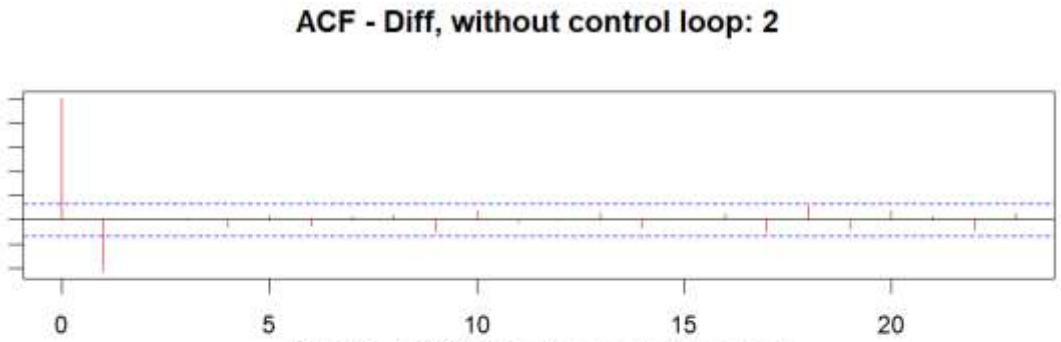
6.3 Tables

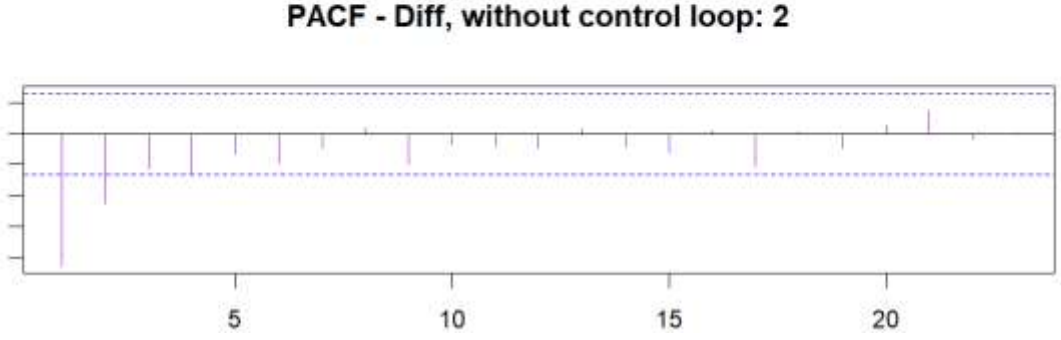
6.4 Figures

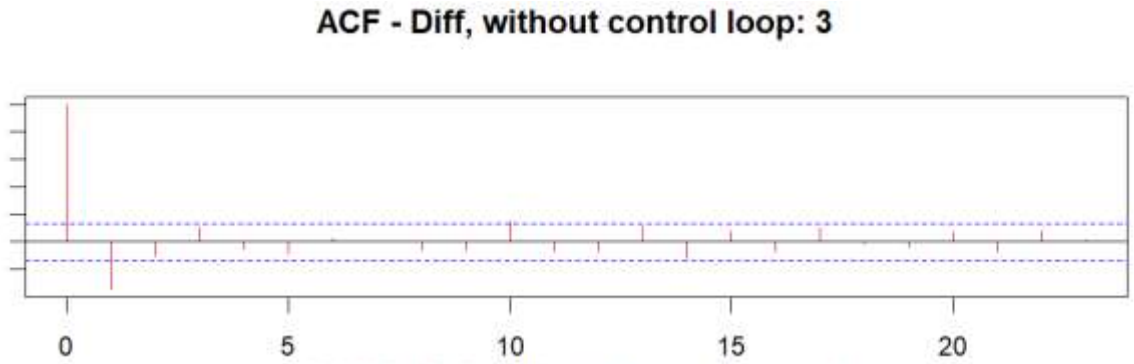
6.5 Appendix

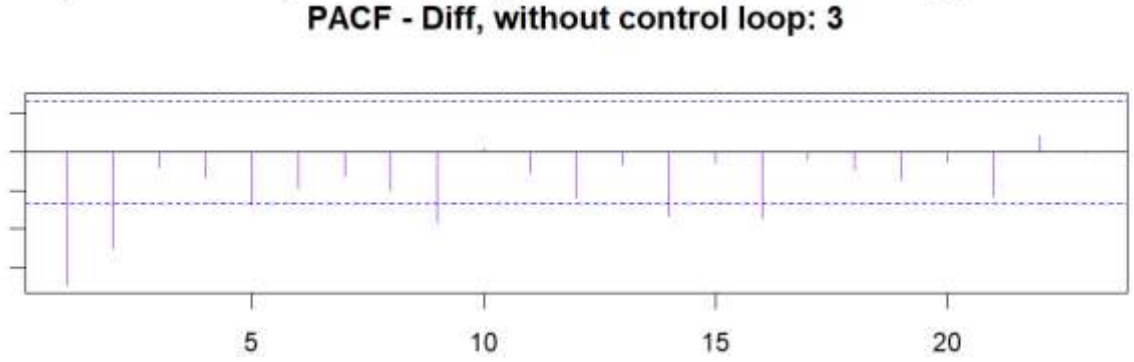

ACF - Diff, without control loop: 1
0
5
10
15
20

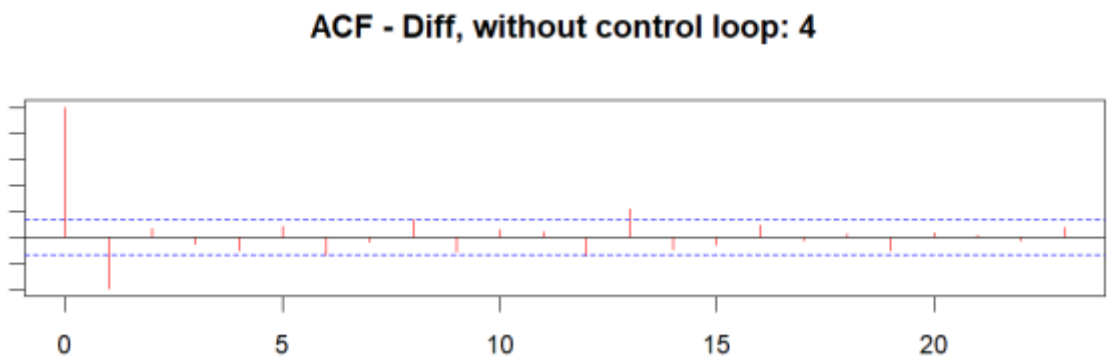

PACF - Diff, without control loop: 1
5
10
15
20

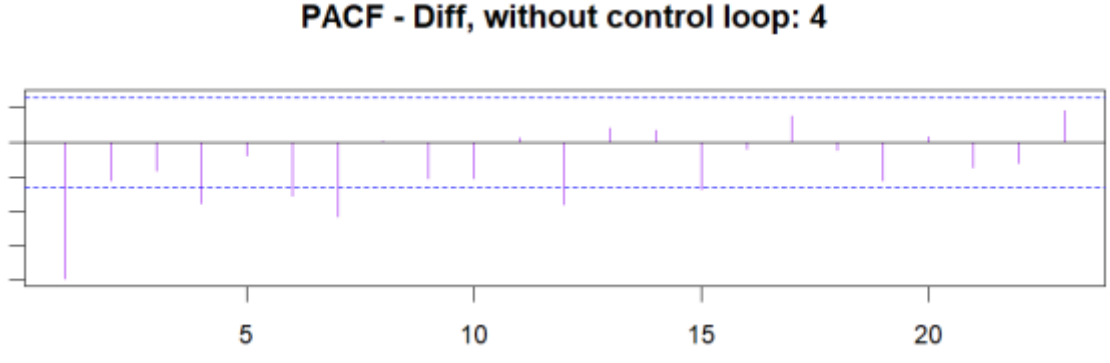

ACF - Diff, without control loop: 2
0
5
10
15
20

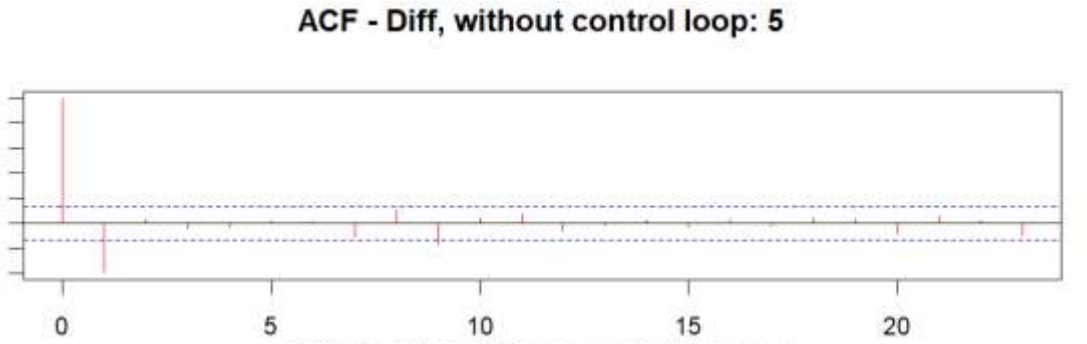

PACF - Diff, without control loop: 2
5
10
15
20

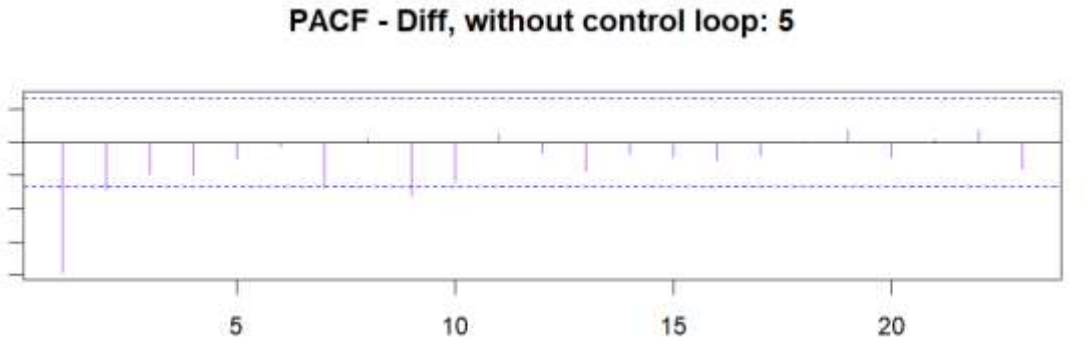

ACF - Diff, without control loop: 3
0
5
10
15
20

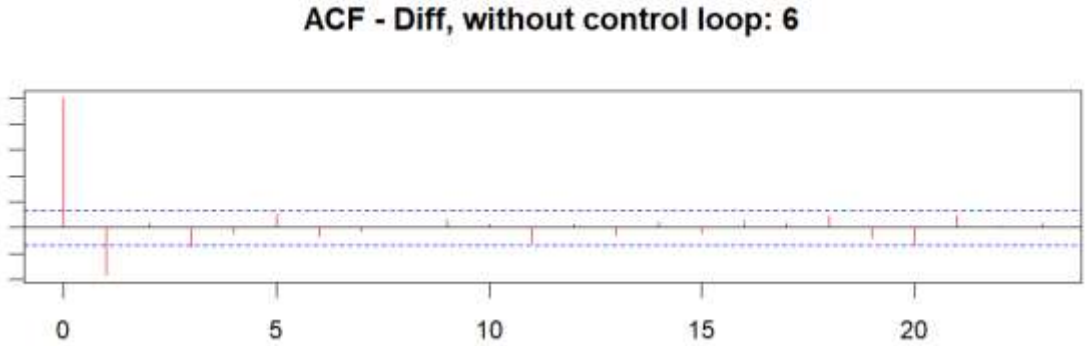

PACF - Diff, without control loop: 3
5
10
15
20

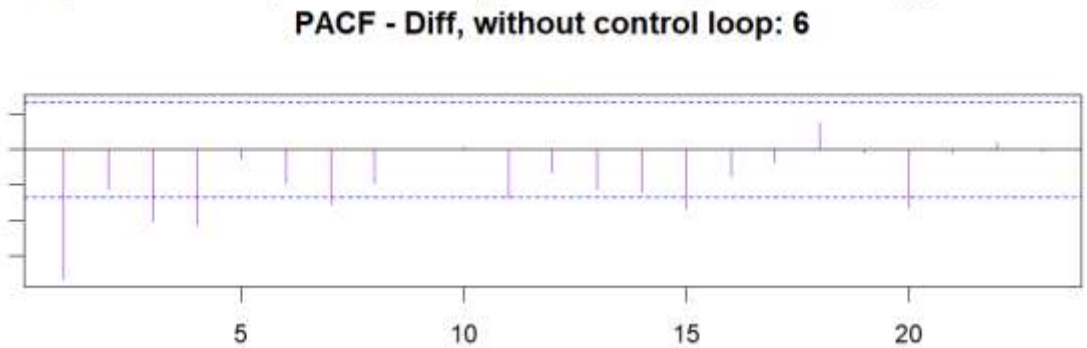

ACF - Diff, without control loop: 4
0
5
10
15
20

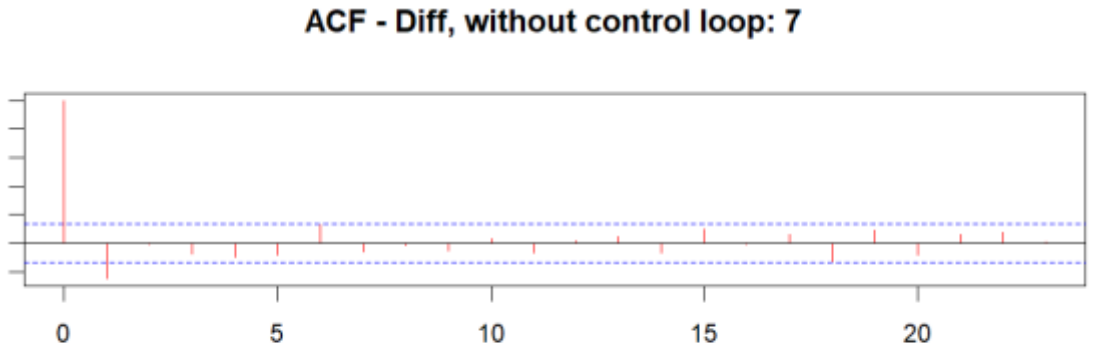

PACF - Diff, without control loop: 4
5
10
15
20

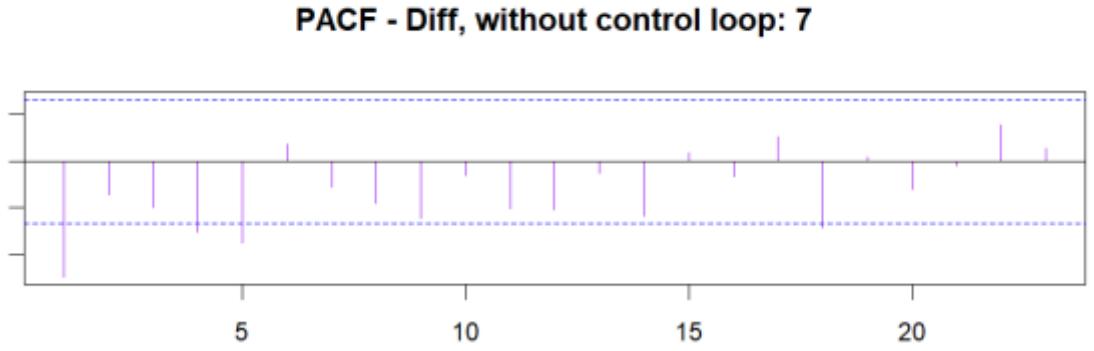

ACF - Diff, without control loop: 5
0
5
10
15
20

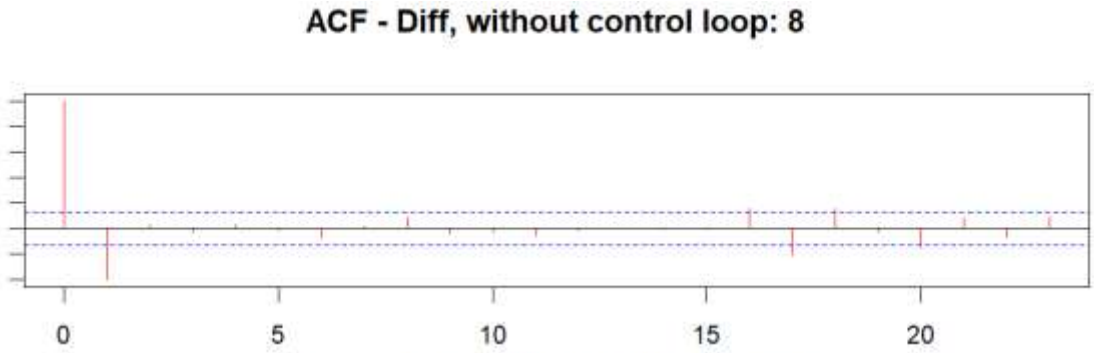

PACF - Diff, without control loop: 5
5
10
15
20

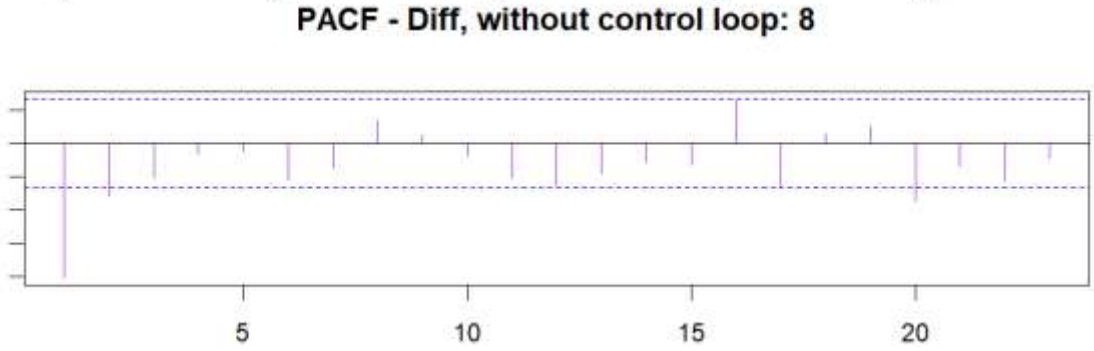

ACF - Diff, without control loop: 6
0
5
10
15
20

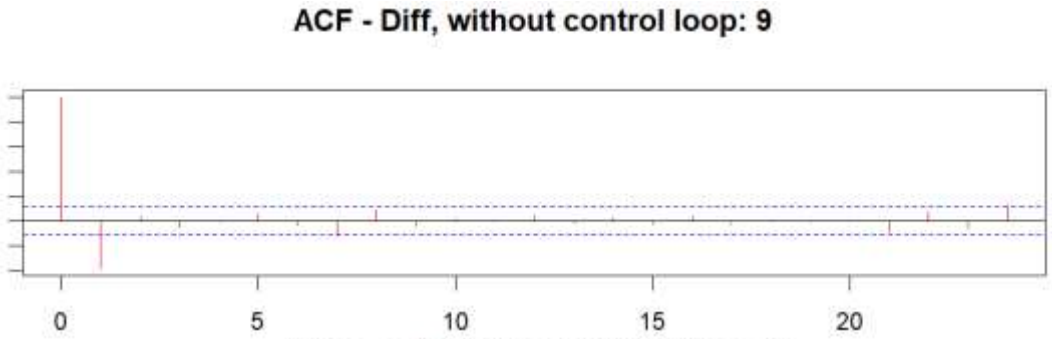

PACF - Diff, without control loop: 6
5
10
15
20

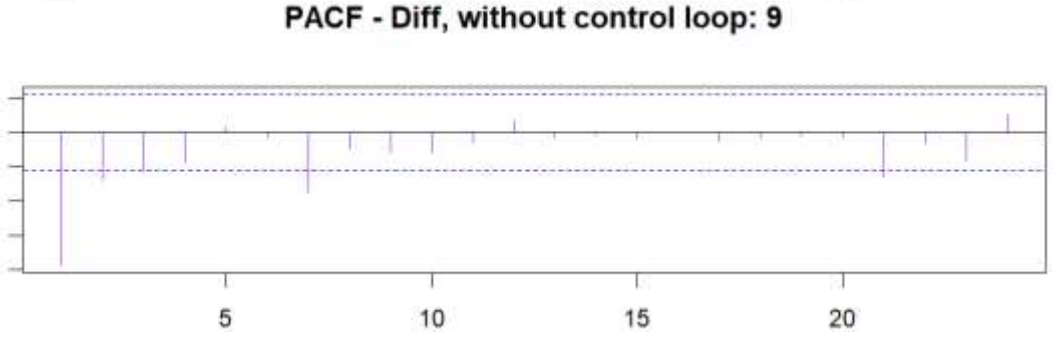

ACF - Diff, without control loop: 7
0
5
10
15
20

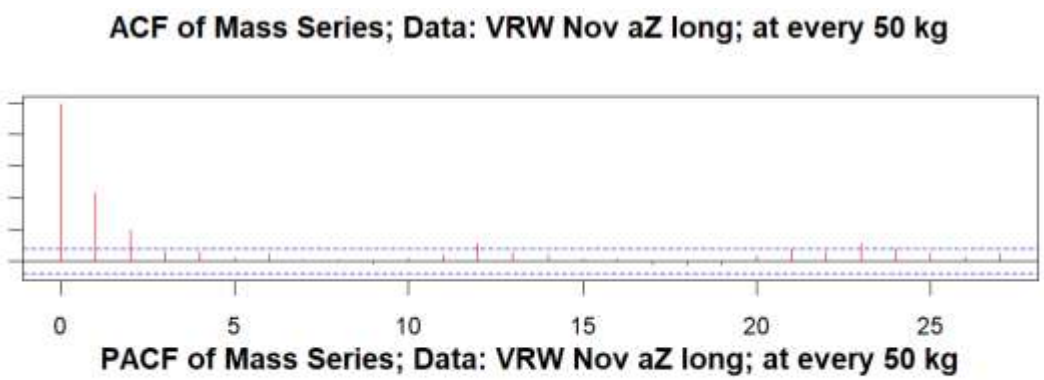

PACF - Diff, without control loop: 7
5
10
15
20

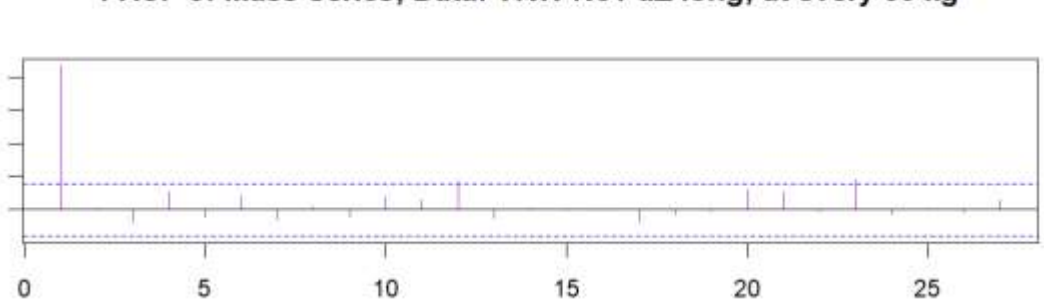

ACF - Diff, without control loop: 8
0
5
10
15
20

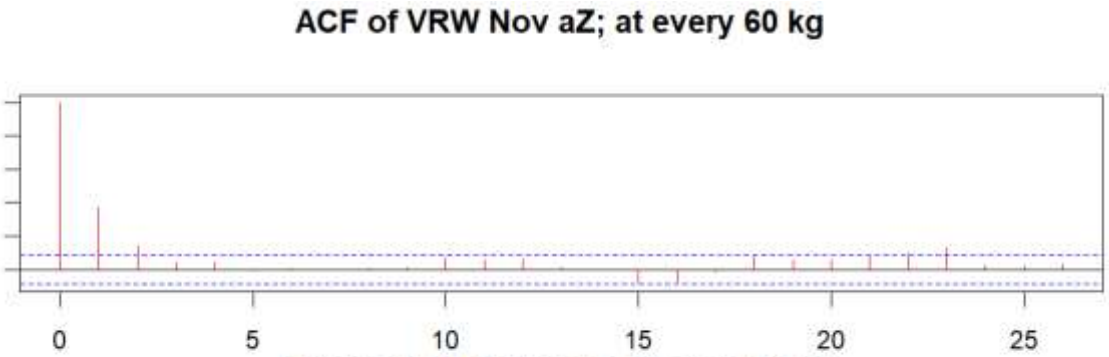

PACF - Diff, without control loop: 8
5
10
15
20

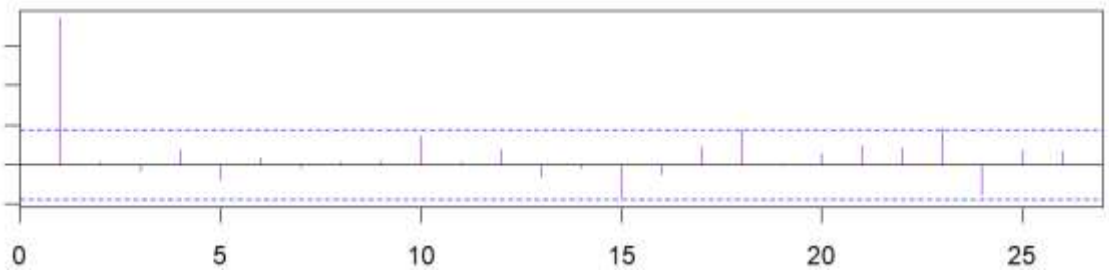

ACF - Diff, without control loop: 9
0
5
10
15
20

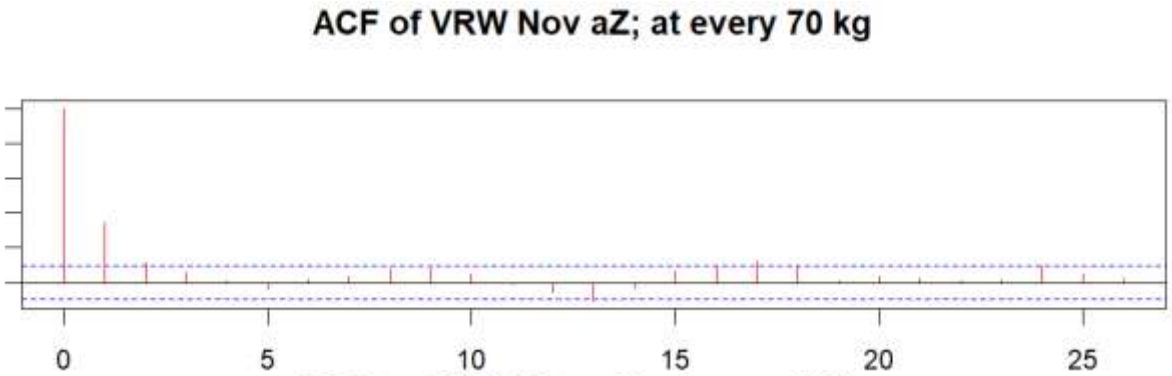

PACF - Diff, without control loop: 9
5
10
15
20

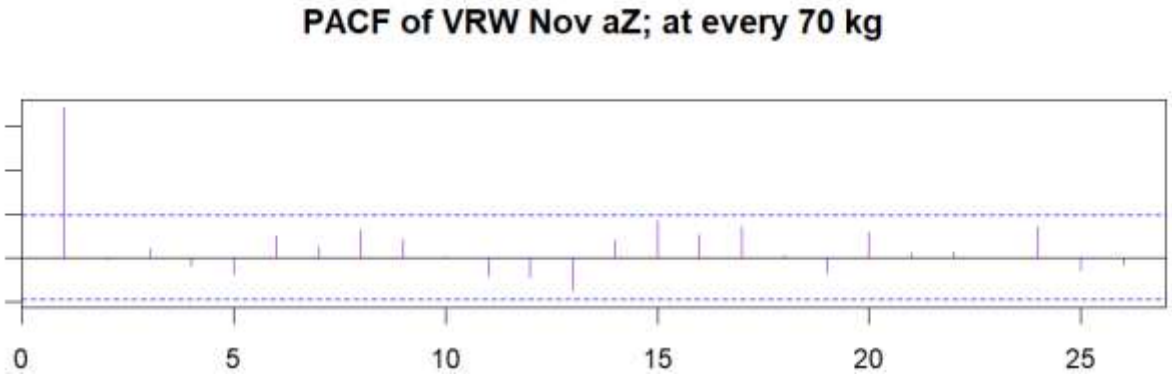

ACF of Mass Series; Data: VRW Nov aZ long; at every 50 kg
0
5
10
15
20
25

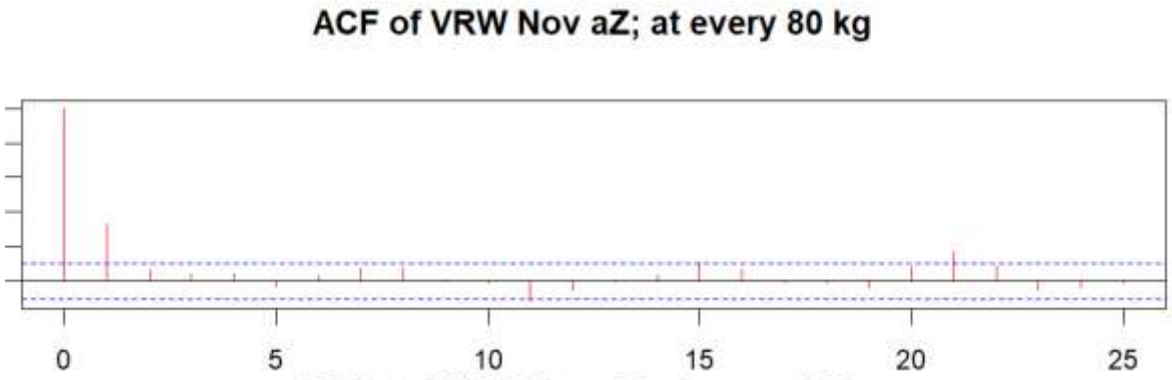

PACF of Mass Series; Data: VRW Nov aZ long; at every 50 kg
0
5
10
15
20
25

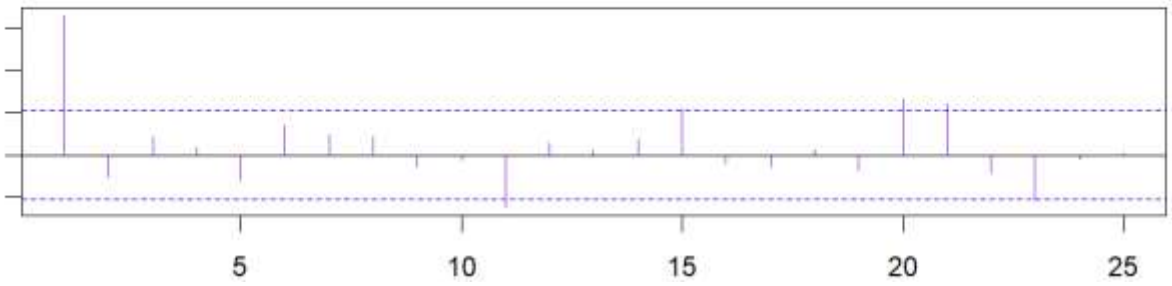

ACF of VRW Nov aZ; at every 60 kg
0
5
10
15
20
25

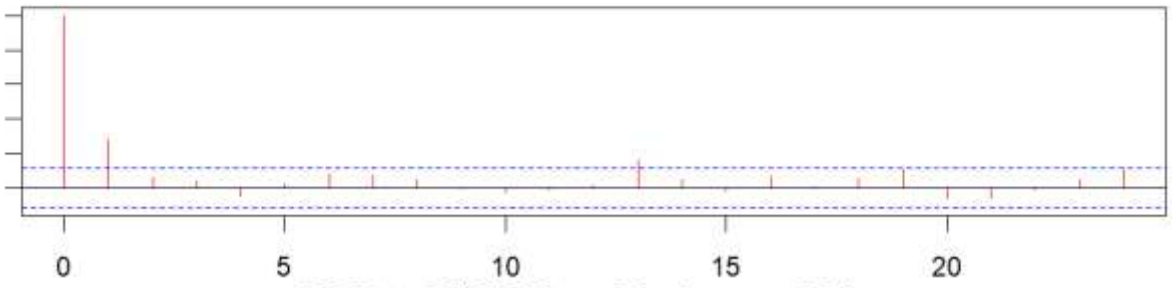

PACF of VRW Nov aZ; at every 60 kg
0
5
10
15
20
25

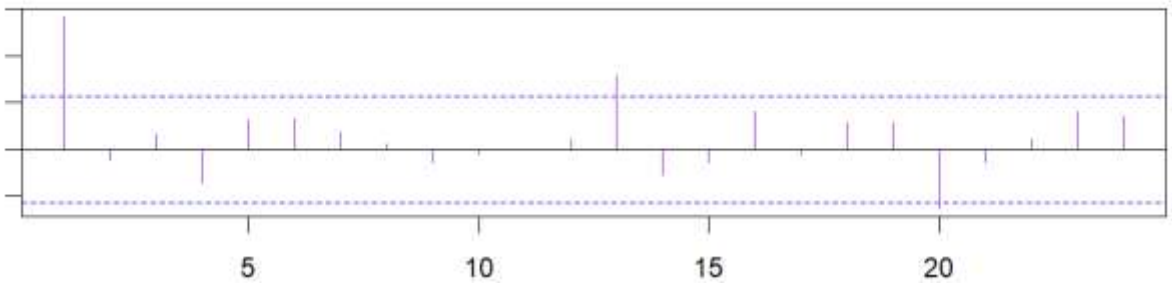

ACF of VRW Nov aZ; at every 70 kg
0
5
10
15
20
25

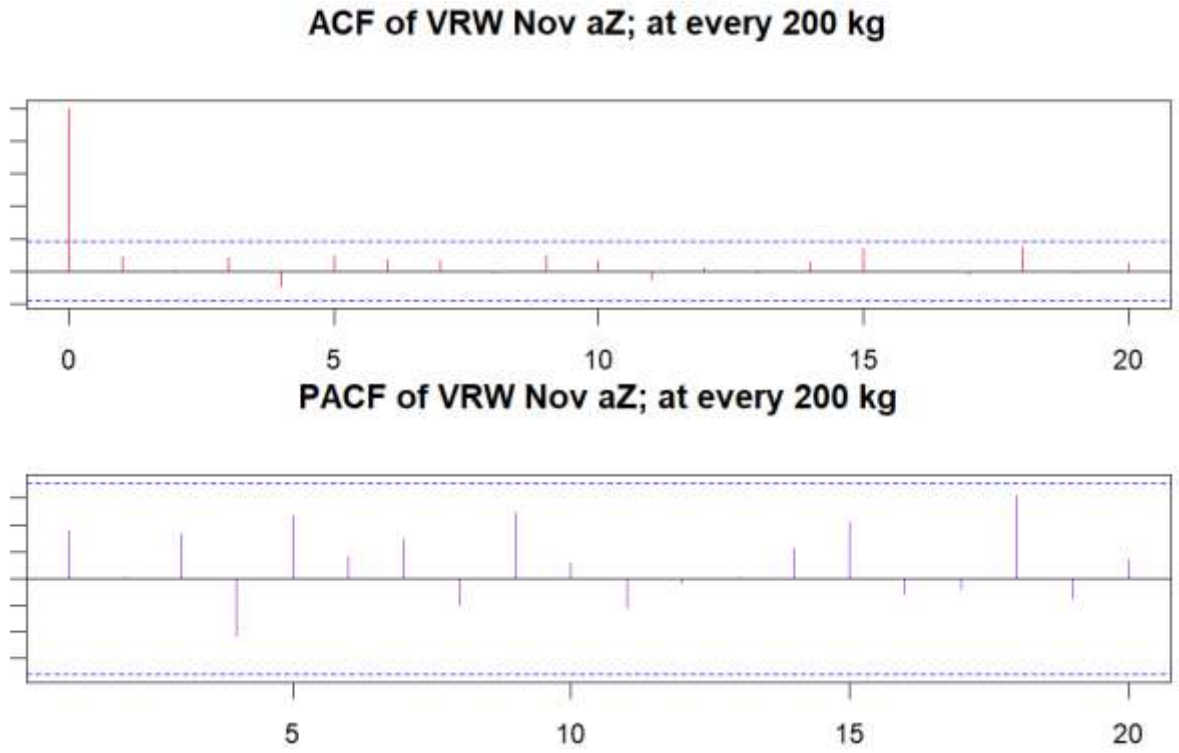

PACF of VRW Nov aZ; at every 70 kg
0
5
10
15
20
25

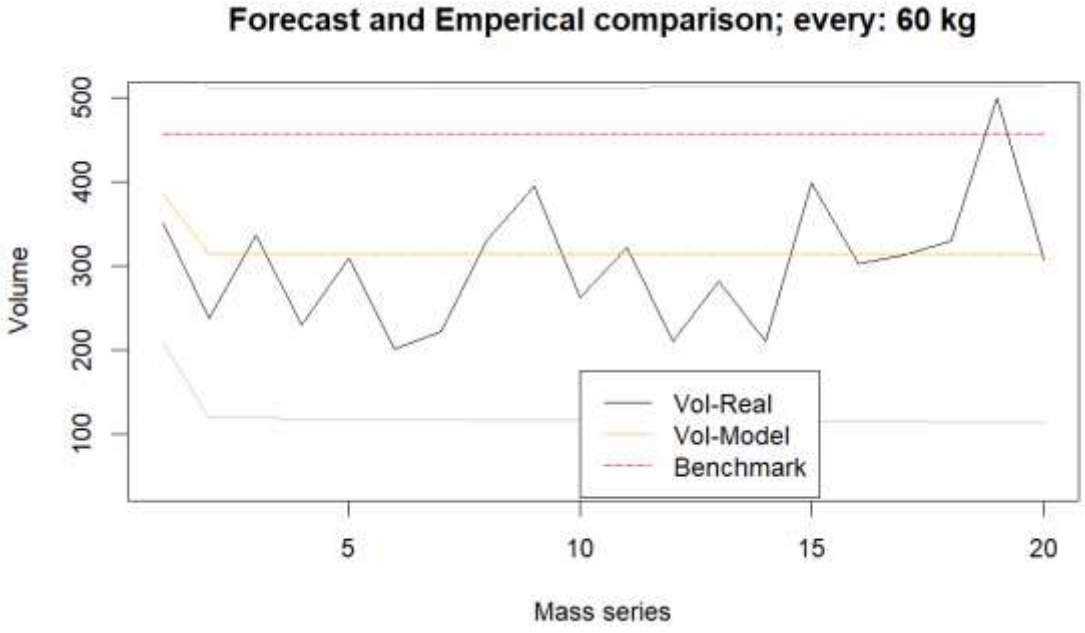

ACF of VRW Nov aZ; at every 80 kg
0
5
10
15
20
25

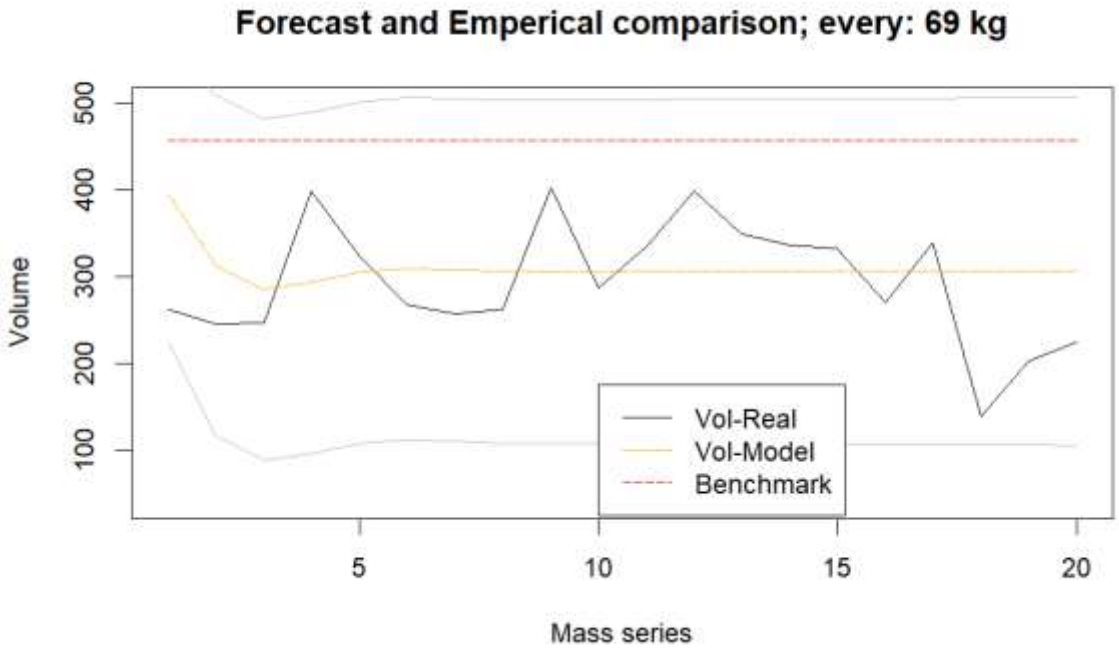

PACF of VRW Nov aZ; at every 80 kg
5
10
15
20
25

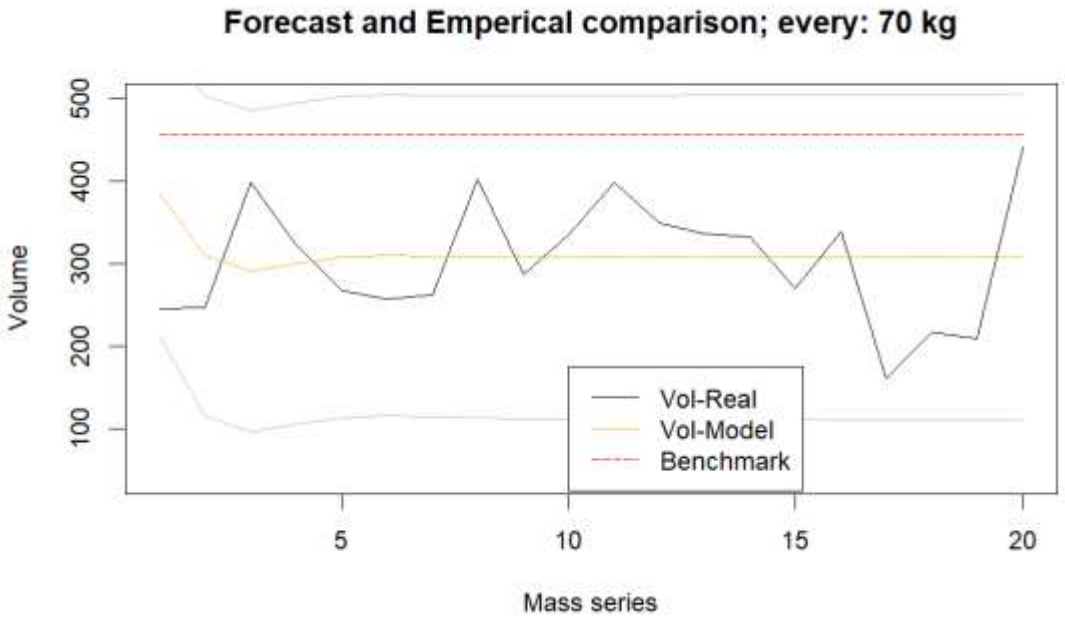

ACF of VRW Nov aZ; at every 90 kg
0
5
10
15
20

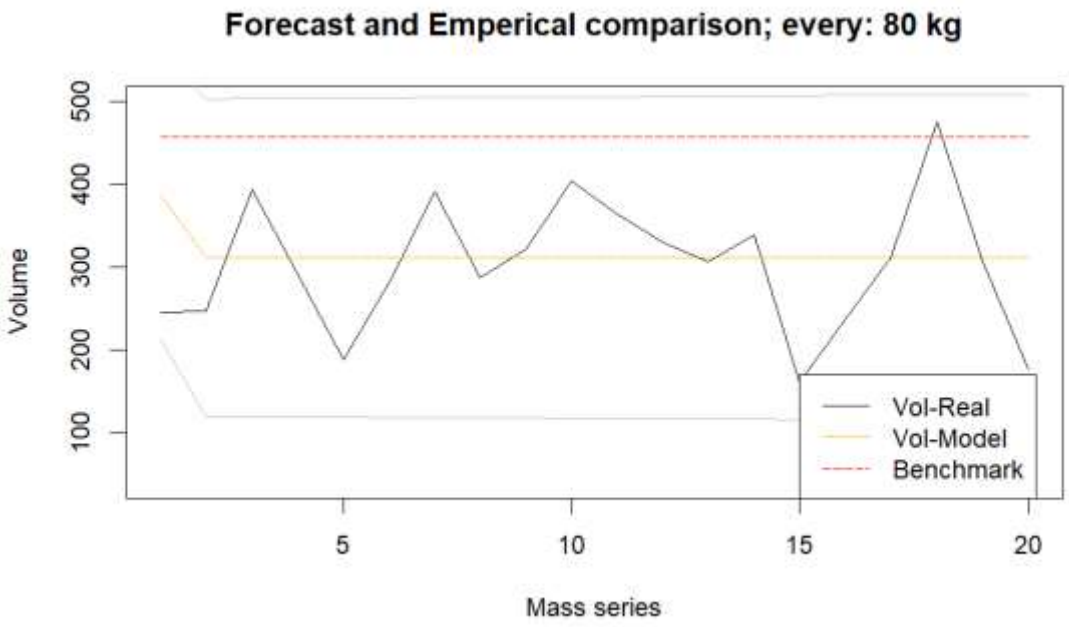

PACF of VRW Nov aZ; at every 90 kg
5
10
15
20

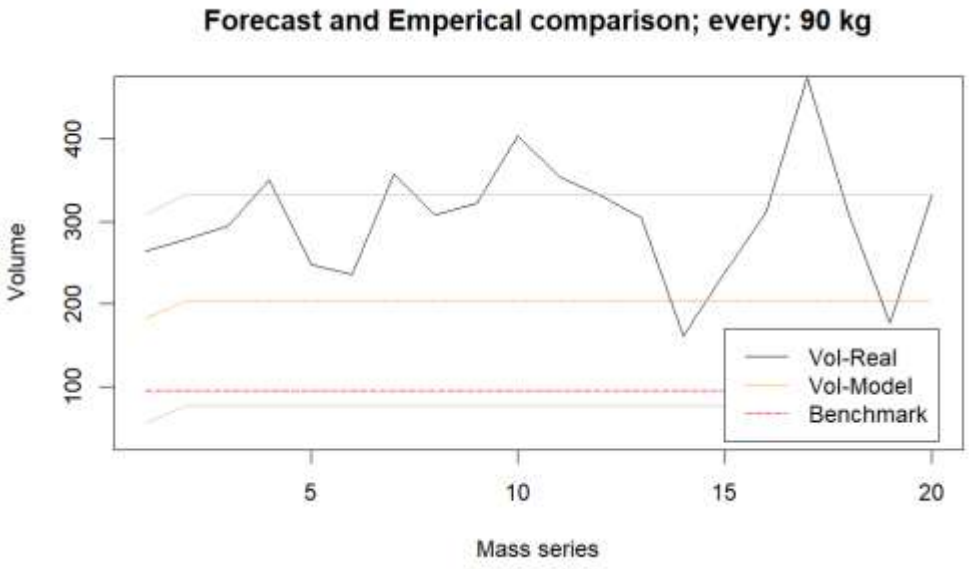

ACF of VRW Nov aZ; at every 200 kg
0
5
10
15
20
PACF of VRW Nov aZ; at every 200 kg
5
10
15
20


Forecast and Emperical comparison; every: 60 kg
Volume
100
200
300
400
500
Vol-Real
Vol-Model
Benchmark
5
10
15
20
Mass series

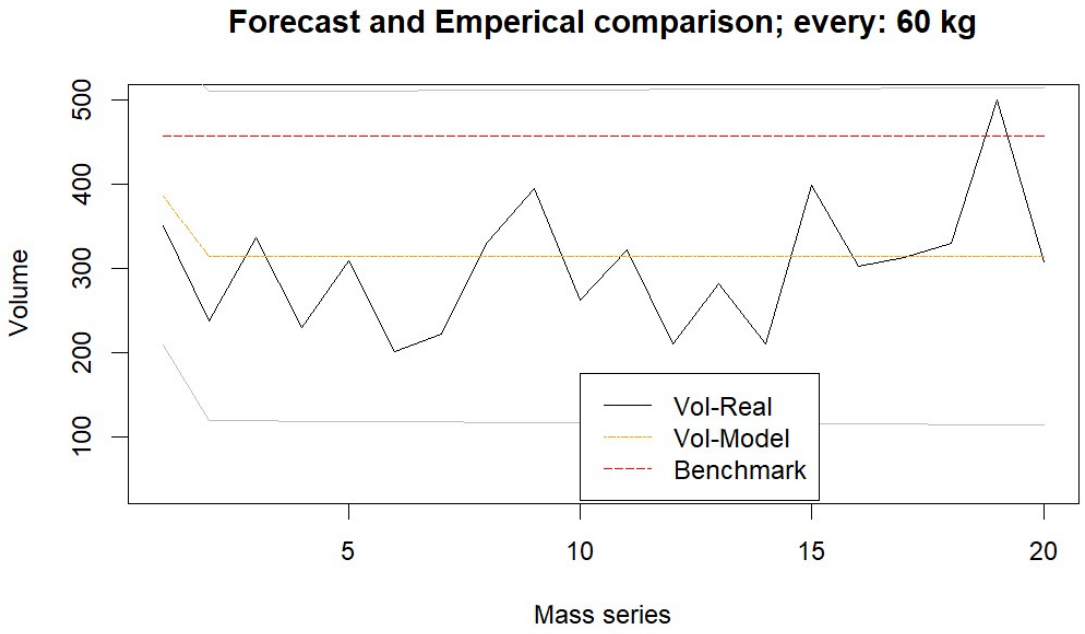

Forecast and Emperical comparison; every: 69 kg
Volume
100
200
300
400
500
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark

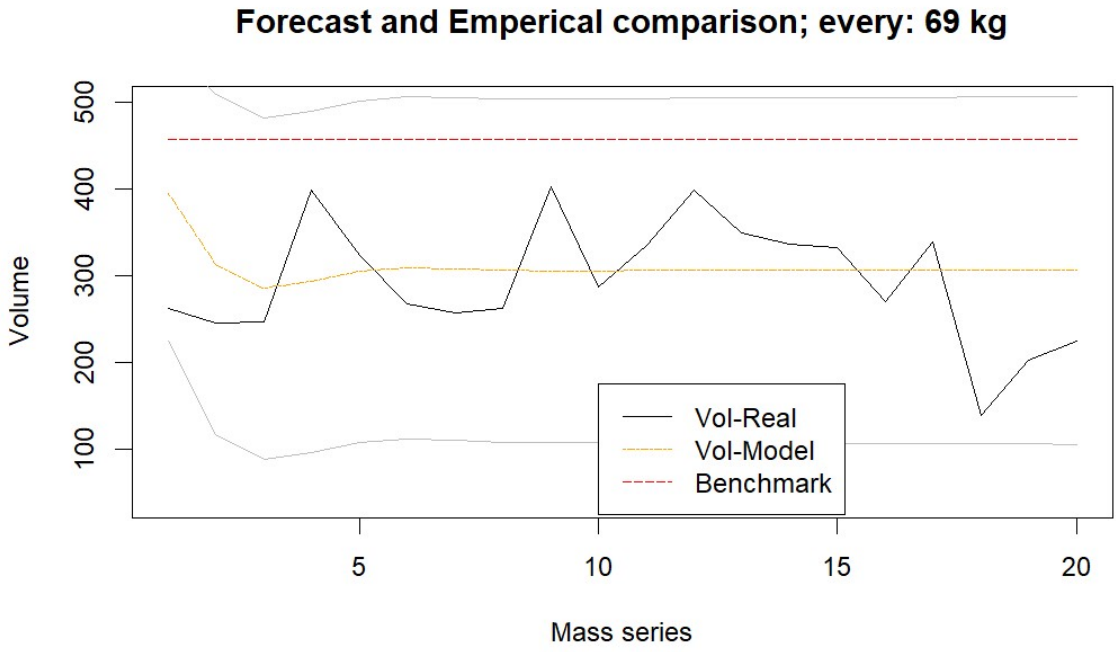

Forecast and Emperical comparison; every: 70 kg
Volume
100
200
300
400
500
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark


Forecast and Emperical comparison; every: 80 kg
Volume
100
200
300
400
500
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark


Forecast and Emperical comparison; every: 90 kg
Volume
100
200
300
400
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark

Appendix


Forecast and Emperical comparison; every: 60 kg
Volume
100
200
300
400
500
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark


Forecast and Emperical comparison; every: 69 kg
Volume
100
200
300
400
500
5
10
15
20
Mass series
Vol-Real
Vol-Model
Benchmark

Zusammenfassung Code

Imhof

2023-09-22

Laden der Daten und definieren von Variablen.Um alle 4s einen neuen Datenpunkt zu erhalten und dadurch die doppelten Datenpunkte zu filtern.

```
#Def von Variablen
dat.num <- 1
nth.ele <- 60 #ergibt ein Abstand von 4s zwischen den Datenpunkten
VMJ_oR <- list()

#Laden der Daten ohne Regelung vom 9.Juni 2022 Versuche Müllex
for(dat.num in 1:9){
  test.data <- read.csv(paste("D:/DATA Jason/Uni/0_Master Unterlagen/0_Masterarbeit/Messdaten/Versuche_
Mullex/9. Messung csv Exports/9_Messung ohne Regelung/9_Messung_ohne_regelung_", dat.num,".csv", sep =
""), sep = ",", header = TRUE, stringsAsFactors = FALSE)

  #Bereinigen der Daten
  test.min <- test.data[seq.int(1, nrow(test.data), by = nth.ele),]
  test.opt <- subset(test.min, test.min[["Volumen.optisch.H.Sensor..m.3.h."]]>10)
  VMJ_oR[[length(VMJ_oR) + 1]] <- test.min[["Volumen.optisch.H.Sensor..m.3.h."]]
}

#Laden der Daten mit Regelung
dat.num <- 1
VMJ_mR <- list()

for(dat.num in 1:9){
  test.data <- read.csv(paste("D:/DATA Jason/Uni/0_Master Unterlagen/0_Masterarbeit/Messdaten/Versuche_
Mullex/9. Messung csv Exports/9_Messung mit Regelung/9_Messung_mit_regelung_", dat.num,".csv", sep =
""), sep = ",", header = TRUE, stringsAsFactors = FALSE)

  #Bereinigen der Daten
  test.min <- test.data[seq.int(1, nrow(test.data), by = nth.ele),]
  test.opt <- subset(test.min, test.min[["Volumen.optisch.H.Sensor..m.3.h."]]>10)
  VMJ_mR[[length(VMJ_mR) + 1]] <- test.min[["Volumen.optisch.H.Sensor..m.3.h."]]
}
```

Jetzt können die eingelesenen Daten auf Kennzahlen wie die 90/10er-Quantile untersucht und die Boxplots erstellt werden.

Kennzahlen

Hier eine Tabelle der Kennzahlen:

```
#Kennzahlen ermitteln und in Dataframe speichern
VMJ_g <- data.frame("Index" = c("10thPercentile", "90thPercentile","Quotient_90/10", "Median", "Avarag
e", "Standarddeveation", "Diff Avarage-Median"))

#for-Schleife für VMJ_oR
for(i in 1:9){
  ovol_p90=quantile(VMJ_oR[[i]],probs=0.9)
  ovol_p10=quantile(VMJ_oR[[i]],probs=0.1)
  oQvol_1090=ovol_p90/ovol_p10

  Median.Vol = median(VMJ_oR[[i]])
  Avarage.Vol = mean(VMJ_oR[[i]])
  Standard.dev = sd(VMJ_oR[[i]])
  AR.Process = VMJ_mR[[i]] - mean(VMJ_oR[[i]])

  VMJ_g[paste("Mes_o_R", i)] <- c(ovol_p10, ovol_p90, oQvol_1090, Median.Vol, Avarage.Vol, Standard.de
v, Avarage.Vol - Median.Vol)
}

#for-Schleife für VMJ_mR
for(i in 1:9){
  ovol_p90=quantile(VMJ_mR[[i]],probs=0.9)
  ovol_p10=quantile(VMJ_mR[[i]],probs=0.1)
  oQvol_1090=ovol_p90/ovol_p10

  Median.Vol = median(VMJ_mR[[i]])
  Avarage.Vol = mean(VMJ_mR[[i]])
  Standard.dev = sd(VMJ_mR[[i]])
  AR.Process = VMJ_mR[[i]] - mean(VMJ_mR[[i]])

  VMJ_g[paste("Mes_m_R", i)] <- c(ovol_p10, ovol_p90, oQvol_1090, Median.Vol, Avarage.Vol, Standard.de
v, Avarage.Vol - Median.Vol)
}

format(VMJ_g)
```

Sowie ein Vergleich der Versuchsreihen als Boxplots:

```
#par(mfrow = c(2, 1), mar = c(2, 1, 2, 1))

boxplot(VMJ_oR, main = "Boxplot of 1-9 without control loop", col = "green", ylab = "Volume")
```

```
#par(mfrow = c(2, 1), mar = c(2, 1, 2, 1))
boxplot(VMJ_mR, main = "Boxplot of 1-9 with control loop", col = "yellow", ylab = "Volume")
```

```
df_oRmR <- data.frame(values = c(VMJ_oR[[1]], VMJ_mR[[1]],
                                 VMJ_oR[[2]], VMJ_mR[[2]],
                                 VMJ_oR[[3]], VMJ_mR[[3]],
                                 VMJ_oR[[4]], VMJ_mR[[4]],
                                 VMJ_oR[[5]], VMJ_mR[[5]],
                                 VMJ_oR[[6]], VMJ_mR[[6]],
                                 VMJ_oR[[7]], VMJ_mR[[7]],
                                 VMJ_oR[[8]], VMJ_mR[[8]],
                                 VMJ_oR[[9]], VMJ_mR[[9]]
                                 ),
                      group = c(rep("1o", length(VMJ_oR[[1]])),
                                rep("1m", length(VMJ_mR[[1]])),
                                rep("2o", length(VMJ_oR[[2]])),
                                rep("2m", length(VMJ_mR[[2]])),
                                rep("3o", length(VMJ_oR[[3]])),
                                rep("3m", length(VMJ_mR[[3]])),
                                rep("4o", length(VMJ_oR[[4]])),
                                rep("4m", length(VMJ_mR[[4]])),
                                rep("5o", length(VMJ_oR[[5]])),
                                rep("5m", length(VMJ_mR[[5]])),
                                rep("6o", length(VMJ_oR[[6]])),
                                rep("6m", length(VMJ_mR[[6]])),
                                rep("7o", length(VMJ_oR[[7]])),
                                rep("7m", length(VMJ_mR[[7]])),
                                rep("8o", length(VMJ_oR[[8]])),
                                rep("8m", length(VMJ_mR[[8]])),
                                rep("9o", length(VMJ_oR[[9]])),
                                rep("9m", length(VMJ_mR[[9]]))
                                )
                      )
boxplot(values ~ group, df_oRmR, col = c("orange", "white"),
        main = "Control loop experiment results",
        xlab = "",
        ylab = "Volume")
```

Augenscheinlich gibt es keinen Unterschied und die Mittelwerte zwischen den Versuchsreihen mit und ohne Regelung. Das soll jetzt Nummerisch bestägt werden.

Auswertung der Kennzahlen

Zuerst ist eine untersuchung notwending ob die Quotienten der Messreihen die Bedingungen für den t-Test erfüllen, nämlich müssen die Daten normalverteilt und gleiche Varianzen darstellen.

```
#Erstellen der notwendigen Vektoren 10/90 Quantiele(Q) und Median (M), jeweils ohne und mit Regelung
VMJ_Q_oR <- unlist(VMJ_g[3, c(2:10)])
VMJ_Q_mR <- unlist(VMJ_g[3, c(11:19)])

VMJ_M_oR <- unlist(VMJ_g[4, c(2:10)])
VMJ_M_mR <- unlist(VMJ_g[4, c(11:19)])
```

Jetzt kann für die 10/90 Quantile die Normalverteilung geprüft werden.

```
swt_VMJ_Q_oR <- shapiro.test(VMJ_Q_oR)
qqnorm(VMJ_Q_oR, main = paste("QQ Quantile 90/10 no control loop, Shapiro p=", round(swt_VMJ_Q_oR$p.val
ue, 4)))
qqline(VMJ_Q_oR)

swt_VMJ_Q_mR <- shapiro.test(VMJ_Q_mR)
qqnorm(VMJ_Q_mR, main = paste("QQ Quantile 90/10 with control loop, Shapiro p=", round(swt_VMJ_Q_mR$p.v
alue, 4)))
qqline(VMJ_Q_mR)
```

Die Quantile sind nicht gut was eine Normalverteilung angeht, dafür sind die p-Werte vom Shapiro-Wilk test deutlich über dem Signifikanz Niveau von 0.05 und damit kann die Nullhypothese(Datensatz ist normalverteilt) NICHT abgelehnt werden. Somit kann an dieser Stelle von einer angenäherten Normalverteilung ausgegangen werden und es wird mit dem t-Test fortgefahren. Mit dem t-Test wird in R gleich der Mittelwert sowie die 95%-Konfidenz Interwallgrenzen bekanntgegeben.

```
library(ggplot2)

VMJ_Qtt <- t.test(VMJ_Q_oR, VMJ_Q_mR)
VMJ_Qtt_oR <- t.test(VMJ_Q_oR)
VMJ_Qtt_mR <- t.test(VMJ_Q_mR)

VMJ_Q_ci <- data.frame(
  group = c("1-without CL", "2-With CL", "3-Difference"),
  mean = c(VMJ_Qtt_oR$estimate[1], VMJ_Qtt_mR$estimate[1], VMJ_Qtt$estimate[1]-VMJ_Qtt$estimate[2]),
  lower_ci = c(VMJ_Qtt_oR$conf.int[1], VMJ_Qtt_mR$conf.int[1], VMJ_Qtt$conf.int[1]),
  upper_ci = c(VMJ_Qtt_oR$conf.int[2], VMJ_Qtt_mR$conf.int[2], VMJ_Qtt$conf.int[2])
)

ggplot(VMJ_Q_ci, aes(x = group, y = mean)) +
  geom_bar(stat = "identity", width = 0.8, fill = "blue", alpha = 0.4) +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.5, color = "black", alpha = 0.8) +
  labs(title = "Mean of 90/10 Quantiles for control loop(CL) experiments with Confidence Intervals", x
= "Group", y = "Mean") +
  theme_minimal()

print(VMJ_Qtt_oR)
print(VMJ_Qtt_mR)
print(VMJ_Qtt)
print(sd(VMJ_Q_oR))
print(sd(VMJ_Q_mR))
```

Wie im Boxplot zu sehen ist, gibt es keinen signifikanten Unterschied zwischen den Verteilungen der zwei Fahrweisen des Schredders.

ACF und PACF

Zuerst eine einfache Darstellung der Zeitreihe und gleich eine Darstellung der Differenzen der Zeitreihe. Die Differenz entfernt jeglichen Trend aus der Reihe um in weiterer Folge den Gleitenden Mittelwert und die Autoregression abzuschätzen.

```
m <- 15
#for(j in 1:m) {
 # geometric_series[j] <- j*2/(m*(m+1))}
LD_series <- seq(1,0,by = -1/(m-1))
LD_series <- LD_series/sum(LD_series)
geometric_series <- 0.7^(1:m)
# Normalize the vector to sum up to one rollmean(VMJ_oR[[j]], k = 15, align = "right", fill = NA)
geometric_series <- geometric_series / sum(geometric_series)
par(mfrow = c(2,1), mar = c(1, 4, 3, 1))
for(j in 1:9){
plot.ts(VMJ_oR[[j]],
        col = "darkgreen",
        main = paste("Time Series VMJ_oR ", j, sep = ""),
        ylab = "Volume [m^3/h]")
lines(filter(VMJ_oR[[j]], LD_series, sides = 1, method = "convolution"), col = "red", lty = 2) #15 data
points for 1min rolling avarage at 4s/Datapoint
#}
#par(mfrow = c(2,1), mar = c(1, 4, 3, 1))
#for (j in 1:9){
plot.ts(diff(VMJ_oR[[j]]),
        main = paste("Difference d=1: VMJ_oR ", j, sep = ""),
        ylab = "Volumedifference")
}
```

```
for(j in 1:9){
par(mfrow = c(2,1), mar = c(1, 4, 3, 1))
plot.ts(VMJ_mR[[j]],
        col = "blue",
        main = paste("Time Series VMJ_mR ", j, sep = ""),
        ylab = "Volume [m^3/h]")
lines(filter(VMJ_mR[[j]], LD_series, sides = 1, method = "convolution"), col = "orange")
plot.ts(diff(VMJ_mR[[j]]),
        main = paste("Differenz d=1: VMJ_mR ", j, sep = ""),
        ylab = "Volume [m^3/h]")

plot.ts(diff(log(VMJ_mR[[j]])),
        main = paste("Differenz d=1: log V_mR ", j, sep = ""),
        ylab = "Volume [m^3/h]")
}
```

Als nächstes wird der Box-Pierce test durchgeführt um die Nullhypothese zu prüfen ob eine Korrelation zwischen den Datenpunkten besteht. Für ausreichend geringe p-Werte wird H_0 : (Die Daten sind unabhängig) abgelehnt.

```
cat(" VMJ_oR1 p-Value:", Box.test(diff(VMJ_oR[[1]]), lag = log(length(VMJ_oR[[1]])))$p.value,
    " VMJ_oR2 p-Value:", Box.test(diff(VMJ_oR[[2]]), lag = log(length(VMJ_oR[[2]])))$p.value,
    " VMJ_oR7 p-Value:", Box.test(VMJ_oR[[7]], lag = log(length(VMJ_oR[[7]])))$p.value,
sep = " ")

cat(" VMJ_mR1 p-Value:", Box.test(VMJ_mR[[1]], lag = log(length(VMJ_mR[[1]])))$p.value,
    " VMJ_mR2 p-Value:", Box.test(VMJ_mR[[2]], lag = log(length(VMJ_mR[[2]])))$p.value,
    " VMJ_mR7 p-Value:", Box.test(VMJ_mR[[7]], lag = log(length(VMJ_mR[[7]])))$p.value,
sep = " ")
#VMJ_g[paste("Mes_m_R", i)]
pvaluesVMJ_oR <- data.frame("Index" = c("p-Value"))

for (i in 1:9){
  pvaluesVMJ_oR[paste("VMJ_oR", i)] <- round(Box.test(diff(VMJ_oR[[i]]), lag = log(length(VMJ_oR
[[i]])))$p.value, 4)
}
#write.csv(pvaluesVMJ_oR, file = "pvalues.csv", row.names = FALSE)
```

Nachdem diese p-Werte weit unter dem Konfidenzwert liegen, wird bei diesen Daten von einer Korrelation ausgegangen. Für die Anwendung auf den Shredder ist jetzt allerdings noch wichtig den Grad der Korrelation zu bestimmen und nach Möglichkeit ein Model zu erstellen das entsprechend Werte besser als ein rein stochastisches Modell hervorsagen kann.

Jetzt wird die ACF und PACF der Daten bestimmt um das Modell besser abschätzen zu können.

```
#estimate Ma and AR order of d=1 data
for(j in 1:9){
par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
acf(diff(VMJ_oR[[j]]), main = paste("ACF - Diff, without control loop: ", j, sep = ""), col = "red")
pacf(diff(VMJ_oR[[j]]), main = paste("PACF - Diff, without control loop: ", j, sep = ""), col = "purpl
e")
}
for(j in 1:9){
  par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
  acf(diff(VMJ_mR[[j]]), main = paste("ACF - Diff, with control loop: ", j, sep = ""), col = "red")
  pacf(diff(VMJ_mR[[j]]), main = paste("PACF - Diff, with control loop: ", j, sep = ""), col = "purpl
e")
}
```

Beim ACF können wir MA(q) für q=4 erkennen und aus dem PACF können wir AR(p) für p=4 und p=8 für alle Datensätze sehen. Eine Sesonalekomponente lässt sich für den PACF nahelegen da immer wieder signifikante Peaks auftauchen mit abständen zwischen ihnen. Hier lässt sich die Periode aber nur schwer visuel abschätzen. Daher kann für die AR(P) mit ein paar Werten probiert werden

```
d=1
Compare_models_mR <- data.frame(p = c(0), d = c(0), q = c(0), AIC = c(0), SSE = c(0), pVa = c(0) )
Compare_models_oR <- data.frame(Regelung = c(0), p = c(0), d = c(0), q = c(0), AIC = c(0), SSE = c(0),
pVa = c(0) )

for(p in 1:9){
  for(q in 1:5){
    if(p+d+q<=7){
      model<-arima(x=VMJ_mR[[1]], order = c((p-1),d,(q-1)))
      pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
      sse<-sum(model$residuals^2)
      Compare_models_mR[nrow(Compare_models_mR) + 1, ] = c(p-1, d, q-1, model$aic, sse, pval$p.value)
      #cat(p-1,d,q-1, "mR", 'AIC=', model$aic, ' SSE=',sse,' p-VALUE=', pval$p.value,'\n')
    }
  }
}
d=1
for(p in 1:9){
  for(q in 1:5){
    if(p+d+q<=8){
      model<-arima(x=VMJ_oR[[1]], order = c((p-1),d,(q-1)))
      pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
      sse<-sum(model$residuals^2)
      Compare_models_oR[nrow(Compare_models_oR) + 1, ] = list("ohne", p-1, d, q-1, model$aic, sse, pval
$p.value)
      #cat(p-1,d,q-1, "oR", 'AIC=', model$aic, ' SSE=',sse,' p-VALUE=', pval$p.value,'\n')
    }
  }
}
Compare_models_mR[order(Compare_models_mR$AIC, decreasing = FALSE), ]
Compare_models_oR[order(Compare_models_oR$AIC, decreasing = FALSE), ]
```

Das Modell (1,1,3) hat zwar nur den zweit niedrigsten AIC und SSE-Wert, aber der p-Wert ist deutich niedriger als beim (4,1,0). Das beste Modell scheint doch das (0,1,4) zu sein das hier die AIC und SSE-Werte am niedrigsten sind und der p-Wert ist noch hoch genug um die Nullhypothese zu akzeptieren.

Neu: Modell (4, 1, 1)

```
d=NULL
DD=NULL
d=1
DD=1
Compare_S_models_oR <- data.frame(Regelung = c(0), p = c(0), d = c(0), q = c(0), P = c(0), D = c(0), Q
= c(0), AIC = c(0), SSE = c(0), pVa = c(0) )
per=4

for(p in 1:2){
  for(q in 1:2){
    for(i in 1:3){
      for(j in 1:4){
        if(p+d+q+i+DD+j<=10){
          model<-arima(x=VMJ_oR[[9]], order = c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-
1)), period=per))
          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
          sse<-sum(model$residuals^2)
          Compare_S_models_oR[nrow(Compare_S_models_oR) + 1, ] = list("ohne", p-1, d, q-1, i-1, DD, j-
1, model$aic, sse, round(pval$p.value, 4))
          #cat(p-1,d,q-1,i-1,DD,j-1,per, 'AIC=', model$aic, ' SSE=',sse,' p-VALUE=', pval$p.value,'\n')
        }
      }
    }
  }
}

Compare_S_models_oR[order(Compare_S_models_oR$AIC, decreasing = FALSE), ]
```

Mit einer Sesonalen Periode von 4

```
library(astsa)
```

```
## Warning: Paket 'astsa' wurde unter R Version 4.2.2 erstellt
```

```
sarima(VMJ_oR[[1]], 4,1,1,0,0,0)
sarima(log(VMJ_oR[[1]]), 4,1,0,0,0,0)
```

Using this Model we can forcast the next lags.

```
library(astsa)
library(forecast)

model1<- arima(x=VMJ_oR[[1]], order = c(4,1,1), seasonal = list(order=c(0,0,0), period=0))

n = 20
plot(forecast(model1, h = n))
forecast(model1, h = n )
```

Vorhersagemodelle

Imhof

2023-09-22

Time and Mass Series

Das Ziel hier ist verschiedene Schredderoutputdaten mit der angestrebten Modellierung zu prüfen. Das bedeutet es soll die Metodik zur erstellung des Vorhersagemodells mit unterschiedlichen Daten überprüft werden. Dazu soll zu Beginn der Datensatz um 5 min gekürzt werden damit am Schluss ein Vergleich zwischen den epirischen Daten und den Vorhergesagten Daten des Zeitreihen modells möglich wird. Außerdem sollen die Daten umgeschrieben werden auf eine Zeitreihe die pro 1t an ausgegegebenem Material einen Datenpunkt zum aktuellen Volumenstrom (m^3/h)

```
#Def von Variablen
dat.num <- 1
nth.ele <- 60 #ergibt ein Abstand von 4s zwischen den Datenpunkten
VMJ_oR <- list()
VRW_Dat_k <- list()

#Laden der Daten ohne Regelung vom 9.Juni 2022 Versuche Müllex
#for(dat.num in 1:9){
  test.data <- read.csv(paste("D:/DATA Jason/Uni/0_Master Unterlagen/0_Masterarbeit/Messdaten/Versuche_
Mullex/9. Messung csv Exports/9_Messung ohne Regelung/9_Messung_ohne_regelung_", dat.num,".csv", sep =
""), sep = ",", header = TRUE, stringsAsFactors = FALSE)

  #Bereinigen der Daten
  test.min <- test.data[seq.int(1, nrow(test.data), by = nth.ele),]
  test.opt <- subset(test.min, test.min[["Volumen.optisch.H.Sensor..m.3.h."]]>10)
  #Um letzte 5 min = 300s zu kürzen bei 4s pro Datenpunkt -> 300/4 = 75 Datenpunkte am Ende entfernen
  VMJ_Dat_k.1 <- if(nrow(test.min) > 200) test.min[seq.int(1, nrow(test.min)-75, by = 1),] #VMJ_oR_gekü
rzt
#}
test.data.2 <- read.csv("D:/DATA Jason/Uni/0_Master Unterlagen/0_Masterarbeit/Messdaten/Vergleichdaten/
Daten Shredderversuche/Daten Shredderversuche/Nov22_Zaehne alt_Gewerbemuell.csv"
, sep = ",", header = TRUE, stringsAsFactors = FALSE)
test.min.2 <- test.data.2[seq.int(1, nrow(test.data.2), by = 4800),] #4s/0.008 differenz between datapo
ints = every 5000th
test.opt.2 <- subset(test.min.2, test.min.2[["Volumen.optisch.H.Sensor..m.3.h."]]>10)
test.opt.2 <- subset(test.opt.2, test.opt.2$Bandwaage..t.h.>0.1)
VRW_Dat_k.2 <- if(nrow(test.opt.2) > 200){test.opt.2[seq.int(1, nrow(test.opt.2)-75, by = 1),]} #Novemb
er Data shortend
```

Es werden Daten von den Versuchen bei Müllex im Juni ohne Regelung benutzt (VMJ_oR) und auch andere Versuchsdaten. Jetzt sind die Daten in der Variable VRW_Mod_k.[n] (Versuche ReWaste Daten kurz) und können jetzt umgeschrieben werden von einer Zeitreihe auf eine Massenreihe. Also Daten in regelmäßigen Masseabständen statt regelmäßigen Zeitabständen. Hier kann dann über die absolute Masse die vom Schredder zerkleinert wurde modelliert werden.Das ergibt im Prinzip das Integral für den Massenstrom über die Zeit, wobei sobald der Ziel Wert mit Toleranzbereich erreicht wird das Integral wieder anfängt die Produkte zu summieren. Zum vereinfachen vom Code auch auf Basis der Überlegung das eine Überschreitung des Toleranzbereiches nach oben nur eine Teilung vom Indexsprung erlaubt um nicht von unter dem Toleranzbereich auf über dem Toleranzbereich zu springen. Da eine Solche Teilung dann den Code erheblich kompliziert durch ausrechnen vom Verbleibenden Anteil bis zum nächsten Regulären Index und dann die Berechnung wieder bis zum nächsten Sprung. Diese Arbeit ist sicherlich möglich aber in dieser Version der Analyse wird die Schwankung um den Zielwert nach oben akzeptiert, wobei die Häufigkeit des Übertretens der oberen Toleranzgrenze auch mitbestimmt werden kann und dadurch das die addierten incrementellen Beträge zu über 97% unter der Toleranzbreite liegt, kann der Feheler des übertretens nach oben hin akzeptiert werden für diese Zwecke.

```
p_tolerence  <- 0.05 # 5% tolerance
target_value <- 0.06 * (1-p_tolerence) #data points per x tonnes
```

```
#Variable für Massendaten
VRW_Dat_k <- VMJ_Dat_k.1
VRW <- list(Vol_pert = c())
Mass_increments <- c()
indexrow <- 0
count <- 0

#transfomr t/h into t/s multiply by 4s per Data entry = Massincrement per datapoint
for (i in 1:nrow(VRW_Dat_k)) {
  Mass_increments[i] <- VRW_Dat_k$Bandwaage..t.h.[i]*diff(VRW_Dat_k$Time..s.)[i]/3600
  }
print(summary(Mass_increments))
print(quantile(Mass_increments, probs = 0.97, na.rm = TRUE))
boxplot(Mass_increments, range = c(1), main = "Boxplot of Massincrements with Quantile Wiskers")

#Umrechnung durch if Bedingung
current_Mass <- Mass_increments[1]
for(i in 1:length(Mass_increments)-1) {
  if (i > 1){
    current_Mass <- current_Mass + Mass_increments[i]
  }
  count <- count + 1

  if (current_Mass >= target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    current_Mass <- Mass_increments[i]
    indexrow <- indexrow + 1
    count <- 0
  }
  if (i == length(Mass_increments) && current_Mass < target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    current_Mass <- Mass_increments[i]
    count <- 0
  }
}

VMJ_ts1_m <- ts(VRW$Vol_pert)
plot(VMJ_ts1_m,main =  paste("Massseries VMJ_oR1; at every",target_value*1000/(1-p_tolerence),"kg", sep
= " "), ylab = "Volume", xlab = "Massincrements")
par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
acf(VMJ_ts1_m, main = paste("ACF of VMJ_oR; at every", target_value*1000/(1-p_tolerence),"kg", sep = "
"), col = "red")
pacf(VMJ_ts1_m, main = paste("PACF of VMJ_oR; at every", target_value*1000/(1-p_tolerence),"kg", sep =
" "), col = "purple")
```

The smaller the Waste increments are choosen the more a correlation between the mass increments is significant. This is indicative that the correlation is dependent on the mass, which makes physical sense since a smaller increment will mean that more of the same material is passing by the scale and laser sensors which will have more similar proporties than two different materials. However this also means that the avarage Timeframe for any significant correlation depends upon the mass stream which is the exact measure that is to be controlled. Which also leads to the realization that a small output (low mass stream) will allow for better models while a high output will only allow for a very short sighted forecast. On that note, since the total time span is 600s and there are 114 data points in the mass series, the average time span between two points is 5.3 s.

ReWaste Versuche November Zähne alt

```
#Variable für Massendaten
VRW_Dat_k <- VRW_Dat_k.2
Mass_increments <- c()

#transfomr t/h into t/s multiply by 4s per Data entry = Massincrement per datapoint
for (i in 1:nrow(VRW_Dat_k)) {
  Mass_increments[i] <- VRW_Dat_k$Bandwaage..t.h.[i]*diff(VRW_Dat_k$Time..s.)[i]/3600
  }
print(summary(Mass_increments))
print(paste("90/10 ratio Massincrements:", quantile(Mass_increments, probs = 0.9, na.rm = TRUE)/quantil
e(Mass_increments, probs = 0.1, na.rm = TRUE), sep = " "))
print(paste("97th Quantile of Massincrements:", quantile(Mass_increments, probs = 0.97, na.rm = TRUE),
sep = " "))
boxplot(Mass_increments, range = c(1), main = "Boxplot of Massincrements with Quantile Wiskers")
hist(Mass_increments, breaks = 20)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## 0.0002212 0.0156582 0.0238159 0.0249021 0.0329084 0.0941673         1
## [1] "90/10 ratio Massincrements: 4.86825090577833"
## [1] "97th Quantile of Massincrements: 0.0494426087273492"
```

```
VRW_Dat_k <- VRW_Dat_k.2
VRW <- list(Vol_pert = c())
p_tolerence  <- 0.05 # 5% tolerance
target_value <- 0.5 * (1-p_tolerence) #data points per x tonnes
indexrow <- 0
count <- 0
#Umrechnung durch if Bedingung
current_Mass <- Mass_increments[1]
for(i in 1:length(Mass_increments)-1) {
  if (i > 1){
    current_Mass <- current_Mass + Mass_increments[i]
  }
  count <- count + 1

  if (current_Mass >= target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    current_Mass <- Mass_increments[i]
    indexrow <- indexrow + 1
    count <- 0
  }
  if (i == length(Mass_increments) && current_Mass < target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    current_Mass <- Mass_increments[i]
    count <- 0
  }
}

VRW_ts2_m <- ts(VRW$Vol_pert)
plot(VRW_ts2_m, main =  paste("Volume per h at mass points; at every", target_value*1000/(1-p_tolerenc
e),"kg", sep = " "), ylab = "Volume per h", xlab = "Massincrements")
Box.test(VRW_ts2_m, lag = log(length(VRW_ts2_m)), type = "Ljung-Box")
par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
acf(VRW_ts2_m, main = paste("ACF of VRW Nov aZ; at every", target_value*1000/(1-p_tolerence),"kg", sep
= " "), col = "red")
pacf(VRW_ts2_m, main = paste("PACF of VRW Nov aZ; at every", target_value*1000/(1-p_tolerence),"kg", se
p = " "), col = "purple")
```

These results show that there is no significant correlation between the lags of the Mass Series using increments of 500 kg. This is an indication that either the Mass increments are too large (the correlation declines after a few hundered kilos) or the correlation would only be time but not mass dependent, however the second option is not likely as the mass is a time dependent function since the data points are created by multiplying the current Mass flow with the time that passed since the last data point. Therefore the Mass points every x kg is a passage of time with irregular increments but still a funktion of time.

Looking at the same Data but with a smaller target Value, we revieve an ACF that has a few 2 significant lags and then some non significant periodicity.

```
VRW_Dat_k <- VRW_Dat_k.2
VRW <- list(Vol_pert = c())
p_tolerence  <- 0.00 # 5% tolerance lower to upper limit
target_value <- 0.07 * (1-p_tolerence) #data points per x tonnes
indexrow <- 0
count <- 0
checkmass <- c(target_value/(1-p_tolerence))
#Umrechnung durch if Bedingung
current_Mass <- Mass_increments[1]
for(i in 1:length(Mass_increments)-1) {
  if (i > 1){
    current_Mass <- current_Mass + Mass_increments[i]
  }
  count <- count + 1

  if (current_Mass >= target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    checkmass <- c(checkmass, current_Mass)
    current_Mass <- Mass_increments[i]
    indexrow <- indexrow + 1
    count <- 0
  }
  if (i == length(Mass_increments) && current_Mass < target_value) {
    if (length(VRW$Vol_pert) == 0){
      VRW$Vol_pert <- mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i])
    } else {
      VRW$Vol_pert <- c(VRW$Vol_pert, mean(VRW_Dat_k$Volumen.optisch.H.Sensor..m.3.h.[(i-count+1):i]))
    }
    current_Mass <- Mass_increments[i]
    count <- 0
  }
}

VRW_ts2_m <- ts(VRW$Vol_pert)
plot(VRW_ts2_m, main =  paste("Volume per h at mass points; at every", target_value*1000/(1-p_tolerenc
e),"kg", sep = " "), ylab = "Volume per h", xlab = "Massincrements")
Box.test(VRW_ts2_m, lag = log(length(VRW_ts2_m)), type = "Ljung-Box")
par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
acf(VRW_ts2_m, main = paste("ACF of VRW Nov aZ; at every", target_value*1000/(1-p_tolerence),"kg", sep
= " "), col = "red")
pacf(VRW_ts2_m, main = paste("PACF of VRW Nov aZ; at every", target_value*1000/(1-p_tolerence),"kg", se
p = " "), col = "purple")
acf(diff(VRW_ts2_m), main = paste("ACF of diff VRW Nov aZ; at every", target_value*1000/(1-p_tolerenc
e),"kg", sep = " "), col = "red")

print(paste("Checkmass Var:", var(checkmass), sep = " "))
summary(checkmass)
par(mfrow = c(1,1), mar = c(2,3,3,2))
boxplot(checkmass, main = paste("Massincrements for target of:", target_value/(1-p_tolerence), "t", sep
= " "))
print(paste("90/10 ratio Mass Separation:", quantile(checkmass, probs = 0.9, na.rm = TRUE)/quantile(che
ckmass, probs = 0.1, na.rm = TRUE), sep = " "))
print(paste("97th Quantile of Mass separation:", quantile(checkmass, probs = 0.97, na.rm = TRUE), sep =
" "))
par(mar=c(4, 4, 2, 2))
```

```
hist(checkmass, main = paste("Mass separation for target of:", target_value/(1-p_tolerence), "t", sep =
```

The Mass series has an MA(q)=2 and AR(p)=1 with a period of 7 which is barely significant.

Preparing the steps to make a time series and mass series Model:

```
Box.test(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h., lag = log(length(VRW_Dat_k.2$Volumen.optisch.H.S
ensor..m.3.h.)), type = "Ljung-Box")
Box.test(VRW_Dat_k.2$Bandwaage..t.h., lag = log(length(VRW_Dat_k.2$Bandwaage..t.h.)), type = "Ljung-Bo
x")

LD_series <- seq(1,0,by = -1/15) #15 steps at 4s/step is 1min rolling weighted mean
LD_series <- LD_series/sum(LD_series) #Normalize LD-series

geometric_series <- 0.7^(1:15) # Normalize the vector to sum up to one (15 steps at 4s/step -> 1min rol
l mean)
geometric_series <- geometric_series / sum(geometric_series)
abscissa <- seq.int(0, to = 900, by = 25)
plot.ts(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h., main = "Volume over Time VRW-Nov", ylab = "Volum
e", xlim = c(200, 600), xaxt = "n")
axis(1, at = abscissa, labels = abscissa )
lines(filter(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h., LD_series, sides = 1, method = "convolutio
n"), col = "orange")

par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
 acf(diff(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.), main ="ACF of diff Vol VRW-November Zähne al
t", col = "red")
 pacf(diff(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.), main ="PACF of diff Vol VRW-November Zähne al
t", col = "purple")

par(mfrow = c(2,1), mar = c(2, 1, 3, 1))
 acf(diff(VRW_Dat_k.2$Bandwaage..t.h.), main ="ACF of diff Mass VRW-November Zähne alt", col = "red")
 pacf(diff(VRW_Dat_k.2$Bandwaage..t.h.), main ="PACF of diff Mass VRW-November Zähne alt", col = "purpl
e")
```

The total data set for the Vol in a ACF Diagram has 3 significant peaks and PACF has 3 with some seasonality of 5, the Mass ACF has 2 peaks and the PACF has 2 with some decaying seasonality. Which would indicate a MA(q)=3 and AR(p)=3. However if we use the Mass-Series then we receive MA(q)=2 and AR(p)=2 with a Period of 3 but the peaks are less significant. There more data points being combined to form a new data point for the mass series over the time series.

Finding a Model to describe the Time Series:

```
VRW_ts <- ts(diff(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.))
d=NULL
DD=NULL
d=1
DD=0
Compare_S_models <- data.frame( p = c(0), d = c(0), q = c(0), P = c(0), D = c(0), Q = c(0), AIC = c(0),
SSE = c(0), pVa = c(0) )
per=5

for(p in 1:3){
  for(q in 1:3){
    for(i in 1:2){
      for(j in 1:2){
        if(p+d+q+i+DD+j<=10){
          model<-arima(x=VRW_ts, order = c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-1)), per
iod=per))
          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
          sse<-sum(model$residuals^2)
          Compare_S_models[nrow(Compare_S_models) + 1, ] = list( p-1, d, q-1, i-1, DD, j-1, model$aic,
sse, round(pval$p.value,4))
        }
      }
    }
  }
}

Compare_S_models <- Compare_S_models[order(Compare_S_models$AIC, decreasing = FALSE), ]
p <- Compare_S_models$p[2]
d <- Compare_S_models$d[2]
q <- Compare_S_models$q[2]
P <- Compare_S_models$P[2]
D <- Compare_S_models$D[2]
Q <- Compare_S_models$Q[2]

print(Compare_S_models)
print(paste("p:",p,"d:",d, "q:",q,"P:",P,"D:",D,"Q:",Q))
```

Next we take the p,d,q,P,D,Q values of the lowest AIC to forecast the next values.

Now we can make a forecast:

```
library(astsa)
```

```
## Warning: Paket 'astsa' wurde unter R Version 4.2.2 erstellt
```

```
library(forecast)
```

```
## Warning: Paket 'forecast' wurde unter R Version 4.2.2 erstellt
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attache Paket: 'forecast'
```

```
## Das folgende Objekt ist maskiert 'package:astsa':
##
##     gas
```

```
VRW_ts <- ts(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.)
sarima(VRW_ts, p,d,q,P,D,Q,per)

model1<- arima(x=VRW_ts, order = c(p,d,q), seasonal = list(order=c(P,D,Q), period=per))

n = 20
VRW_fore_k2_ts <- forecast(model1, h = n )
plot(VRW_fore_k2_ts, main = "Forecast for VRW_Dat_k.2 with ARIMA", xlim = c(600, length(VRW_ts)+n))
```

With Time Series Forecast we compare it with real data: test.opt.2

```
VRW_ts_end <- test.opt.2$Volumen.optisch.H.Sensor..m.3.h.[(length(VRW_Dat_k.2$Volumen.optisch.H.Senso
r..m.3.h.)+1):(length(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.)+n)]

VRW_ts_resid <- VRW_fore_k2_ts$mean - VRW_ts_end
#compared with taking the LD mean of the last 5 values as a constant.
LD_5 <- seq(0,1,by = 1/5)
LD_5 <- LD_5/sum(LD_5)
Bench_Value <- sum(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.[(length(VRW_Dat_k.2$Volumen.optisch.H.S
ensor..m.3.h.)-5):length(VRW_Dat_k.2$Volumen.optisch.H.Sensor..m.3.h.)]*LD_5)
VRW_bench <- Bench_Value - VRW_ts_end

VRW_SSE <- sum(VRW_ts_resid^2)
VRW_bench_SSE <- sum(VRW_bench^2)
print(paste("Time Series forecast", "SSE of Benchmark:", round(VRW_bench_SSE,1), " SSE of Forecast:", r
ound(VRW_SSE,1), "Forecast is better:", VRW_SSE<=VRW_bench_SSE, "at (%)", round((VRW_SSE/VRW_bench_SSE)
*100, 3), sep = " "))
```

Comparison of Time Series:

```
plot.ts(VRW_ts_end, type = "l", lty = 1, main = paste("Forecast and Emperical comparison; Time Series",
sep = " "), ylab = "Volume output [m^3/h]", xlab = "Time series", ylim = c(0,490))
lines(as.vector(VRW_fore_k2_ts$mean), col ="orange", lty = 6)
lines(as.vector(VRW_fore_k2_ts$upper[,2]), col ="grey")#95% Interval
lines(as.vector(VRW_fore_k2_ts$lower[,2]), col ="grey")
lines(rep(Bench_Value, times = 20), col = "red", lty = 5)
#lines(rep(VRW_ts2_m[length(VRW_ts2_m)]+diff(VRW_ts2_m[(length(VRW_ts2_m)-1):length(VRW_ts2_m)]), times
= 20), col = "green", lty = 4)
legend(14,175,legend=c("Vol-Real","Vol-Model","Benchmark"),col=c("black","orange","red"),lty=c(1,6,5),p
ch=c())
```

Finding a Model to describe the Mass series:

```
d=NULL
DD=NULL
d=1
DD=0
Compare_S_models <- data.frame( p = c(0), d = c(0), q = c(0), P = c(0), D = c(0), Q = c(0), AIC = c(0),
SSE = c(0), pVa = c(0) )
per=7
VRW_ts_m <- VRW_ts2_m

for(p in 1:3){
  for(q in 1:3){
    for(i in 1:2){
      for(j in 1:2){
        if(p+d+q+i+DD+j<=10){
          model<-arima(x=VRW_ts_m, order = c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-1)), p
eriod=per))
          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
          sse<-sum(model$residuals^2)
          Compare_S_models[nrow(Compare_S_models) + 1, ] = list( p-1, d, q-1, i-1, DD, j-1, model$aic,
sse, round(pval$p.value,4))
        }
      }
    }
  }
}

Compare_S_models[order(Compare_S_models$AIC, decreasing = FALSE), ]
```

Following the Parsimony Principle and using the lowest AIC while observing that the model with the lowest SSE is very close to the simpler Model with the lowest AIC, the simpler Model is chosen. (1,0,0)(0,1,1) per 5 AIC = 6530.6 (1,0,0)(1,1,2) per 3 AIC = 6537.2 (2,0,2)(0,1,1) per 7 AIC = 6514.7 SSE = 4063960 (2,0,1)(0,1,1) per 7 AIC = 6513.4 SSE = 4065669 (1,0,0)(0,1,1) per 7 AIC = 6515.5 SSE = 4112116 (1,1,1)(0,1,1) per 7 AIC = 6511.5 SSE = 4020819 (2 - Niedriger AIC und SSE) (2,1,2)(0,0,0) per 7 AIC = 6552.5 SSE = 4016650 (1 - Niedriger SSE und keine Periode berücksichtigen müssen)

The Step of choosing the best Model could likley be automated, for our goals. And would need to be if it is used as part of a control loop.

```
library(astsa)
#VRW_ts2_m <- VRW_ts2_m[!is.na(VRW_ts2_m)]
sarima(VRW_ts2_m, 0,1,2,0,0,0,7)
sarima(VRW_ts2_m, 2,1,1,0,0,0,7)
sarima(VRW_ts2_m, 1,1,2,0,0,0,7)
sarima(VRW_ts2_m, 1,1,1,0,1,1,7)
sarima(VRW_ts2_m, 2,0,1,0,1,1,7)
```

Several models seem to fit rather well, since the Residuals have no obvious trend and seem to be normally distributed in the QQ plot, while the ACF is not significant and the p-Values of the Ljung-Box test are above the threshold for rejecting H_0. The simpler model with a DD = 0 even has higher p-Values and the ACF of Residuals is lower. Therefore the Model (2,1,2)(0,0,0) is chosen.

Now we can make a forecast:

```
library(astsa)
library(forecast)

model1<- arima(x=VRW_ts2_m, order = c(2,1,2), seasonal = list(order=c(1,0,0), period=7))

n = 20
plot(forecast(model1, h = n))
VRW_fore_k2 <- forecast(model1, h = n )
```

Convert the last minutes to a mass series to compare with the model forecast

```
#Variable für Massendaten
VRW_Dat_k <- test.opt.2
Mass_increments_l <- c()

#transfomr t/h into t/s multiply by 4s per Data entry = Massincrement per datapoint
for (i in 1:nrow(VRW_Dat_k)) {
  Mass_increments_l[i] <- VRW_Dat_k$Bandwaage..t.h.[i]*diff(VRW_Dat_k$Time..s.)[i]/3600
  }
print(summary(Mass_increments_l))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## 0.0002212 0.0150962 0.0233143 0.0244787 0.0325780 0.0941673         1
```

```
print(paste("97th Quantile of Massincrements:", quantile(Mass_increments_l, probs = 0.97, na.rm = TRU
E), sep = " "))
```

```
## [1] "97th Quantile of Massincrements: 0.0489065227508545"
```