UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Studying dehydration kinetics of tissue-mimicking hydrogels

Trabajo Fin de Grado-Treball Final de Grau

Grado en Ingeniería Biomédica-Grau en Enginyeria Biomèdica

AUTOR/A: Navarro Sabater, Erika

Tutores: Vilariño Ferrer, Guillermo; Lenk, Kerstin; Kainz, Manuel; Polz, Mathias;

CURSO ACADÉMICO 2023-2024

Erika Navarro Sabater

Studying dehydration kinetics of tissue-mimicking hydrogels.

BACHELOR'S THESIS
to achieve the university degree of
Bachelor of Biomedical Engineering

submitted to
Graz University of Technology

Supervisors
Lenk, Kerstin. Dipl.-Inf. [FH] Dr.rer.nat.
Kainz, Manuel, Dipl.-Ing. BSc
Polz, Mathias, Dipl.-Ing. BSc
Institute of Neural Engineering
Institute of Biomechanics
Institute of Health Care Engineering with European Testing Center of Medical Devices

Graz, September 2024

Abstract

Hydrogels are materials with the capacity to absorb and release significant amounts of fluid in response to external factors such as mechanical stress, pH changes, or temperature variations. This property is utilized in applications like drug delivery and impurity absorption. The fluid absorption process is called swelling or hydration, and its reverse is known as deswelling or dehydration. This study evaluates several models for characterizing the deswelling behavior of hydrogels, including the First-Order Kinetic Model, Fickian Diffusion Model, Non-Fickian Diffusion Model, Double Exponential Non-Linear Regression Model, Broken Stick Model, and Polynomial Model (2nd Degree). Among these, the Polynomial Model (2nd Degree) proved to be the most effective in capturing deswelling dynamics, providing superior fit quality. Future research could explore applying these models to different hydrogels or conditions to further enhance their practical applicability.

Contents

CHAPTER 1

Introduction

The study of hydrogels holds paramount importance within the field of biomedical engineering. Hydrogels, hydrated networks of polymers, possess the remarkable ability to retain substantial amounts of water within their intricate three-dimensional structure, rendering them highly relevant for applications such as tissue engineering and advanced drug delivery systems [1]. These materials can mimic the properties of biological tissues, providing a supportive environment for cell growth and proliferation [2]. Furthermore, the high water content and soft, (visco)elastic nature of hydrogels make them ideal candidates for replicating the extracellular matrix of various tissues, including brain tissue [2].

Hydrogels are widely researched due to their biocompatibility, adjustable physical properties, and ability to mimic the extracellular matrix of biological tissue [2]. This makes them ideal candidates for creating tissue-like scaffolds that can support cell growth and tissue regeneration [3]. In particular, they can provide a suitable environment for neuronal cells, facilitating studies on neural development, disease modelling, and potential treatments for neurological disorders [4]. The motivation for this research stems from the need to improve our understanding of the fundamental hydration and dehydration mechanisms of these materials to optimise their design and application in biomedical engineering [5].

This thesis explores the hydration kinetics of brain tissue-mimicking hydrogels, with their potential applications, diverse physical and chemical properties, and the fundamental processes of hydration and dehydration [2]. Understanding these aspects is crucial for advancing their use in regenerative medicine and other critical biomedical applications [1].

CHAPTER 2

Background and Terminology

2.1 Hydrogels: Definition and Key Properties

Hydrogels are polymer networks that can absorb and retain substantial amounts of water without dissolving [2]. These materials are characterised by their unique ability to swell and maintain their structure, thanks to the crosslinking of polymer chains [6]. Key properties of hydrogels include:

- **Insolubility in Water**: Hydrogels maintain their structure in aqueous environments due to cross-linking, which prevents dissolution [3]. This insolubility is crucial for their function as scaffolds and drug delivery systems, as it ensures stability in the biological milieu [2].
- **Covalent and Non-covalent Interactions**: The polymer chains in hydrogels can be linked through covalent bonds or held together by non-covalent interactions, such as hydrogen bonding and van der Waals forces [3]. These interactions influence the mechanical properties and responsiveness of hydrogels to environmental changes [2].
- **Diffusivity and Permeability**: The porous nature of hydrogels facilitates the diffusion of molecules, which is crucial for applications requiring the controlled release or absorption of substances [2, 7]. This property is especially valuable in biomedical applications where hydrogels can be used to deliver drugs or absorb impurities [8, 5].
- **Mechanical Properties**: The inherent softness of hydrogels makes them suitable for mimicking the mechanical properties of soft tissues, such as the brain [4, 9]. This softness is essential for applications that require flexibility and adaptability, ensuring that the hydrogels can conform to and interact with various biological environments effectively [1, 10].
- **Biocompatibility**: Hydrogels are widely recognized for their biocompatibility, particularly in medical and pharmaceutical applications. They are designed to be compatible with biological tissues, supporting essential processes such as cell attachment, proliferation, and differentiation [1, 5]. This property is crucial for applications in

tissue engineering, where hydrogels must integrate with living tissues and promote the regeneration of damaged or diseased areas [8, 9]. Additionally, their biocompatibility extends to interactions with various cell types and tissues, ensuring minimal adverse effects in medical implants and drug delivery systems [7, 10].

2.2 Applications of Soft Hydrogels in Biomedical Engineering

Soft hydrogels have numerous applications in biomedical engineering, particularly in tissue engineering and drug delivery [1]:

- **Tissue Engineering in Neuroscience**: Hydrogels serve as scaffolds that support cell growth and neural tissue regeneration. Their high water content and softness provide a permeable environment for the growth of delicate neural tissues [2]. For instance, hydrogels can be used to create models of brain tissue for studying neurodegenerative diseases or for developing new therapies [1].
- **Controlled Delivery Systems**: Hydrogels can be used to create controlled release systems of drugs and genes, ensuring a sustained and localized delivery [2]. This is particularly useful for delivering therapeutic agents directly to the brain, bypassing the blood-brain barrier [1].
- **Dermal Patches and Dressings**: Hydrogels are used in wound care products to maintain a moist environment, which is beneficial for healing. Their ability to conform to the wound surface and provide a protective barrier makes them ideal for treating burns and other injuries [3].

2.3 Hydration and Dehydration of Hydrogels

The processes of hydration (swelling) and dehydration (deswelling) are fundamental to the behaviour of hydrogels [11] [12]:

- **Hydration (Swelling)**: Hydrogels absorb water, expanding as they retain water between the polymer chains [3]. This swelling process increases the volume of the hydrogel. The degree of swelling can be influenced by factors such as the polymer composition, crosslinking density, and the presence of ions or other solutes in the surrounding environment [11] [12].
- **Dehydration (Deswelling)**: Upon losing water, hydrogels contract and return to their original volume [3]. The rate and extent of dehydration can be influenced by various factors, including temperature, humidity, and the chemical structure of the hydrogel [11].

The mechanisms underlying these processes are complex, involving the interplay of osmotic pressure, polymer elasticity, and environmental conditions [12]. For example, the osmotic pressure difference between the hydrogel and the surrounding solution drives water into the hydrogel during swelling, while the polymer network's elasticity resists this expansion [11].

2.3.1 Bound Water evaporates differently

In complex materials such as hydrogels, water is not uniformly distributed. There are two primary types of water: **free water** and **bound water**. These two types of water behave differently during evaporation processes due to their distinct characteristics [13].

Free Water

Free water is water that is not tightly bound to the structure of the material. In hydrogels, free water is typically found in the intercellular spaces or between molecules. This type of water behaves similarly to pure water, with a vapour pressure almost equal to that of pure water. As a result, free water evaporates easily when heat is applied, as it requires less energy to change from a liquid to a vapour state. During drying or dehydration processes, free water is the first to evaporate. The rapid evaporation of free water does not cause significant changes to the structure of the material, as it is not strongly attached to the solid matrix [13].

Bound Water

In contrast, bound water is water that is strongly attached to the solid matrix of the material, either at the molecular level or through chemical bonds with components such as proteins, carbohydrates, and other polymers. Bound water possesses unique characteristics that set it apart from free water [13]:

1. **Lower Vapour Pressure**: Bound water has a significantly lower vapour pressure compared to free water, which means that more energy is required for it to evaporate [13].
2. **Higher Binding Energy**: Bound water is held within the material by stronger bonds than free water, such as hydrogen bonds and Van der Waals forces. This requires a higher amount of energy to break these bonds during evaporation [13].
3. **Reduced Mobility**: Due to the nature of its bonds, the molecules of bound water have lower mobility compared to free water molecules. This also means that bound water is less accessible for evaporation processes [13].
4. **Non-Freezing at Low Temperatures**: Unlike free water, bound water does not freeze at low temperatures, making it particularly difficult to remove in some dehydration and preservation processes [13].
5. **Dielectric Properties and Higher Boiling Point**: Bound water has a higher boiling point and different dielectric properties compared to free water. This influences the dehydration and preservation methods that can be applied to effectively remove it [13].

2.3.2 Parameters Affecting Dehydration

Several parameters change during the dehydration process and influence its dynamics [14]:

- **Volume and Weight**: As hydrogels dehydrate, both their volume and weight decrease. This is due to the loss of water from the polymer network [14] [15].
- **Mechanical Properties**: Dehydration often results in increased stiffness and brittleness [2]. The loss of water reduces the hydrogel's flexibility and can lead to changes in its mechanical integrity [15].
- **Porosity and Water Content**: The porosity of the hydrogel decreases as water is expelled [2]. This reduction in porosity can affect the diffusion of molecules through the hydrogel and its overall functionality [8].
- **Environmental Factors**: Temperature, humidity, airflow, and pressure can significantly impact the dehydration rate. For example, higher temperatures and lower humidity levels generally accelerate dehydration [15].
- **Hydrogel Composition**: The chemical makeup of the hydrogel, including crosslinking density and solute concentration, plays a critical role in its dehydration behaviour [2]. Hydrogels with higher crosslinking densities tend to retain water more effectively, leading to slower dehydration rates [3].

These parameters are categorized into physical (describing properties and environmental conditions) and model-specific (used in mathematical models) [8]. Some parameters overlap both categories, as classified in the Table 2.1, aiding in systematic hydrogel research [2].

Table 2.1: Classification and description of parameters used in hydrogel dehydration studies, including specific parameters for various models.

Parameter	**Classification**	**Description**
Deswelling Degree (DS)	Physical	Quantifies the extent of water loss from the swollen hydrogel.
Time to Dehydrate (T)	Physical	Represents the time taken for the hydrogel to dehydrate to one-third of its original hydration level. Indicates how quickly or slowly the material loses water under specific environmental conditions.
Rate of Dehydration (R)	Generic	Quantifies the speed at which the hydrogel dehydrates. Derived from the slope of the dehydration profile curve obtained from experimental data.
Diffusion Coefficients	Both	Reflect the mobility of water molecules within the hydrogel network. Influenced by polymer chain interactions and crosslinking density.

Parameter	Classification	Description
Crosslinking Density	Physical	Determines the mesh size of the hydrogel network. Affects both the rate and extent of dehydration.
Temperature and pH	Both	External conditions that can affect the swelling equilibrium and dehydration kinetics of hydrogels.
Reaction Rates	Both	For chemically reactive hydrogels, parameters such as reaction rate constants and activation energies are essential for modelling dehydration-induced structural changes.
Initial Conditions	Both	Initial water content, polymer composition, and swelling state influence the kinetics of dehydration and subsequent structural changes.
Mechanical Properties	Physical	Shear moduli (G_1, G_2, ...), viscosity parameter (Z), and other mechanical parameters define the resistance of the hydrogel to deformation and its ability to store and dissipate energy.
First-Order Kinetic Model Parameters	Model-Specific	For the First-Order Kinetic Model: $M(t)$ (amount of water lost at time t), M_∞ (maximum water loss), k (rate constant), t (time).
Double Exponential Model Parameters	Model-Specific	For the Double Exponential Model: A (relative weight of the first exponential term), τ_1 and τ_2 (time constants for the two exponential terms), M_∞ (equilibrium water loss).
Fickian Diffusion Model Parameters	Model-Specific	For the Fickian Diffusion Model: $M(t)$ (amount of water lost at time t), M_∞ (equilibrium water loss), D (diffusion coefficient), L (characteristic length), t (time), erf (error function).
Non-Fickian Diffusion Model Parameters	Model-Specific	For the Non-Fickian (Anomalous) Diffusion Model: $M(t)$ (amount of water lost at time t), M_∞ (equilibrium water loss), τ (characteristic time), n (diffusion exponent).
Polynomial Model Parameters	Model-Specific	For a second-degree polynomial model: a_2 (coefficient of t^2), a_1 (coefficient of t), a_0 (constant term). These parameters define the curvature and linear behaviour of the polynomial fit.

Parameter	Classification	Description
Broken Stick Model Parameters	Model-Specific	For the Broken Stick model: A_1 (amplitude of the first exponential term), B_1 (decay rate of the first exponential term), A_2 (amplitude of the second exponential term), B_2 (decay rate of the second exponential term). These parameters characterise the exponential decay behaviour and the transition points of the model.
Key Diffusion and Thermodynamic Parameters	Generic	Diffusion coefficient (D), relaxation time (τ), water content (M_t and M_∞), diffusion exponent (n), rate constants (k), temperature (T), Gibbs free energy (ΔG), and other parameters specific to diffusion and thermodynamic models.

2.3.3 (Bio)Mechanical Implications of Dehydration

The dehydration of hydrogels has notable biomechanical implications:

- **Increased Stiffness and Brittleness**: As water is lost, hydrogels become stiffer and more brittle, losing some of their flexibility [2]. This change in mechanical properties can impact their performance in applications where flexibility and softness are required [6] [3].
- **Contraction and Volume Decrease**: Dehydration leads to a reduction in volume and porosity, potentially altering the hydrogel's load-bearing capacity and impact absorption ability [2]. This is particularly important in applications where the hydrogel needs to maintain a specific shape or volume [14] [11].

2.4 Formulations for Modelling Dehydration of Hydrogels

Hydrogels exhibit complex swelling and deswelling behaviors due to their porous structure and interaction with fluids. Understanding these behaviors is crucial for optimizing their performance in various applications, from drug delivery to tissue engineering [10, 1]. To capture these dynamics, several mathematical models have been developed. These models aim to describe the fluid transport mechanisms within hydrogels under various conditions, such as temperature, pH changes, or mechanical stress [16, 11]. Each model offers a different perspective on the dehydration process, focusing on specific mechanisms and assumptions [17, 18].

Several models are used to describe the dehydration of hydrogels, each with its specific focus and assumptions:

- **Diffusion Models**: These models, based on Fick's laws of diffusion, characterise the transport of water within the hydrogel matrix. They describe how the concentration gradient drives the movement of water molecules, accounting for factors such as diffusion coefficients and boundary conditions [8].
- **Swelling Equilibrium Models**: These models predict the equilibrium swelling behaviour of hydrogels in response to external stimuli such as temperature and pH [12]. They consider the balance between osmotic and elastic forces within the hydrogel [8].
- **Mechanical Models**: These models describe the mechanical response of hydrogels during dehydration, including stress, strain, and deformation. They help in understanding how the hydrogel's mechanical properties evolve as it loses water [8].

This thesis delves into the following models for analysing the dehydration behaviour of hydrogels: the **kinetic models**, which include the First-Order Kinetic Model for Deswelling [19]; the **diffusion models**, such as the Fickian Diffusion Model [20] and the Non-Fickian (Anomalous) Diffusion Model [21]; and the **empirical fitting models**, which comprise the Double Exponential Non-Linear Regression Model [22], the Second-Degree Polynomial Model [23], and the Broken Stick Model [18].

First-Order Kinetic Model for Deswelling:

- This model is used to describe the deswelling behaviour of hydrogels [24].
- The deswelling degree (DS) is calculated using the equation [7]:

$$DS = \frac{M_e - M_t}{M_e - M_0} \tag{2.1}$$

  where:
  - M_t represents the mass of the hydrogel at time t,
  - M_e is the weight of the swollen gel at equilibrium,
  - M_0 is the initial dry weight.
- The model is further described by the equation [19]:

$$M(t) = M_\infty \cdot \left(1 - e^{-kt}\right) \tag{2.2}$$

  where:
  - $M(t)$ is the amount of water lost at time t,
  - M_∞ is the maximum water loss (equilibrium water loss),
  - k is the rate constant, indicating the speed of deswelling,
  - t represents time.

- This model quantifies the water expulsion rate from the hydrogel network over time [16].
- Key assumptions include:
  - It assumes that deswelling follows first-order kinetics, where the rate of change in deswelling degree is directly proportional to the remaining water content relative to the equilibrium state [7].
  - It assumes uniform deswelling throughout the hydrogel matrix [7].
  - Environmental conditions, such as temperature and humidity, are assumed to remain constant during deswelling [7].

Double Exponential Non-Linear Regression Model:

- This model is commonly used to fit thermogravimetric analysis (TGA) data, allowing the determination of both dehydration time and rate [25] [22].
- It is described by the equation:

$$M(t) = M_\infty \left[A \cdot e^{-t/\tau_1} + (1 - A) \cdot e^{-t/\tau_2} \right] \tag{2.3}$$

  where:
  - $M(t)$ represents the amount of water lost at time t,
  - M_∞ is the equilibrium water loss,
  - A is the relative weight of the first exponential term,
  - τ_1 and τ_2 represent the dehydration rates of the two exponential terms.
- The model captures the dehydration profiles of various hydrogel materials under different environmental conditions [25].
- Assumptions include:
  - It assumes that the dehydration process can be effectively represented by a double exponential decay function, capturing multiple stages of water loss [22].
  - It assumes that TGA data accurately reflects the dehydration kinetics of the hydrogel [25].
  - No significant experimental errors or artifacts are assumed to influence the TGA measurements [25].

Fickian Diffusion Model:

- This model describes dehydration as a diffusion-controlled process following Fick's laws of diffusion [20].

- It is expressed by the equation:

$$M_t = M_\infty \left(1 - \text{erf}\left(\frac{L}{2\sqrt{Dt}}\right)\right) \tag{2.4}$$

  where:
  - M_t is the amount of water lost at time t,
  - M_∞ is the equilibrium water loss,
  - erf represents the error function,
  - L is the characteristic length of the material,
  - D is the diffusion coefficient,
  - t represents time.
- This model is characterised by a linear relationship between the square root of time and the amount of water lost [20].
- It is particularly applicable when the diffusion rate of water is slower than the polymer chain relaxation [20].
- Key assumptions are:
  - Water molecules diffuse through the hydrogel matrix according to Fick's laws, with the rate of diffusion proportional to the concentration gradient of water [20].
  - The hydrogel structure is assumed to be uniform and isotropic, facilitating even diffusion pathways [20].
  - It assumes no significant interactions between water molecules and polymer chains other than simple diffusion [20].

Non-Fickian (Anomalous) Diffusion Model:

- This model accounts for scenarios where diffusion and polymer relaxation occur on comparable time scales [21].
- It is described by the equation:

$$M_t = M_\infty \left(1 - \left(\frac{t}{\tau}\right)^n\right) \tag{2.5}$$

  where:
  - M_t represents the amount of water lost at time t,
  - M_∞ is the equilibrium water loss,
  - τ is the characteristic time,
  - n is the diffusion exponent, indicating the transport mechanism.

- Key assumptions are:
  - It assumes the diffusion process exhibits non-uniformity or heterogeneity, possibly due to polymer relaxation, swelling gradients, or internal structural changes [21].
  - The diffusion coefficient is assumed to vary over time or space within the hydrogel matrix [21].
  - Multiple transport mechanisms are assumed to influence water migration, such as polymer relaxation or hindered diffusion [21].

Second-Degree Polynomial Model:

- This model captures non-linear relationships between a dependent variable and one or more independent variables by incorporating a quadratic term [26].
- The model is expressed as:

$$Y = a_2X^2 + a_1X + a_0 + A_1 \cdot e^{-B_1 \cdot X} + A_2 \cdot e^{-B_2 \cdot X} \tag{2.6}$$

  where:
  - Y is the dependent variable,
  - X is the independent variable,
  - a_2 is the coefficient of the quadratic term,
  - a_1 is the coefficient of the linear term,
  - a_0 is the intercept,
  - A_1 and A_2 are the coefficients of the exponential terms,
  - B_1 and B_2 are the rate constants of the exponential terms [26].
- Assumptions include:
  - The relationship between the dependent and independent variables involves both quadratic and exponential components [26].
  - Polynomial models are useful for capturing non-linear relationships in data where simple linear models might not be sufficient [26].
  - The degree of the polynomial affects the model's complexity and its ability to fit the data, with higher degrees potentially leading to overfitting [26].

Broken Stick Model:

- This model analyses irregular longitudinal data by approximating each sample's trajectory with a series of connected straight lines, or "broken sticks," defined by user-specified breakpoints. This approach is effective for managing and interpreting datasets from drying experiments where the timing of measurements is inconsistent [18].

- The model is expressed by the equation:

$$Y(t) = \begin{cases} a_2 t^2 + a_1 t + a_0 & \text{for } t \leq t_c \\ b_2 t + b_1 & \text{for } t > t_c \end{cases} \tag{2.7}$$

  where:

  - $Y(t)$ is the amount of water lost at time t,
  - t_c is the critical time (breakpoint) where the transition between the two phases occurs,
  - a_2, a_1, and a_0 are the coefficients of the polynomial segment before t_c,
  - b_2 and b_1 are the coefficients of the linear segment after t_c.

- Assumptions include:
  - It assumes linear trajectories between breakpoints, which simplifies the representation of drying phases by standardising the timing of measurements across samples [18].
  - The breakpoint t_c represents a significant transition between the rapid and slow drying phases [18].

- Applications include:
  - Modelling drying kinetics and structural changes in materials with irregular data collection times, such as hydrogels and other porous materials [18].
  - Combining with other models like the Two-Phase Drying Model to enhance understanding of drying dynamics and improve accuracy in predicting drying behaviour [27].

CHAPTER 3

Methods

This section presents the methods used for processing and analyzing data from six tests on hydrogel deswelling. It outlines the data acquisition, cleaning procedures, and model fitting techniques applied. The aim is to identify the model that best represents the hydrogel's deswelling behavior.

3.1 Experimental Data

Data were provided from six different tests which involved measuring the deswelling (water loss) of hydrogels over time showing the weight and volume evolution of a hydrogel sample , as we can see in table 3.1, (5 mm diameter, 2 mm height) at temperatures ranging from 21 to 24 °C and a humidity of about 50%.

All experiments were repeated at 37 °C with humidity levels of 40%, 60%, and 80%, and the hydrogel discs had a height of 1 mm.

Time	S S	Weight (g)	Area Side	Area Top
13:56:11	S S	0.0393	5188100.0	4450.0
13:56:41	S S	0.0391	5196700.0	4450.0
13:57:11	S S	0.0390	5195550.0	4450.0
13:57:41	S S	0.0388	5193300.0	4450.0
13:58:12	S S	0.0388	5164750.0	4450.0
13:58:42	S S	0.0387	5171650.0	4450.0
13:59:13	S S	0.0385	5170450.0	4450.0
13:59:43	S S	0.0385	5127450.0	4450.0
14:00:13	S S	0.0383	5106800.0	4450.0
14:00:43	S S	0.0382	5105400.0	4450.0

Table 3.1: First 10 rows of the pilot test dataset.

3.2 Data Processing

The data processing was conducted in MATLAB and involved the following steps:

- Loading Data: The experimental data from various Excel files were imported into MATLAB.
- Data Cleaning: Non-numeric entries were removed from the data, as the Excel files contained both numerical and textual data.
- Extracting Relevant Columns: The columns corresponding to time and weight were extracted for further analysis.
- Outlier Removal: To enhance the accuracy of the data analysis:
  - For the Pilot Test, many rows contained missing information (NaN). These rows were removed from the dataset as they did not provide valid data for model fitting.
  - For Test 3, Mahalanobis distance was employed to identify and remove outliers. This method detects data points that significantly deviate from the overall pattern of the multidimensional dataset.
  - For Test 2 and Test 4, anomalous behaviours were observed at specific time points. In Test 2, the dataset was truncated at second 0.57 to exclude subsequent data exhibiting unusual patterns. Similarly, for Test 4, the dataset was truncated at second 0.48400 due to unexpected data trends. For the remaining tests, outlier removal was not performed as visual inspection did not reveal any significant anomalies.
- Rescaling Time: Time data were rescaled by a factor of 100,000 to facilitate the analysis.

The data included the following parameters:

- Weight (Mt): The mass of the hydrogel sample at different time points.
- Time ($Time$): The time at which weight measurements were taken.
- Area Side ($area_{\mathbf{side}}$): The side area of the hydrogel sample (not used directly in the kinetic models).
- Area Top ($area_{\mathbf{top}}$): The top area of the hydrogel sample (not used directly in the kinetic models).

3.2.1 Model Fitting

The following models were fitted to the data:

First-Order Kinetic Model

The First-Order Kinetic Model was fitted using non-linear regression. The deswelling degree (DS) was calculated using the equation (2.2). The ‘nlinfit‘ function was used to estimate the model parameters by minimising the sum of squared residuals between the observed and predicted values.

Double Exponential Non-Linear Regression Model

For the Double Exponential Non-Linear Regression Model, a non-linear fitting approach was employed. The model combines two exponential functions to describe the data (2.3). Initial guesses for the parameters were based on preliminary analyses. The ‘lsqcurvefit‘ function was used to optimise the model parameters, minimising the residuals.

Fickian Diffusion Model

The Fickian Diffusion Model was fitted using a non-linear least squares approach. The model describes diffusion behaviour based on Fick’s laws (2.4). Parameter estimation was performed using the ‘nlinfit‘ function, with the model’s parameters being adjusted to fit the experimental data.

Non-Fickian (Anomalous) Diffusion Model

For the Non-Fickian (Anomalous) Diffusion Model, the fitting process involved using a power-law function to capture the anomalous diffusion behaviour (2.5). Parameters were estimated by fitting the model to the data using ‘nlinfit‘, with the focus on accurately capturing deviations from Fickian diffusion.

Polynomial Model (2nd Degree)

The Polynomial Model of second degree (2.6) was fitted using the ‘polyfit‘ function to estimate the coefficients of a quadratic polynomial. The model’s fit was evaluated by plotting the observed data against the predicted values and assessing the residuals.

Two-Phase Drying Model

The Two-Phase Drying Model included a polynomial model for the initial phase and a linear model for the subsequent phase. An optimisation approach was used to determine the breakpoint time, ‘tc‘, which delineates the transition between the polynomial and linear phases. The ‘lsqcurvefit‘ function was employed to optimise the breakpoint parameter. The overall model was evaluated using the segmented function (2.7) that combines the polynomial and linear models.

3.2.2 Initial Parameter Values

Despite evaluating several models, including the First-Order Kinetic Model, Double Exponential Non-Linear Regression Model, Fickian Diffusion Model, and Non-Fickian (Anomalous) Diffusion Model, it was found that these models did not sufficiently capture the deswelling behavior of hydrogels. Consequently, only the Polynomial Model (2nd Degree) and the Broken Stick Model were used to extract parameters, as these models provided a more accurate fit to the data.

For the **Polynomial Model (2nd Degree)**, the initial parameter values for the quadratic coefficients were determined based on preliminary data analysis and literature insights [26]. Initially, the coefficients were set to approximate values close to zero, as polynomial coefficients are often initialized based on the data range or preliminary observations [26]. In the MATLAB code, these initial guesses were systematically adjusted through iterative fitting procedures to improve the model's accuracy. The optimization process involved refining these estimates to capture the quadratic trends in the data effectively, with adjustments made to minimize the residuals and improve the fit quality [26].

For the **Two-Phase Drying Model**, the initial value for the breakpoint time t_c, which distinguishes the polynomial and linear phases, was determined using an optimisation approach. The initial guess for t_c was based on visual inspection of the drying curves and adjusteding iteratively to find the value that provided the best fit to the experimental data. This optimisation process involved systematically varying t_c to minimise the difference between the model predictions and the observed data, ensuring that the model accurately represented the drying process [9].

3.2.3 Results Visualization and Error Handling

For each model, the results were visualized by plotting experimental data against model predictions. MATLAB's plotting functions were used to generate these plots, providing a clear comparison between observed and modeled data. Error handling mechanisms were incorporated throughout the process to address potential issues such as invalid data formats or numerical problems during fitting.

3.2.4 Evaluation of Fit Quality

Evaluation of the fit quality was performed by calculating several goodness-of-fit metrics including Sum of Squared Errors (SSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared values to assess the models performance.

The SSE offers an overall indication of the fit by summing the squared differences between predicted and actual values, while the RMSE and MAE provide scaled and absolute error measures, respectively, which help in understanding the model's precision [23]. The R^2 value further evaluates the model's effectiveness by indicating the fraction of the data's variability

that is captured by the model [23]. A high R^2 suggests a good fit, while lower values may indicate that the model does not fully capture the data's underlying patterns [23].

Residual plots and goodness-of-fit metrics were computed only for models that demonstrated the best fit to the data. Detailed discussions of these metrics and their implications are provided in the Discussions section.

CHAPTER 4

Results

The main objective of this study is to evaluate and compare the performance of the Broken Stick Model (BSM) and the Polynomial Model in describing the deswelling behaviour of hydrogels. To achieve this, the experimental data from various tests were analysed, and the results are systematically presented in this section. The findings are organised based on the research question of model fitting accuracy, followed by a detailed presentation of the fitting parameters and residuals for each test.

4.1 Broken Stick Model Analysis

The Broken Stick Model was applied to the experimental data from several tests to assess its suitability for representing hydrogel deswelling dynamics. The results of the fitting for each test are summarised below.

4.1.1 Pilot Test

Figure 4.1 presents the overall fit of the Broken Stick Model to the data from the Pilot Test. The model demonstrates a strong fit, with a high level of agreement between the predicted and observed data points. To provide more insight into the fitting accuracy, a zoomed-in view of the model in a specific time range is shown in Figure 4.2, highlighting the precision of the model in this interval.

Residuals for the Pilot Test are shown in Figure 4.3, where small differences between the model's predictions and the actual data can be observed, confirming the accuracy of the model. The model parameters for the Pilot Test, including the quadratic and linear coefficients, are summarised in Table 4.1. These parameters were optimised to achieve the best fit. Additionally, Table 4.2 presents the goodness-of-fit metrics, including R^2, SSE, RMSE, and MAE. The model achieves an R^2 value of 0.9994, indicating an excellent fit to the data.

4.1.2 Test T2

The fitting results for Test T2 are illustrated in Figure 4.4. The model accurately captures the overall deswelling behaviour. Figure 4.5 provides a closer look at the fitting in a specific time

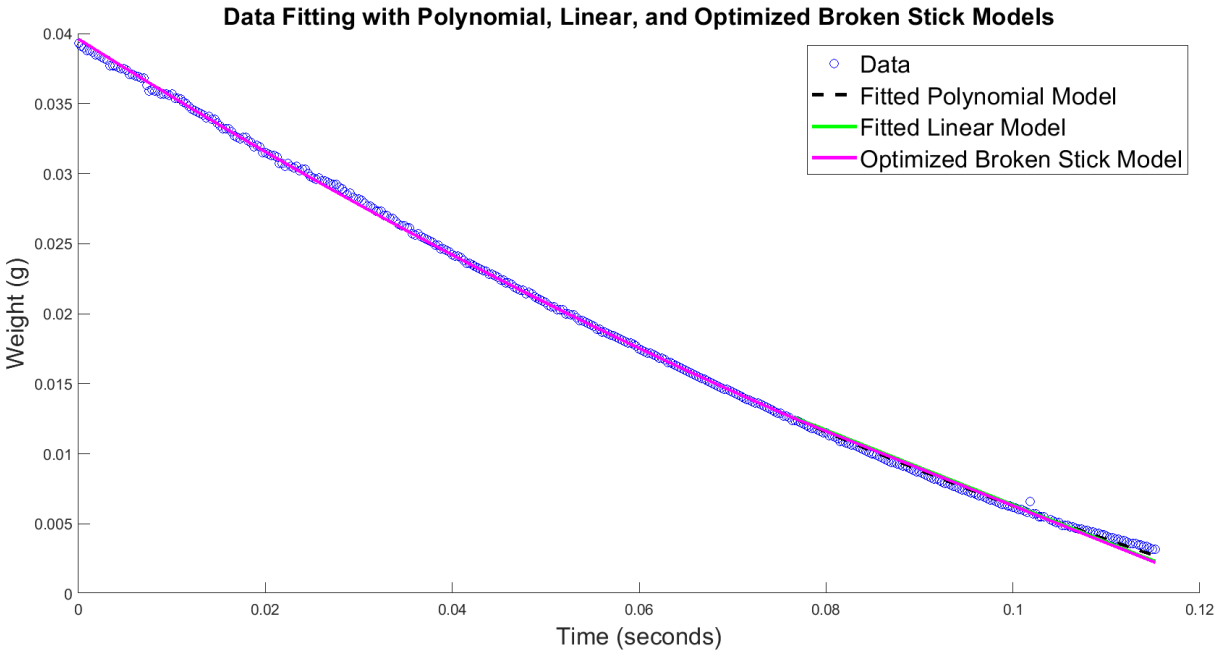

Figure 4.1: General Broken Stick Model (BSM) plot for the Pilot Test. This plot illustrates the overall fitting of the Broken Stick Model to the experimental data.

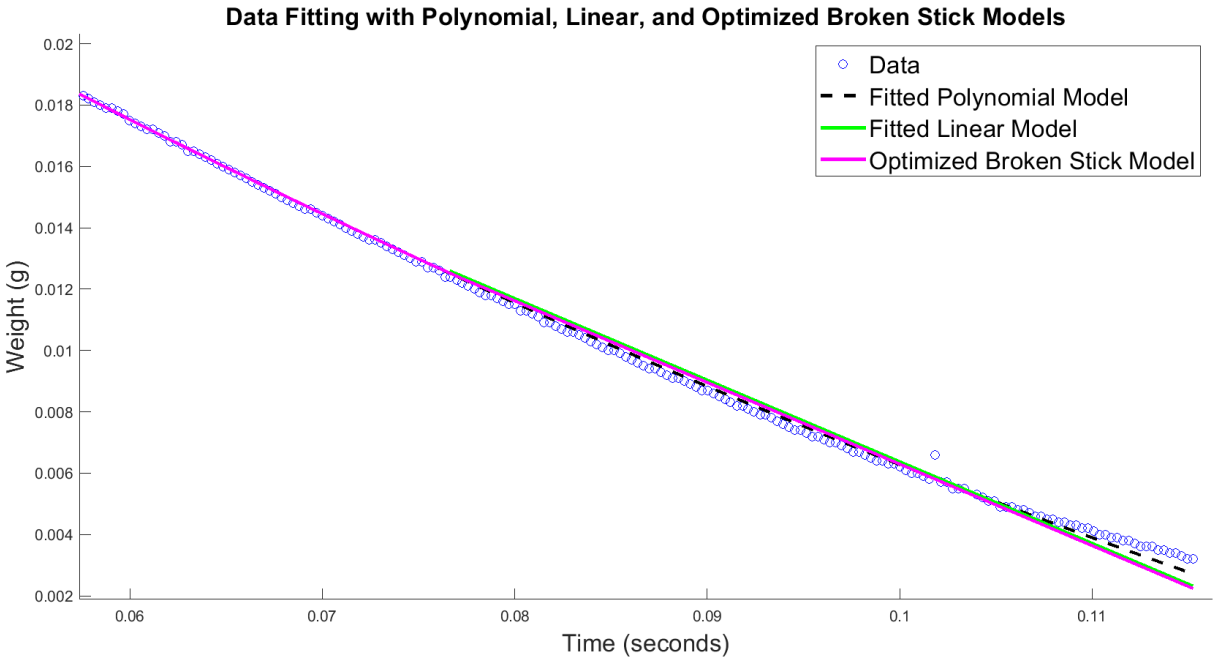

Figure 4.2: Zoomed-in view of the Broken Stick Model (BSM) plot for the Pilot Test, highlighting the details of the fit in a specific time range.

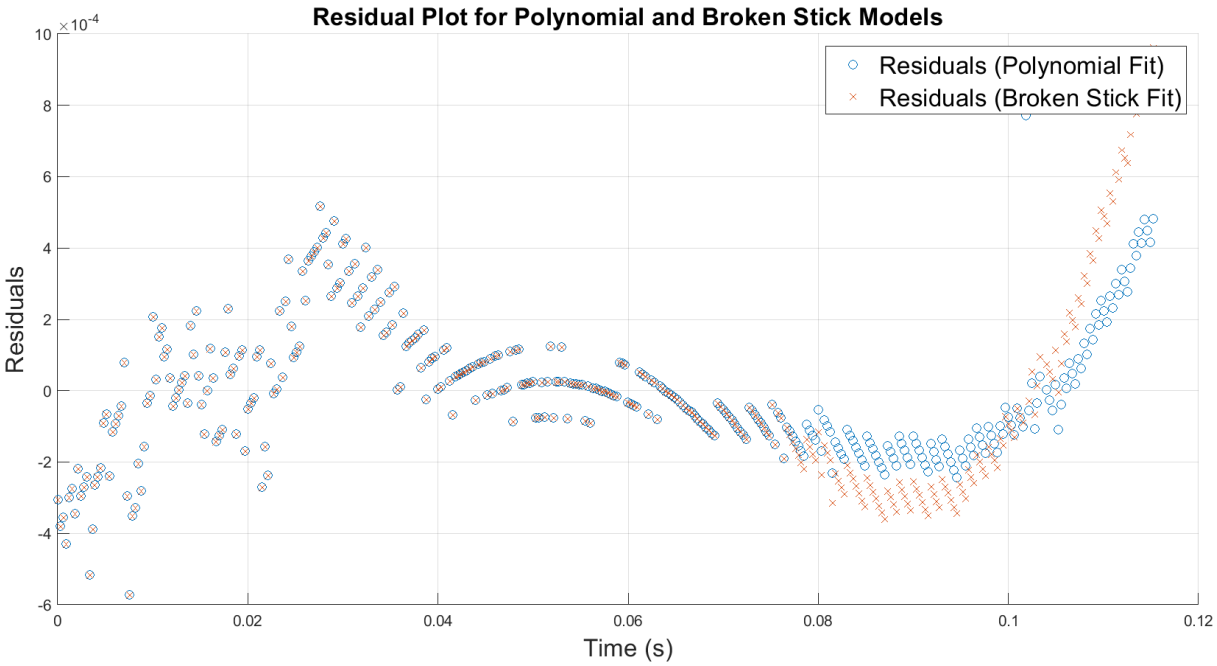

Figure 4.3: Residuals for the Broken Stick Model (BSM) fit for the Pilot Test. The plot shows the difference between the observed data and the model's predictions.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	0.86613
	a_1 (Linear Coefficient)	-0.41991
	a_0 (Intercept)	0.039606
Linear Model (After t_c)	a_1 (Slope)	-0.26613
	a_0 (Intercept)	0.032988
Piecewise Model	t_c (Optimized Critical Time)	0.076615

Table 4.1: Broken Stick Model Parameters for Pilot Test

Metric	**Value**
SSE (Broken Stick)	0.0000
RMSE (Broken Stick)	0.0003
MAE (Broken Stick)	0.0002
R^2 (Broken Stick)	0.9994

Table 4.2: Goodness of Fit for Broken Stick Model (Pilot Test)

range, showing a precise match between the model and experimental data.

Residuals for Test T2 are displayed in Figure 4.6, showing small deviations from the observed data, further validating the model's reliability. The model parameters, as optimised for Test T2, are presented in Table 4.3. These values reflect the model's adjustment to the data. Table 4.4 summarises the goodness-of-fit metrics, with an R^2 of 0.9993, confirming the model's strong performance.

Figure 4.7 illustrates the outliers that were removed during the analysis, enhancing the model's precision by excluding data points that significantly deviated from the general trend.

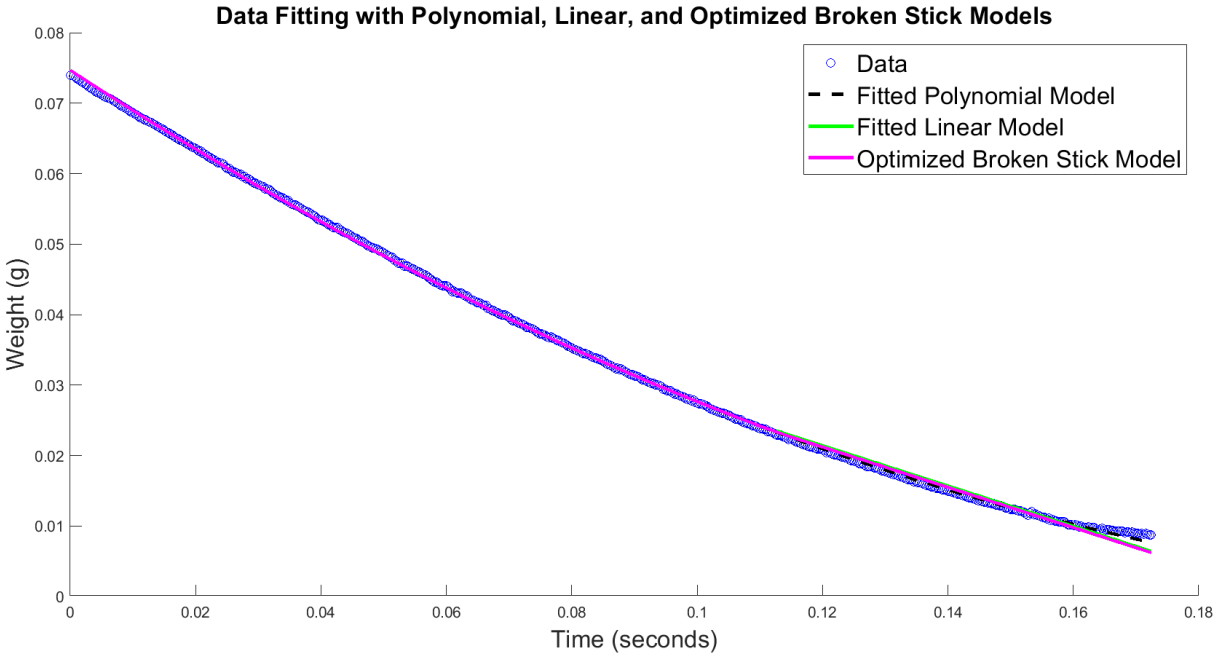

Figure 4.4: General Broken Stick Model (BSM) plot for Test T2. The plot shows the overall fitting of the model to the data from Test T2.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	1.1277
	a_1 (Linear Coefficient)	-0.58311
	a_0 (Intercept)	0.074672
Linear Model (After t_c)	a_1 (Slope)	-0.28577
	a_0 (Intercept)	0.055703
Piecewise Model	t_c (Optimized Critical Time)	0.11259

Table 4.3: Broken Stick Model Parameters for Test 2

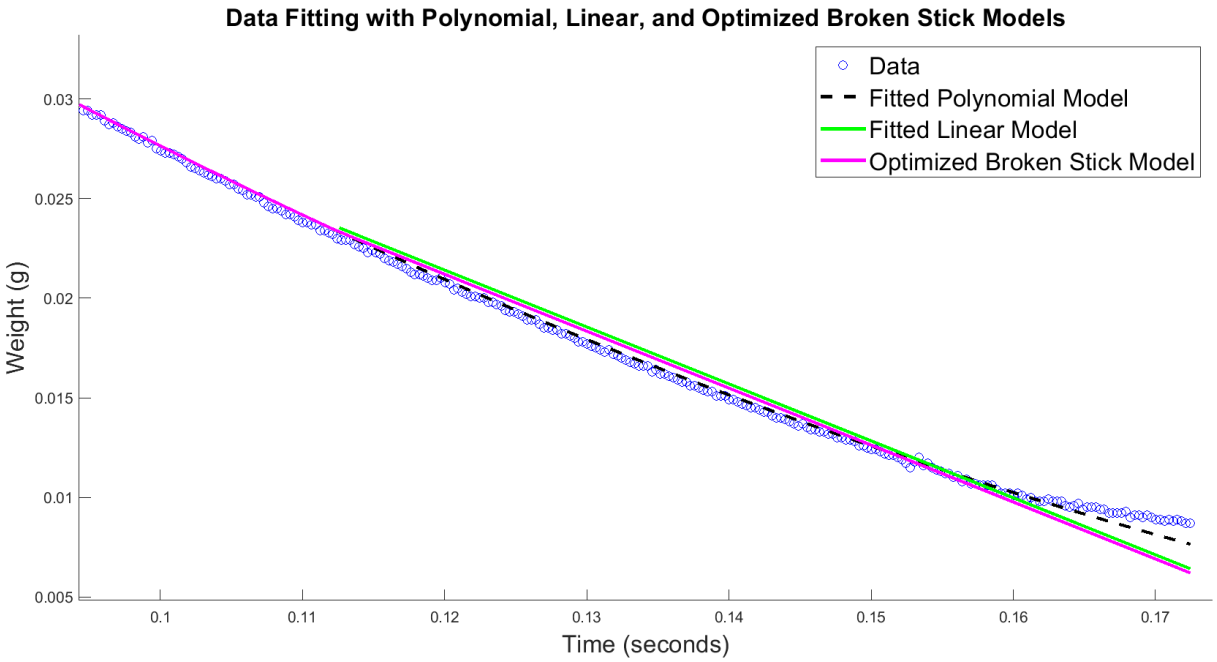

Figure 4.5: Zoomed-in view of the Broken Stick Model (BSM) plot for Test T2, focusing on a specific time range.

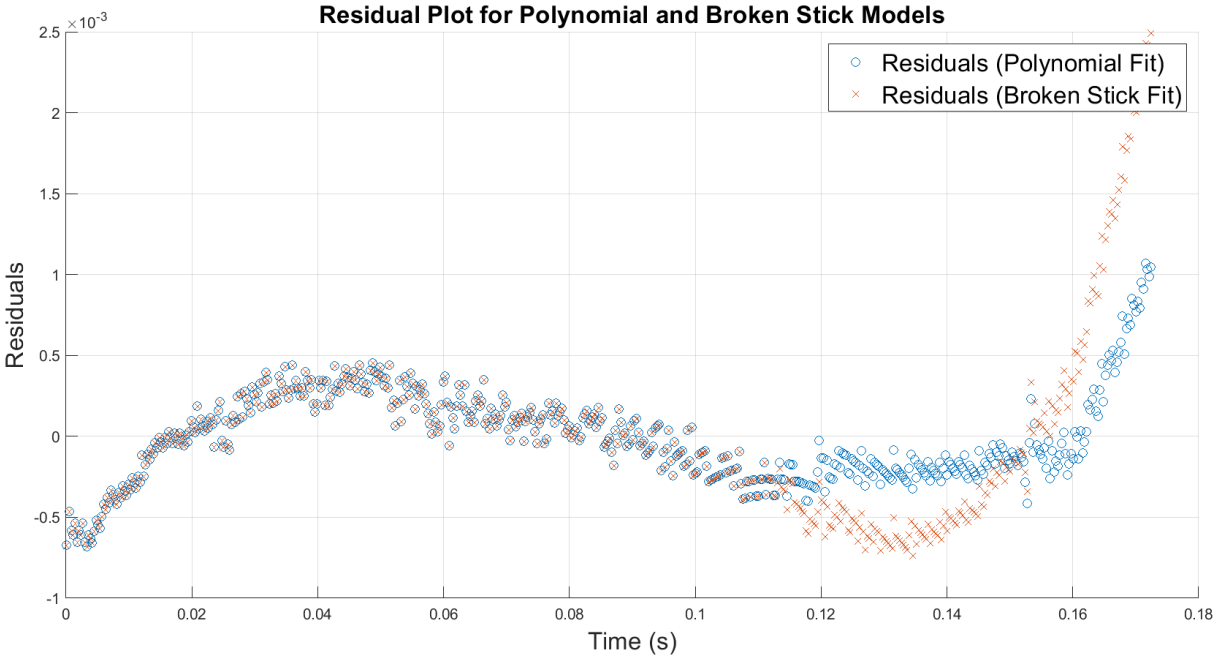

Figure 4.6: Residuals for the Broken Stick Model (BSM) fit for Test T2. The residuals are the differences between the observed data and the model predictions.

Metric	Value
SSE (Broken Stick)	0.0002
RMSE (Broken Stick)	0.0005
MAE (Broken Stick)	0.0004
R^2 (Broken Stick)	0.9993

Table 4.4: Goodness of Fit for Broken Stick Model (Test 2)

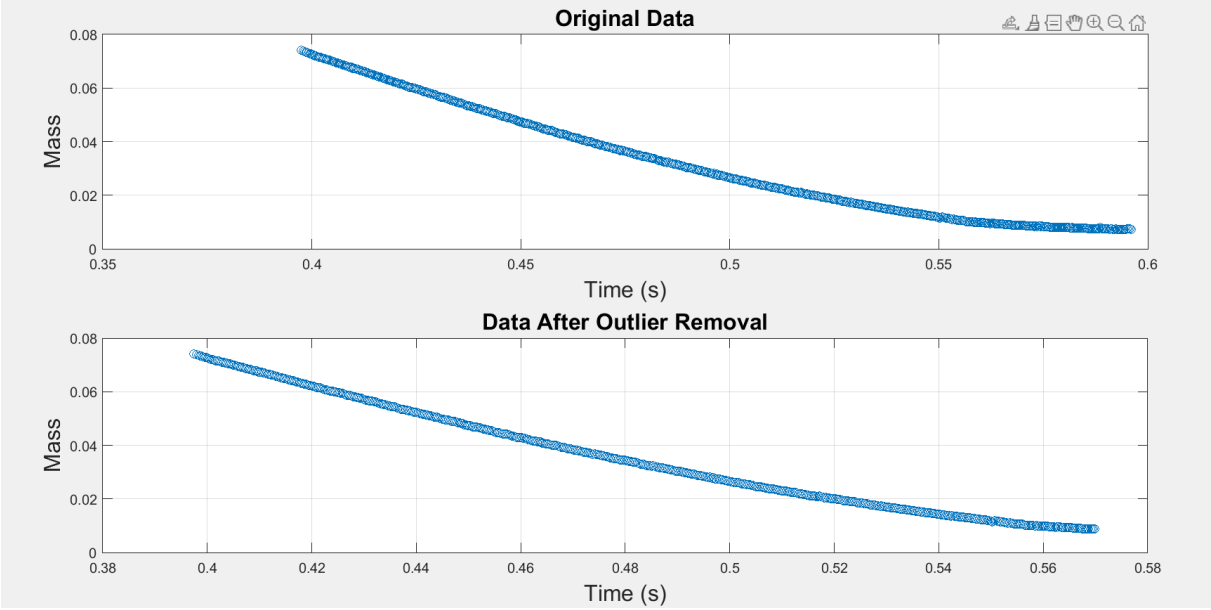

Figure 4.7: Outlier removal for Test T2. The plot illustrates the data points removed based on the outlier detection method.

4.1.3 Test T3

For Test T3, the overall fitting of the Broken Stick Model is shown in Figure 4.8, with the model displaying a good match with the experimental data. A zoomed-in view of a specific time range, shown in Figure 4.9, highlights the precision of the model in capturing subtle variations in the data.

Residuals for Test T3 are presented in Figure 4.10, where minimal discrepancies between the model and the actual data can be seen. The model parameters for Test T3, shown in Table 4.5, were fine-tuned for accuracy. Table 4.6 presents the goodness-of-fit metrics, with an impressive R^2 value of 0.9998, indicating an almost perfect fit.

Figure 4.11 illustrates the outliers identified and removed from the analysis, further improving the model's fit.

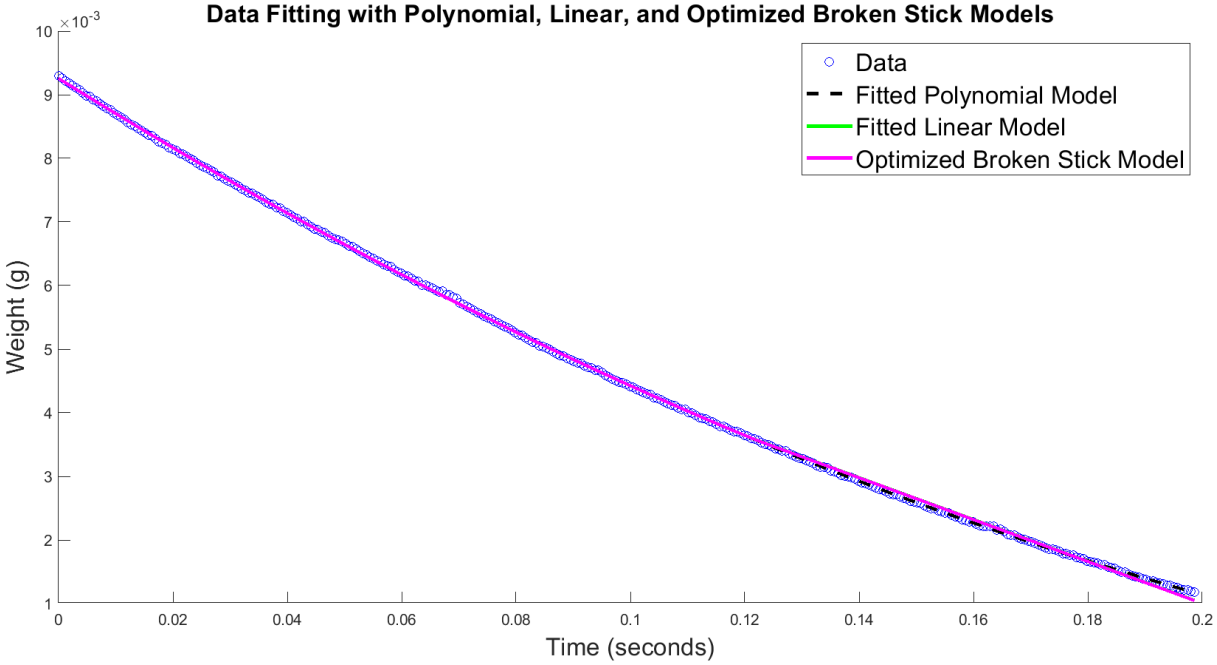

Figure 4.8: General Broken Stick Model (BSM) plot for Test T3. This plot shows the fitting of the model to the data from Test T3.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	0.077303
	a_1 (Linear Coefficient)	-0.056043
	a_0 (Intercept)	0.0092557
Linear Model (After t_c)	a_1 (Slope)	-0.032919
	a_0 (Intercept)	0.0075899
Piecewise Model	t_c (Optimized Critical Time)	0.12151

Table 4.5: Broken Stick Model Parameters for Test 3

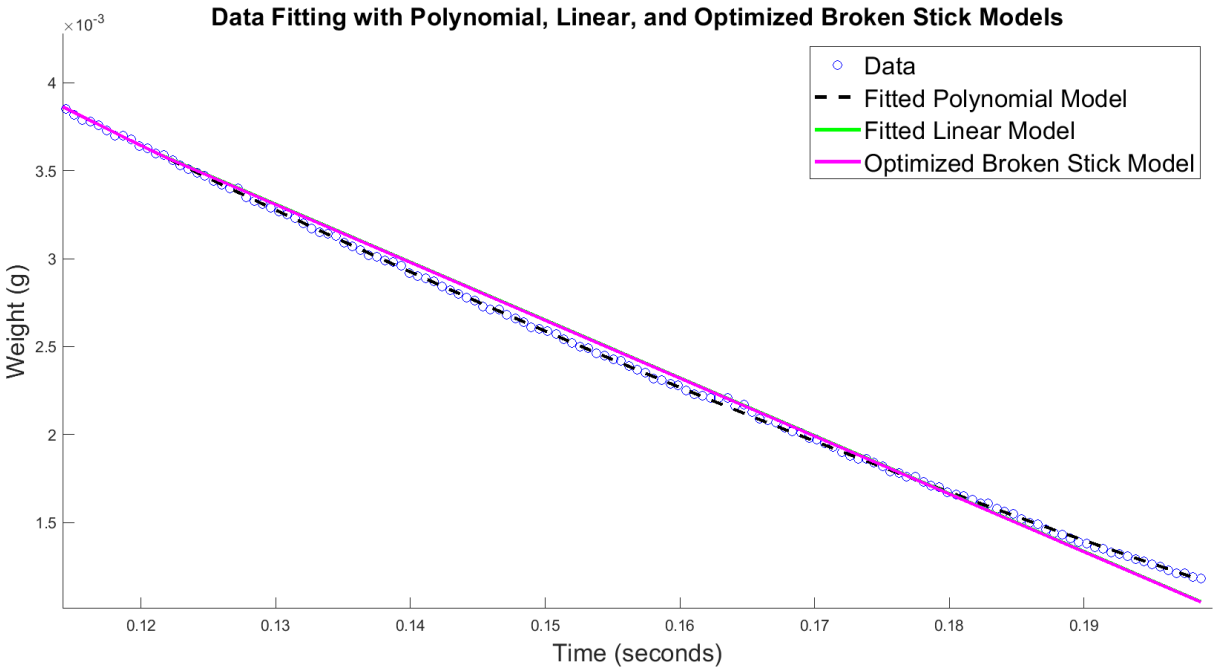

Figure 4.9: Zoomed-in view of the Broken Stick Model (BSM) plot for Test T3, focusing on a specific time range.

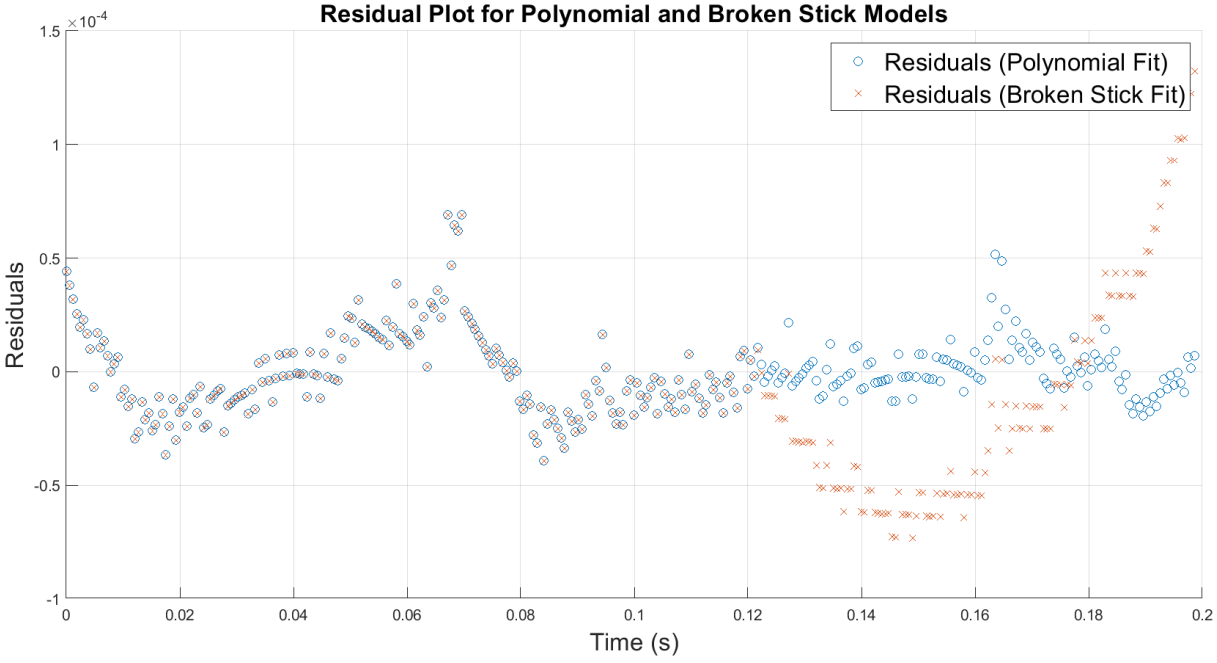

Figure 4.10: Residuals for the Broken Stick Model (BSM) fit for Test T3. The plot shows the discrepancies between the observed data and the model's predictions.

Metric	Value
SSE (Broken Stick)	0.0000
RMSE (Broken Stick)	0.0000
MAE (Broken Stick)	0.0000
R^2 (Broken Stick)	0.9998

Table 4.6: Goodness of Fit for Broken Stick Model (Test 3)

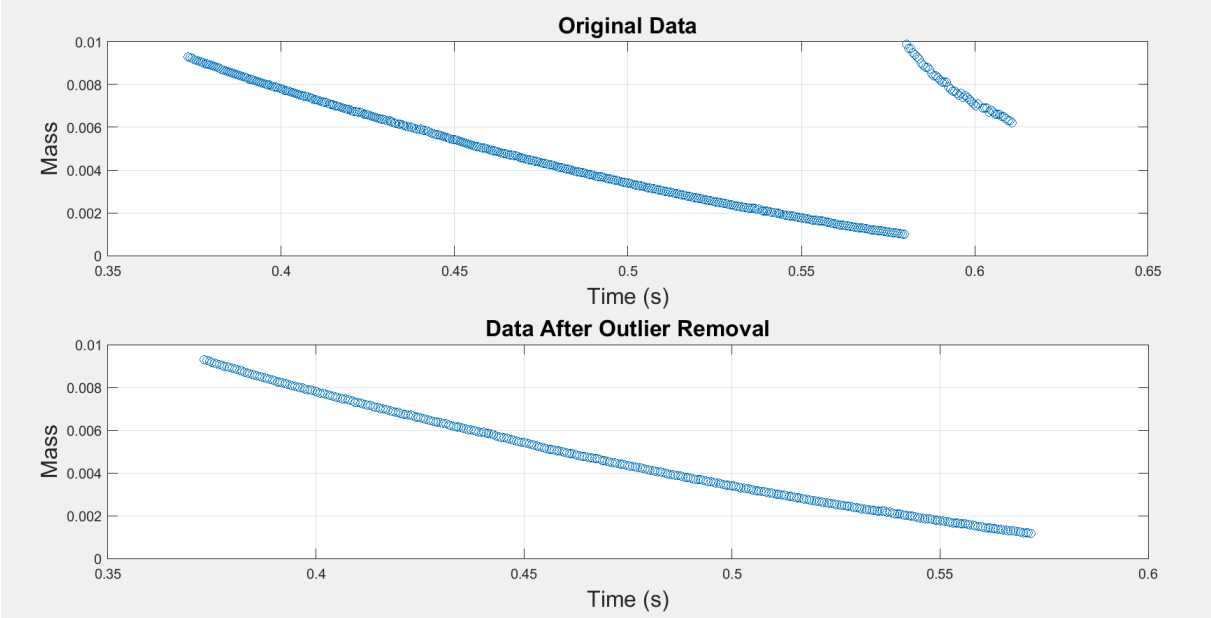

Figure 4.11: Outlier removal for Test T3. This plot shows the data points excluded due to outlier detection.

4.1.4 Test T4

Figure 4.12 shows the fit of the Broken Stick Model for Test T4. The model captures the overall trend effectively. A zoomed-in view of a specific section of the data is provided in Figure 4.13, illustrating the model's high accuracy in this range.

Residuals for Test T4 are shown in Figure 4.14. The small residuals suggest that the model performs well in representing the data. Table 4.7 lists the optimised parameters for Test T4, and Table 4.8 shows the corresponding goodness-of-fit metrics, with an R^2 value of 0.9993, demonstrating a very strong fit.

Figure 4.15 shows the outliers removed during the fitting process, enhancing the accuracy of the results.

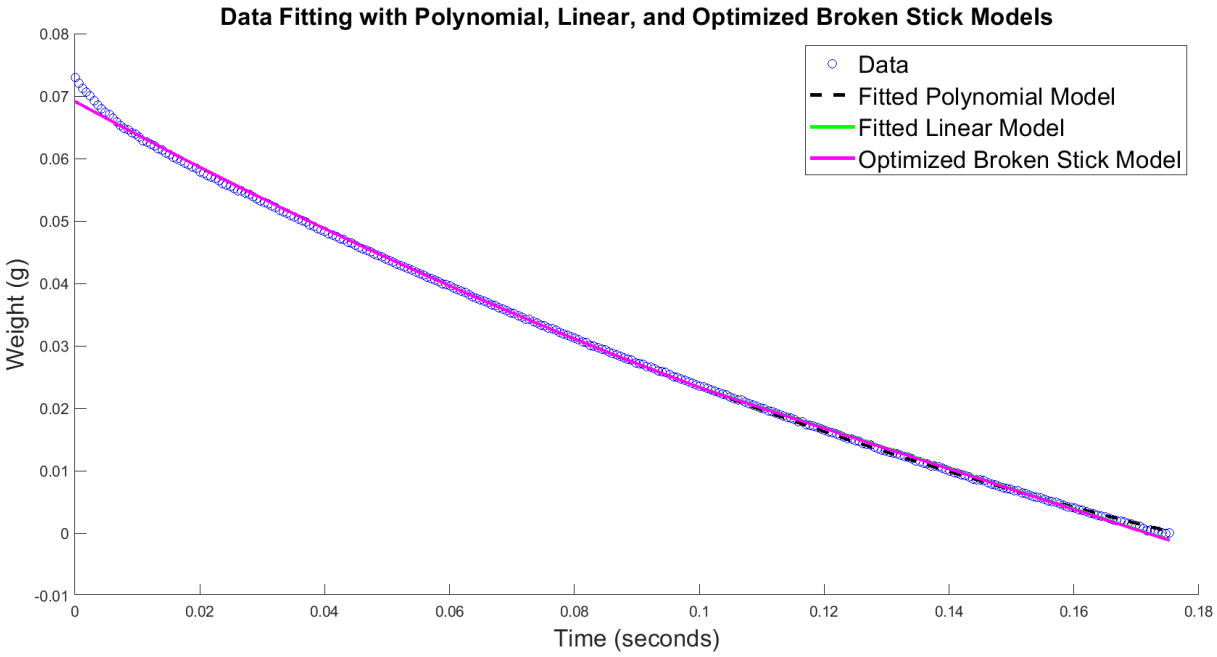

Figure 4.12: General Broken Stick Model (BSM) plot for Test T4. The plot illustrates the overall fitting of the model to the data from Test T4.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	0.86122
	a_1 (Linear Coefficient)	-0.5439
	a_0 (Intercept)	0.06914
Linear Model (After t_c)	a_1 (Slope)	-0.3246
	a_0 (Intercept)	0.055836
Piecewise Model	t_c (Optimized Critical Time)	0.10161

Table 4.7: Broken Stick Model Parameters for Test 4

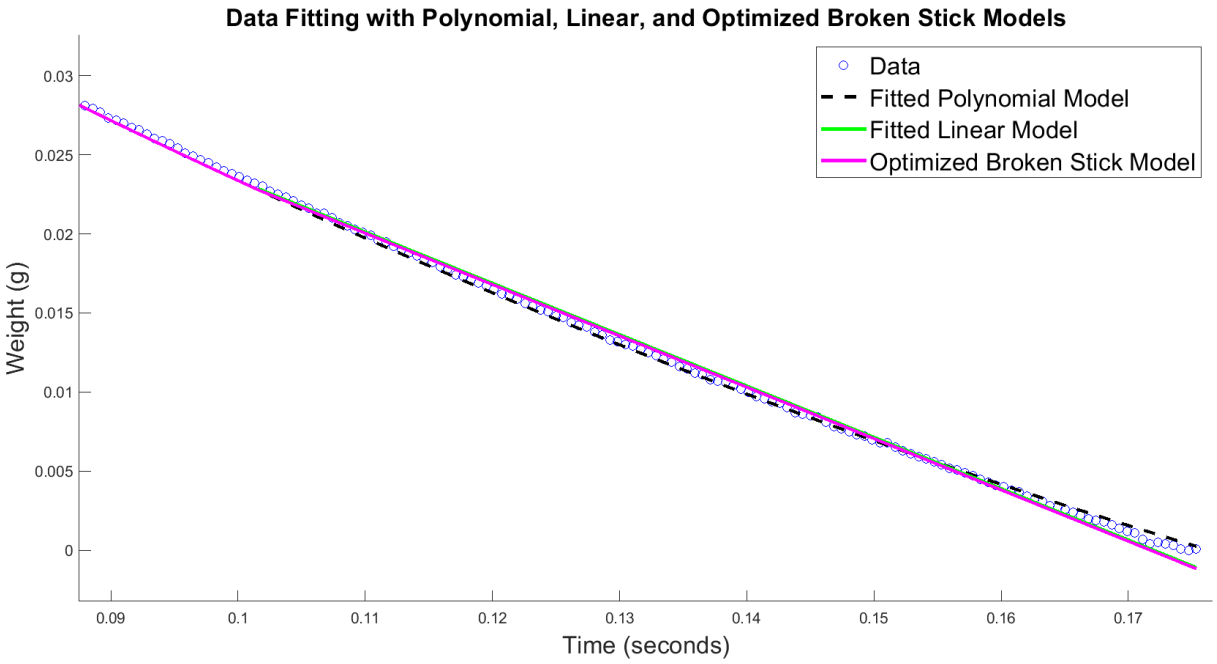

Figure 4.13: Zoomed-in view of the Broken Stick Model (BSM) plot for Test T4, showing details of the fit within a particular time range.

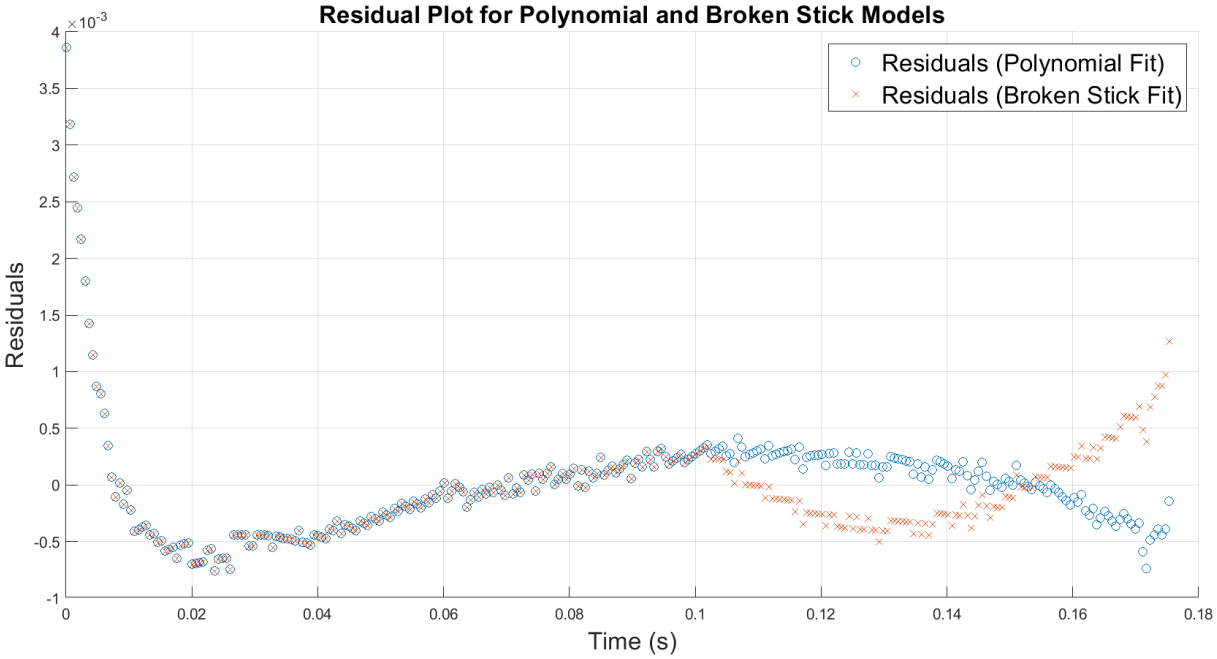

Figure 4.14: Residuals for the Broken Stick Model (BSM) fit for Test T4. This plot depicts the differences between observed data and model predictions.

Metric	Value
SSE (Broken Stick)	0.0001
RMSE (Broken Stick)	0.0005
MAE (Broken Stick)	0.0003
R^2 (Broken Stick)	0.9993

Table 4.8: Goodness of Fit for Broken Stick Model (Test 4)

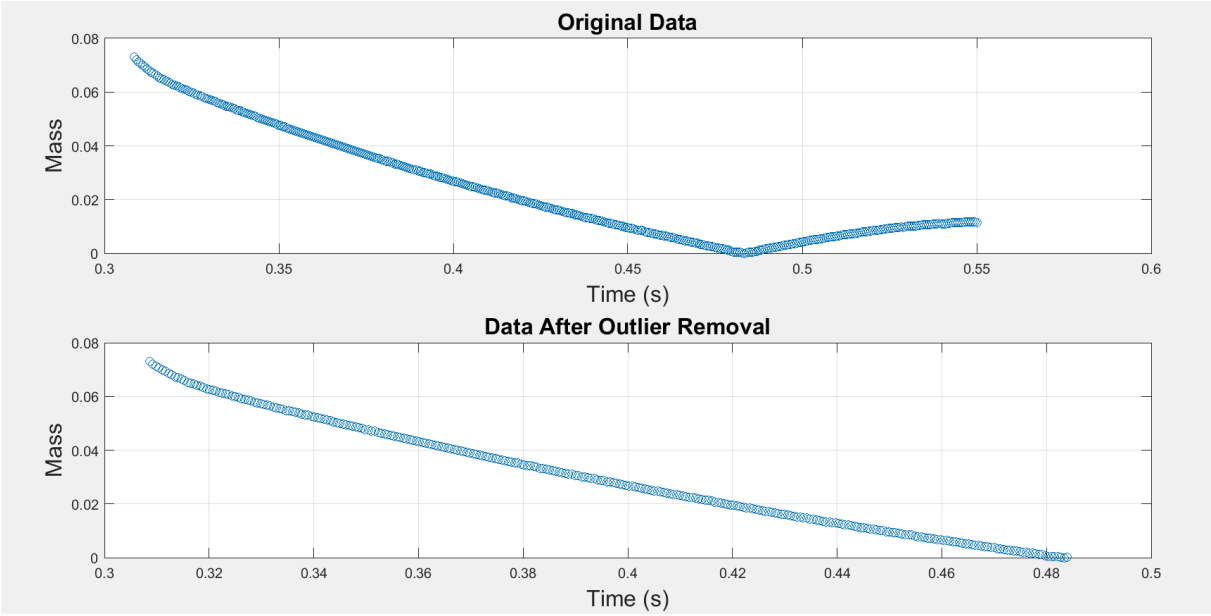

Figure 4.15: Outlier removal for Test T4. This plot illustrates the removal of data points identified as outliers.

4.1.5 Test T5

Figure 4.16 illustrates the overall fit of the Broken Stick Model to the data from Test T5. The model captures the general drying behaviour of the material, showing a strong correlation between the experimental data and the predicted values.

A more detailed view of the model fit within a specific time range is provided in Figure 4.17. This zoomed-in section highlights the model's precision in describing the dynamics of the deswelling process during this period.

Residuals for Test T5 are shown in Figure 4.18, presenting the differences between the observed data and the model's predictions. The residuals remain small, indicating that the model accurately follows the experimental data.

Table 4.9 summarises the optimised model parameters for Test T5, including the quadratic and linear coefficients, as well as the critical time t_c. These parameters have been adjusted to provide the best possible fit.

Table 4.10 presents the goodness-of-fit metrics for Test T5. The model achieves an R^2 value of 0.9980, indicating a very good fit to the experimental data, with low values for SSE, RMSE, and MAE.

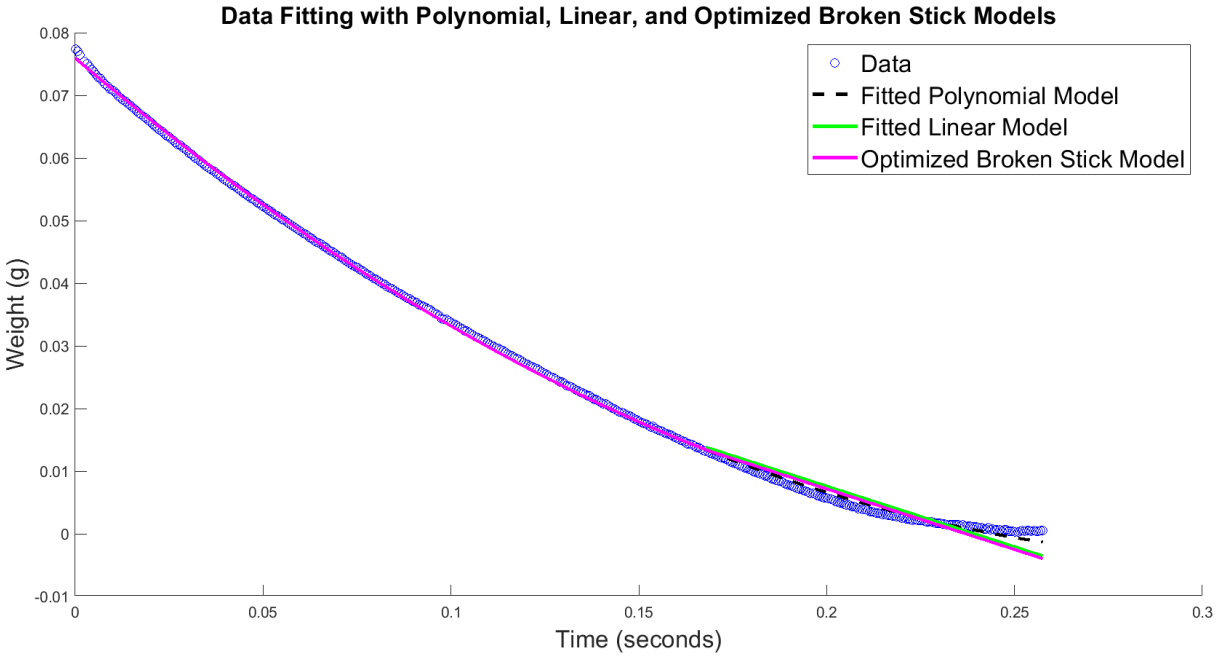

Figure 4.16: General plot for Test T5 showing the overall drying behaviour.

4.1.6 Test T6

Figure 4.19 shows the overall fit of the Broken Stick Model for Test T6. Similar to previous tests, the model accurately represents the drying behaviour of the material, closely following the experimental data throughout the entire time range.

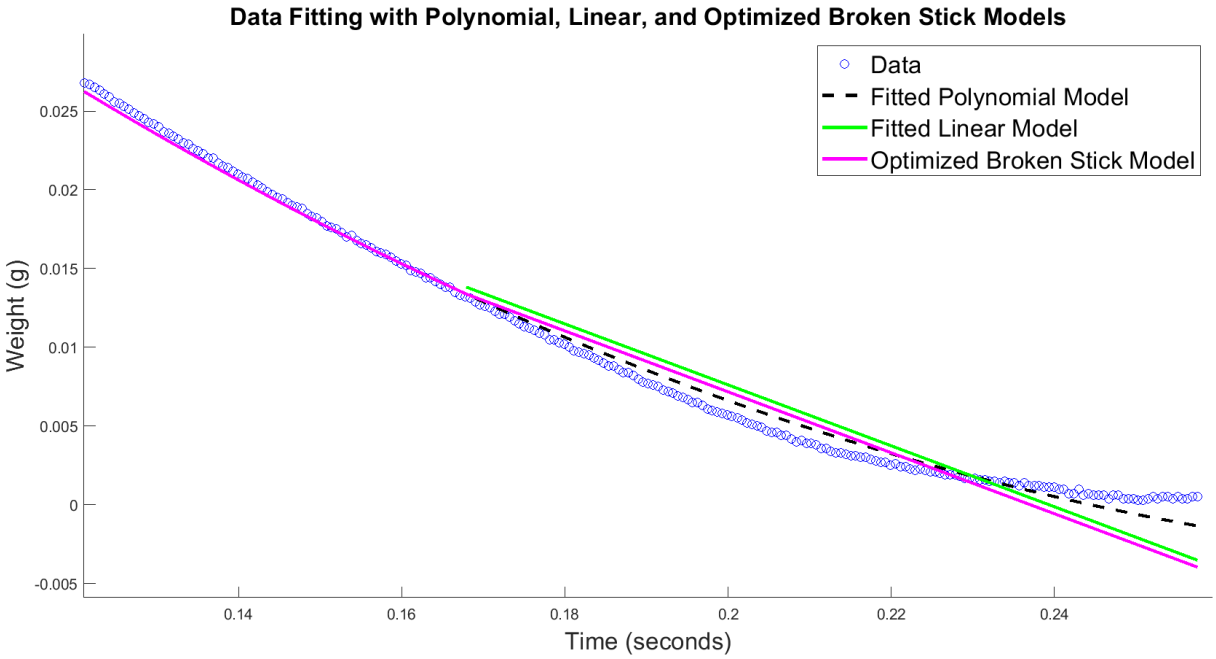

Figure 4.17: Zoomed-in plot for Test T5 highlighting specific details of the drying process.

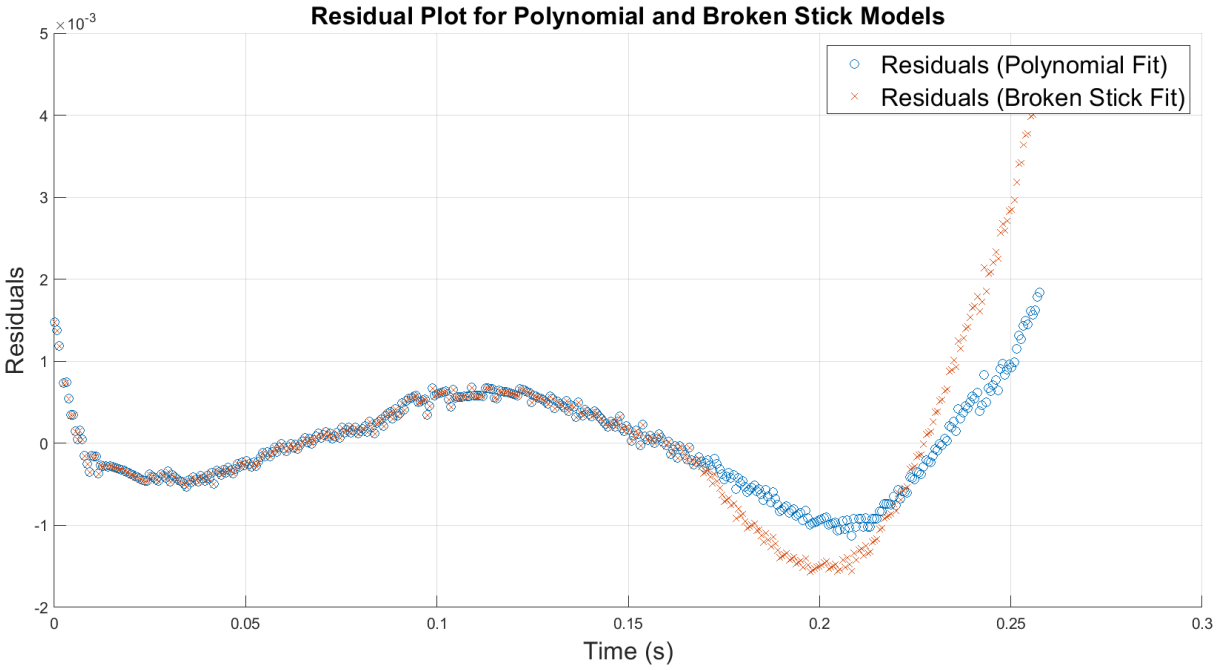

Figure 4.18: Residuals for the Broken Stick Model (BSM) fit for Test T5. This plot depicts the differences between observed data and model predictions.

Model	Parameter	Value
Polynomial Model	a_2 (Quadratic Coefficient)	0.80859
	a_1 (Linear Coefficient)	-0.50826
	a_0 (Intercept)	0.075926
Linear Model (After t_c)	a_1 (Slope)	-0.19352
	a_0 (Intercept)	0.046317
Piecewise Model	t_c (Optimized Critical Time)	0.1679

Table 4.9: Broken Stick Model Parameters for Test 5

Metric	Value
SSE (Broken Stick)	0.0004
RMSE (Broken Stick)	0.0010
MAE (Broken Stick)	0.0007
R^2 (Broken Stick)	0.9980

Table 4.10: Goodness of Fit for Broken Stick Model (Test 5)

A zoomed-in view of the model fit is shown in Figure 4.20, focusing on a narrower time range. This detailed view demonstrates the model's ability to accurately capture smaller variations in the data during this interval.

Figure 4.21 presents the residuals for the Broken Stick Model in Test T6. The residuals are small, showing that the model closely predicts the experimental data.

Table 4.11 lists the parameters for the Broken Stick Model in Test T6. The coefficients and critical time were optimised to ensure the best fit to the data.

Table 4.12 summarises the goodness-of-fit metrics for Test T6. The model achieves an R^2 of 0.9989, reflecting a very high level of accuracy, with low SSE, RMSE, and MAE values.

Model	Parameter	Value
Polynomial Model	a_2 (Quadratic Coefficient)	0.67403
	a_1 (Linear Coefficient)	-0.52043
	a_0 (Intercept)	0.086772
Linear Model (After t_c)	a_1 (Slope)	-0.28583
	a_0 (Intercept)	0.067361
Piecewise Model	t_c (Optimized Critical Time)	0.13798

Table 4.11: Broken Stick Model Parameters for Test 6

4.2 Polynomial Model Analysis

The Polynomial Model (second degree) was also applied to the data, and its performance was evaluated for each test. The findings for the Polynomial Model are summarised as follows.

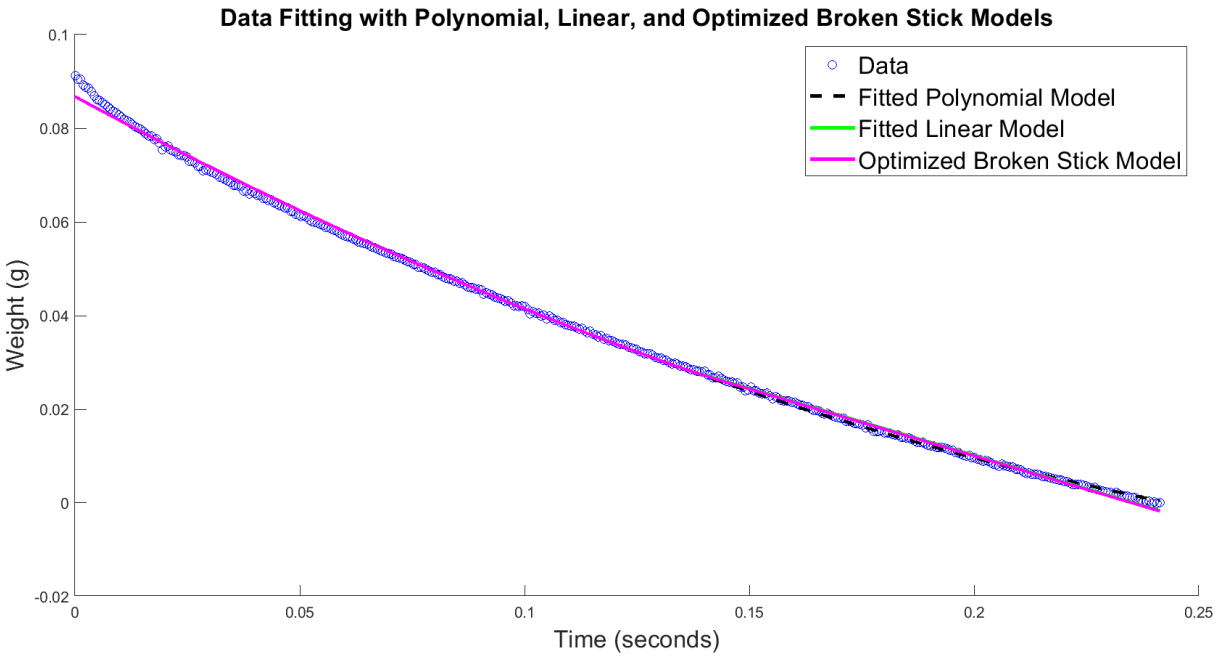

Figure 4.19: General plot for Test T6 showing the overall drying behaviour.

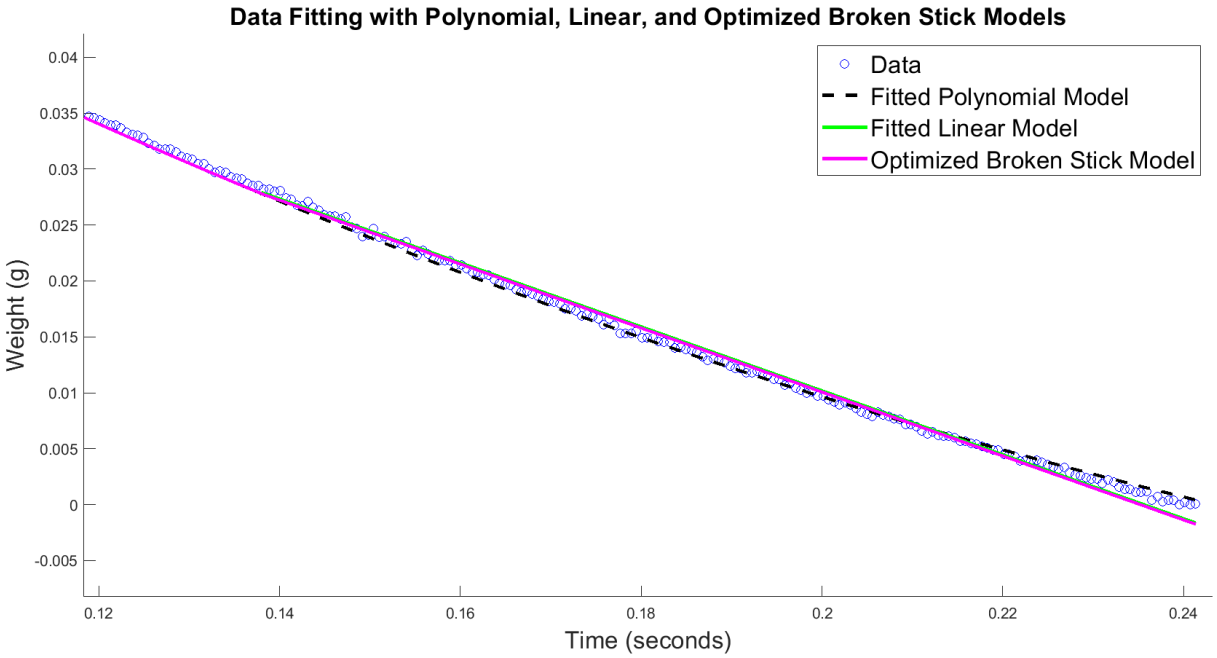

Figure 4.20: Zoomed-in plot for Test T6 highlighting specific details of the drying process.

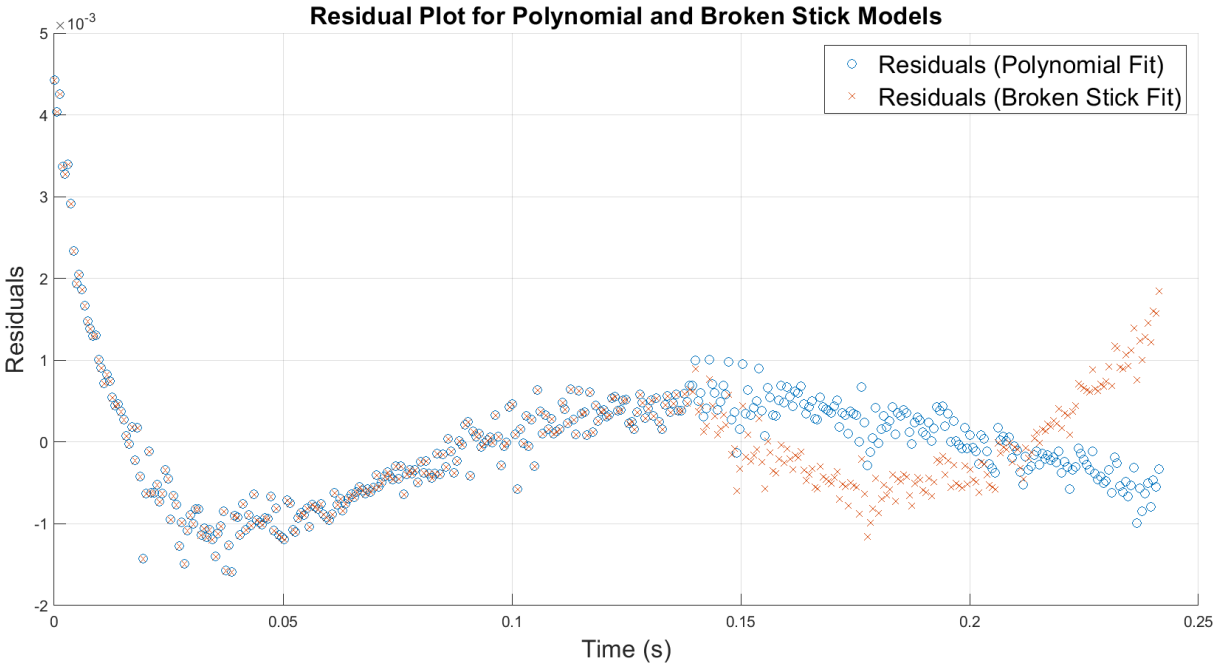

Figure 4.21: Residuals for the Broken Stick Model (BSM) fit for Test T6. This plot depicts the differences between observed data and model predictions.

Metric	**Value**
SSE (Broken Stick)	0.0003
RMSE (Broken Stick)	0.0008
MAE (Broken Stick)	0.0006
R^2 (Broken Stick)	0.9989

Table 4.12: Goodness of Fit for Broken Stick Model (Test 6)

4.2.1 Pilot Test

Figure 4.22 shows the overall fitting of the Polynomial Model to the Pilot Test data. The model accurately follows the experimental data, providing a close match.

Figure 4.23 displays the residuals of the Polynomial Model for the Pilot Test, where the small residuals indicate a strong fit. Table 4.13 presents the model parameters, and Table 4.14 provides the goodness-of-fit metrics. With an R^2 value of 0.9997, the Polynomial Model performs exceptionally well.

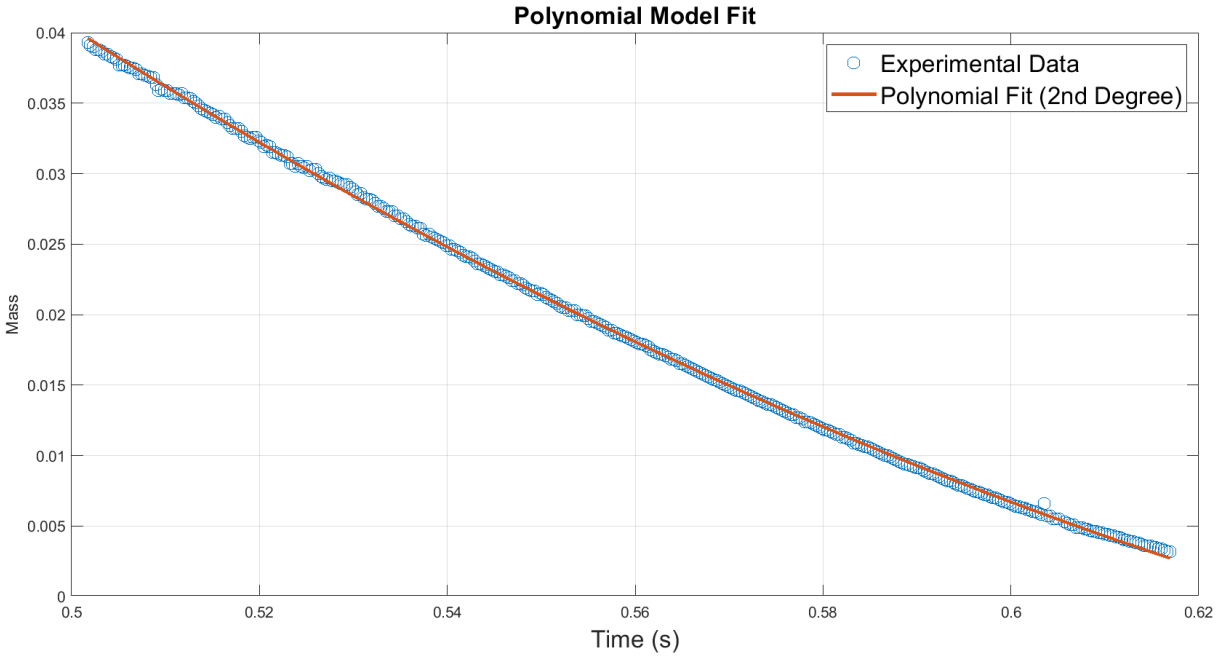

Figure 4.22: General Polynomial Model (PM) plot for the Pilot Test. This plot illustrates the overall fitting of the Polynomial Model to the experimental data.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	0.8661
	a_1 (Linear Coefficient)	-1.2890
	a_0 (Intercept)	0.4683

Table 4.13: Polynomial Model Parameters for Pilot Test

4.2.2 Test T2

For Test T2, the overall fit of the Polynomial Model is shown in Figure 4.24. The model accurately represents the deswelling behaviour, with small residuals as shown in Figure 4.25.

The parameters of the Polynomial Model for Test T2 are summarised in Table 4.15, and the goodness-of-fit metrics in Table 4.16 indicate an R^2 of 0.9998, reflecting a very accurate fit.

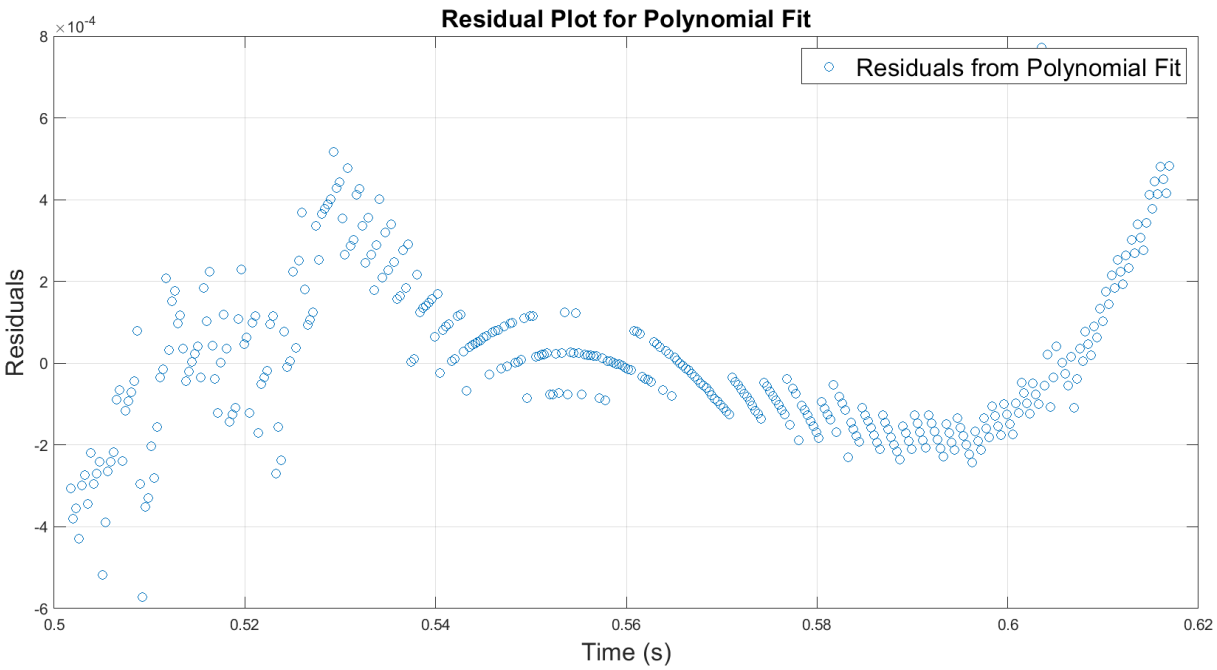

Figure 4.23: Residuals for the Polynomial Model (PM) fit for the Pilot Test. The plot shows the difference between the observed data and the model's predictions.

Metric	**Value**
SSE (Polynomial)	0.0000
RMSE (Polynomial)	0.0002
MAE (Polynomial)	0.0001
R^2 (Polynomial)	0.9997

Table 4.14: Goodness of Fit for Polynomial Model (Pilot Test)

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	1.1277
	a_1 (Linear Coefficient)	-1.4795
	a_0 (Intercept)	0.4846

Table 4.15: Polynomial Model Parameters for Test 2

Metric	**Value**
SSE (Polynomial)	0.0000
RMSE (Polynomial)	0.0003
MAE (Polynomial)	0.0002
R^2 (Polynomial)	0.9998

Table 4.16: Goodness of Fit for Polynomial Model (Test 2)

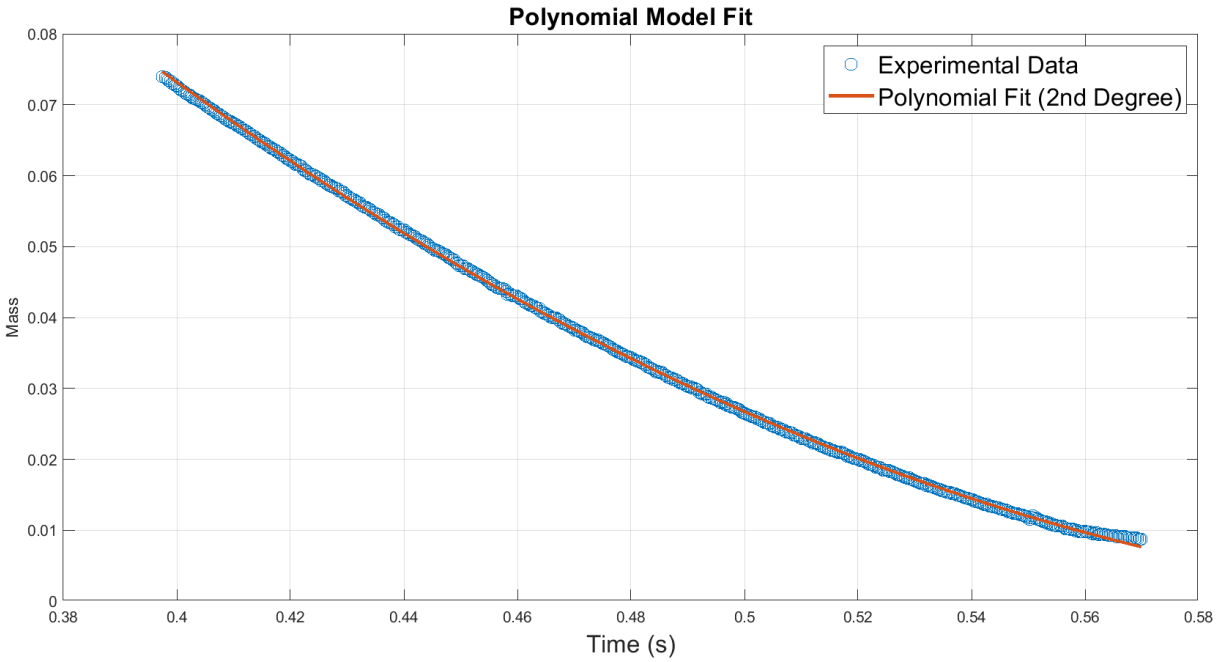

Figure 4.24: General Polynomial Model (PM) plot for Test T2. The plot shows the overall fitting of the model to the data from Test T2.

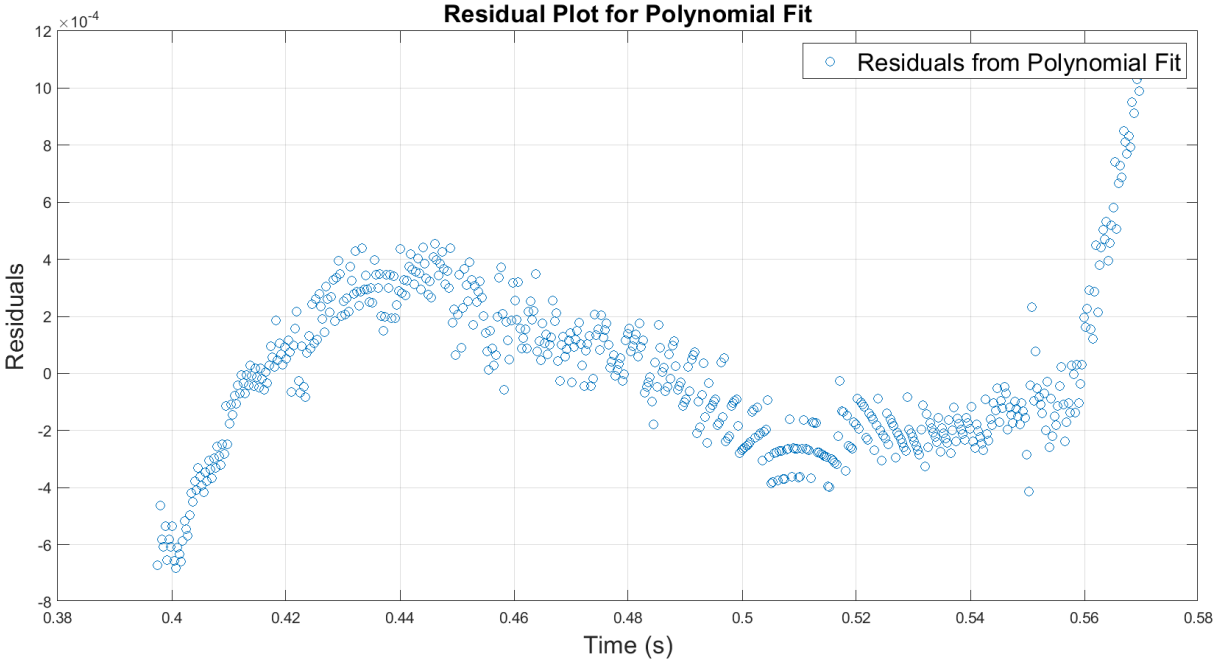

Figure 4.25: Residuals for the Polynomial Model (PM) fit for Test T2. The residuals are the differences between the observed data and the model predictions.

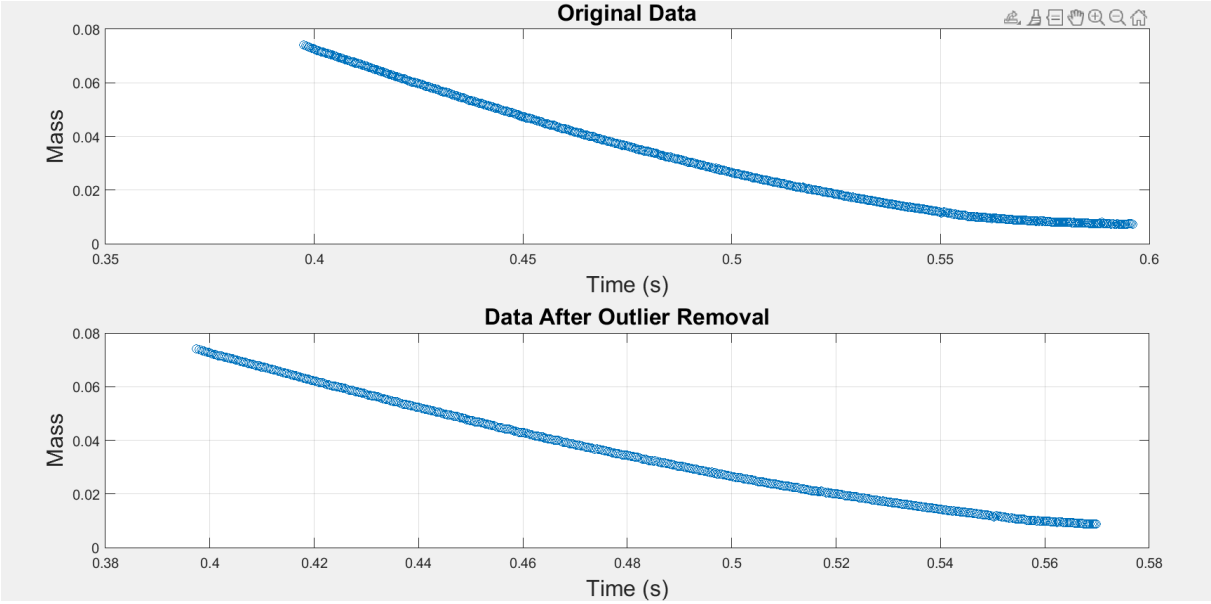

Figure 4.26: Outlier removal for Test T2. The plot illustrates the data points removed based on the outlier detection method.

4.2.3 Test T3

The fitting results for Test T3 are shown in Figure 4.27, with the Polynomial Model providing a good fit across the entire time range. Figure 4.28 shows the residuals, which remain small, indicating the model's accuracy.

Table 4.17 lists the optimised parameters for the Polynomial Model, while Table 4.18 presents the goodness-of-fit metrics, with an R^2 of 0.9999, showing a near-perfect fit.

Model	**Parameter**	**Value**
Polynomial Model	a_2 (Quadratic Coefficient)	0.0773
	a_1 (Linear Coefficient)	-0.1137
	a_0 (Intercept)	0.0409

Table 4.17: Polynomial Model Parameters for Test 3

4.2.4 Test T4

For Test T4, Figure 4.30 illustrates the fit of the Polynomial Model, where it captures the overall trend effectively. The residuals for Test T4, displayed in Figure 4.31, are minimal, indicating a good fit.

Table 4.19 shows the model parameters for Test T4, and Table 4.20 presents the goodness-of-fit metrics, with an R^2 of 0.9993, indicating strong performance.

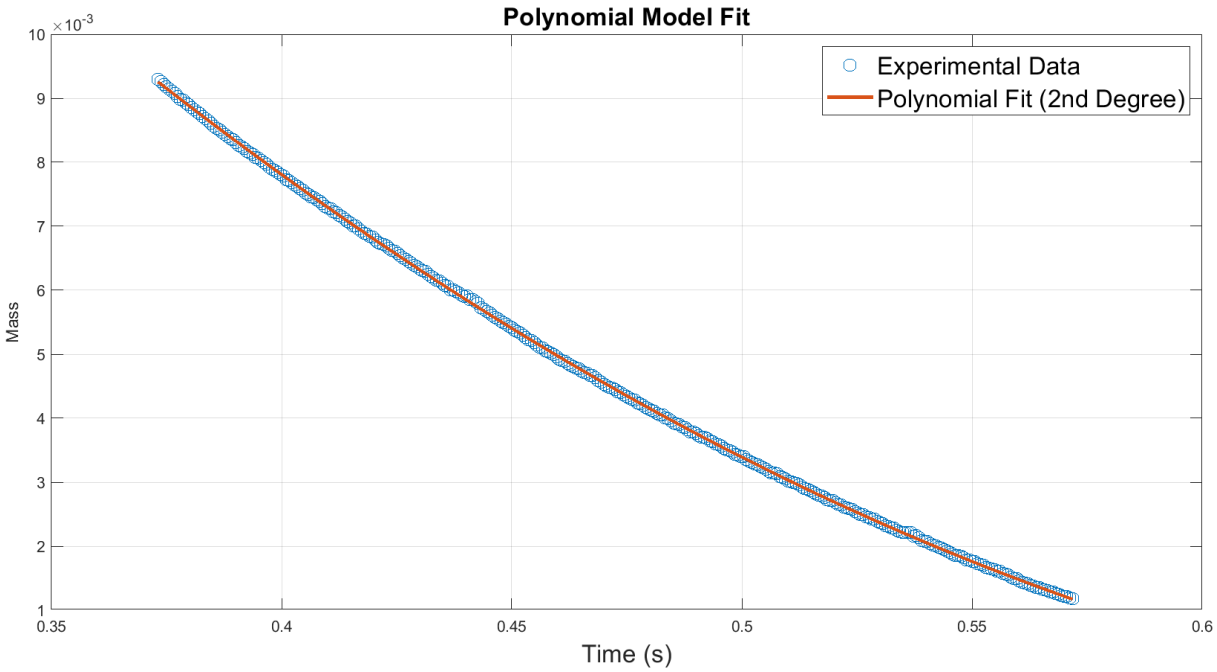

Figure 4.27: General Polynomial Model (PM) plot for Test T3. This plot shows the fitting of the model to the data from Test T3.

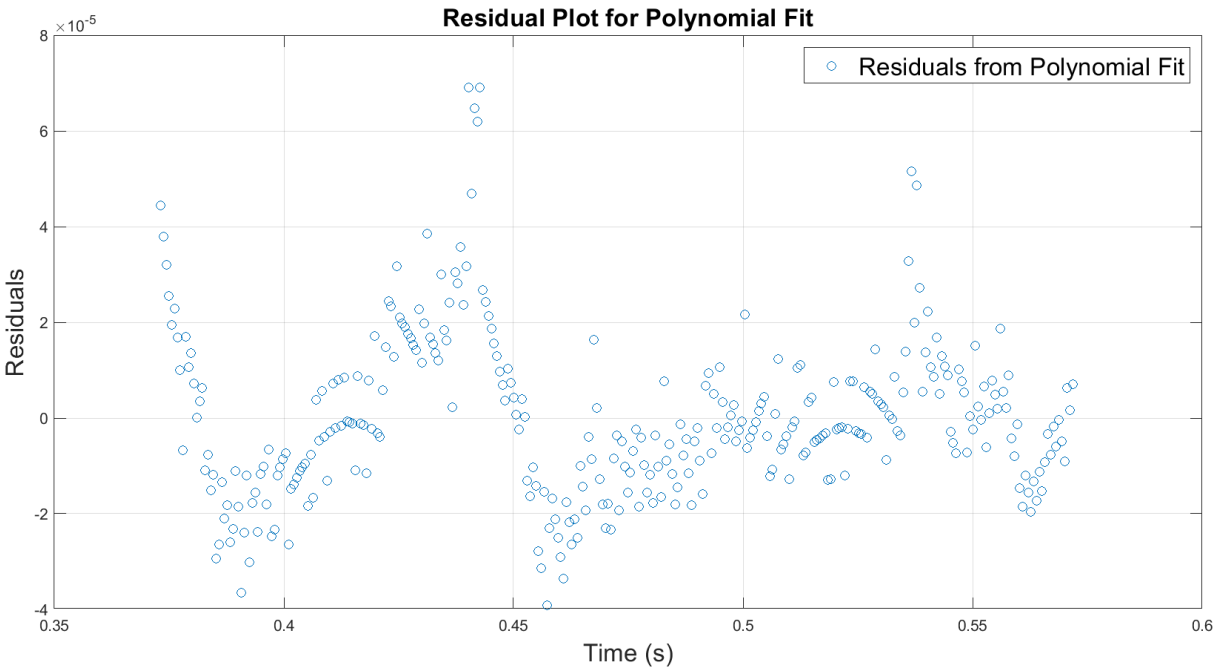

Figure 4.28: Residuals for the Polynomial Model (PM) fit for Test T3. The plot shows the discrepancies between the observed data and the model's predictions.

Metric	Value
SSE (Polynomial)	0.0000
RMSE (Polynomial)	0.0000
MAE (Polynomial)	0.0000
R^2 (Polynomial)	0.9999

Table 4.18: Goodness of Fit for Polynomial Model (Test 3)

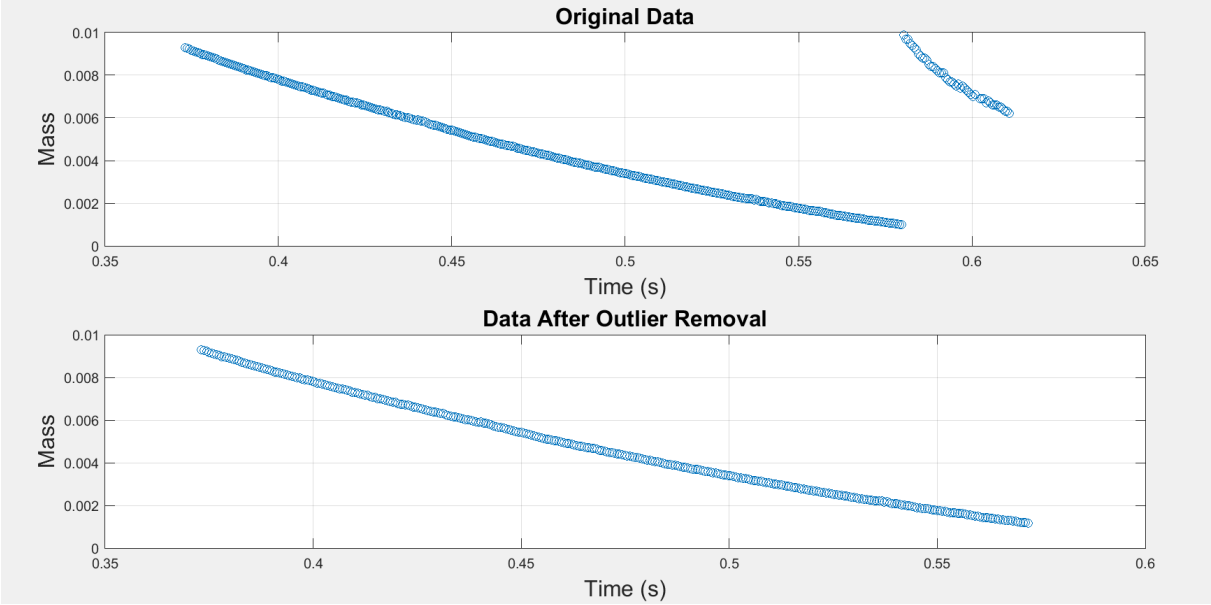

Figure 4.29: Outlier removal for Test T3. This plot shows the data points excluded due to outlier detection.

Model	Parameter	Value
Polynomial Model	a_2 (Quadratic Coefficient)	0.8612
	a_1 (Linear Coefficient)	-1.0755
	a_0 (Intercept)	0.3190

Table 4.19: Polynomial Model Parameters for Test 4

Metric	Value
SSE (Polynomial)	0.0001
RMSE (Polynomial)	0.0005
MAE (Polynomial)	0.0003
R^2 (Polynomial)	0.9993

Table 4.20: Goodness of Fit for Polynomial Model (Test 4)

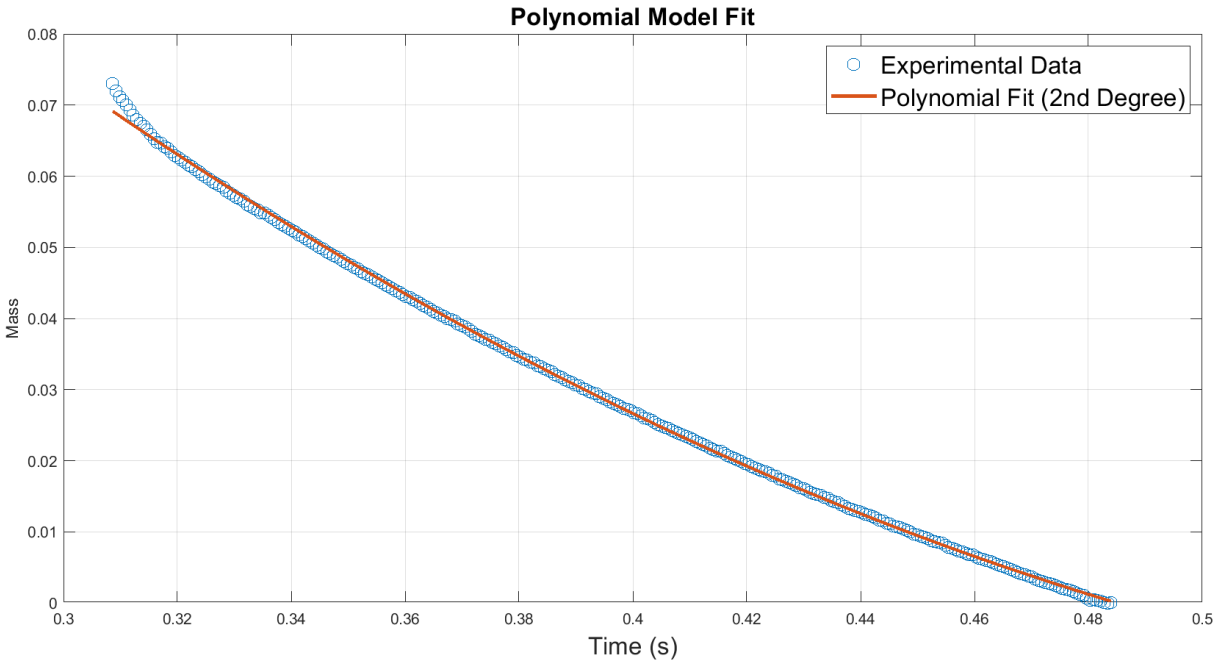

Figure 4.30: General Polynomial Model (PM) plot for Test T4. The plot illustrates the overall fitting of the model to the data from Test T4.

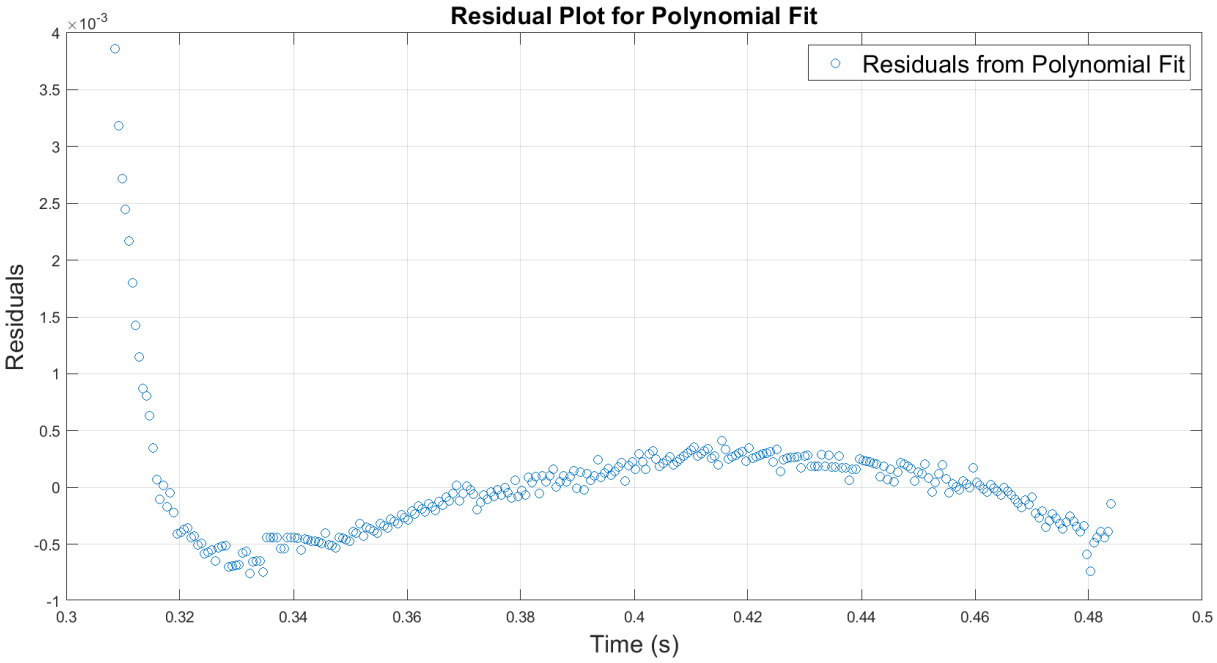

Figure 4.31: Residuals for the Polynomial Model (PM) fit for Test T4. This plot depicts the differences between observed data and model predictions.

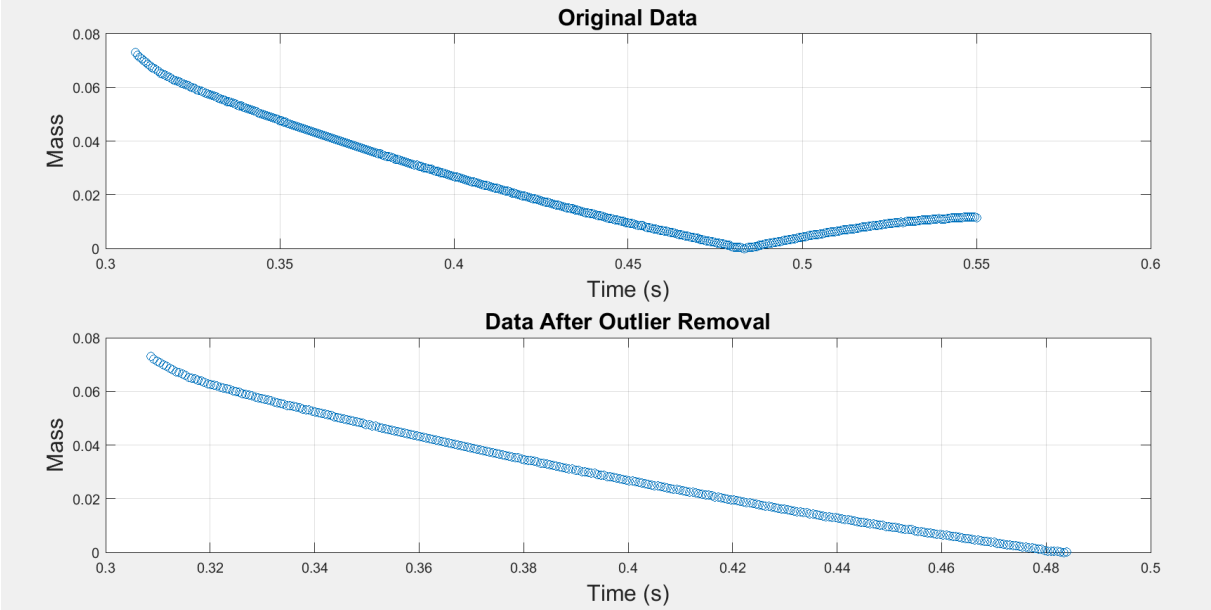

Figure 4.32: Outlier removal for Test T4. This plot illustrates the removal of data points identified as outliers.

4.2.5 Test T5

Figure 4.33 shows the overall fitting of the Polynomial Model to the data from Test T5. The model provides a very close match to the experimental data, accurately capturing the drying behaviour over the full time range.

Residuals for Test T5 are shown in Figure 4.34, where the small residuals suggest that the Polynomial Model performs very well in this test, with minimal discrepancies between the predicted and observed values.

Table 4.21 summarises the Polynomial Model parameters for Test T5, including the quadratic and linear coefficients, which have been optimised to provide a good fit.

Table 4.22 presents the goodness-of-fit metrics for Test T5. The model achieves an R^2 of 0.9994, indicating an excellent fit to the data with low error values.

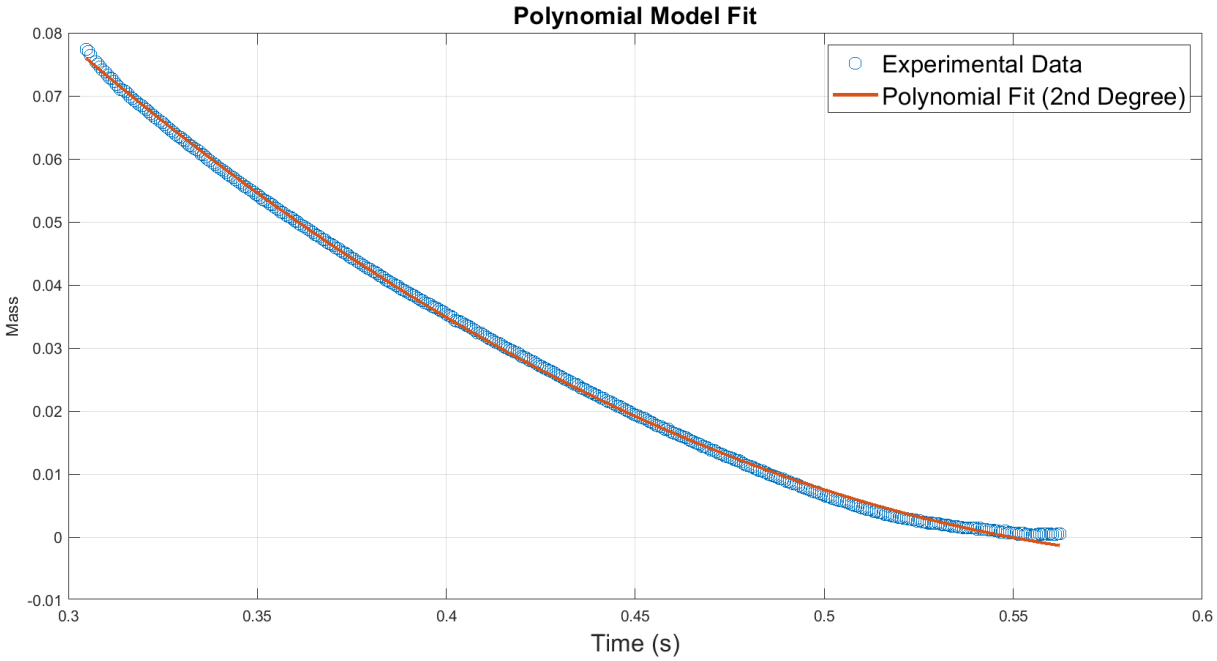

Figure 4.33: General plot for Test T5 showing the overall drying behaviour.

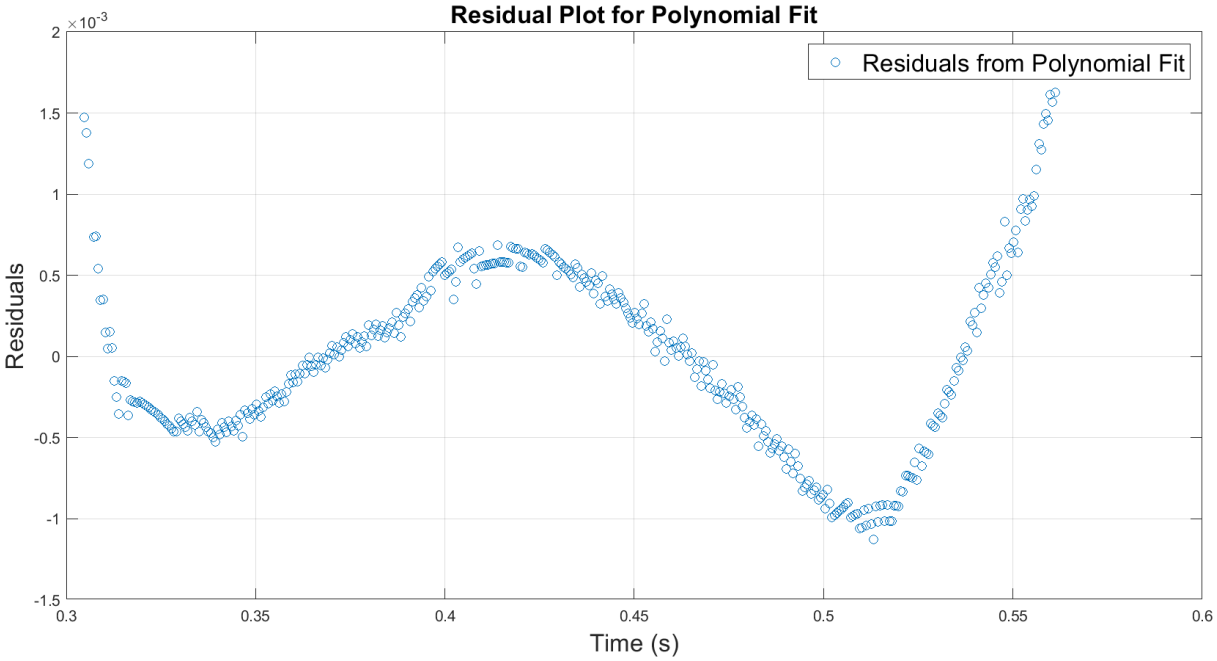

Figure 4.34: Residuals for the Polynomial Model (PM) fit for Test T5. This plot depicts the differences between observed data and model predictions.

Model	Parameter	Value
Polynomial Model	a_2 (Quadratic Coefficient)	0.8086
	a_1 (Linear Coefficient)	-1.0011
	a_0 (Intercept)	0.3059

Table 4.21: Polynomial Model Parameters for Test 5

4.2.6 Test T6

Figure 4.35 shows the overall fit of the Polynomial Model for Test T6. The model accurately captures the overall trend in the data, providing a good fit across the full time range.

Figure 4.36 presents the residuals for Test T6, where the residuals are very small, indicating the model's high level of accuracy in predicting the experimental data.

Table 4.23 lists the parameters of the Polynomial Model for Test T6, including the quadratic and linear terms, which were optimised for the best fit.

Table 4.24 summarises the goodness-of-fit metrics for Test T6. The Polynomial Model achieves an R^2 of 0.9990, reflecting a very strong fit with minimal error values.

Metric	Value
SSE (Polynomial)	0.0001
RMSE (Polynomial)	0.0006
MAE (Polynomial)	0.0005
R^2 (Polynomial)	0.9994

Table 4.22: Goodness of Fit for Polynomial Model (Test 5)

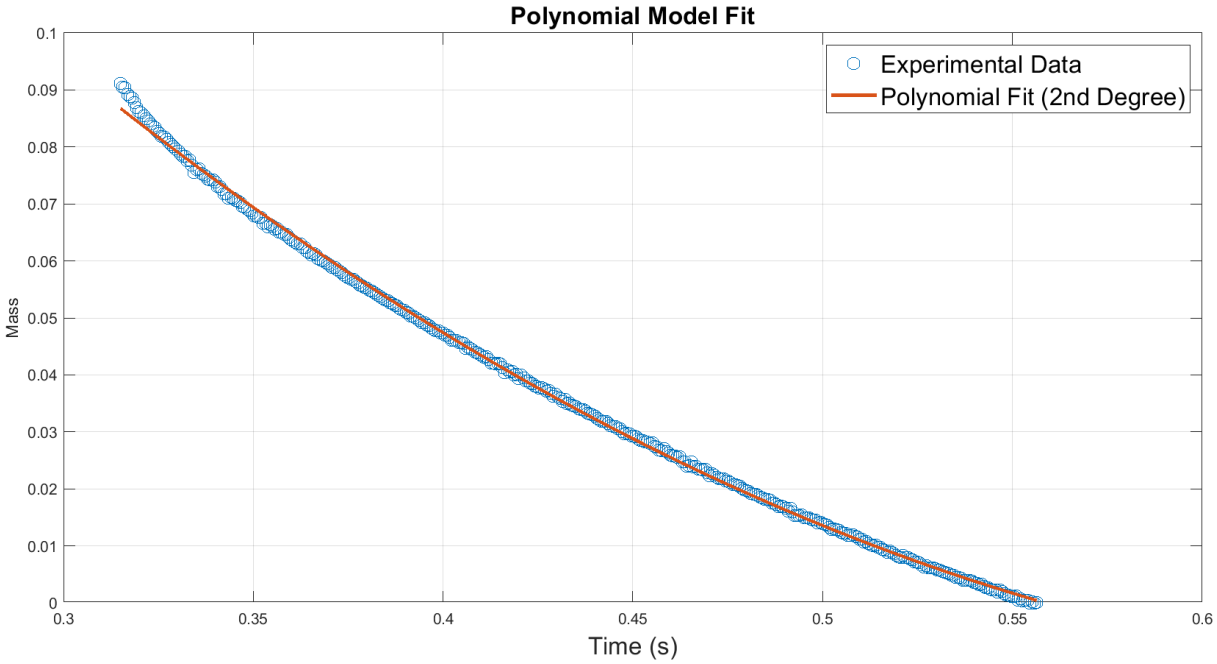

Figure 4.35: General plot for Test T6 showing the overall drying behaviour.

Model	Parameter	Value
Polynomial Model	a_2 (Quadratic Coefficient)	0.6740
	a_1 (Linear Coefficient)	-0.9451
	a_0 (Intercept)	0.3176

Table 4.23: Polynomial Model Parameters for Test 6

Metric	Value
SSE (Polynomial)	0.0002
RMSE (Polynomial)	0.0008
MAE (Polynomial)	0.0005
R^2 (Polynomial)	0.9990

Table 4.24: Goodness of Fit for Polynomial Model (Test 6)

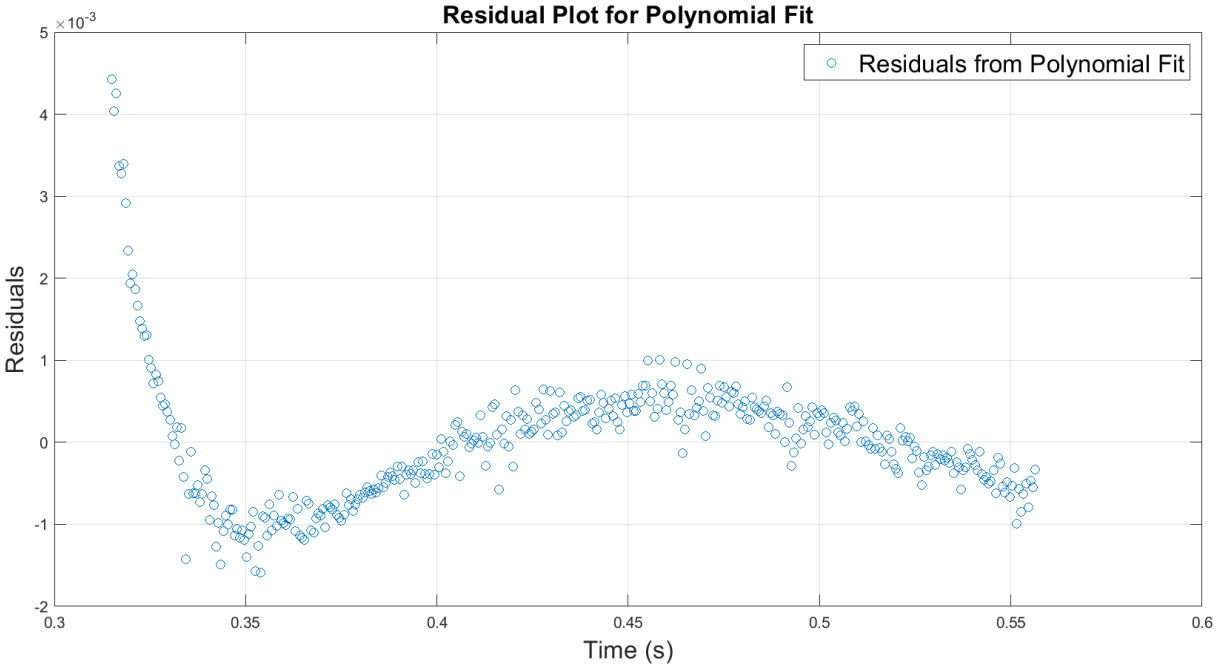

Figure 4.36: Residuals for the Polynomial Model (PM) fit for Test T6. This plot depicts the differences between observed data and model predictions.

CHAPTER 5

Discussion

The primary goal of this thesis was to determine which model best describes the behaviour of hydrogels during deswelling. While extensive research has been conducted on the dehydration behaviour of hydrogels, most studies have focused on simpler exponential or linear models to describe deswelling kinetics [11, 16]. However, these models often fail to capture the complexity observed in materials with more intricate structures. To address this gap, this study applied more sophisticated models, including the Two-Phase Drying Model and Polynomial models [27, 28].

The performance of the models is assessed through graphical visualisations, residual plots, and goodness-of-fit metrics. These aspects help determine the accuracy of each model in representing the deswelling behaviour of hydrogels.

5.1 Model Performance

5.1.1 Graphical visualisation

Upon reviewing the graphical representations of various models, it was observed that the First-Order Kinetic Model for Deswelling (see Figure A.1), Double Exponential Non-Linear Regression Model (see Figure A.4), Fickian Diffusion Model (see Figure A.2), and Non-Fickian (Anomalous) Diffusion Model (see Figure A.3) did not provide a sufficient fit to the data. These models failed to accurately represent the deswelling behaviour of hydrogels, as evidenced by their graphical outputs presented in the Appendix section A.1.

In examining the comparative fits of the Broken Stick Model and the Polynomial Model to the hydrogel deswelling data, it is evident that both models provide a high degree of accuracy, but differences emerge upon closer inspection. The Polynomial Model consistently aligns with the data across all time intervals, capturing more subtle variations in the data trends. This is particularly evident in the zoomed-in figures 4.2 4.5 4.9 4.13 4.17 4.20, where the Polynomial Model, represented by the black dashed line, closely follows the observed fluctuations.

Conversely, the Broken Stick Model maintains a robust fit initially but starts to deviate slightly after reaching the optimised critical time (t_c) as we can see in figures 4.2 4.5 4.9 4.13 4.17 4.20.

At this point, the model transitions into following the linear model trend, which causes minor discrepancies in the fit, especially when compared to the Polynomial Model's ability to capture non-linear variations in the later stages. While both models perform well, the Polynomial Model generally offers a more accurate fit across all test sets.

Consequently, a more focused analysis was conducted on the Goodness-of-Fit metrics and residual plots for the Broken Stick Model and the Polynomial 2nd Degree Model. This comparative analysis aimed to identify which of these models offered a superior fit to the data, ensuring a more accurate representation of the hydrogels' deswelling characteristics.

5.1.2 Residual Analysis

Residual plots are a critical diagnostic tool used to assess the fit of a regression model. These plots display the residuals, which are the differences between the observed and predicted values, against the predicted values. Ideally, residuals should be randomly scattered around zero with no clear pattern, indicating that the model adequately captures the relationship between the variables and that the assumptions of linearity and constant variance are met [29]. If the residuals show a non-random pattern or curvature, it could suggest model inadequacy or that a different model might be more appropriate [29].

An ideal residual plot would show no patterns, confirming that the model has accounted for all systematic variation and leaving only random errors. Visual inspection of these plots, as supported by statistical tests, helps in determining the suitability of the model [29].

The residual plots for both the polynomial and Broken Stick models reveal notable differences in the ability of each model to fit the data. In both cases, residuals display clear curvature patterns, which indicate that the models capture some non-linear trends in the data, although imperfectly. Particularly in the later stages of the time series, both models show increasing residuals, suggesting heteroscedasticity, where the variance of the residuals is not constant across all fitted values [29].

According to theoretical expectations [29], residual plots should exhibit a random scatter around zero without any discernible patterns and should maintain a consistent spread [29]. Neither model fully achieves this, as both present evident trends and increasing variability in their residuals [29].

Comparing the two models, the Broken Stick Model appears to fit the data better, especially towards the end of the time period analysed. The residuals for this model remain smaller and more controlled compared to the polynomial model, which shows a significant increase in residual magnitude in these regions [29]. However, the polynomial model might capture the mid-range data more accurately, with smaller residuals in that portion of the time series. Its tendency for larger errors at the end suggests that the Broken Stick Model might still be a better overall fit due to more stable residuals across the full time span [29].

This nuanced interpretation depends on which part of the data is valued more: if the end-

time performance is critical, the Broken Stick Model performs better, while the Polynomial Model may be preferable for mid-range time data [29].

5.1.3 Goodness-of-fit analysis

The analysis of the goodness-of-fit metrics for the Broken Stick Model and the Polynomial Model highlights several key findings regarding their performance across different test sets.

For the Broken Stick Model, the Sum of Squared Errors (SSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 values consistently indicate a high level of accuracy. Specifically, R^2 values range from 0.9980 to 0.9998 across various test sets, demonstrating an exceptional fit to the data. Notably, the lowest R^2 value of 0.9980 is observed in Test 5, with corresponding SSE, RMSE, and MAE values remaining relatively low, reflecting a strong performance of the model.

In comparison, the Polynomial Model exhibits even higher R^2 values, ranging from 0.9990 to 0.9999 across the test sets. The highest R^2 value of 0.9999 is recorded in Test 3, alongside lower SSE, RMSE, and MAE values compared to the Broken Stick Model. These metrics suggest that the Polynomial Model generally provides a slightly better fit, capturing data nuances more effectively.

5.1.4 Comparison between models

Overall, while both models demonstrate high-quality fits, the Polynomial Model stands out with marginally superior R^2 values and error metrics, as well as a better fit with regard to the visualisation, indicating a more accurate representation of the data across the test sets. However, the Broken Stick Model's stability in residuals suggests that it may be a more reliable choice when considering the full time span of the data.

The superior performance of the Polynomial Model is due to its ability to handle the complexity of the data. Unlike simpler models, the polynomial model's coefficients let it fit the data more flexibly. The second-degree polynomial, in particular, captures the curvature in the deswelling data, making it effective at representing non-linear trends. The coefficients of the polynomial show how the input variables relate to the response, reflecting both linear and quadratic effects. This allows for a better understanding of variations such as acceleration or deceleration in deswelling. As noted by Ostertagová [26], polynomial regression provides a detailed interpretation of these parameters, helping to explain the physical processes behind the deswelling behaviour.

5.2 Limitations and Future Research

This study has several limitations that should be considered. One of the main constraints is the potential impact of the model's assumptions on the accuracy of the results. Specifically, the

simplicity of the Polynomial and Broken Stick Models may not fully capture all the nuances of the dataset, such as non-linear behaviour at extreme values of the time range. Additionally, the limited number of test sets used in this study might restrict the generalisability of the findings across a broader range of scenarios [6, 5].

To address these limitations and guide future research, there are several potential avenues for improvement. Firstly, investigating models that account for more complex, non-linear behaviours could lead to better accuracy, especially for time series data where extreme values are important. Additionally, exploring hybrid models that combine aspects of both the Broken Stick and Polynomial Models might offer a more robust solution, balancing mid-range accuracy with stable performance across the entire data set [18, 25]. Future work could also include testing the models on a more extensive dataset to enhance the reliability and applicability of the findings to a wider variety of contexts.

5.3 Conclusion

This study has demonstrated that the Polynomial Model outperforms other models, including the Two-Phase Drying Model, in predicting the dehydration behaviour of hydrogels [22, 28]. The Polynomial Model's ability to capture non-linear trends through its coefficients, which reflect both linear and quadratic effects, provides a more accurate representation of the drying process compared to simpler models. This enhanced fit offers a better understanding of hydrogel dehydration and highlights the model's robustness in capturing the underlying complexities of the data [26]. These findings not only contribute to a deeper insight into hydrogel behaviour but also have implications for advancing the development of hydrogel-based materials [7]. Future research should address the identified limitations and explore new modelling approaches to further enhance the accuracy and applicability of predictions in this field.

Bibliography

[1] K. S. SUSlick, "Kirk-othmer encyclopedia of chemical technology," *J. Wiley & Sons: New York*, vol. 26, pp. 517–541, 1998. → [p1], [p3], [p4], [p8]

[2] H. Omidian and K. Park, "Engineered high swelling hydrogels," *Biomedical applications of hydrogels handbook*, pp. 351–374, 2010. → [p1], [p3], [p4], [p6], [p8]

[3] A. Patel and K. Mequanint, "Hydrogel biomaterials," in *Biomedical engineering-frontiers and challenges*, IntechOpen, 2011. → [p1], [p3], [p4], [p6], [p8]

[4] S. Yang, L. Jang, S. Kim, J. Yang, K. Yang, S.-W. Cho, and J. Y. Lee, "Polypyrrole/alginate hybrid hydrogels: electrically conductive and soft biomaterials for human mesenchymal stem cell culture and potential neural tissue engineering applications," *Macromolecular Bioscience*, vol. 16, no. 11, pp. 1653–1661, 2016. → [p1], [p3]

[5] T.-C. Ho, C.-C. Chang, H.-P. Chan, T.-W. Chung, C.-W. Shu, K.-P. Chuang, T.-H. Duh, M.-H. Yang, and Y.-C. Tyan, "Hydrogels: properties and applications in biomedicine," *Molecules*, vol. 27, no. 9, p. 2902, 2022. → [p1], [p3], [p54]

[6] F. Ullah, M. B. H. Othman, F. Javed, Z. Ahmad, and H. M. Akil, "Classification, processing and application of hydrogels: A review," *Materials Science and Engineering: C*, vol. 57, pp. 414–433, 2015. → [p3], [p8], [p54]

[7] N. A. Peppas and A. R. Khare, "Preparation, structure and diffusional behavior of hydrogels in controlled release," *Advanced drug delivery reviews*, vol. 11, no. 1-2, pp. 1–35, 1993. → [p3], [p4], [p9], [p10], [p54]

[8] D. Caccavo, S. Cascone, G. Lamberti, and A. Barba, "Hydrogels: experimental characterization and mathematical modelling of their mechanical and diffusive behaviour," *Chemical Society Reviews*, vol. 47, no. 7, pp. 2357–2373, 2018. → [p3], [p4], [p6], [p9]

[9] M. P. Kainz, A. Greiner, J. Hinrichsen, D. Kolb, E. Comellas, P. Steinmann, S. Budday, M. Terzano, and G. A. Holzapfel, "Poro-viscoelastic material parameter identification of brain tissue-mimicking hydrogels," *Frontiers in bioengineering and biotechnology*, vol. 11, p. 1143304, 2023. → [p3], [p4], [p18]

[10] D. Caccavo, "An overview on the mathematical modeling of hydrogels' behavior for drug delivery systems," *International journal of pharmaceutics*, vol. 560, pp. 175–190, 2019. → [p3], [p4], [p8]

[11] S. Nieuwenhuis, Q. Zhong, E. Metwalli, L. Bießmann, M. Philipp, A. Miasnikova, A. Laschewsky, C. M. Papadakis, R. Cubitt, J. Wang, *et al.*, "Hydration and dehydration kinetics: Comparison between poly (n-isopropyl methacrylamide) and poly (methoxy diethylene glycol acrylate) films," *Langmuir*, vol. 35, no. 24, pp. 7691–7702, 2019. → [p4], [p5], [p8], [p51]

[12] F. Horkay, M.-H. Han, I. S. Han, I.-S. Bang, and J. J. Magda, "Separation of the effects of ph and polymer concentration on the swelling pressure and elastic modulus of a ph-responsive hydrogel," *Polymer*, vol. 47, no. 21, pp. 7335–7338, 2006. → [p4], [p5], [p9]

[13] M. U. Joardder, M. Mourshed, M. Hasan Masud, M. U. Joardder, M. Mourshed, and M. Hasan Masud, "Characteristics of bound water," *State of Bound Water: Measurement and Significance in Food Processing*, pp. 29–45, 2019. → [p5]

[14] R. Singhal and K. Gupta, "A review: tailor-made hydrogel structures (classifications and synthesis parameters)," *Polymer-Plastics Technology and Engineering*, vol. 55, no. 1, pp. 54–70, 2016. → [p6], [p8]

[15] M. B. Łabowska, M. Skrodzka, H. Sicińska, I. Michalak, and J. Detyna, "Influence of cross-linking conditions on drying kinetics of alginate hydrogel," *Gels*, vol. 9, no. 1, p. 63, 2023. → [p6]

[16] K. Takahashi, T. Takigawa, and T. Masuda, "Swelling and deswelling kinetics of poly (n-isopropylacrylamide) gels," *The Journal of chemical physics*, vol. 120, no. 6, pp. 2972–2979, 2004. → [p8], [p10], [p51]

[17] Z. Zhao, Z. Li, Q. Xia, E. Bajalis, H. Xi, and Y. Lin, "Swelling/deswelling kinetics of pnipaam hydrogels synthesized by microwave irradiation," *Chemical Engineering Journal*, vol. 142, no. 3, pp. 263–270, 2008. → [p8]

[18] S. Van Buuren, "Broken stick model for irregular longitudinal data," *Journal of Statistical Software*, vol. 106, pp. 1–51, 2023. → [p8], [p9], [p12], [p13], [p54]

[19] H. Moussout, H. Ahlafi, M. Aazza, and H. Maghat, "Critical of linear and nonlinear equations of pseudo-first order and pseudo-second order kinetic models," *Karbala International Journal of Modern Science*, vol. 4, no. 2, pp. 244–254, 2018. → [p9]

[20] J. S. Vrentas and J. L. Duda, "Molecular diffusion in polymer solutions," *AIChE Journal*, vol. 25, no. 1, pp. 1–24, 1979. → [p9], [p10], [p11]

[21] A. G. Cherstvy, S. Thapa, C. E. Wagner, and R. Metzler, "Non-gaussian, non-ergodic, and non-fickian diffusion of tracers in mucin hydrogels," *Soft Matter*, vol. 15, no. 12, pp. 2526–2551, 2019. → [p9], [p11], [p12]

[22] I. G. Mason, R. I. McLachlan, and D. T. Gérard, "A double exponential model for biochemical oxygen demand," *Bioresource Technology*, vol. 97, no. 2, pp. 273–282, 2006. → [p9], [p10], [p54]

[23] J. O. Rawlings, S. G. Pantula, and D. A. Dickey, *Applied regression analysis: a research tool*. Springer, 1998. → [p9], [p18], [p19]

[24] J. D. Jovanović and B. K. Adnadjević, "Summary of investigations in regard to the kinetics of absorbed water dehydration from different hydrogels," in *A Comprehensive Review of the Versatile Dehydration Processes*, IntechOpen, 2023. → [p9]

[25] B. Potkonjak, J. Jovanović, B. Stanković, S. Ostojić, and B. Adnadjević, "Comparative analyses on isothermal kinetics of water evaporation and hydrogel dehydration by a novel nucleation kinetics model," *Chemical Engineering Research and Design*, vol. 100, pp. 323–330, 2015. → [p10], [p54]

[26] E. Ostertagová, "Modelling using polynomial regression," *Procedia engineering*, vol. 48, pp. 500–506, 2012. → [p12], [p18], [p53], [p54]

[27] L. Brown and C. Zukoski, "Experimental tests of two-phase fluid model of drying consolidation," *AIChE journal*, vol. 49, no. 2, pp. 362–372, 2003. → [p13], [p51]

[28] A. Setapa, N. Ahmad, S. Mohd Mahali, and M. C. I. Mohd Amin, "Mathematical model for estimating parameters of swelling drug delivery devices in a two-phase release," *Polymers*, vol. 12, no. 12, p. 2921, 2020. → [p51], [p54]

[29] C.-L. Tsai, Z. Cai, and X. Wu, "The examination of residual plots," *Statistica Sinica*, pp. 445–465, 1998. → [p52], [p53]

CHAPTER A

Appendix

A.1 Additional Figures

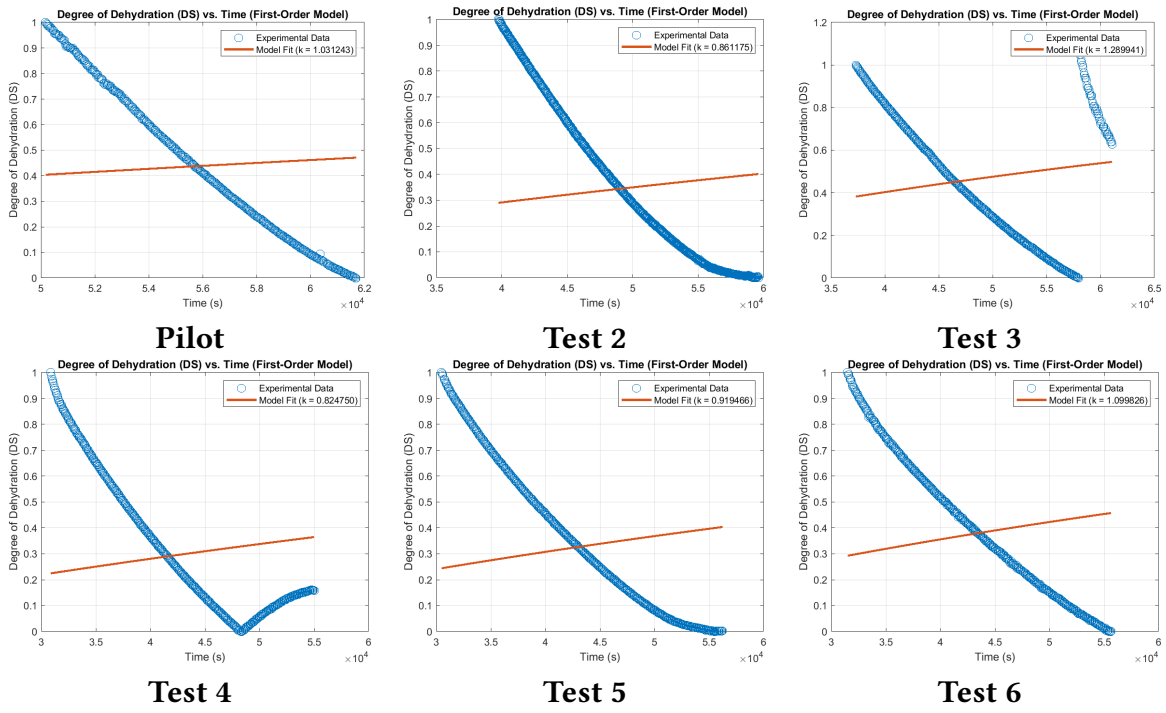

Figure A.1: Performance of the First-Order Kinetic Model for Deswelling.

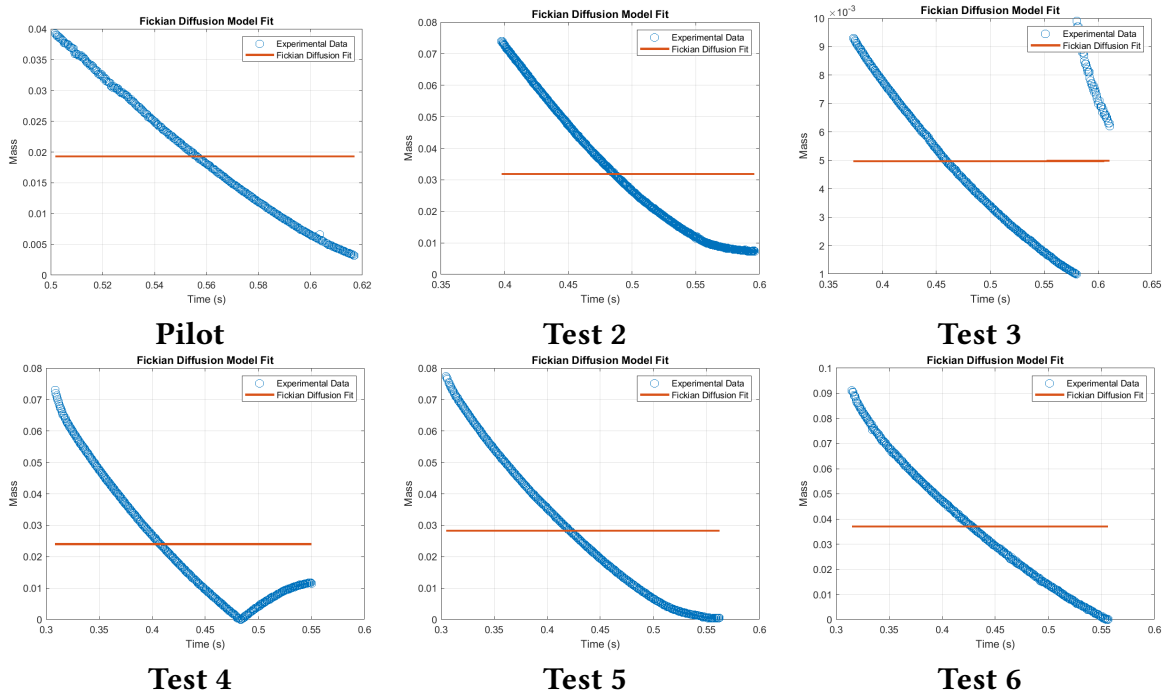

Figure A.2: Performance of the Fickian Diffusion Model.

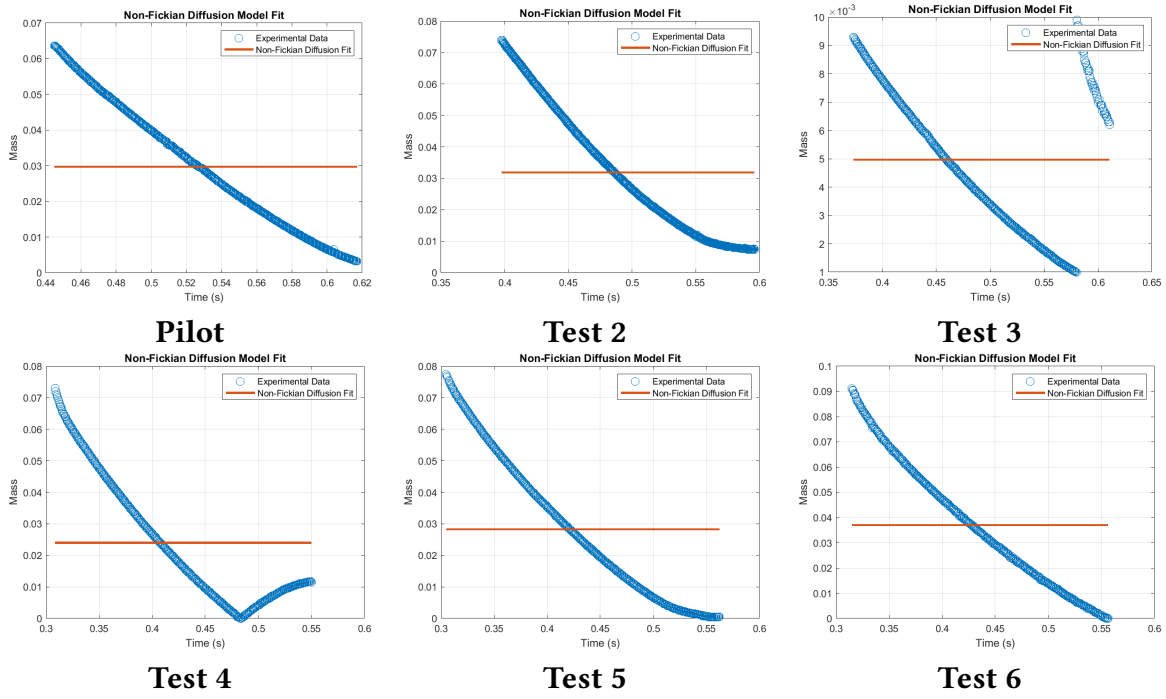

Figure A.3: Performance of the Non-Fickian (Anomalous) Diffusion Model

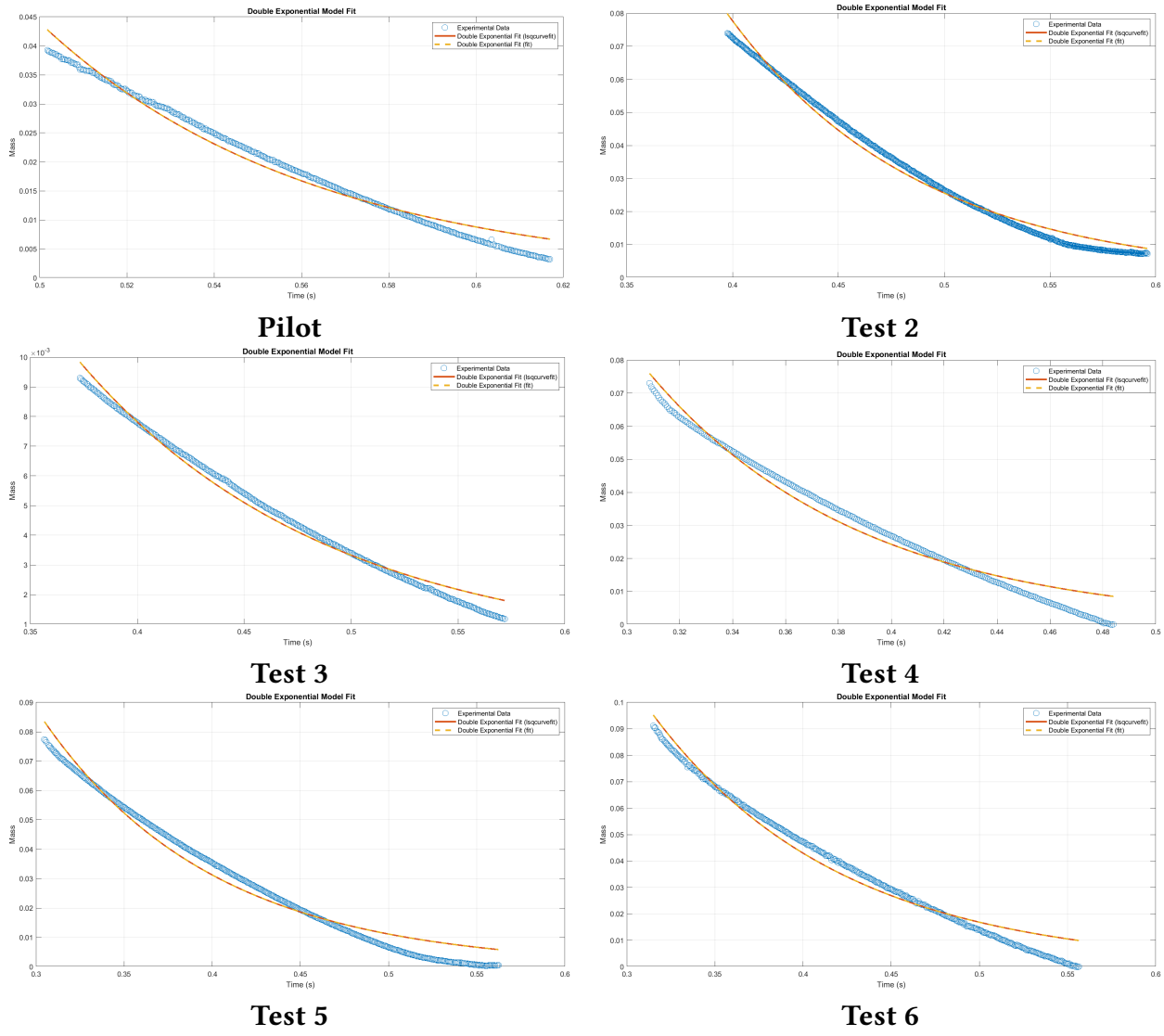

Figure A.4: Performance of the Double Exponential Model

CHAPTER B

Code Used

B.1 MATLAB Scripts

B.1.1 First-Order Kinetic Model for Deswelling

```
% Carga los datos desde el archivo de Excel
data = readtable("pilot2.csv");

%% CLEANING
% Clean and convert cells to numbers
time = seconds(data.Var1); %time
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); %initia
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false);

% Convert cleaned cell arrays to numeric arrays
Mt = cell2mat(Mt);
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

%% Deducir M0 (masa inicial) y Me (masa en equilibrio)
M0 = Mt(1);        % Asumimos que la masa al tiempo t=0 es la masa inicial
Me = min(Mt);      % Asumimos que la masa mínima registrada es la masa en equilibri

%% Calcula el Deswelling Degree (DS)
DS = (Me - Mt) ./ (Me - M0);

%% Agrega la columna DS a la tabla original
data.DS = DS;

% % Muestra la tabla con el Deswelling Degree
% disp(data);
```

```
%Guarda la tabla con el Deswelling Degree en un nuevo archivo de Excel
writetable(data, 'test_2DS.xlsx');

% %% Plotear DS vs. masa
% figure;
% plot(Mt, DS, '-o', 'LineWidth', 2);
% xlabel('Mt');
% ylabel('DS');
% title('DS vs. Mt');
% grid on;

%% Definición de la función del modelo exponencial

% Reescalar el tiempo si es necesario
time_rescaled = time / 100000;

% % Define la función modelo exponencial
% modelFun = @(k, t) 1 - exp(-k * t);
%
% % Define la función de error para el ajuste
% errorFun = @(k) sum((DS - modelFun(k, time_rescaled)).^2);
%
% % Probar diferentes valores iniciales de la constante k
% k0 = 0.01; % Ajusta este valor inicial si es necesario
%
% % Utiliza 'lsqcurvefit' para ajustar el modelo a los datos
% kFit = lsqcurvefit(modelFun, k0, time_rescaled, DS);
%
% % Calcular el ajuste del modelo
% DS_fit = modelFun(kFit, time_rescaled);
%
% % Verifica el valor ajustado de k
% fprintf('Valor ajustado de k: %.6f\n', kFit);
%
% % Plot the data and model fit
% figure;
% plot(time, DS, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental Data');
% hold on;
% plot(time, DS_fit, '-', 'LineWidth', 2, 'DisplayName', sprintf('Model Fit (k = %.
% xlabel('Time (s)');
% ylabel('Degree of Dehydration (DS)');
% title('Degree of Dehydration (DS) vs. Time (First-Order Model)');
```

```
% legend;
% grid on;

%% Define el tipo de ajuste usando 'fittype'
modelType = fittype('1 - exp(-k * t)', 'independent', 't', 'coefficients', 'k');

% Define las opciones para el ajuste no lineal
fitOptions = fitoptions('Method', 'NonlinearLeastSquares', 'StartPoint', 0.01);

% Realiza el ajuste usando 'fit'
[fitResult, gof] = fit(time_rescaled, DS, modelType, fitOptions);

% Extrae el valor ajustado de k
kFit = fitResult.k;

% Verifica el valor ajustado de k
fprintf('Valor ajustado de k: %.6f\n', kFit);

% Calcular el ajuste del modelo
DS_fit = modelType(kFit, time_rescaled);

% Plot the data and model fit
figure;
plot(time, DS, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental Data');
hold on;
plot(time, DS_fit, '-', 'LineWidth', 2, 'DisplayName', sprintf('Model Fit (k = %.6f)
xlabel('Time (s)');
ylabel('Degree of Dehydration (DS)');
title('Degree of Dehydration (DS) vs. Time (First-Order Model)');
legend;
grid on;
```

B.1.2 Double Exponential Non-Linear Regression Model

```
 % Load data from the Excel file
data = readtable("pilot2.csv");

%% CLEANING
% Clean and convert cells to numbers
time = seconds(data.Var1); % time
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); % init
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false);
```

```
% Convert cleaned cell arrays to numeric arrays
Mt = cell2mat(Mt);
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

% Rescale the time if necessary
time_rescaled = time / 100000;

%% OUTLIER REMOVAL

% % TEST 3
% % Calculate Mahalanobis distance
% dataMatrix = [time_rescaled, Mt, Area_side, Area_top];
% meanData = mean(dataMatrix);
% covData = cov(dataMatrix);
% mahalDist = mahal(dataMatrix, dataMatrix);
%
% % Determine the threshold for identifying outliers
% threshold = chi2inv(0.81, size(dataMatrix, 2));
%
% % Identify outliers
% isOutlier = mahalDist > threshold;
%
% % Filter the data to remove outliers
% time_filtered = time_rescaled(~isOutlier);
% Mt_filtered = Mt(~isOutlier);
% Area_side_filtered = Area_side(~isOutlier);
% Area_top_filtered = Area_top(~isOutlier);
%
% % Display the number of outliers removed
% fprintf('Outliers removed: %d\n', sum(isOutlier));

% TEST 4
% Filtrar datos para excluir muestras a partir del segundo 0.48400
time_threshold = 100000 %57000 %48400; % Tiempo en segundos a partir del cual se del
valid_indices = time_rescaled <= time_threshold;

% Filtrar los datos
time_filtered = time_rescaled(valid_indices);
Mt_filtered = Mt(valid_indices);
Area_side_filtered = Area_side(valid_indices);
Area_top_filtered = Area_top(valid_indices);
```

```
% Mostrar el número de datos eliminados
fprintf('Muestras eliminadas (tiempo > %.3f s): %d\n', time_threshold, sum(~valid_in

% Plot the data with and without outliers for verification
figure;
subplot(2,1,1);
plot(time_rescaled, Mt, 'o');
title('Original Data');
xlabel('Time (s)');
ylabel('Mass');
grid on;

subplot(2,1,2);
plot(time_filtered, Mt_filtered, 'o');
title('Data After Outlier Removal');
xlabel('Time (s)');
ylabel('Mass');
grid on;

%% MODEL FITTING

% Define the double exponential model
double_exp_model = @(params, t) params(1) * (params(2) * exp(-t / params(3)) + (1 -

% Define the error function for fitting
error_function = @(params) Mt_filtered - double_exp_model(params, time_filtered);

% Initial parameter values [M_inf, A, t1, t2]
M_inf_guess = max(Mt_filtered);
A_guess = 0.5; % Initial guess for A
t1_guess = range(time_filtered) / 2000; % Estimate t1 as 1/100th of the time range
t2_guess = range(time_filtered) / 1;  % Estimate t2 as 1/10th of the time range
initial_params = [M_inf_guess, A_guess, t1_guess, t2_guess];

% Fit the model using lsqcurvefit
options = optimoptions('lsqcurvefit', 'Display', 'iter', 'Algorithm', 'trust-region-
[params_fit, resnorm] = lsqcurvefit(double_exp_model, initial_params, time_filtered

% Extract the fitted parameters
M_inf_fit = params_fit(1);
A_fit = params_fit(2);
t1_fit = params_fit(3);
```

```
t2_fit = params_fit(4);
fprintf('Fitted parameters with lsqcurvefit:\nM_inf = %.4f\nA = %.4f\nt1 = %.4f\nt2
% Calculate fitted values with lsqcurvefit
Mt_fit_lsqcurvefit = double_exp_model(params_fit, time_filtered);

%% Define fittype for nonlinear fitting with fit
ft = fittype('M_inf * (A * exp(-x / t1) + (1 - A) * exp(-x / t2))', ...
    'independent', 'x', 'coefficients', {'M_inf', 'A', 't1', 't2'});

% Options for nonlinear fitting with fit
opts = fitoptions('Method', 'NonlinearLeastSquares', 'StartPoint', initial_params);

% Fit the model using fit
[fit_result, gof] = fit(time_filtered, Mt_filtered, ft, opts);

% Extract the fitted parameters
M_inf_fit2 = fit_result.M_inf;
A_fit2 = fit_result.A;
t1_fit2 = fit_result.t1;
t2_fit2 = fit_result.t2;
fprintf('Fitted parameters with fit:\nM_inf = %.4f\nA = %.4f\nt1 = %.4f\nt2 = %.4f\n

% Calculate fitted values with fit
Mt_fit_fit = M_inf_fit2 * (A_fit2 * exp(-time_filtered / t1_fit2) + (1 - A_fit2) * e

%% Results Plot
figure;
plot(time_filtered, Mt_filtered, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental
hold on;
plot(time_filtered, Mt_fit_lsqcurvefit, '-', 'LineWidth', 2, 'DisplayName', 'Double
plot(time_filtered, Mt_fit_fit, '--', 'LineWidth', 2, 'DisplayName', 'Double Expone
xlabel('Time (s)');
ylabel('Mass');
title('Double Exponential Model Fit');
legend;
grid on;
```

B.1.3 Fickian Diffusion Model

```
% Carga los datos desde el archivo de Excel
data = readtable("test_6.csv");
```

```
%% CLEANING
% Clean and convert cells to numbers
time = seconds(data.Var1); %time
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); %initia
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false);

% Convert cleaned cell arrays to numeric arrays
Mt = cell2mat(Mt);
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

% %% Verificar que los datos no contengan valores complejos o NaNs
% if any(~isreal(time)) || any(~isreal(Mt))
%     error('Los datos contienen valores complejos.');
% end
%
% if any(isnan(time)) || any(isnan(Mt))
%     error('Los datos contienen valores NaN.');
% end
%
% % Asegurarse de que el tiempo no contenga ceros
% if any(time == 0)
%     error('Los datos de tiempo contienen ceros, lo cual puede causar problemas co
% end
%
% % Asegurarse de que los tiempos sean positivos
% if any(time < 0)
%     error('Los datos de tiempo contienen valores negativos.');
% end

%% Definir el modelo de difusión Fickiana
model = @(params, t) params(1) * (1 - erf(params(2) ./ sqrt(4 * params(3) * t)));

% Asegurarse de que el modelo no devuelve valores complejos
time = time(:); % Asegurar que time sea un vector columna
initial_params = [max(Mt), 1, 1]; % Valores iniciales para los parámetros [M_inf, L

% Reescalar el tiempo si es necesario
time_rescaled = time / 100000;

% Definir límites inferiores y superiores
LB = [0, 0, 0]; % Límite inferior
```

```
UB = [Inf, Inf, Inf]; % Límite superior

% Definir la función de error para ajuste
error_function = @(params) Mt - model(params, time_rescaled);

% Opciones para el ajuste no lineal
options = optimoptions('lsqcurvefit', 'Display', 'iter', 'Algorithm', 'trust-region-

% Intentar ajustar el modelo
try
    [params_fit, resnorm] = lsqcurvefit(model, initial_params, time_rescaled, Mt, LB
catch ME
    error('Error en lsqcurvefit: %s', ME.message);
end

% Resultados
M_inf_fit = params_fit(1);
L_fit = params_fit(2);
D_fit = params_fit(3);
fprintf('Parámetros ajustados:\nM_inf = %.4f\nL = %.4f\nD = %.4f\n', M_inf_fit, L_fi

% Calcular los valores ajustados
Mt_fit = model(params_fit, time_rescaled);

% Gráfica de resultados
figure;
plot(time_rescaled, Mt, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental Data');
hold on;
plot(time_rescaled, Mt_fit, '-', 'LineWidth', 2, 'DisplayName', sprintf('Fickian Di
xlabel('Time (s)');
ylabel('Mass');
title('Fickian Diffusion Model Fit');
legend;
grid on;
```

B.1.4 Non-Fickian (Anomalous) Diffusion Model

```
% Carga los datos desde el archivo de Excel
data = readtable("test_6.csv");

%% CLEANING
% Clean and convert cells to numbers
time = seconds(data.Var1); %time
```

```
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); %initia
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false);

% Convert cleaned cell arrays to numeric arrays
Mt = cell2mat(Mt);
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

%% Verificar que los datos no contengan valores complejos o NaNs
if any(~isreal(time)) || any(~isreal(Mt))
    error('Los datos contienen valores complejos.');
end

if any(isnan(time)) || any(isnan(Mt))
    error('Los datos contienen valores NaN.');
end

% Asegurarse de que el tiempo no contenga ceros
if any(time == 0)
    error('Los datos de tiempo contienen ceros, lo cual puede causar problemas en el
end

% Asegurarse de que los tiempos sean positivos
if any(time < 0)
    error('Los datos de tiempo contienen valores negativos.');
end

%% Definir el modelo de difusión anómala
model = @(params, t) params(1) * (1 - exp(- (t / params(2)).^params(3)));

% Asegurarse de que el modelo no devuelve valores complejos
time = time(:); % Asegurar que time sea un vector columna
initial_params = [max(Mt), 1, 1]; % Valores iniciales para los parámetros [M_inf, ta

% Reescalar el tiempo si es necesario
time_rescaled = time / 100000;

% Definir límites inferiores y superiores
LB = [0, 0, 0]; % Límite inferior
UB = [Inf, Inf, 2]; % Límite superior (exponente alfa generalmente se limita a un ma

% Opciones para el ajuste no lineal
```

```
options = optimoptions('lsqcurvefit', 'Display', 'iter', 'Algorithm', 'trust-region-

% Intentar ajustar el modelo
try
    [params_fit, resnorm] = lsqcurvefit(model, initial_params, time_rescaled, Mt, LB
catch ME
    error('Error en lsqcurvefit: %s', ME.message);
end

% Resultados
M_inf_fit = params_fit(1);
tau_fit = params_fit(2);
alpha_fit = params_fit(3);
fprintf('Parámetros ajustados:\nM_inf = %.4f\nTau = %.4f\nAlpha = %.4f\n', M_inf_fit

% Calcular los valores ajustados
Mt_fit = model(params_fit, time_rescaled);

% Gráfica de resultados
figure;
plot(time_rescaled, Mt, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental Data');
hold on;
plot(time_rescaled, Mt_fit, '-', 'LineWidth', 2, 'DisplayName', sprintf('Non-Fickian
xlabel('Time (s)');
ylabel('Mass');
title('Non-Fickian Diffusion Model Fit');
legend;
grid on;
```

B.1.5 Second-Degree Polynomial Model

```
% Load data from the Excel file
data = readtable("test_3.csv");

%% CLEANING
% Clean and convert cells to numbers
time = seconds(data.Var1); % time
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); % initi
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false);

% Convert cleaned cell arrays to numeric arrays
Mt = cell2mat(Mt);
```

```
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

% Rescale the time if necessary
time_rescaled = time / 100000;

%% OUTLIER REMOVAL

% TEST 3
% Calculate Mahalanobis distance
dataMatrix = [time_rescaled, Mt, Area_side, Area_top];
meanData = mean(dataMatrix);
covData = cov(dataMatrix);
mahalDist = mahal(dataMatrix, dataMatrix);

% Determine the threshold for identifying outliers
threshold = chi2inv(0.81, size(dataMatrix, 2));

% Identify outliers
isOutlier = mahalDist > threshold;

% Filter the data to remove outliers
time_filtered = time_rescaled(~isOutlier);
Mt_filtered = Mt(~isOutlier);
Area_side_filtered = Area_side(~isOutlier);
Area_top_filtered = Area_top(~isOutlier);

% Display the number of outliers removed
fprintf('Outliers removed: %d\n', sum(isOutlier));

% % TEST 4 i 2
% time_threshold = 1 %0.57 %0.48400; % Time in seconds after which samples are exclu
% valid_indices = time_rescaled <= time_threshold;
%
% % Filter the data
% time_filtered = time_rescaled(valid_indices);
% Mt_filtered = Mt(valid_indices);
% Area_side_filtered = Area_side(valid_indices);
% Area_top_filtered = Area_top(valid_indices);
%
% % Display the number of samples removed
% fprintf('Samples removed (time > %.3f s): %d\n', time_threshold, sum(~valid_indice
```

```
% Plot the data with and without outliers for verification
figure;
subplot(2,1,1);
plot(time_rescaled, Mt, 'o');
title('Original Data', FontSize=16);
xlabel('Time (s)', FontSize=16);
ylabel('Mass', FontSize=16);
grid on;

subplot(2,1,2);
plot(time_filtered, Mt_filtered, 'o');
title('Data After Outlier Removal', FontSize=16);
xlabel('Time (s)', FontSize=16);
ylabel('Mass', FontSize=16);
grid on;

%% MODEL FITTING

% Define the polynomial model (2nd degree)
poly_coeffs = polyfit(time_filtered, Mt_filtered, 2); % Fit a 2nd degree polynomial
poly_model = @(t) polyval(poly_coeffs, t); % Polynomial model as a function

% Calculate fitted values
Mt_fit_poly = poly_model(time_filtered);

% Display Model Parameters
% Extract polynomial coefficients
a2 = poly_coeffs(1); % Coefficient for the quadratic term
a1 = poly_coeffs(2); % Coefficient for the linear term
a0 = poly_coeffs(3); % Intercept

% Display the polynomial parameters
fprintf('Polynomial Model Parameters:\n');
fprintf('a2 (Quadratic Coefficient) = %.4f\n', a2);
fprintf('a1 (Linear Coefficient) = %.4f\n', a1);
fprintf('a0 (Intercept) = %.4f\n', a0);

%% Results Plot
figure;
plot(time_filtered, Mt_filtered, 'o', 'MarkerSize', 8, 'DisplayName', 'Experimental
hold on;
plot(time_filtered, Mt_fit_poly, '-', 'LineWidth', 2, 'DisplayName', 'Polynomial Fit
xlabel('Time (s)', FontSize=16);
```

```
ylabel('Mass');
title('Polynomial Model Fit', FontSize=16);
legend(FontSize=16);
grid on;

%% Evaluation of the fit quality

% Residual Analysis
residuals_poly = Mt_filtered - Mt_fit_poly;

%% Goodness-of-Fit Metrics
SSE = sum(residuals_poly.^2); % Sum of Squared Errors
RMSE = sqrt(mean(residuals_poly.^2)); % Root Mean Squared Error
MAE = mean(abs(residuals_poly)); % Mean Absolute Error
R_squared = 1 - (SSE / sum((Mt_filtered - mean(Mt_filtered)).^2)); % R-squared

fprintf('Goodness-of-Fit Metrics for Polynomial Model:\n');
fprintf('SSE = %.4f\n', SSE);
fprintf('RMSE = %.4f\n', RMSE);
fprintf('MAE = %.4f\n', MAE);
fprintf('R_squared = %.4f\n', R_squared);

%% Residual Plot
figure;
plot(time_filtered, residuals_poly, 'o', 'DisplayName', 'Residuals from Polynomial
xlabel('Time (s)', FontSize=16);
ylabel('Residuals', FontSize=16);
title('Residual Plot for Polynomial Fit', FontSize=16);
legend(FontSize=16);
grid on;
```

B.1.6 Broken Stick Model

```
% Load data from the Excel file
data = readtable("test_2.csv");

%% CLEANING

% Clean and convert cells to numbers
time = seconds(data.Var1); % time
Mt = cellfun(@cleanNumeric, table2cell(data(:, 2)), 'UniformOutput', false); % init
Area_side = cellfun(@cleanNumeric, table2cell(data(:, 3)), 'UniformOutput', false);
Area_top = cellfun(@cleanNumeric, table2cell(data(:, 4)), 'UniformOutput', false); %
```

```
% Convert cleaned cell arrays to numeric matrices
Mt = cell2mat(Mt);
Area_side = cell2mat(Area_side);
Area_top = cell2mat(Area_top);

% Rescale time if necessary
time_rescaled = time / 100000;

%% OUTLIER REMOVAL

% % TEST 3: Mahalanobis distance-based outlier removal
% % Calculate Mahalanobis distance
% dataMatrix = [time_rescaled, Mt, Area_side, Area_top];
% meanData = mean(dataMatrix);
% covData = cov(dataMatrix);
% mahalDist = mahal(dataMatrix, dataMatrix);
%
% % Determine the threshold for identifying outliers
% threshold = chi2inv(0.81, size(dataMatrix, 2));
%
% % Identify outliers
% isOutlier = mahalDist > threshold;
%
% % Filter the data to remove outliers
% time_filtered = time_rescaled(~isOutlier);
% Mt_filtered = Mt(~isOutlier);
% Area_side_filtered = Area_side(~isOutlier);
% Area_top_filtered = Area_top(~isOutlier);
%
% % Display the number of outliers removed
% fprintf('Outliers removed: %d\n', sum(isOutlier));

% TEST 4 i 2: Time-based outlier removal
time_threshold = 0.57 %0.57; % 0.48400
valid_indices = time_rescaled <= time_threshold;

% Filter the data
time_filtered = time_rescaled(valid_indices);
Mt_filtered = Mt(valid_indices);
Area_side_filtered = Area_side(valid_indices);
Area_top_filtered = Area_top(valid_indices);
```

```
% Display the number of samples removed
fprintf('Samples removed (time > %.3f s): %d\n', time_threshold, sum(~valid_indices)

% Plot the data with and without outliers for verification
figure;
subplot(2,1,1);
plot(time_rescaled, Mt, 'o');
title('Original Data', FontSize=16);
xlabel('Time (s)',FontSize=16);
ylabel('Mass',FontSize=16);
grid on;

subplot(2,1,2);
plot(time_filtered, Mt_filtered, 'o');
title('Data After Outlier Removal',FontSize=16);
xlabel('Time (s)',FontSize=16);
ylabel('Mass',FontSize=16);
grid on;

%% MODEL FITTING

% Normalize time to start from 0
time_filtered = time_filtered - time_filtered(1);

% Polynomial model (2nd degree)
poly_coeffs = polyfit(time_filtered, Mt_filtered, 2);
poly_model = @(t) polyval(poly_coeffs, t);

% Initial guess for the breakpoint tc (critical time)
initial_tc_guess = time_filtered(floor(length(time_filtered)/2)); % Use midpoint of

% Linear model Mt = a1 * t + a0 using only data after the initial tc
linear_coeffs = polyfit(time_filtered(time_filtered > initial_tc_guess), ...
    Mt_filtered(time_filtered > initial_tc_guess), 1);
linear_model = @(t) polyval(linear_coeffs, t);

% Broken stick model defined as a function of tc
broken_stick_model = @(params, t) (t <= params(1)).*polyval(poly_coeffs, t) + ...
                                  (t > params(1)).*(polyval(linear_coeffs, t) - poly

% The term (polyval(linear_coeffs, t) - polyval(linear_coeffs, params(1)) + polyval(
% ensures that the linear part of the model starts where the polynomial model ends
```

```
% Bounds for tc
lower_bounds = min(time_filtered); % tc should not be less than the minimum time
upper_bounds = max(time_filtered); % tc should not be greater than the maximum time

% Optimization of tc using lsqcurvefit
options = optimset('Display', 'off'); % Suppress output
[optimized_params, resnorm] = lsqcurvefit(broken_stick_model, initial_tc_guess, time

% Extraction of the optimized tc
optimized_tc = optimized_params(1);

% Optimized value of tc displayed
disp(['Optimized tc: ', num2str(optimized_tc)]);

% Recalculation of the broken stick model using the optimized tc
optimized_broken_stick_model = @(t) broken_stick_model(optimized_params, t);

%% PLOT RESULTS

% Plot of the data, polynomial model, linear model, and optimized broken stick model
figure;
scatter(time_filtered, Mt_filtered, 'blue', 'DisplayName', 'Data');
hold on;

% Plot of the polynomial model
fplot(poly_model, [min(time_filtered), max(time_filtered)], 'black', 'LineWidth', 2

% Plot of the linear model (only after tc)
fplot(@(t) linear_model(t), [optimized_tc, max(time_filtered)], 'green', 'LineWidth

% Plot of the optimized broken stick model
fplot(optimized_broken_stick_model, [min(time_filtered), max(time_filtered)], 'mager

xlabel('Time (seconds)', FontSize=16);
ylabel('Weight (g)', FontSize=16);
title('Data Fitting with Polynomial, Linear, and Optimized Broken Stick Models', For
legend(FontSize=16);
hold off;

%% DISPLAY FITTED PARAMETERS

% Display the coefficients of the polynomial model
disp('Polynomial Model Coefficients:');
```

```
disp(['a2 (Quadratic Coefficient): ', num2str(poly_coeffs(1))]);
disp(['a1 (Linear Coefficient): ', num2str(poly_coeffs(2))]);
disp(['a0 (Intercept): ', num2str(poly_coeffs(3))]);

% Display the coefficients of the linear model
disp('Linear Model Coefficients (After tc):');
disp(['a1 (Slope): ', num2str(linear_coeffs(1))]);
disp(['a0 (Intercept): ', num2str(linear_coeffs(2))]);

% Display the optimized parameters of the broken stick model
disp('Piecewise Model Parameters:');
disp(['tc (Optimized Critical Time): ', num2str(optimized_tc)]);

%% Evaluation of the fit quality

% Calculation of the fitted values for both polynomial and broken stick models
Mt_fit_poly = poly_model(time_filtered); % Fitted values using the polynomial model
Mt_fit_broken_stick = optimized_broken_stick_model(time_filtered); % Fitted values u

% Residual Analysis
residuals_poly = Mt_filtered - Mt_fit_poly; % Residuals for the polynomial fit
residuals_broken_stick = Mt_filtered - Mt_fit_broken_stick; % Residuals for the brok

%% Goodness-of-Fit Metrics for the Polynomial Fit
SSE_poly = sum(residuals_poly.^2); % Sum of Squared Errors
RMSE_poly = sqrt(mean(residuals_poly.^2)); % Root Mean Squared Error
MAE_poly = mean(abs(residuals_poly)); % Mean Absolute Error
R_squared_poly = 1 - (SSE_poly / sum((Mt_filtered - mean(Mt_filtered)).^2)); % R-squ

fprintf('Goodness-of-Fit Metrics for the Polynomial Model:\n');
fprintf('SSE (Polynomial) = %.4f\n', SSE_poly);
fprintf('RMSE (Polynomial) = %.4f\n', RMSE_poly);
fprintf('MAE (Polynomial) = %.4f\n', MAE_poly);
fprintf('R_squared (Polynomial) = %.4f\n', R_squared_poly);

%% Goodness-of-Fit Metrics for the Broken Stick Model
SSE_broken_stick = sum(residuals_broken_stick.^2); % Sum of Squared Errors
RMSE_broken_stick = sqrt(mean(residuals_broken_stick.^2)); % Root Mean Squared Erro
MAE_broken_stick = mean(abs(residuals_broken_stick)); % Mean Absolute Error
R_squared_broken_stick = 1 - (SSE_broken_stick / sum((Mt_filtered - mean(Mt_filtered

fprintf('\nGoodness-of-Fit Metrics for the Broken Stick Model:\n');
fprintf('SSE (Broken Stick) = %.4f\n', SSE_broken_stick);
```

```
fprintf('RMSE (Broken Stick) = %.4f\n', RMSE_broken_stick);
fprintf('MAE (Broken Stick) = %.4f\n', MAE_broken_stick);
fprintf('R_squared (Broken Stick) = %.4f\n', R_squared_broken_stick);

%% Residual Plot
figure;
hold on;
plot(time_filtered, residuals_poly, 'o', 'DisplayName', 'Residuals (Polynomial Fit)
plot(time_filtered, residuals_broken_stick, 'x', 'DisplayName', 'Residuals (Broken S
xlabel('Time (s)', FontSize=16);
ylabel('Residuals', FontSize=16);
title('Residual Plot for Polynomial and Broken Stick Models', FontSize=16);
legend(FontSize=16);
grid on;
hold off;
```