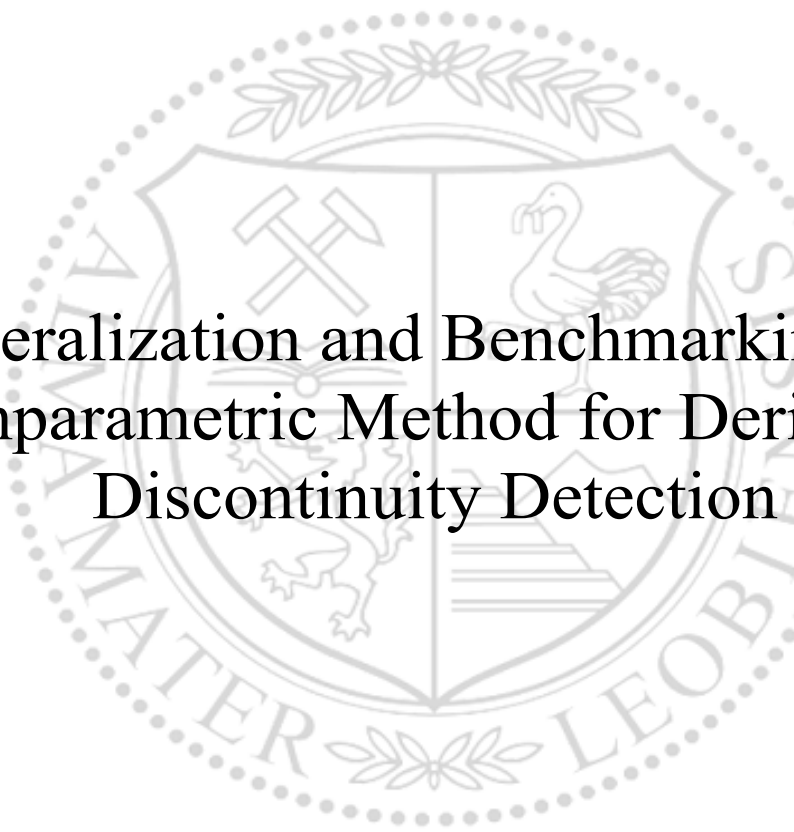




Chair of Automation

Master's Thesis



Generalization and Benchmarking of a
Nonparametric Method for Derivative
Discontinuity Detection

Karl Heinz Deutsch, BSc (WU) BSc

March 2021



Lehrstuhl für Automation

Masterarbeit

Generalisierung und Evaluierung einer
nicht-parametrischen Methode zur
Erkennung von Unstetigkeiten in
Sensordaten

Karl Heinz Deutsch, BSc (WU) BSc

März 2021

Danksagung - Dedication

Diese Arbeit wurde im Wintersemester 2020/21 am Lehrstuhl für Automation der Montanuniversität Leoben während der COVID-19-Pandemie geschrieben. Ein Umstand der zu ungewöhnlichen Bedingungen für das Verfassen der Arbeit führte. Folglich möchte ich mich besonders bei Professor O’Leary bedanken, der stets versuchte die Betreuung unter Anbetracht der Umstände so normal wie möglich zu gestalten. Dabei waren insbesondere die wöchentlichen Besprechungen mit allen Kollegen und Betreuern von besonderem Wert, um durchwegs fokussiert zu bleiben. Selbstverständlich bin ich auch sehr dankbar darüber die Möglichkeit zur Bearbeitung dieses Themas bekommen zu haben.

Besonderer Dank gebührt zudem meinem Mitbetreuer Dimitar Ninevski, welcher immer verfügbar war, um offene Fragen zu besprechen, mir durchwegs hilfreiche Empfehlungen zur Bearbeitung meines Themas gab und mir zudem äußerst schnell sehr detailliertes Feedback auf die von mir verfassten Teilleistungen gab. Selbstverständlich ebenfalls sehr dankbar bin ich meinen Eltern, meinem Bruder und meiner Freundin, die mich während meiner Masterarbeit und meines Studiums stets in jeglicher Form unterstützt haben. Zudem danke ich allen Kollegen und Freunden die während meiner Studienzeit für mich da waren.

This thesis was written in the winter semester of 2020/21 at the Chair of Automation, during the COVID-19 pandemic. Hence, the conditions have been quite unusual. Therefore, I would like to thank professor O’Leary especially for the good handling of the situation during this time. Weekly meetings with all fellow students, supervisors and him, have been really valuable to keep focused. Of course, I am also thankful for the opportunity to write this thesis at the institute in general.

Special thanks are, of course, dedicated to my additional supervisor Dimitar Ninevski, who was always available to discuss even the smallest issues, gave very good recommendations and delivered detailed feedback to parts of my work in an outstanding speed on a regular basis. Furthermore, I would also like to thank my parents, my brother and my girlfriend for always supporting me in any kind, both during the process of writing this thesis and throughout my entire time as a student. Thank you very much. I would also like to thank all peers who supported me in one way or the other.



EIDESSTÄTTLICHE ERKLÄRUNG

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

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

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

Datum 05.03.2021

Unterschrift Verfasser/in
Karl Heinz Deutsch

Kurzfassung

Diese Arbeit beschäftigt sich mit der Erkennung von Unstetigkeiten in Echtzeit-Sensordaten von Anlagen und Maschinen. Das ist insbesondere von Bedeutung, da physikalische Systeme deren Verhalten über Differentialgleichungen beschrieben wird, ausschließlich kontinuierlicher Funktionen und Ableitungen ausgesetzt sein sollten. Hierbei baut diese Arbeit auf Forschungen im Bereich der Erkennung von C^n -Unstetigkeiten auf, wobei in weiterer Folge eine Verallgemeinerung des Ansatzes durchgeführt wird. Dabei ermöglicht der erweiterte Ansatz, bei einer Menge von n Ableitungen einer Funktion, jene n Ordnungen zu definieren, welche auf Unstetigkeiten untersucht werden sollen.

Alle dazu notwendigen Herleitung werden in Matrixschreibweise bereitgestellt. Anschließend erfolgt die numerische Umsetzung der erweiterten Methode, welche mittels synthetischer und realer Datensätze getestet wird. Im Zuge dieser Evaluierung wurde die Detektionsmethode mit anderen gängigen Algorithmen aus diesem Feld gegenübergestellt. Dabei wurden zwei, in der Literatur verwendete, Leistungskennzahlen herangezogen.

Der neue Ansatz kann als allgemeinste Formulierung der Detektionsmethoden, die in dieser Arbeit behandelt wurden, gesehen werden, da keine genauere Spezifizierung des Anwendungsfalls erforderlich ist. Generell basiert dieser auf einer allgemeinen mathematischen Formulierung einer Unstetigkeit, welche eine Verallgemeinerung einer C^n -Unstetigkeit darstellt. Die durchgeführte Evaluierung zeigt, dass der Detektions-Algorithmus, im verwendeten Testumfeld, überdurchschnittliche Ergebnisse aufweist. Dabei ist allerdings anzumerken, dass einzelne andere Anwendungen, durchaus höhere Kennzahlen lieferten. Zusätzlich zu erwähnen ist hierbei, dass die verwendeten Testdatensätze ausschließlich C^0 - und C^1 -Unstetigkeiten aufweisen. Der neue Ansatz, zeigt zwar gute Ergebnisse bei diesen, es ist allerdings davon auszugehen, dass er für Unstetigkeiten höherer Ordnung besser geeignet ist.

Abstract

This thesis addresses the issue of detecting discontinuities in real-time observational data from plant and machinery. This is highly relevant, since systems whose dynamics are well modelled by differential equations should exhibit continuity in the real-time signals and their derivatives. This work builds upon previous research into the detection of C^n discontinuities and extends it to a more general case. Considering a set of n derivative orders, the new approach permits defining which of these n orders are to be inspected for discontinuities.

All the derivations for the method, based on matrix algebraic formulations, are provided. Furthermore, a numerical solution for the method is implemented. Testing has been performed with a wide set of data sets derived from strongly differing areas of application. Performance estimates are computed using two different metrics and are compared with the results from other discontinuity and change detection methods.

This new approach is the most generic of all the methods considered, since it does not require application specific adaption. It is based on a formal mathematical definition of a discontinuity, which is a generalization of a C^n discontinuity. The comparative results show that the algorithm, on average, outperforms the other methods. There are, however, specific cases where the application specific methods perform better. Previous literature and test datasets only consider C^0 and C^1 type discontinuities. Whereas, the new approach, not only performs well for these type of discontinuities, but also functions for higher order derivative discontinuities.

Contents

1	Introduction	1
2	Algebraic Fundamentals	3
2.1	Polynomials	3
2.2	Polynomial Interpolation	5
2.3	Polynomial Approximation	7
2.4	Constrained Polynomial Approximation	10
2.5	Computational Variants of the Moore-Penrose Pseudo Inverse	13
2.5.1	QR decomposition	13
2.5.2	Singular Value Decomposition	13
2.6	Covariance Propagation	15
3	Time Series Analysis	17
3.1	Challenges in Time Series Processing	18
3.2	Change Point Detection	20
3.2.1	Mean-shift	20
3.2.2	Variance Change	21
3.2.3	Regression-slope switch	22
3.2.4	Distribution change	23
3.3	Detection Methods in Change Point Analysis	24
3.3.1	Parametric Methods	26
3.3.2	Non-parametric Methods	26
3.3.3	Bayesian Methods	27
3.3.4	Sequential Analysis	29
3.4	Condition Monitoring	33
4	Derivative Discontinuity Detection via Constrained Local Polynomial Approximation	35
4.1	Algebraic Discontinuities	35
4.2	Constrained Coupled Polynomial Approximation	37

Contents	vi
4.2.1 General Principle	38
4.2.2 Covariance Propagation	41
4.2.3 Error Analysis	42
5 Generalization of the Constrained Polynomial Discontinuity Detection Approach	47
5.1 Detection of D^n Discontinuities	47
5.2 Synthetic Datasets	50
5.2.1 D^2D^0 - Dataset	50
5.2.2 $D^3D^1D^0$ - Dataset	52
5.2.3 $D^4D^3D^1D^0$ - Dataset	54
5.3 Triple Peak Feature	55
5.3.1 D^2D^0 - Dataset	56
5.3.2 $D^3D^1D^0$ - Dataset	60
5.4 Multiple Discontinuities	62
5.5 Noise Testing	64
5.5.1 $D^2D^0 - Noise$ - Dataset	64
5.5.2 $D^3D^1D^0 - Noise$ - Dataset	67
5.5.3 Noise Impact - Monte Carlo Simulation	72
6 Benchmark Current Version with other Change Point Detection Methods	75
6.1 Real World Datasets	75
6.2 Evaluation Metrics	75
6.2.1 Segmentation Covering Metric	76
6.2.2 F1 - Score	78
6.3 Benchmark Testing	79
7 Conclusion	83
List of figures	84
List of tables	86
Bibliography	
A Detailed Results of the Benchmark Tests	92
B Mean-shift - Discontinuity Detection Plots	94
C Slope-Switch - Discontinuity Detection Plots	132

Chapter 1

Introduction

A model of a physical system, bound to the laws of physics, is directly part of a data science environment dependent on the provided measurement data to the system. Hence, abrupt changes in the observed data could lead to an abnormal behaviour of the model. An area in that context which seems to attract quite some attention in the recent past is the so called *physics informed data science* [2], [3]. For further illustration of the described issue consider the following example of the differential equation of the simple gravity pendulum,

$$\frac{\partial^2 \theta(t)}{\partial t^2} + \frac{g}{l} \sin(\theta(t)) = 0. \quad (1.1)$$

Hereby, a discontinuity in $\frac{\partial^2 \theta(t)}{\partial t^2}$ - a second order discontinuity - would lead to a completely false behaviour of the model given in Equation 1.1. Thus, the detection of such changes might be of interest. Exactly such a detection problem can be considered as the main subject of a field called *change point* or *discontinuity detection*, where mathematical concepts are applied in order to detect changes in data. From an algebraic perspective the literature distinguishes between three different types of discontinuities in univariate real valued functions. Namely, those are removable, jump and infinite discontinuities. A more in-depth definition of the different types is going to be executed later in this thesis. As a result of higher practical relevance the major focus of this work is on the detection of jump discontinuities of functions and their derivatives. Irregularities of that type can affect the behaviour of a model describing a physical system by differential equations as examined in [1].

Additional use cases for discontinuity detection would be segmented regression [4], detection of specific events for example in economic time series data [5], geophysics [6] or image processing [7], [8]. Considering the field of image processing, the detection of jumps is equivalent to the finding of edges in an image. Likewise, possible change points in the derivative of image data may be of interest in the task of edge detection.

Even though, this thesis is primarily dealing with a non-parametric discontinuity detection method with constrained local polynomial approximation, an outline of the field of change point detection in general as part of time series analysis is given briefly in Chapter 3. However, before the different approaches for change point detection are introduced, in Chapter 2 an outline for the relevant algebraic concepts necessary for this thesis is made. Afterwards, the derivative discontinuity detection method proposed in [1] is presented in Chapter 4.

Subsequently, a more general approach for the introduced method is described in Chapter 5, which is then tested with synthetic datasets, generated for the evaluation of change point detection algorithms. Whereas, the results are compared to the first version of the detection approach. After the application of the method to datasets without noise, the impact of noise to the detection method is also tested with a Monte Carlo simulation.

Finally, in Chapter 6 the method is compared to other algorithms used in the field of change point analysis. This comparison is performed with an evaluation framework dedicated to measure the performance of change detection algorithms tested with a specific dataset collection.

At the end, Chapter 7 gives a conclusion of the thesis and some possible future tasks are proposed.

Chapter 2

Algebraic Fundamentals

In this chapter some algebraic fundamentals, which are required for the derivative discontinuity detection method later in this thesis, are introduced.

2.1 Polynomials

As a highly applied and very popular set of functions the *algebraic polynomials* can be defined as functions which map the set of real numbers into itself. [9]

Generally, a polynomial $y(x, \alpha)$, as the weighted sum of *monomials*, is formulated as,

$$y(x, \alpha) = \alpha_d x^d + \alpha_{d-1} x^{d-1} + \dots + \alpha_1 x^1 + \alpha_0, \quad (2.1)$$

with $d \in \mathbb{N}$ being a non-negative integer. The weights used for each monomial are called polynomial coefficients $\alpha = [\alpha_d \ \alpha_{d-1} \ \dots \ \alpha_0]^T$, which are real valued constants, $\alpha \in \mathbb{R}$. Alternatively, *the sum notation* is also commonly used with,

$$y(x, \alpha) = \sum_{i=0}^d \alpha_i x^i. \quad (2.2)$$

As a broadly applied way to approximate data points, *polynomials* are well known in a lot of different subjects and fields. This mainly comes from the fact that they are relatively flexible in their application and pretty easy to understand.

Concerning the usage of polynomials in general two theorems are essential. One of which is the *Weierstrass Theorem* [10] describing the application range of polynomials.

Theorem 1. (*Weierstrass approximation theorem*) *A real continuous function $f = f(x)$ over an interval $x \in [a, b]$, can be approximated by a polynomial $p(x)$ such that the supremum norm*

$$\|f(x) - p(x)\|_\infty < \varepsilon, \text{ with } \varepsilon > 0, \text{ for } a \leq x \leq b. \quad (2.3)$$

This means that it is possible to approximate a continuous function $f(x)$ by a polynomial $p(x)$ with arbitrary accuracy ε as long as the degree of the polynomial is high enough.

The second fundamental theorem stated here is *Taylor's theorem*, which proposes a way to determine a n^{th} order polynomial approximation around a location $x = a$ of a continuous function $f = f(x)$. This is accomplished by evaluating the respective derivatives of the function around the point $x = a$.

Theorem 2. (*Taylor's theorem*) *For a continuous function $f = f(x)$ with $n - 1$ continuous derivatives, the function can be defined as,*

$$f(x) = f(a) + \frac{f^{(1)}(a)}{1!}(x-a) + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n(x,a), \quad (2.4)$$

whereas, the theorem also delivers an estimate of the residual error ε with,

$$\varepsilon = R_n(x,a) = \int_a^x \frac{f^{(n+1)}(t)}{n!}(x-t)^n dt \quad (2.5)$$

Consequently, the *Maclaurin series* is defined as a *Taylor series* evaluated at the point $x = 0$. In other words, it can be constructed by performing the coordinate transformation $y = x - a$ and subsequently evaluate the approximation at the point y with,

$$f(y) = f(0) + \frac{f^{(1)}(0)}{1!}y + \dots + \frac{f^{(n)}(0)}{n!}y^n + R_n(y,0). \quad (2.6)$$

So, by considering the formulation of the *Maclaurin expansion* in Equation 2.6 compared to the definition of a polynomial in Equation 2.1 it can be shown that the polynomial coefficients α are equivalent to the *Taylor coefficients*, both evaluated around the origin. Now a Taylor expansion the first d terms delivers an estimate $\tilde{f}(x)$ for a respective continuous function $f = f(x)$ with at the point a ,

$$\tilde{f}(x) = \sum_{i=0}^d \frac{f^{(i)}(a)}{i!}(x-a)^i. \quad (2.7)$$

Thus, the estimate $\tilde{f}(x)$ delivered by the *Maclaurin expansion* at the point $a = 0$ would then be,

$$\tilde{f}(x) = \sum_{i=0}^d \frac{f^{(i)}(0)}{i!}(x)^i. \quad (2.8)$$

with,

$$t_f^{(i)} \triangleq \frac{f^{(i)}(0)}{i!}. \quad (2.9)$$

Whereas, $t_f^{(i)}$ is the i^{th} order *Maclaurin coefficient*. Then, as already stated above, by performing the polynomial approximation at the origin of the coordinate system the polynomial coefficients α

are closely related to the corresponding *derivatives* at that point. Hence, the connection between the *Maclaurin coefficients* t_f and *polynomial coefficients* α can be made by,

$$t_f^{(i)} = \alpha_i = \frac{f^{(i)}(0)}{i!} \quad (2.10)$$

Another way to formulate a polynomial is in *matrix form*. In particular this variant is of importance when computational resources are required to perform calculations with polynomials, since many programming languages, such as *mcode* in MATLAB or *Python*, support matrix notation directly. Hence, the application of polynomials gets easier for the user. Considering the notation variants above, a general polynomial is denoted as,

$$\begin{aligned} \mathbf{y}(\mathbf{x}, \boldsymbol{\alpha}) &= \begin{bmatrix} x^d & x^{d-1} & \dots & 1 \end{bmatrix} \begin{bmatrix} \alpha_n \\ \alpha_{n-1} \\ \vdots \\ \alpha_0 \end{bmatrix} \\ &= \mathbf{v}_d(\mathbf{x})\boldsymbol{\alpha} \end{aligned} \quad (2.11)$$

in matrix form, where the vector $\mathbf{v}_d(\mathbf{x})$ stores all d powers of x and can be part of a bigger matrix, if the polynomial $y(x, \boldsymbol{\alpha})$ is evaluated at n sample points. Hereby, n row vectors with the structure of $\mathbf{v}_d(\mathbf{x})$ are combined to the so called *Vandermonde matrix*

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_d(\mathbf{x}_1) \\ \mathbf{v}_d(\mathbf{x}_2) \\ \vdots \\ \mathbf{v}_d(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} x_1^d & x_1^{d-1} & \dots & x_1 & 1 \\ x_2^d & x_2^{d-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^d & x_n^{d-1} & \dots & x_n & 1 \end{bmatrix}. \quad (2.12)$$

Which finally leads to the handy formulation for a set of n polynomial evaluation in matrix form with,

$$\mathbf{y} = \mathbf{V}\boldsymbol{\alpha}. \quad (2.13)$$

2.2 Polynomial Interpolation

Now that the principle formulation is defined, the subsequent part of this subchapter will focus on general application examples of polynomials and the impact of the chosen *degree* d and the number of sample points n .

Here mainly three different cases are possible. One of which is defined with,

$$n = d + 1. \quad (2.14)$$

Here the solution is *exactly determined* and the evaluation of the polynomial is also referred to as *polynomial interpolation*. A classical application example of this special case is the calibration of a sensor for example, where only some measurement points are known, but the range within those points is also of interest. The polynomial coefficients α_d are then derived by,

$$\alpha_d = \mathbf{V}^{-1} \mathbf{y}. \quad (2.15)$$

The Vandermonde matrix \mathbf{V} is always invertible in the *determined* case. In general, this is true for Vandermonde matrices of the form,

$$\mathbf{V} = \begin{bmatrix} x_1^d & x_1^{d-1} & \dots & x_1 & 1 \\ x_2^d & x_2^{d-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^d & x_n^{d-1} & \dots & x_n & 1 \end{bmatrix} \in \mathbb{R}^{n \times (d+1)}, \text{ with } n = d + 1 \quad (2.16)$$

because the column vectors are linearly independent and the structure of the matrix is squared. The fact that the columns of the Vandermonde matrix \mathbf{V} are linearly independent is supported by the matter that n unique points are sampled, $x_i \neq x_j, \forall i \neq j$. Note that this circumstance is of theoretical nature. Numerically, due accuracy limits, columns of the matrix \mathbf{V} can be linearly depended.

Even tough inverting the matrix is possible, the computational effort can get very high, since the inversion of matrices is of $O(n^3)$ time complexity.

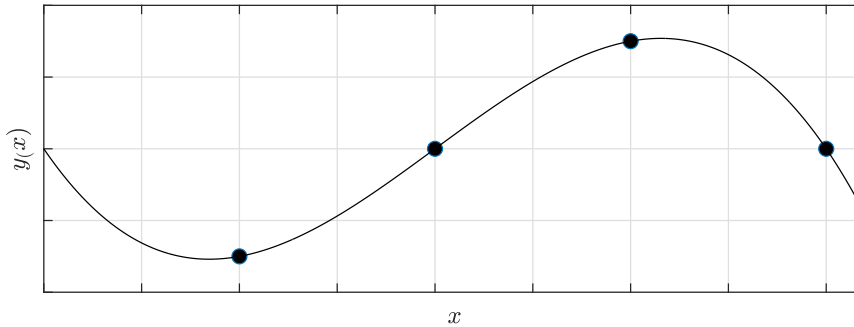


Fig. 2.1: Example of a polynomial interpolation with $n = 4$ sample points and a polynomial of degree $d = 3$.

The second variant on the other hand delivers an *under-determined solutions* and the approximation of the given sample points can be done not only by one polynomial, but by a *family of polynomials*. This scenario is given by,

$$n < d + 1 \quad (2.17)$$

This would then lead to a *system of polynomial equations* with more unknowns $d + 1$ than equations n . Where the unknowns describing the degrees of freedom of the system and the equations constraining these. In other words, in the under-determined case at least one degree of freedom remains.

Hereby, the Vandermonde matrix \mathbf{V} is of the form,

$$\mathbf{V} = \begin{bmatrix} x_1^d & x_1^{d-1} & \dots & x_1 & 1 \\ x_2^d & x_2^{d-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^d & x_n^{d-1} & \dots & x_n & 1 \end{bmatrix} \in \mathbb{R}^{n \times (d+1)}, \text{ with } n < d + 1, \quad (2.18)$$

which has a non-square structure $[n \times (d + 1)]$ and is therefore not invertible.

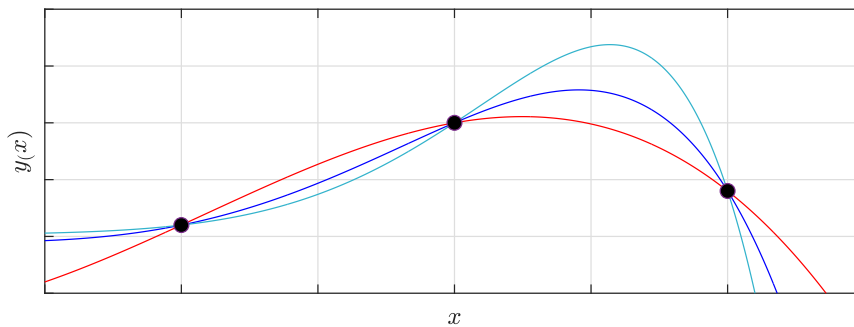


Fig. 2.2: Example of a polynomial interpolation with $n = 3$ sample points and polynomials of degree $d_1 = 3$ (red), $d_2 = 5$ (blue) and $d_3 = 7$ (green).

Various polynomials can be used to interpolate between sample points in the under-determined case, as shown in Figure 2.2. In fact, the family of polynomials consists of infinitely many variants.

2.3 Polynomial Approximation

Then the third and final case is given if

$$n > d + 1, \quad (2.19)$$

which yields an *overdetermined solution* by an approximated evaluation of the polynomials.

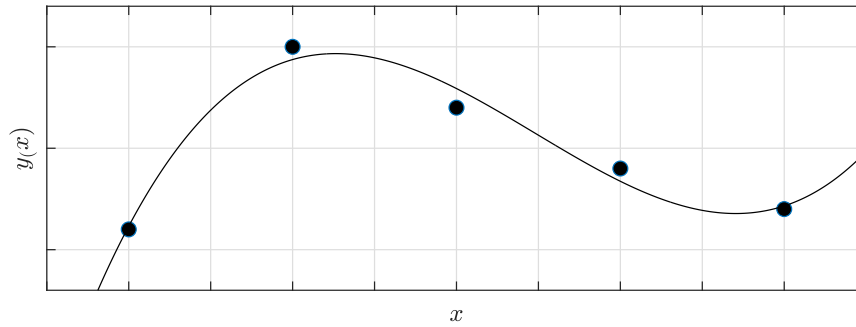


Fig. 2.3: Example of an overdetermined polynomial approximation with $n = 5$ sample points and polynomial of degree $d = 3$, optimized in the least squares sense.

Again, just like in the under-determined case, the structure of the Vandermonde matrix V is *non-square* in the overdetermined case, where $V \in \mathbb{R}^{n \times (d+1)}$ with $n > d + 1$. Consequently, Formula 2.13 has no exact solution, since the V is not invertible.

So, as there is no exact solution to solve for α the best alternative would be to get an approximated result for the coefficients. In general, a measure to rate the quality of the approximation is then also formulated.

Now, a brief introduction is given to the algebraically most straight forward approach - the *normal equations*. An over-determined system of equations can be written as,

$$Ax = b. \quad (2.20)$$

Note that, the system is overdetermined, hence the non-square matrix A is not invertible.

So, as already stated, an exact solution for the given problem setting is not feasible, thus an approximation is performed with the *residual*,

$$r = b - Ax, \quad (2.21)$$

describing the deviation between the approximated values and the actual observations as a measure of error. Consequently, to get as close to an exact solution as possible, the goal is to minimize this residual vector r . This minimization can be achieved by the definition of a *cost function* $\epsilon(x)$, with the *2-norm* of the residual r , as the sum of squared residuals, with,

$$\begin{aligned}
\varepsilon(x) &= \|\mathbf{r}\|_2^2 \\
&= \mathbf{r}^T \mathbf{r} \\
&= (\mathbf{b} - \mathbf{A}\mathbf{x})^T (\mathbf{b} - \mathbf{A}\mathbf{x}) \\
&= (\mathbf{b}^T - \mathbf{x}^T \mathbf{A}^T) (\mathbf{b} - \mathbf{A}\mathbf{x}) \\
&= \mathbf{b}^T \mathbf{b} - \mathbf{b}^T \mathbf{A}\mathbf{x} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{x}^T \mathbf{A}^T \mathbf{A}\mathbf{x}
\end{aligned} \tag{2.22}$$

Now, given that the *cost function* $\varepsilon(x)$ is a scalar, each term of the sum in Equation 2.22 is also a scalar, so the formulation can be simplified even further with,

$$\varepsilon(x) = \mathbf{b}^T \mathbf{b} - 2\mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{x}^T \mathbf{A}^T \mathbf{A}\mathbf{x} \tag{2.23}$$

Furthermore, as the optimization of the approximation is performed with respect to the vector \mathbf{x} , the following condition can be formulated,

$$\begin{aligned}
\frac{\delta \varepsilon(x)}{\delta \mathbf{x}} &= 0 \\
&= -2\mathbf{A}^T \mathbf{b} + 2\mathbf{A}^T \mathbf{A}\mathbf{b}.
\end{aligned} \tag{2.24}$$

Subsequently, the system of linear equations,

$$\mathbf{A}^T \mathbf{A}\mathbf{x} = \mathbf{A}^T \mathbf{b}, \tag{2.25}$$

which are commonly referred to as *normal equations*, can be formulated. Now, in case the condition is fulfilled, that the matrix \mathbf{A} consists of real values only, the approximate solution $\hat{\mathbf{x}}$ can be derived with,

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \tag{2.26}$$

Here, the term $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ can be combined to the so called *Moore-Penrose Pseudo Inverse* \mathbf{A}^+ , which is the most generalized form of the inverse of a matrix, finally yielding the optimized approximation in the least squares sense with,

$$\hat{\mathbf{x}} = \mathbf{A}^+ \mathbf{b}. \tag{2.27}$$

Now, considering the previously described over-determined approximation, the system of polynomial equations can be written as,

$$\mathbf{y} = \mathbf{V}_o \boldsymbol{\alpha}_o. \tag{2.28}$$

Hence, the coefficients can be approximated by,

$$\hat{\alpha}_o = V_o^+ \mathbf{y}, \quad (2.29)$$

which are solving the least squares problem stated as,

$$\min_{\alpha_0} \|\mathbf{y} - V_o^+ \alpha_0\|_2^2. \quad (2.30)$$

Another important factor to keep in mind in this context is the computational efficiency of the approximation. Therefore, in general the Moore-Penrose Inverse A^+ with $(A^T A)^{-1} A^T$, is derived by various approximation techniques, like the *QR* or the *singular value decomposition*. Therefore, both method are briefly introduced in a following subchapter.

2.4 Constrained Polynomial Approximation

Occasionally, it might be the case that an approximation of a set of sample points is required, where some specific values or derivatives are already known before. These a priori known conditions can then be considered in the approximation process by placing constraints on the solution.

Furthermore, four different types of constraints are distinguished.

1. Incomplete sum of monomials, which could for example be,

$$y(x) = \alpha_3 x^3 + \alpha_2 x^2 + \alpha_0. \quad (2.31)$$

In that case, the value of a coefficient could be known to be zero, like $\alpha_1 = 0$ for the example above.

2. Zero constraints, where the evaluated polynomial at the location c is constrained to,

$$y(c) = 0. \quad (2.32)$$

3. Value constraints, with the evaluated polynomial at the location c , implying a value d in the form of,

$$y(c) = d. \quad (2.33)$$

This constraint is also known as *Dirichlet condition*.

4. Differential constraints, enforcing a value d for the p^{th} derivative of the polynomial $y(x)$ at the location c , with,

$$y_{(c)}^{(p)} = d \quad (2.34)$$

In this context, a specific constraint for the first order derivative $p = 1$ is the *Neumann condition* with,

$$y_{(c)}^{(1)} = d \quad (2.35)$$

Furthermore, these different constraints can be implemented into a system of polynomial equations with,

1. Modifying the polynomials, which are forming a vector basis sets spanning the space of the constraints
2. Modifying the coefficients (e.g. by a permutation matrix)
3. Modifying the span and null space of the matrix

In the following, only the third approach is going to be described in more detail, since it is most relevant for this thesis.

Here the constraint t_i is defined by the triplet,

$$t_i = \begin{bmatrix} c_i \\ k_i \\ a_i \end{bmatrix}, \quad (2.36)$$

where c_i is the location of the constraint, k_i is its order and a_i accounts for the respective value. As a highly general formulation t_i enables the implementation of initial, interior or boundary conditions.

Subsequently, the constraints defined by a triple t_i can be implemented into a system of polynomial equations with the constraint matrix C and the value vector a with,

$$C\alpha = a \quad (2.37)$$

Hereby, the constraint matrix C is constructed by m Vandermonde vectors $v_d^{k_i}(c_i)$ of k_i^{th} order at the location c_i

$$C \triangleq \begin{bmatrix} v_d^{k_1}(c_1) \\ v_d^{k_2}(c_2) \\ \vdots \\ v_d^{k_m}(c_m) \end{bmatrix} \quad (2.38)$$

Now the solution of Equation 2.37 returns the coefficients,

$$\alpha = C^+ a + N_C \gamma. \quad (2.39)$$

Here, the matrix N_C forms an orthonormal vector basis set spanning the null space of the matrix C .

Then the previously defined formulation of a polynomial approximation yields,

$$\mathbf{y} = \mathbf{V}\{\mathbf{C}^+ \mathbf{a} + \mathbf{N}_C \boldsymbol{\gamma}\}. \quad (2.40)$$

So, as the general implementation of polynomial constraints has been shown, the following paragraphs are dealing with a special case of constraints for a system of polynomial equations, which is relevant for the discontinuity detection method introduced later in this work. Namely, the special case is the implementation of *zero constraints* into the system. Hereby, Equation 2.37 can be formulated with,

$$\mathbf{C}\boldsymbol{\alpha} = \mathbf{0}, \quad (2.41)$$

as the value vector \mathbf{a} is equal to the zero vector. What is more, the polynomial coefficients are then derived by,

$$\boldsymbol{\alpha} = \mathbf{N}_C \boldsymbol{\gamma}, \quad (2.42)$$

and the polynomial approximation results in,

$$\mathbf{y} = \mathbf{V}\mathbf{N}_C \boldsymbol{\gamma}, \quad (2.43)$$

which is a new system of polynomial equations. Hence, considering the over-determined case, no exact solution is available. So, the task here is once more to find the vector $\boldsymbol{\gamma}$ which minimizes the least squares problem stated with,

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{V}\mathbf{N}_C \boldsymbol{\gamma}\|_2^2. \quad (2.44)$$

As described previously, the optimal solution in the least squares sense can be achieved by the pseudo-inverse with,

$$\boldsymbol{\gamma} = (\mathbf{V}\mathbf{N}_C)^+ \mathbf{y} \quad (2.45)$$

Thus, the polynomial coefficients $\boldsymbol{\alpha}$ considering the implemented constraints can be computed with,

$$\boldsymbol{\alpha} = \mathbf{N}_C (\mathbf{V}\mathbf{N}_C)^+ \mathbf{y}. \quad (2.46)$$

Hereby, the necessary pseudo-inverse can be computed by the normal equation method or different approximation methods. Whereas, two representatives of the later are part of the following subsection.

2.5 Computational Variants of the Moore-Penrose Pseudo Inverse

Since the method of deriving the *Moore-Penrose pseudo-inverse* of a matrix A directly from the *normal equations* is not the most efficient approach, various methods are used to approximate it. In this context, two main methods are broadly used. Namely, those are the *QR decompositions* on the one hand and the *singular value decomposition* on the other side.

2.5.1 QR decomposition

By applying the *QR decomposition* to a matrix A , this matrix is decomposed into,

$$A = QR. \quad (2.47)$$

Here the matrix Q is orthonormal with, $Q^T Q = I$, and R is of *upper triangular* form. Now, considering the formulation for the Moore-Penrose pseudo-inverse A^+ from Equation 2.26 yields,

$$A^+ = (R^T Q^T Q R)^{-1} (Q R)^T \quad (2.48)$$

Now, since the matrix Q is orthonormal, the formulation can be simplified to,

$$\begin{aligned} A^+ &= (R^T R)^{-1} (Q R)^T \\ &= R^+ Q^T. \end{aligned} \quad (2.49)$$

Finally, the coefficients, as stated in Equation 2.29, for the over-determined case can be derived by,

$$\alpha_o = R^+ Q^T y \quad (2.50)$$

Although, the pseudo-inverse R^+ need to be computed, this can be achieved simply through back substitution, as the matrix R is of upper triangular structure.

2.5.2 Singular Value Decomposition

Another possible way to compute the pseudo inverse of a matrix is to use the *singular value decomposition (SVD)*. Besides the ability to compute the pseudo-inverse of a matrix efficiently, the SVD is also broadly used for *dimensionality reduction of high-dimensional data* or to *denoise datasets*. The following section should give a quick introduction of the method and show the application

in terms of pseudo-inverse computation. Of course, there is a lot of literature about the SVD, for example a good overview is given in [11].

In general, the *SVD* of matrix $\mathbf{A} \in \mathbb{C}^{n \times m}$ delivers a distinct matrix decomposition in the form,

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^*. \quad (2.51)$$

Where the matrices $\mathbf{U} \in \mathbb{C}^{n \times n}$ and $\mathbf{V} \in \mathbb{C}^{m \times m}$ are *unitary* and their columns are *orthonormal*. The matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$ on the other side, is a diagonal matrix with *real* and *nonnegative* values. Note that the form \mathbf{V}^* is the *conjugate transpose*, which is the equivalent of the regular formulation \mathbf{V}^T of a transpose in the *complex space* \mathbb{C} .

Even though, a matrix \mathbf{A} could be described exactly with the SVD, a more practical approach - if not the most important one - is to approximate this matrix. In fact, the decomposition method delivers *optimal low-rank approximation* for a matrix \mathbf{A} as stated by the *Eckhart-Young theorem* [12], which postulates,

Theorem 3. (Eckhart-Young [12]) *The optimal rank- r approximation to \mathbf{A} , in a least squares sense, is given by the rank- r SVD truncation $\tilde{\mathbf{A}}$ with,¹*

$$\underset{\tilde{\mathbf{A}}, s.t. \text{rank}(\tilde{\mathbf{A}})=r}{\operatorname{argmin}} \quad \|\mathbf{A} - \tilde{\mathbf{A}}\|_F = \tilde{\mathbf{U}} \tilde{\mathbf{\Sigma}} \tilde{\mathbf{V}}^*. \quad (2.52)$$

Note that the matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}^*$ are constructed by the first r columns of \mathbf{U} or \mathbf{V} respectively. The estimate $\tilde{\mathbf{\Sigma}}$ on the other hand, is established by the first $r \times r$ sub-matrix of the matrix $\mathbf{\Sigma}$. In general, $\mathbf{\Sigma}$ in combination with a predefined threshold is used to define r . Here, the first r entries on the diagonal of $\mathbf{\Sigma}$ bigger than the threshold are considered. This approximation variant is then called the *truncated SVD*, which yields an estimate of a matrix \mathbf{A} with,

$$\mathbf{A} \approx \tilde{\mathbf{U}} \tilde{\mathbf{\Sigma}} \tilde{\mathbf{V}}^* \quad (2.53)$$

Consequently, the Moore-Penrose pseudo-inverse \mathbf{A}^+ - or an estimate $\tilde{\mathbf{A}}^+$ - can be derived from the SVD, by inverting each of the matrices separately resulting in,

$$\tilde{\mathbf{A}}^+ \triangleq \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}^* \quad (2.54)$$

This estimate can afterwards be used to derive an optimized solution of a over-determined system of linear or polynomial equations in the least squares sense, just like described previously with the *normal equations* approach. This method, however, is computational more efficient. One application example of the SVD in this context might be the *MATLAB* function *pinv()*, which delivers the Moore-Penrose pseudo-inverse of a matrix by making use of the truncated SVD.

¹ $\|\cdot\|_F$ denotes as the *Frobenius norm*

2.6 Covariance Propagation

In the context of polynomial approximation it might be of interest, how the uncertainty of a measurement impacts that approximation. Hereby, the task is to compute to which extend variance propagates through a calculation. In other words, to which extend does changes to an measurement vector \mathbf{y} affect the coefficients $\boldsymbol{\alpha}$.

Considering the over-determined polynomial approximation with,

$$\boldsymbol{\alpha} = \mathbf{V}^+ \mathbf{y}. \quad (2.55)$$

The mean of the coefficient vector $\boldsymbol{\alpha}$ is derived by,

$$\begin{aligned} \boldsymbol{\mu}_\alpha &= \frac{1}{n} \sum_{i=1}^n \boldsymbol{\alpha}_i \\ &= \mathbf{V}^+ \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i \\ &= \mathbf{V}^+ \boldsymbol{\mu}_y. \end{aligned} \quad (2.56)$$

Subsequently, the covariance of the coefficients are computed with,

$$\begin{aligned} \boldsymbol{\Lambda}_\alpha &= \frac{1}{n} \sum_{i=1}^n (\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)(\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)^T \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{V}^+ \mathbf{y}_i - \mathbf{V}^+ \boldsymbol{\mu}_y)(\mathbf{V}^+ \mathbf{y}_i - \mathbf{V}^+ \boldsymbol{\mu}_y)^T \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{V}^+ \mathbf{y}_i - \mathbf{V}^+ \boldsymbol{\mu}_y)(\mathbf{y}_i^T (\mathbf{V}^+)^T - \boldsymbol{\mu}_y^T (\mathbf{V}^+)^T) \\ &= \mathbf{V}^+ \left\{ \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}_y)(\mathbf{y}_i^T - \boldsymbol{\mu}_y^T) \right\} (\mathbf{V}^+)^T \end{aligned} \quad (2.57)$$

Whereas the covariance of the vector \mathbf{y} are,

$$\boldsymbol{\Lambda}_y = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}_y)(\mathbf{y}_i^T - \boldsymbol{\mu}_y^T), \quad (2.58)$$

which leads to the simplification,

$$\boldsymbol{\Lambda}_\alpha = \mathbf{V}^+ \boldsymbol{\Lambda}_y (\mathbf{V}^+)^T. \quad (2.59)$$

Now, under the consideration that the error of \mathbf{y} is independent and identically distributed (i.i.d), the covariance $\Lambda_{\mathbf{y}}$ can be formulated as,

$$\Lambda_{\mathbf{y}} = \sigma^2 \mathbf{I} \quad (2.60)$$

then the covariance of the coefficients Λ_{α} yield,

$$\Lambda_{\alpha} = \mathbf{V}^+ \sigma^2 \mathbf{I} (\mathbf{V}^+)^T = \sigma^2 \mathbf{V}^+ (\mathbf{V}^+)^T. \quad (2.61)$$

Chapter 3

Time Series Analysis

The following chapter is dedicated to give an brief overview of *time series* in general, as well as of some chosen *challenges* for dealing with time series, which are especially relevant for *change point detection*. Furthermore, an outline for the different problem settings in change point detection in combination with different types of methods is given.

In general, a *time series* is a set of data points which are sorted chronologically or, to put it in an other way, a *time series* displays quantities that vary over time. Apart from that characteristic there are many other specifications by which time series can differ from each other. Therefore, in the following chapter the most important ones are going to be described in more detail.

Algebraically more precise, a *time series* maps a given time index, $t \in \tau$ to an observed measurement $y_t \in \mathbb{R}$. Hereby, two different cases are possible. On one hand, the time series data can be sampled with *continuous* time with,

$$\tau = \mathbb{R} \tag{3.1}$$

and on the other hand, a *discrete* representation would be possible as well, given,

$$\tau = \mathbb{Z}. \tag{3.2}$$

An example for a continuous time series is for instance the measurement of ambient pressure values at random time intervals or with varying sampling rates. Whereas, the observation of the same ambient pressure at constant time intervals, for example every minute, would result in a discrete time series. Taking samples in constant time steps is generally referred to as *uniform sampling*. Out of practical reasons, most of the processed time series in real world applications are discrete ones.

Considering the example stated above, measuring only the ambient pressure would lead to a so called *univariate* time series. Furthermore, by taking measurements of two or more values, the resulting time series would be a *multivariate* one. Considering the same example again, by collecting ambient temperature values additionally at the same time point, the time series object would than be called *multivariate*. What is more, *multivariate* time series $y_t \in \mathbb{R}^D$ are considered *discrete* if they are synchronized and sampled at uniformed time steps.

3.1 Challenges in Time Series Processing

In addition to the basic characteristics described above *time series* show also different challenges or specifications. Therefore, some of the more important ones, in terms of change point detection, are described in the subsequent paragraphs of this chapter.

Trends

A simple and very common example would be an underlying linear trend. Hereby, it might not be very advisable to use a model which tends to drift to the mean for extrapolation, since the mean is constantly increasing or decreasing over time.

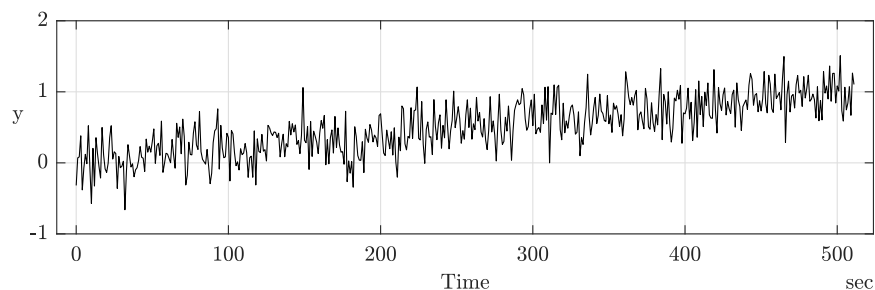


Fig. 3.1: Example of a positive trend in a time series dataset with constant Gaussian noise $\sigma = 0.25$.

Periodicities

Time series datasets showing some sort of *periodicity* or *seasonality* are very common in observations of natural processes. Hereby, some sort of repeating pattern can be observed of certain periods. This could be especially challenging, if a time series dataset is analysed for change points, because the observed data could have a lot of change points algebraically inside those repeating patterns. However, the abrupt change which might be of interest, could be in an underlying function. An example for a periodic time series is given in Figure 3.1, with a pattern of three local mean-shift change points repeating four times. In that case, one might be interested if there was a change besides those inside the patterns, which would not be present in that example.

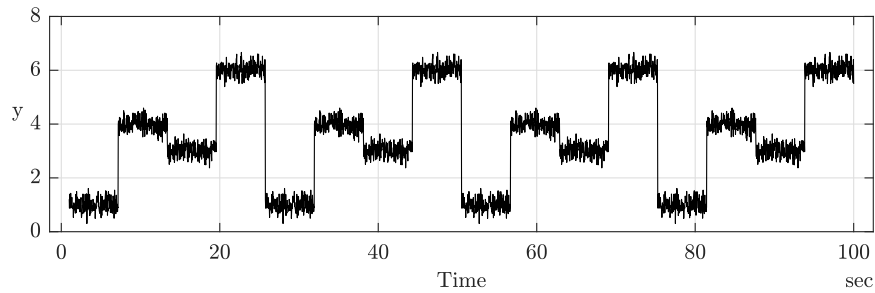


Fig. 3.2: Example of periodicity in a time series dataset with constant Gaussian noise σ .

Outliers

As another prominent representative of the challenges in time series processing, *outliers* are shown as an unexpected behaviour of the dataset. In fact, those could be false peaks due to sensor anomalies or zero values due to failed observations, to name just a few examples.

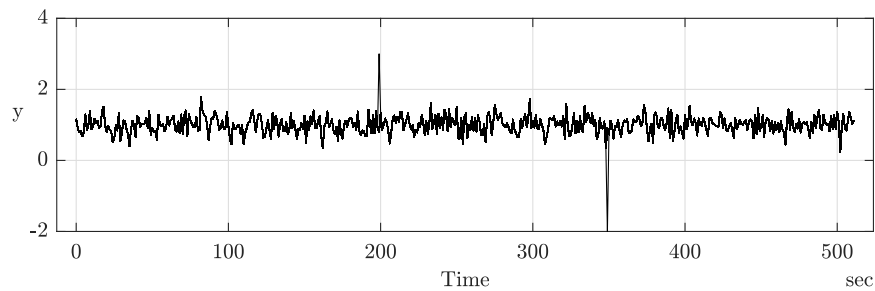


Fig. 3.3: Example of a two outliers at $t_1 = 200s$ and $t_2 = 350s$ in a time series dataset with constant Gaussian noise $\sigma = 0.25$.

Similar to the issue stated with periodicity, outliers can also lead to problems in the course of change point detection, since algebraically they are discontinuities in the dataset.

Change Points

Even though, a *change point* could be confused with an *outlier* in some special cases, the inherent reason for the former is entirely different. Furthermore, a time series dataset can have a change point in various aspects, like in the mean, in the variance or in a derivative to name some of the possibilities. Since change points and in a more special case *discontinuities* in sensor data and their detection are the main topic of this thesis, the upcoming chapters give a more extensive overview of the possible versions of change points.

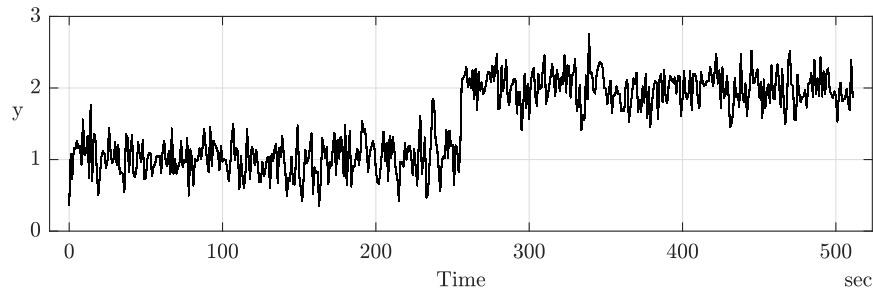


Fig. 3.4: Example of a mean-shift change point at $t = 250s$ from mean $\mu_1 = 1$ to mean $\mu_2 = 2$ with constant Gaussian noise $\sigma = 0.25$.

3.2 Change Point Detection

The field of Change Point Analysis mainly deals with the detection of abrupt changes in data. These changes could of course be of various nature and show different characteristics. First and foremost, in this context one tries to answer the question whether or not there is an unsteadiness in the observed dataset. Consequently, the resulting findings might lead to several sub-problems. Namely, those could be information about the location, size and amount of discontinuities. Furthermore, it might be of interest how certain the detected irregularity is a proper change point and with which probability location and size can be defined.

In 1954 Page [13] published one of the first papers tackling a change point detecting problem. Hereby, he defined the still well known CUSUM (cumulative sum) method combined with a quality criteria h to detect a potential break point. Initially, this procedure was motivated by quality control in manufacturing. Due to the historical relevance and the fact that the technique is still widely used in the field, a more detailed description of the method is part a following subsection in this chapter.

However, before methods for change detection are described in more detail, different variants of problem settings are presented. Hereby, typical change point problems, which are processed in the literature, can be divided in five different groups as follows [14].

3.2.1 Mean-shift

Beginning with the most common representative of change point problems, the *mean-shift*, is present in a dataset, if an abrupt jump in the observation leads to a sustained variation of the underlying mean of the dataset. Hereby, a former mean model can not describe the observation any more, resulting in the mean models for both subsets with the means μ_1 and μ_2 separated by the change point at $\tau = v$.

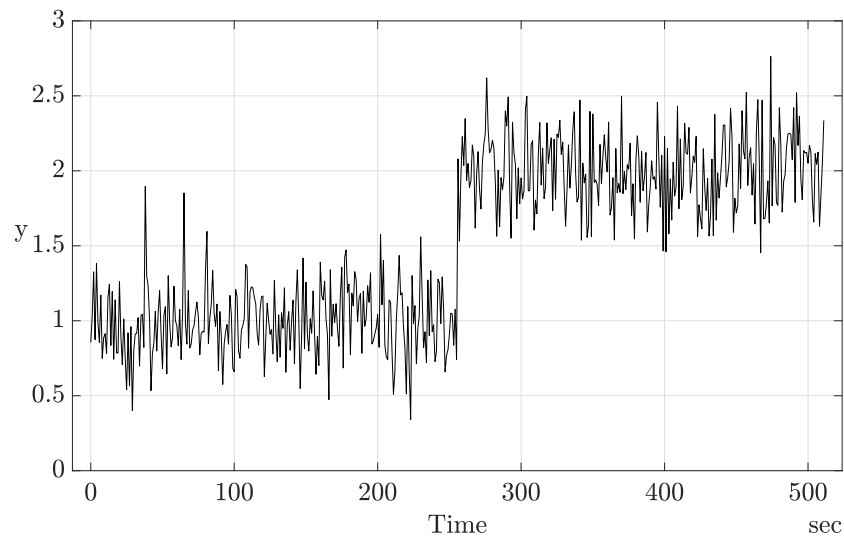


Fig. 3.5: Example for a change point in the mean at a time of $\tau = 256$ s of the dataset. Both parts of the observation are constant within the subset and have Gaussian error distribution with standard deviation of $\sigma = 0.25$.

3.2.2 Variance Change

Another, also quite prominent example might be the *variance change*, which is present if the variance σ_1^2 of a dataset changes suddenly to another level σ_2^2 at the change point location $\tau = \nu$. This type is especially from importance in the context of *frequency* or *vibration analysis*.

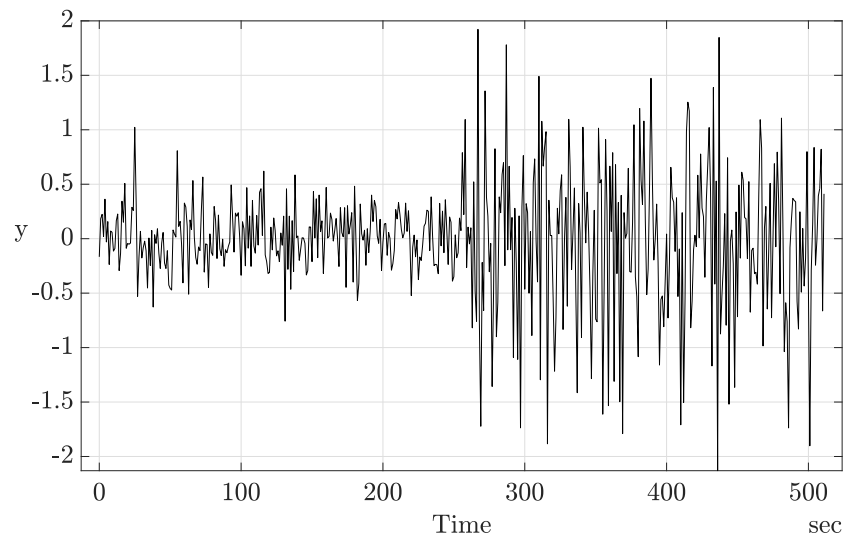


Fig. 3.6: Example for a discontinuity in the variance of a dataset. Two subset are exposed to two different Gaussian error distributions. Hereby, the first one features a standard deviation of $\sigma_1 = 0.25$, whereas the second one has a standard deviation of $\sigma_2 = 0.75$. Resulting in a variance change point at time $\tau = 256 s$.

3.2.3 Regression-slope switch

The next possible type of a change point problem could be the *slope switch*, which is characterised by an abrupt change in the slope - the first order derivative - of a dataset. Algebraically, this change behaviour can also be seen as a *mean-shift* in the first derivative. Of course, with the same principle also higher derivative discontinuities or change point problems are possible.

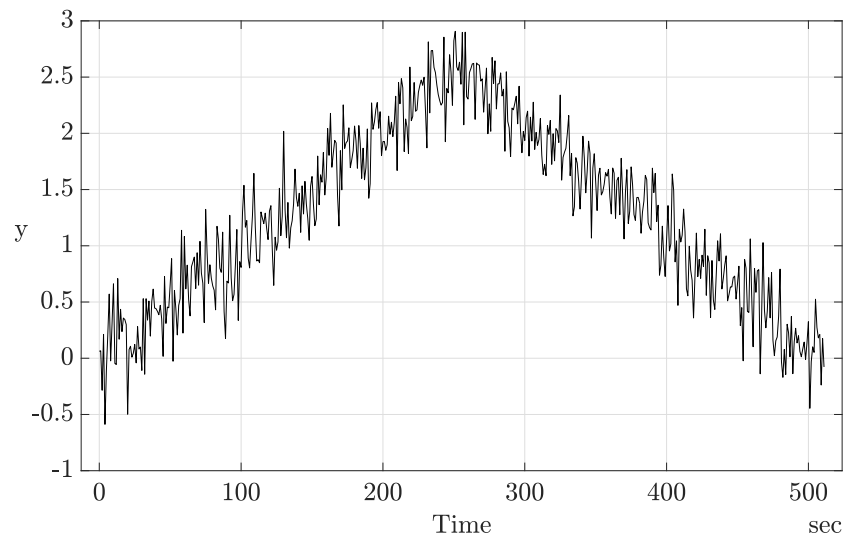


Fig. 3.7: The slope (first derivative) of the time series function is the object of abrupt change happening at $\tau = 256\text{ s}$ with a constant Gaussian error distribution of $\sigma = 0.25$.

3.2.4 Distribution change

The *distribution change* embodies another change point variant, where the type of the underlying distribution changes abruptly at a certain change point location. Such a change could for example be the transition from a *Gaussian* distribution to an *Inverse Gaussian* distribution, as displayed in Figure 3.8.

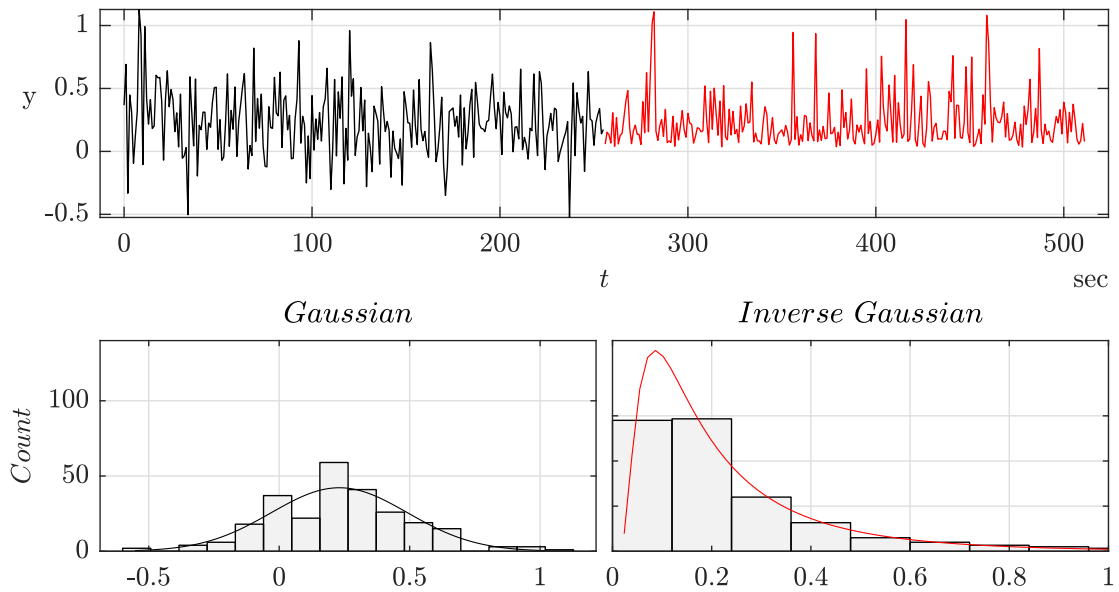


Fig. 3.8: Example for a change in the type of distribution of an observation, from a Gaussian with mean $\mu = 0$ and $\sigma = 0.25$ to an Inverse Gaussian error distribution with original mean $\mu^* = 1$ and shape parameter $\lambda = 1$ at $\tau = 256.s$. (Note: The second signal was shifted to have effectively the same mean as the first one.)

Additionally, some other problem types like the hazard rate change [15] are also subject of the change point detection literature. However, as only those types described above are relevant for this thesis, the others are not discussed in more detail.

3.3 Detection Methods in Change Point Analysis

The main goal of this section is to give an overview of the field of change point detection and describe some of the most popular methods in more detail. Subsequently, this outline of the different algorithms should also assist to properly categorize the recently proposed method by Ninevski and O’Leary [1] in the field and describe the application limits of the algorithm.

As a first basis for segmentation the literature regularly distinguishes between *off-line* and *on-line* detection methods. This differentiation can be seen as fairly general. For instance *off-line* settings benefit of the availability of higher processing time, since the detection process does not need to happen in real-time. Hereby, the data is provided in an entire batch. In contrast a method could be described as *on-line*, if the possible change points or defined batches of measurement values can be processed in real-time. Theoretically, *on-line* methods could be used in *off-line* settings and vice versa. In regards of an algorithm typically used for *on-line* change detection the data which is directly available just needs to be observed one by one to simulate an *on-line* behaviour. In compar-

ison a similar approach is used to perform *off-line* detection mechanism on *on-line* environment. Hereby, data points are collected in smaller batches and then analysed with the *off-line* method. However, one could certainly argue that by using the later approach the real-time behaviour of the technique gets worse with a rising batch size.

Although, so far the method in [1] was only used on *off-line* settings, it could also be used in an *on-line* environment with some adaptations.

One other way to differentiate methods for discontinuity analysis is to distinguish between algorithms for a *single change-point* or for *multiple change-points*. Not to surprisingly the second one is way more difficult and is connected to a much higher calculation effort. This mainly comes from the possibility of having $\binom{n-1}{k-1}$ different ways of n observations to be separated in k segments. Resulting in

$$\sum_{k=1}^{n-1} \binom{n-1}{k-1} = 2^{(n-1)} \quad (3.3)$$

absolute potential outcomes. In analogy to the differentiation between on-line and off-line settings, it might be achievable to use methods generally designed for multiple change point problems in single change-point tasks. Even algorithms dedicated for problems with just a single discontinuity could be used to solve those with several abrupt changes.

In fact, a way this might be obtained is the procedure of *binary segmentation*, like in [16]. The corresponding iterative routine works as follows:

1. Search for the first change point in a dataset
2. Split the used dataset at the location of the explored discontinuity
3. Begin a new detection run in the new segments

Whereas the classification for now was just due to the given characteristics of the data itself, another way to categorize the various techniques is with respect to the used methodology. Hence, the following categorization into five different groups covered in the literature emerges.

1. Parametric Methods
2. Non-Parametric Methods
3. Bayesian Methods
4. Sequential Analysis
5. Miscellaneous Methods

Collected by Lee [14], Table 3.1 shows the distribution of published articles sorted by problem type and used methodology. According to the overview most papers are dealing with parametric methods, at least until 2010.

Subsequently, the following subsections are covering some selected algorithms from those categories.

Type of Problem	Methodology				
	Parametric	Non-parametric	Bayesian	Sequential	Miscellaneous
Mean-shift	40	29	15	33	8
Variance	19	10	7	2	1
Slope-switch	60	9	17	1	2
Hazard-rate	26	6	5	2	7
Distribution	44	32	10	28	4
Total	189	86	54	66	22

Table 3.1: Published articles in the field of change point analysis by Lee, [14] sorted by change point problem type and detection methodology.

3.3.1 Parametric Methods

The group of parametric change point detection methods does make assumptions about certain parameters of a given dataset. Hence, they mainly focus on irregularities in a finite-dimensional parameter vector. This vector could describe the distribution of the given observation belonging to a family of distributions.

Truong et al. [17] are stating three different concepts in the context of parametric change point determination:

1. Maximum likelihood estimation
2. Piecewise linear regression
3. Mahalonobis-type metric

For a more detailed description the reader is referred to [17].

3.3.2 Non-parametric Methods

Occasionally, the assumption of an underlying parametric model could be not suitable for certain observations. As a result the application of non-parametric approaches might be more reliable or robust in specific circumstances.

Hereby, the literature provides a variety of non-parametric methods. For instance, with regard to detection of derivative discontinuities Hall and Titterington (1992) [18] tackled this problem by using an estimation method based on a kernel to approximate peak and edge containing curves, whereas the aim of the method was to preserve those irregularities. Jose and Ismail (1997) [19] on the other hand approached a similar problem by residual analysis. Yet another approach was realized through non-parametric kernel estimation by Müller(1992) [20] as well as Wu and Chu(1993) [21].

Furthermore, a technique on the basis of splines was used by Eubank and Speckman (1994) [22]

among others.

McDonald and Owen (1986) [23], Qui and Yandell (1998) [24] and Spokoiny (1998)[25] are some examples where local polynomial estimation was used for change preserving estimation or change point detection.

Wang (1995) [5] as well as Raimondo (1998) [26] in contrast, rely on wavelet-based approaches. Even though, all of the mentioned solutions are part of the group of non-parametric change point detection methods, all are united by the fact, that their performance is strongly governed by a specific smoothing parameter, which has to be chosen correspondingly.

In general, the literature [17] also distinguishes between three core principles:

1. Non-parametric maximum likelihood estimation
2. Rank-based detection
3. Kernel-based detection

Also for non-parametric methods, a more extensive overview is give in [17].

3.3.3 Bayesian Methods

Another group of popular detection methods are Bayesian algorithms. The majority of them are dealing with off-line problems. To name just a few, early publications in the area of Bayesian change point analysis are *Chernoff and Zacks* [27], *Broemeling* [28] and *Smith* [29]. All of the above have in common, that they are describe a problem in a general form, such that the distribution of the observed dataset prior and after a potential change point is known or at least the distribution of one side is not known. Furthermore, most of the methods in the literature based on Bayesian principles make use of the *product partitioning model* introduced by *Barry and Hartigan* [30]. Hereby, the likelihood of a dataset is segmented into a product of sub-likelihoods.

Because it is a highly used and very effective detection technique [31], the *Bayesian Online Changepoint Detection* method by *Adams and MacKay* [32] is going to be described in more detail in the following part. A similar approach was made by *Fearnhead and Liu* [33].

Note that in contrary to the majority of Bayesian algorithms, the two methods mentioned above are focusing on on-line change detection. However, as already described, these methods can easily be adapted for off-line usage, to some extent.

As the core principle of *online change point detection* the time elapsed since the last change point - referred to as *run length* r_t - is modelled with respect to *a priori* and *posterior* knowledge. Basically the evolution of r_t proceeds as,

$$r_t = \begin{cases} 0 & \text{if a changepoint is likely at } t \\ r_{t-1} + 1 & \text{otherwise} \end{cases} \quad (3.4)$$

Formula 3.4 describes the two ways r_t can evolve from one time step to another. Given the only two possibilities are either to increase by *one*, if the probability of a change point is low enough at the certain data point t , or to set to *zero* directly, should there be a high probability of a change point at t .

Note: As the *Bayesian change point detection* is an probabilistic approach the results are only *estimates* of the changepoint.

Hereby, the *changepoint prior* $P(r_t|r_{t-1})$ models the probability of the transition. Finally, for the prediction of the upcoming point $t + 1$ one deduce the *run-length posterior distribution* $P(r_t|x_{1..t})$, as well as the *predictive distribution* $P(x_{t+1}|x_{1..t})$.

The following overview should give an intuitive understanding of the method. Hence, the procedure is defined by the key measures described subsequently.

1. Value of the *probability density function* at $R_{[r,t]}$ using the distribution parameters from $R_{[r-1,t-1]}$
 Note: If the *pdf* is high, it is likely that the point is on the path.
2. New distribution parameter values
 Calculate μ' and σ' for $[r,t]$ in case of a Gaussian distribution.
3. Hazard value H , depended on the current run length
4. Growth probability
 Probability for the run length to increase from $r - 1$ to r , which is the product of the following values:
 - a. Growth probability at $R_{[r-1,t-1]}$
 - b. Probability that the Hazard did not occur: $1 - H$
 - c. *probability density function* for the new data point
5. Change Probability Probability for the run length to decrease to 0. Just like the *growth probability* also the *change probability* is a product of three values. Whereas here the actual hazard rate H is part of the equation.
 - a. Growth probability at $R_{[r-1,t-1]}$
 - b. Probability that the Hazard did occur: H
 - c. *probability density function* for the new data point

3.3.4 Sequential Analysis

In general, the statistical discipline of *sequential analysis* [34] is dealing with variable sample sizes. Hereby, the data is processed just as it is provided and the testing is stopped based on an predefined stopping criteria. Hence, change point detection methods belonging to this field are typically capable of dealing with on-line detection settings.

Given that the majority [14] of all sequential analysis methods are related to the CUSUM method established by Page (1954) [13], this chapter is dedicated to give a more in-depth explanation of the algorithm. Apart from being the main part of the group, the principle is also one of the most popular techniques in change point analysis. Additionally, it is still widely used.

Even though, a vast amount of extensions and adaptations was performed to the general CUSUM method, the principle idea stays the same. In general, the cumulative sum from the observed data - maybe exposed to a weighting factor or some other transformation - is calculated. Subsequently, a potential change point can be found in the region of the point where the calculated sum has its absolute maximum. In order to guarantee significant and reliable results some form of threshold needs to be defined and compared to the found maximum of cumulative sum. Should the detected extreme point exceed the threshold, then a change point might be in this region.

Page (1954) [13] was - as already mentioned - one of the first authors who tackled the problem of change point detection. In fact, he proposed several different variations of the CUSUM method:

1. direct form
2. recursive form
3. one-sided form
4. two-sided form

The task hereby can be defined as follows.

Let $\mathcal{X}_{1..n} = \{x_1 \dots x_n\}$ be a set of random discrete observations having samples that are independent and identically distributed (i.i.d.). Each and every sample can be described by a probability density function (PDF), $p(x_{1..n}, \theta)$ which is a function of the deterministic variable θ . Such an observation may contain sudden changes at a specific change point ν . Resulting in an instantaneous variation of θ at the point ν . Consequently, this very observation can not be described by the same PDF any more. Subsequently, the data will be modelled with $\theta = \theta_1$ before ν and with $\theta = \theta_2$ afterwards. Assuming that there is only one potential change point in the dataset the following assumptions can be made:

1. \mathcal{H}_0 - Null hypothesis (no change point)

$$p_{\mathcal{X}|\mathcal{H}_0} = \prod_{n=0}^k p(x(n), \theta_1) \quad (3.5)$$

2. \mathcal{H}_1 - Alternative hypothesis (one change point)

$$p_{\mathcal{X}|\mathcal{H}_1} = \prod_{n=0}^{v-1} p(x(n), \theta_1) \prod_{n=v}^k p(x(n), \theta_2) \quad (3.6)$$

Hereby, it is assumed that both the probability density function of every sample, as well as the parameters θ_1 and θ_2 are known. Hence, the remaining question one may wish to answer is, if there is a change point in the dataset. Depending on the result one subsequent task might be to find the point of change v . A popular technique for solving this problem is the popular likelihood ratio test [35], which is part of *detection theory*.

Hereby the *log-likelihood ratio* $\mathcal{L}_{\mathcal{X}}$ denotes as:

$$\mathcal{L}_{\mathcal{X}} = \ln \left(\frac{p_{\mathcal{X}|\mathcal{H}_1}}{p_{\mathcal{X}|\mathcal{H}_0}} \right) \quad (3.7)$$

For, $\mathcal{L}_{\mathcal{X}} \leq h$, where h is a predefined threshold, the null hypothesis \mathcal{H}_0 is accepted meaning there is no abrupt change in the observed data point. On the other hand, should, $\mathcal{L}_{\mathcal{X}} > h$, the null gets rejected resulting in a detected change point at the point v . In other words, for an accepted alternative hypothesis, $p_{\mathcal{X}|\mathcal{H}_1}$ describes the PDF for the given data significantly better than $p_{\mathcal{X}|\mathcal{H}_0}$.

Under consideration of Formula 3.5 and Formula 3.6, Formula 3.7 can be written as follows:

$$\begin{aligned} \mathcal{L}_{\mathcal{X}} &= \ln \left(\frac{\prod_{n=0}^{v-1} p(x(n), \theta_1) \prod_{n=v}^k p(x(n), \theta_2)}{\prod_{n=0}^k p(x(n), \theta_1)} \right) \\ &= \ln \left(\frac{\prod_{n=0}^{v-1} p(x(n), \theta_1) \prod_{n=v}^k p(x(n), \theta_2)}{\prod_{n=0}^{v-1} p(x(n), \theta_1) \prod_{n=v}^k p(x(n), \theta_1)} \right) \\ &= \ln \left(\frac{\prod_{n=v}^k p(x(n), \theta_2)}{\prod_{n=v}^k p(x(n), \theta_1)} \right) \quad (3.8) \\ &= \sum_{n=v}^k \ln \left(\frac{p(x(n), \theta_2)}{p(x(n), \theta_1)} \right) \end{aligned}$$

Even though the *log-likelihood ratio* $\mathcal{L}_{\mathcal{X}}$ is simplified now, the potential change point v is still unknown. As a consequence, it is still not possible to make a statement about the acceptance of the

null hypothesis \mathcal{H}_0 .

Fortunately, *detection theory* [35] delivers yet another solution. In fact, the *log-likelihood ratio* \mathcal{L}_x is going to be replaced by a *generalized likelihood ratio test* \mathcal{G}_x , whereas the maximum of the *log-likelihood ratio* \mathcal{L}_x is of particular interest.

Hence \mathcal{G}_x denotes as:

$$\begin{aligned}\mathcal{G}_x &= \max_{1 \leq v \leq k} \mathcal{L}_x \\ &= \max_{1 \leq v \leq k} \sum_{n=v}^k \ln \left(\frac{p(x(n), \theta_2)}{p(x(n), \theta_1)} \right)\end{aligned}\quad (3.9)$$

Where \mathcal{G}_x is frequently referred to as *decision function*, since the null hypothesis is tested as

$$\mathcal{H}_0 : \mathcal{G}_x \leq h$$

against the alternative

$$\mathcal{H}_1 : \mathcal{G}_x > h.$$

Just like before, h is again a predefined threshold, mainly for the purpose of detecting *significant* points of change.

Given, that \mathcal{H}_0 has been rejected this finding subsequently yields another problem setting. Namely, as a discontinuity is detect, the following task is to make an efficient estimate about the location v of the point in question. One solution to this problem is the *maximum likelihood estimate* [36], which is the estimate \hat{v} of the change point v maximizing the *probability density function* $p_{x|\mathcal{H}_1}$.

$$\begin{aligned}\hat{v} &= \operatorname{argmax}_{1 \leq v \leq k} \mathcal{L}_x \\ &= \operatorname{argmax}_{1 \leq v \leq k} \sum_{n=v}^k \ln \left(\frac{p(x(n), \theta_2)}{p(x(n), \theta_1)} \right)\end{aligned}\quad (3.10)$$

In order to get a more or less straight forward form of the algorithm, the following transformations can be performed. (*direct form*) The *log-likelihood ratio* at the point n is

$$s[n] = \mathcal{L}_{\mathcal{X}}[n, n] = \ln \left(\frac{p(x(n), \theta_2)}{p(x(n), \theta_1)} \right) \quad (3.11)$$

and the cumulative sum of the *log-likelihood ratio* from 0 to k then yields

$$S[k] = \sum_{n=0}^k s[n]. \quad (3.12)$$

As a result, the *log-likelihood ratio* for a potential change point $\mathcal{L}_{\mathcal{X}}[k, v]$ can easily be derived

$$\mathcal{L}_{\mathcal{X}}[k, v] = S[k] - S[v - 1]. \quad (3.13)$$

Furthermore, the *generalized likelihood ratio* as well as the *maximum likelihood estimate* resulting in

$$\mathcal{G}_{\mathcal{X}}[k] = S[k] - \min_{1 \leq v \leq k} S[v - 1]. \quad (3.14)$$

$$\hat{v} = \underset{1 \leq v \leq k}{\operatorname{argmin}} S[v - 1] \quad (3.15)$$

Note: This is where the name of the *CUSUM* methods comes from.

Another version, which is computationally more efficient and additionally better suited for on-line or real-time change point problems is the *recursive form* of the *CUSUM* method.

Hereby, the cumulative sum is calculated as

$$S[k] = S[k-1] + s[k] \quad (3.16)$$

then

$$\mathcal{G}_{\mathcal{X}}[k] = \{\mathcal{G}_{\mathcal{X}}[k-1] + s[k]\}^+ \quad (3.17)$$

with

$$\mathcal{G}_{\mathcal{X}}[-1] = 0 \quad (3.18)$$

In equation 3.17 $\{.\}^+$ considers the fact that the threshold h only has positive values. According to equation 3.18 \mathcal{G}_x is initialized with zero and the following values are stored for each subsequent step. This brings the benefit that, unlike in the *direct form* \mathcal{G}_x has not to be calculated for each sample.

Apart from distinguishing between the *direct form* and the *recursive form* of the *CUSUM* method, a further differentiation can be made with respect to the application limit. By doing so, the techniques are separated in *one-sided* and *two-sided* algorithms.

Considering the *CUSUM* method described so far, one would speak about an *one-sided* method, since an abrupt change can only be detected in one direction. For example, just positive change points may trigger the threshold h . Of course, this could be quite unfavourable in many real-world applications, as most of the time one might be looking for both positive and negative discontinuities. Fortunately, this problem has been addressed by many authors. Unsurprisingly, already Page [13] proposed a solution for this. In general, the approach consists basically of two *one-sided* methods combined. One for positive and one for negative change points.

Of course, there are many other variations or adaptations of the *CUSUM* method in the literature. Among many others there are for instance the enhancement for multivariate data [37], the *fast initial response CUSUM* [38] and the *combined Shewhart-CUSUM* methods [39].

3.4 Condition Monitoring

Now, a quick introduction is given to the field of *condition monitoring*, as it is one important application example of change point detection. Here some important terms are stated and described briefly.

In order to ensure and maintain a proper operation of a machine in an industrial environment, it is necessary to conduct various measures in the context of process observation. Namely, those could be oil pressure, oil temperature, engine speed and many more. Besides, the monitoring of the aforementioned operational parameters, it is at least equally important to regularly maintain, inspect and repair the machine.

As a modern approach for machine maintenance, the field of *Condition Monitoring* (also *Machinery Monitoring*) is dedicated to the observation of certain condition variables of a machine. Hereby, physical factors are measured and analysed to make inferences about the present state of an observed object. Consequently the main goals can be summarized as follows.

1. Identification of early potential damages
2. Machine safety
3. Breakdown prevention
4. Machine Efficiency

Consequently there are various concepts for observation systems, dependent on the desired purpose.

Observation

The task of a simple *observation* is to detect the presence of an error. Furthermore, in the case of an automated system a detected failure can lead to further measures.

Early Error Recognition

Early error recognition systems detects possible future failures and ensures the operation of a machine over a certain time frame.

Error Diagnosis

After the collection of relevant sensor data, the subsequent task is to derive an *error diagnosis*. This can be achieved by various signal processing methods for example.

Trend Analysis

Methods for *trend analysis* are often embedded in systems for *early error recognition*. At first, the operation of the monitored machine is still possible, even if a possible future error has been detected.

Forecast

After the *trend analysis* the *forecast* tries to estimate the residual run time of the machine until a critical state is reached.

Another possible separation for observation systems could be due to the frequency of monitoring. On the one hand the *permanent observation* yields a high degree of security with the downside of considerably more effort. Whereas, the *intermediate observation* is way more flexible and therefore less reliable.

Chapter 4

Derivative Discontinuity Detection via Constrained Local Polynomial Approximation

The following chapter is dedicated to give a detailed explanation of the method for derivative discontinuity detection in [1].

4.1 Algebraic Discontinuities

For univariate functions, three different types of discontinuities can be distinguished. Beginning with the simplest form, the so called *removable discontinuity* is part of a *real-valued univariate function* $f = f(x)$ at the sample point x_0 if,

$$f(x_0) < \infty \quad (4.1)$$

and also

$$\lim_{x \rightarrow x_0} f(x) = L < \infty, \quad (4.2)$$

but

$$f(x_0) \neq L. \quad (4.3)$$

Removable discontinuities have their name from the fact, that the point x_0 causing the discontinuity, can be removed, yielding an almost identical function $F = F(x)$. A simple example of a removable discontinuity at the point x_0 is shown in Figure 4.1.

Again, as the name suggests the given discontinuity can be removed, resulting in an almost identical continuous function,

$$F(x) = \begin{cases} 3x + 1 & \text{for } x \neq x_0 \\ 1.6 & \text{for } x = x_0. \end{cases} \quad (4.4)$$

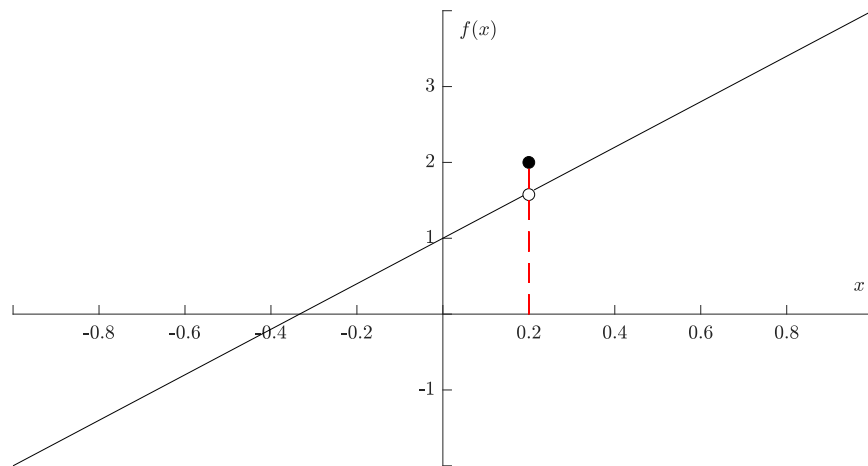


Fig. 4.1: Example of a *removable discontinuity* in a univariate real-valued function $f = f(x)$ at the point $x = x_0$.

The so called *jump discontinuity* is the second variant of possible discontinuities in univariate functions $f = f(x)$.

Algebraically, this form of singularity is defined with,

$$\lim_{x \rightarrow x_0^-} f(x) = L_1 < \infty \quad (4.5)$$

and

$$\lim_{x \rightarrow x_0^+} f(x) = L_2 < \infty \quad (4.6)$$

whereas $L_1 \neq L_2$.

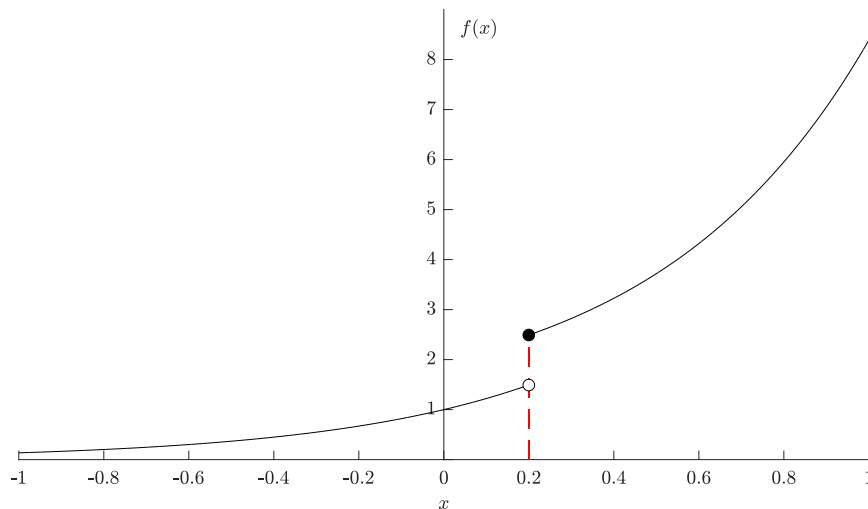


Fig. 4.2: Example of a *jump discontinuity* in a univariate real-valued function $f = f(x)$ at the point $x = x_0$.

An example for a jump discontinuity at $x_0 = 0.2$ is shown in Figure 4.2 for the function

$$f(x) = \begin{cases} e^x & \text{for } x < x_0 \\ e^x + 1 & \text{for } x \geq x_0. \end{cases} \quad (4.7)$$

Finally, the third version of a discontinuity in a univariate function $f = f(x)$ is known as an *infinite discontinuity*. Hereby, a typical example would be the function $f(x) = \tan(x)$ with infinite discontinuities at $\frac{\pi}{2} x = n\frac{\pi}{2}$ with $n \in \mathbb{R}$ as shown in Figure 4.3.

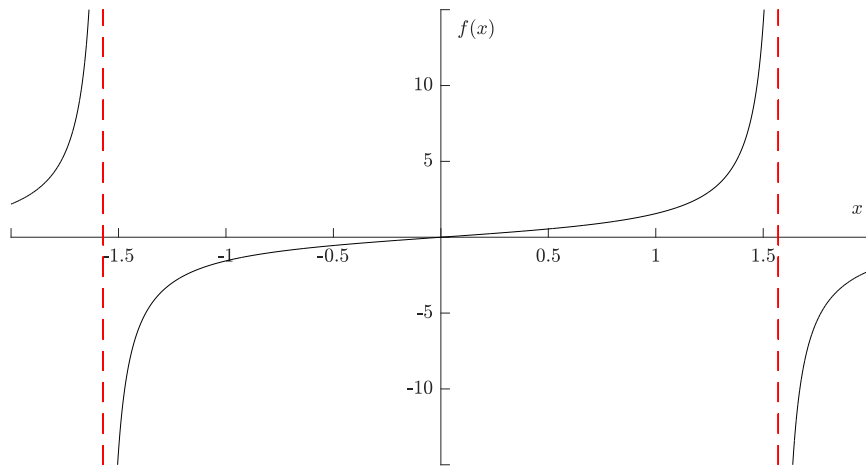


Fig. 4.3: Example of an *infinite discontinuity* in a univariate real-valued function $f = f(x)$ at the point $x = x_0$

Even though there are different forms of discontinuities as described above, the focus of this thesis is only on *jump discontinuities* in functions and their derivatives. These are also closely related to the also already described *mean-shift* and *slope-shift change point problems*.

4.2 Constrained Coupled Polynomial Approximation

The upcoming subchapter should give an overview of a derivative discontinuity detection method [1] which applies two constrained coupled polynomial approximations sequentially to set of observations with suspected discontinuities. Theoretically, the method can be applied in off-line and on-line setting. In addition, the detection of multiple change points is possible as well.

4.2.1 General Principle

Now, the general idea is briefly introduced and followed by a more detailed description of the necessary computational steps. Hereby, one polynomial $f_{(x,\alpha)}$ is approximated to the left and another one to the right $g_{(x,\beta)}$ of an interstitial point ξ_i , under the condition that at the point ξ_i the coefficients up the $(n-1)^{th}$ order are constrained with,

$$\alpha_i = \beta_i, \text{ for every } i \in [0 \dots n-1]. \quad (4.8)$$

While constraining the first $n-1$ coefficients the method yields an estimate for the n^{th} order polynomial coefficients α and β . These are then closely related to coefficients $t_f^{(n)}$ and $t_g^{(n)}$ of a Taylor expansion at the interstitial point ξ_i in both directions, which are by itself proportional to the n^{th} derivative at the evaluation point ξ_i . Subsequently, inferences about possible n^{th} order discontinuities can be drawn with respect to the magnitude of the calculated Taylor differences Δt_{fg}^n . In other words, should the n^{th} order Taylor coefficients of the left approximation $t_f^{(n)}$ and of the right approximation $t_g^{(n)}$ differ significantly, chances are high that there is a discontinuity in the n^{th} derivative of the observed data at the interstitial point ξ_i .

There are two reasons why interstitial points are used for the method. On the one hand, the usage of them makes sure that each observed point x_i is only used in one polynomial approximation. Moreover, especially for sensor data it is most likely that the suspected discontinuity might happen between two sample points. As a result, a method for interpolation between the interstitial point is necessary. Hereby, this is accomplished by polynomial approximation, as the Weierstrass approximation theorem [10] recommends. (see Chapter 2)

A generic illustration of the principle is displayed in the following Figure 4.4.

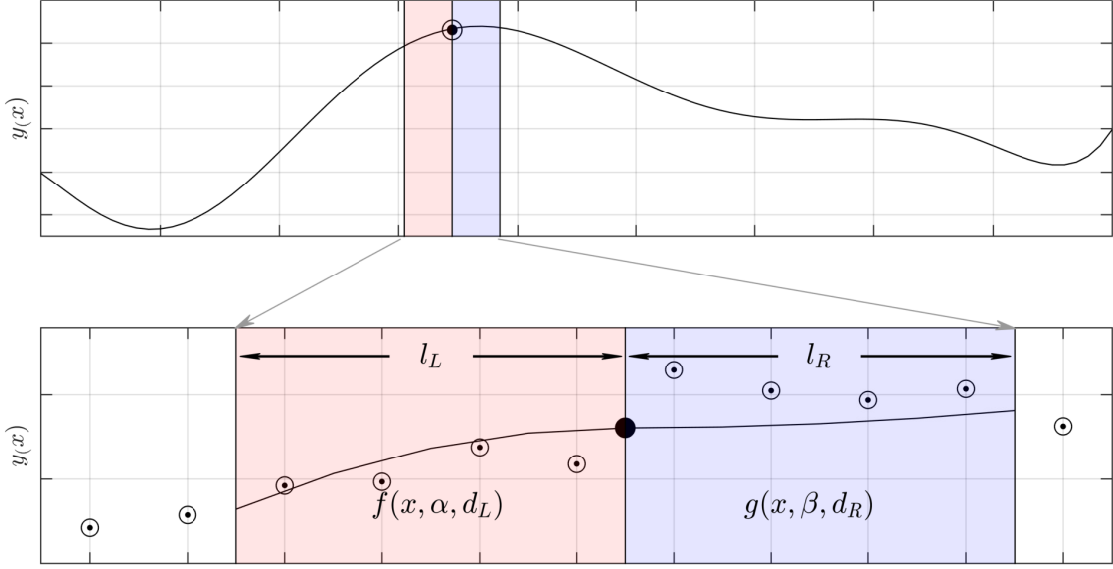


Fig. 4.4: Schematic visualization of the approximation polynomials $f(x, \alpha, d_L)$ and $g(x, \beta, d_R)$ to the left and to the right of the interstitial point ξ_i over the support lengths l_L and l_R respectively.

As already described, the basic principle of the method is to evaluate a polynomial to the left and to the right of an interstitial point ξ_i . Subsequently, these approximations yield two sets of coefficients α and β , whereas the coefficients up to the order $(n - 1)$ are constrained at the interstitial point ξ_i to guarantee C^{n-1} continuity. As a result the degrees of freedom of the approximation are reduced and an improved variance of the solution can be maintained.

Consequently, the polynomial approximation at the interstitial point ξ_i can be evaluated for \mathbf{y}_L to the left and for \mathbf{y}_R to the right with the corresponding *Vandermonde* matrices \mathbf{V}_L for the left approximation and \mathbf{V}_R for the right approximation. The two vectors α and β are dedicated to store the respective polynomial coefficients.

$$\mathbf{y}_L = \mathbf{V}_L \alpha \quad (4.9)$$

$$\mathbf{y}_R = \mathbf{V}_R \beta \quad (4.10)$$

Furthermore, the approximation Formulas 4.9 and 4.10 can be written in block matrix form as,

$$\begin{bmatrix} \mathbf{V}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \quad (4.11)$$

As stated in Formula 4.8, a C^{n-1} continuity has constrained coefficients, except for those of the $n - th$ order.

$$\alpha_n \neq \beta_n \quad (4.12)$$

This characteristic can be written in a structured manner in matrix form as,

$$\left[0 \quad \mathbf{I}_{n-1} \mid 0 \quad -\mathbf{I}_{n-1} \right] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0}. \quad (4.13)$$

In the light of the above the following definitions can be made for the *Vandermonde* matrix storing the left and the right *Vandermonde* matrices in block matrix form

$$\mathbf{V} \triangleq \begin{bmatrix} \mathbf{V}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix}, \quad (4.14)$$

the coefficient vector

$$\boldsymbol{\gamma} \triangleq \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \quad (4.15)$$

an observation vector

$$\mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \quad (4.16)$$

and for the *constraint matrix*

$$\mathbf{C} \triangleq \left[0 \quad \mathbf{I}_{n-1} \mid 0 \quad -\mathbf{I}_{n-1} \right]. \quad (4.17)$$

Subsequently, the *least squares minimization* with *homogeneous linear constraints* can be written as,

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{V}\boldsymbol{\gamma}\|_2^2 \quad (4.18)$$

with

$$\mathbf{C}\boldsymbol{\gamma} = \mathbf{0}. \quad (4.19)$$

In order to satisfy Equation 4.19 $\boldsymbol{\gamma}$ has to lie in the null-space of the constraint matrix \mathbf{C} . In other words, the coefficient vector $\boldsymbol{\gamma}$ gets mapped by the constraint matrix \mathbf{C} to the *null vector* $\mathbf{0}$. Consequently, the orthonormal vector basis set

$$\mathbf{N} = \text{null}\{\mathbf{C}\} \quad (4.20)$$

can be derived. Subsequently, the coefficient vector $\boldsymbol{\gamma}$ can be stated as,

$$\boldsymbol{\gamma} = \mathbf{N}\boldsymbol{\delta} \quad (4.21)$$

Now, by substituting Equation 4.21 into Equation 4.18 the least squares minimization problem results in,

$$\min_{\delta} \|\mathbf{y} - \mathbf{V}\mathbf{N}\delta\|_2^2. \quad (4.22)$$

Consequently, the δ which minimizes equation 4.22 is,

$$\delta = (\mathbf{V}\mathbf{N})^+ \mathbf{y} \quad (4.23)$$

Hence, the desired coefficients are derived by substituting equation 4.23 into equation 4.21,

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{N}(\mathbf{V}\mathbf{N})^+ \mathbf{y} \quad (4.24)$$

Finally, in order to compute the *Taylor coefficients* $\Delta \mathbf{t}_{fg}^{(n)}$ the difference vector \mathbf{d} is defined as,

$$\mathbf{d} = \begin{bmatrix} 1 \\ \mathbf{0}_{d_{L-1}} \\ -1 \\ \mathbf{0}_{d_{R-1}} \end{bmatrix} \quad (4.25)$$

Consisting only of two *zero vectors* $\mathbf{0}_{d_{L-1}}$ and $\mathbf{0}_{d_{R-1}}$ as well as one entry of 1 and another one of -1 the sole purpose of this vector is to pick out the coefficients we are interested in from the vector $\boldsymbol{\gamma}$. Since, according to equation 4.8 all coefficients up to the order $n - 1$ are constrained, there respected differences would be zero anyway. Furthermore, by looking for discontinuities of order n , all other orders are not of interest in this case. Finally, the respective *Taylor differences* $\Delta \mathbf{t}_{fg}^{(n)}$ can be found by,

$$\Delta \mathbf{t}_{fg}^{(n)} = \begin{bmatrix} 1 \\ \mathbf{0}_{d_{L-1}} \\ -1 \\ \mathbf{0}_{d_{R-1}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{d}\boldsymbol{\gamma} = \mathbf{d}\mathbf{N}(\mathbf{V}\mathbf{N})^+ \mathbf{y}. \quad (4.26)$$

4.2.2 Covariance Propagation

In order to estimate how the covariance of the observed data propagates through the method to the derived *Taylor differences* $\Delta \mathbf{t}_{fg}^{(n)}$ the following calculations can be made.

Given,

$$\mathbf{K} = \mathbf{N}(\mathbf{V}\mathbf{N})^+ \quad (4.27)$$

the coefficient vector $\boldsymbol{\gamma}$ is denoted by,

$$\gamma = \mathbf{K} \mathbf{y}. \quad (4.28)$$

Consequently, with the covariance of the data Λ_y , the covariance of the coefficients Λ_γ results in,

$$\Lambda_\gamma = \mathbf{K} \Lambda_y \mathbf{K}^T. \quad (4.29)$$

Hence, the final covariance of the *Taylor coefficients* Λ_Δ equates to,

$$\Lambda_\Delta = \mathbf{d} \Lambda_\gamma \mathbf{d}^T. \quad (4.30)$$

$$\begin{aligned} \Lambda_\Delta &= \mathbf{d} \mathbf{K} \Lambda_y \mathbf{K} \mathbf{K}^T \mathbf{d}^T \\ &= \mathbf{d} \mathbf{N} (\mathbf{V} \mathbf{N})^+ \Lambda_y (\mathbf{N} (\mathbf{V} \mathbf{N})^+)^T \mathbf{d}^T \end{aligned} \quad (4.31)$$

4.2.3 Error Analysis

Finally, to get additional measures for the determination of a possible derivative discontinuity the following figures are derived as well.

1. Norm of the approximation residual
2. Combined approximation and extrapolation error
3. Extrapolation error

4.2.3.1 Approximation Error

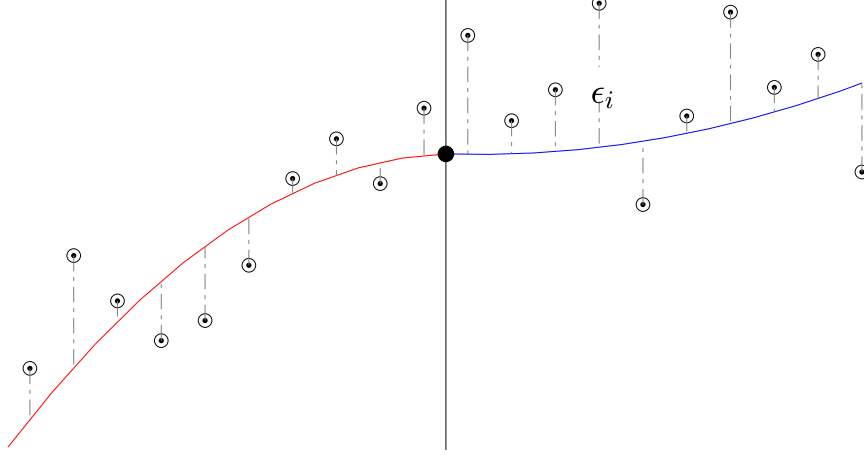


Fig. 4.5: Schematic visualization of the approximation error with two second order polynomials approximations at an interstitial point ξ including the difference ε_i between the approximation and the actual observation.

By comparing the observations with the evaluated approximation over an interval, the *approximation error* can be seen as a measure for the accuracy or the quality of the approximation. Numerically, this measure can be computed by comparing these actual values with its approximation for each sample, as Figure 4.5 suggests. Whereas, ε_i is the just mentioned difference.

Consequently, the residual vector \mathbf{r} is derived as the difference between the observed data vector \mathbf{y} and its estimated counterpart $\mathbf{V}\boldsymbol{\gamma}$ as follows,

$$\mathbf{r} = \mathbf{y} - \mathbf{V}\boldsymbol{\gamma} = \begin{bmatrix} \mathbf{y}_L - \mathbf{V}_L\boldsymbol{\alpha} \\ \mathbf{y}_R - \mathbf{V}_R\boldsymbol{\beta} \end{bmatrix} \quad (4.32)$$

Thus, the *quadratic norm* of the residual vector \mathbf{r} delivers the *approximation error* E_a with,

$$\begin{aligned} E_a &= \|\mathbf{r}\|_2^2 \\ &= \|\mathbf{y}_L - \mathbf{V}_L\boldsymbol{\alpha}\|_2^2 + \|\mathbf{y}_R - \mathbf{V}_R\boldsymbol{\beta}\|_2^2 \\ &= (\mathbf{y}_L - \mathbf{V}_L\boldsymbol{\alpha})^T (\mathbf{y}_L - \mathbf{V}_L\boldsymbol{\alpha}) + (\mathbf{y}_R - \mathbf{V}_R\boldsymbol{\beta})^T (\mathbf{y}_R - \mathbf{V}_R\boldsymbol{\beta}) \\ &= \mathbf{y}^T \mathbf{y} - 2\boldsymbol{\alpha}^T \mathbf{V}_L^T \mathbf{y}_L + \boldsymbol{\alpha}^T \mathbf{V}_L^T \mathbf{V}_L \boldsymbol{\alpha} - 2\boldsymbol{\beta}^T \mathbf{V}_R^T \mathbf{y}_R + \boldsymbol{\beta}^T \mathbf{V}_R^T \mathbf{V}_R \boldsymbol{\beta}. \end{aligned} \quad (4.33)$$

4.2.3.2 Extrapolation Error

Considering a dataset is not exposed to a discontinuity, the extrapolation over a defined interval should give a relative good estimate for the upcoming observations. However, if a discontinuity is present, the extrapolation and the corresponding data point differ significantly. Hence, the *ex-*

trapolation error can be used as another measure for the detection of a discontinuity. In order to compute the extrapolation error the differences ϵ_i between the extrapolation and the respective sample point is calculated, as shown in Figure 4.6.

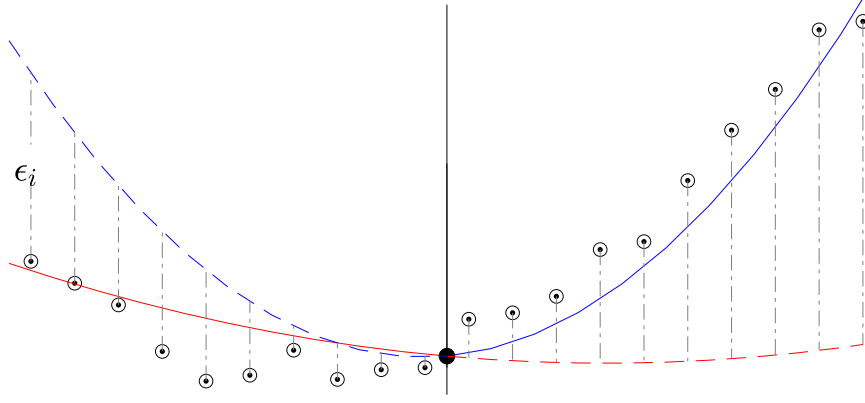


Fig. 4.6: Schematic visualization of the extrapolation error with two second order polynomials approximations at an interstitial point ξ including the difference ϵ_i between the extrapolation and the actual observation.

In general, by performing the aforementioned computation over an interval with the length l_L or l_R respectively, the *extrapolation residual* vectors \mathbf{r}_{ef} and \mathbf{r}_{eg} denote as,

$$\mathbf{r}_{ef} = \mathbf{y}_L - g(\mathbf{x}_L, \boldsymbol{\beta}) = \mathbf{y}_L - \mathbf{V}_L \boldsymbol{\beta} \quad (4.34)$$

and

$$\mathbf{r}_{eg} = \mathbf{y}_R - f(\mathbf{x}_R, \boldsymbol{\alpha}) = \mathbf{y}_R - \mathbf{V}_R \boldsymbol{\alpha}. \quad (4.35)$$

Whereas, the *extrapolation residual* vectors \mathbf{r}_{ef} and \mathbf{r}_{eg} are storing the respective residual errors ϵ_i and have the structure,

$$\mathbf{r}_{ef} = \begin{bmatrix} \epsilon_{ef,1} \\ \epsilon_{ef,2} \\ \vdots \\ \epsilon_{ef,l} \end{bmatrix} \quad \mathbf{r}_{eg} = \begin{bmatrix} \epsilon_{eg,1} \\ \epsilon_{eg,2} \\ \vdots \\ \epsilon_{eg,l} \end{bmatrix}$$

Finally, the sum of the *squared* residual vectors delivers the actual *extrapolation error* E_a . In matrix notation this can be written as,

$$\begin{aligned}
E_e &= \mathbf{r}_{ef}^T \mathbf{r}_{ef} + \mathbf{r}_{eg}^T \mathbf{r}_{eg} \\
&= (\mathbf{y}_L - \mathbf{V}_L \boldsymbol{\beta})^T (\mathbf{y}_L - \mathbf{V}_L \boldsymbol{\beta}) + (\mathbf{y}_R - \mathbf{V}_R \boldsymbol{\alpha})^T (\mathbf{y}_R - \mathbf{V}_R \boldsymbol{\alpha}) \\
&= \mathbf{y}_L^T \mathbf{y}_L - 2\boldsymbol{\beta}^T \mathbf{V}_L^T \mathbf{y}_L + \boldsymbol{\beta} + \mathbf{y}_R^T \mathbf{y}_R - 2\boldsymbol{\alpha}^T \mathbf{V}_R^T \mathbf{y}_R + \boldsymbol{\alpha}^T \mathbf{V}_R^T \mathbf{V}_R \boldsymbol{\alpha}
\end{aligned} \tag{4.36}$$

4.2.3.3 Combined Error

As a third and final measure for a potential derivative discontinuity, the *combined error* is introduced. Hereby, the residual over the entire approximation range is calculated twice. Numerically, this process can be split into two steps. Namely, for each side - left and right to the interstitial point ξ_i - the combined residual error e_f (or e_g) is calculated as the difference between the actual observation and the combined approximation including the interpolation. Considering the left hand side, this would mean the entire observation \mathbf{y} consisting of \mathbf{y}_L and \mathbf{y}_R are compared to the combined approximated values with $f(x_L, \alpha)$ (= approximation) and $f(x_R, \alpha)$ (= extrapolation).

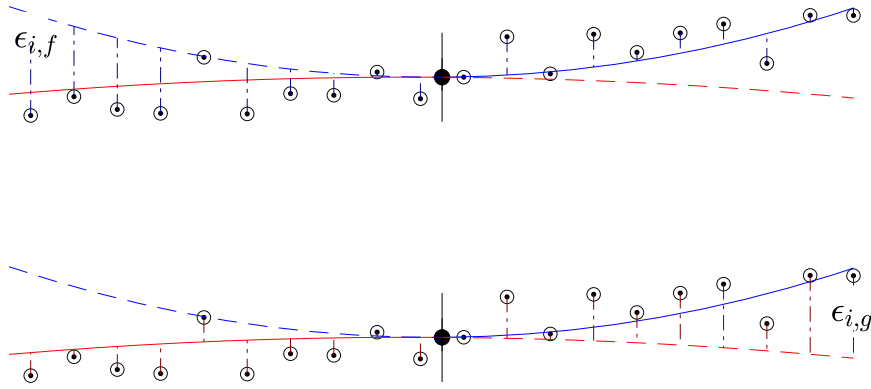


Fig. 4.7: Schematic visualization of the combined error with two second order polynomials approximations at an interstitial point ξ including the difference $\epsilon_{i,f}$ and $\epsilon_{i,g}$ between the combination of approximation and extrapolation and the actual observation. Here the upper subplot is showing several residuals $\epsilon_{i,f}$ for the combined error of the LHS. On the other hand, the lower one is displaying the residuals $\epsilon_{i,g}$ concerning the RHS.

Analytical Combined Error

In the following, the approach to calculate the combined error is shown in an analytical form.

$$\epsilon_x = \int_{x_{min}}^{x_{max}} \{f(x, \boldsymbol{\alpha}) - g(x, \boldsymbol{\beta})\}^2 dx \tag{4.37}$$

Under the consideration, that the coefficients up to the n th order are constrained

$$\epsilon_x = \int_{x_{min}}^{x_{max}} \{(\boldsymbol{\alpha}_n - \boldsymbol{\beta}_n)x^n\}^2 dx \tag{4.38}$$

Consequently, evaluating the integral leads to,

$$\varepsilon_x = (\alpha_n - \beta_n)^2 \frac{x_{max}^{2n+1} - x_{min}^{2n+1}}{2n+1}. \quad (4.39)$$

Since x_{max} and x_{min} are constant for a calculation the factor,

$$k = \frac{x_{max}^{2n+1} - x_{min}^{2n+1}}{2n+1} \quad (4.40)$$

is a constant as well. Hence, the *combined error* ε_x is directly proportional the squared Taylor difference Δt_{fg}^n ,

$$\varepsilon_x \propto (\alpha_n - \beta_n)^2 \quad (4.41)$$

Numerical Combined Error

Compared to the analytical way, the combined error can also be formulated numerically. Hence, the measure can be derived as the 2-norm of combined residual vectors e_f (or e_g) as,

$$e_f = \mathbf{y} - \mathbf{f}(\mathbf{x}, \alpha) \quad (4.42)$$

$$e_g = \mathbf{y} - \mathbf{g}(\mathbf{x}, \beta) \quad (4.43)$$

$$E_{fg} = \|e_f - e_g\|_2^2 = \|(a_n - b_n)z\|_2^2 = (a_n - b_n)^2 z^T z^n = (a_n - b_n)^2 \sum x_i^n \quad (4.44)$$

with,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_L \\ \mathbf{x}_R \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix}, \quad \mathbf{z} \triangleq \mathbf{x}^n$$

In coherence with the analytical case, the numerical implementation of the combined error measure also shows the direct proportionality to the squared Taylor difference Δt_{fg}^n .

Chapter 5

Generalization of the Constrained Polynomial Discontinuity Detection Approach

5.1 Detection of D^n Discontinuities

In order to generalize the method described in Chapter 4, the subsequent goal is to detect a discontinuity in the k -th derivative of a function, while higher and lower orders might be continuous. In the following part of this thesis, these discontinuities are referred to as D^n discontinuities.

Definition 1. A function f has a D^k discontinuity at an interstitial point ξ_i , if that very function shows a jump in the k -th derivative, but the $(k + 1)$ -th derivatives to the left and to the right of ξ_i are identical.

Given Definition 1, the *constrained coupled polynomial approach* can be formulated with the following coefficients, for a function with a discontinuity in the first derivative,

$$\begin{aligned}\alpha_0 &= \beta_0 \\ \alpha_1 &\neq \beta_1 \\ \alpha_2 &= \beta_2.\end{aligned}\tag{5.1}$$

This condition then afterwards results in the constraint vector

$$\mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\tag{5.2}$$

In general, this formulation can be constructed in matrix form with the binary *constraint vector* filled with *ones* except for the predefined orders i_1, i_2, \dots, i_r ,

$$\begin{aligned}\mathbf{c} &= [c_1 \ c_2 \ \dots \ c_n]^T \\ &= \left[1 \ \dots \ \underbrace{0}_{n-i_1+1} \ 1 \ \dots \ \underbrace{0}_{n-i_r+1} \ 1 \right]^T\end{aligned}\tag{5.3}$$

Consequently, for every *one* in the *constraint vector* c the corresponding coefficients are coupled and continuity is enforced for the respective derivative. Whereas, for all *zeros* no constraints are defined. As a result, D^n discontinuities can be detected for those orders.

For example the *constraint vector* c required to detect a *first* order derivative discontinuity with a polynomial of degree three would result in,

$$c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad (5.4)$$

In general, the *coefficients* for the constrained coupled polynomial approximation are given as,

$$\alpha_i = \beta_i, \text{ for } i = 1, \dots, n, \quad i \neq i_1, \dots, i_r. \quad (5.5)$$

This could be written in matrix form similar to Equation 4.19 as,

$$C \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = 0 \quad (5.6)$$

with the *constraint matrix* C defined as

$$C = [C_s \ -C_s]. \quad (5.7)$$

whereas the diagonal matrix C_s is constructed from the *constraint vector* c as follows,¹

$$\begin{aligned} C_s &= \text{diag}(c) \\ &= \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix} \end{aligned} \quad (5.8)$$

Consequently, the *constraint matrix* C has the following structure,

$$C = \left[\begin{array}{cccc|cccc} c_1 & 0 & \dots & 0 & -c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 & 0 & -c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n & 0 & 0 & \dots & -c_n \end{array} \right] \quad (5.9)$$

¹ Even though algebraically more precise the constraint sub matrix C_s is written as, $C_s = I_n \circ (\mathbf{1}_n^T \otimes c)$, to maintain a more readable formula structure, the notation $\text{diag}(c)$ was chosen in Formula 5.8.

From here onwards, the procedure remains similar to the one described in Equation 4.21 to Equation 4.26 with the goal to calculate the *Taylor differences* $\Delta t_{fg}^{(n)}$ of the desired derivatives.

Beginning with Equation 4.11, the approximation to the left and the right is stated in block matrix form as,

$$\begin{bmatrix} \mathbf{V}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \quad (5.10)$$

Unlike the approach before, the implementation of the constraint matrix \mathbf{C} looks slightly different, resulting in the following structure,

$$\begin{bmatrix} c_1 & 0 & \dots & 0 & -c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 & 0 & -c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n & 0 & 0 & \dots & -c_n \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0}. \quad (5.11)$$

Again the following definitions are considered,

$$\mathbf{V} \triangleq \begin{bmatrix} \mathbf{V}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix}, \quad \boldsymbol{\gamma} \triangleq \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \quad \mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix}.$$

From here onwards the procedure gets very similar again, with the least squares optimization problem stated as,

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{V}\boldsymbol{\gamma}\|_2^2 \quad (5.12)$$

and

$$\mathbf{C}\boldsymbol{\gamma} = \mathbf{0}. \quad (5.13)$$

Since $\boldsymbol{\gamma}$ has to belong to the null-space of \mathbf{C} ,

$$\mathbf{N} = \text{Null}\{\mathbf{C}\} \quad (5.14)$$

leads to

$$\boldsymbol{\gamma} = \mathbf{N}\boldsymbol{\delta}. \quad (5.15)$$

As a result, the least squares minimization becomes,

$$\min_{\boldsymbol{\delta}} \|\mathbf{y} - \mathbf{V}\mathbf{N}\boldsymbol{\delta}\|_2^2. \quad (5.16)$$

Consequently, $\boldsymbol{\delta}$ minimizes the least squares problem with,

$$\delta = (\mathbf{V}\mathbf{N})^+ \mathbf{y}. \quad (5.17)$$

Thus, the coefficients are given by,

$$\gamma = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{N}(\mathbf{V}\mathbf{N})^+ \mathbf{y} \quad (5.18)$$

Finally the *Taylor differences* $t_{fg}^{(n)}$ are derived by,

$$\Delta t_{fg}^{(n)} = \begin{bmatrix} 1 \\ \mathbf{0}_{d_{L-1}} \\ -1 \\ \mathbf{0}_{d_{R-1}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = d\gamma = d\mathbf{N}(\mathbf{V}\mathbf{N})^+ \mathbf{y}. \quad (5.19)$$

For further details regarding the principles of this method, the reader is referred to Chapter 4.

5.2 Synthetic Datasets

In order to validate a method for change point detection or in the case of this thesis discontinuity detection, an algorithm needs to be tested on a practical dataset. Since, the generalization of the discontinuity detection method discussed and validated in this thesis (Ninevski / O'Leary 2018 [1]) is still in progress the logical first step to start a validation process is to begin with *synthetic datasets*. For this reason various time series datasets, with known statistical parameters including *change point size, location* and *order* were created. Consequently, these datasets with all relevant parameters are going to be introduced in the following section of this chapter.

To ensure a considerably easier structure and overview regarding the usage of the synthetic datasets a more or less descriptive notation was chosen for their naming. Hereby, the name indicates in which derivatives the observation is continuous or discontinuous. For example, an artificial dataset which is called D^2D^0 has one or more first order discontinuities and is continuous in the zeroth and second order.

5.2.1 D^2D^0 - Dataset

As the first dataset, to test the generalized approach, the D^2D^0 - Dataset was designed to be continuous in the signal itself and in its second order derivative, whereas the first order derivative is discontinuous at a point v . This characteristic is also referred to as being D^2 and D^0 continuous,

but D^1 discontinuous. The created dataset is shown in Figure 5.1 with no added white Gaussian noise. In the course of this work, this and other datasets are tested under the corruption of noise. In that case, the artificial observations presented in this subchapter are used with additional noise.

As the dataset was created by evaluating one polynomial to the left and one polynomial to the right of an interstitial point. Where, the coefficients of these polynomials only differ in one order - the discontinuous order. Thus, the parameters describing the dataset are,

$$\alpha = \begin{bmatrix} -2 \\ -8 \\ -5 \\ 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} -2 \\ -8 \\ 10 \\ 1 \end{bmatrix}, \quad v = 256, \quad n = 512, \quad \sigma = 0.$$

α ... coefficients left polynomial

β ... coefficients right polynomial

v ... discontinuity location

n ... number of observation points

σ ... standard deviation

Hence, the dataset and its relevant derivatives looks as follows.

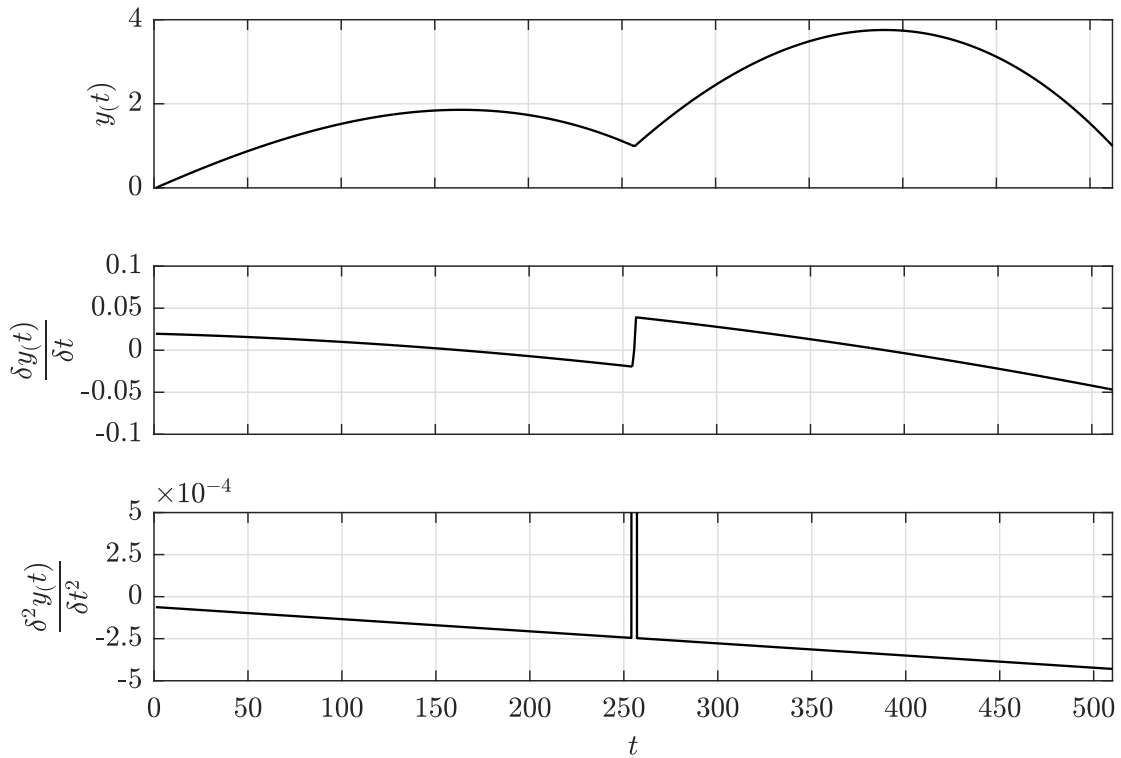


Fig. 5.1: Function $y(t)$, first order derivative $\frac{\delta y(t)}{\delta t}$ and second order derivative $\frac{\delta^2 y(t)}{\delta t^2}$ of the synthetic dataset $D^2 D^0$, which is continuous in the function and the second order derivative and discontinuous in the first order at the point $v = 256$.

As shown in Figure 5.1, the constructed $D^2 D^0$ dataset has the desired properties - a first order D^1 discontinuity, while being continuous in the observation itself D^0 and in its second derivative D^2 . These properties are visualized in the respective subplots, whereas as the second on shows the jump discontinuity clearly. Note that the jump in the second derivative - third subplot - is due to numeric reasons. Given, in the numerical case, the jump in the second order derivative does not happen at one point, but between two. Thus, the slope at this point is relatively high. Analytically, this jump should be infinitely high over an infinitely narrow time step. Anyway, as in the numeric case, shown here, the time step is still considerably short.

5.2.2 $D^3 D^1 D^0$ - Dataset

Another artificial use case to test the generalization even further was created by the $D^3 D^1 D^0$ - dataset. As the name suggests, this example is continuous in the third D^3 , first D^1 and zeroth D^0 order derivative, while showing an abrupt change in form of a jump discontinuity at a point v . Again, the base version of the data is presented without added white Gaussian noise. However, other versions used later have added noise in addition to the basic properties described below.

To create the dataset the following parameters were used,

$$\alpha = \begin{bmatrix} -2 \\ -7 \\ -3 \\ 3 \end{bmatrix}, \quad \beta = \begin{bmatrix} -2 \\ 5 \\ -3 \\ 3 \end{bmatrix}, \quad v = 256, \quad n = 512, \quad \sigma = 0.$$

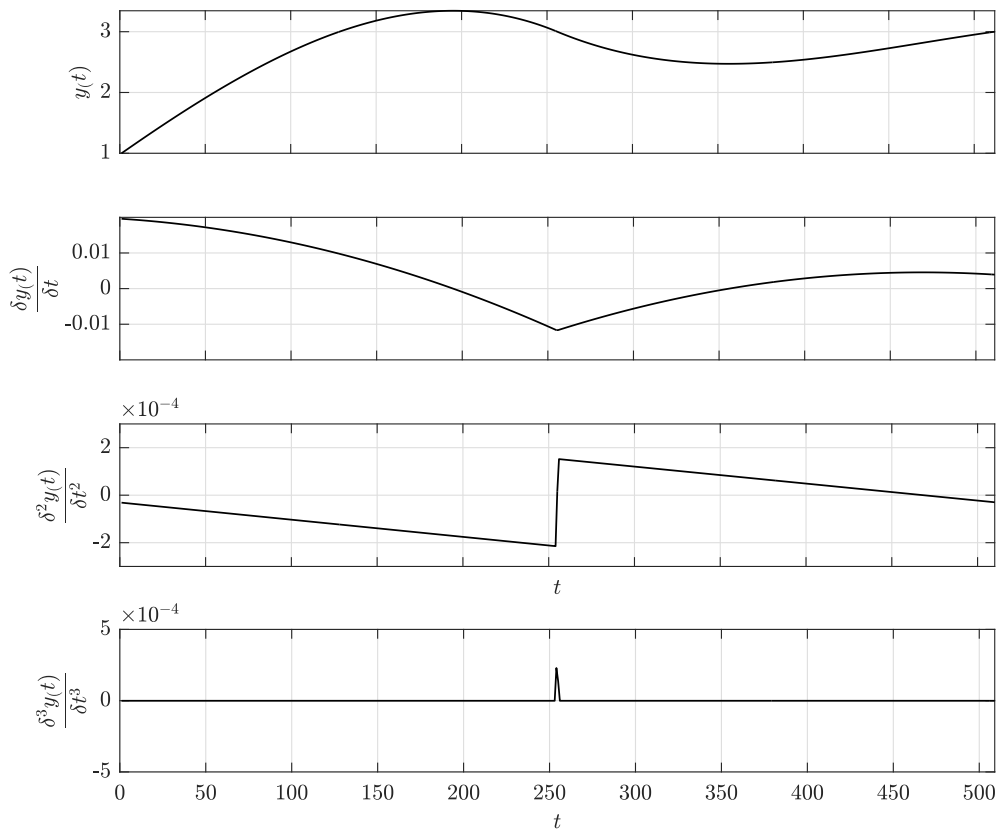


Fig. 5.2: Function $y(t)$, first order derivative $\frac{\delta y(t)}{\delta t}$, second order derivative $\frac{\delta^2 y(t)}{\delta t^2}$ and third order derivative $\frac{\delta^3 y(t)}{\delta t^3}$ of the synthetic dataset $D^3 D^1 D^0$, which is continuous in the function, first order and third order derivative and discontinuous in the second order one at the point $v = 256$.

Again, the artificial signal has the properties as designed, showing a clear D^2 jump discontinuity in the second order derivative - as Figure 5.2 suggests. Similar, to the case before the peak in the fourth subplot of Figure 5.2 is due to the already described numerical reasons.

5.2.3 $D^4D^3D^1D^0$ - Dataset

In order to test another use case, namely the detection of multiple derivative change points, the $D^4D^3D^1D^0$ dataset was designed, having two second order discontinuities at the points v_1 and v_2 . By doing so, the artificial observation is continuous in every other order.

Regarding the construction of this test sample, the approach was different to the one before. In fact, a bottom up approach was used to create the signal, by starting with a continuous function representing the fourth derivative - a sine in that case. Afterwards, the continuous functions was integrated numerically, yielding the third derivative of the desired dataset, which is still continuous. Then the second order derivative is computed by another integration, whereas it is also still continuous. As a next step the desired second order jump discontinuities are added to the second order derivative at the points v_1 and v_2 . Finally, after two more numerical integration steps, the base function - zeroth order derivative - is created. A visualization of the dataset creation can be seen in Figure 5.3, where the initial function, the sine, is displayed in the last subplot and the jump discontinuities are added in the third subplot.

$$v_1 = 200, \quad v_2 = 400, \quad n = 450, \quad \sigma = 0.$$

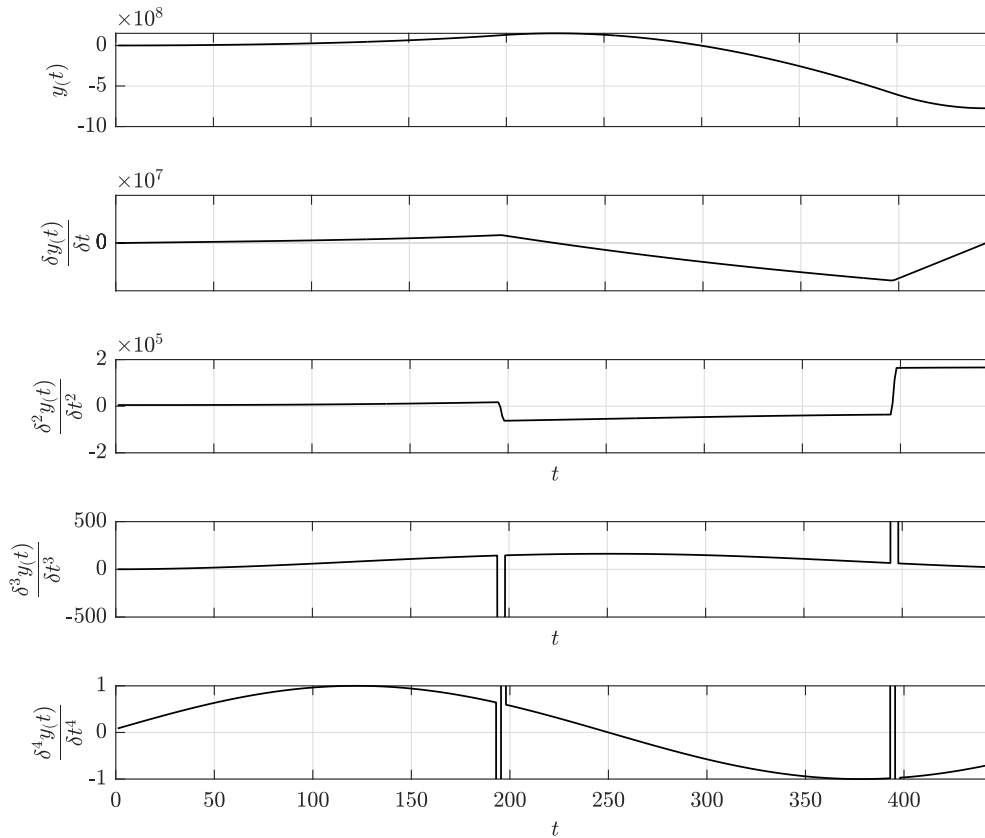


Fig. 5.3: Function $y(t)$, first order derivative $\frac{\delta y(t)}{\delta t}$, second order derivative $\frac{\delta^2 y(t)}{\delta t^2}$, third order derivative $\frac{\delta^3 y(t)}{\delta t^3}$ and fourth order derivative $\frac{\delta^4 y(t)}{\delta t^4}$ of the synthetic dataset $D^4 D^3 D^1 D^0$, which is continuous in the function, the first, the third and fourth order derivative and discontinuous in the second order at the points $v_1 = 200$ and $v_2 = 400$.

Once again, just as before, the desired properties - second order discontinuity - of the created dataset can be seen in Figure 5.3. Also similar as before, the outliers at the to change point locations v_1 and v_2 due to known numerical reasons are visualized in the fourth and fifth subplot of Figure 5.3.

5.3 Triple Peak Feature

This section is dedicated to the analysis of a behaviour occurring in the discontinuity detection algorithm after the implementation of a constraint vector (rather than just a scalar). More precisely, the issue hereby is that after the implementation the relevant plots of Taylor differences and residual as well as extrapolation residual errors were showing three instead of one peak around the suspected discontinuous point. In other words, the fit around the discontinuity was significantly different with constrained zeroth and second order (looking for first order discontinuity) compared

to the case where just the first order was constrained. See Figure 5.4 for the described characteristic.

5.3.1 D^2D^0 - Dataset

As a first test, the generalized method was applied to the synthetic D^2D^0 dataset with an artificial discontinuity in the first order derivative at $t = 256$.

Thus, the necessary parameters were set as follows for the generalized approach,

$$C = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array} \right], \quad l_s = [6, 6]$$

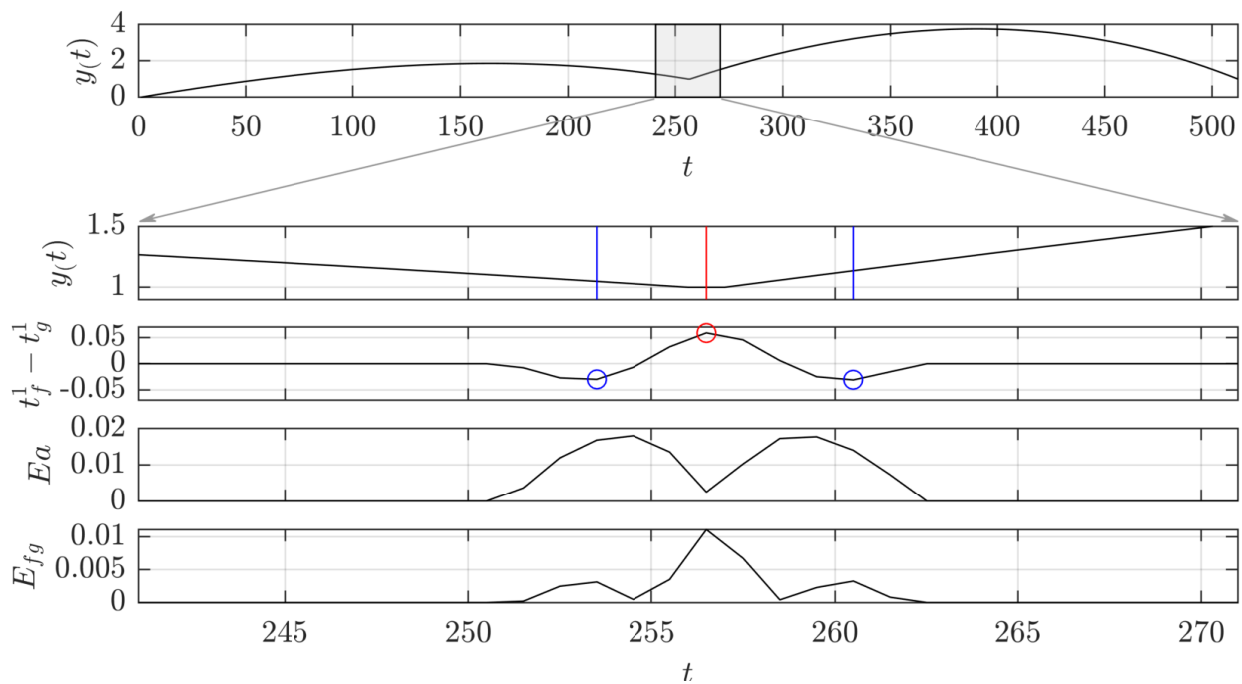


Fig. 5.4: Detecting a possible D1 discontinuity with enforced D^0 and D^2 continuities with support length of $l_s = 6$ to the left and to the right. The dataset tested here is a synthetic one with a first order discontinuity at $t = 256$ and no added noise. The observation itself and the second derivative of the artificial signal are continuous.

Now the goal is to get a better understanding where this behaviour is coming from. Furthermore, a more precise explanation is part of the following paragraphs. To do so, the first step for getting a more detailed view of the issue was to plot all respective polynomials for the approximation around the change point. The approximation around the point of change is shown in Figure 5.5.

Furthermore, it gives a more intuitive understanding, where the two local minima of the first order Taylor differences are coming from in Figure 5.4. One can see easily, that, as soon as the change point ($t = 256$) is in the range of the right approximation window, the best fit of the right polynomial changes significantly to the left one, such that the slope of the polynomial decreases. The same is true for the local minimum on the right side of the discontinuity.

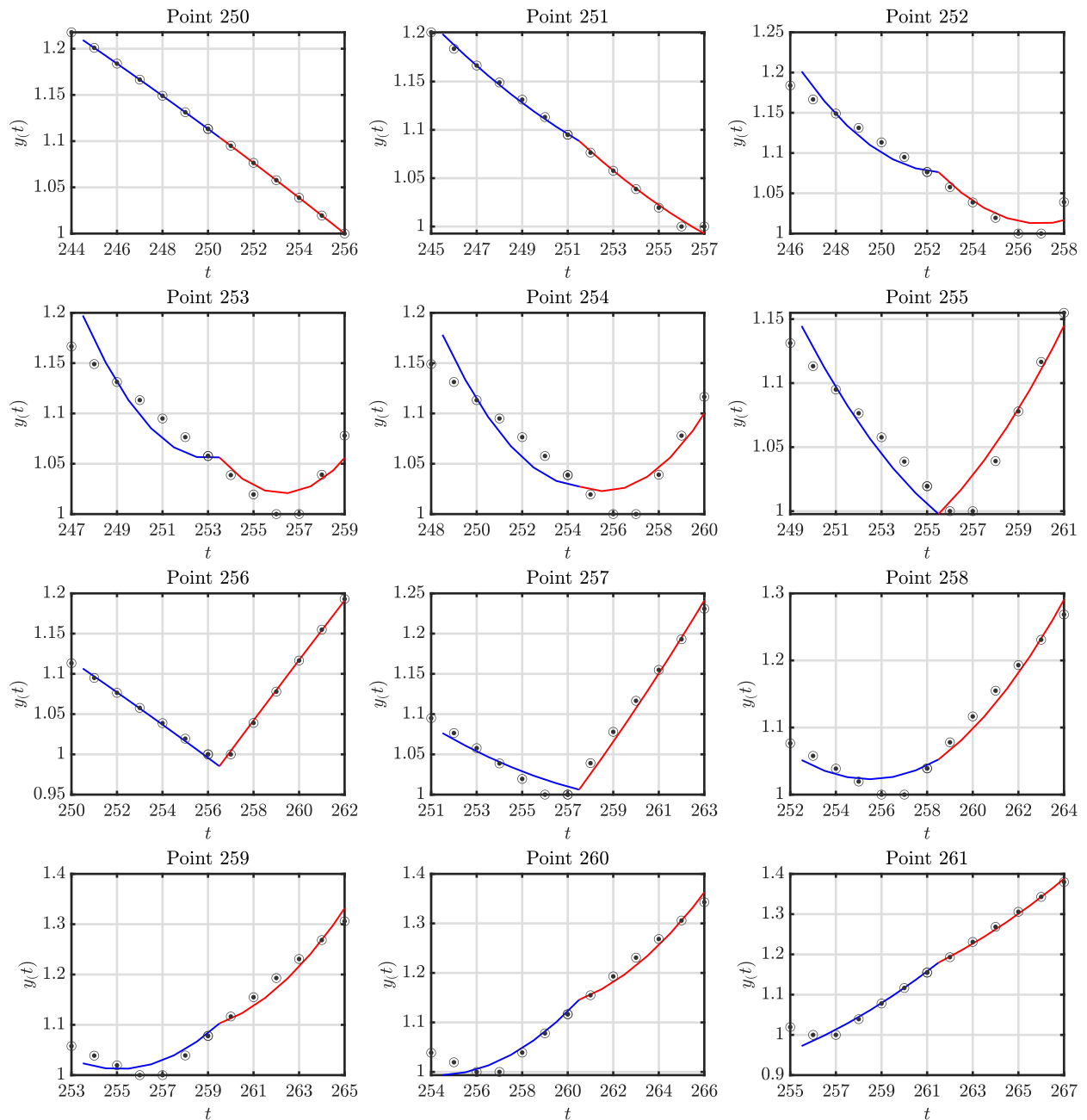


Fig. 5.5: Evaluation of the left and the right polynomial (second order) around the discontinuous point $t = 256$ with the obtained coefficients under constrained D^0 and $D1$ continuity.

Similar to the approach to detect a D^1 discontinuity in Figure 5.4, Figure 5.6 delivers the outcome of a C^1 discontinuity detection with constrained C^0 continuity. In contrast to the approach presented in Figure 5.4 there is, just as expected, only one peak in the first order Taylor differences. Subsequently, the same analysis as before in Figure 5.5 is going to be performed for this case with the following parameters,

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \quad l_s = [6, 6]$$

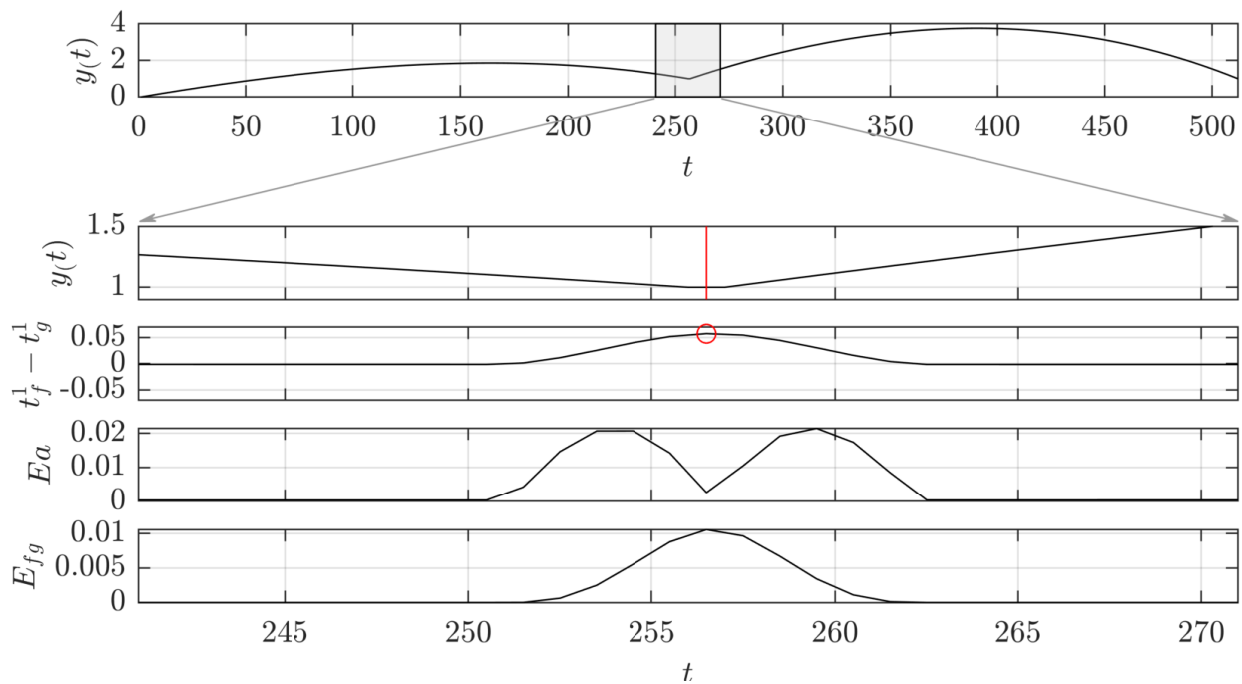


Fig. 5.6: Detecting a possible C^1 discontinuity with enforced C^0 continuity and a support length of $l_s = 6$ to the left and to the right. The dataset tested here is a synthetic one with a first order discontinuity at $t = 256$ and no added noise. The observation itself and the second derivative of the artificial signal are continuous.

Unsurprisingly, one can see in Figure 5.7 that, just like before, the right polynomial adapts to the data points after the first order jump. However, now the slope directly increases until the interstitial point reaches the discontinuity at $t = 256$ resulting in a rise in first order Taylor differences till the change point.

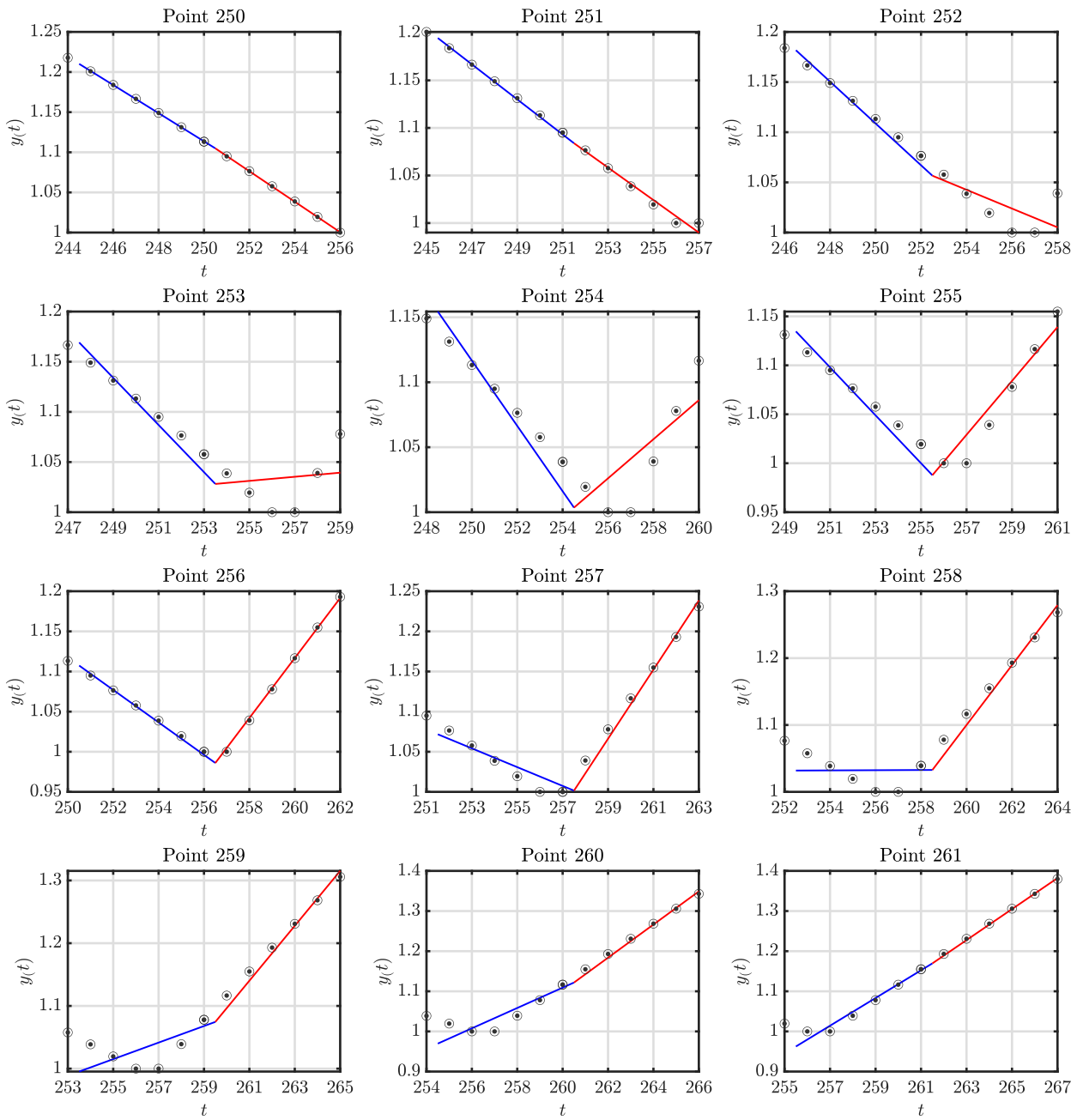


Fig. 5.7: Evaluation of the left and the right polynomial around the discontinuous point $t = 256$ with the obtained coefficients under C^0 continuity.

Findings so far:

1. No impact of the constraint vector
2. Number of additional peaks is directly dependent on the order of the approximation polynomial
3. Position of additional peaks varies with support length
4. The pattern can be considered as normal under the given conditions (no problem in the algorithm or the generalization)

5.3.2 $D^3D^1D^0$ - Dataset

In order to further evaluate the behaviour of the algorithm with the new constraint vector implementation, a similar test like before was conducted. However, this time the second order Taylor coefficients were the ones to take a closer look at, to get a measure for a possible D^2 discontinuity. For this purpose, a synthetic dataset with the following specifications was created. The artificial data shows a D^2 discontinuity at the point $v = 256$, while being D^3 , D^1 and D^0 continuous.

In order to ensure the necessary constraints the constraint matrix C was set to,

$$C = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array} \right],$$

with the support length vector

$$l_s = [20, 20].$$

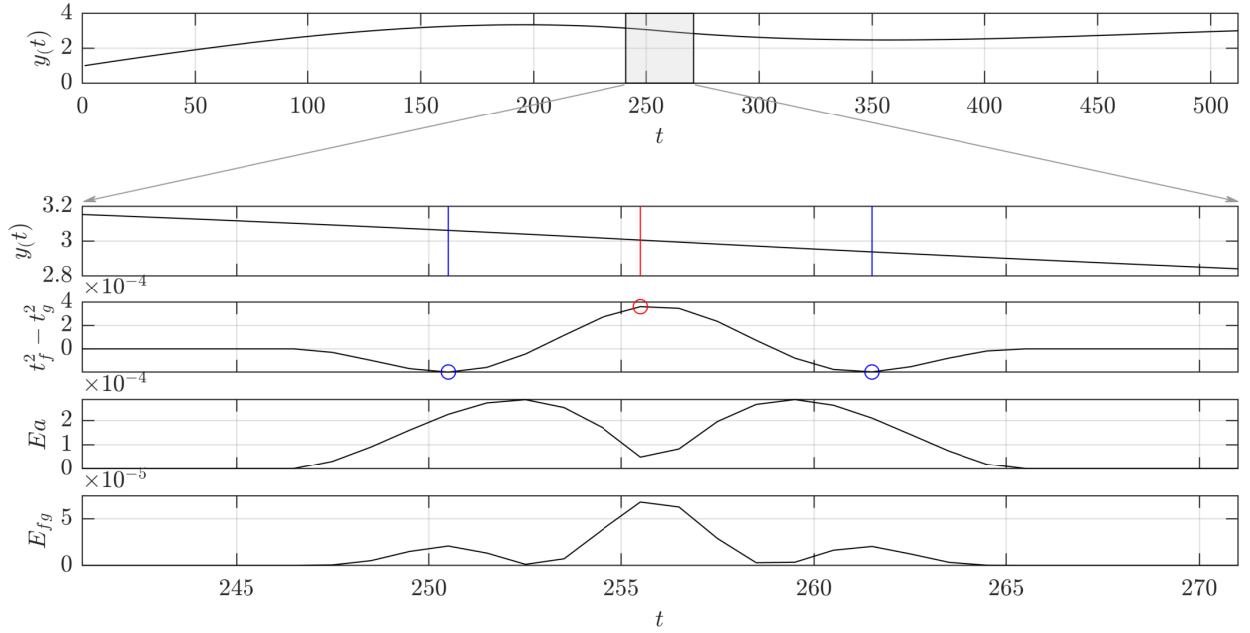


Fig. 5.8: Detecting a possible D^2 discontinuity with enforced D^0 , D^1 and D^3 continuities with support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a second order discontinuity at $t = 256$ and no added noise.

As shown in Figure 5.8 the algorithm detects the D^2 discontinuity just like it should. However, as before with the D^2D^0 -dataset, there are also two additional additional negative peaks around the change point location v . This finding further confirms that the triple peak pattern is a phenomena coming up consistently in settings with the following constraints.

$$\begin{aligned}
 \alpha_3 &= \beta_3 \\
 \alpha_2 &\neq \beta_2 \\
 \alpha_1 &= \beta_1 \\
 \alpha_0 &= \beta_0
 \end{aligned} \tag{5.20}$$

To compare these results, with the method before the generalization - continuity enforced only up to the $(n - 1)$ th derivative - the tunable parameters where set to,

$$\mathbf{C} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array} \right], \quad \mathbf{l}_s = [20, 20]$$

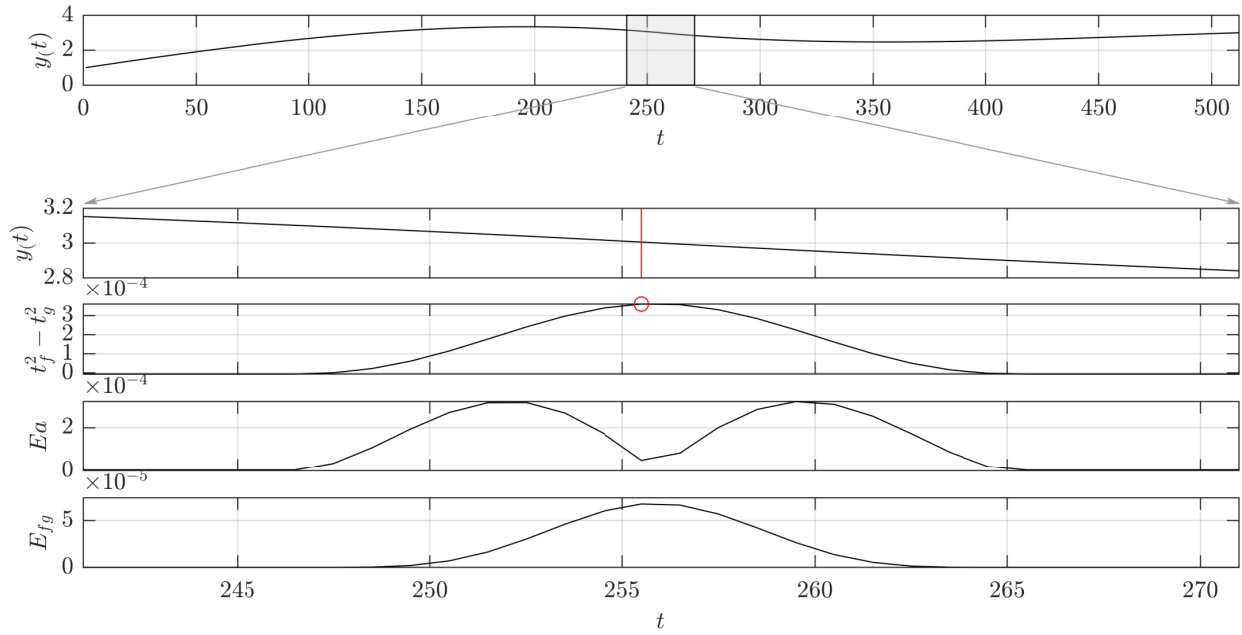


Fig. 5.9: Detecting a possible C^2 discontinuity with enforced C^0 and C^1 continuities with support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a second order discontinuity at $t = 256$ and no added noise.

In contrast, Figure 5.9 gives an insight in the analysis of the identical dataset with enforced C^0 and C^1 continuity. Hereby, only the second order discontinuity at the sample point $v = 256$ was detected.

5.4 Multiple Discontinuities

With the purpose to validated the performance of the generalized method, with a different dataset. The algorithm was applied to the $D^4D^3D^1D^0$ dataset, with two second order jump discontinuities D^2 , already introduced earlier in this chapter.

The method was applied under following conditions,

$$C = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{array} \right], \quad l_s = [20, 20]$$

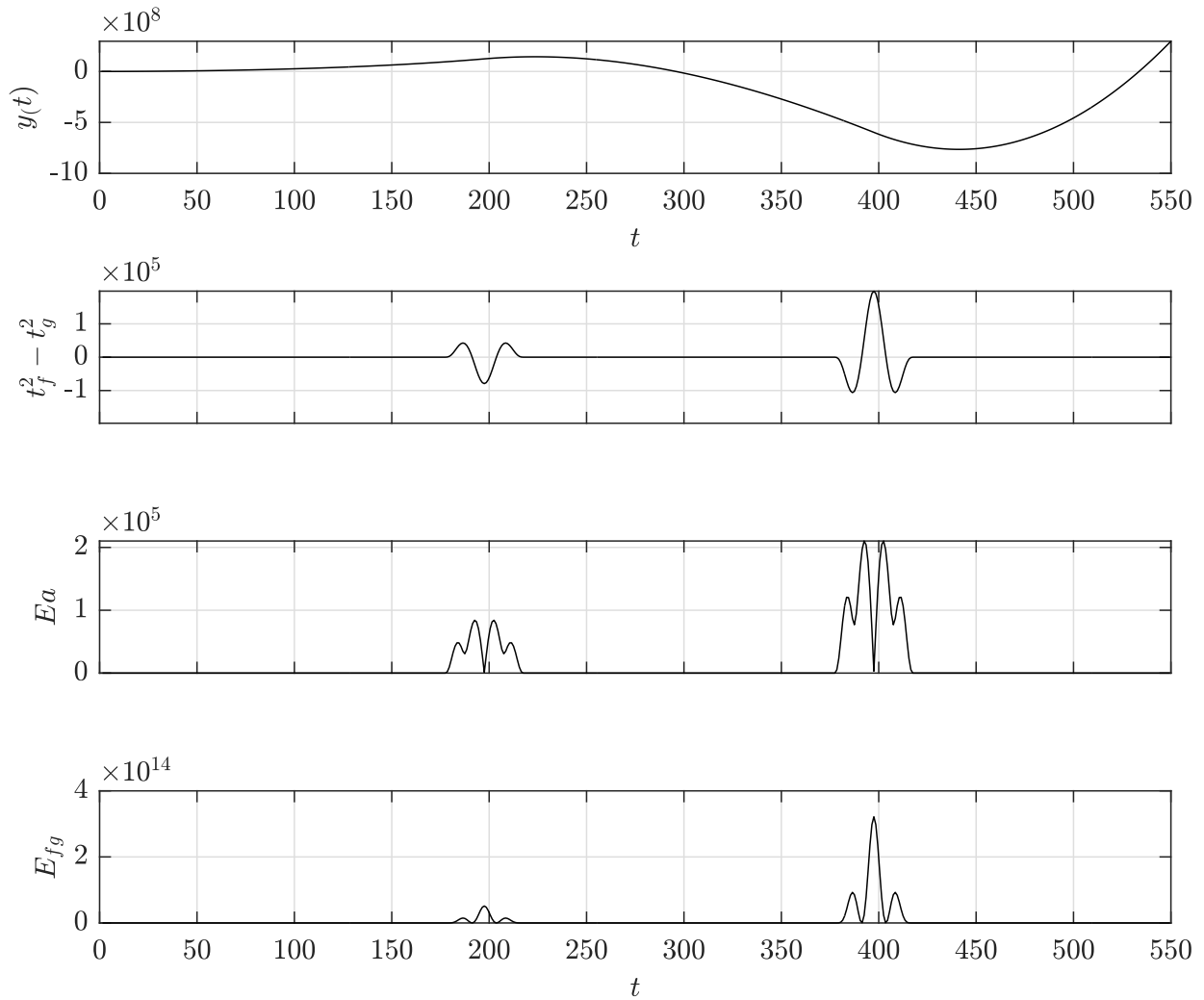


Fig. 5.10: Detecting a possible D^2 discontinuity with enforced D^0 , D^1 , D^3 and D^4 continuities with support length of $ls = 20$ to the left and to the right. The dataset tested here is a synthetic one with two second order discontinuities at $v_1 = 200$ and $v_2 = 400$ and no added noise.

Clear characteristic patterns, as investigated in the subsection before in this chapter, around the suspected locations of the discontinuities in the second order Taylor differences $t_f^2 - t_g^2$, are shown in the second subplot of Figure 5.10. Hereby, it is important to notice that the first second order jump discontinuity is a negative one, whereas the other one is positive. Subsequently, the *Triple Peak* indicates this difference, by having its peak in the respective direction.

As a result, the generalized method was tested successfully with more than one change point and the direction of the detected discontinuity has a direct impact on the direction of the pattern.

5.5 Noise Testing

Contrary to the tests executed before, in this section the focus is on the performance of the *generalized method* with noisy data.

For this purpose the same datasets as before with added white Gaussian noise were used. In the first validation phase the main goal was to test if the algorithm is showing characteristic patterns in the *Taylor differences* Δt_{fg}^n . Hence, the peak finding algorithm used in the tests executed before (without noise) is not part of this test phase. In that case, only a qualitative decision is made, whether or not a characteristic pattern is present.

5.5.1 D^2D^0 – Noise - Dataset

In order to validate the performance of the *generalized method*, the next task is to detect a known first order discontinuity in the D^2D^0 dataset with added white Gaussian noise of $\sigma = 0.05$, both with constrained D^2 and D^0 continuities (generalized method; quadratic polynomial) and only with C^0 continuity (special case; linear polynomial).

Again with the parameters set to,

$$C = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array} \right], \quad \mathbf{l}_s = [20, 20]$$

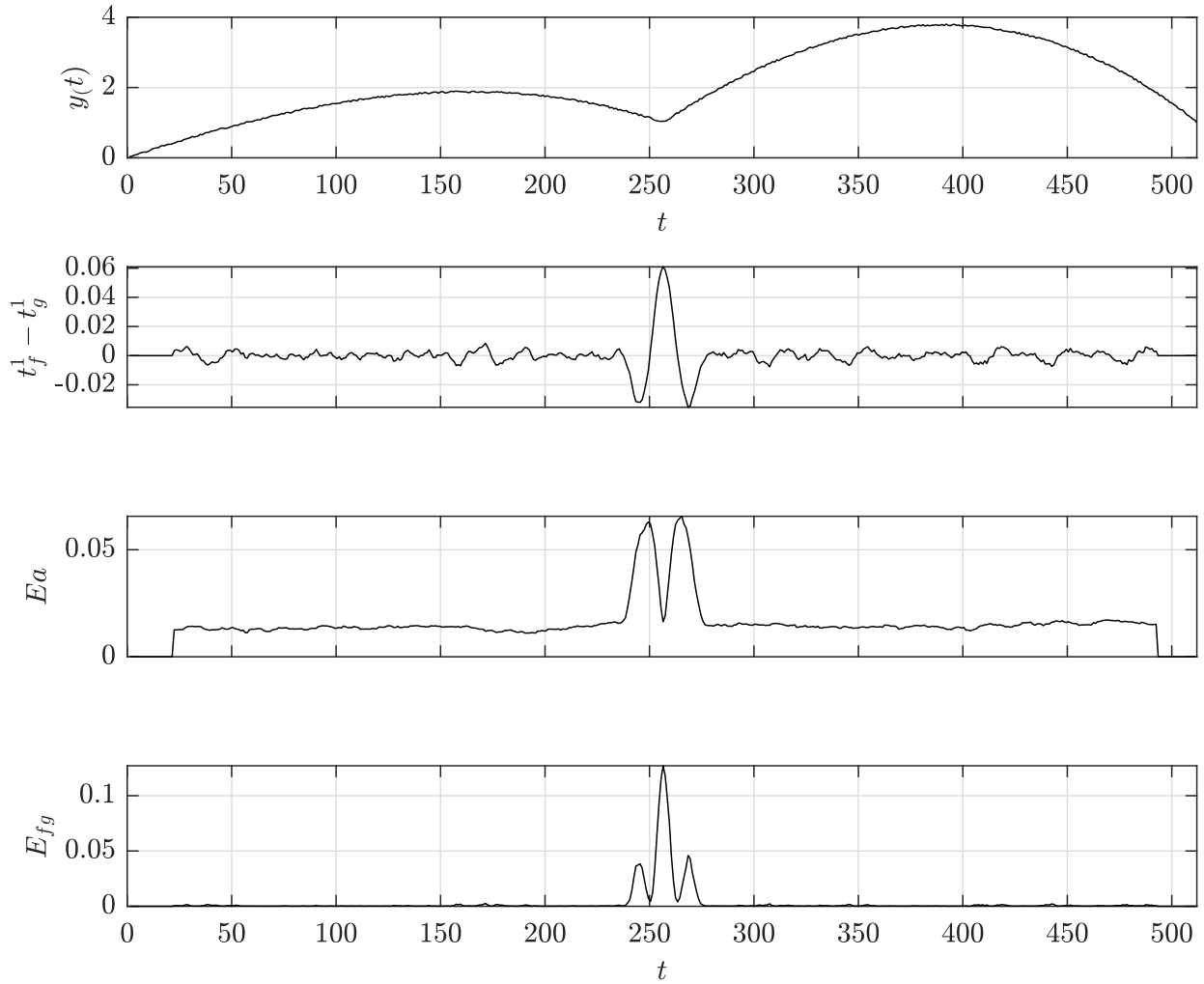


Fig. 5.11: Detecting a possible D^1 discontinuity with enforced D^0 and D^2 continuities with support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a first order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 0.05$. The observation itself and the second derivative of the artificial signal are continuous.

While the added white Gaussian noise is still relatively low in comparison to the detected signal, the first detection procedure executed with generalized version of the constrained polynomial method does show a characteristic pattern in the Taylor differences $t_f^1 - t_g^2$, as Figure 5.11 suggests. Hence, the first noise validation step can be seen as quite successful. With the aim to compare the results of the method before the generalization (implementation of the constraint matrix C), the identical dataset was investigated with the former method.

Hence, the parameters were set to,

$$C = \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{array} \right], \quad l_s = [20, 20]$$

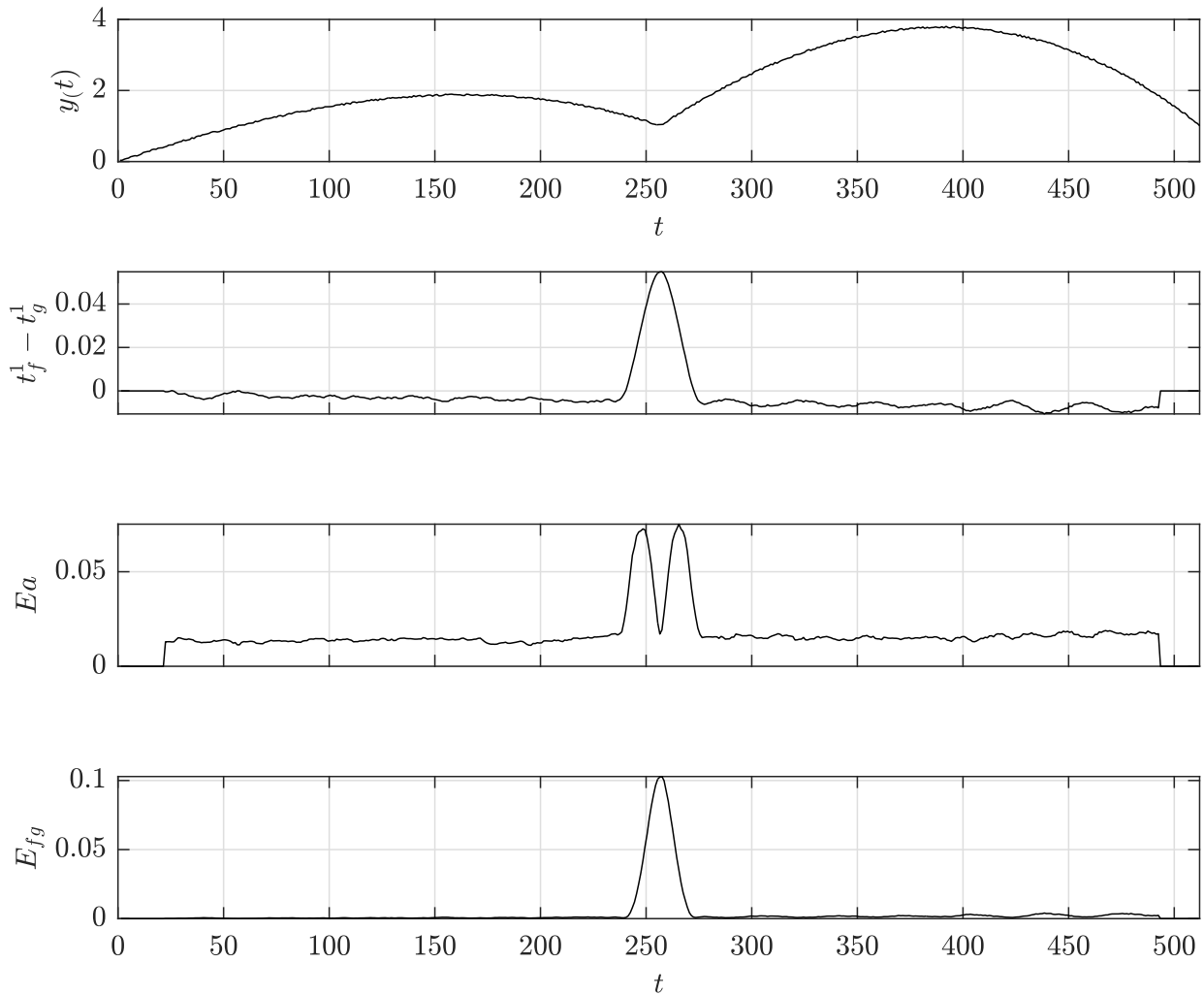


Fig. 5.12: Detecting a possible C^1 discontinuity with enforced C^0 continuity and a support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a first order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 0.05$. The observation itself and the second derivative of the artificial signal are continuous.

As described above, the comparison between the results of the now generalized method with the former approach was performed. The later results are shown in Figure 5.12. As a result, the conclusion can be made that, both approach deliver a clear characteristic pattern in the measures of interest, such as Taylor differences $t_f^1 - t_g^2$, the extrapolation error E_a and the interpolation error E_{fg} , albeit the former approach can be seen as slightly less sensitive to the noise corruption of the analysed dataset. This finding is going to be tested even further in the upcoming comparison.

5.5.2 $D^3D^1D^0$ – Noise - Dataset

$$C = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array} \right], \quad l_s = [20, 20]$$

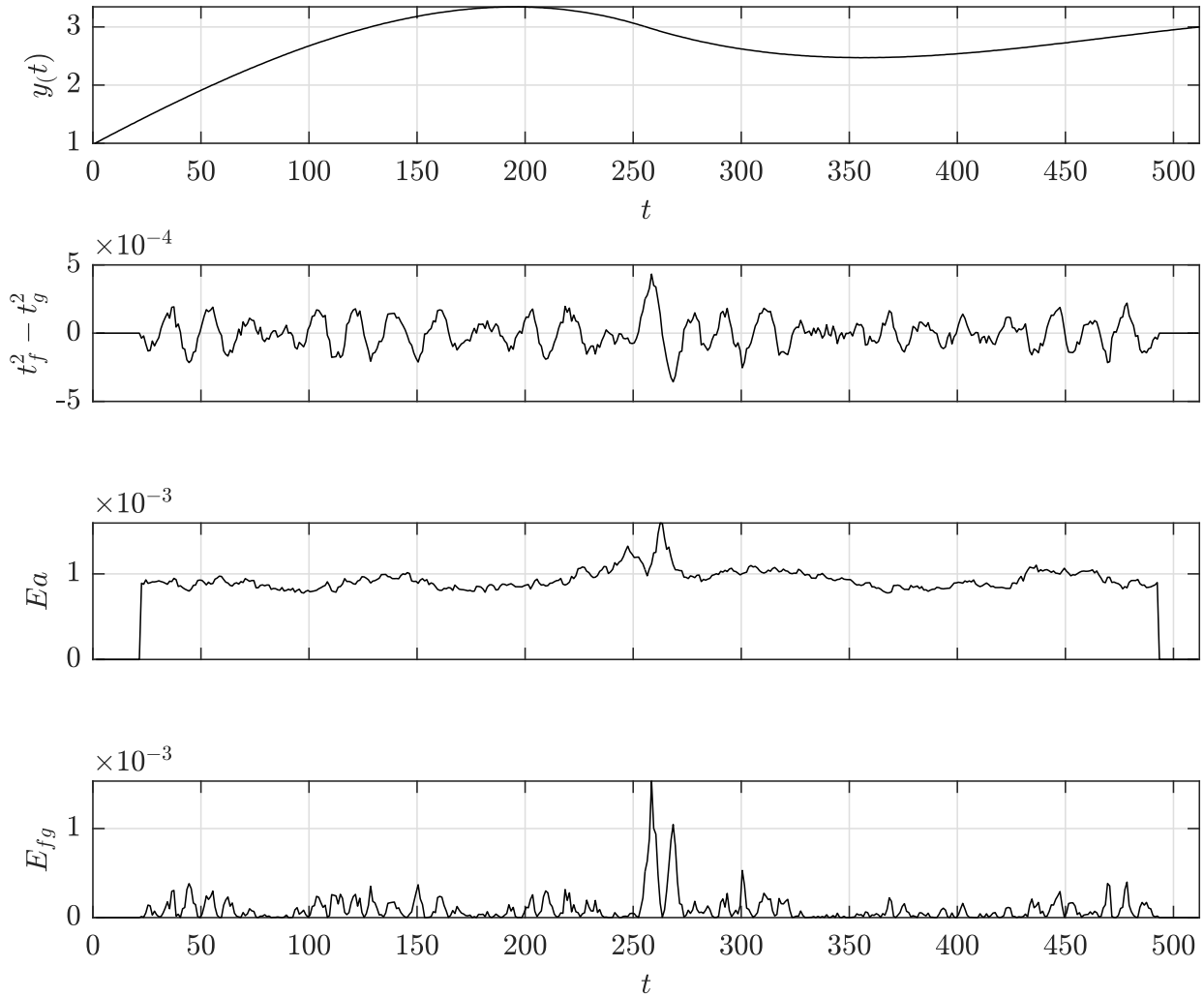


Fig. 5.13: Detecting a possible D^2 discontinuity with enforced D^0 , D^1 and D^3 continuities with support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a second order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 10^{-3}$.

Even though, the white Gaussian noise added to the $D^3D^1D^0$ dataset is relatively low, a characteristic pattern is not really established under the current circumstances. Therefore, the same test case is analysed with the former setting, with the parameters set to.

$$C = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array} \right], \quad l_s = [20, 20]$$

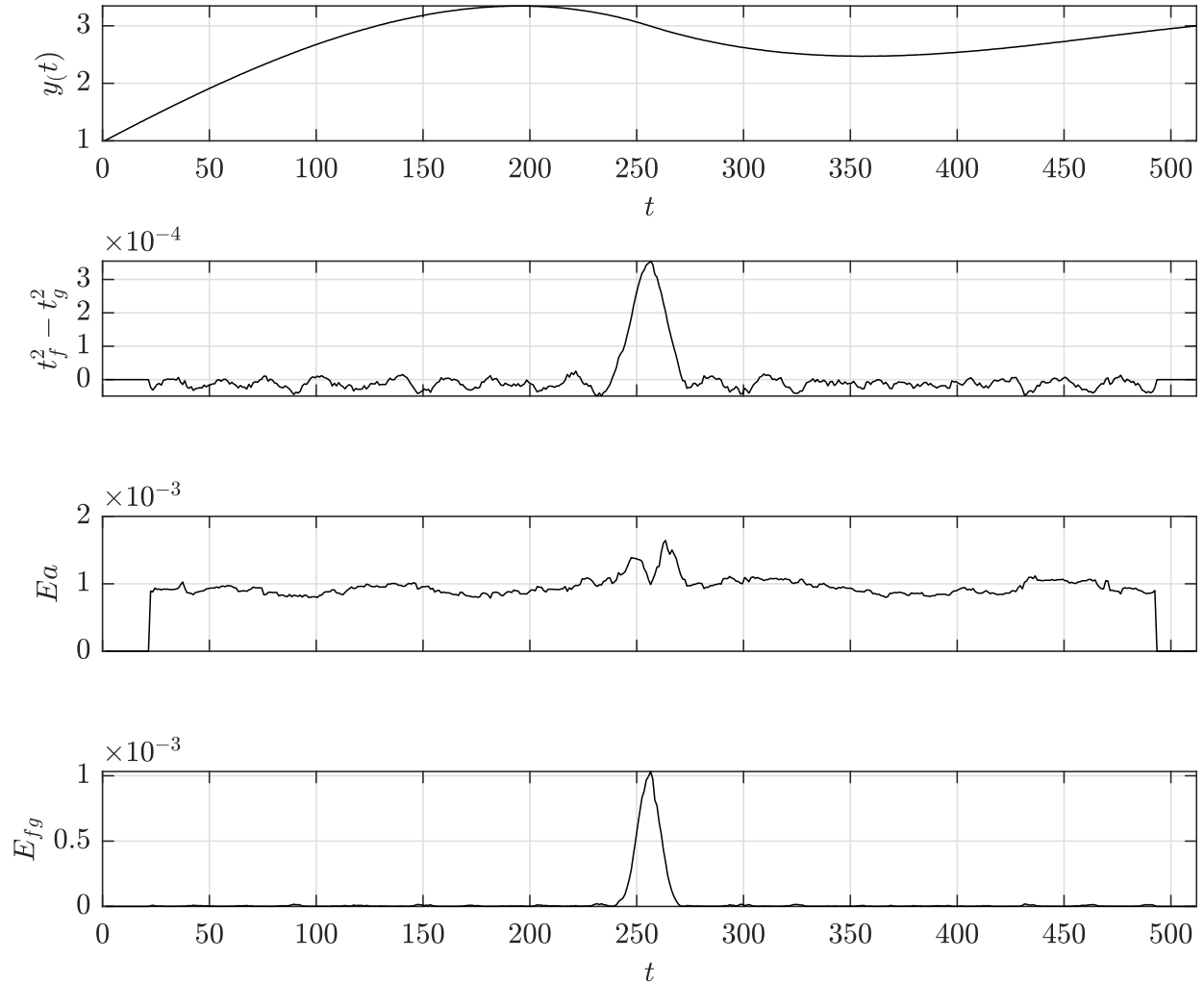


Fig. 5.14: Detecting a possible C^2 discontinuity with enforced C^0 and C^1 continuities with support length of $l_s = 20$ to the left and to the right. The dataset tested here is a synthetic one with a second order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 10^{-3}$.

Whereas, the detection of the D^1 discontinuity in the D^2D^0 dataset with added white Gaussian noise with $\sigma = 0.05$ was delivering clear peaks in the respective Taylor differences, the advanced case of the D^2 discontinuity detection of the $D^3D^1D^0$ dataset with added white Gaussian noise with $\sigma = 10^{-3}$ was less successful. Even though, the identical parameters were used to detect the suspected change point, the new generalized method hardly shows a characteristic profile in the second order Taylor differences $t_f^2 - t_g^2$ as shown in Figure 5.13. On the other hand, the not

generalized version of the detection method does show way better results, as Figure 5.14 suggests. Whereas, the same parameters were used to compare both variants. The reason for this behaviour could be similar to the mechanism described in the context of the *Triple Peak Feature* before in this chapter.

Consequently, the approach to deal with this discrepancy is to first adapt the *support length* l_s to improve the resolution of the method in this example, yielding the parameters,

$$C = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array} \right], \quad l_s = [30, 30]$$

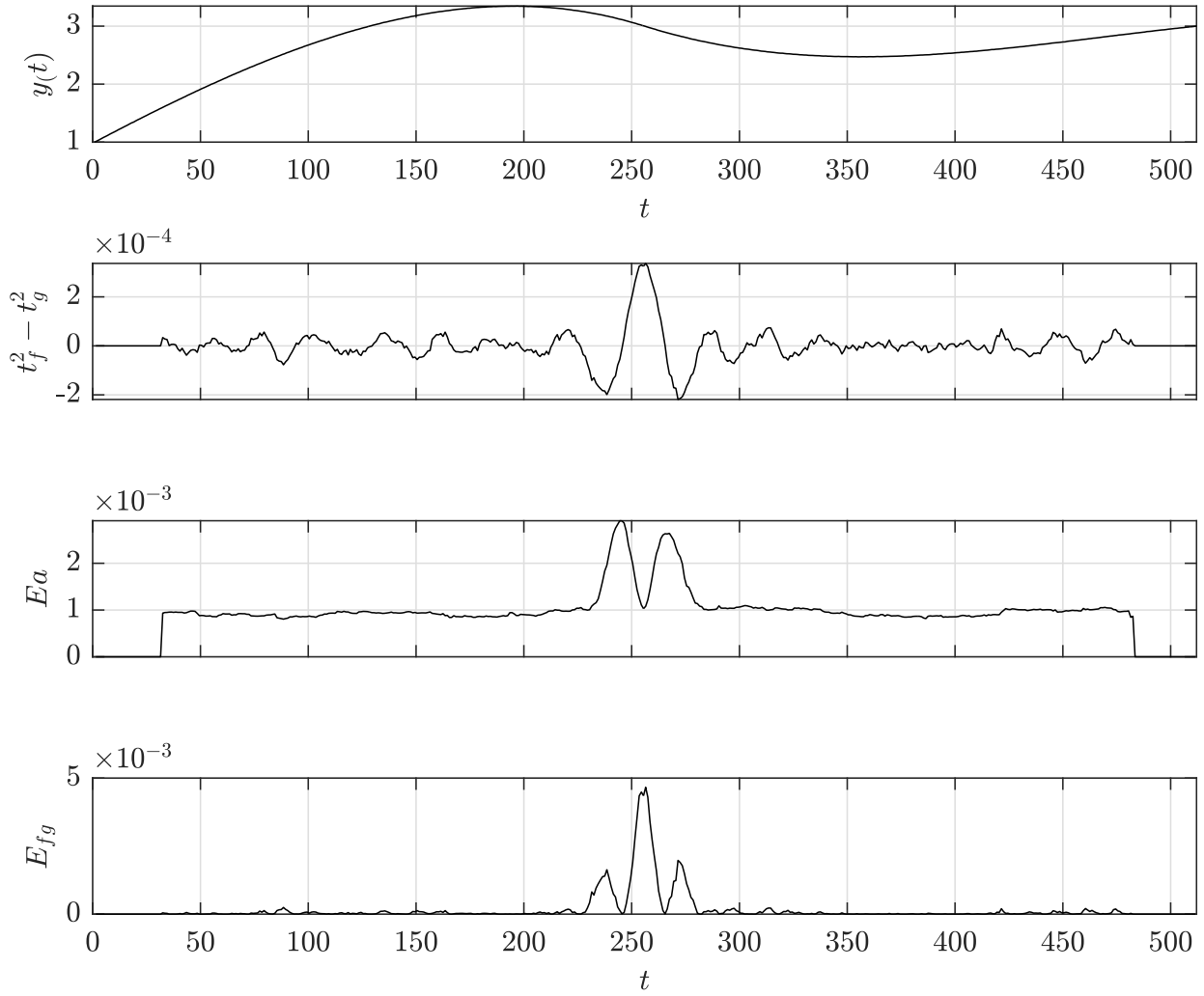


Fig. 5.15: Detecting a possible D^2 discontinuity with enforced D^0 , D^1 and D^3 continuities with support length of $l_s = 30$ to the left and to the right. The dataset tested here is a synthetic one with a second order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 10^{-3}$.

After increasing the support length vector l_s the analysis shows significantly better results in terms of a characteristic pattern in the second order Taylor differences $t_f^2 - t_g^2$.

As a next test case, relatively high white Gaussian noise $\sigma = 0.5$ was added to the $D^2 D^0$ dataset. Consequently, the generalized detection approach was applied with,

$$C = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array} \right], \quad l_s = [100, 100]$$

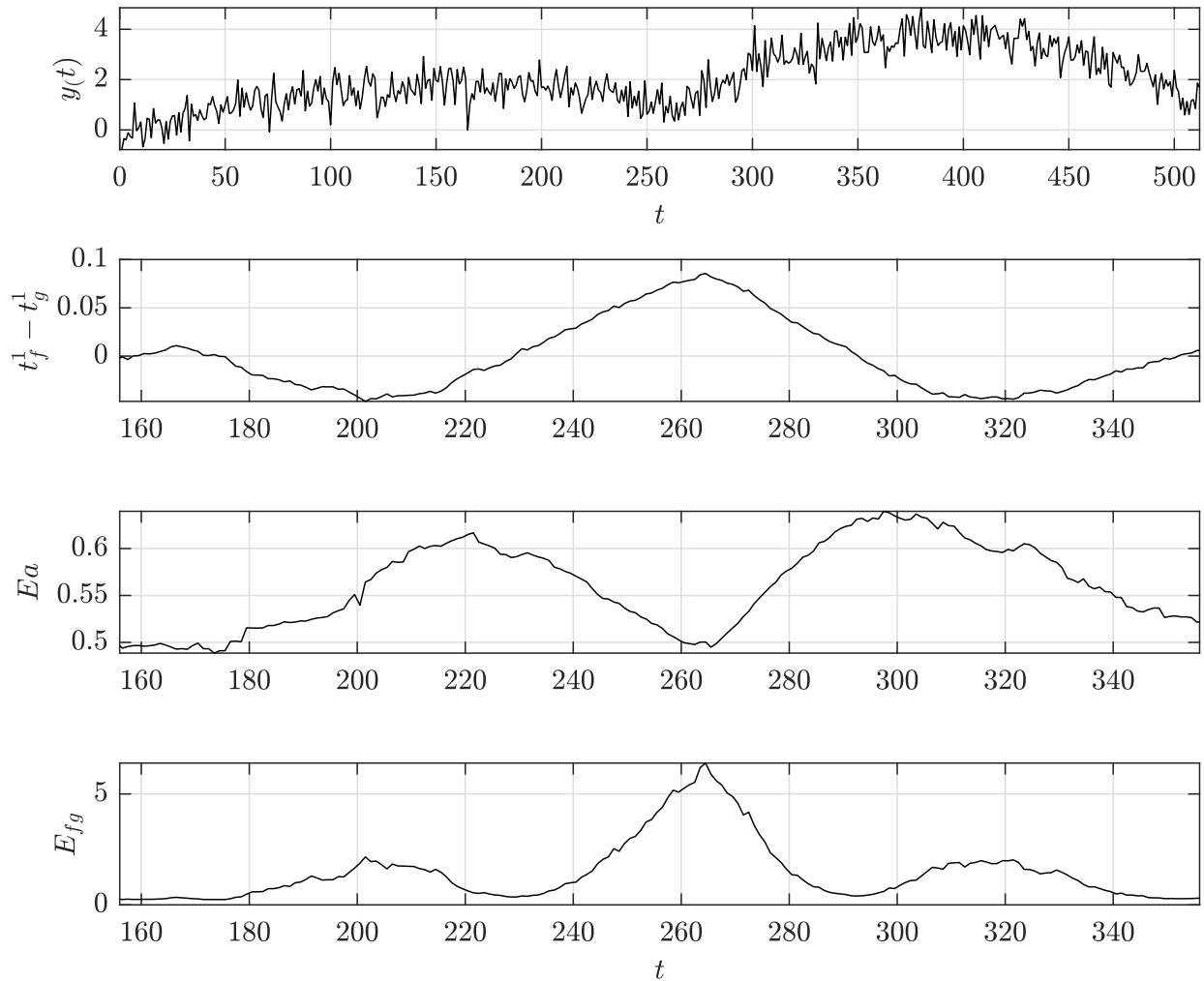


Fig. 5.16: Detecting a possible D^1 discontinuity with enforced D^0 and D^2 continuities with support length of $ls = 100$ to the left and to the right. The dataset tested here is a synthetic one with a first order discontinuity at $t = 256$ and added white Gaussian noise with $\sigma = 0.5$. The observation itself and the second derivative of the artificial signal are continuous.

As an other example to show the impact of *support length* ls variations Figure 5.16 shows the detection of the D^1 discontinuity in the D^2D^0 dataset with higher added white Gaussian noise of $\sigma = 0.5$. Furthermore, the possible application of the *generalized* method is further validated, since the *Taylor differences* $t_f^1 - t_g^1$, the *extrapolation error* E_a , as well as the *interpolation error* E_{fg} all show characteristic patterns. The detection and evaluation of these characteristic patterns is then subject of a following task. Note, that the noisier the analysed dataset gets, the higher the support length ls has to be. As a result, the detection of multiple discontinuities gets restricted, as with higher support lengths characteristic patterns tend to overlap.

5.5.3 Noise Impact - Monte Carlo Simulation

Now, after the impact of noise and the comparison between different parameter settings was shown in the previous subsections in a qualitative manner, this one is now dedicated to give a quantitative evaluation of the impact of noise on the two variants of the detection method with the created synthetic datasets corrupted by Gaussian noise of different magnitude. This task is performed by *Monte Carlo simulation* with $n = 1000$ iterations for each testing routine. The parameter which was mainly adapted to the tested dataset was the support length l_s , whereas no variations have been performed within tests of the same dataset. As additional tunable parameters, those defining the sensitivity of the peak detection algorithm were constant over all test runs. Furthermore, in order to have a reference value for each tested dataset, the detection procedure was also executed without any constraints, $c = 0$. The results of the performed Monte Carlo simulations are presented in Table 5.1.

Dataset	Noise level σ	Constraints	Support Length l_s	Mean Error μ_{detect}	σ_{detect}	
D^2D^0	0.010	[1,0,1]	[30 ,30]	0.0490	0.2161	
	0.020	[1,0,1]	[30 ,30]	0.1529	0.5029	
	0.040	[1,0,1]	[30 ,30]	0.2049	3.2755	
	0.010	[0,1]	[30 ,30]	0.0523	0.2798	
	0.020	[0,1]	[30 ,30]	0.1038	0.5940	
	0.040	[0,1]	[30 ,30]	0.0759	1.0578	
	0.040	[0,0]	[30 ,30]	0.1035	0.8863	
	$D^3D^1D^0$	0.005	[1,0,1,1]	[100 ,100]	0.5316	1.4898
		0.010	[1,0,1,1]	[100 ,100]	0.6590	3.0690
0.020		[1,0,1,1]	[100 ,100]	0.6009	6.3281	
0.005		[0,1,1]	[100 ,100]	0.5825	1.1187	
0.010		[0,1,1]	[100 ,100]	0.6216	2.2033	
0.020		[0,1,1]	[100 ,100]	0.6063	4.3499	
0.020		[0,0,0]	[100 ,100]	0.6966	15.8251	
$D^4D^2D^1D^0$		0.005	[1,1,0,1,1]	[40 ,40]	2.0213	0.8108
		0.010	[1,1,0,1,1]	[40 ,40]	2.0106	1.5514
	0.020	[1,1,0,1,1]	[40 ,40]	1.9123	8.5202	
	0.005	[0,1,1]	[40 ,40]	2.0186	0.5800	
	0.010	[0,1,1]	[40 ,40]	2.0620	1.1069	
	0.020	[0,1,1]	[40 ,40]	2.0454	2.0796	
	0.020	[0,0,0]	[40,40]	1.9653	6.8331	

Table 5.1: Overview of the Monte Carlo simulation results for the detection of change points with various parameter settings.

Within each dataset a clear trend of the impact of increasing noise levels can be seen in the standard deviation of the detection results σ_{detect} . Hereby, the accuracy of the detection method gets worse with increasing noise level σ .

Another observation that could be made over all three datasets is a decreasing detection accuracy after the implementation of higher order constraints to the approximation. This behaviour can be seen for all noise levels, see for example dataset D^2D^0 , with the constraint vector $c = [1, 0, 1]$ and $c = [0, 1]$ respectively.

What is more, there seems to be a clear impact of an increasing noise level, if higher order discontinuities are suspected in a dataset. Hereby, one could for example compare the results of the datasets D^2D^0 and $D^3D^1D^0$, where even with a higher support length l_s , the accuracy of the detection is still considerably worse for comparable noise levels.

Chapter 6

Benchmark Current Version with other Change Point Detection Methods

Now, to further evaluate a change point detection method, real world datasets are commonly used for testing in the literature [31]. Hence, a real world dataset collection was created and is going to be presented briefly in this chapter as well. Furthermore, the applied evaluation frame work, as well as the results of the benchmark testing are also described in the upcoming paragraphs.

6.1 Real World Datasets

The collection of time series datasets for the subsequent evaluation process was inspired by the benchmark frame work of van den Burg [31]. In this frame work various datasets are collected, which have interesting properties to evaluate the performance of change point detection methods. In general, these are discontinuities in the observation as well is in the first derivative. However, some of the datasets do not even show proper change points, but have some other interesting features to evaluate the effectiveness of the algorithm, like outliers or seasonality. Many of the datasets in the collection were already used in the literature of change point analysis like the *Nile* dataset ([40], [41], [42]) or the well log dataset [43]. Even though, also multivariate datasets were used in the frame work of van den Burg [31], the evaluation executed in this thesis, test the detection method only with univariate datasets. This is mainly due to practical reasons. Further testing with multivariate observations might be a possible task of a future work.

A detailed overview of the used dataset can be found in [31].

6.2 Evaluation Metrics

Now, before the results of the tested real world change point datasets can be evaluated, specific *evaluation* or *benchmark metrics* are going to be defined in this section. The choice of these very

metrics was mainly inspired by the evaluation frame work for change point detection problems by *van den Burg* [31].

Hereby, commonly used measures in change point analysis can be separated in *clustering metrics* and *classification metrics*. Whereas both describe the *change point problem* in a distinct manner.

Furthermore, as the already mentioned, framework from *van den Burg* [31] is used to evaluate the detected discontinuities, it is important to state that the authors made use of a so called *annotation tool* to get a measure for the *ground truth* of the analysed datasets. This annotation tool was used to let experienced data scientists from various fields estimate the given change points in the datasets manually. Subsequently, each annotator $k \in \{1, \dots, K\}$ defines the locations of suspected discontinuities given by the ordered set $\mathcal{T}_k = \{\tau_1, \dots, \tau_{n_k}\}$ with $\tau_i \in [1, T]$ for $i = 1, \dots, n_k$ and $\tau_i < \tau_j$ for $i < j$. Hence, \mathcal{T}_k can be seen as a partition, G_k , of the time series dataset with the interval $[1, T]$, into distinct sets A_j from τ_{j-1} to $\tau_j - 1$ for $j = 1, \dots, n_k + 1$.

6.2.1 Segmentation Covering Metric

Typical measures for change point evaluation methods are separated into clustering and classification metrics, as already stated above. Whereas, applications of the former, tend to illustrate the change point problem as a setting which inherently follows the goal to divide a time series into multiple different subsets with individual data models. Considering a simple mean-shift change point problem for example, the time series can be divided into two distinctive mean models. Some clustering metrics used in the literature are for instance the *variation of information (VI)*, the *adjusted Rand index* [44], the *Hausdorff distance* [45] and the *segmentation covering metric* [46]. Because the evaluation quality of change point methods, delivering many false positives, suffers by using the *Hausdorff metric* and the validation of methods with multiple change points is not suited for the application of the VI-metric, the *segmentation covering metric* is used in the following part of this thesis. (see *van den Burg* 2020 [31])

Consequently, the *Jaccard Index* J [47] (*intersection over union*) of the subsets $A, A' \subseteq [1, T]$ is defined as,¹

$$J(A, A') = \frac{|A \cap A'|}{|A \cup A'|}. \quad (6.1)$$

Then the *segmentation covering metric* [48] of the partition G by the partition G' can be computed with,

¹ In the following, the expression $|\cdot|$ denotes the cardinality of a set.

$$C(G', G) = \frac{1}{T} \sum_{A \in G} |A| \max_{A' \in G'} J(A, A'). \quad (6.2)$$

Finally, considering a collection of ground truth partitions $\{G_k\}_{k=1}^K$ (defined by K annotators), as well as the partition S returned by a change point detection algorithm, the average of the *covering metric* $C(S, G_k)$ can be derived as a measure to evaluate the performance of the used algorithm. Basically, the concept of the measure is to compare the subsets of a dataset due to the detected change points with the ground truth subsets. Hereby, the highest Jaccard index $J(A, A')$ for each subset is computed, where every subset in the partition S is compared to every subset of the ground truth partition G_k . This comparison then yields the highest Jaccard index $J(A, A')$ for each subset of the detected partition S , which can be seen as the score of the best match for each subset. Finally, for those detected scores the weighted average over the dataset length T of all subsets yields the segmentation covering measure $C(S, G_k)$ of the partitions S and G_k . Hereby, the respective length of each subset $|A|$ is used as weighting factor.

To further illustrate the covering metric C the following simple example should show how it is used.

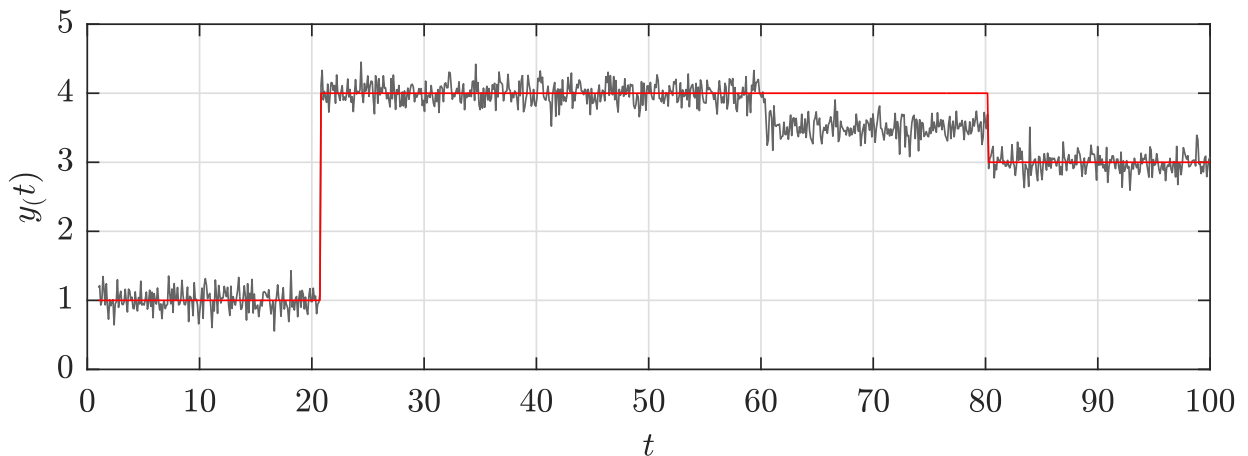


Fig. 6.1: Example of a dataset with three mean-shift change points and the segmentation due to a set of detected change points by a detecting method (red).

Here the dataset shown in Figure 6.1 has three mean-shift change points at the locations combined in the set \mathcal{T}_1 , resulting in the partitioning G_1 .

$$\begin{aligned} \mathcal{T}_1 &= \{20, 60, 80\} \\ G_1 &= \{[1 \dots 20], (20 \dots 60], (60 \dots 80], (80 \dots 100]\} \end{aligned}$$

Furthermore, the detected change points by a detection method lead to the partitioning S with,

$$S = \{[1...20], (20...80], (80...100]\}.$$

Now, the segmentation covering measure can be computed with,

$$\begin{aligned} C(S, G_1) &= \frac{1}{100} \left(|[1...20]| \frac{|[1...20] \cup [1...20]|}{|[1...20] \cap [1...20]|} + |(20...60)] \frac{|(20...60] \cup (20...80]|}{|(20...60] \cap (20...80]|} \right. \\ &\quad \left. + |(80...100)] \frac{|(80...100] \cup (80...100]|}{|(80...100] \cap (80...100]|} \right) \\ &= \frac{1}{100} \left(20 \frac{20}{20} + 40 \frac{40}{60} + 20 \frac{20}{20} \right) = 0.667 \end{aligned} \quad (6.3)$$

This result can be interpreted as to proportion of the correctly detected partitioning S , considering the ground truth partitioning G_1 . For further test, as K annotators defined the ground truth locations of the change points of the different datasets, the average of all covering metrics is used as the final evaluation metric.

6.2.2 F1 - Score

Now, considering another approach for change point detection method evaluation, the setting can be formulated as a *classification problem*. Hereby, the classes "change point" and "non-change point" are distinguished.[49], [50].

However, it is important to mention, that typical classification metrics like the *accuracy score* are not well suited for the application in this context, because the number of discontinuities in the dataset is considerably small compared to all samples. As a result, the *accuracy score* would overrate certain detection methods, where datasets with higher sample size are analysed.

The F_β -score is used as a measure for accuracy effectiveness of a detection algorithm. Hereby, the so called *precision* P as well as the *recall* R are involved. Where the precision P is derived as the ratio between properly detected discontinuities to the overall number of detected discontinuities. The recall R , on the other hand, yields the ratio of properly detected discontinuities to number of given discontinuities in the time series. Both are then combined in the F_β -score, where the precision P is weighted in β times.

Hence, the F_β -score is calculated with,

$$F_\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R}. \quad (6.4)$$

The more specific, F_1 -score, with $\beta = 1$, is then derived as the harmonic mean of precision P and recall R with,

$$F_1 = 2 \frac{PR}{P+R}. \quad (6.5)$$

In this context, a *margin of error* M is introduced to account for minor deviations between the suspected change points and the annotated ones. [50], [17], [31] Let X be as set of detected discontinuities returned by a detection method and $\mathcal{T}^* = \cup_k \mathcal{T}_k$ denotes the combined annotations of all manual annotators. Then for the ground truth discontinuities \mathcal{T} the true positives $TP(\mathcal{T}, X)$ are all $\tau \in \mathcal{T}$ for which $x \in X$ exists such that $|\tau - x| \leq M$, with $M \geq 0$.

Subsequently, the precision P is derived by,

$$P = \frac{|TP(\mathcal{T}^*, X, M)|}{|X|}, \quad (6.6)$$

and the recall R can be computed with,

$$R = \frac{1}{K} \sum_{k=1}^K \frac{|TP(\mathcal{T}_k, X, M)|}{|\mathcal{T}_k|}. \quad (6.7)$$

Consequently, only those detected sample points are evaluated as false positive ones, which do not correspond to any annotated change point.

So, by making use of the segmentation metric $C(S, G_k)$ as well as of the F_1 -Score, the benchmark frame work takes the clustering view and the classification view into account.

6.3 Benchmark Testing

As described in the previous subchapter, the evaluation of the method introduced in this thesis was executed with the *segmentation covering metric* and the F_1 -Score. In fact, two separate benchmark testing routines were performed on the entire dataset collection, once for the *constrained coupled polynomial* approach before and once after the *generalization*. In fact, if the dataset was analysed for discontinuities in the observation directly (zeroth order), no constraints were applied on the approximation for the first testing routine and if slope discontinuities were suspected (first order) the zeroth order polynomial coefficients,

$$\alpha_0 = \beta_0 \quad (6.8)$$

were constrained. All tests executed under these conditions are referred to as *CCP* in the following.

In contrast to the mentioned configuration (CCP), an other set of tests with the *change point method validation dataset collection* was performed by making use of the *generalization* described in this thesis. Namely, additional constraints were applied to the approximation for the two different test cases. On the one hand, potential mean-shift change points (zeroth order) were detected with enforced first order continuity,

$$\alpha_1 = \beta_1, \text{ but } \alpha_0 \neq \beta_0. \quad (6.9)$$

Whereas on the other hand, the detection of slope change points was performed with enforced second and zeroth order continuity,

$$\alpha_2 = \beta_2, \quad (6.10)$$

$$\alpha_0 = \beta_0, \quad (6.11)$$

by allowing first order discontinuities with,

$$\alpha_1 \neq \beta_1. \quad (6.12)$$

Here the results are labelled with *GCCP*, for *generalized constrained coupled polynomials*.

The results of this evaluation for both variants are shown in Table 6.1. Hereby, the average of each metric for the entire dataset collection is compared to the other methods evaluated in [31]. The results shown in the table are compared to the *best* possible results for each method, which means that the hyper parameters are optimized for the respective dataset. Another evaluation variant would be the comparison with *default* hyper parameters. See [31] for the difference between both variants. The parameters for the newly tested algorithms have been optimized manually. Hence, another task for future work could be to optimize the parameters automatically, which was performed in [31] by a *grid search*.

Detailed results, as well as the used parameters, for each dataset and detection method can be found in the appendix.

In general, the performance of the method can be seen as quite comparable, as displayed in Table 6.1. Especially, the *segmentation covering metric* for the CCP configuration delivers good results on the dataset collection compared to the other algorithms. Considering the F_1 -score, the CCP results are also clearly above average.

Slightly under the average performance of the detection methods are the metrics for the tests executed with the GCCP configuration of the algorithm. This finding is strongly in accordance with

the observation of performed Monte Carlo simulations in Chapter 5, where a higher impact of noise to the detections with higher constraints was present.

Detection Method	Covering Metric	F_1 Score	Description	Reference
AMOC	0.746	0.799	At Most One Change	Hinkley (1970) [51]
BINSEG	0.780	0.856	Binary Segmentation	Scott and Knott (1974) [52]
BOCPD	0.789	0.880	Bayesian Online Change Point Detection	Adams and MacKay (2007) [32]
BOCPDMS	0.744	0.620	BOCPD with Model Selection	Knoblauch and Daoulas (2018) [53]
CPNP	0.552	0.666	Nonparametric Change Point Detection	Haynes et al. (2017) [54]
ECP	0.720	0.797	Energy Change Point	Matteson and James (2014) [55]
KCPA	0.626	0.683	Kernel Change-Point Analysis	Harchaoui et al. (2009) [56]
PELT	0.725	0.787	Pruned Exact Linear Time	Killick et al. (2012) [50]
PROPHET	0.576	0.534	Prophet	Taylor and Letham (2018) [57]
RBOCPDMS	0.629	0.447	Robust BOCPDMS	Knoblauch et. al. (2018) [58]
RFPOP	0.414	0.531	Robust Functional Pruning Optimal Partitioning	Fearnhead and Rigaiil (2019) [59]
SEGNEIGH	0.784	0.855	Segment Neighbourhoods	Auger and Lawrence (1989) [60]
WBS	0.428	0.533	Wild Binary Segmentation	Fryzlewicz (2014) [61]
ZERO	0.579	0.662	No Change Points	
AVG	0.652	0.698	Average of the evaluation metrics	
CCP	0.763	0.788	Constrained Coupled Polynomials	Ninevski and O’Leary (2018) [1]
GCCP	0.690	0.711	Generalized Constrained Coupled Polynomials	

Table 6.1: Comparison of the evaluation results for several well-known change point detection algorithms by van den Burg and Williams [31] including the evaluation of the constrained coupled polynomial discontinuity detection approach [1]

After the comparison of all datasets of the collection at once, the datasets were separated into those with suspected *slope* change points and mean-shift change points. The idea behind this split was the fact that the CCP, as well as the GCCP approach might perform better on the detection of derivative constraints. Hence, a difference in favour of the slope change point problems could be seen in the evaluation.

In fact, a clear improvement of both benchmark metrics, except for the CCP covering metric, can be seen when only the slope change point problems are tested, as Table 6.2 suggests. Hereby, it is also notable that both configurations, CCP and GCCP, show a performance increase, whereas both are clear above average. Moreover, all other change point detection algorithms delivered significantly worse results compared to the average detection results in Table 6.1 and consequently to the mean-shift results in Table 6.3.

Detection Method	Covering Metric	F_1 Score	Description	Reference
AMOC	0.697	0.797	At Most One Change	Hinkley (1970) [51]
BINSEG	0.716	0.822	Binary Segmentation	Scott and Knott (1974) [52]
BOCPD	0.727	0.875	Bayesian Online Change Point Detection	Adams and MacKay (2007) [32]
BOCPDMS	0.666	0.602	BOCPD with Model Selection	Knoblauch and Daoulas (2018) [53]
CPNP	0.514	0.656	Nonparametric Change Point Detection	Haynes et al. (2017)) [54]
ECP	0.688	0.797	Energy Change Point	Matteson and James (2014) [55]
KCPA	0.546	0.638	Kernel Change-Point Analysis	Harchaoui et al. (2009) [56]
PELT	0.680	0.784	Pruned Exact Linear Time	Killick et al. (2012) [50]
PROPHET	0.517	0.509	Prophet	Taylor and Letham (2018) [57]
RBOCPDMS	0.673	0.640	Robust BOCPDMS	Knoblauch et. al. (2018) [58]
RFPOP	0.378	0.504	Robust Functional Pruning Optimal Partitioning	Fearnhead and Rigaiil (2019) [59]
SEGNEIGH	0.725	0.842	Segment Neighbourhoods	Auger and Lawrence (1989) [60]
WBS	0.329	0.420	Wild Binary Segmentation	Fryzlewicz (2014) [61]
ZERO	0.590	0.688	No Change Points	
AVG	0.622	0.683	Average of the evaluation metrics	
CCP	0.743	0.813	Constrained Coupled Polynomials	Ninevski and O’Leary (2018) [1]
GCCP	0.661	0.748	Generalized Constrained Coupled Polynomials	

Table 6.2: First derivative change points - Comparison of the evaluation results for several well-known change point detection algorithms by van den Burg and Williams [31] including the evaluation of the constrained coupled polynomial discontinuity detection approach [1]

Finally, just like for the slope change point datasets, a distinct comparison was also carried out for the mean-shift detection problems. Hereby, a similar trend could be observed in terms of the configurations CCP and GCCP, where the former performs significantly better.

Detection Method	Covering Metric	F_1 Score	Description	Reference
AMOC	0.788	0.801	At Most One Change	Hinkley (1970) [51]
BINSEG	0.706	0.885	Binary Segmentation	Scott and Knott (1974) [52]
BOCPD	0.636	0.883	Bayesian Online Change Point Detection	Adams and MacKay (2007) [32]
BOCPDMS	0.633	0.637	BOCPD with Model Selection	Knoblauch and Daoulas (2018) [53]
CPNP	0.584	0.675	Nonparametric Change Point Detection	Haynes et al. (2017)) [54]
ECP	0.731	0.782	Energy Change Point	Matteson and James (2014) [55]
KCPA	0.684	0.716	Kernel Change-Point Analysis	Harchaoui et al. (2009) [56]
PELT	0.764	0.790	Pruned Exact Linear Time	Killick et al. (2012) [50]
PROPHET	0.540	0.562	Prophet	Taylor and Letham (2018) [57]
RBOCPDMS	0.629	0.477	Robust BOCPDMS	Knoblauch et. al. (2018) [58]
RFPOP	0.392	0.554	Robust Functional Pruning Optimal Partitioning	Fearnhead and Rigaiil (2019) [59]
SEGNEIGH	0.676	0.866	Segment Neighbourhoods	Auger and Lawrence (1989) [60]
WBS	0.513	0.628	Wild Binary Segmentation	Fryzlewicz (2014) [61]
ZERO	0.558	0.632	No Change Points	
AVG	0.694	0.709	Average of the evaluation metrics	
CCP	0.790	0.762	Constrained Coupled Polynomials	Ninevski and O’Leary (2018) [1]
GCCP	0.718	0.674	Generalized Constrained Coupled Polynomials	

Table 6.3: Mean-Shift change points - Comparison of the evaluation results for several well-known change point detection algorithms by van den Burg and Williams [31] including the evaluation of the constrained coupled polynomial discontinuity detection approach [1]

Chapter 7

Conclusion

This new approach to the detection of discontinuities in observational data is the most generic of all the methods considered in this thesis, since it does not require application specific adaption. It is based on a formal mathematical definition of a discontinuity, which is a generalization of a C^n discontinuity. In the new definition, any one of the n derivative can be tested for discontinuity, while enforcing continuity for the others. The extensive benchmarking of the new method in comparison to others, for a wide range of test cases, reveals that the new approach is among the best. This combined with the numerical efficient computation makes it very attractive. There are, however, test cases where the application specific methods perform better. Previous literature and test data sets only consider C^0 and C^1 type discontinuities. Whereas, the new approach, not only performs well for these type of discontinuities, but also functions for higher order derivative discontinuities.

Future Tasks

Finally, a brief overview of possible future avenues for investigation is given:

1. More relevant and reliable and fundamental evaluation metric
2. Application of the evaluation environment to higher order discontinuity datasets
3. Grid search optimization in the course of the evaluation environment
4. Comparison of computational efficiency of different detection methods
5. Variable margin of error for the *Segmentation Covering metric*
6. Testing of multivariate datasets
7. Simulation of on-line change point detection
8. Directly detecting of the *triple peak pattern*

List of Figures

2.1	Exactly determined polynomial interpolation	6
2.2	Under-determined polynomial interpolation	7
2.3	Overdetermined polynomial approximation	8
3.1	Positive trend in time series	18
3.2	Periodicity in time series	19
3.3	Outliers in time series	19
3.4	Change Point in time series	20
3.5	Example for an abrupt change in the mean	21
3.6	Example for an abrupt variance change	22
3.7	Example for an abrupt change in the slope of an observation	23
3.8	Example for an abrupt change in the distribution type of an observation	24
4.1	Removable discontinuity	36
4.2	Jump discontinuity	36
4.3	Infinite discontinuity	37
4.4	Constrained polynomial discontinuity detection	39
4.5	Approximation error	43
4.6	Extrapolation error	44
4.7	Combined error	45
5.1	Synthetic dataset D^2D^0	52
5.2	Synthetic dataset $D^3D^1D^0$	53
5.3	Synthetic dataset $D^4D^3D^2D^0$	55
5.4	Detection of a possible D1 discontinuity	56
5.5	Evaluation for D1 discontinuity	57
5.6	Detection of a possible C^1 discontinuity	58
5.7	Evaluation for C^1 discontinuity	59
5.8	Detection of a possible D^2 discontinuity	61
5.9	Detection of a possible C^2 discontinuity	62

List of Figures	85
5.10 Detection of a possible D^2 discontinuity	63
5.11 Detection of a possible D1 discontinuity	65
5.12 Detection of a possible C1 discontinuity	66
5.13 Detection of a possible D1 discontinuity	67
5.14 Detection of a possible C2 discontinuity	68
5.15 Detection of a possible D2 discontinuity	70
5.16 Detection of a possible D1 discontinuity	71
6.1 Segmentation Covering Metric	77

List of Tables

3.1	Articles in the change point detection literature	26
5.1	Monte Carlo Simulation results for noise impact testing	73
6.1	Average evaluation results	81
6.2	Average evaluation results - Slope switch	82
6.3	Average evaluation results - Mean-shift	82
A.1	Detailed evaluation results for all mean-shift datasets.	92
A.2	Detailed evaluation results for all slope-switch datasets.	93

References

- [1] Dimitar Ninevski and Paul O’Leary. “Detection of Derivative Discontinuities in Observational Data”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2020. ISBN: 9783030445836. DOI: 10.1007/978-3-030-44584-3_29. arXiv: 1911.12724.
- [2] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* (2019). ISSN: 10902716. DOI: 10.1016/j.jcp.2018.10.045.
- [3] Harshit Saxena, Omar Aponte, and Katie T. McConky. “A hybrid machine learning model for forecasting a billing period’s peak electric load days”. In: *International Journal of Forecasting* (2019). ISSN: 01692070. DOI: 10.1016/j.ijforecast.2019.03.025.
- [4] Astrid Dempfle and Winfried Stute. “Nonparametric estimation of a discontinuity in regression”. In: *Statistica Neerlandica*. 2002. DOI: 10.1111/1467-9574.00196.
- [5] Yazhen Wang. “Jump and sharp cusp detection by wavelets”. In: *Biometrika* (1995). ISSN: 00063444. DOI: 10.1093/biomet/82.2.385.
- [6] M. Roggero. “Discontinuity Detection and Removal from Data Time Series”. In: *International Association of Geodesy Symposia*. 2012. ISBN: 9783642220777. DOI: 10.1007/978-3-642-22078-4_20.
- [7] Larry S. Davis. “A survey of edge detection techniques”. In: *Computer Graphics and Image Processing* (1975). ISSN: 0146664X. DOI: 10.1016/0146-664x(75)90012-x.
- [8] D. Marr. “Vision: a computational investigation into the human representation and processing of visual information.” In: *Vision: a computational investigation into the human representation and processing of visual information*. (1982). ISSN: 00222496. DOI: 10.1016/0022-2496(83)90030-5.
- [9] Annette Burden, Richard L. Burden, and J Douglas Faires. *Numerical Analysis, 10th ed.* 2016. ISBN: 1305253663.
- [10] K. Weierstrass. “Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen.” In: *Sitzungsbericht der Königlich Preußischen Akademie der Wissenschaften zu Berlin II* (1885).
- [11] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering*. 2019. DOI: 10.1017/9781108380690.
- [12] Carl Eckart and Gale Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* (1936). ISSN: 00333123. DOI: 10.1007/BF02288367.
- [13] E. S. Page. “Continuous Inspection Schemes”. In: *Biometrika* (1954). ISSN: 00063444. DOI: 10.2307/2333009.

- [14] Tze San Lee. “Change-point problems: Bibliography and review”. In: *Journal of Statistical Theory and Practice* (2010). ISSN: 15598616. DOI: 10.1080/15598608.2010.10412010.
- [15] Robert C. Wilson, Matthew R. Nassar, and Joshua I. Gold. *Bayesian online learning of the hazard rate in change-point problems*. 2010. DOI: 10.1162/NECO_a_00007.
- [16] L.Y. Vostrikova. “Detecting disorder in multidimensional random process.” In: *In Soviet Mathematics Doklady* (1981), pp. 55–59.
- [17] Charles Truong, Laurent Oudre, and Nicolas Vayatis. *Selective review of offline change point detection methods*. 2020. DOI: 10.1016/j.sigpro.2019.107299. arXiv: 1801.00718.
- [18] Peter Hall and D. M. Titterton. “Edge-preserving and peak-preserving smoothing”. In: *Technometrics* (1992). ISSN: 15372723. DOI: 10.1080/00401706.1992.10484954.
- [19] C. T. Jose and B. Ismail. “Estimation of jump points in nonparametric regression through residual analysis”. In: *Communications in Statistics - Theory and Methods* (1997). ISSN: 03610926. DOI: 10.1080/03610929708832067.
- [20] Hans-Georg Muller. “Change-Points in Nonparametric Regression Analysis”. In: *The Annals of Statistics* (1992). ISSN: 0090-5364. DOI: 10.1214/aos/1176348654.
- [21] J. S. Wu and C. K. Chu. “Modification for boundary effects and jump points in nonparametric regression”. In: *Journal of Nonparametric Statistics* (1993). ISSN: 10290311. DOI: 10.1080/10485259308832563.
- [22] R. L. Eubank and P. L. Speckman. “Nonparametric estimation of functions with jump discontinuities”. In: 1994. DOI: 10.1214/lnms/1215463119.
- [23] John Alan McDonald and Art B. Owen. “Smoothing with split linear fits”. In: *Technometrics* (1986). ISSN: 15372723. DOI: 10.1080/00401706.1986.10488127.
- [24] Peihua Qiu and Brian Yandell. “A Local Polynomial Jump-Detection Algorithm in Nonparametric Regression”. In: *Technometrics* (1998). ISSN: 00401706. DOI: 10.2307/1270648.
- [25] V. G. Spokoiny. “Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice”. In: *Annals of Statistics* (1998). ISSN: 00905364. DOI: 10.1214/aos/1024691246.
- [26] Marc Raimondo. “Minimax estimation of sharp change points”. In: *Annals of Statistics* (1998). ISSN: 00905364. DOI: 10.1214/aos/1024691247.
- [27] H. Chernoff and S. Zacks. “Estimating the Current Mean of a Normal Distribution which is Subjected to Changes in Time”. In: *The Annals of Mathematical Statistics* (1964). ISSN: 0003-4851. DOI: 10.1214/aoms/1177700517.
- [28] Lyle David Broemeling. “Bayesian Inferences about a Changing Sequence of Random Variables”. In: *Communications in Statistics - Simulation and Computation* (1974). ISSN: 0361-0918. DOI: 10.1080/03610917408548374.

- [29] A. F.M. Smith. “A Bayesian approach to inference about a change-point in a sequence of random variables”. In: *Biometrika* (1975). ISSN: 00063444. DOI: 10.1093/biomet/62.2.407.
- [30] Daniel Barry and J. A. Hartigan. “Product Partition Models for Change Point Problems”. In: *The Annals of Statistics* (1992). ISSN: 0090-5364. DOI: 10.1214/aos/1176348521.
- [31] Gerrit J. J. van den Burg and Christopher K. I. Williams. “An Evaluation of Change Point Detection Algorithms”. In: (2020). arXiv: 2003.06222. URL: <https://arxiv.org/abs/2003.06222>.
- [32] Ryan Prescott (University of Cambridge) Adams and David J. C. (University of Cambridge) MacKay. “Bayesian Online Changepoint Detection”. In: (2007). arXiv: 0710.3742. URL: <https://arxiv.org/abs/0710.3742>.
- [33] Paul Fearnhead and Zhen Liu. “On-line inference for multiple changepoint problems”. In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* (2007). ISSN: 13697412. DOI: 10.1111/j.1467-9868.2007.00601.x.
- [34] Abraham Wald. *Sequential Analysis*. 1st. John Wiley and Sons, 1947. URL: <http://books.google.com/books?id=oVYDHHzZtdIC%7B%5C%7Dprintsec=frontcover%7B%5C%7Ddq=editions:oVYDHHzZtdIC%7B%5C%7Dhl=en%7B%5C%7Dei=P5zFTYbWNdK1twelSfCYBA%7B%5C%7Dsa=X%7B%5C%7Doi=book%7B%5C%7Dresult%7B%5C%7Dct=book-thumbnail%7B%5C%7Dresnum=1%7B%5C%7Dved=0CCwQ6wEwAA%7B%5C%7Dv=onepage%7B%5C%7Dq%7B%5C%7Df=false>.
- [35] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Prentice Hall, 1998, p. 577. ISBN: 013504135X.
- [36] Sailes K. Sengupta and Steven M. Kay. “Fundamentals of Statistical Signal Processing: Estimation Theory”. In: *Technometrics* (1995). ISSN: 00401706. DOI: 10.2307/1269750.
- [37] M Basseville and I V Nikiforov. “Detection of Abrupt Changes : Mich ‘ele Basseville”. In: *Change* (1993). ISSN: 09670661. DOI: 10.1016/0967-0661(94)90196-1.
- [38] James M. Lucas and Ronald B. Crosier. “Fast initial response for cusum quality-control schemes: Give your cusum a head start”. In: *Technometrics* (1982). ISSN: 15372723. DOI: 10.1080/00401706.1982.10487759.
- [39] J. O. Westgard et al. “Combined Shewhart cusum control chart for improved quality control in clinical chemistry”. In: *Clinical Chemistry* (1977). ISSN: 00099147. DOI: 10.1093/clinchem/23.10.1881.
- [40] George W. Cobb. “The problem of the Nile: Conditional solution to a changepoint problem”. In: *Biometrika* (1978). ISSN: 00063444. DOI: 10.1093/biomet/65.2.243.
- [41] Nathan S. Balke. “Detecting Level Shifts In Time Series”. In: *Journal of Business and Economic Statistics* (1993). ISSN: 15372707. DOI: 10.1080/07350015.1993.10509934.

- [42] James Durbin, Siem Jan Koopman, and Eleanor Bash. *Time series analysis by state space methods: J. Durbin and S.J. Koopman, Oxford Statistical Series 24, 2001, Oxford University Press*. 2001. ISBN: 0198523548. arXiv: arXiv:1011.1669v3.
- [43] Kurt S. Riedel, Joseph J. K. O. Ruanaidh, and William J. Fitzgerald. “Numerical Bayesian Methods Applied to Signal Processing”. In: *Technometrics* (1997). ISSN: 00401706. DOI: 10.2307/1270786.
- [44] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* (1985). ISSN: 01764268. DOI: 10.1007/BF01908075.
- [45] G. T. Whyburn and F. Hausdorff. “Mengenlehre.” In: *The American Mathematical Monthly* (1928). ISSN: 00029890. DOI: 10.2307/2300010.
- [46] Mark Everingham et al. “The pascal visual object classes (VOC) challenge”. In: *International Journal of Computer Vision* (2010). ISSN: 09205691. DOI: 10.1007/s11263-009-0275-4.
- [47] Paul Jaccard. “Lois de distribution florale dans la zone alpine”. In: *Bulletin de la Société vaudoise des sciences naturelles* (1902). ISSN: ;null;.
- [48] Pablo Arbeláez et al. “Contour detection and hierarchical image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011). ISSN: 01628828. DOI: 10.1109/TPAMI.2010.161.
- [49] Samaneh Aminikhanghahi and Diane J. Cook. “A survey of methods for time series change point detection”. In: *Knowledge and Information Systems* (2017). ISSN: 02193116. DOI: 10.1007/s10115-016-0987-z.
- [50] R. Killick, P. Fearnhead, and I. A. Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* (2012). ISSN: 01621459. DOI: 10.1080/01621459.2012.737745. arXiv: 1101.1438.
- [51] David V. Hinkley. “Inference about the change-point in a sequence of random variables”. In: *Biometrika* (1970). ISSN: 00063444. DOI: 10.1093/biomet/57.1.1.
- [52] A. J. Scott and M. Knott. “A Cluster Analysis Method for Grouping Means in the Analysis of Variance”. In: *Biometrics* (1974). ISSN: 0006341X. DOI: 10.2307/2529204.
- [53] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. “Doubly robust Bayesian inference for non-stationary streaming data with β -divergences”. In: *Advances in Neural Information Processing Systems*. 2018. arXiv: 1806.02261.
- [54] Kaylea Haynes, Paul Fearnhead, and Idris A. Eckley. “A computationally efficient non-parametric approach for changepoint detection”. In: *Statistics and Computing* (2017). ISSN: 15731375. DOI: 10.1007/s11222-016-9687-5. arXiv: 1602.01254.
- [55] David S. Matteson and Nicholas A. James. “A nonparametric approach for multiple change point analysis of multivariate data”. In: *Journal of the American Statistical Association* (2014). ISSN: 1537274X. DOI: 10.1080/01621459.2013.849605. arXiv: 1306.4933.

- [56] Zaïd Harchaoui, Francis Bach, and Éric Moulines. “Kernel change-point analysis”. In: *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*. 2009. ISBN: 9781605609492.
- [57] Sean J. Taylor and Benjamin Letham. “Forecasting at Scale”. In: *American Statistician* (2018). ISSN: 15372731. DOI: 10.1080/00031305.2017.1380080.
- [58] Jeremias Knoblauch and Theodoras Damoulas. “Spatio-temporal Bayesian on-line change-point detection with model selection”. In: *35th International Conference on Machine Learning, ICML 2018*. 2018. ISBN: 9781510867963. arXiv: 1805.05383.
- [59] Paul Fearnhead and Guillem Rigail. “Changepoint Detection in the Presence of Outliers”. In: *Journal of the American Statistical Association* (2019). ISSN: 1537274X. DOI: 10.1080/01621459.2017.1385466. arXiv: 1609.07363.
- [60] Ivan E. Auger and Charles E. Lawrence. “Algorithms for the optimal identification of segment neighborhoods”. In: *Bulletin of Mathematical Biology* (1989). ISSN: 15229602. DOI: 10.1007/BF02458835.
- [61] Piotr Fryzlewicz. “Wild binary segmentation for multiple change-point detection”. In: *Annals of Statistics* (2014). ISSN: 00905364. DOI: 10.1214/14-AOS1245. arXiv: 1411.0858.

Appendix A

Detailed Results of the Benchmark Tests

Dataset	CCP		GCCP	
	Cover	F1	Cover	F1
bank	0.628	0.667	0.398	0.206
bitcoin	0.768	0.591	0.761	0.5112
brent_spot	0.718	0.747	0.718	0.747
homeruns	0.539	0.766	0.476	0.627
measles	0.952	0.947	0.650	0.643
nile	0.892	1.000	0.649	0.368
quality_control_1	0.992	0.667	0.978	0.500
quality_control_2	0.907	0.571	0.920	0.800
quality_control_3	0.914	0.444	0.500	0.667
quality_control_4	0.760	0.936	0.672	0.781
quality_control_5	1.000	1.000	1.000	1.000
rail_lines	0.779	0.889	0.811	0.857
ratner_stock	0.816	0.713	0.838	0.500
scanline_126007	0.428	0.741	0.458	0.779
scanline_42049	0.871	0.951	0.836	0.869
seatbelts	0.778	0.736	0.527	0.621
unemployment_nl	0.605	0.904	0.655	0.812
usd_isk	0.748	0.797	0.877	0.876
well_log	0.920	0.409	0.926	0.643
Average	0.790	0.762	0.7180	0.674

Table A.1: Detailed evaluation results for all mean-shift datasets.

Dataset	CCP		GCCP	
	Cover	F1	Cover	F1
businv	0.816	0.727	0.616	0.650
centralia	0.498	0.800	0.523	0.720
children_per_woman	0.789	0.769	0.792	0.555
co2_canada	0.635	0.719	0.549	0.535
construction	0.820	0.889	0.751	0.727
debt_ireland	0.792	0.667	0.565	0.929
gdp_argentina	0.646	1.000	0.580	0.818
gdp_croatia	0.750	0.800	0.752	1.000
gdp_iran	0.832	0.974	0.616	0.630
gdp_japan	0.666	0.800	0.801	0.889
global_co2	0.744	0.929	0.720	0.929
iceland_tourism	0.827	0.571	0.828	0.391
jfk_passengers	0.755	0.571	0.378	0.523
lga_passengers	0.618	0.644	0.477	0.693
ozone	0.850	0.966	0.850	0.966
robocalls	0.804	1.000	0.566	0.500
shanghai_license	0.905	0.776	0.922	0.966
uk_coal_employ	0.635	0.896	0.531	0.799
us_population	0.730	0.947	0.735	1.000
Average	0.743	0.813	0.7180	0.661

Table A.2: Detailed evaluation results for all slope-switch datasets.

Appendix B

Mean-shift - Discontinuity Detection Plots

In the following the evaluation results for the CCP configuration presented in Table 6.1 are displayed.

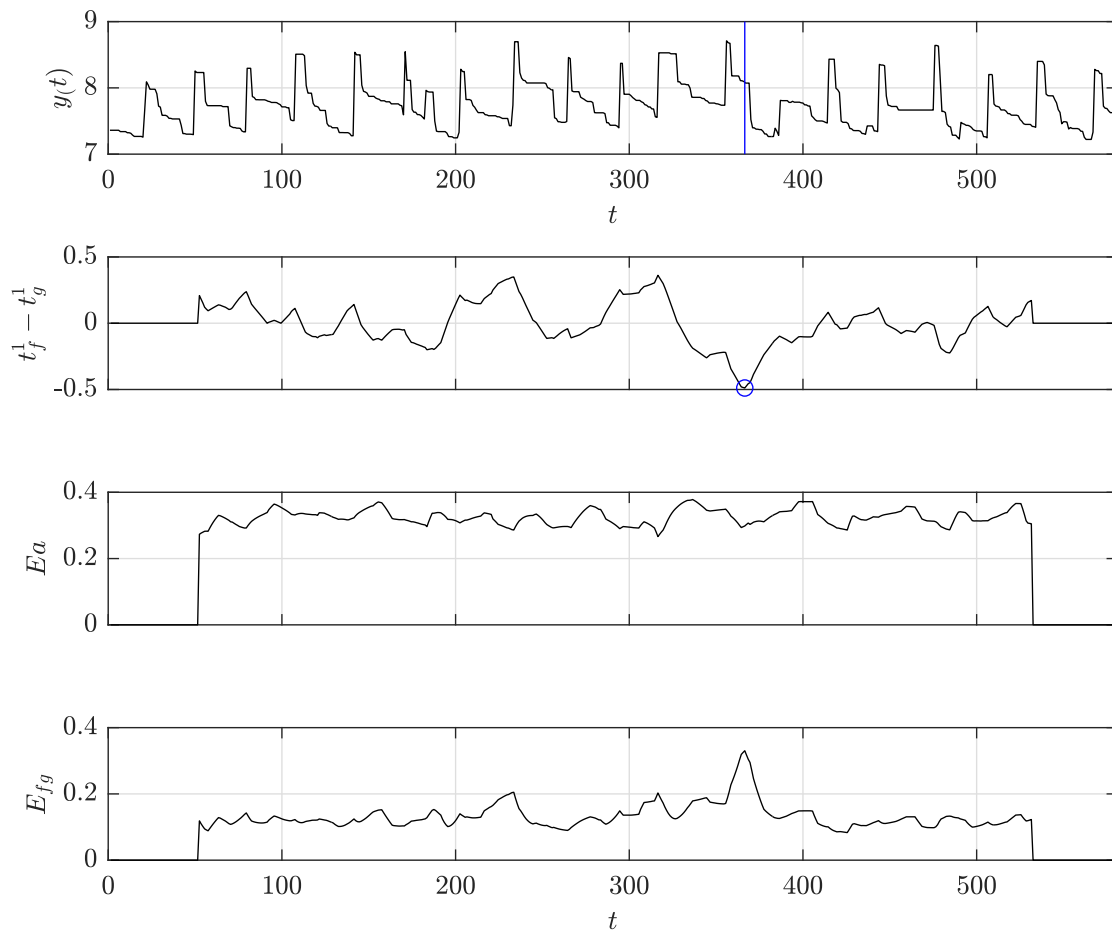


Fig. B.1: Detection of a possible mean-shift change point in the dataset "bank", by considering a present periodicity, with support length $l_s = [50, 50]$ and constraint vector of $c = [0]$.

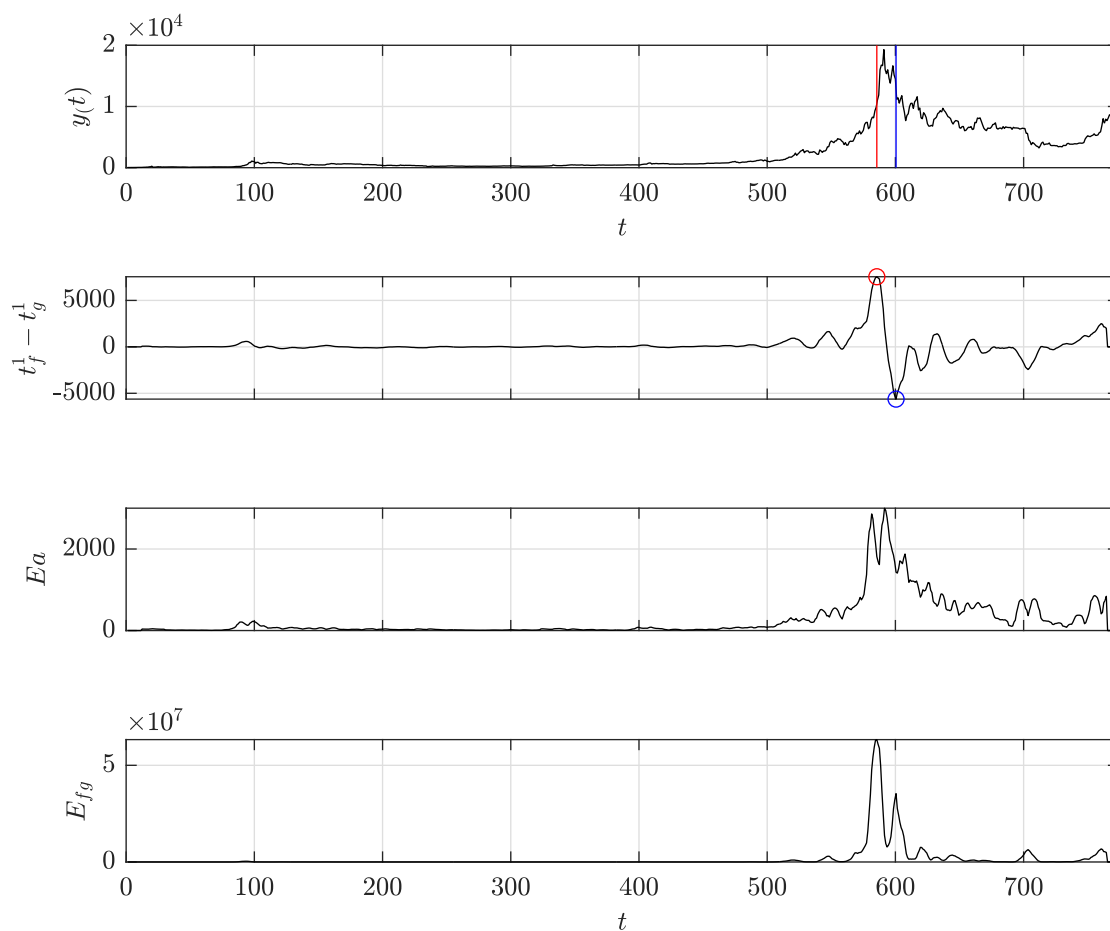


Fig. B.2: Detection of a possible mean-shift change point in the dataset "bitcoin" with support length $l_s = [10, 10]$ and constraint vector of $c = [0]$.

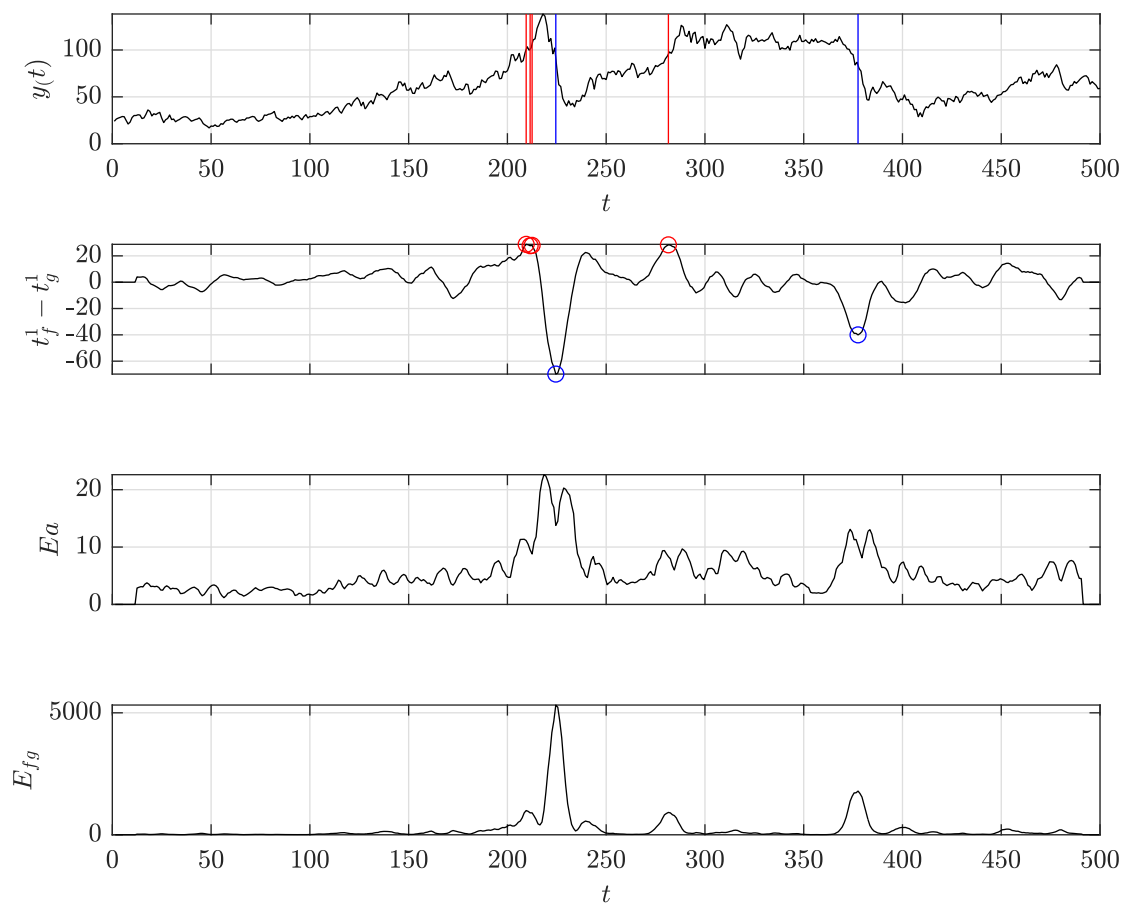


Fig. B.3: Detection of a possible mean-shift change point in the dataset "brent_spot" with support length $l_s = [10, 10]$ and constraint vector of $c = [0]$.

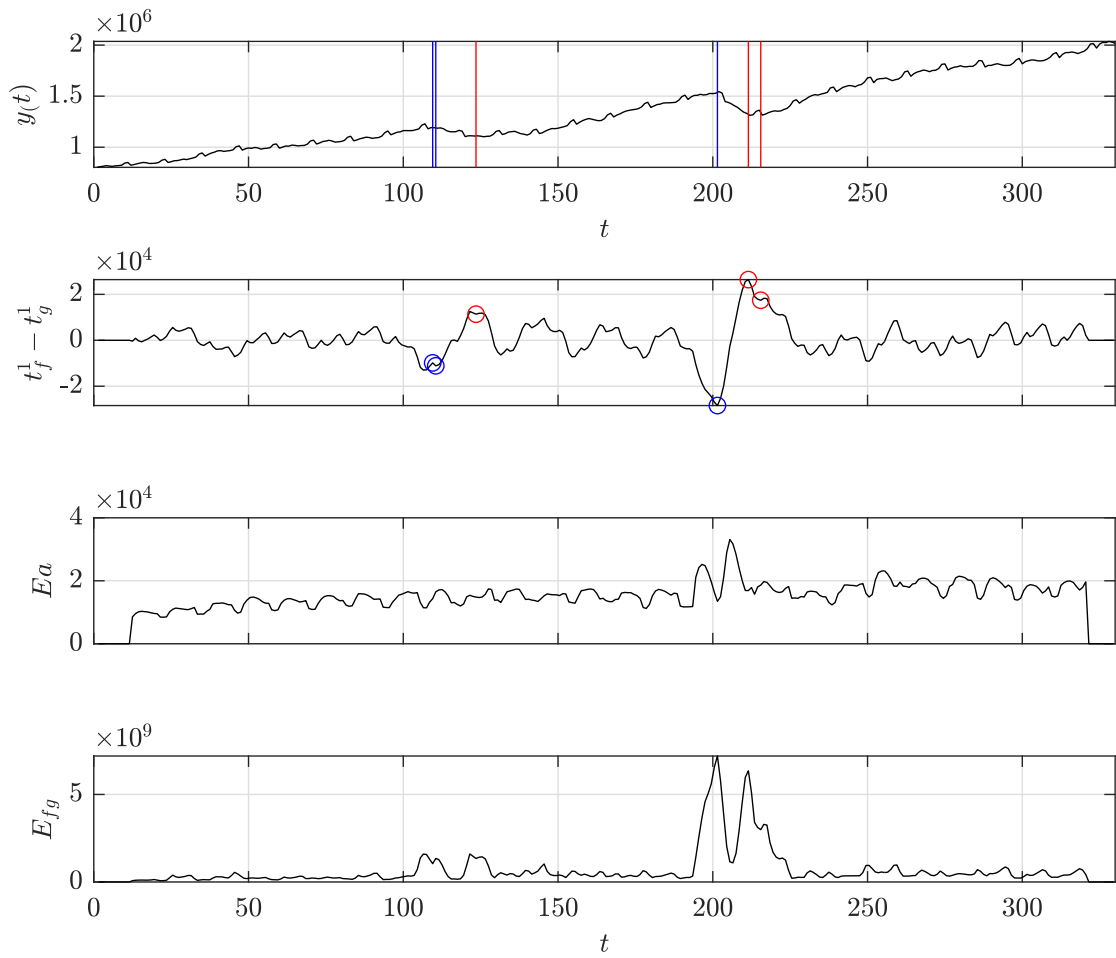


Fig. B.4: Detection of a possible slope-switch change point in the dataset "businv" with support length $l_s = [10, 10]$ and constraint vector of $c = [0, 1]$.

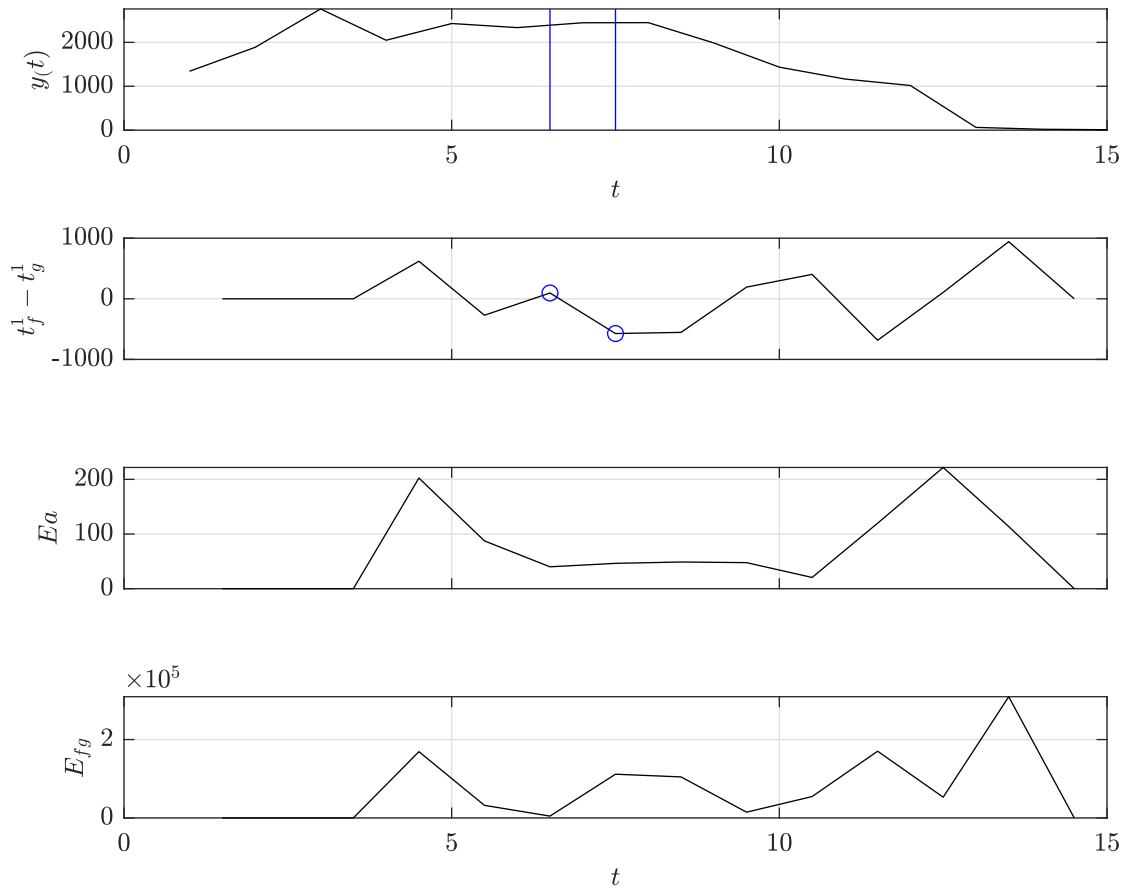


Fig. B.5: Detection of a possible slope-switch change point in the dataset "centralia" with support length $l_s = [2, 2]$ and constraint vector of $c = [0, 1]$.

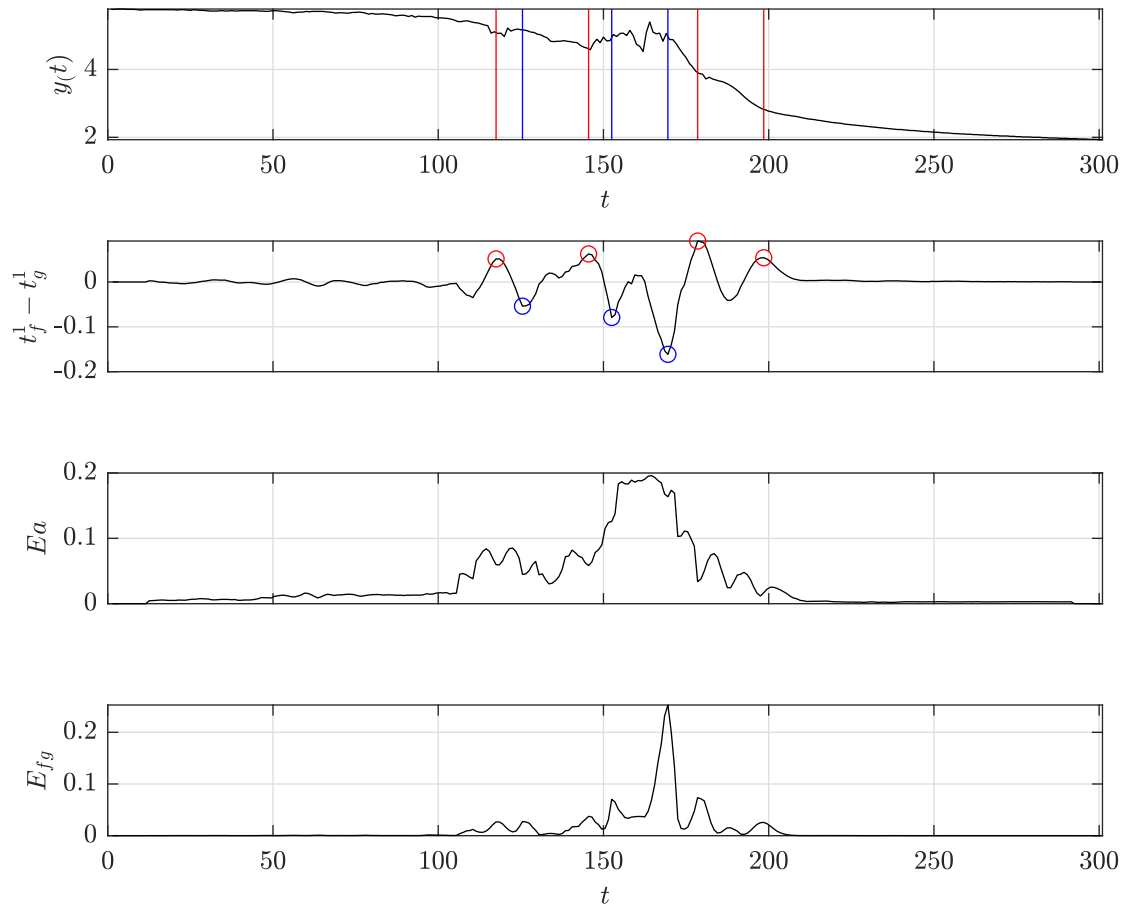


Fig. B.6: Detection of a possible slope-switch change point in the dataset "children_per_woman" with support length $l_s = [10, 10]$ and constraint vector of $c = [0, 1]$.

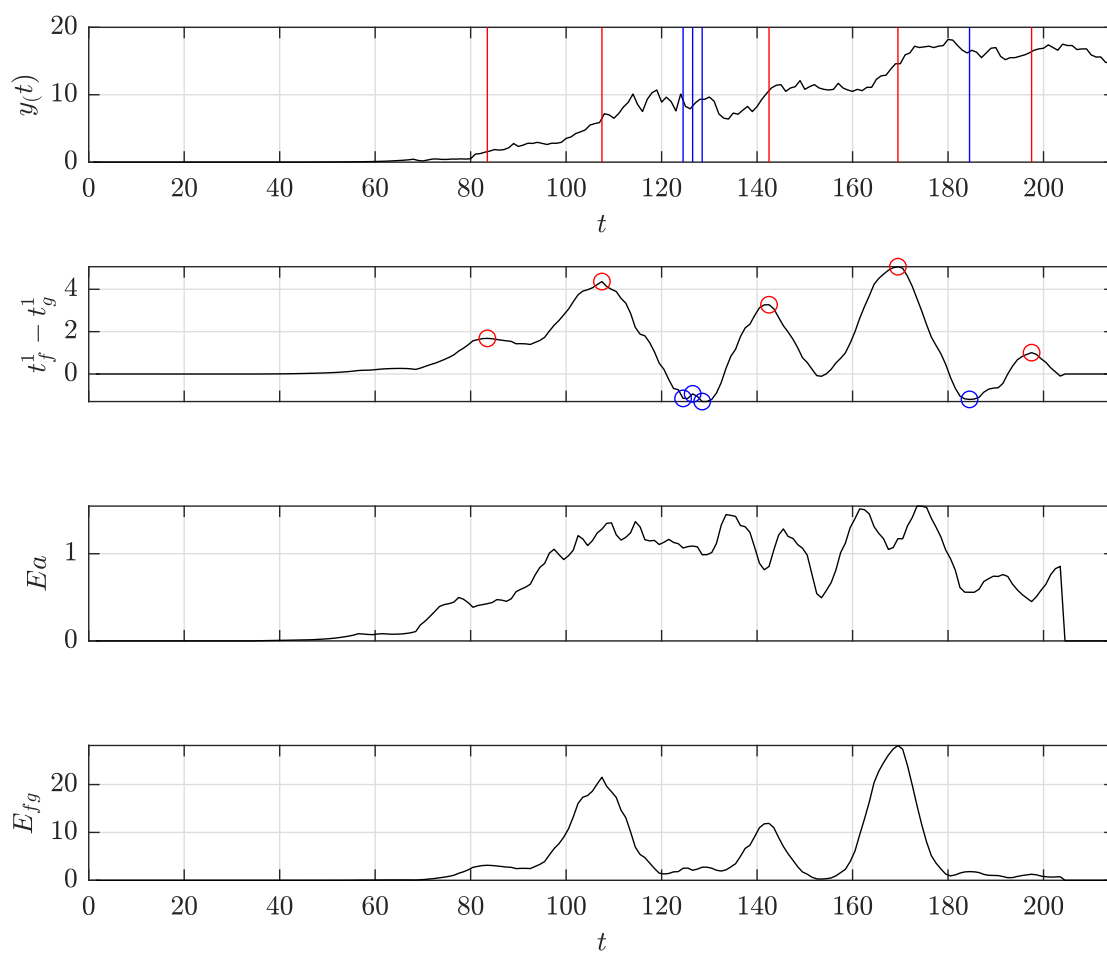


Fig. B.7: Detection of a possible mean-shift change point in the dataset "co2.canada" with support length $l_s = [12, 12]$ and constraint vector of $c = [0]$.

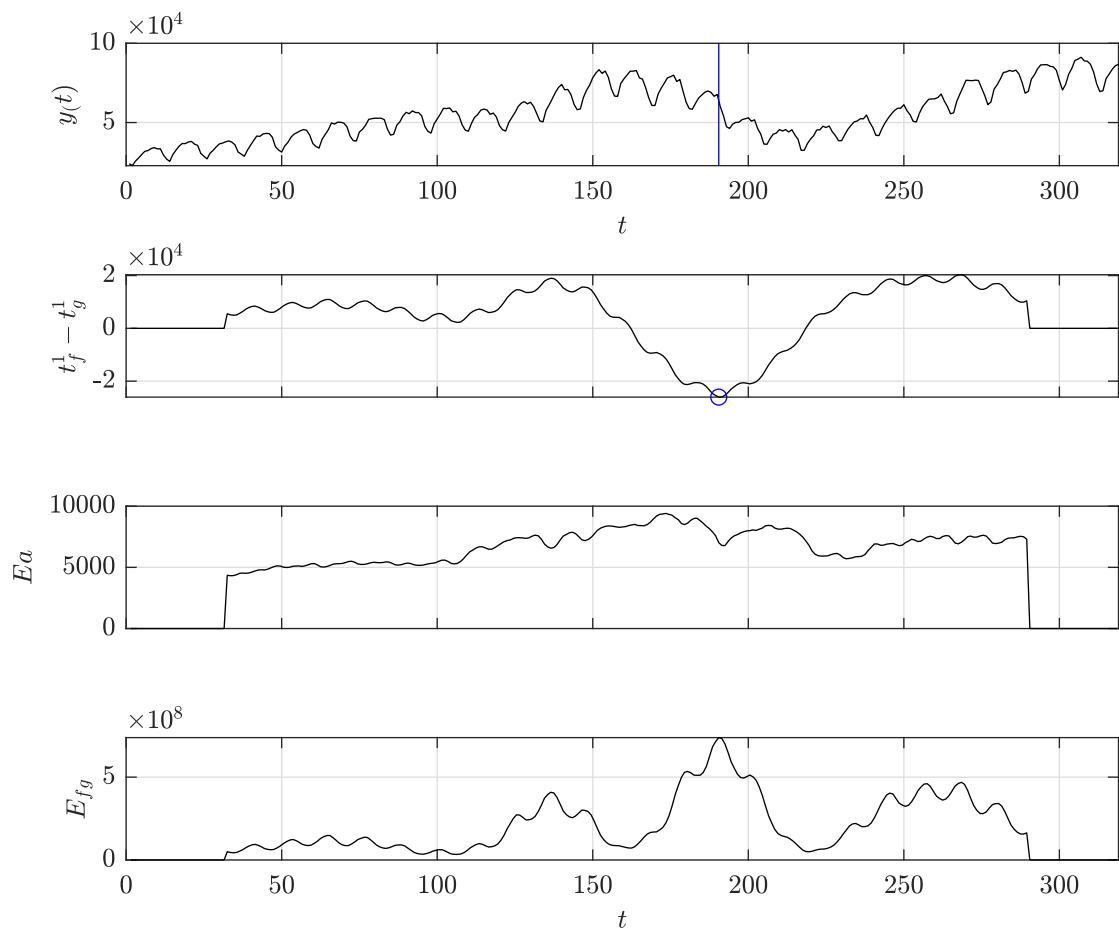


Fig. B.8: Detection of a possible mean-shift change point in the dataset "construction" with support length $l_s = [30, 30]$ and constraint vector of $c = [0]$.

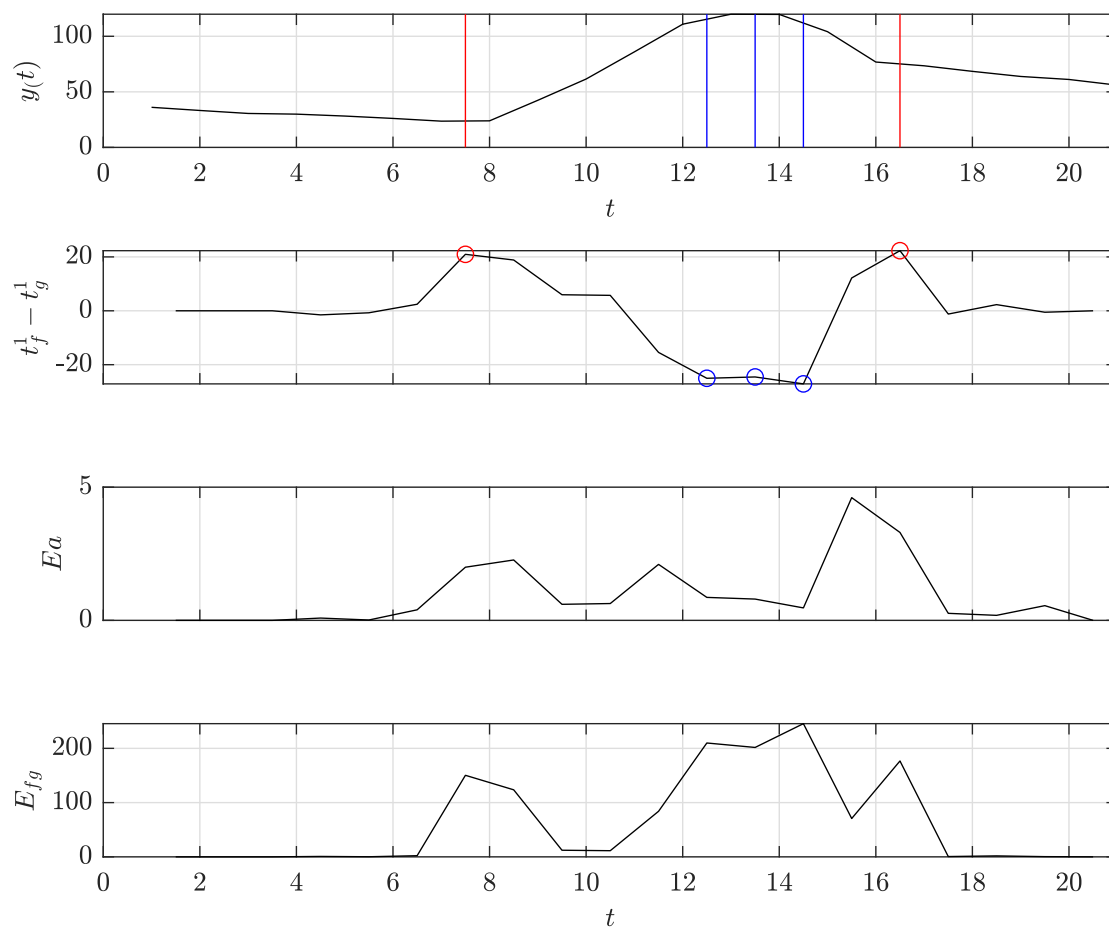


Fig. B.9: Detection of a possible slope-switch change point in the dataset "debt_ireland" with support length $l_s = [2, 2]$ and constraint vector of $c = [0, 1]$.

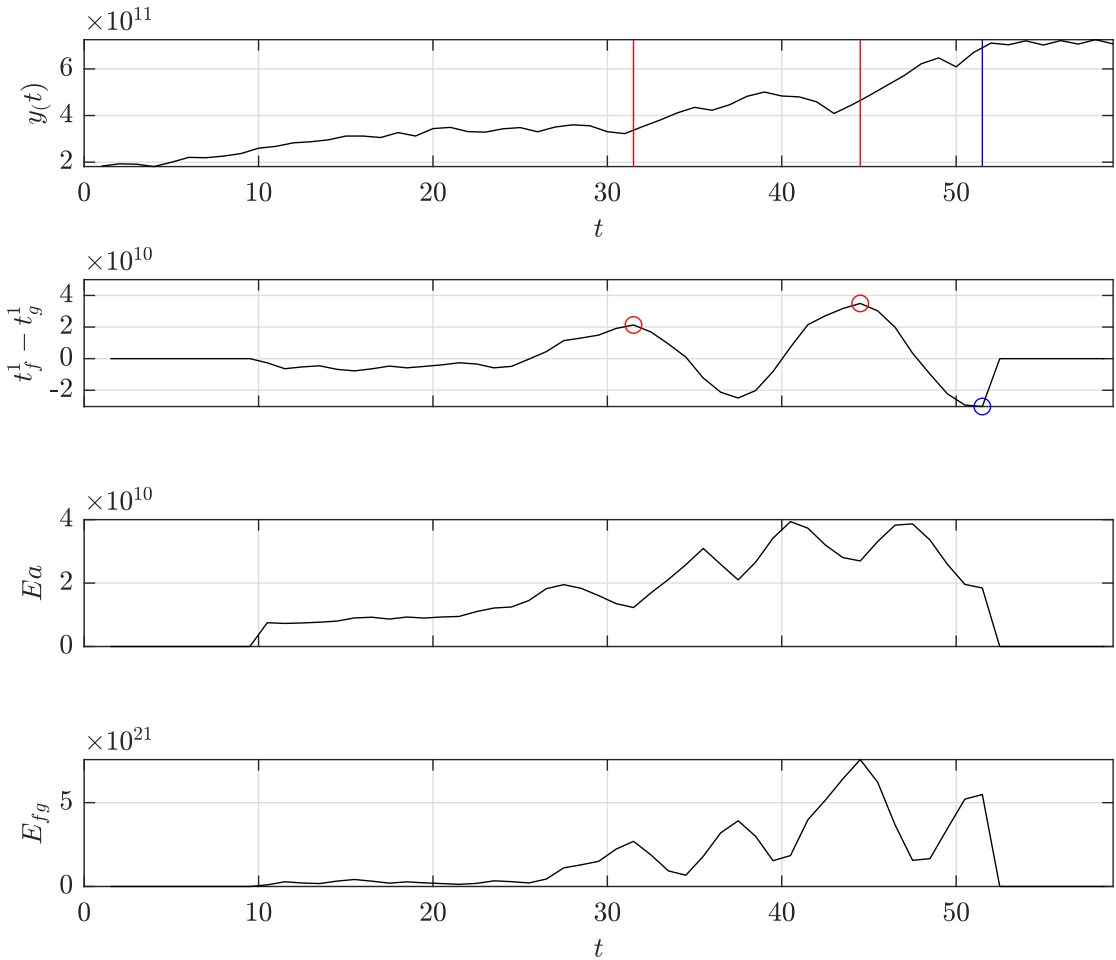


Fig. B.10: Detection of a possible slope-switch change point in the dataset "gdp_argentina" with support length $l_s = [8, 8]$ and constraint vector of $c = [0, 1]$.

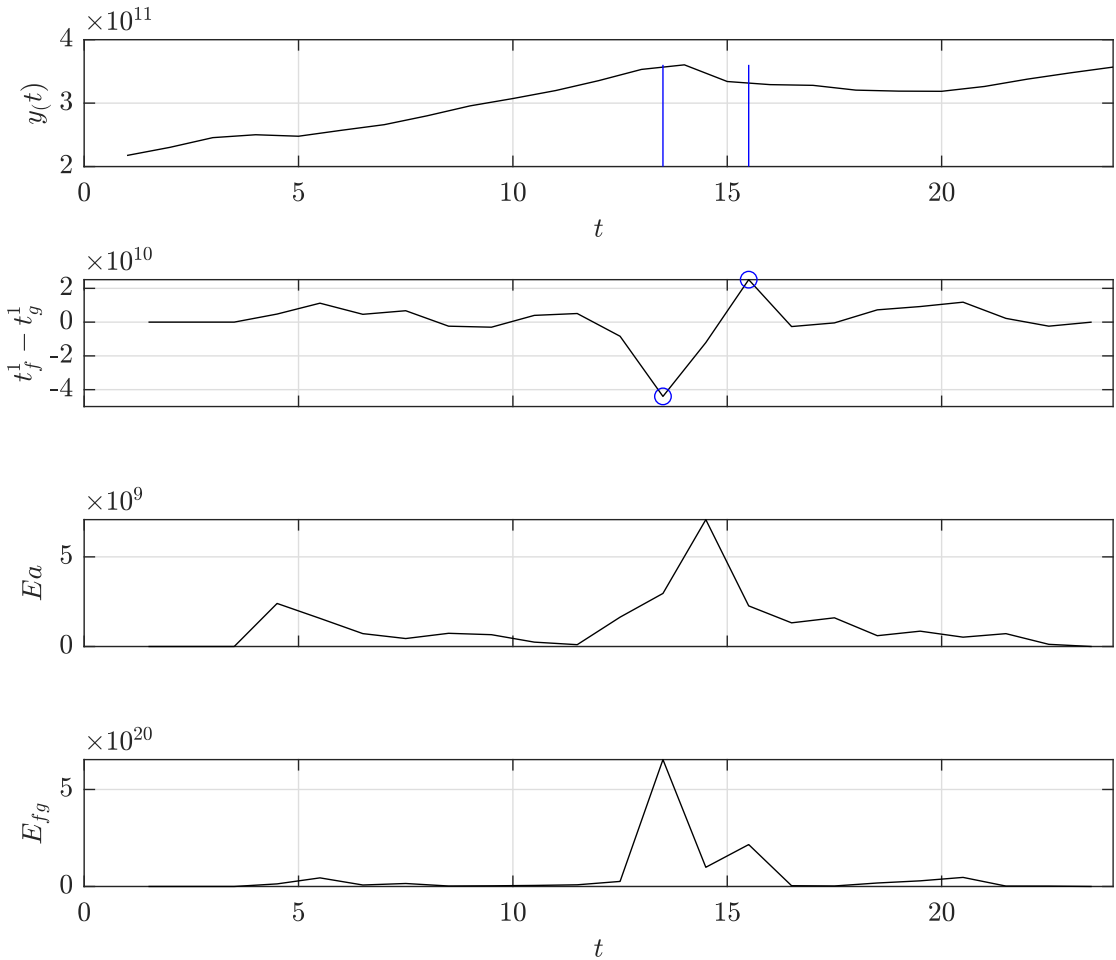


Fig. B.11: Detection of a possible slope-switch change point in the dataset "gdp_croatia" with support length $l_s = [2, 2]$ and constraint vector of $c = [0, 1]$.

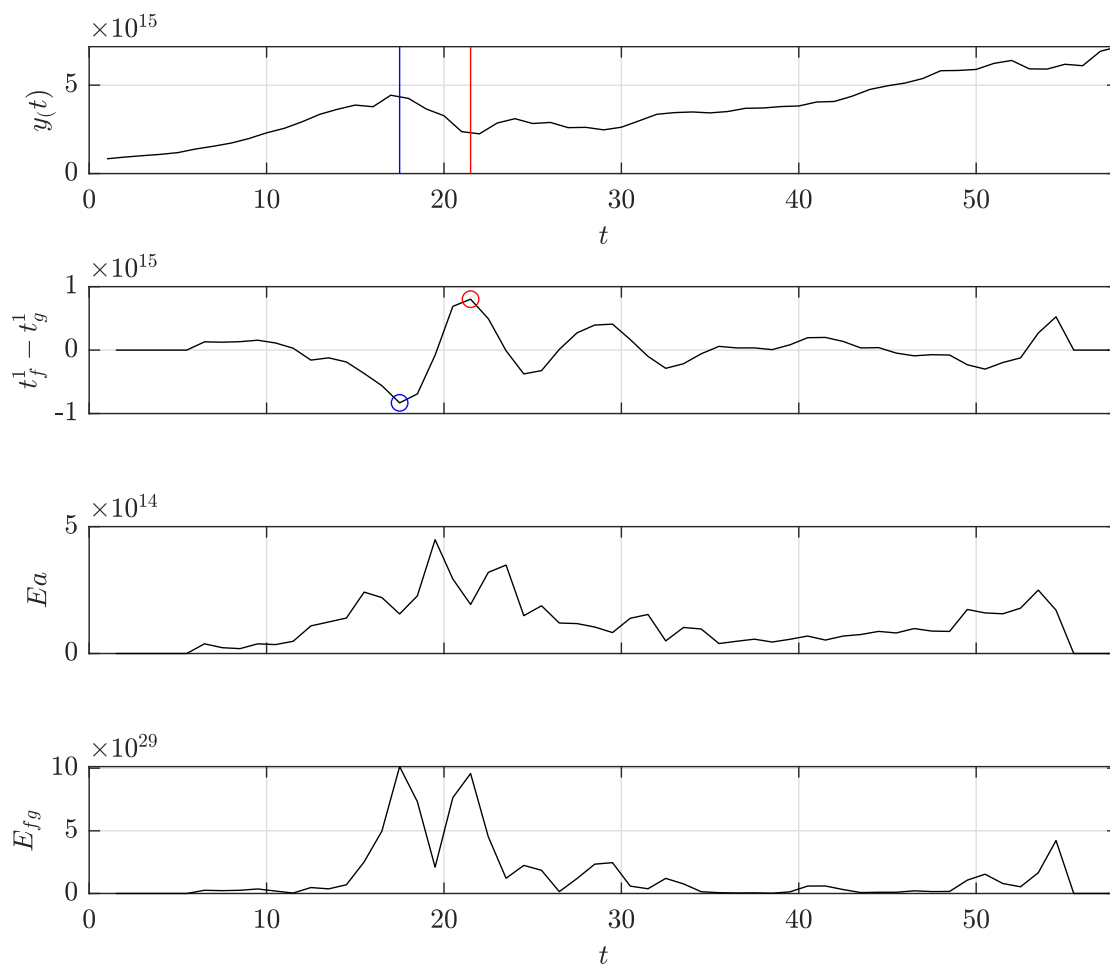


Fig. B.12: Detection of a possible slope-switch change point in the dataset "gdp_iran" with support length $l_s = [4, 4]$ and constraint vector of $c = [0, 1]$.

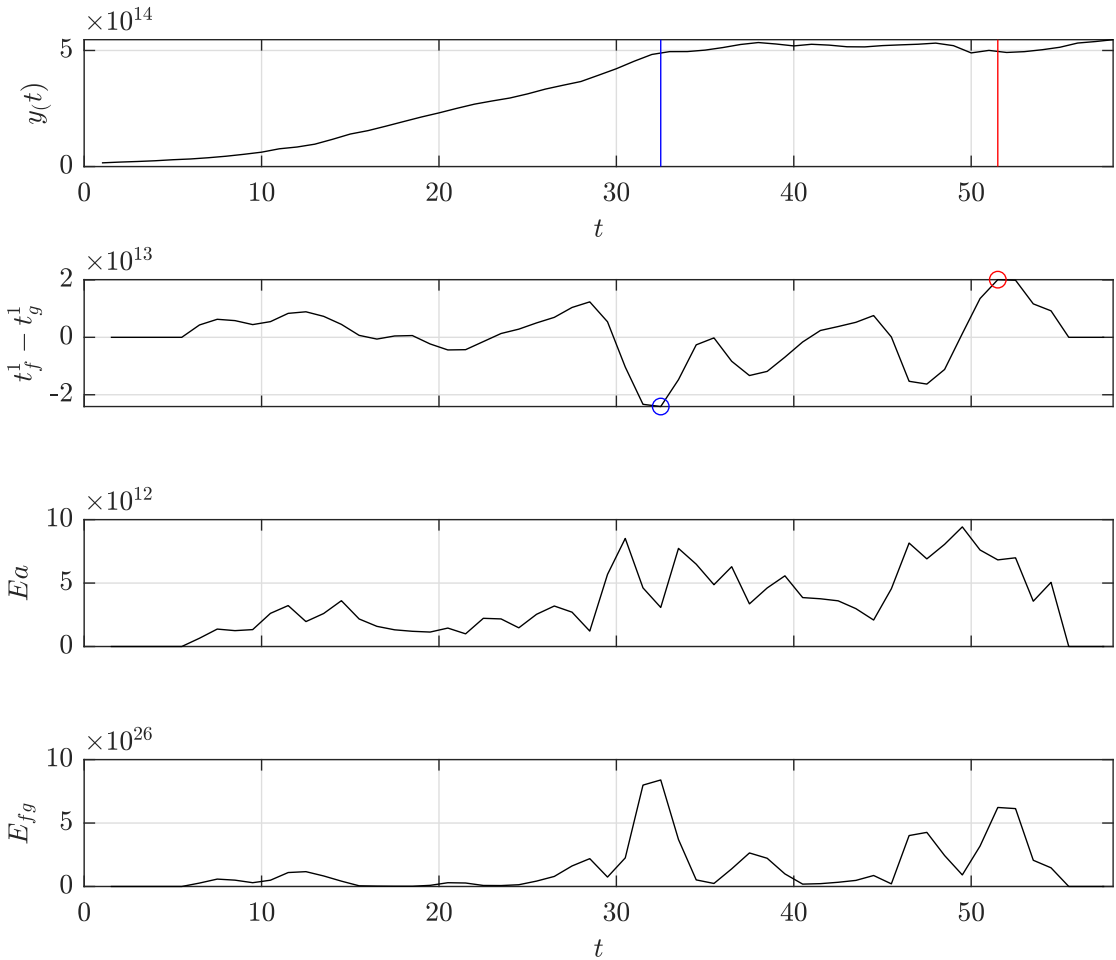


Fig. B.13: Detection of a possible slope-switch change point in the dataset "gdp_japan" with support length $l_s = [4, 4]$ and constraint vector of $c = [0, 1]$.

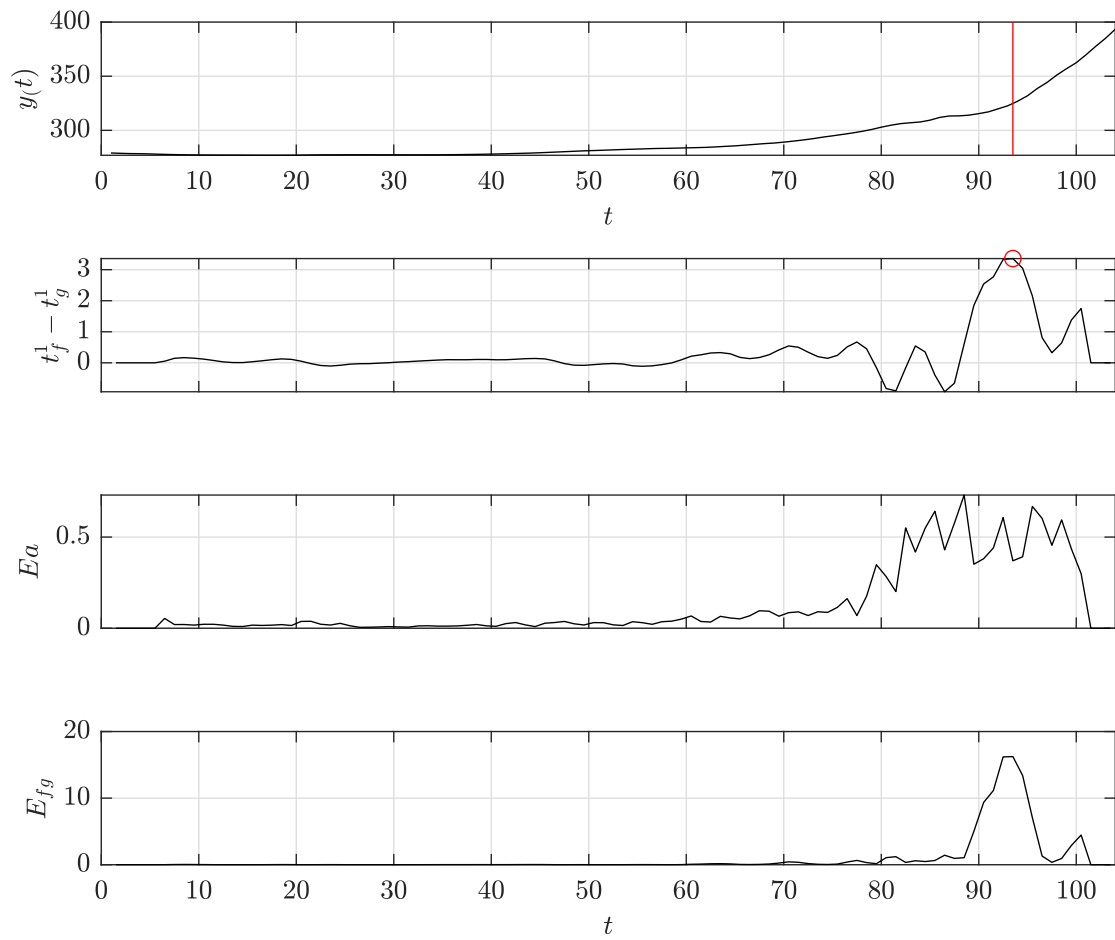


Fig. B.14: Detection of a possible slope-switch change point in the dataset "global_co2" with support length $l_s = [4, 4]$ and constraint vector of $c = [0, 1]$.

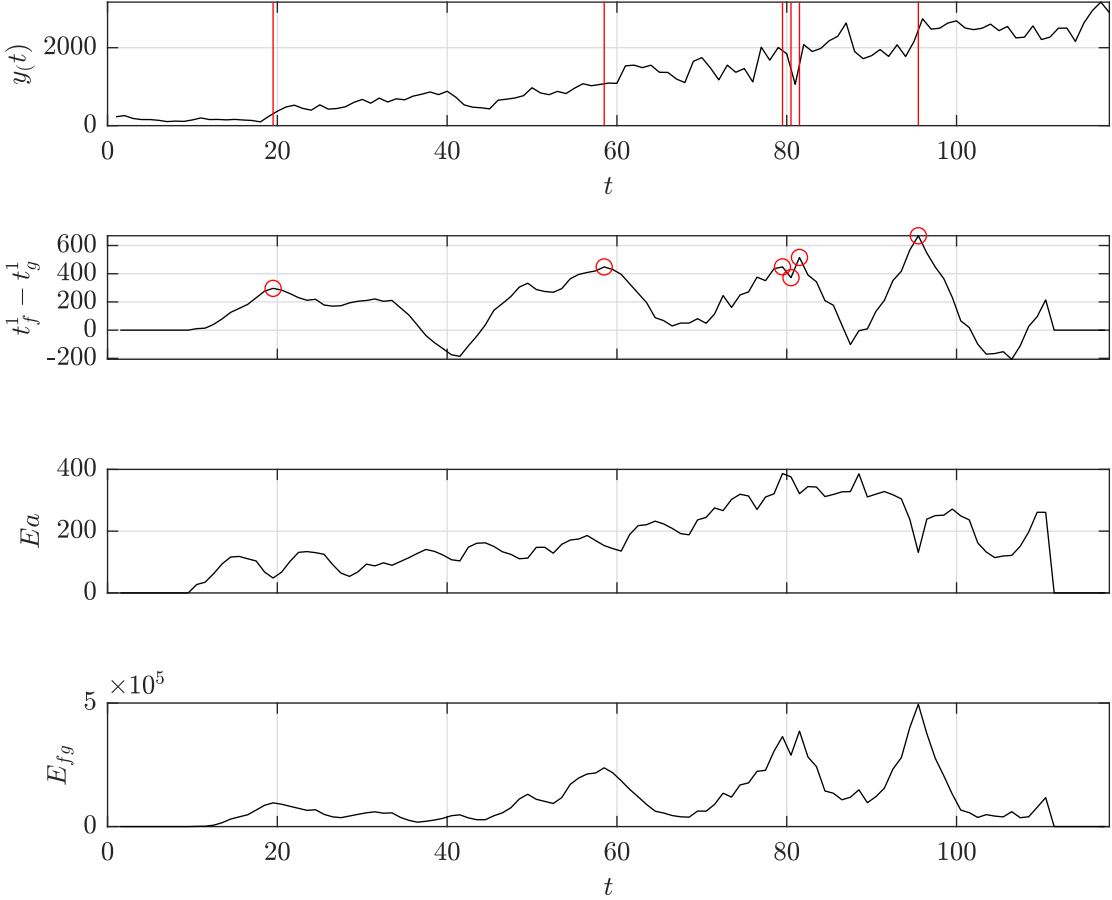


Fig. B.15: Detection of a possible mean-shift change point in the dataset "homeruns" with support length $l_s = [8, 8]$ and constraint vector of $c = [0]$.

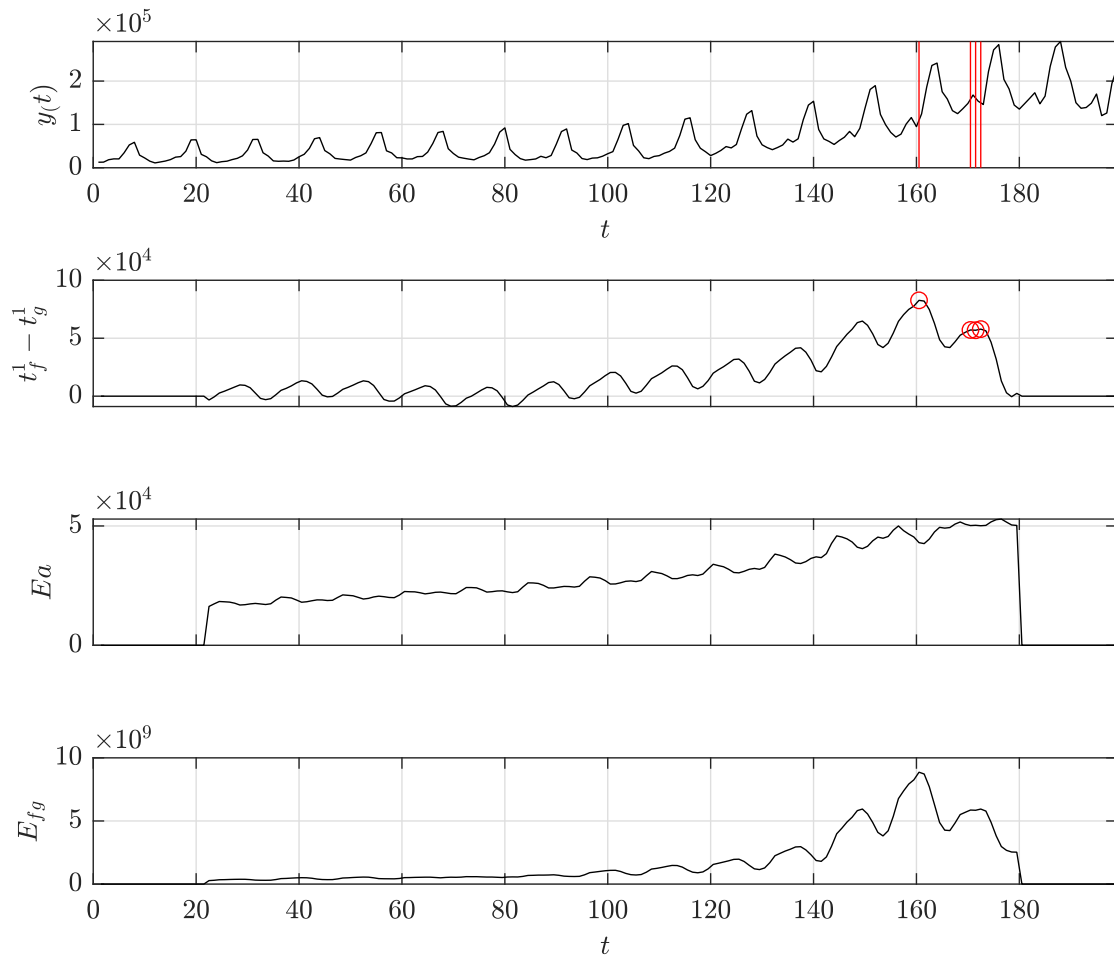


Fig. B.16: Detection of a possible mean-shift change point in the dataset "iceland_tourism" with support length $l_s = [20, 20]$ and constraint vector of $c = [0]$.

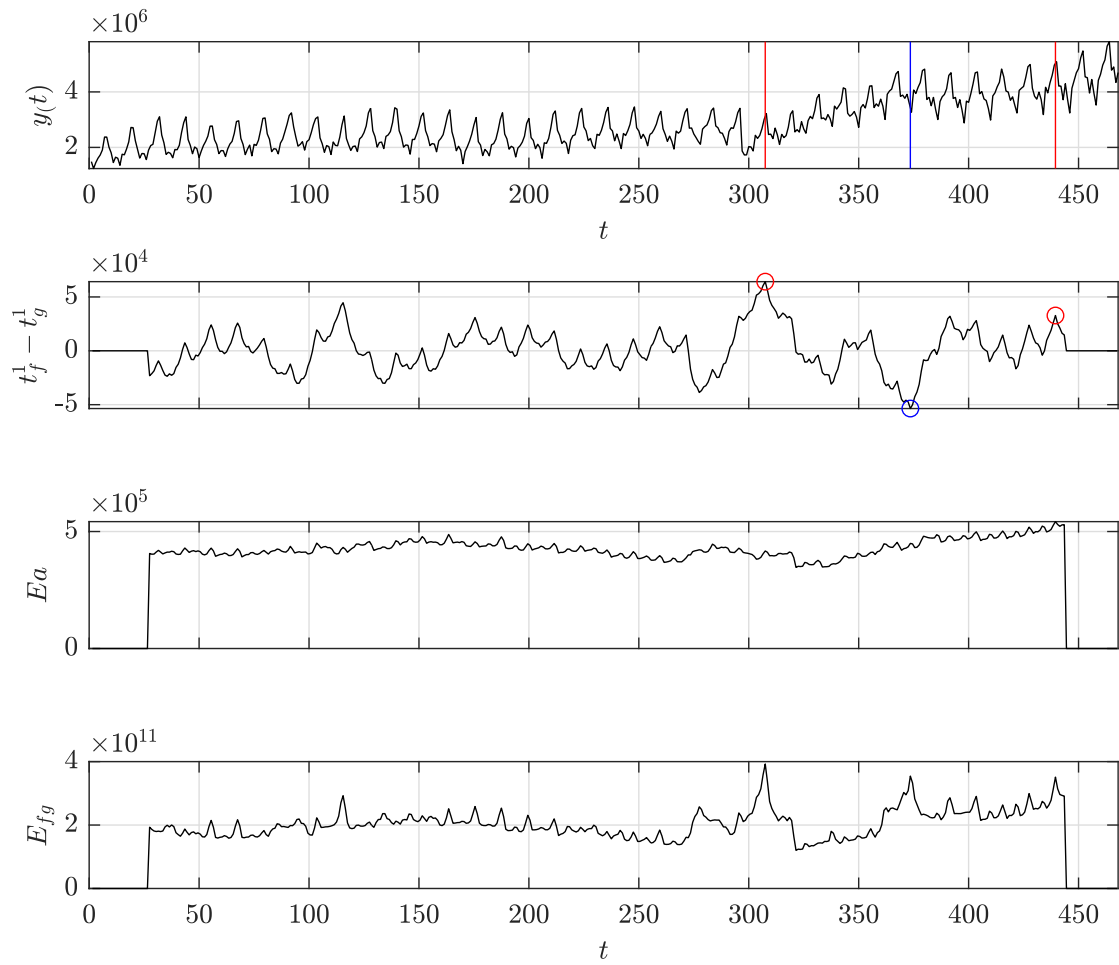


Fig. B.17: Detection of a possible slope-switch change point in the dataset "jfk_passengers" with support length $l_s = [25, 25]$ and constraint vector of $c = [0, 1]$.

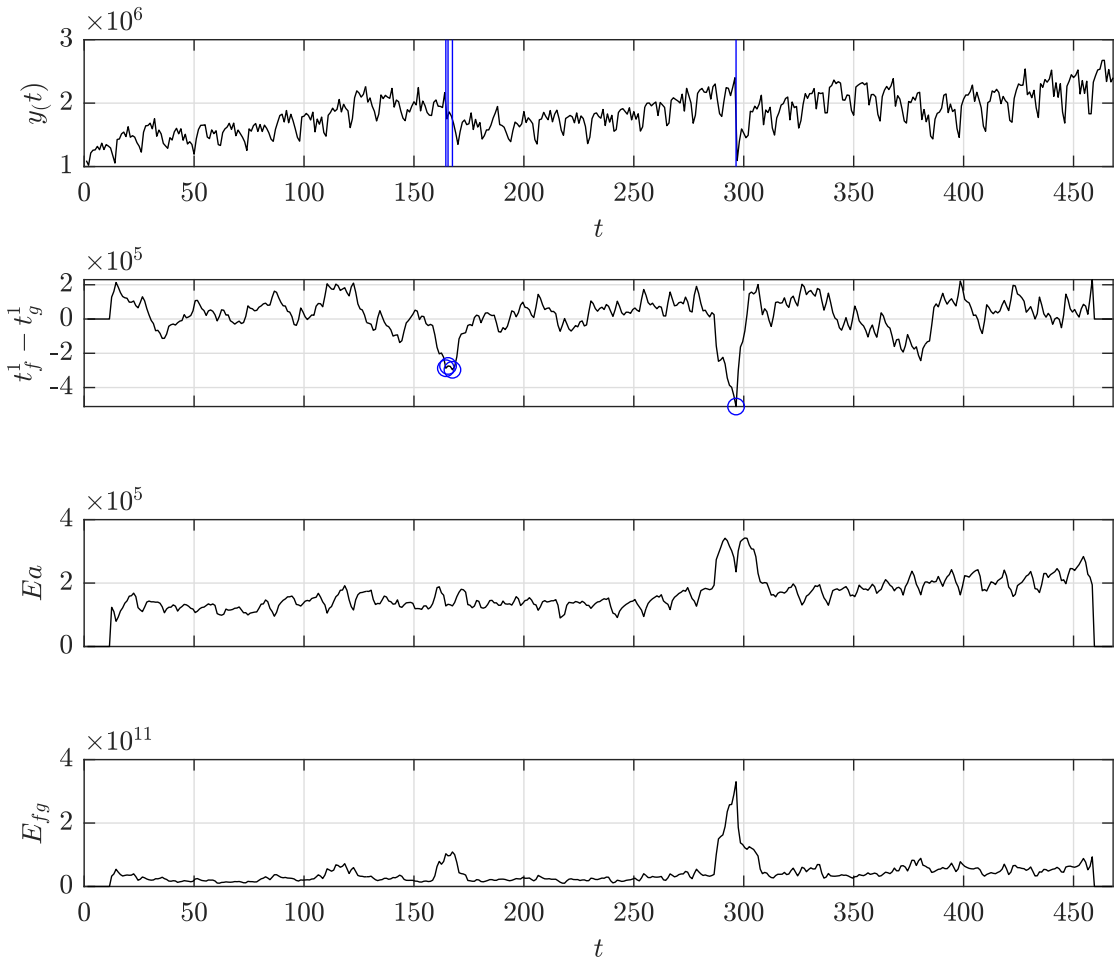


Fig. B.18: Detection of a possible mean-shift change point in the dataset "lga_passengers" with support length $l_s = [10, 10]$ and constraint vector of $c = [0]$.

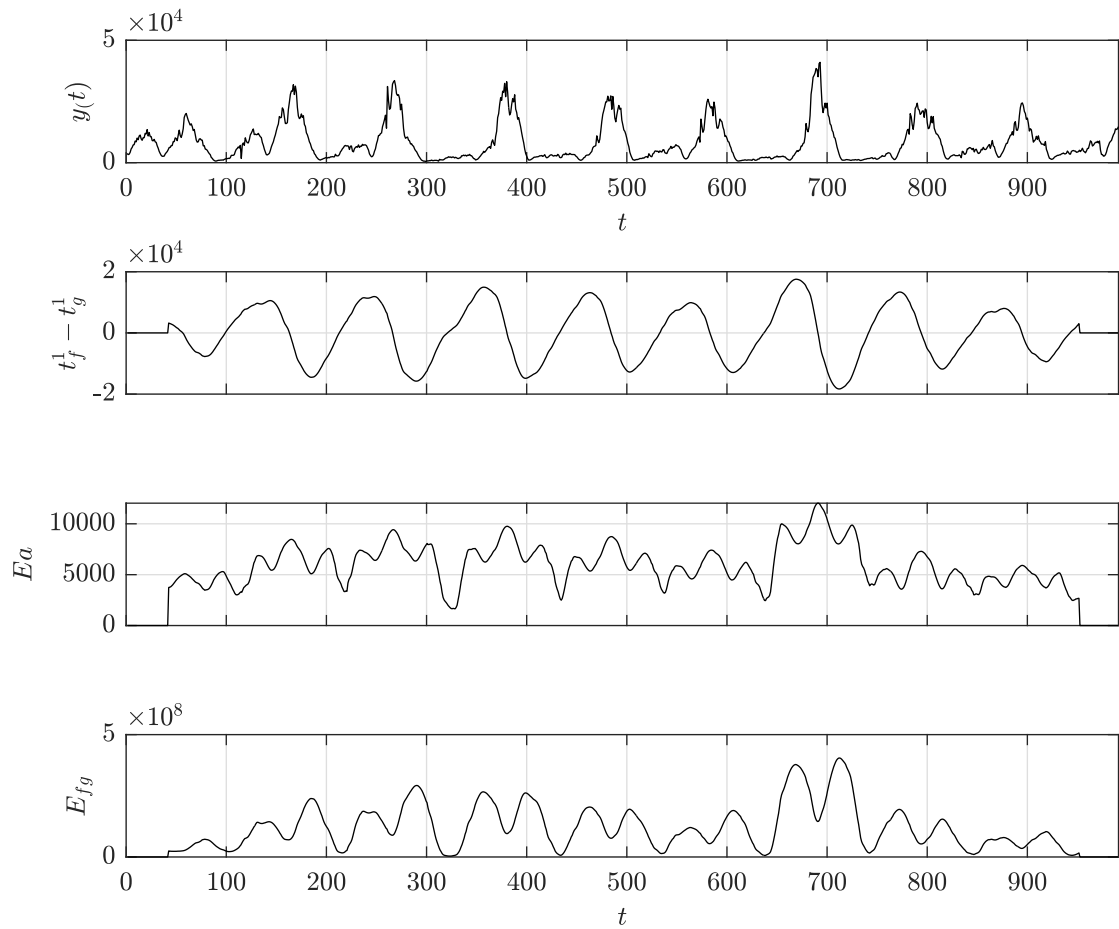


Fig. B.19: Detection of a possible mean-shift change point in the dataset "measles" with support length $l_s = [40, 40]$ and constraint vector of $c = [0]$.

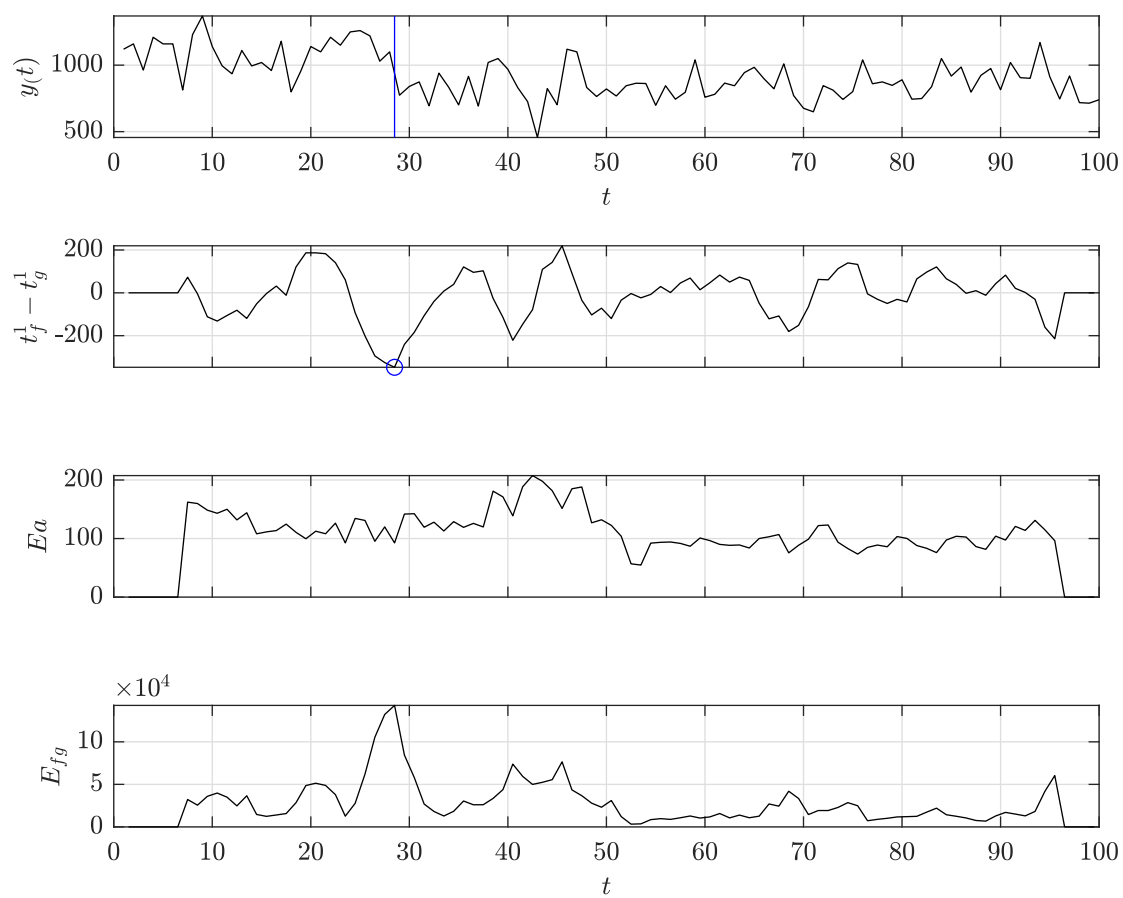


Fig. B.20: Detection of a possible mean-shift change point in the dataset "nile" with support length $l_s = [5, 5]$ and constraint vector of $c = [0]$.

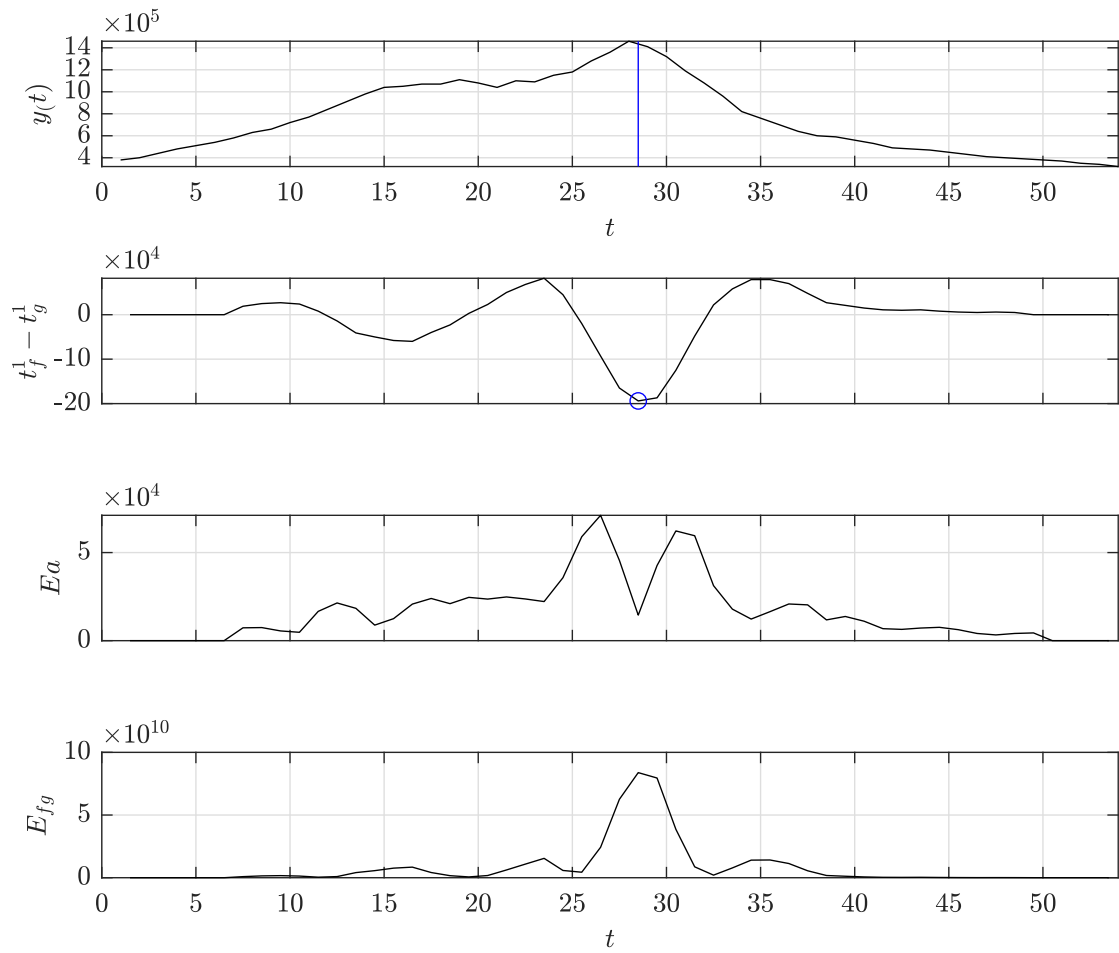


Fig. B.21: Detection of a possible slope-switch change point in the dataset "ozone" with support length $l_s = [5, 5]$ and constraint vector of $c = [0, 1]$.

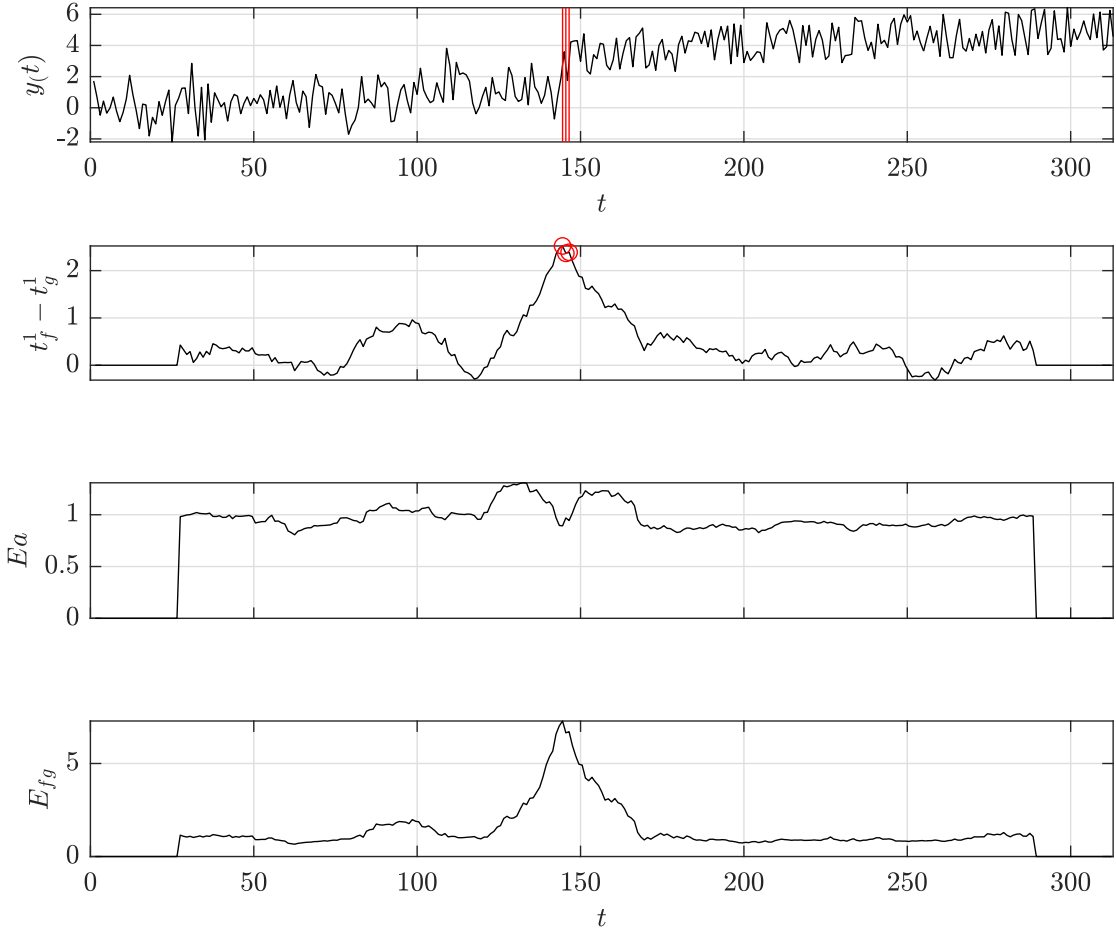


Fig. B.22: Detection of a possible mean-shift change point in the dataset "quality_control_1" with support length $l_s = [25, 25]$ and constraint vector of $c = [0]$.

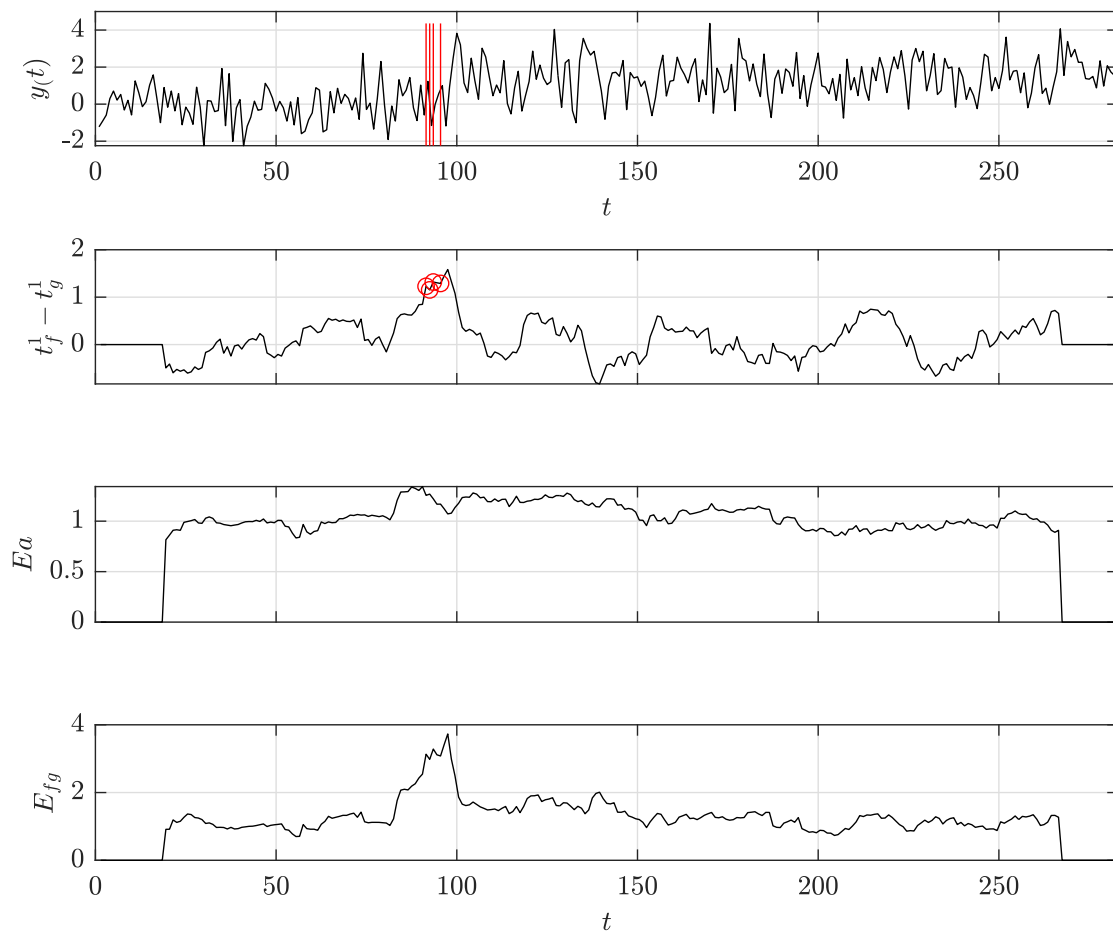


Fig. B.23: Detection of a possible mean-shift change point in the dataset "quality_control_2" with support length $l_s = [17, 17]$ and constraint vector of $c = [0]$.

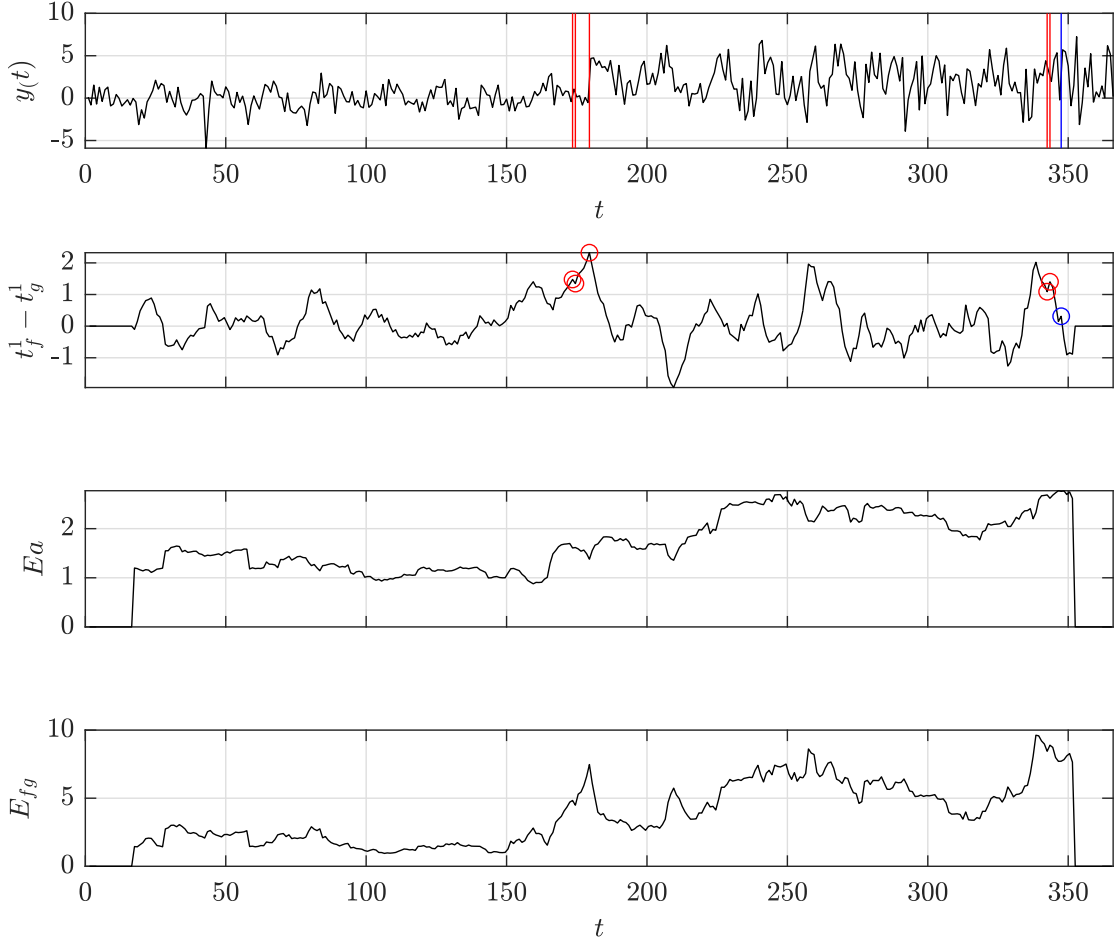


Fig. B.24: Detection of a possible mean-shift change point in the dataset "quality_control.3" with support length $l_s = [15, 15]$ and constraint vector of $c = [0]$.

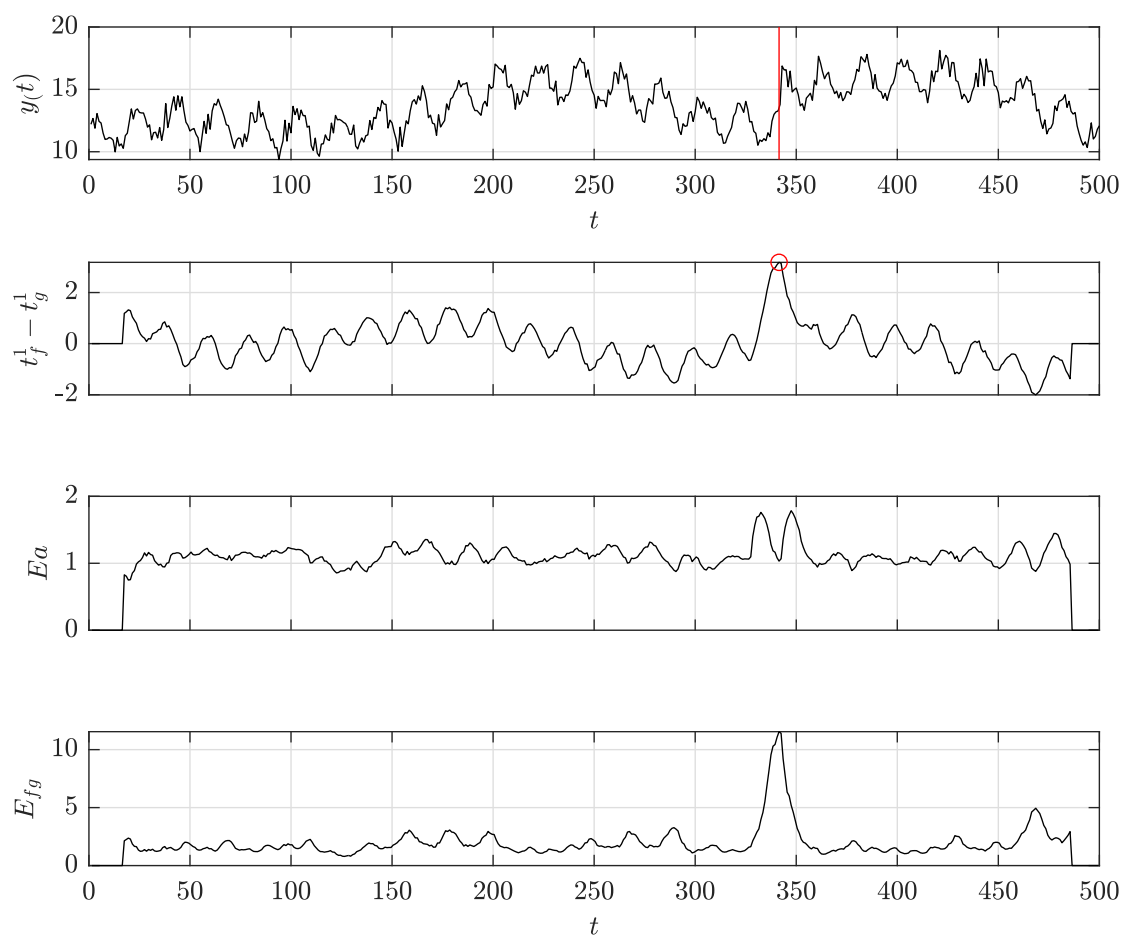


Fig. B.25: Detection of a possible mean-shift change point in the dataset "quality_control.4" with support length $l_s = [15, 15]$ and constraint vector of $c = [0]$.

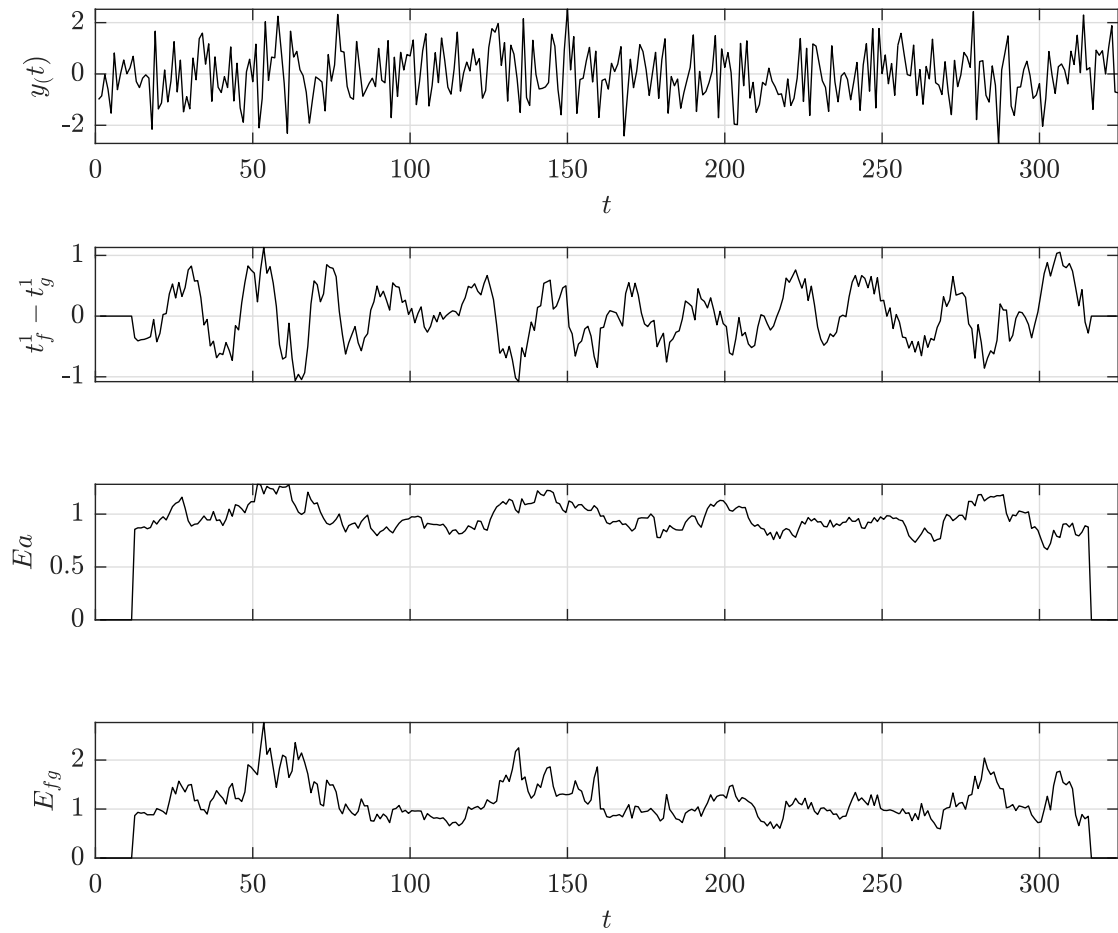


Fig. B.26: Detection of a possible mean-shift change point in the dataset "quality_control.5" with support length $l_s = [10, 10]$ and constraint vector of $c = [0]$.

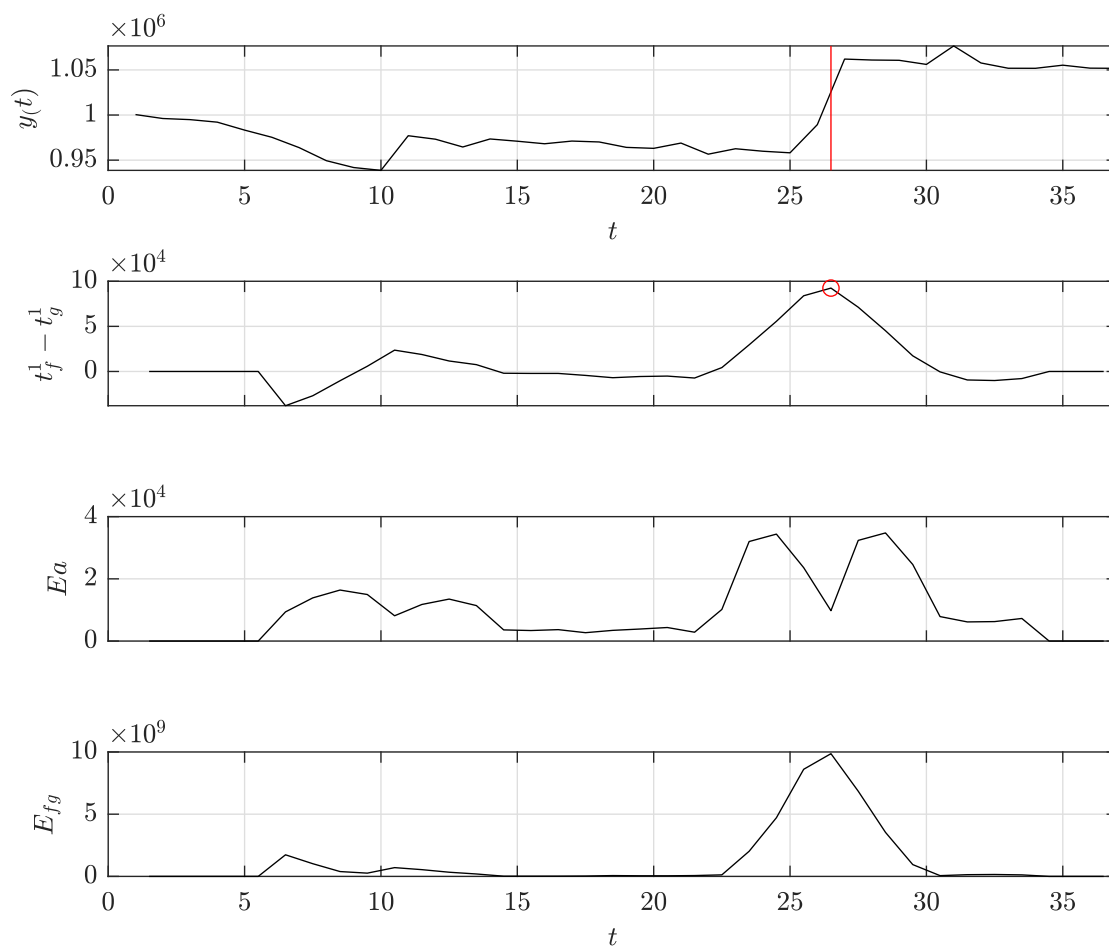


Fig. B.27: Detection of a possible mean-shift change point in the dataset "rail_lines" with support length $l_s = [4, 4]$ and constraint vector of $c = [0]$.

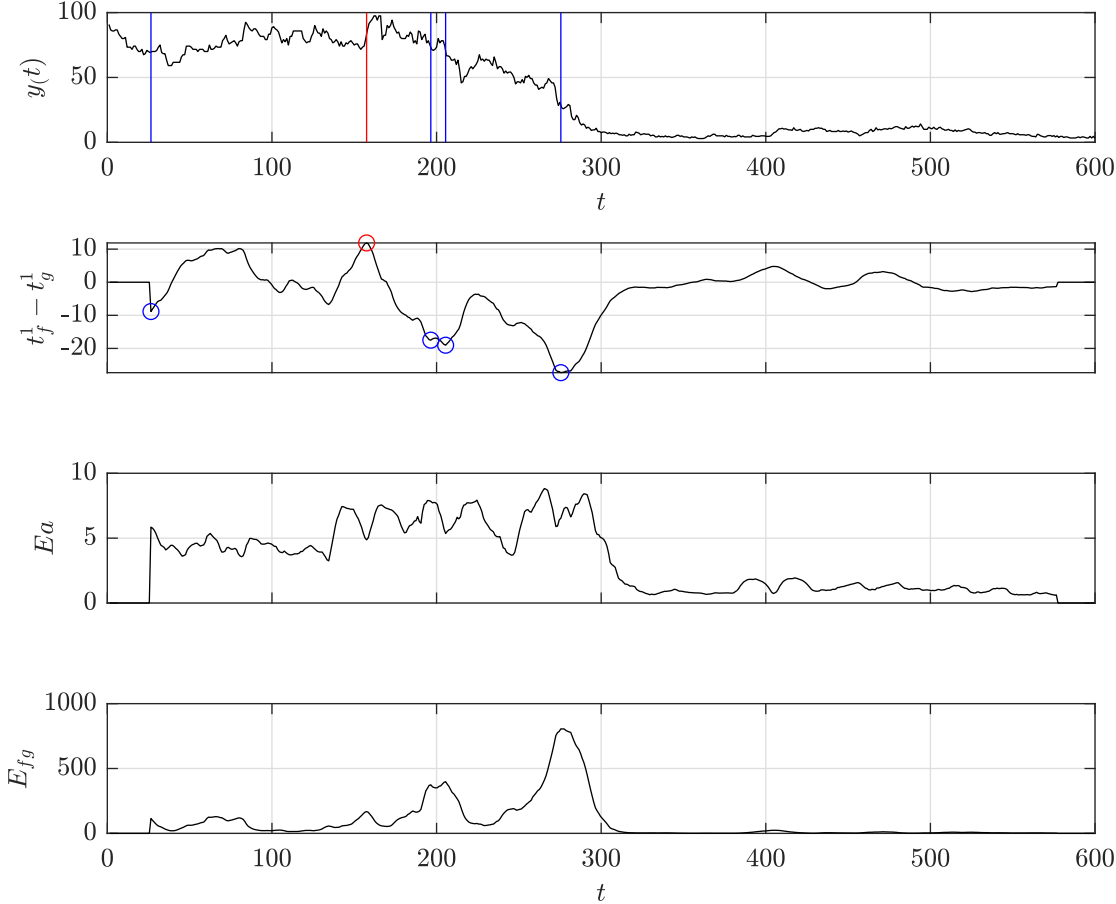


Fig. B.28: Detection of a possible mean-shift change point in the dataset "ratner_stock" with support length $l_s = [24, 24]$ and constraint vector of $c = [0]$.

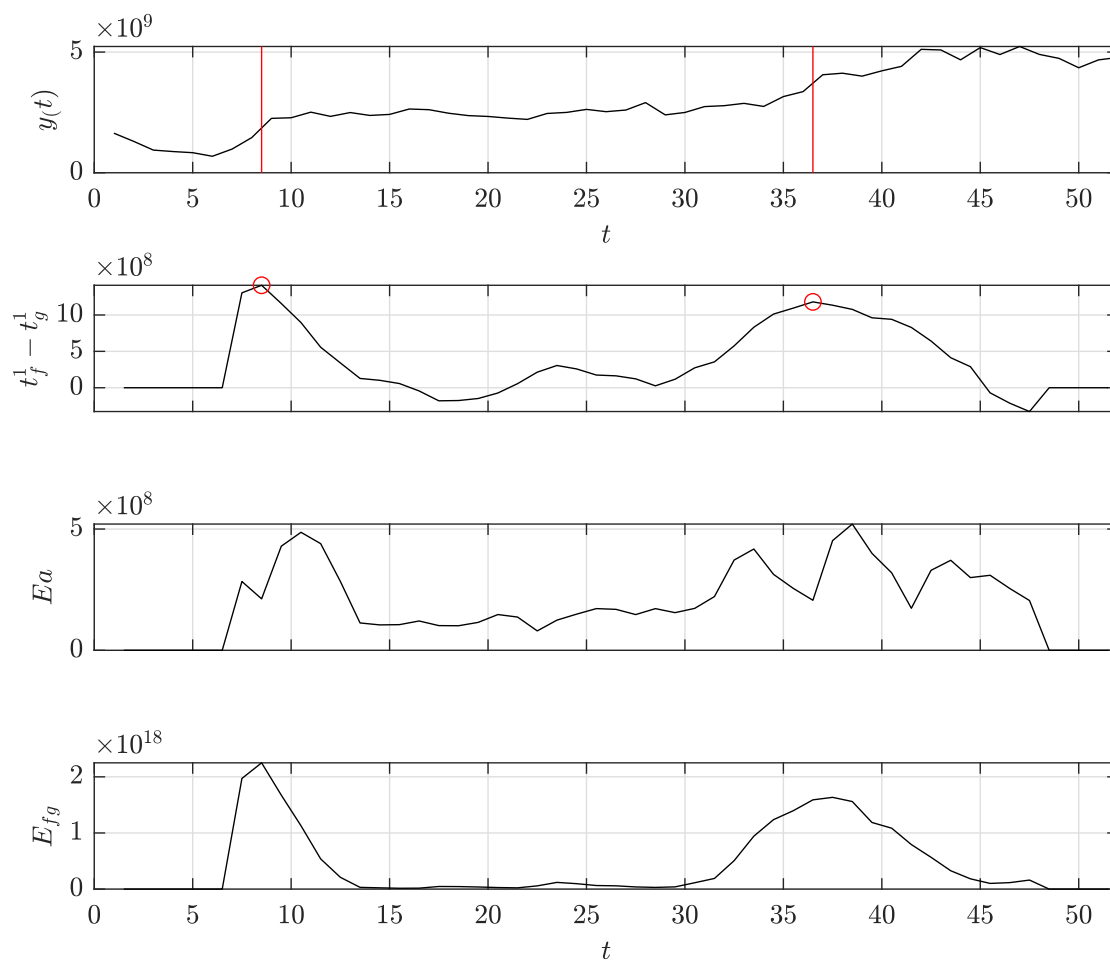


Fig. B.29: Detection of a possible mean-shift change point in the dataset "robocalls" with support length $l_s = [5, 5]$ and constraint vector of $c = [0]$.

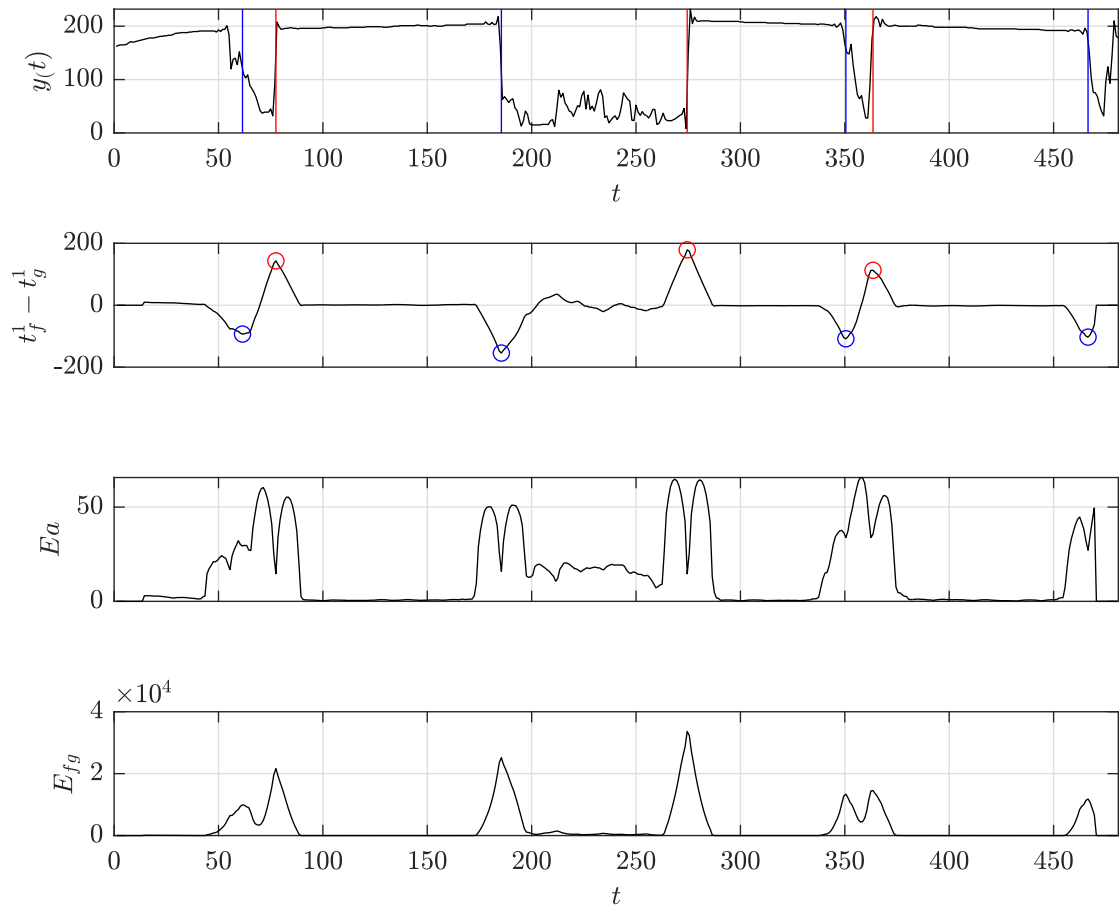


Fig. B.30: Detection of a possible mean-shift change point in the dataset "scanline_420491" with support length $l_s = [12, 12]$ and constraint vector of $c = [0]$.

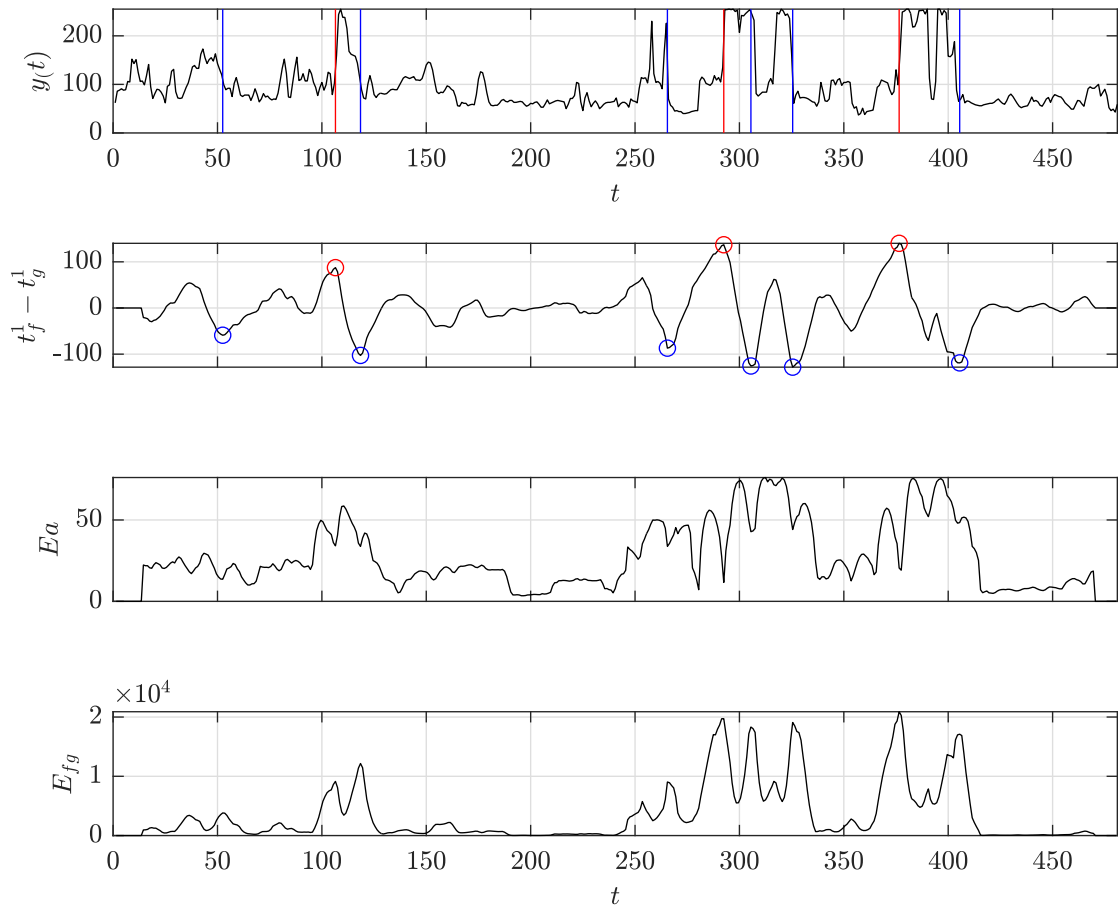


Fig. B.31: Detection of a possible mean-shift change point in the dataset "scanline_126007" with support length $l_s = [12, 12]$ and constraint vector of $c = [0]$.

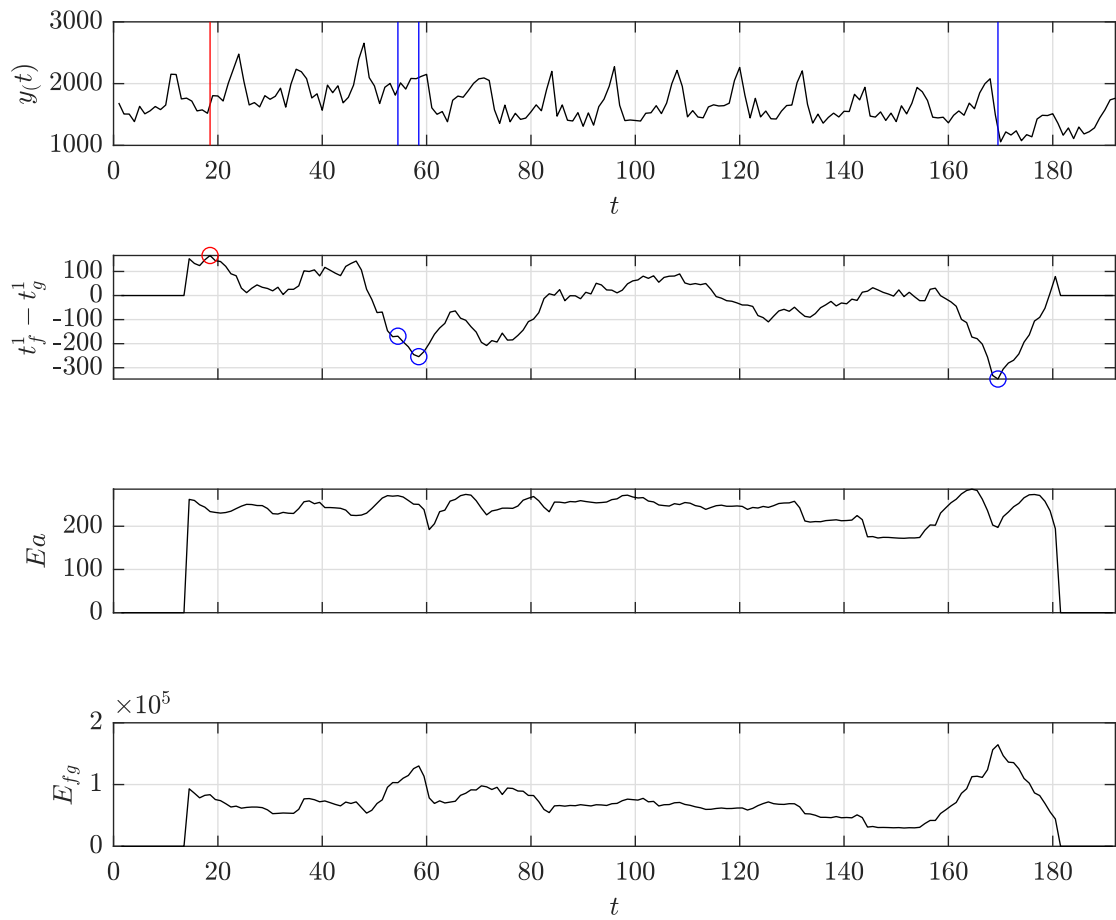


Fig. B.32: Detection of a possible mean-shift change point in the dataset "seatbelts" with support length $l_s = [12, 12]$ and constraint vector of $c = [0]$.

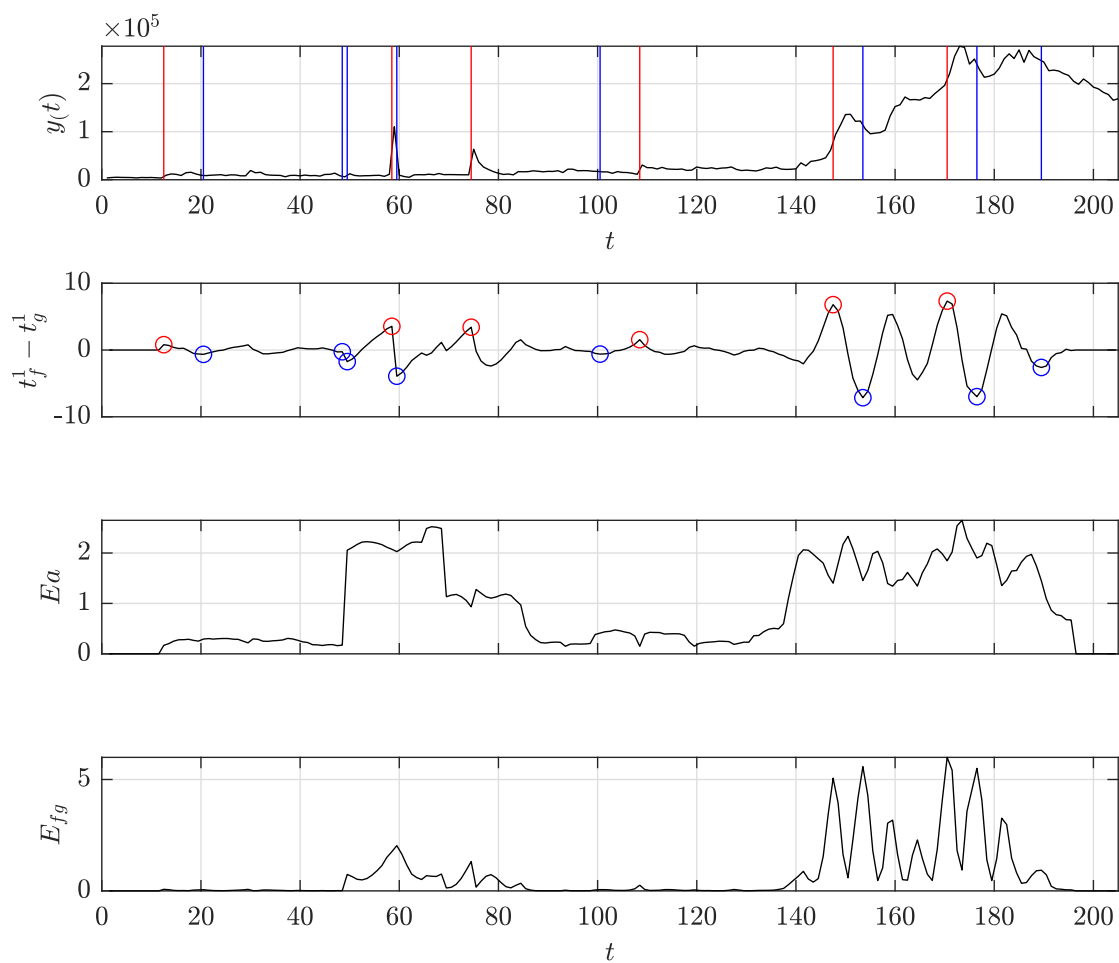


Fig. B.33: Detection of a possible slope-switch change point in the dataset "shanghai_license" with support length $l_s = [15, 15]$ and constraint vector of $c = [0, 1]$.

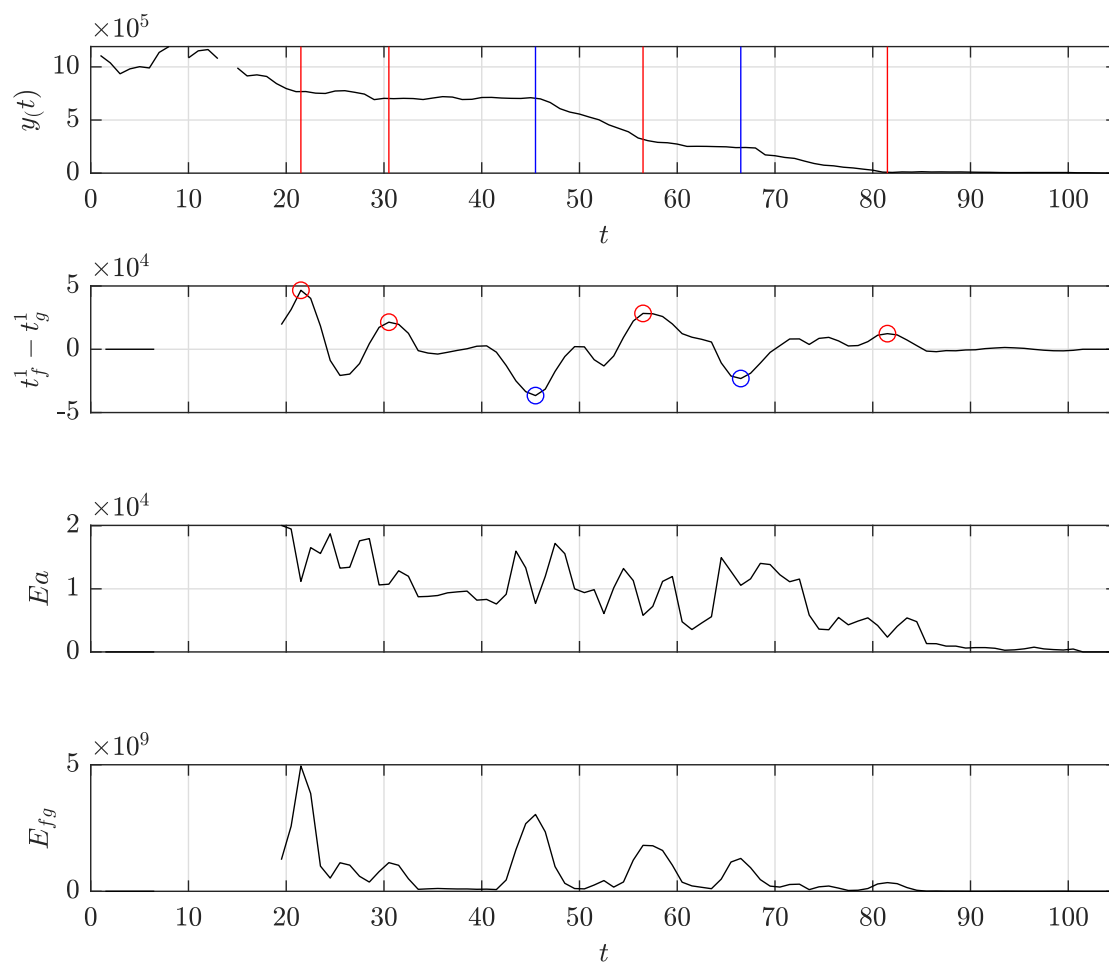


Fig. B.34: Detection of a possible slope-switch change point in the dataset "uk_coal_employ" with support length $l_s = [5, 5]$ and constraint vector of $c = [0, 1]$.

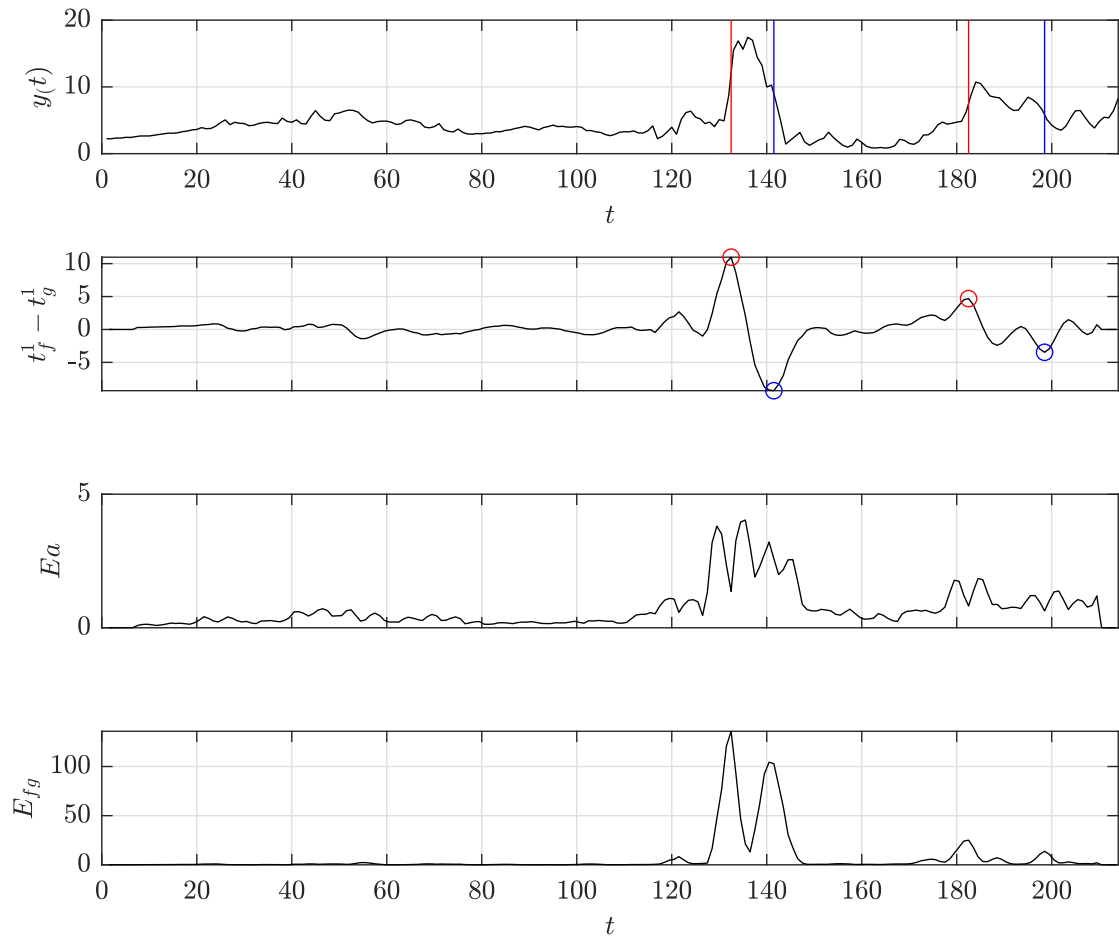


Fig. B.35: Detection of a possible mean-shift change point in the dataset "unemployment_nl" with support length $l_s = [5, 5]$ and constraint vector of $c = [0]$.

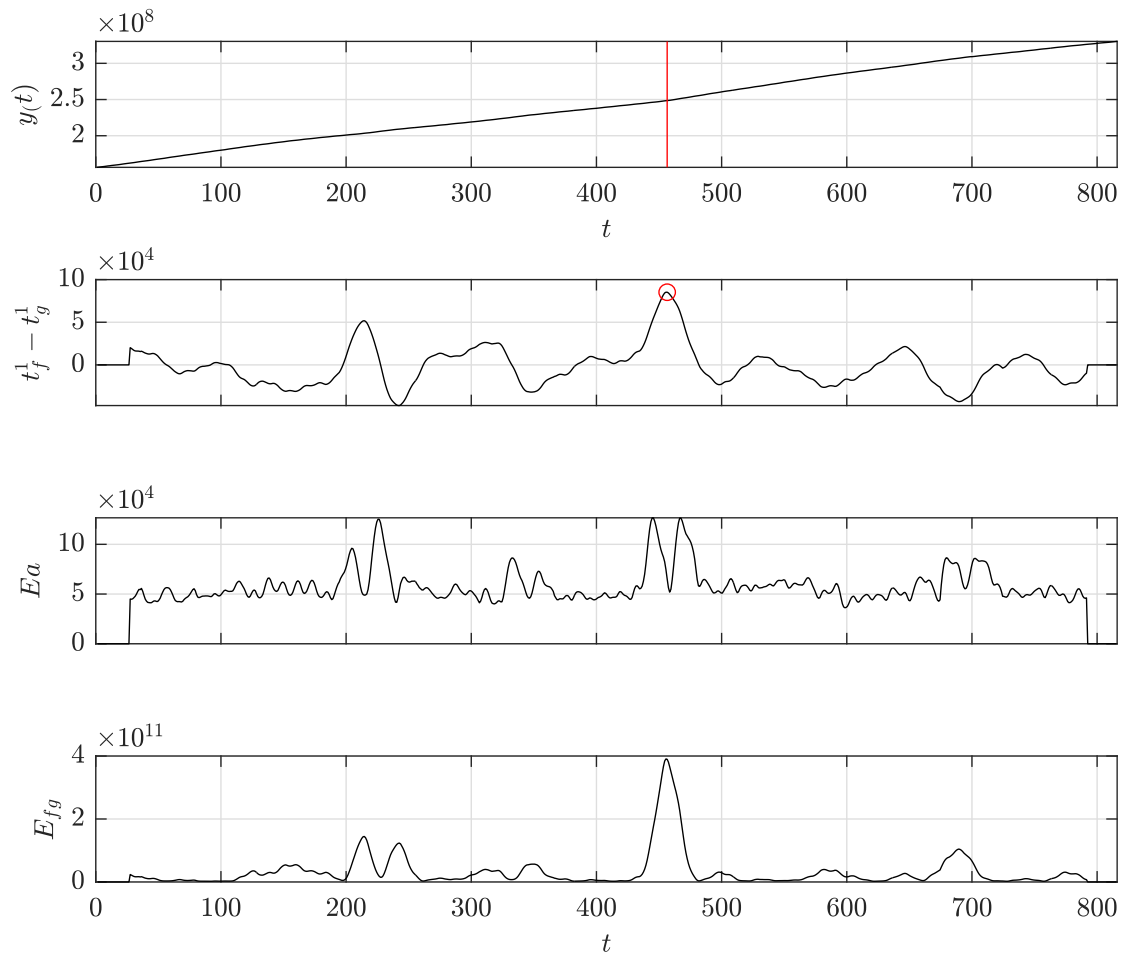


Fig. B.36: Detection of a possible slope-switch change point in the dataset "us_population" with support length $l_s = [25, 25]$ and constraint vector of $c = [0, 1]$.

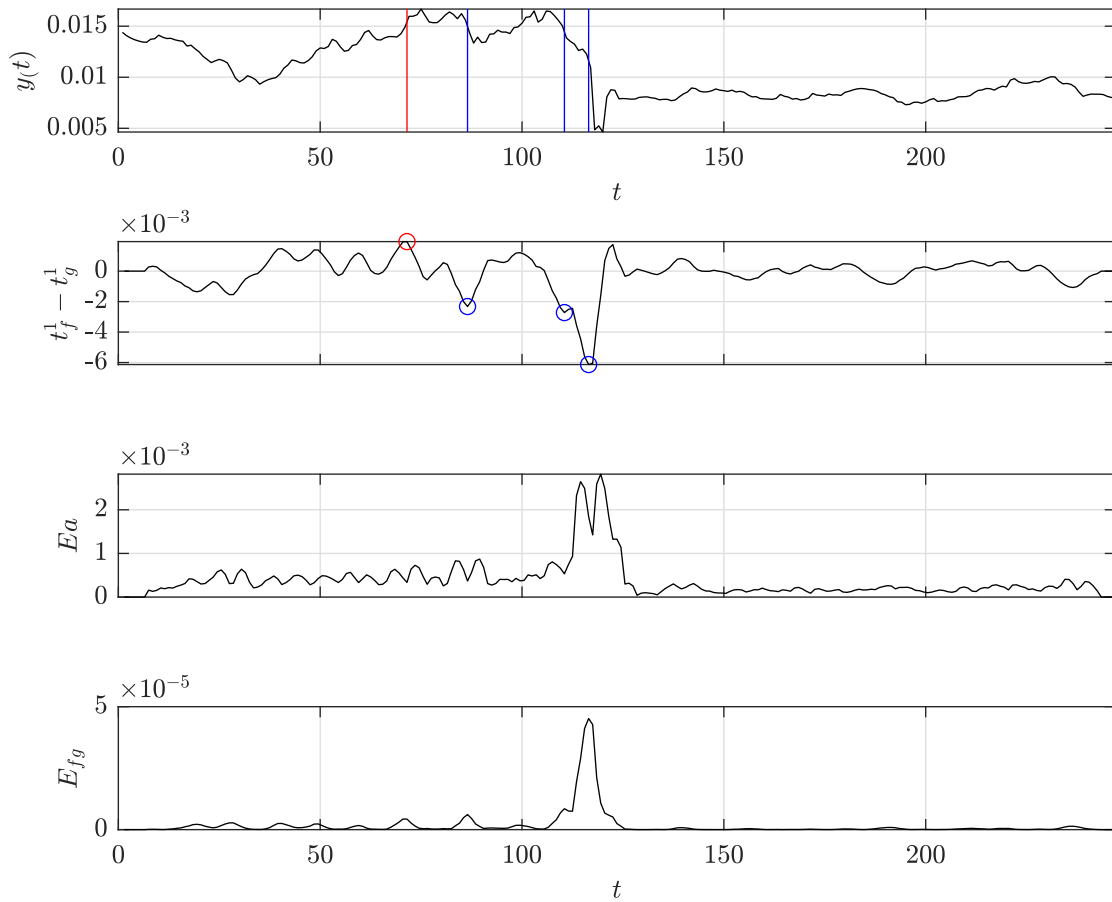


Fig. B.37: Detection of a possible mean-shift change point in the dataset "usd_isk" with support length $l_s = [5, 5]$ and constraint vector of $c = [0]$.

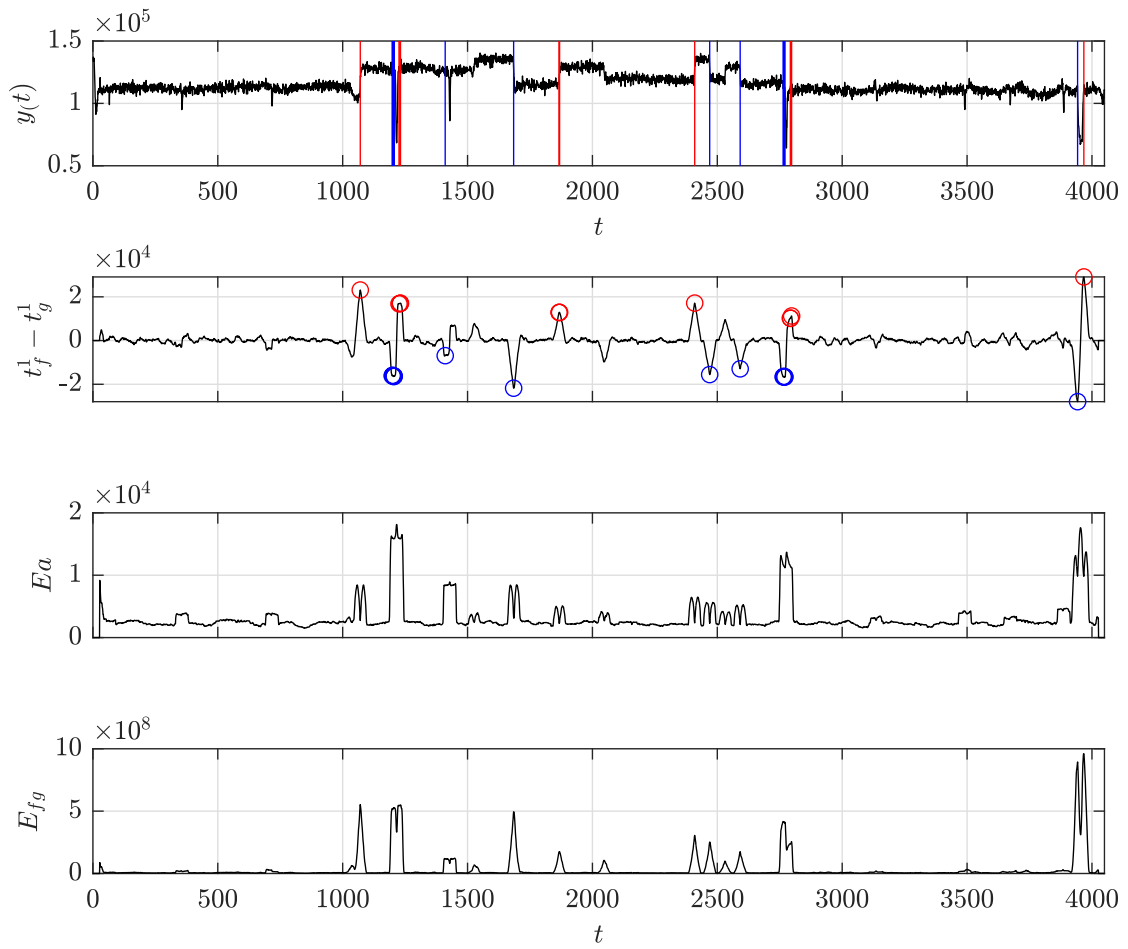


Fig. B.38: Detection of a possible mean-shift change point in the dataset "well_log" with support length $l_s = [25, 25]$ and constraint vector of $c = [0]$.

Appendix C

Slope-Switch - Discontinuity Detection Plots

In the following the evaluation results for the GCCP configuration presented in Table 6.1 are displayed.

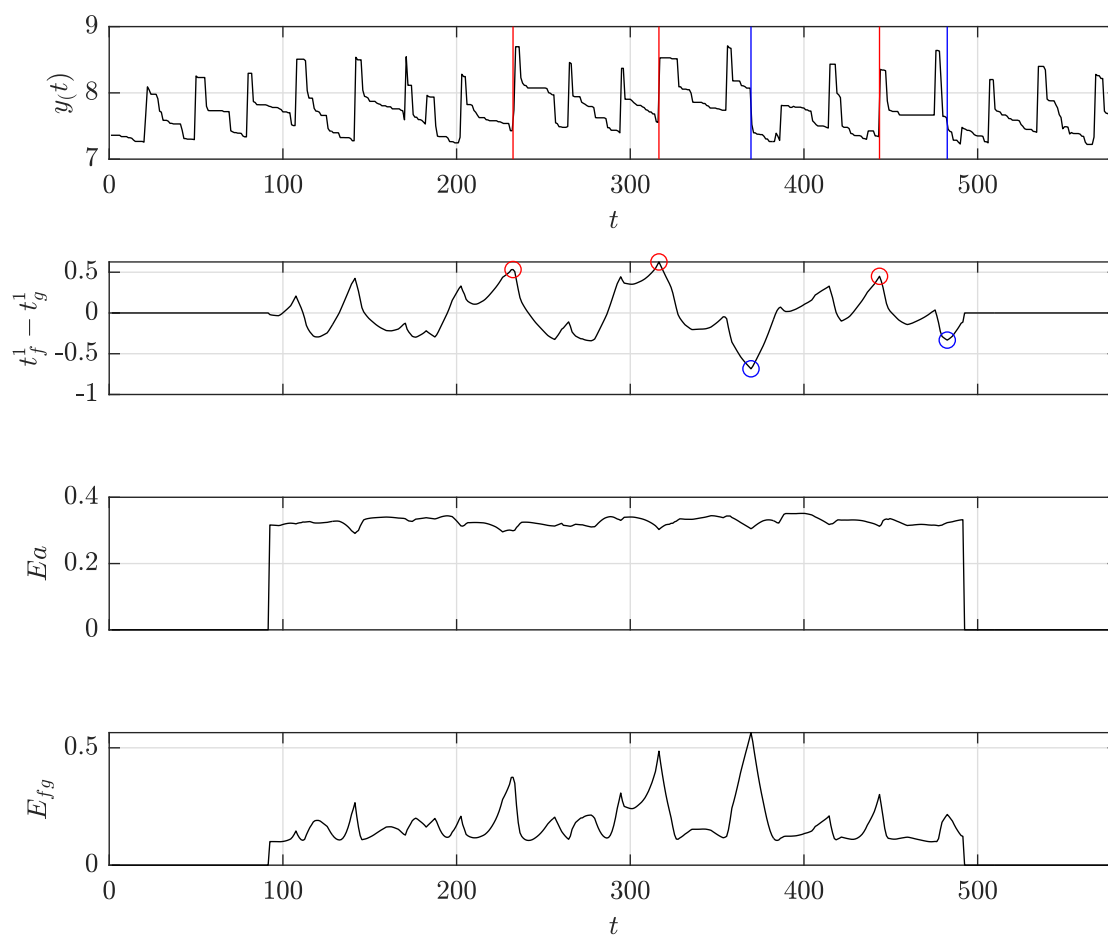


Fig. C.1: Detection of a possible mean-shift change point in the dataset "bank", by considering a present periodicity, with support length $l_s = [90, 90]$ and constraint vector of $c = [1, 0]$.

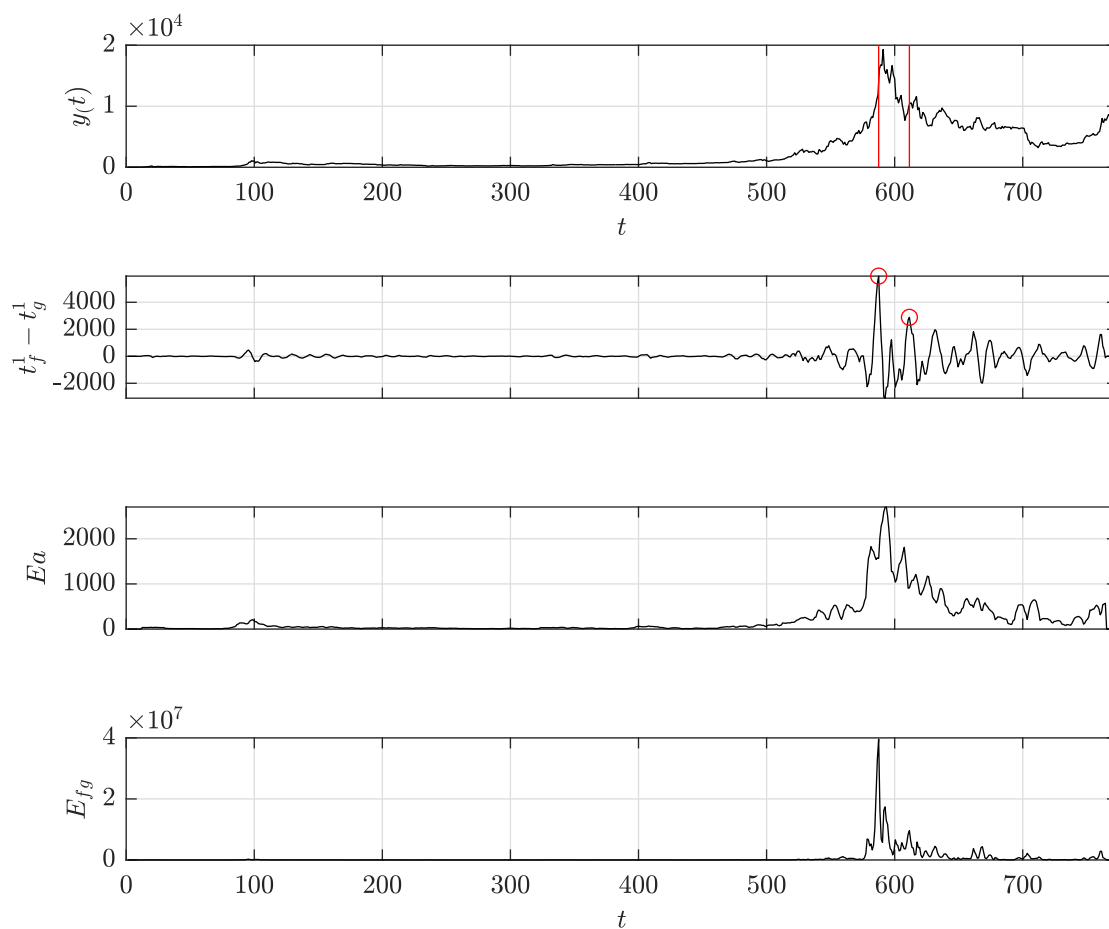


Fig. C.2: Detection of a possible mean-shift change point in the dataset "bitcoin" with support length $l_s = [10, 10]$ and constraint vector of $c = [1, 0]$.

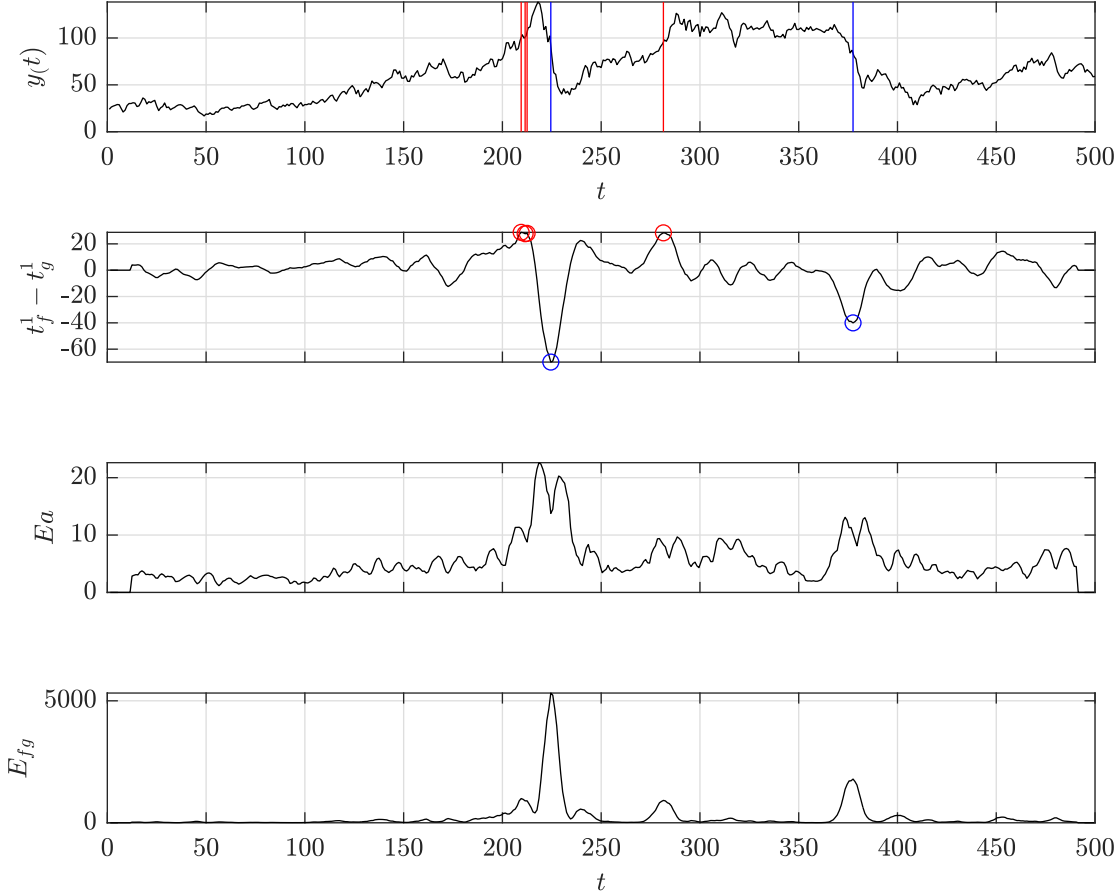


Fig. C.3: Detection of a possible mean-shift change point in the dataset "brent_spot" with support length $l_s = [20, 20]$ and constraint vector of $c = [1, 0]$.

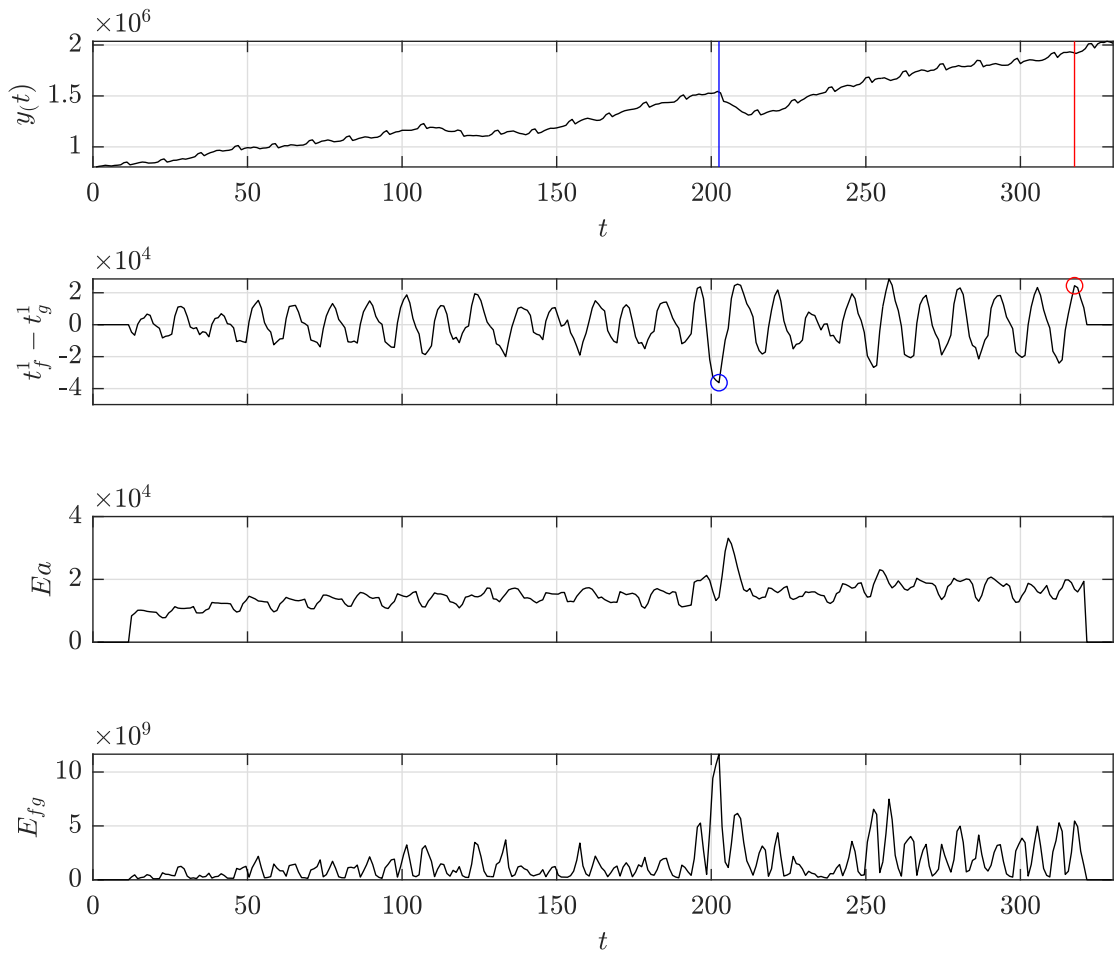


Fig. C.4: Detection of a possible slope-switch change point in the dataset "businv" with support length $l_s = [10, 10]$ and constraint vector of $c = [1, 0, 1]$.

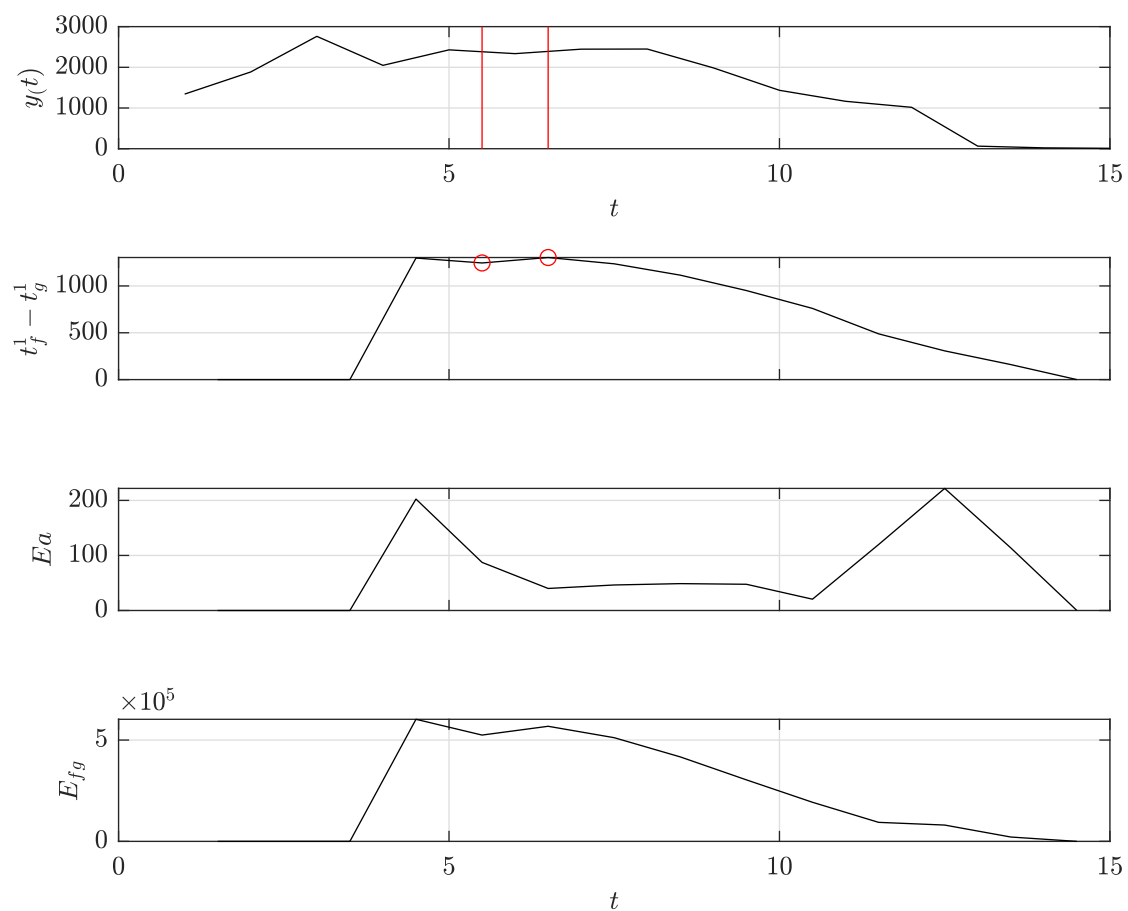


Fig. C.5: Detection of a possible slope-switch change point in the dataset "centralia" with support length $l_s = [2, 2]$ and constraint vector of $c = [1, 0, 1]$.

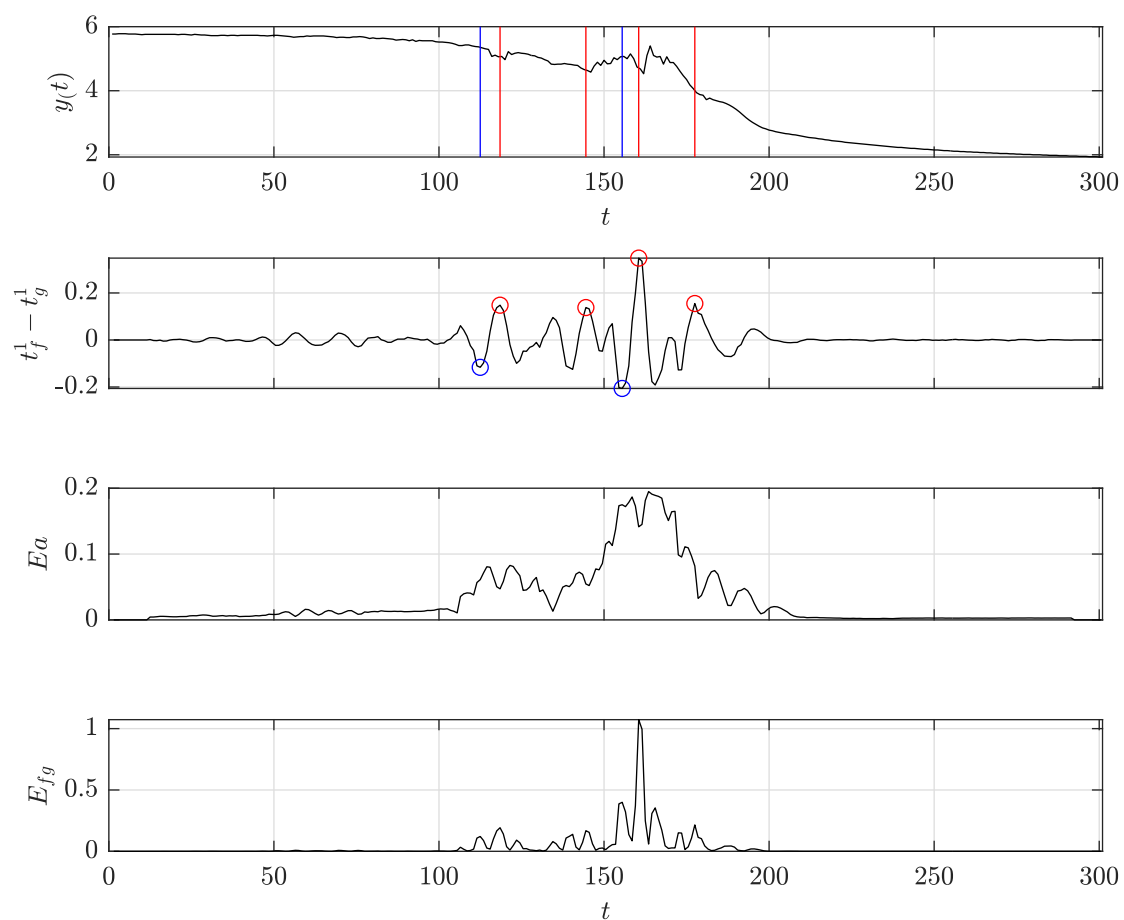


Fig. C.6: Detection of a possible slope-switch change point in the dataset "children_per_woman" with support length $l_s = [10, 10]$ and constraint vector of $c = [1, 0, 1]$.

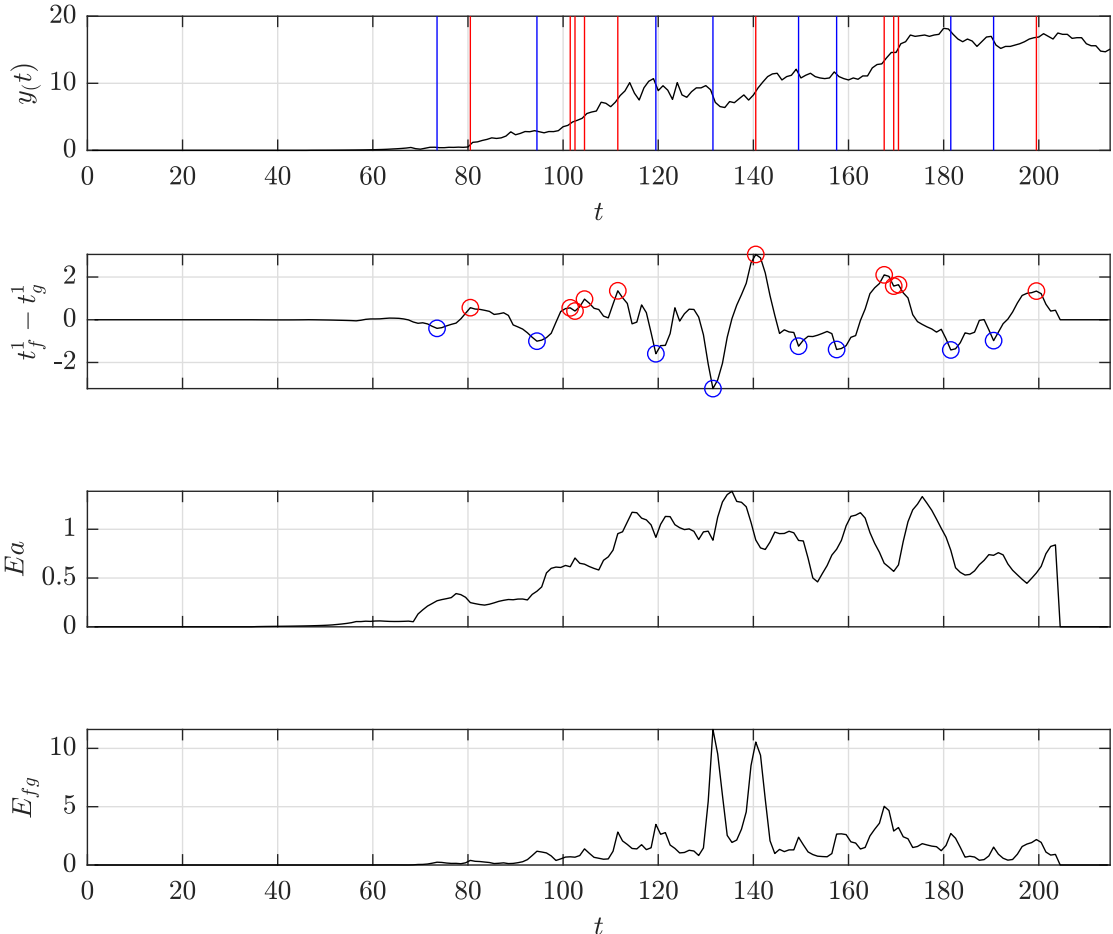


Fig. C.7: Detection of a possible mean-shift change point in the dataset "co2.canada" with support length $l_s = [12, 12]$ and constraint vector of $c = [1, 0]$.

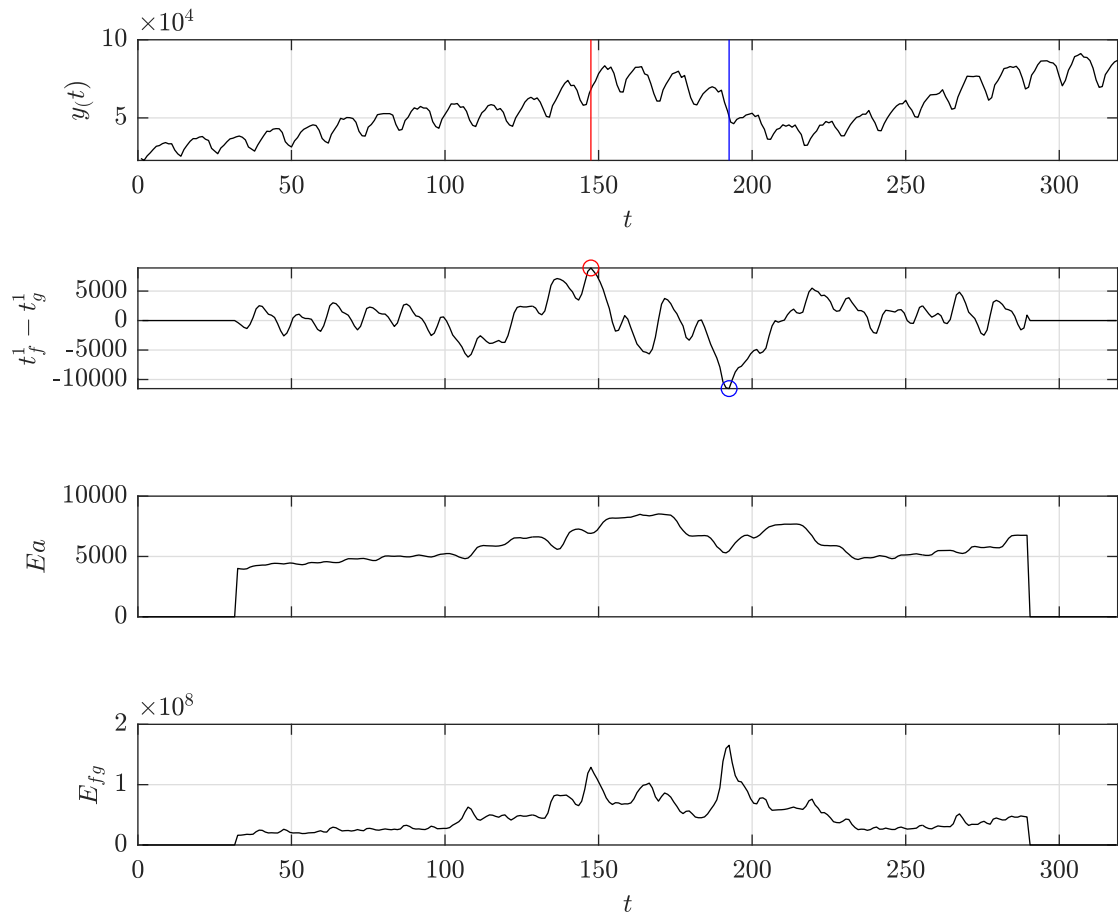


Fig. C.8: Detection of a possible mean-shift change point in the dataset "construction" with support length $l_s = [30, 30]$ and constraint vector of $c = [1, 0]$.

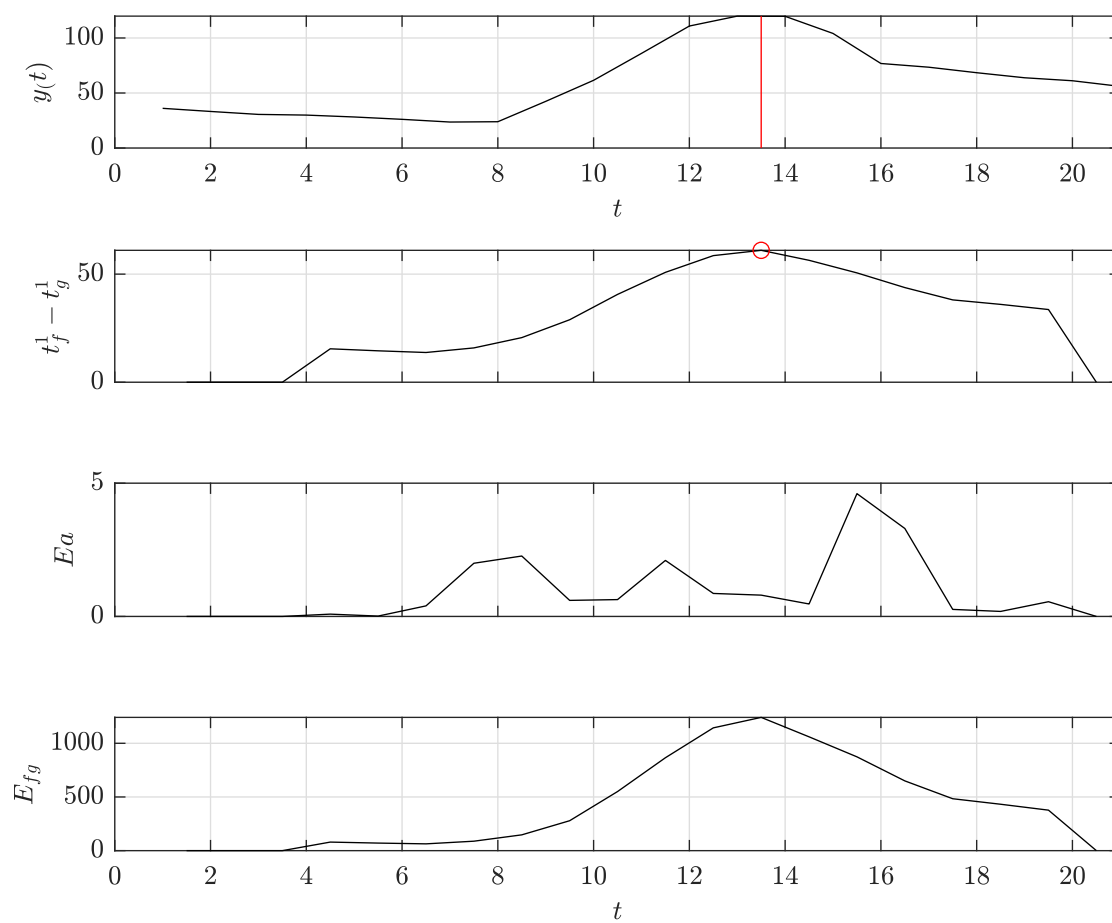


Fig. C.9: Detection of a possible slope-switch change point in the dataset "debt_ireland" with support length $l_s = [2, 2]$ and constraint vector of $c = [1, 0, 1]$.

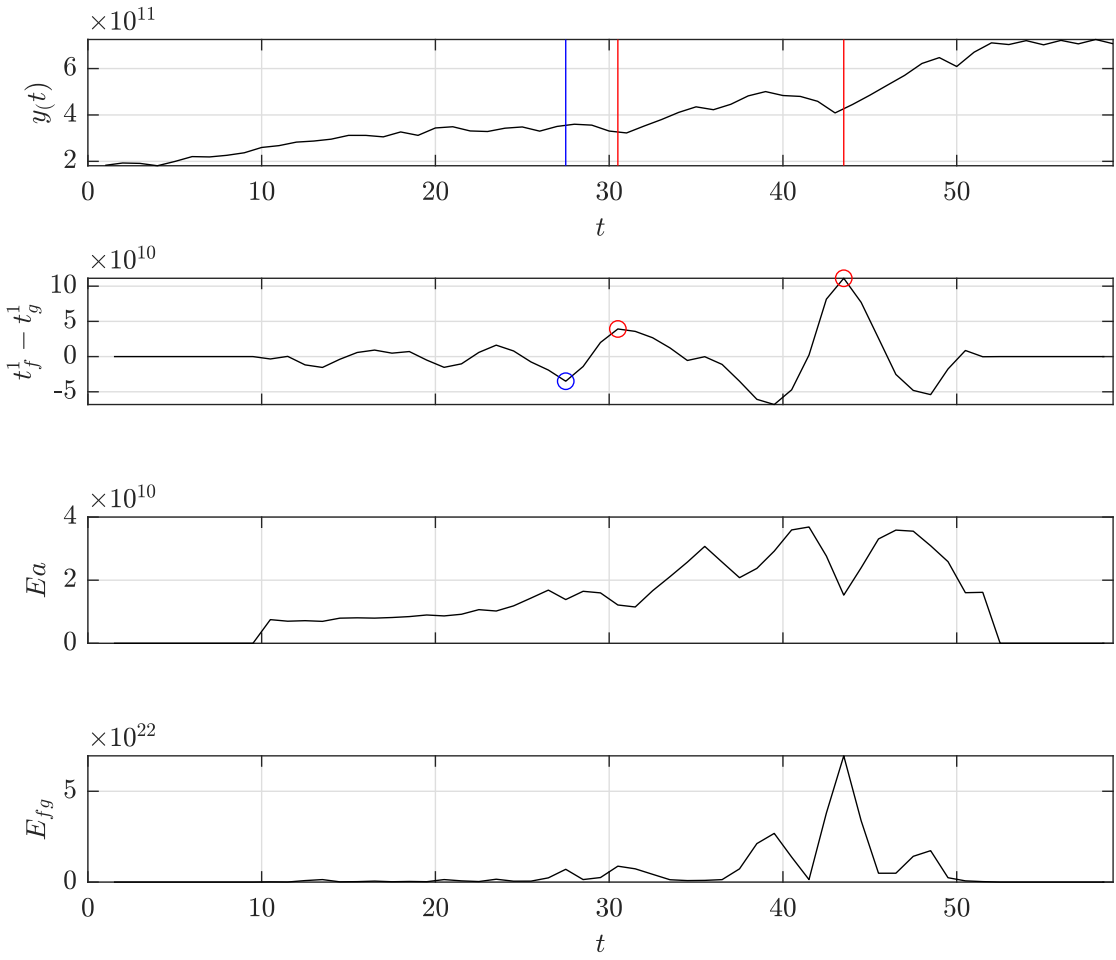


Fig. C.10: Detection of a possible slope-switch change point in the dataset "gdp_argentina" with support length $l_s = [8, 8]$ and constraint vector of $c = [1, 0, 1]$.

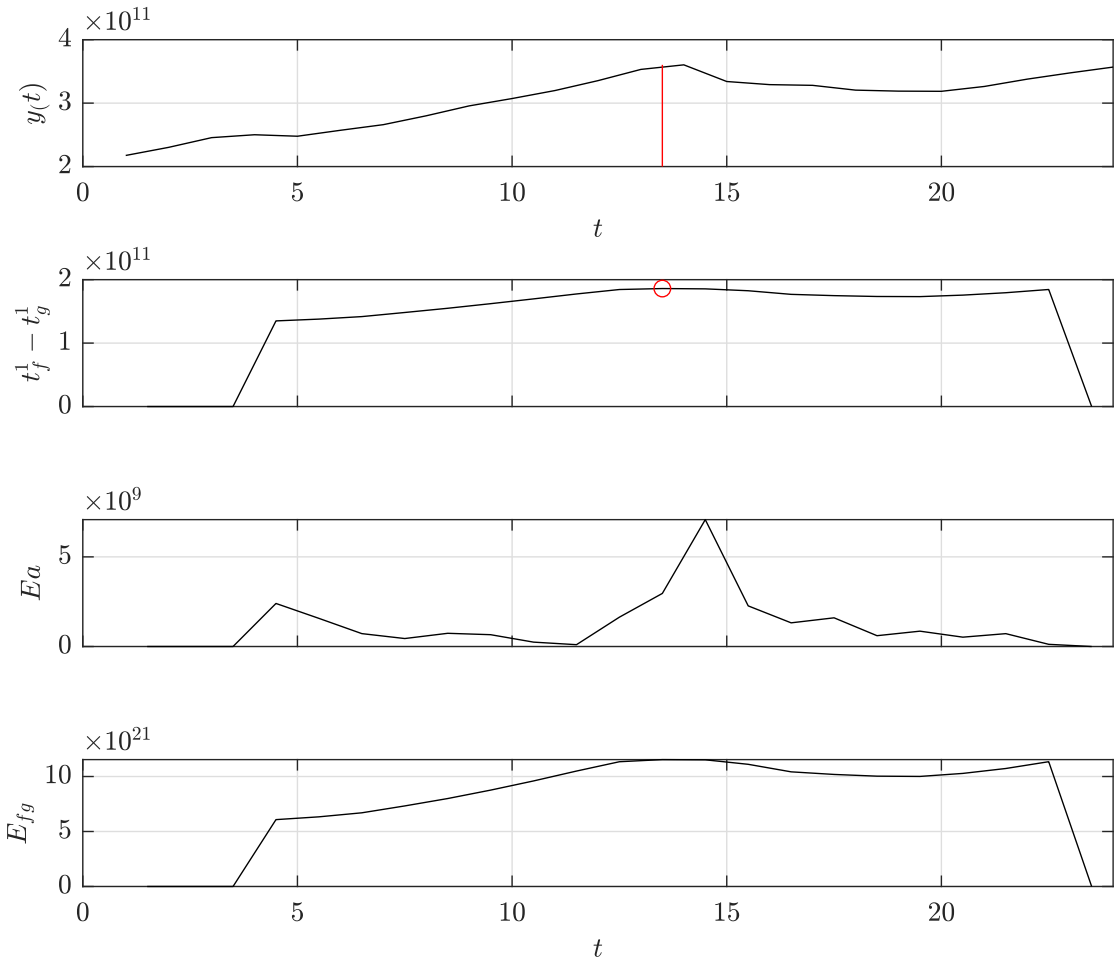


Fig. C.11: Detection of a possible slope-switch change point in the dataset "gdp_croatia" with support length $l_s = [2, 2]$ and constraint vector of $c = [1, 0, 1]$.

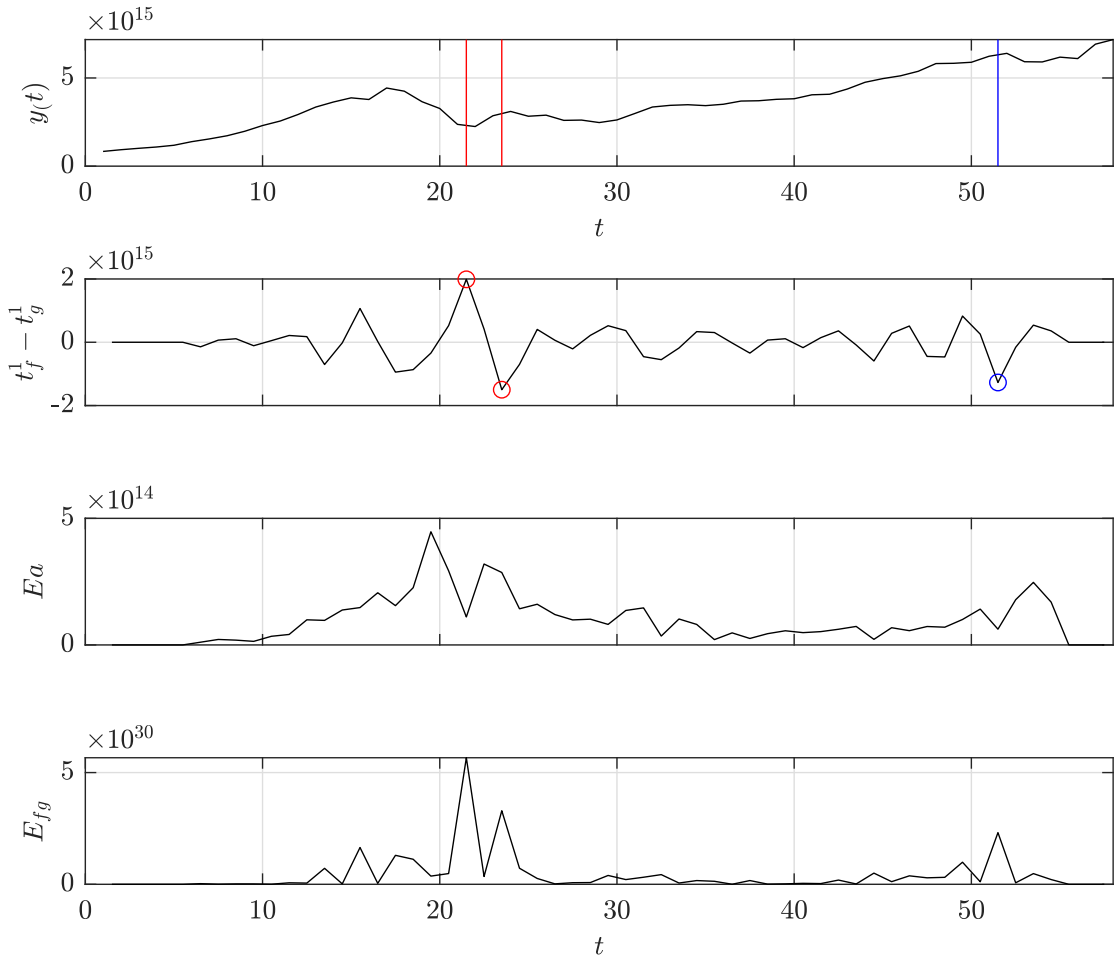


Fig. C.12: Detection of a possible slope-switch change point in the dataset "gdp_iran" with support length $l_s = [4, 4]$ and constraint vector of $c = [1, 0, 1]$.

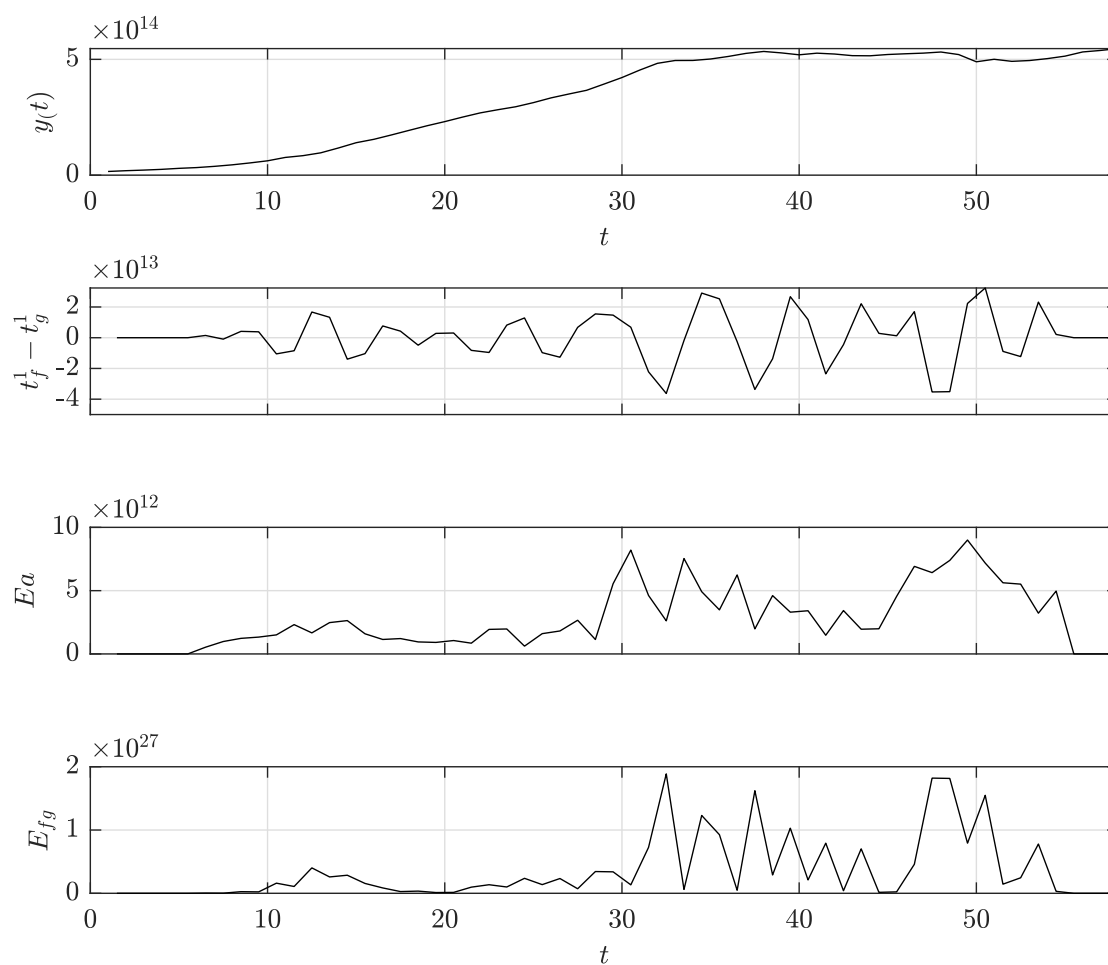


Fig. C.13: Detection of a possible slope-switch change point in the dataset "gdp_japan" with support length $l_s = [4, 4]$ and constraint vector of $c = [1, 0, 1]$.

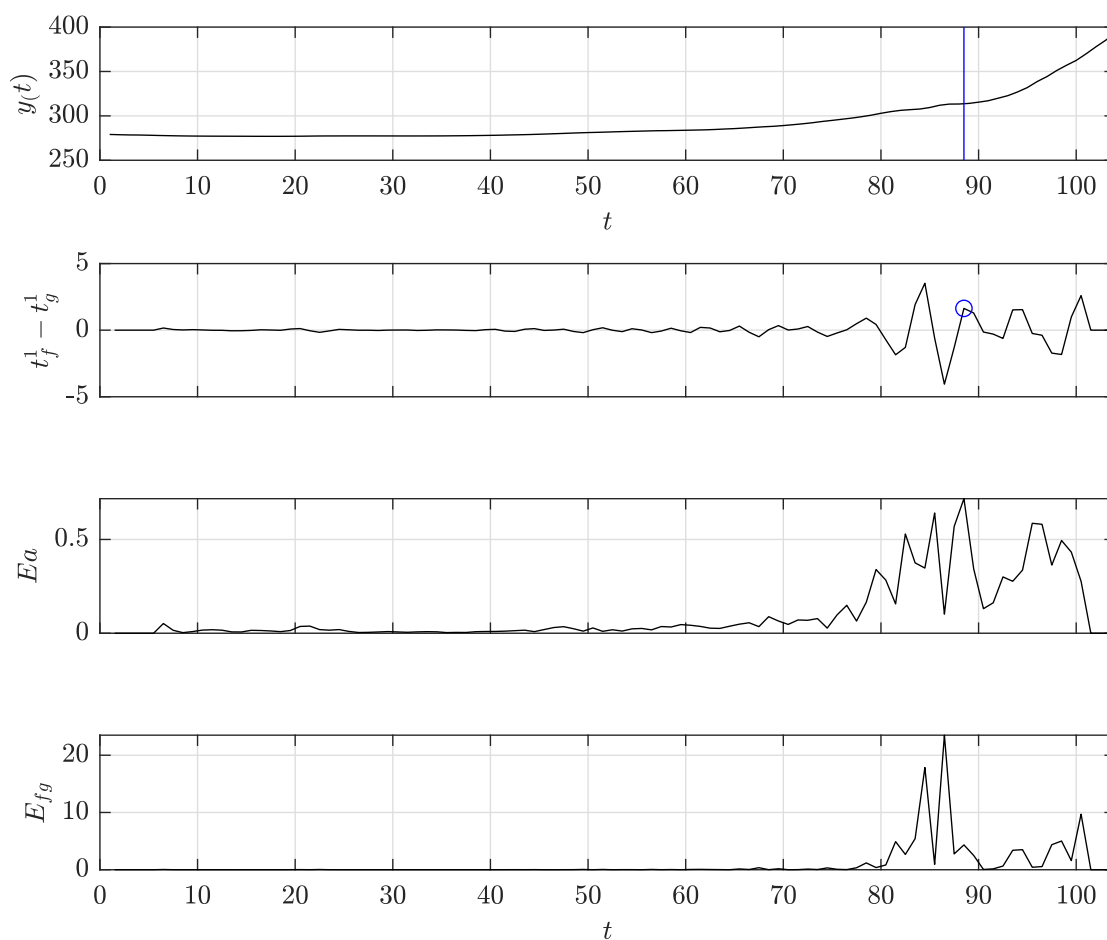


Fig. C.14: Detection of a possible slope-switch change point in the dataset "global_co2" with support length $l_s = [4, 4]$ and constraint vector of $c = [1, 0, 1]$.

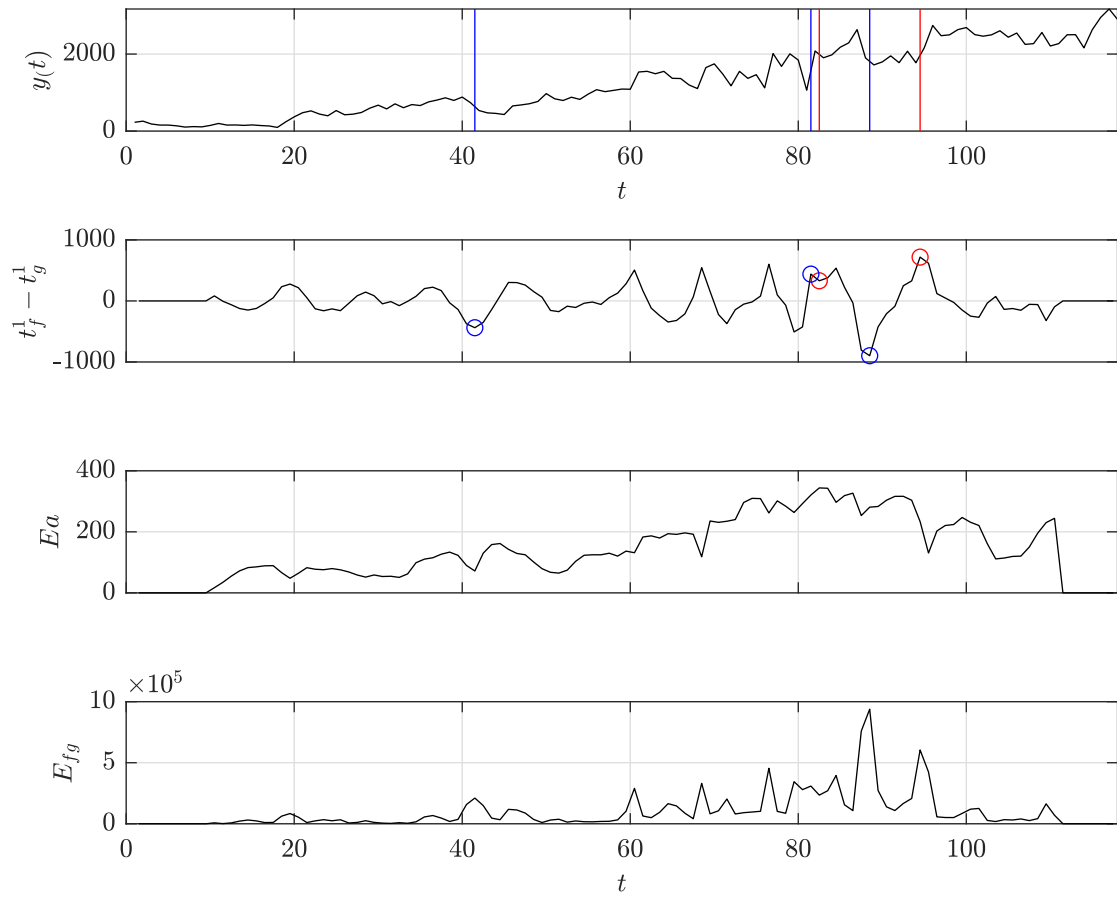


Fig. C.15: Detection of a possible mean-shift change point in the dataset "homeruns" with support length $l_s = [8, 8]$ and constraint vector of $c = [1, 0]$.

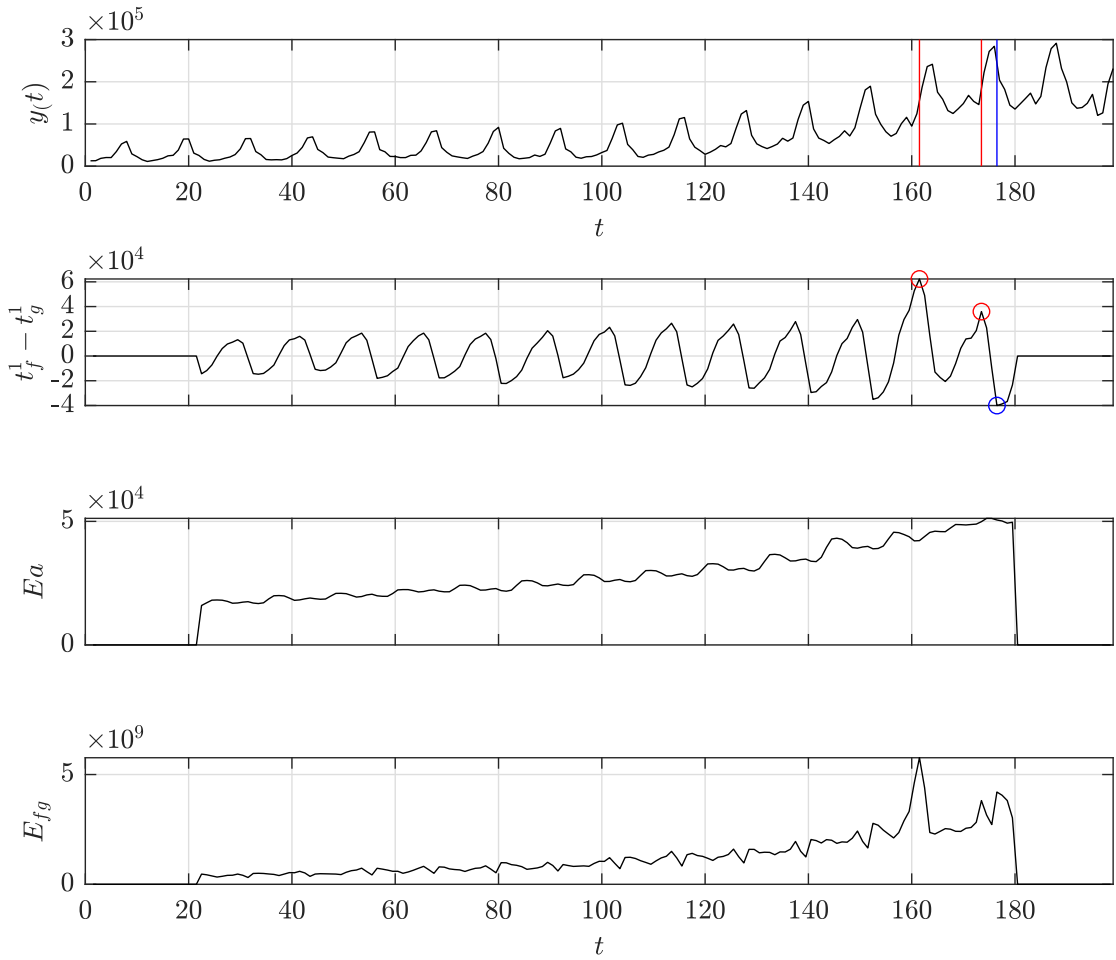


Fig. C.16: Detection of a possible mean-shift change point in the dataset "iceland_tourism" with support length $l_s = [20, 20]$ and constraint vector of $c = [1, 0]$.

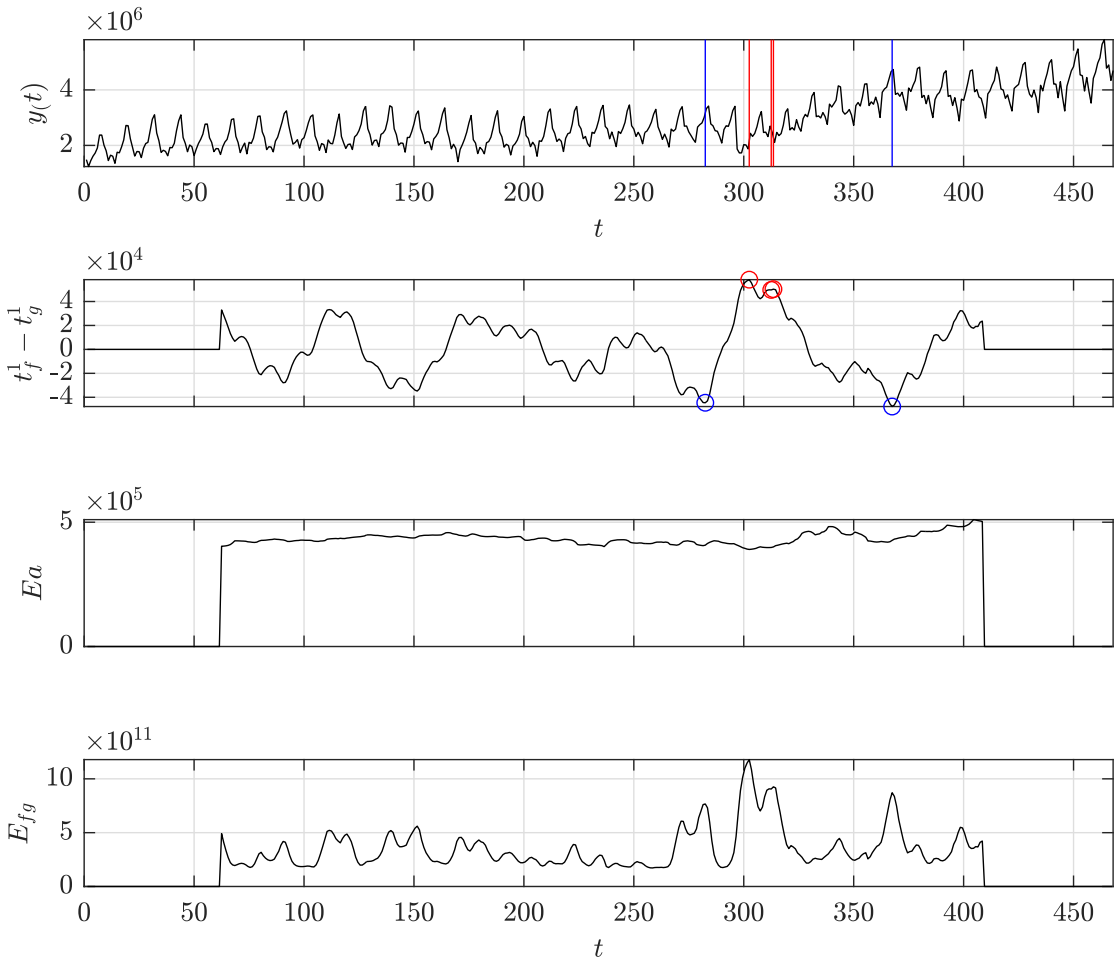


Fig. C.17: Detection of a possible slope-switch change point in the dataset "jfk_passengers" with support length $l_s = [60, 60]$ and constraint vector of $c = [1, 0, 1]$.

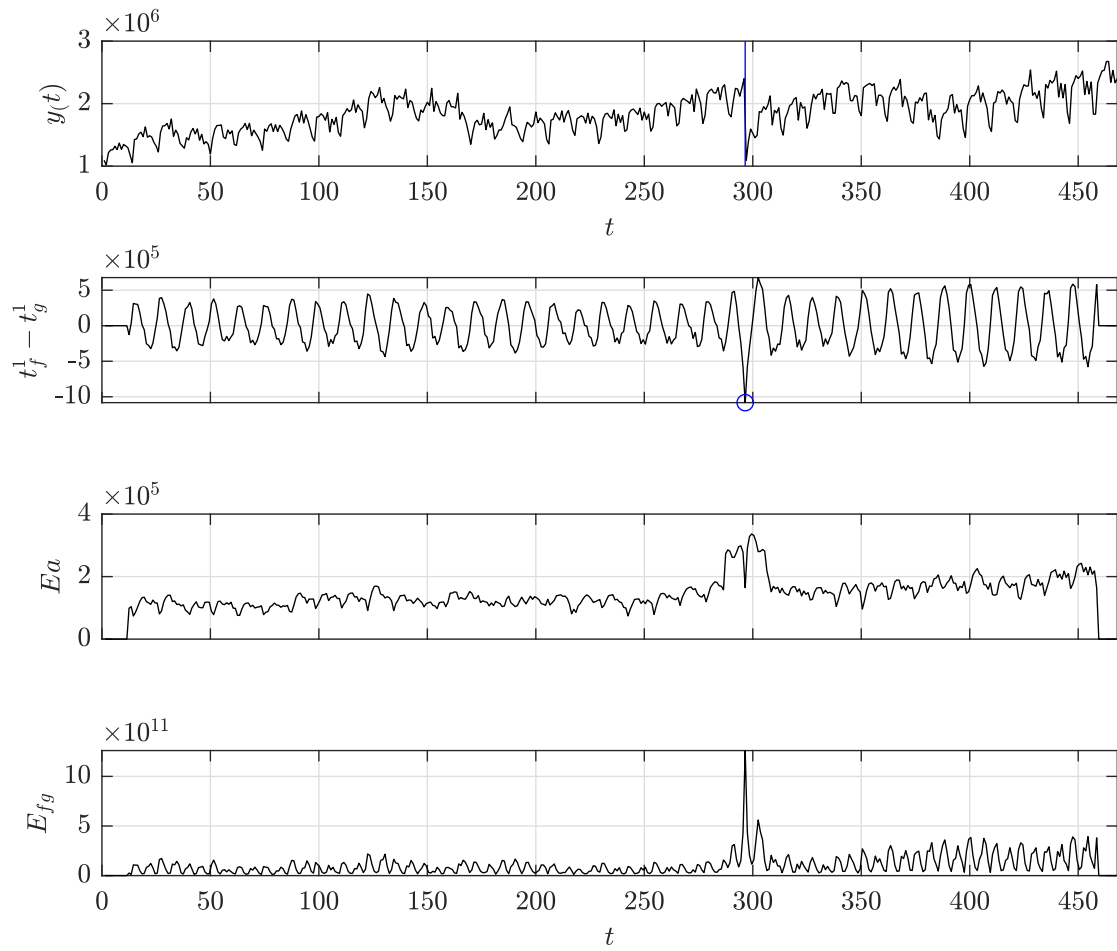


Fig. C.18: Detection of a possible mean-shift change point in the dataset "lga_passengers" with support length $l_s = [10, 10]$ and constraint vector of $c = [1, 0]$.

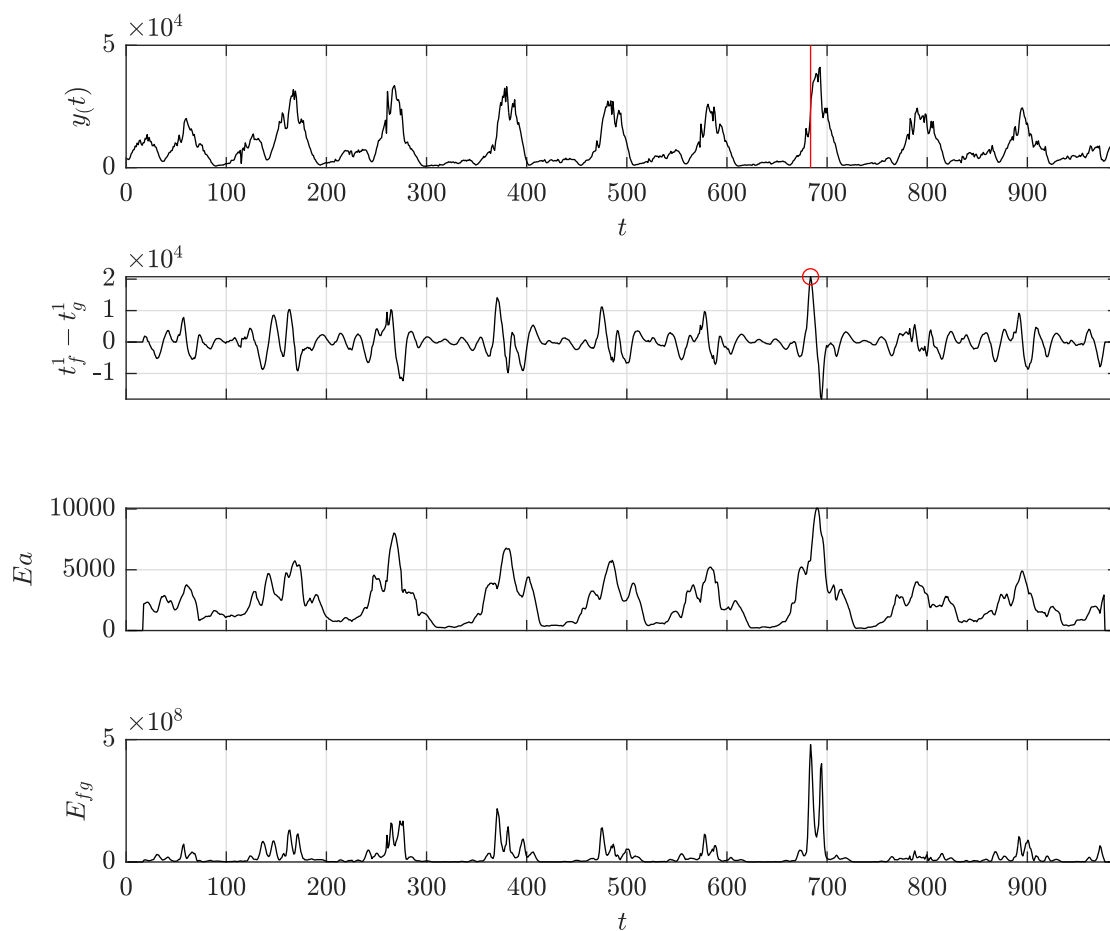


Fig. C.19: Detection of a possible mean-shift change point in the dataset "measles" with support length $l_s = [15, 15]$ and constraint vector of $c = [1, 0]$.

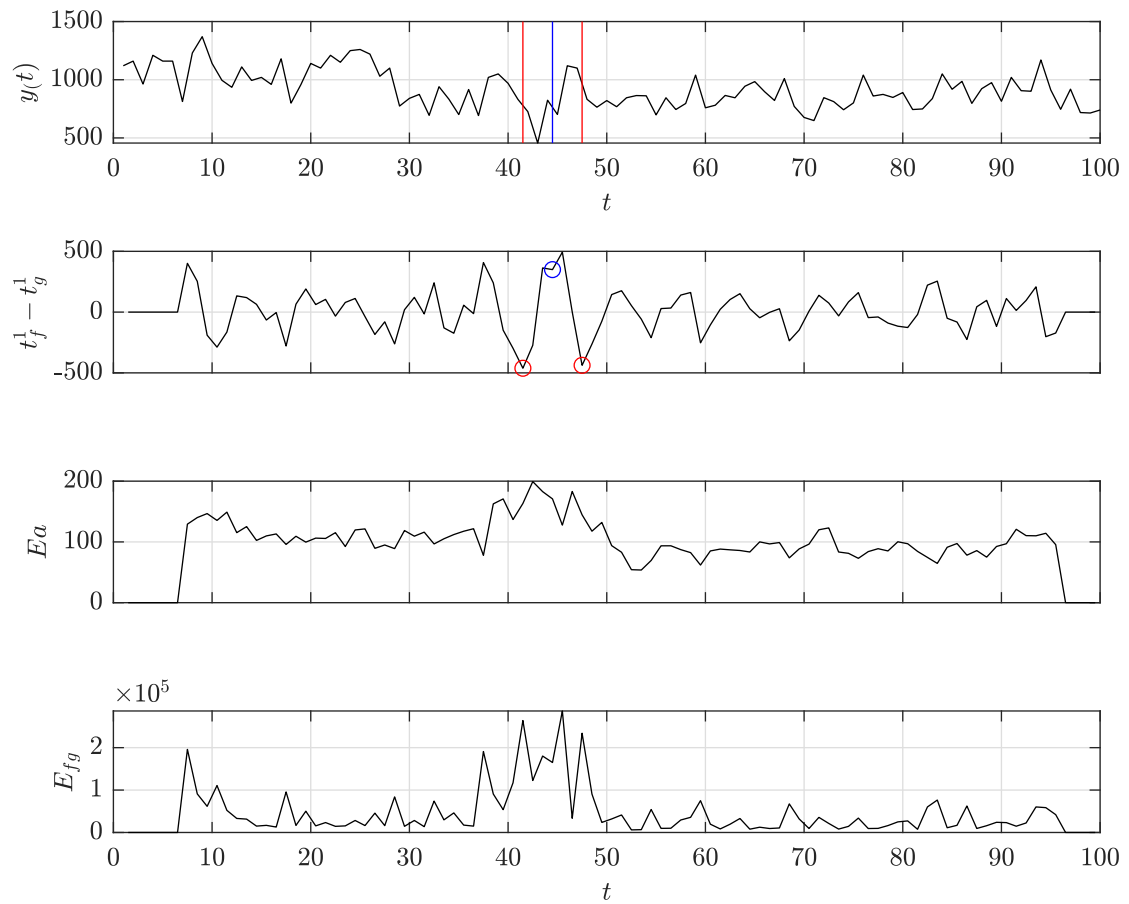


Fig. C.20: Detection of a possible mean-shift change point in the dataset "nile" with support length $l_s = [5, 5]$ and constraint vector of $c = [1, 0]$.

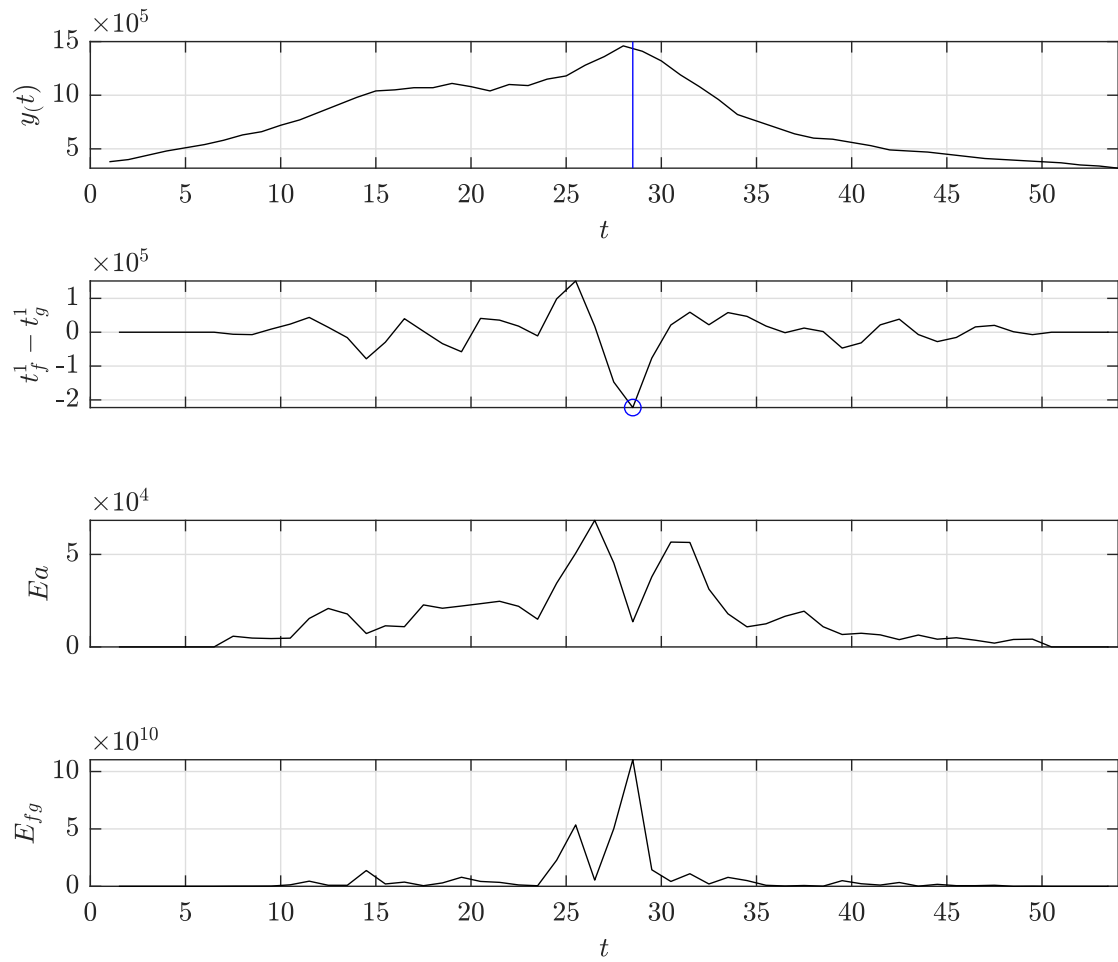


Fig. C.21: Detection of a possible slope-switch change point in the dataset "ozone" with support length $l_s = [5, 5]$ and constraint vector of $c = [1, 0, 1]$.

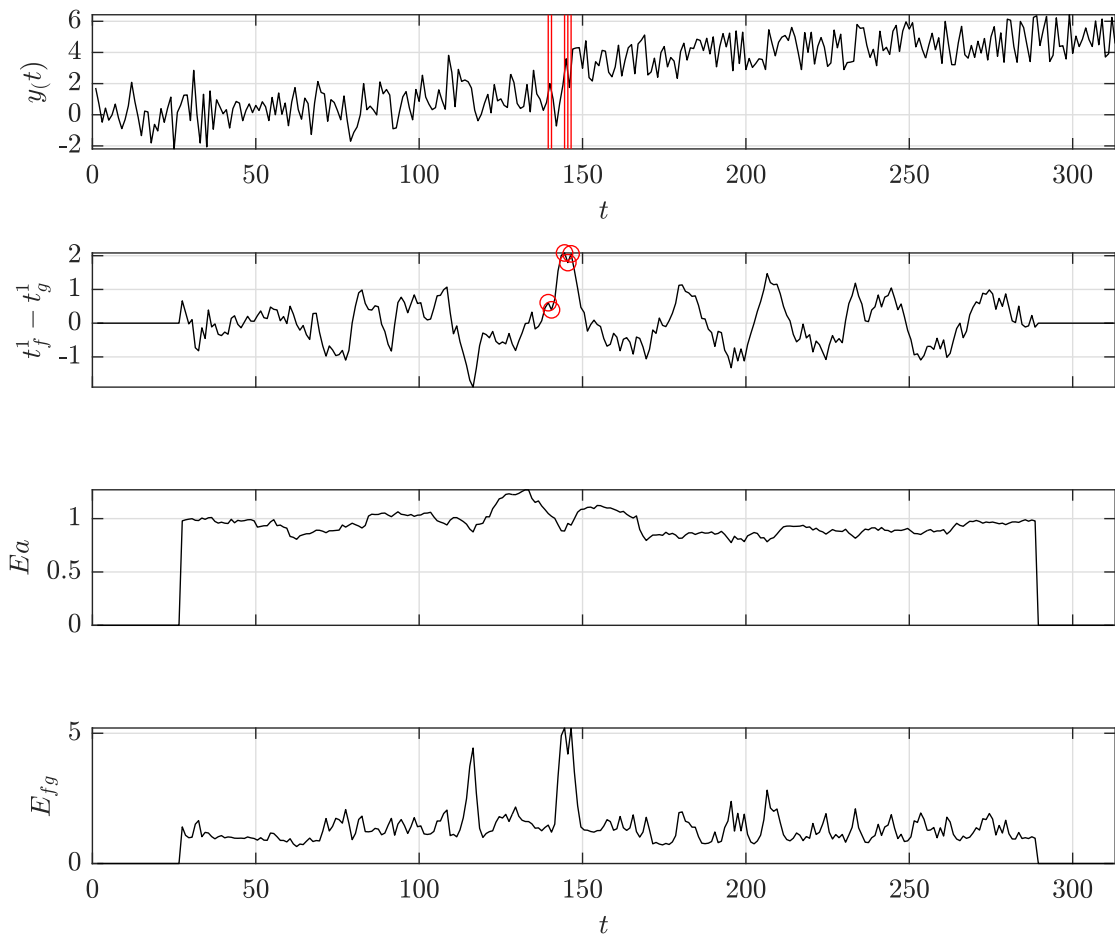


Fig. C.22: Detection of a possible mean-shift change point in the dataset "quality_control_1" with support length $l_s = [25, 25]$ and constraint vector of $c = [1, 0]$.

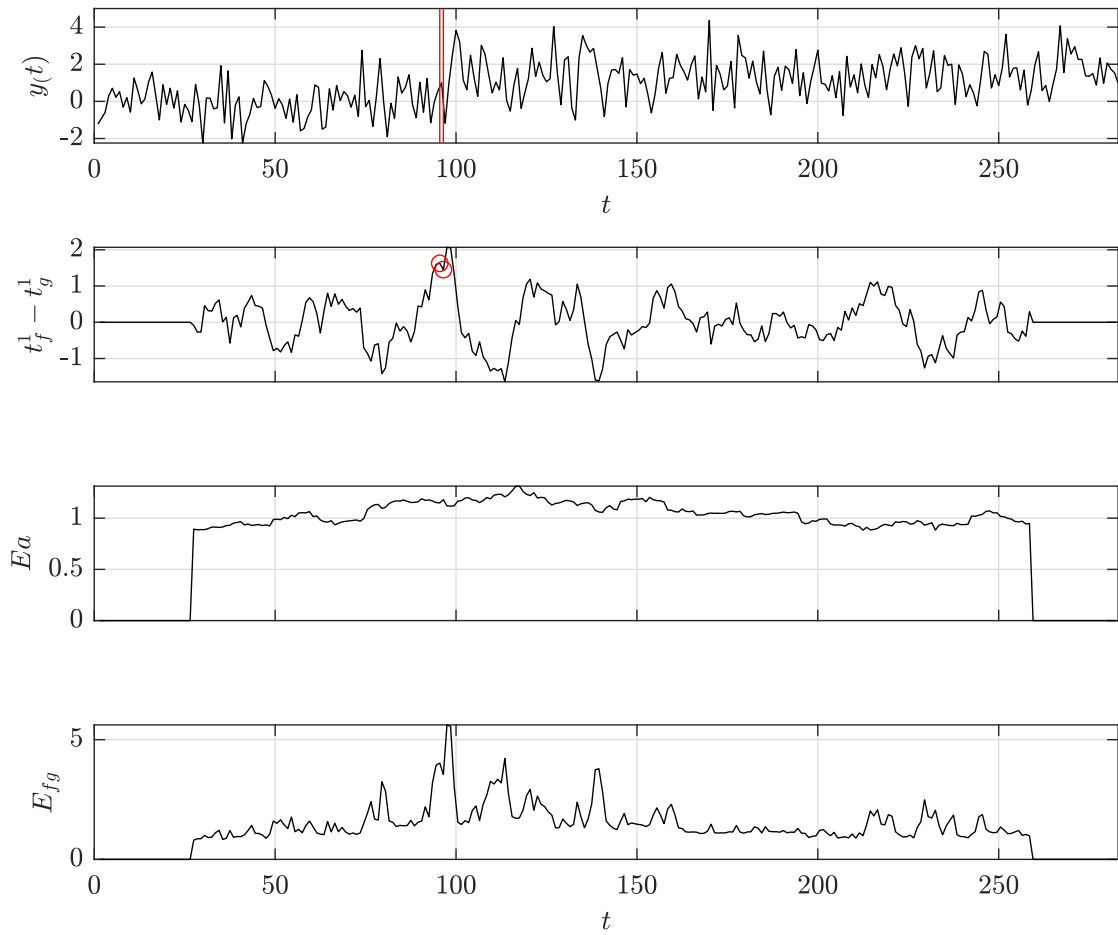


Fig. C.23: Detection of a possible mean-shift change point in the dataset "quality_control.2" with support length $l_s = [25, 25]$ and constraint vector of $c = [1, 0]$.

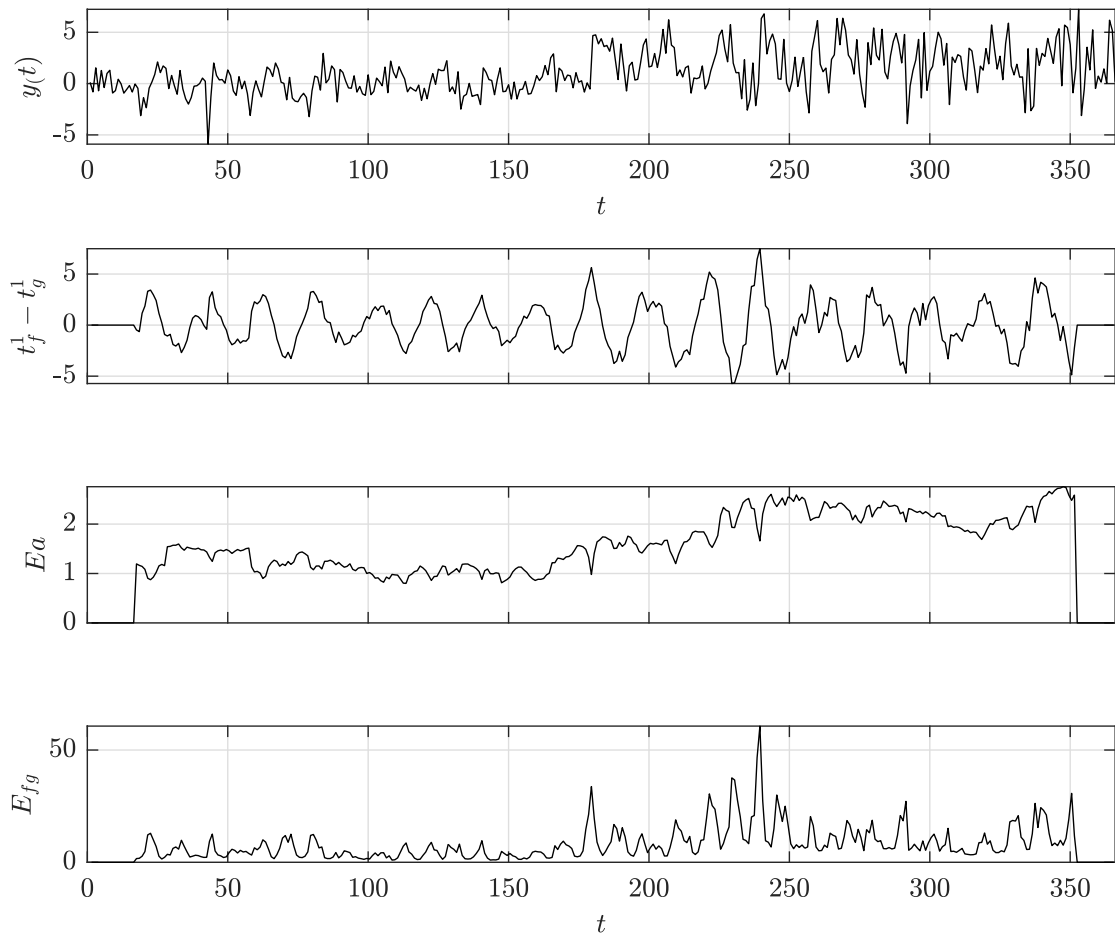


Fig. C.24: Detection of a possible mean-shift change point in the dataset "quality_control_3" with support length $l_s = [15, 15]$ and constraint vector of $c = [1, 0]$.

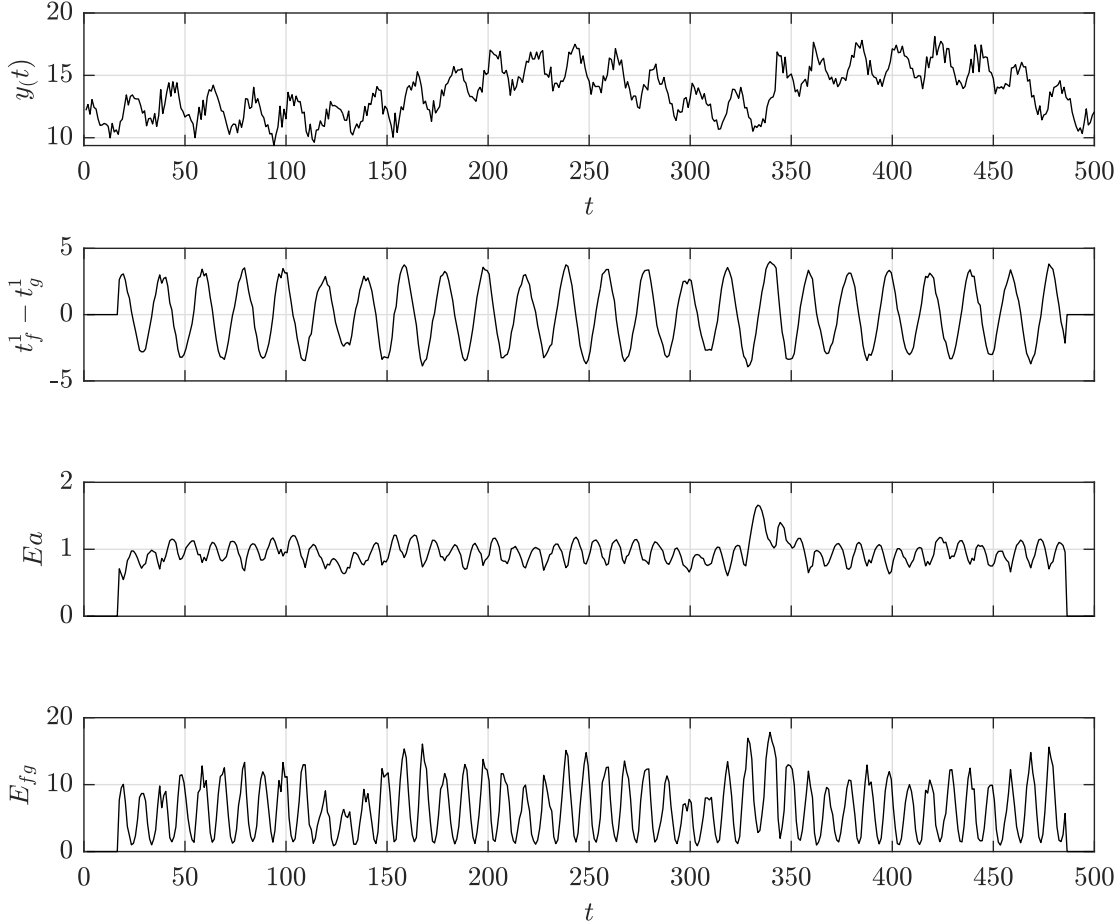


Fig. C.25: Detection of a possible mean-shift change point in the dataset "quality_control.4" with support length $l_s = [15, 15]$ and constraint vector of $c = [1, 0]$.

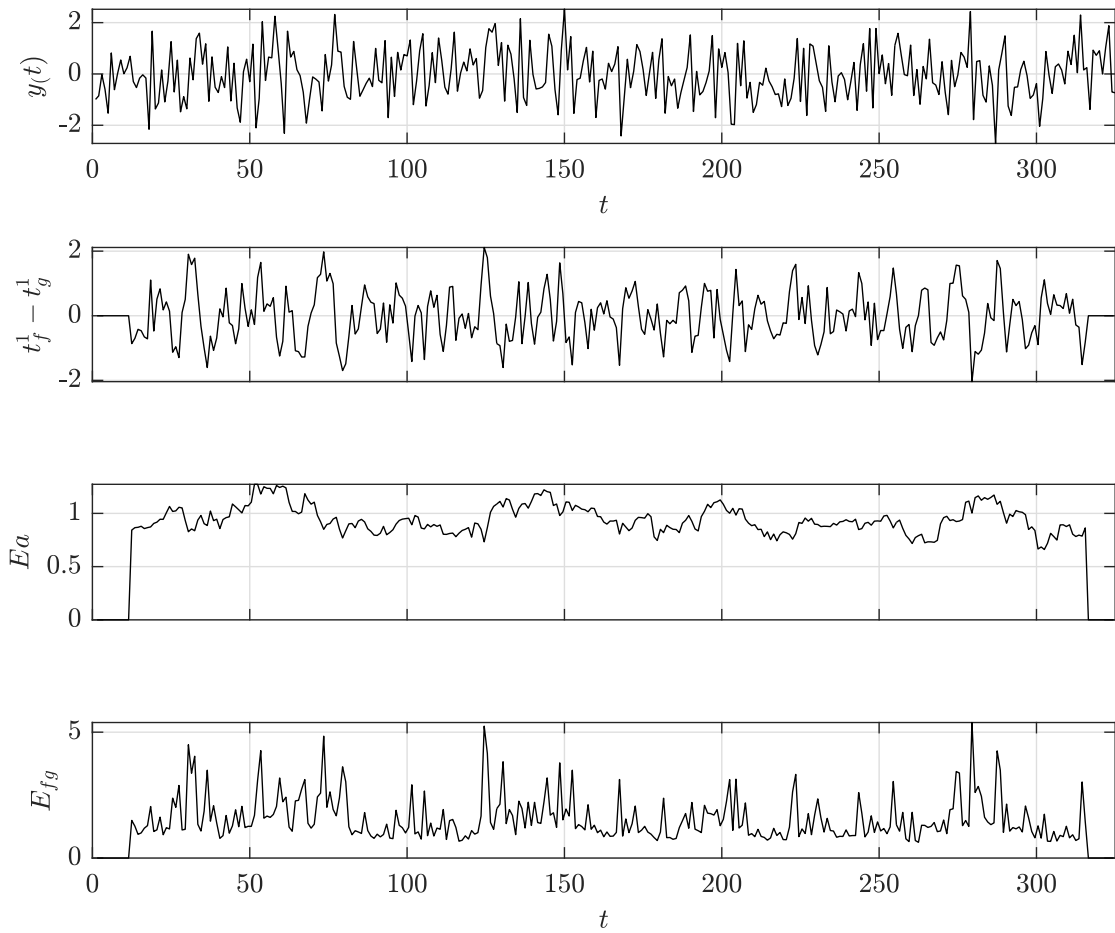


Fig. C.26: Detection of a possible mean-shift change point in the dataset "quality_control.5" with support length $l_s = [10, 10]$ and constraint vector of $c = [1, 0]$.

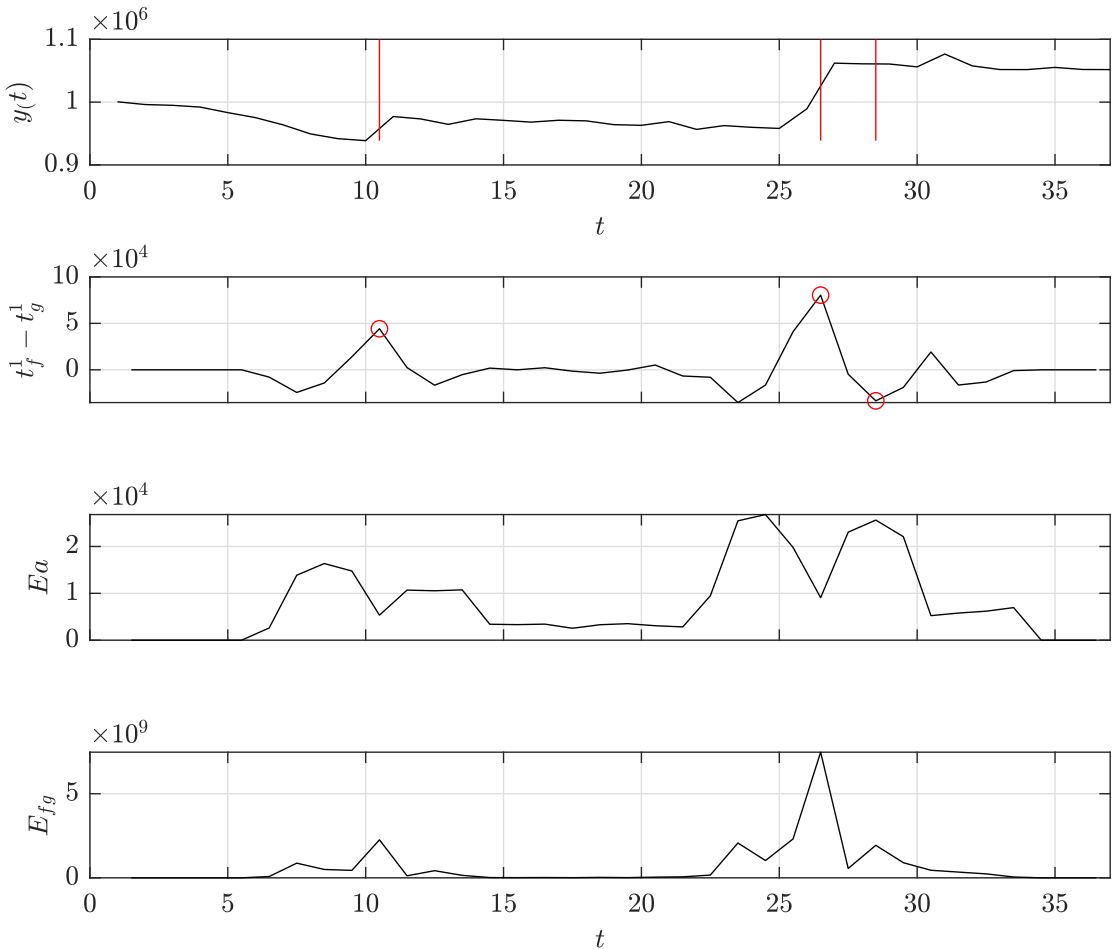


Fig. C.27: Detection of a possible mean-shift change point in the dataset "rail_lines" with support length $l_s = [4, 4]$ and constraint vector of $c = [1, 0]$.

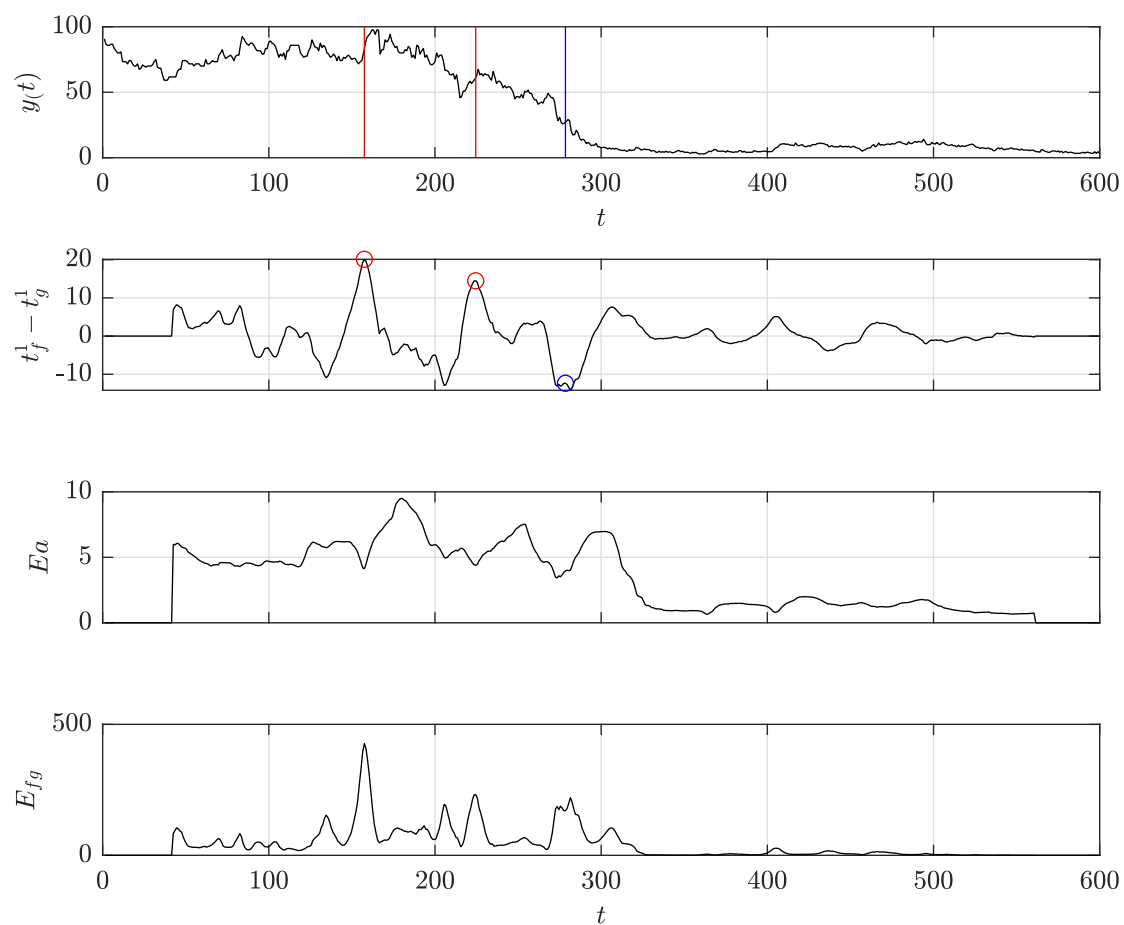


Fig. C.28: Detection of a possible mean-shift change point in the dataset "ratner_stock" with support length $l_s = [40, 40]$ and constraint vector of $c = [1, 0]$.

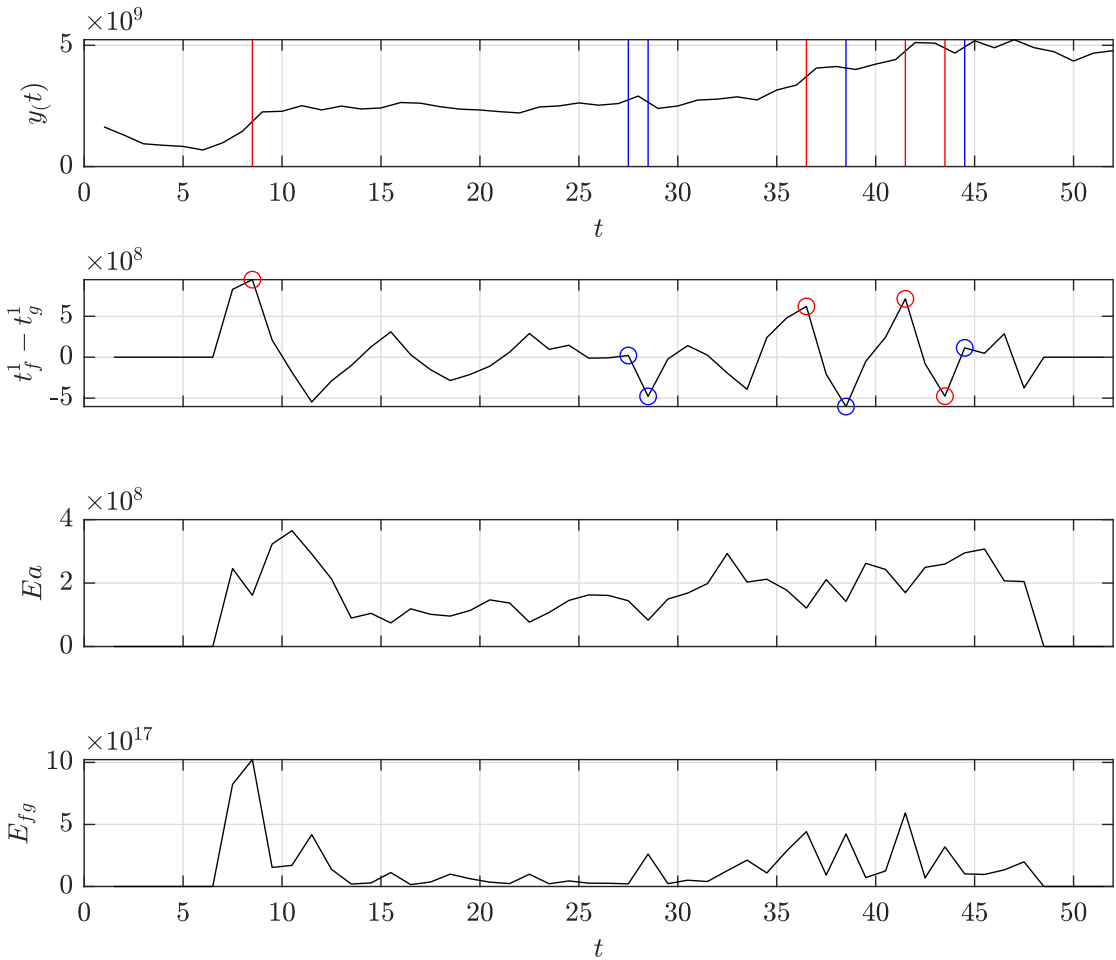


Fig. C.29: Detection of a possible mean-shift change point in the dataset "robocalls" with support length $l_s = [5, 5]$ and constraint vector of $c = [1, 0]$.

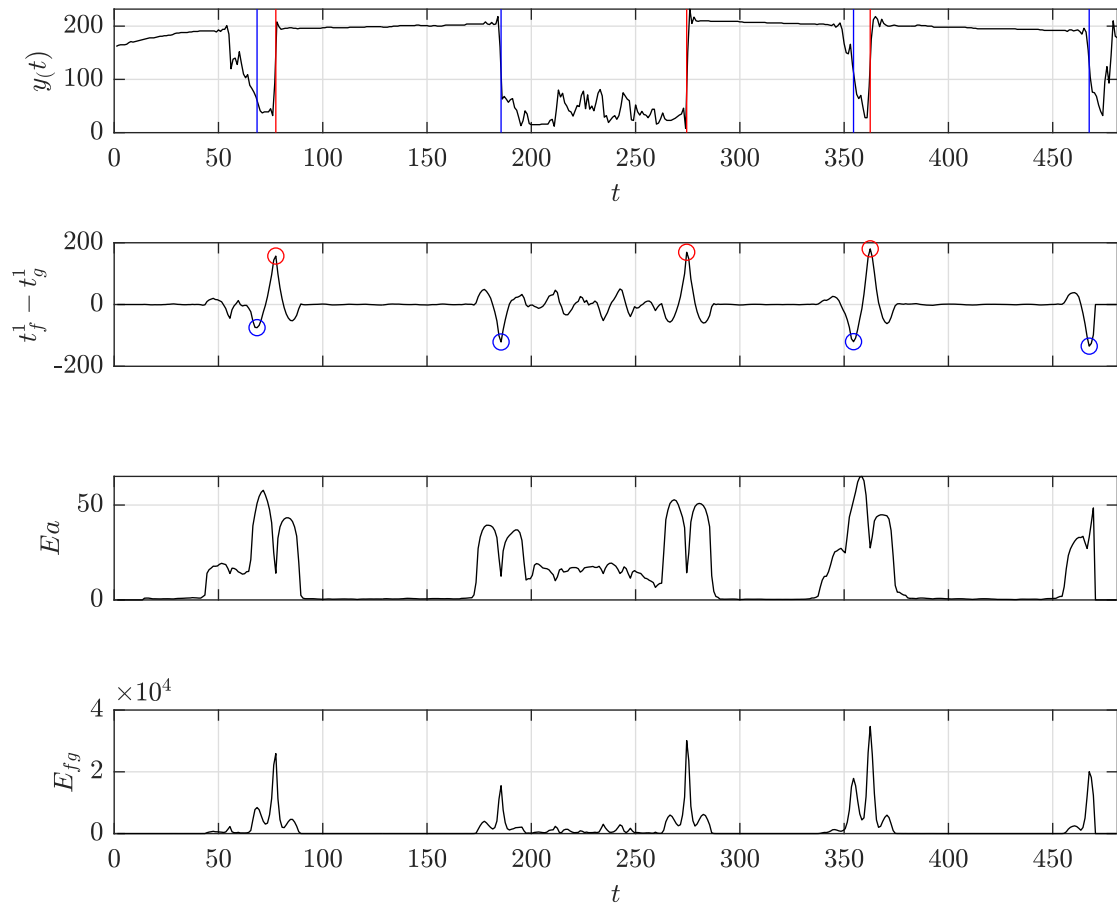


Fig. C.30: Detection of a possible mean-shift change point in the dataset "scanline_420491" with support length $l_s = [12, 12]$ and constraint vector of $c = [1, 0]$.

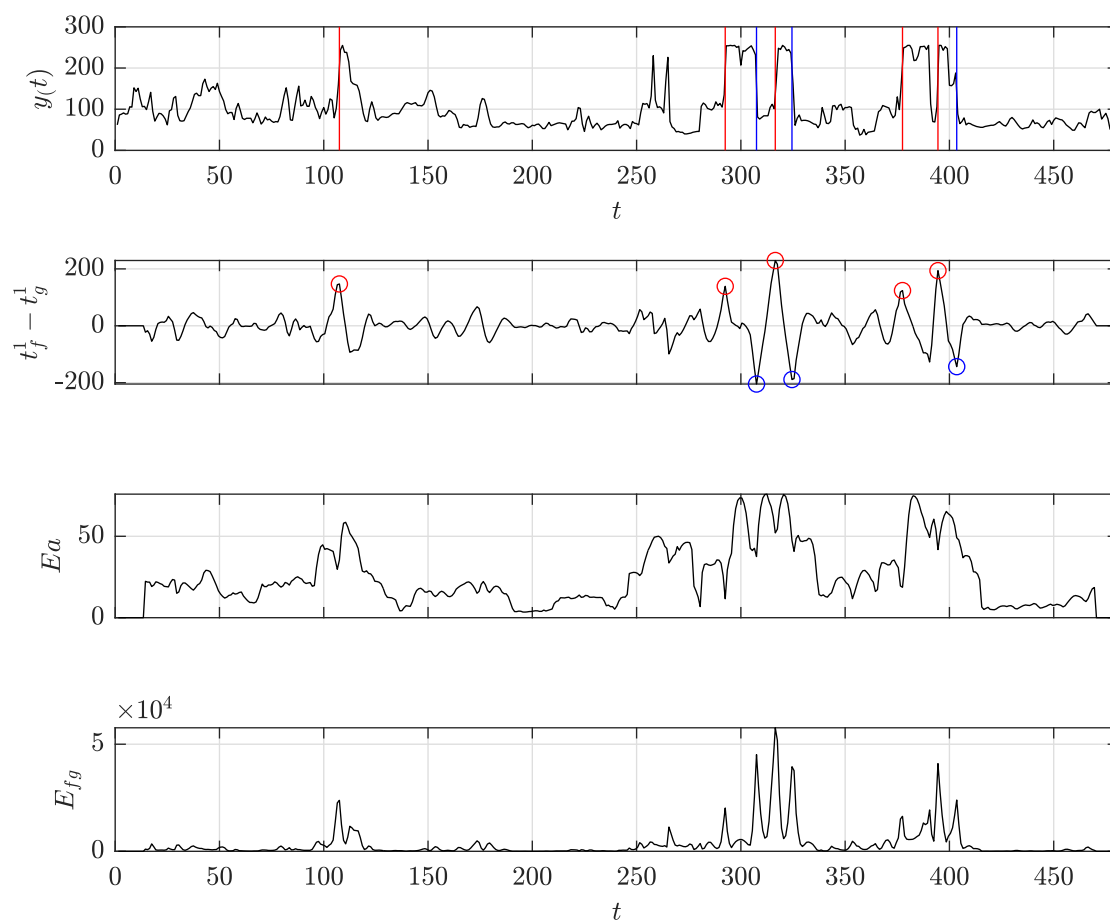


Fig. C.31: Detection of a possible mean-shift change point in the dataset "scanline_126007" with support length $l_s = [12, 12]$ and constraint vector of $c = [1, 0]$.

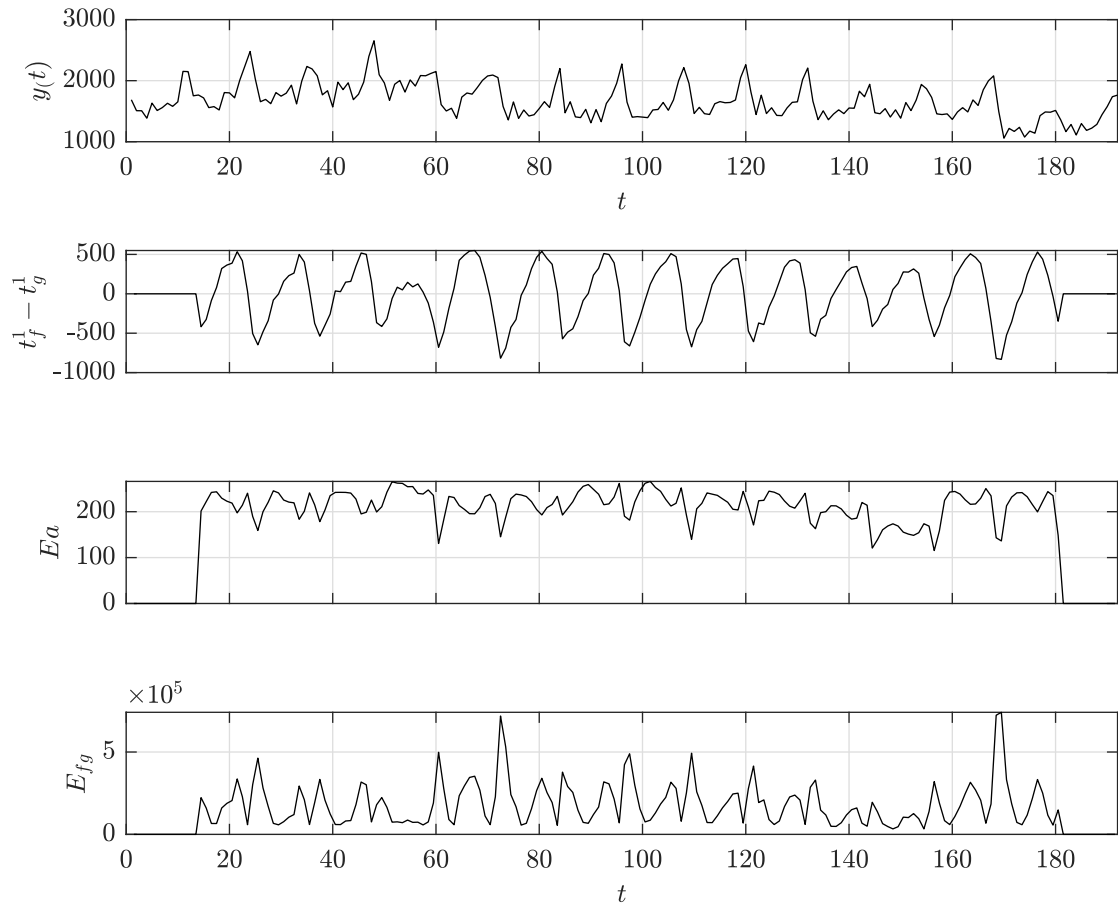


Fig. C.32: Detection of a possible mean-shift change point in the dataset "seatbelts" with support length $l_s = [12, 12]$ and constraint vector of $c = [1, 0]$.

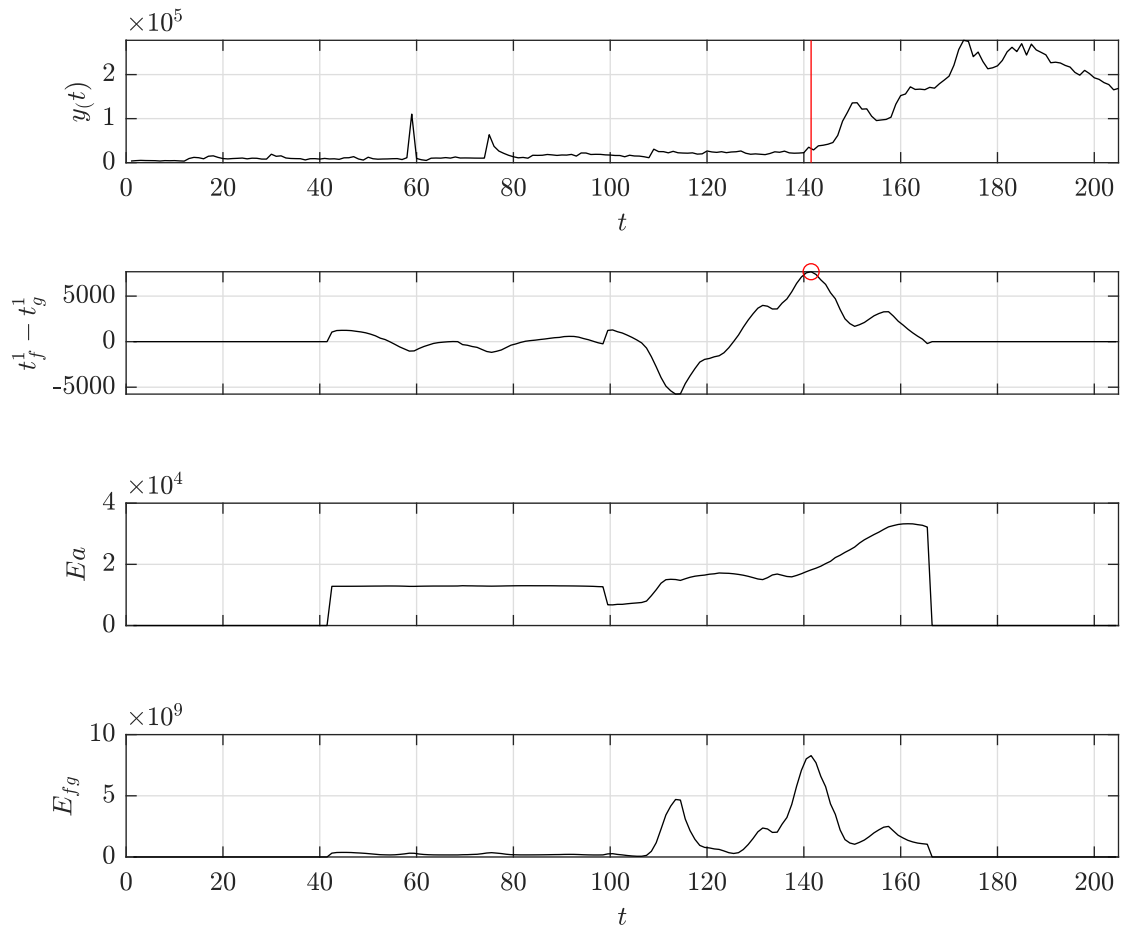


Fig. C.33: Detection of a possible slope-switch change point in the dataset "shanghai_license" with support length $l_s = [40, 40]$ and constraint vector of $c = [1, 0, 1]$.

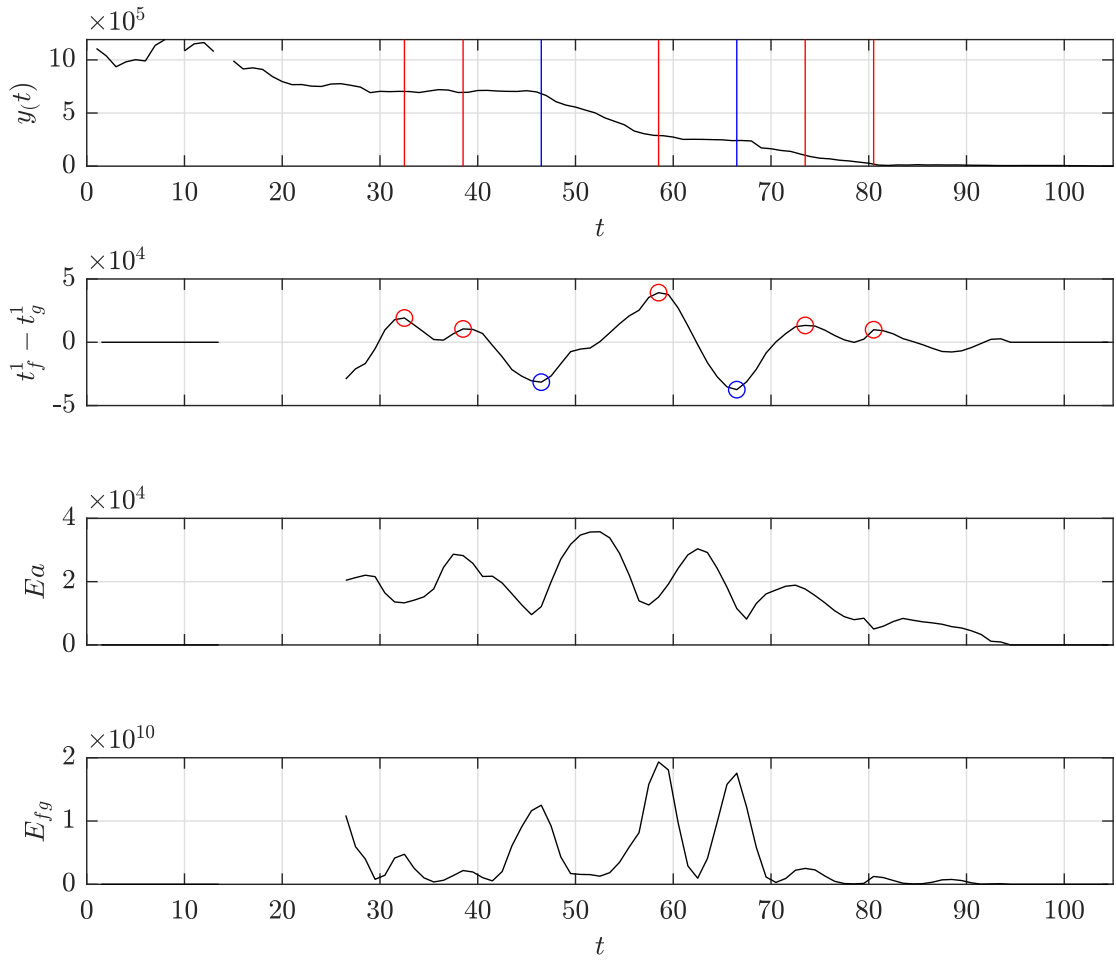


Fig. C.34: Detection of a possible slope-switch change point in the dataset "uk_coal_employ" with support length $l_s = [12, 12]$ and constraint vector of $c = [1, 0, 1]$.

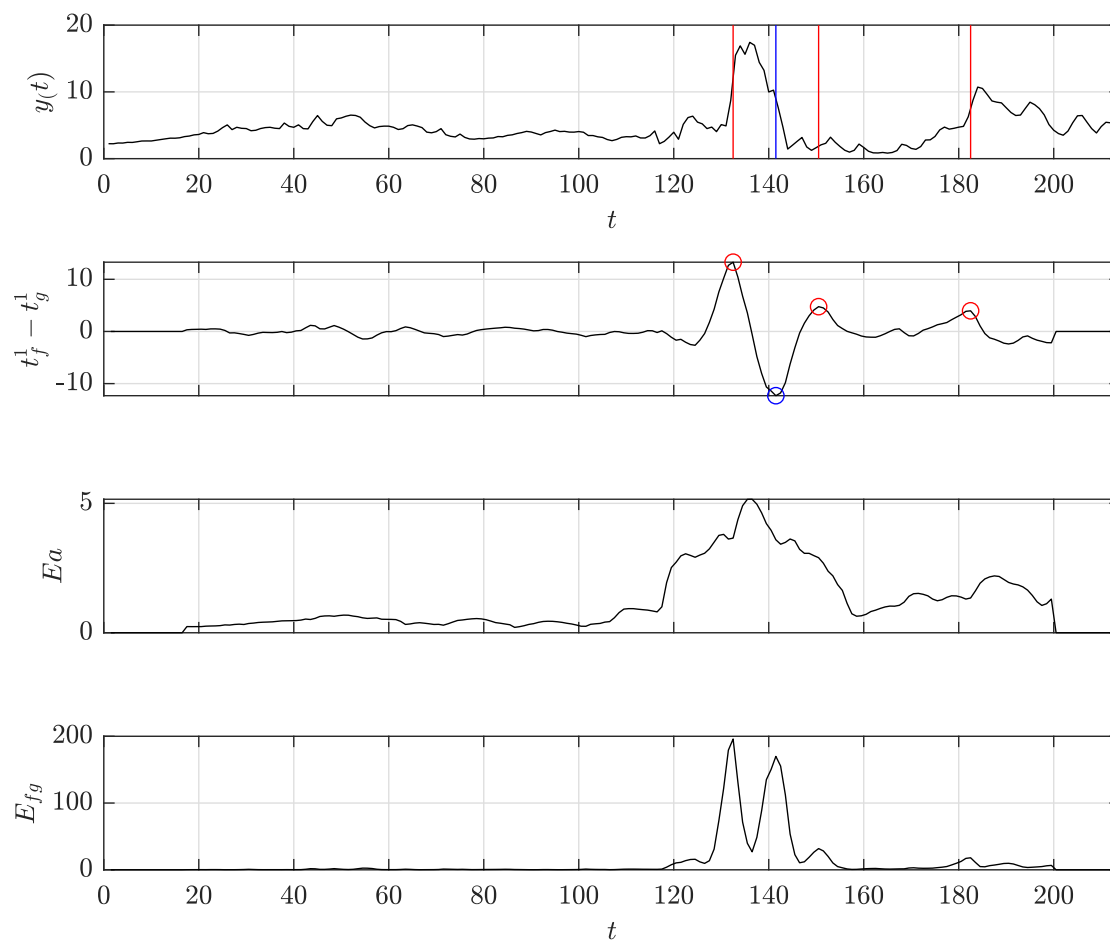


Fig. C.35: Detection of a possible mean-shift change point in the dataset "unemployment_nl" with support length $l_s = [15, 15]$ and constraint vector of $c = [1, 0]$.

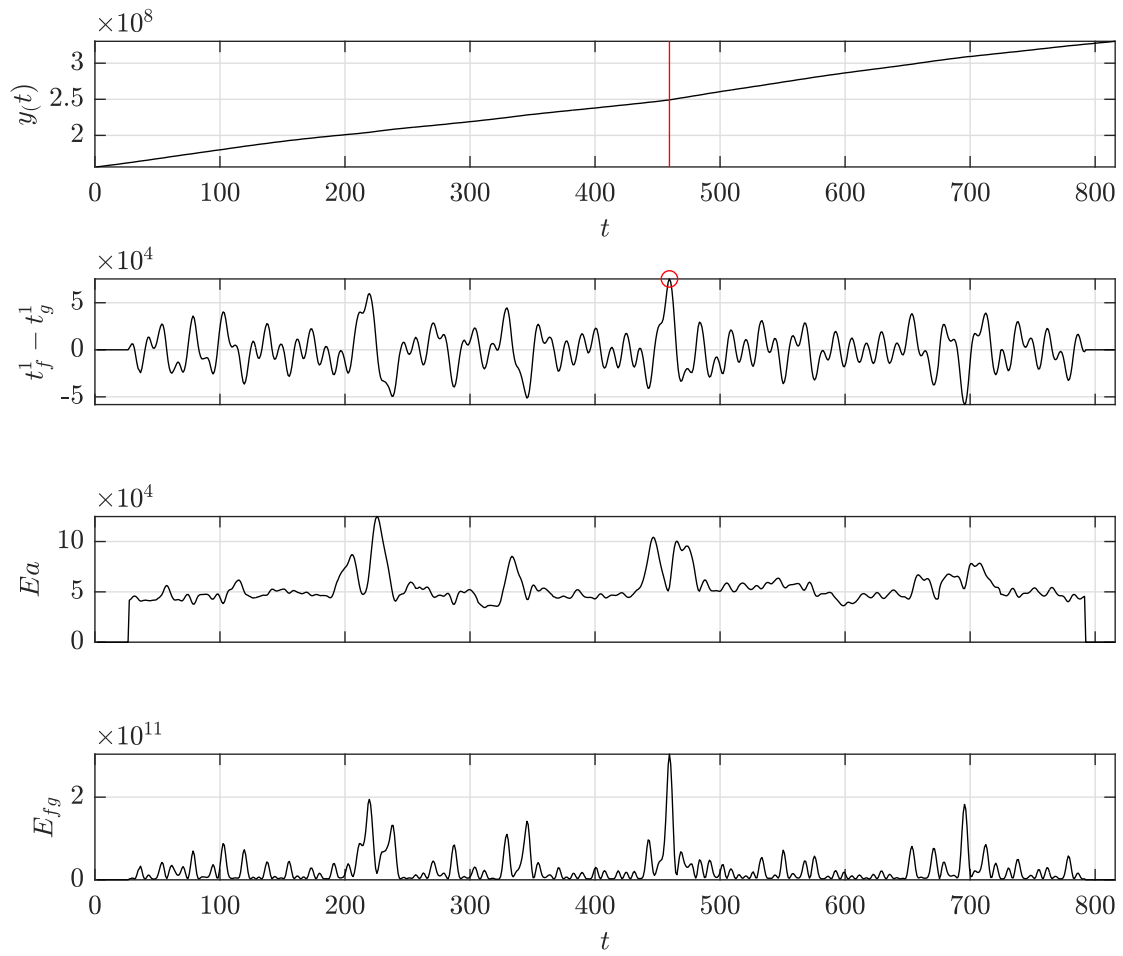


Fig. C.36: Detection of a possible slope-switch change point in the dataset "us_population" with support length $l_s = [25, 25]$ and constraint vector of $c = [1, 0, 1]$.

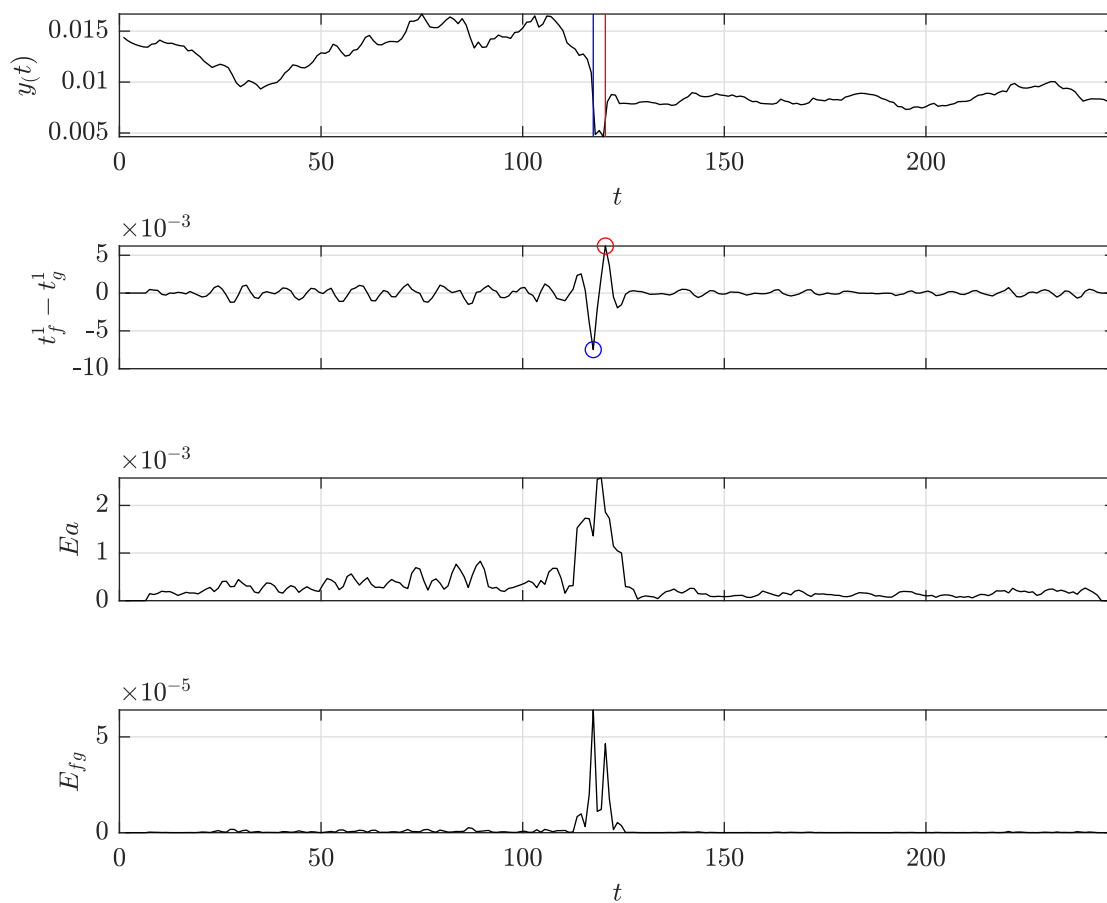


Fig. C.37: Detection of a possible mean-shift change point in the dataset "usd.isk" with support length $l_s = [5, 5]$ and constraint vector of $c = [1, 0]$.

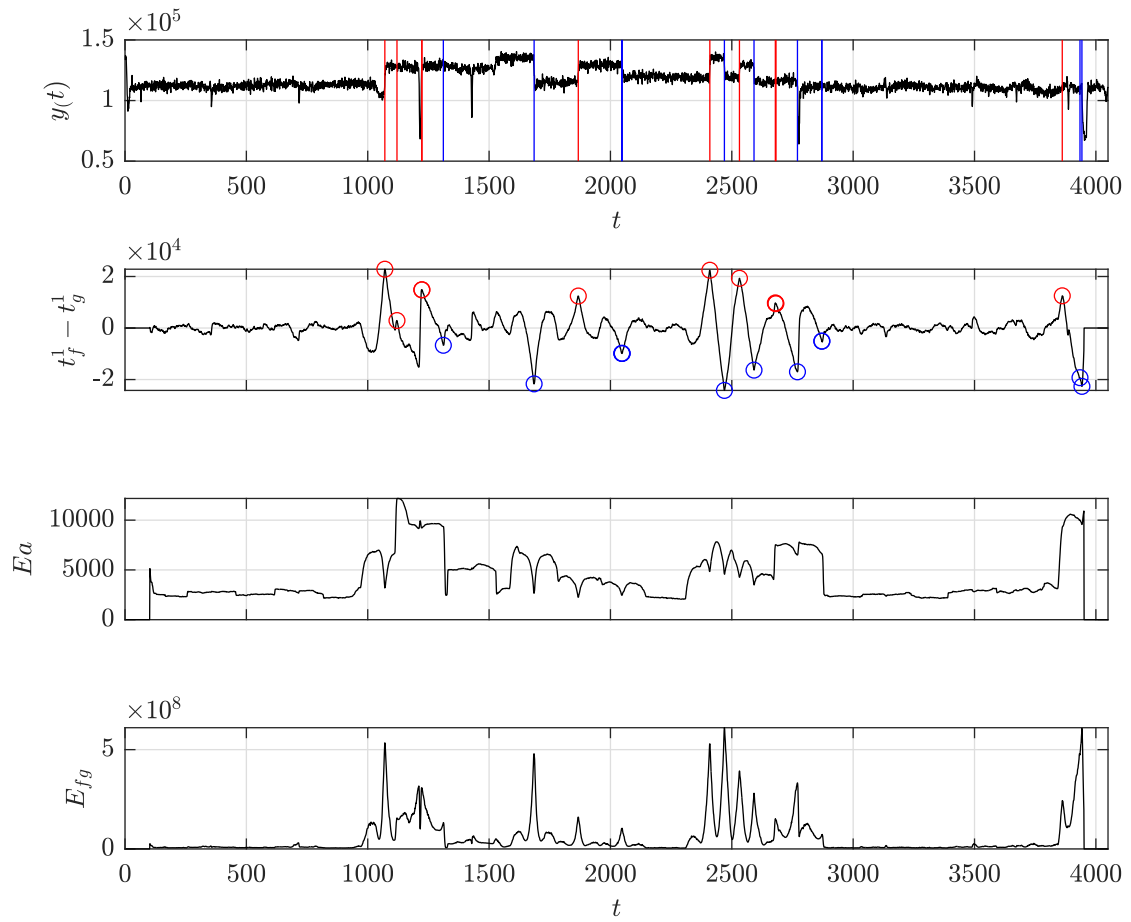


Fig. C.38: Detection of a possible mean-shift change point in the dataset "well_log" with support length $l_s = [100, 100]$ and constraint vector of $c = [1, 0]$.