POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

SP&L Homework 3

Tesi di Laurea Magistrale in
Telecommunication Engineering

Author: **Hong CHEN**

Student ID: 11024836
Advisor: Prof. Name Surname
Co-advisors: Name Surname, Name Surname
Academic Year: 2025-07

Contents

Contents **i**

1 Adaptive interference mitigation on audio signals **1**

1.1 Part 1-1 1

1.1.1 Problem Description 1

1.1.2 Prediction model and derivation 1

1.1.3 Implementation Strategy 2

1.1.4 Results and Analysis 3

1.2 Part 1-2 4

1.2.1 **a)** 4

1.2.2 **b)** 5

1.2.3 c) 5

1.3 Part 1-3 7

1.3.1 Repeat 2a) 7

1.3.2 Repeat 2b) 7

1.4 Part 2: Detection of non-stationary processes 7

1.4.1 Part 2-1 9

1.4.2 MAP Estimation 10

1.4.3 Part 2 - Q(2): Unknown Model Parameters Estimation 11

1.5 Part 2-4: Relationship Between ρ and MSE 13

1 | Adaptive interference mitigation on audio signals

1.1. Part 1-1

1.1.1. Problem Description

In digital marketing scenarios, companies use advertising to influence customer behavior. In this task, the goal is to assess the causal effect of an advertising campaign by analyzing the time series of a selected KPI (e.g., Clicks) for one company. The analysis aims to detect if and how the KPI behavior changes after a known activation time T_{20} (given as 397).
Specifically, the task asks to:

- Predict future KPI values using a linear model based on a window of past values;
- Train the model only using data **before the advertisement is activated**;
- After the advertisement, recursively use predicted values instead of real values;
- Quantitatively evaluate the prediction accuracy and illustrate performance.

1.1.2. Prediction model and derivation

We use a linear autoregressive model of the form:
where:

- $x[n]$ is the value of the KPI at day n;
- k is the prediction gap;
- p is the window size (i.e., number of past days used);
- α_i are the learned prediction coefficients.

$$x[n+k] = \sum_{i=0}^{p-1} \alpha_i \cdot x[n-i] + \epsilon \tag{1.1}$$

$$\epsilon_n = x[n+k] - \sum_{i=0}^{p-1} \alpha_i \cdot x[n-i] \tag{1.2}$$

$$\text{MSE} = \mathbb{E}[\epsilon_n^2] = \mathbb{E}\left[\left(x[n+k] - \sum_{i=0}^{p-1} \alpha_i \cdot x[n-i]\right)^2\right] \tag{1.3}$$

$$\text{MSE} = \mathbb{E}\left[x[n+k]^2\right] - 2\sum_{i=0}^{p-1} \alpha_i \mathbb{E}\left[x[n+k]x[n-i]\right] + \sum_{i=0}^{p-1}\sum_{j=0}^{p-1} \alpha_i \alpha_j \mathbb{E}\left[x[n-i]x[n-j]\right] \tag{1.4}$$

To minimize the mean squared error:

$$\frac{\partial \text{MSE}}{\partial \alpha_i} = -2\mathbb{E}\left[x[n+k]x[n-i]\right] + 2\sum_{j=0}^{p-1} \alpha_j \mathbb{E}\left[x[n-i]x[n-j]\right] \tag{1.5}$$

$$\mathbb{E}\left[x[n+k]x[n-i]\right] = \sum_{j=0}^{p-1} \alpha_j \mathbb{E}\left[x[n-i]x[n-j]\right] \tag{1.6}$$

$$R \cdot \alpha = r \tag{1.7}$$

We use the **Wiener-Hopf** equation:

$$\alpha = R^{-1} r \tag{1.8}$$

1.1.3. Implementation Strategy

The process is divided into two phases:

- **Training phase:** Use only data where $n + k \leq T_{20}$ to estimate coefficients α.
- **Prediction phase:**
  - If the prediction window is entirely before T_{20}, use real values.
  - If the prediction window crosses T_{20}, use real values before T_{20} and estimated values after.
  - If the prediction point is beyond $T_{20} + k$, only estimated values are used.

1.1.4. Results and Analysis

To evaluate the prediction performance of the linear estimator, we tested four configurations by varying the window size (W) and the prediction gap (Gap). The figures below compare the predicted KPI values (red dashed) with the true values (blue solid) under different settings.

- **Observation:** As shown in Figure 1.1 and 1.3, both models with Gap = 1 provide close tracking of the true values before the ad activation point ($T_{20} = 397$). After the ad, the predictions begin to deviate but still maintain a relatively flat trend.
- **Explanation:** A small prediction gap allows the model to capture short-term dynamics effectively. Larger window size ($W = 80$) slightly improves stability, leading to lower MSE.

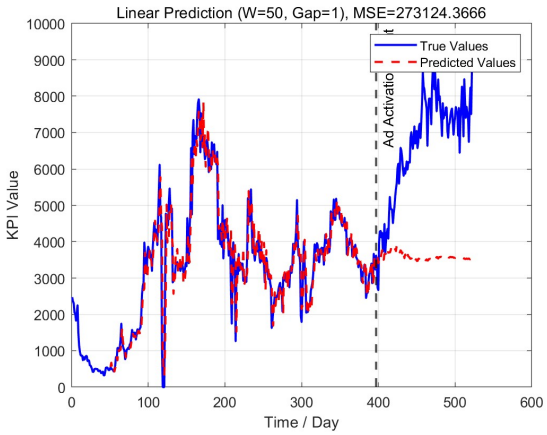

Figure 1.1: Part1: Pred vs. True KPI(Click) - W=50, Gap = 1

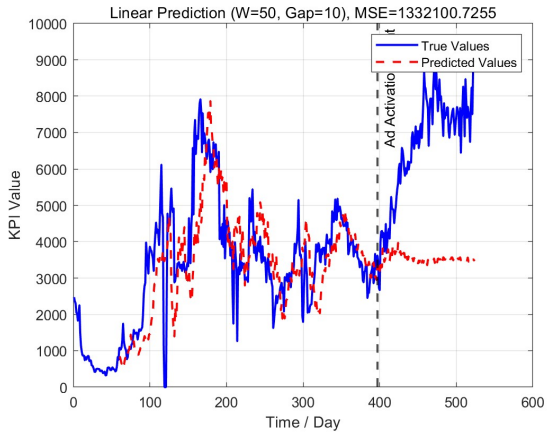

Figure 1.2: Part1: Pred vs. True KPI(Click) - W=50, Gap = 10

Case: $W = 50, 80$, Gap = 10

- The prediction diverges from the real value more after the ad.
- But the shape of the predicted curve is similar to the true one, just **delayed by about 10 days**.
- This delay still helps detect the ad's effect, since the model (trained before the ad) fails to follow the sudden rise.

Conclusion:
Based on the simulation results, we conclude that the advertisement added on day 397 had a significant effect on the KPI of the 20th company.
This is shown by the sudden divergence between the predicted values and the true values

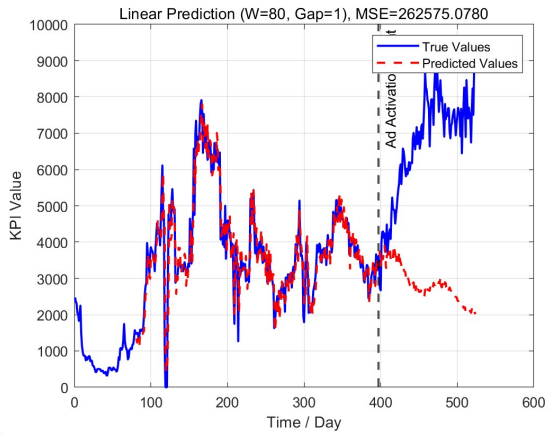

Figure 1.3: Part1: Pred vs. True KPI(Click) - W=80, Gap = 1

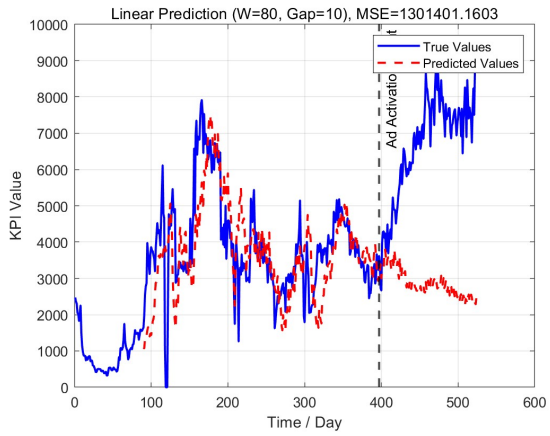

Figure 1.4: Part1: Pred vs. True KPI(Click) - W=80, Gap = 10

after the ad activation point. Before the ad, the model predicted accurately; after the ad, the predictions failed to follow the real growth trend.

1.2. Part 1-2

1.2.1. a)

$$\hat{x}_i(t) = f_i\left[x_1(t), x_2(t), \ldots, x_K(t) \mid k \neq i\right] \tag{1.9}$$

$$\hat{x}_i(t) = \sum_{k \neq i} \alpha_{ik} x_k(t) \tag{1.10}$$

$$\mathrm{MSE}_i = \mathbb{E}\left[(x_i(t) - \hat{x}_i(t))^2\right] \tag{1.11}$$

$$\frac{\partial \mathrm{MSE}_i}{\partial \alpha_{ik}} = 0 \tag{1.12}$$

We can calculate a cross-correlation matrix R of size 20×20, where:

$$R_{ij} = \mathrm{corr}(x_i, x_j) = \frac{\sum_{n=1}^{N}(x_i(n) - \bar{x}_i)(x_j(n) - \bar{x}_j)}{\sqrt{\sum_{n=1}^{N}(x_i(n) - \bar{x}_i)^2 \cdot \sum_{n=1}^{N}(x_j(n) - \bar{x}_j)^2}} \tag{1.13}$$

1.2.2. b)

Importance of Normalization in Correlation Analysis:
When comparing different companies' KPI time series, it is crucial to account for differences in magnitude. Some companies may have KPI values in the thousands, while others may only have values in the tens. If not normalized, this imbalance can distort both correlation analysis and regression-based predictions.
To address this, we apply the normalization z-score to each company time series:

$$\tilde{x}_k(t) = \frac{x_k(t) - \mu_k}{\sigma_k} \tag{1.14}$$

This transformation ensures that all time series are on the same scale (zero mean, unit variance), allowing for fair comparison of trends. Moreover, the Pearson correlation coefficient between two normalized series $\tilde{x}$ and $\tilde{y}$ is equivalent to their covariance.

$$\rho_{xy} = \frac{\mathrm{Cov}(x, y)}{\sigma_x \sigma_y} = \mathrm{Cov}(\tilde{x}, \tilde{y}) \tag{1.15}$$

Simulation Results Analysis:
From the results, we find that several companies can be predicted well by their similar peers, suggesting that they belong to the same industry or exhibit similar business behavior. However, some companies are relatively independent in behavior and cannot be effectively predicted by others.

1.2.3. c)

Modeling and Assumptions
To evaluate whether the advertisement for Company 20 is effective, we use the data from other companies that are highly correlated (correlation coefficient $|\rho| \geq 0.3$) as predictors. Importantly, the linear prediction model is trained only using the pre-advertisement data (i.e., days 1 to 396), and the learned weight vector $\boldsymbol{\alpha}$ is fixed for predicting the entire timeline.
Conclusion:
As shown in Figure 1.7, the predicted values diverge significantly from the ground truth immediately after the advertisement starts. Specifically, during the first 60 days after day 397, the real KPI values (clicks) are clearly higher than the predictions, indicating a strong short-term impact of the advertisement. However, after this 60-day period, the predicted values start to catch up with the real values, showing that the advertisement's effect gradually diminishes.

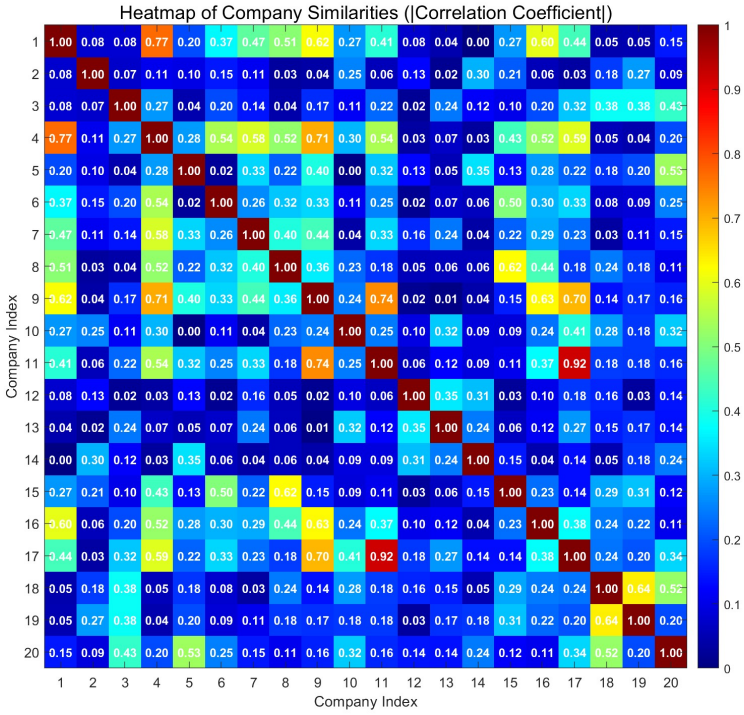

Figure 1.5: Part1-2b: heatmap of Similarity (Click)

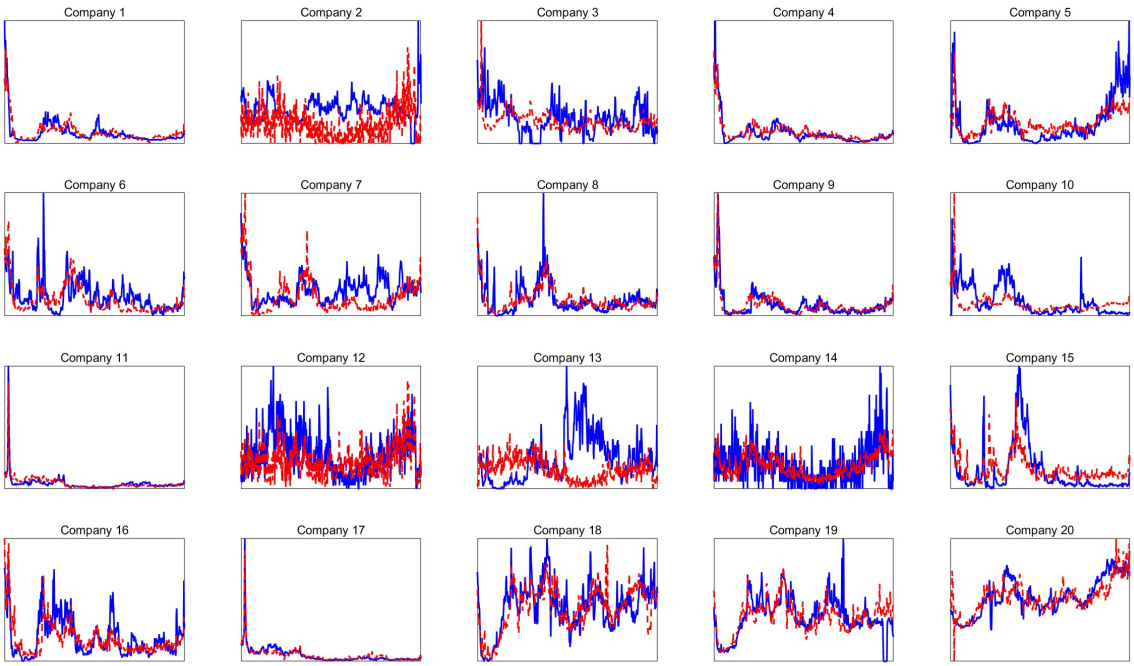

Figure 1.6: Part1-2b: Pred vs. True KPI(Click)

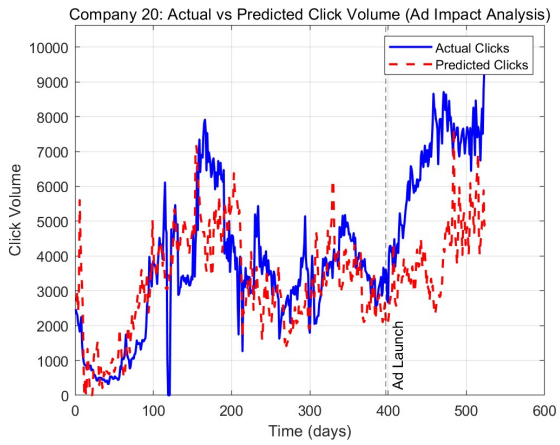

Figure 1.7: Part 1-2c

1.3. Part 1-3

Method: For each company i, we use all three KPIs from the remaining $K-1$ companies as input features, i.e.,

$$\text{Input features: } \{\text{Click}_j, \text{Cost}_j, \text{Conv}_j\}_{j \neq i}, \quad \text{Total dimension: } 3 \times (K-1)$$

We then use Least Squares Estimation (LSE) to predict the Click(or Conversion) KPI of company i.

1.3.1. Repeat 2a)

Observation:
As shown in Figure **??**, for most companies, using all KPIs gives slightly better or comparable performance than using only Click data. This indicates that **Cost** and **Conversion** are useful auxiliary signals in certain cases.

1.3.2. Repeat 2b)

1.4. Part 2: Detection of non-stationary processes

For $n \leq T_0$, the signal $x[n]$ follows the AR(1) model:

$$x[n] = \rho \cdot x[n-1] + w[n] \tag{1.16}$$

where ρ is the AR(1) coefficient, and $w[n] < \mathcal{N}(0, {}^{2}_{\alpha})$ is white noise.

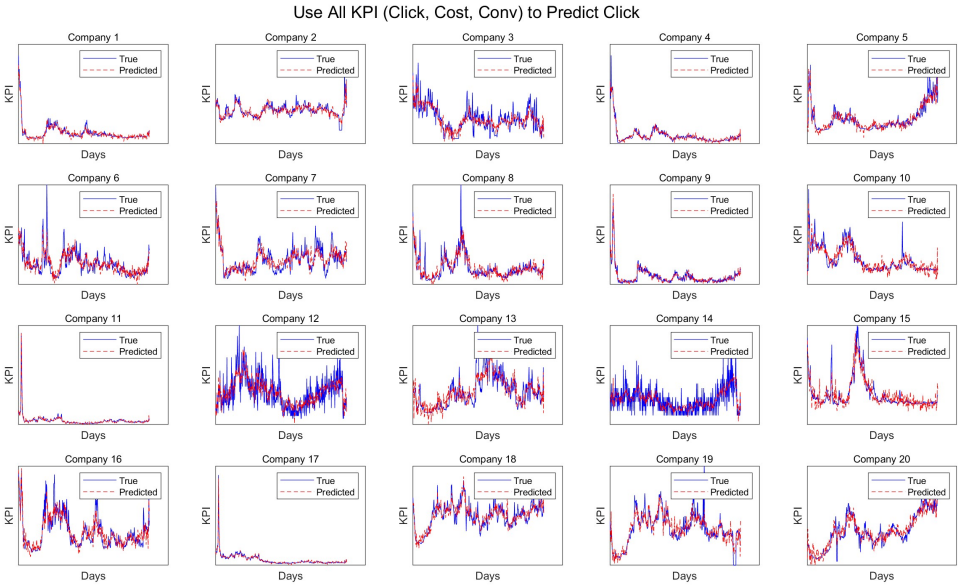

Figure 1.8: Part1: 3) Pred vs. Truth (repeat 2a)

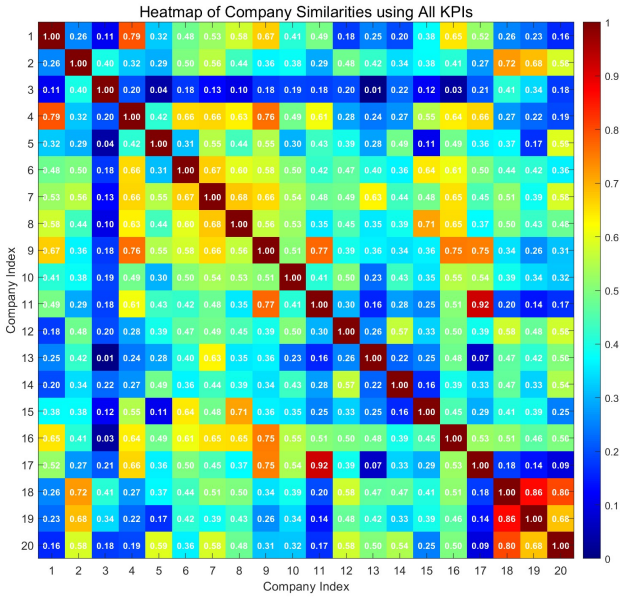

Figure 1.9: Part1: 3) Heatmap of Similarity (repeat 2b)

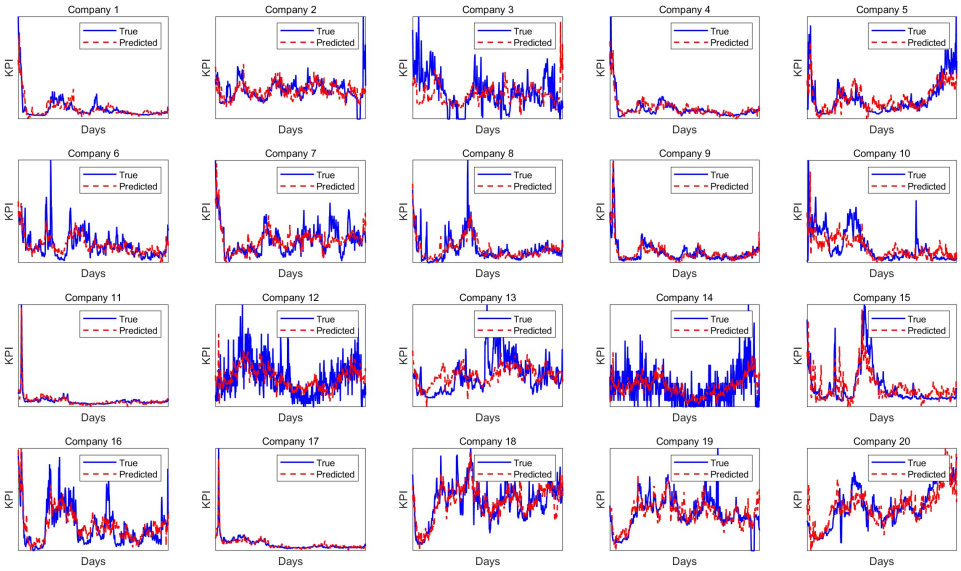

Figure 1.10: Part1: 3) Pred vs. Truth (repeat 2b)

- $x[n]$ depends on its previous value $x[n-1]$ and the noise $w[n]$.
- The pole location ρ indicates the autocorrelation of the signal; the closer ρ is to 1, the stronger the temporal dependence of the signal.

For $n > T_0$, the signal $x[n]$ follows the AR(2) model:

$$x[n] = a_1 \cdot x[n-1] + a_2 \cdot x[n-2] + w[n] \tag{1.17}$$

where:

$$a_1 = \rho \cdot \cos(\phi), \quad a_2 = -\frac{\rho^2}{4} \tag{1.18}$$

are the AR(2) coefficients, and $w[n] < \mathcal{N}(0, \frac{2}{\beta})$ is white noise.

- This process uses two previous values, $x[n-1]$ and $x[n-2]$, to determine the current value $x[n]$, which indicates a stronger dependence of the signal.
- The pole location $\rho/2 \cdot expa(\pm j\phi)$defines the oscillatory characteristics of the signal.

1.4.1. Part 2-1

Goal:

To design a method that can detect when a stochastic process $x[n]$ undergoes a change in its statistical properties, i.e., it becomes non-stationary after a certain unknown point

T_o. The process $x[n]$ is defined as follows:

$$x[n] = \begin{cases} \alpha[n], & n < T_o \\ \beta[n], & n \geq T_o \end{cases}$$

where $\alpha[n]$ and $\beta[n]$ are two different autoregressive (AR) random processes:

- $\alpha[n]$ is an AR(1) process with one pole;
- $\beta[n]$ is an AR(2) process with two complex-conjugate poles;

1.4.2. MAP Estimation

The objective is to estimate T_0 via Maximum A Posteriori (MAP) estimation. According to Bayes' rule:

$$P(T_0|X) \propto P(X|T_0) \cdot P(T_0)$$

Assume $P(T_0)$ is uniform. The key term is the likelihood: - Before T_0: fit AR(1) and compute residuals:

$$e_{AR1}(n) = x[n] - \rho x[n-1]$$

- After T_0: fit AR(2) and compute residuals:

$$e_{AR2}(n) = x[n] - a_1 x[n-1] - a_2 x[n-2]$$

Define total Residual Sum of Squares (RSS):

$$RSS(T_0) = \sum_{n \leq T_0} e_{AR1}^2(n) + \sum_{n > T_0} e_{AR2}^2(n)$$

Then posterior is:

$$P(T_0|X) \propto \exp\left(-\frac{1}{2}(RSS(T_0))\right)$$

The MAP estimate is the T_0 that minimizes RSS.

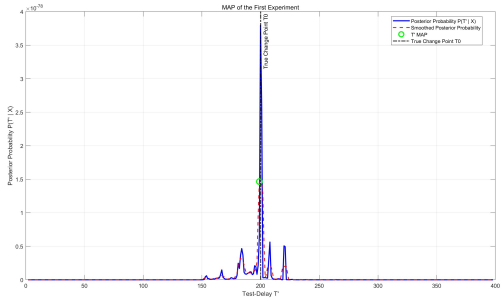

Figure 1.11: Part2: 1): MAP

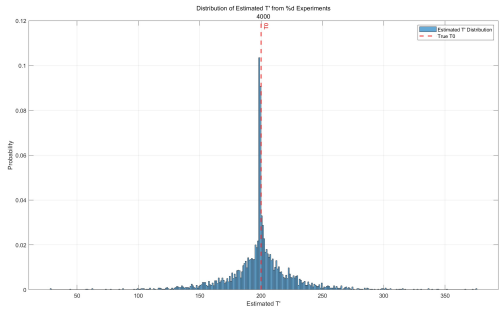

Figure 1.12: Part2: 1): Distribution of Estimation

1.4.3. Part 2 - Q(2): Unknown Model Parameters Estimation

In this case, the AR(1) and AR(2) models are unknown. We estimate their parameters using Least Squares Estimation (LSE) and apply MAP to detect the change point T_0.

AR(1) Estimation:

For $n \leq T'$:

$$x[n] = \rho x[n-1] + w[n], \quad \hat{\rho} = \frac{\sum x[n]x[n-1]}{\sum x[n-1]^2}$$

$$RSS_1 = \sum (x[n] - \hat{\rho} x[n-1])^2$$

AR(2) Estimation:

For $n > T'$:

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + v[n]$$

$$\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \end{bmatrix} = (X^T X)^{-1} X^T Y, \quad RSS_2 = \sum (x[n] - \hat{a}_1 x[n-1] - \hat{a}_2 x[n-2])^2$$

Simulation Results:

- $\hat{\rho}_{mean} = 0.7924$

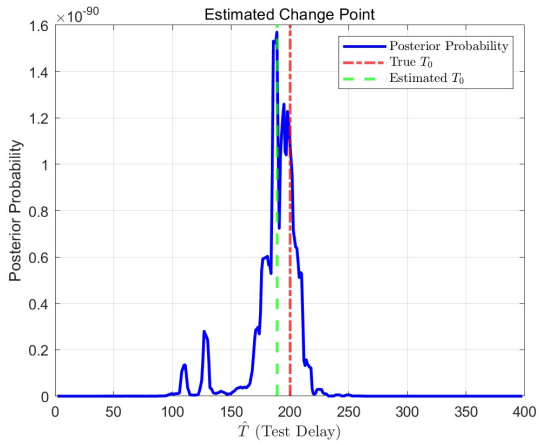

Figure 1.13: Estimated change point

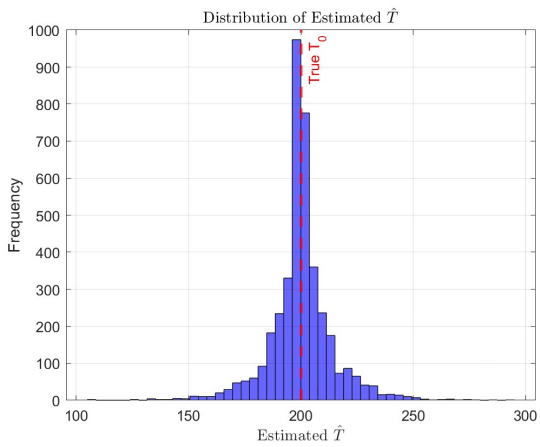

Figure 1.14: Distribution of Estimated T

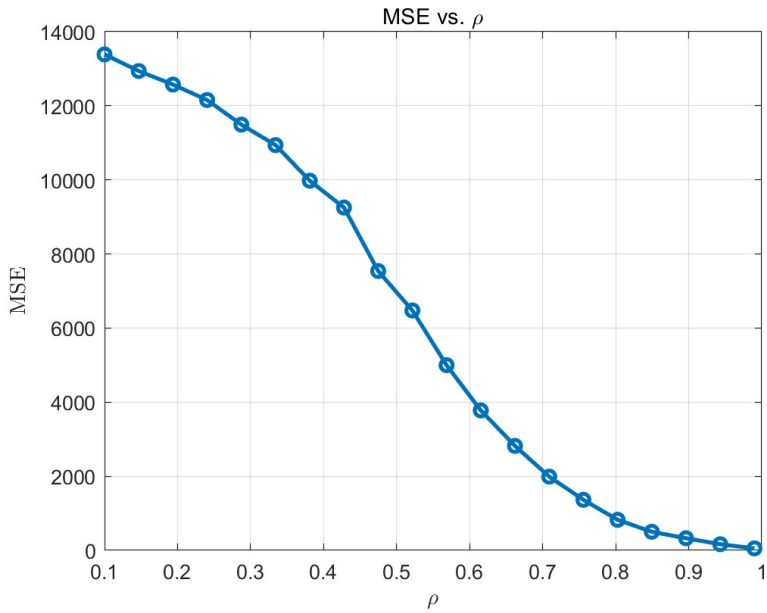

Figure 1.15: Part2-4: MSE vs. ρ

- $\hat{a}_{1mean} = 0.4955$, $\hat{a}_{2mean} = -0.2988$
- MSE (Monte Carlo) = 226.19

1.5. Part 2-4: Relationship Between ρ and MSE

Observation: As the AR pole ρ increases, the temporal dependency of the signal strengthens, improving model fitting and detection accuracy.

- As ρ increases, the fit of the AR(1) and AR(2) models improves, reducing residuals, and increasing detection accuracy, which results in a lower MSE.
- As ρ decreases, the model fit worsens, residuals increase, and detection accuracy decreases, leading to a higher MSE.

Conclusion: The higher the value of ρ, the lower the MSE due to improved model fit and clearer statistical separation before and after T_0.