# ROBUST CONIC QUADRATIC PROGRAMMING
# APPLIED TO QUALITY IMPROVEMENT
# – A ROBUSTIFICATION OF CMARS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞE ÖZMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

SEPTEMBER 2010

Approval of the thesis:

**ROBUST CONIC QUADRATIC PROGRAMMING<br>APPLIED TO QUALITY IMPROVEMENT<br>-A ROBUSTIFICATION OF CMARS**

submitted by **AYŞE ÖZMEN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing**, **Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız ____________
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen ____________
Head of Department, **Scientific Computing**

Prof. Dr. Gerhard-Wilhelm Weber ____________
Supervisor, **Institute of Applied Mathematics, METU**

Assoc. Prof. Dr. İnci Batmaz ____________
Co-Supervisor, **Department of Statistics, METU**

**Examining Committee Members:**

Assist. Prof. Dr. Tolga Can ____________
Department of Computer Engineering, METU

Prof. Dr. Gerhard-Wilhelm Weber ____________
Institute of Applied Mathematics, METU

Assist. Prof. Dr. Hakan Öktem ____________
Institute of Applied Mathematics, METU

**Date:** ____________

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name: AYŞE ÖZMEN

Signature :

# ABSTRACT

ROBUST CONIC QUADRATIC PROGRAMMING
APPLIED TO QUALITY IMPROVEMENT
-A ROBUSTIFICATION OF CMARS

Özmen, Ayşe
M.Sc., Department of Scientific Computing
Supervisor: Prof. Dr. Gerhard-Wilhelm Weber
Co-Supervisor: Assoc. Prof. Dr. İnci Batmaz

September 2010, 139 pages

In this thesis, we study and use Conic Quadratic Programming (CQP) for purposes of operational research, especially, for quality improvement in manufacturing. In previous works, the importance and benefit of CQP in this area became already demonstrated. There, the complexity of the regression method Multivariate Adaptive Regression Spline (MARS), which especially means sensitivity with respect to noise in the data, became penalized in the form of so-called Tikhonov regularization, which became expressed and studied as a CQP problem. This was leading to the new method CMARS; it is more model-based and employs continuous, actually, well-structured convex optimization which enables the use of Interior Point Methods and their codes such as MOSEK. In this study, we are generalizing the regression problem by including uncertainty in the model, especially, in the input data, too.

CMARS, recently developed as an alternative method to MARS, is powerful in overcoming complex and heterogeneous data. However, for MARS and CMARS method, data are assumed to contain fixed variables. In fact, data include noise in both output and input variables. Consequently, optimization problem's solutions can show a remarkable sensitivity to perturbations in the parameters of the problem. In this study, we include the existence of uncertainty in the future scenarios into CMARS and robustify it with robust optimization which is dealt with data uncertainty. That kind of optimization was introduced by Aharon Ben-Tal and Arkadi Nemirovski, and used by Laurent El Ghaoui in the area of data mining. It incorporates various kinds of noise and perturbations into the programming problem. This robustification of CQP with robust optimization is compared with previous contributions that based on Tikhonov regularization, and with the traditional MARS method.

Keywords: Multi Adaptive Regression Splines, Conic Multi Adaptive Regression Splines, Robust Optimization, Data Uncertainty, Conic Quadratic Programming.

# ÖZ

KALİTE GELİŞTİRMESİNE UYGULANAN
SAĞLAM KONİK KUADRATİK PROGRAMLAMA
–CMARS METODUNDA BİR SAĞLAMLAŞTIRMA

Özmen, Ayşe
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi: Prof. Dr. Gerhard-Wilhelm Weber
Ortak Tez Yöneticisi: Assoc. Prof. Dr. İnci Batmaz

Eylül 2010, 139 sayfa

Bu çalışmada, yöneylem araştırması amacıyla, özellikle, imalat içindeki kalite geliştirmesi için Konik Karasel Programlamayı inceliyor ve kullanıyoruz. Önceki dönemde ODTÜ Uygulamalı Matematik Enstitüsü'nde yapılan ve tamamlanan Master tezi projesinde Konik Karasel Programlamanın bu alandaki önemi ve yararı zaten açıklanmış oldu. O projede, özellikle veri içindeki parazitle ilgili duyarlık anlamındaki regresyon yöntemi MARS'ın karmaşıklığı, bir Konik Karasel programlama problemi olarak incelenmiş ve açıklanmış olan sözde Tikhonov düzenleme formu içinde cezalandırılmış oldu. Bu durum yeni metod CMARS'ın oluşmasına öncülük etmiştir. CMARS daha fazla model tabanlıdır ve MOSEK gibi kodlarıyla İç Nokta Metodlarının kullanılmasına olanak sağlayan sürekli, doğrusu, iyi yapılandırılmış konveks optimizasyona hizmet verir. Bu çalışmada, biz model içinde özelliklede girdi değişkenlerinde belirsizlik içerecek şekilde regresyon problemimizi geneleleştiriyoruz.

Çok değişkenli uyarlanabilir regresyon eğrileri (MARS) algoritmasına seçenek olarak son zamanlarda geliştirilmiş yeni bir yaklaşım olan konik (konveks, sürekli) çok değişkenli uyarlanabilir regresyon eğrileri (CMARS) algoritması karmaşık ve türdeş olmayan veri kümelerini başarı ile modelleyen bir yöntem olarak karşımıza çıkmaktadır. Ancak MARS ve

CMARS yöntemleri bağımsız (girdi) değişkenlerin sabit olduğunu varsaymaktadır. Aslında yaşam verilerinin tümümde (yani, girdi ve çıktı değerlerinde) gürültü bulunmaktadır ve optimizasyon probleminin çözümleri problem değişkenlerindeki belirsizliklere karşı kayda değer bir duyarlılık gösterebilmektedir. Bu yeni çalışmada, bağımsız değişkenlerin rastgele olduğu varsayılarak CMARS modeline belirsizlik kavramı eklenmiş ve verilerdeki belirsizlikleri ele alan sağlam optimizasyon tekniği ile CMARS model ve algoritması sağlamlaştırılmaktadır. Bu çeşit optimization Aharon Ben-Tal ve Arkadi Nemirovski tarafından tanıtıldı ve Laurent El Ghaoui tarafından veri madenciliği alanında kullanıldı. Bu optimizasyon, programlama problemleri içindeki değişik tarzda belirsizlikleri ve karışıklıkları içermektedir. Sağlam optimizasyon ile bu Konik Karasel Program Sağlamlaştırma, Tikhonov düzenlemesini temel alan önceki katkılarla ve geleneksel MARS yöntemiyle karşılaştırılmaktadır.

Anahtar Kelimeler: Çok Değişkenli Uyarlanabilir Regresyon Eğrileri, Konik Çok Değişkenli Uyarlanabilir Regresyon Eğrileri, Sağlam Optimizasyon, Veri Belirsizliği, Konik Karesel Programlama.

*To my father*

# ACKNOWLEDGMENTS

I owe my deepest gratitude to my supervisor, Prof. Dr. Gerhard Wilhelm Weber who encouraged, guided, helped and supported me from the initial to the final level of the study. It is an honor for me to work/study with him.

I would like to thank my co-supervisor Assoc. Prof. Dr. İnci Batmaz for motivating and helping me.

I am grateful to my friends Melike Kaya, Bengisen Pekmen and Miray Hanım Aslan for their helping, understanding and guidance and throughout this study.

I am thankful to Vefa Jafarova and Mehlika Kocabaş for her friendship and help.

I am indebted to personel of Institute of Applied Mathematics for their friendship, understanding and help.

I would like to show my gratitude to each member of my family. I would like to thank my mother Ayşin Özmen, my older sister Özlem Özmen, my brother Ali Özmen and my sister Pınar Özmen for their patience and love. This thesis would not have been possible without my family’s continuous support, endless love and patient.

Special thanks to my father Mehmet Zeki Özmen who always believed and encourged me during my life. I would like to dedicate this study to my father whom I am proud of.

Thank you...

# TABLE OF CONTENTS

APPENDICES

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# ABBREVIATIONS

| | |
|---|---|
| Additive Model | : AM |
| Artificial Neural Network | : ANN |
| Average Absolute Error | : AAE |
| Basis Functions | : BFs |
| Classification and Regression Tree | : CART |
| Confidence Intervals | : CIs |
| Conic Dual Problem | : CD |
| Conic Multivariate Adaptive Regression Splines | : CMARS |
| Conic Quadratic | : CQ |
| Conic Quadratic Programming | : CQP |
| Conic Primal Problem | : CP |
| Cross-Validation | : CV |
| Correlation Coefficient | : r |
| Decision Tree | : DT |
| Generalized Additive Models | : GAMs |
| Generalized Cross-Validation | : GCV |
| Generalized Linear Models | : GLMs |
| Generalized Partial Linear Model | : GPLM |
| Interior Point Methods | : IPMs |
| Least-Squares | : LS |
| Least-Squares Estimation | : LSE |
| Linear Matrix Inequality | : LMI |
| Linear Programming | : LP |
| Mean Square Error | : MSE |
| Multivariate Adaptive Regression Spline | : MARS |
| Multivariate Linear Regression | : MLR |
| Penalized Residual Sum of Squares | : PRSS |
| Quadratic Constrained Quadratic Programming | : QCQP |

| | |
|---|---|
| Recursive Partitioning Regression | : RPR |
| Regression Tree | : RT |
| Residual Sum of Squares | : RSS |
| Robust Conic Multivariate Adaptive Regression Splines | : RCMARS |
| Robust Optimization | : RO |
| Root Mean Squared Error | : RMSE |
| Second-Order Cone Programming | : SOCP |
| Semidefinite Programming | : SDP |
| Singular Value Decomposition | : SVD |
| Support Vector Machine | : SVM |
| Tikhonov Regularization | : TR |
| Weak Robust Conic Multivariate Adaptive Regression Splines | : WRCMARS |

# CHAPTER 1

# INTRODUCTION

## 1.1 Contributions of the Thesis

*Multivariate Adaptive Regression Spline* (*MARS*) has been applied successfully too many fields of science, economy and technology in recent years. It bases on a modern methodology from statistical learning, which is important in both regression and classification. MARS builds flexible high-dimensional nonparametric regression models, and presents a great promise for fitting nonlinear multivariate functions. It generates an additive model in two-stage process: the forward and backward stepwise algorithms. In *Conic Multivariate Adaptive Regression Spline* (*CMARS*) method, the backward stepwise algorithm is not applied. Instead of this, a *Penalized Residual Sum of Squares* (*PRSS*) is employed for MARS as a *Tikhonov Regularization* (*TR*) problem [89]. This two-objective optimization problem is treated using the continuous optimization technique called *Conic Quadratic Programming* (*CQP*).

*CMARS* is an alternative method to a well-known regression tool MARS from data mining and estimation theory. With this study, we further improve CMARS so that it can treat uncertainty in the data. In fact, generally, data may include noise in both input and output variable. This means that the data of the regression problem are not exactly known or may not be exactly measured, or the exact solution of the problem may not be carried out because of intrinsic inaccuracy of the devices [17]. Furthermore, the data can undergo small changes by variations in the optimal experimental design. These altogether leads to uncertainty in the objective function and in possible constraints. To handle this, we refine our CMARS algorithm by an important robust optimization developed by Ben-Tal and Nemirovski [7, 9, 10], and El-Ghaoui and Lebret [27], and call it as RCMARS.

*Robust optimization* (*RO*) is a modeling methodology to process optimization problems in which the data are uncertain and are only known to belong to some uncertainty set, except for outliers. The purpose of RO is to find an optimal or near optimal solution which is feasible for every possible realization of the uncertain scenarios [15].

In a previous study, “a comparison of data mining methods for prediction and classification types of quality problems” reveals that MARS outperforms with respect to several performance criteria [1]. Based on this finding, in this study, we further improve MARS to be able to deal with not only fixed but also random type input data.

## 1.2 Purpose of the Study

CMARS models depend on the parameters. Small perturbations in data may give different model parameters. This can cause unstable solutions. In CMARS, the aim is to reduce the estimation error while keeping efficiency as high as possible. In order to achieve this aim, we apply some approaches like scenario optimization, robust counterpart and usage of more robust estimators. By using robustification in CMARS, we try to reduce the estimation variance.

We firstly analyze how uncertainty incorporated into the CMARS model with complexity terms in the form of integrals of squared first- and second-order derivatives of the model functions and, then, the discretized TR, and finally, the CQP form of the problem. Then, we introduce a *robustification* of CMARS with robust optimization under polyhedral uncertainty and ellipsoidal uncertainty [67, 68]. Because of the computational effort which our robustification of CMARS easily need, we also present the concept of a *weak robustification*.

## 1.3 Scope of the Thesis

This thesis is comprised of six main chapters and an Appendix. Briefly summarizing, the contents are organized as follows:

Chapter 1: Introduction of the thesis. The objectives and outlines of the study is given in this chapter.

Chapter 2: The background information about Regression, Tikhonov Regularization, Conic and Robust Optimization.

Chapter 3: Theory and approaches of MARS and CMARS methods.

Chapter 4: Robust CMARS is introduced in theory and method. RCMARS results with different uncertainty scenarios for the numerical example studied in our study.

Chapter 5: Simulation results for comparison of MARS, CMARS and RCMARS methods. The results are discussed at the end of this chapter.

Chapter 6: Conclusion and further studies are stated in the last chapter.

# CHAPTER 2

# BACKGROUND

## 2.1 Regression

Regression analysis is a mathematical and statistical technique which is very useful for many types of problems in engineering and science analyzing the relationship between dependent variable and one or more independent variables. Regression analysis is widely used for prediction and estimation and most commonly estimates the conditional expectation of the dependent variable given the independent variables [57].

There exist many regression models such as Linear Regression Models, Nonlinear Regression Models, Generalized Linear Models, Nonparametric Models, and Generalized Additive Models.

### 2.1.1 Linear Regression Model

The case of simple linear regression regards a single *predictor* (regressor) variable *X* and a *response* (dependent) variable *Y*. It is assumed that the true relationship between *Y* and *X* is a straight line and that the observation *Y* at each level of *x* is a random variable [57]. The conditional expected value of *Y* for each value of *X* (=*x*) is

$$E(Y \mid X) = \alpha_0 + \alpha_1 X. \tag{2.1.1}$$

Here, the intercept $\alpha_0$ and the slope $\alpha_1$ are unknown regression coefficients. It is supposed that each observation, *Y*, may be defined by the model [57]

$$Y = \alpha_0 + \alpha_1 X + \varepsilon. \tag{2.1.2}$$

Here, $\varepsilon$ is a random error with zero mean, and $\sigma^2$ is the unknown variance. Also, we suppose that the random errors corresponding to different observations are uncorrelated random variables.

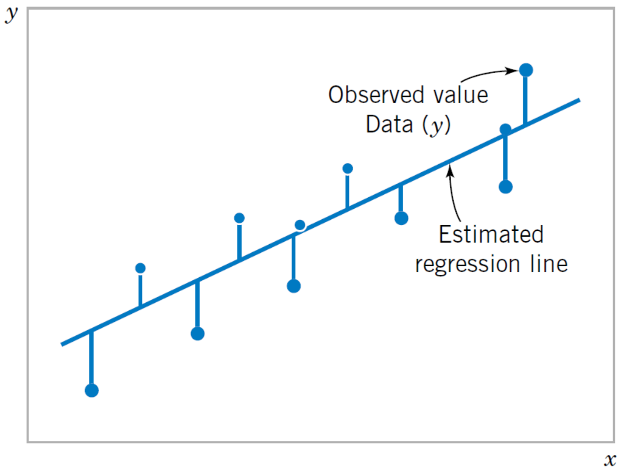


Figure 2.1 Deviation of the points about a line from the estimated regression model [57].

To select the **best-fitting** line for a set of data, the unknown parameters of the simple linear regression model in Figure 2.1, $\alpha_0$ and $\alpha_1$ should be estimated. *Maximum Likelihood Estimation* and *Least-Squares Estimation* (*LSE*) could be used to estimate unknown regression parameters. However, the easiest method is least-squares (*LS*) method. We can clearly define the *n* observations in the sample by using the equation (2.1.2) as follows [57]

$$y_k = \alpha_0 + \alpha_1 X_k + \varepsilon_k \quad (k = 1, 2, ..., n), \tag{2.1.3}$$

and the deviation of *k*th value of *y* from *k*th predicted value is equal to $y_k - (\alpha_0 + \alpha_1 X_k)$. Then, for all of the *n* data points, the sum of the squares of the deviations of the observations from the true regression line is [62]

$$L(\alpha) = \sum_{k=1}^{n} \varepsilon_k^2 = \sum_{k=1}^{n} \left(y_k - (\alpha_0 + \alpha_1 X_k)\right)^2 . \tag{2.1.4}$$

The least squares estimators of $\alpha_0$ and $\alpha_1$ is defined as $\hat{\alpha}_0$ and $\hat{\alpha}_1$ which minimize *L*, and the prediction equation represented as

$$\hat{y} = \hat{\alpha}_0 + \hat{\alpha}_1 X . \tag{2.1.5}$$

The values of $\hat{\alpha}_0$ and $\hat{\alpha}_1$ should satisfy the two partial derivatives, $\partial L / \partial \alpha_0$ and $\partial L / \partial \alpha_1$, equal to zero. Here, *L*, hence, is the likelihood function. The equation (2.1.5) is called the **LS line**. Each of pair of observation satisfies the relationship [57, 62]

$$y_k = \alpha_0 + \alpha_1 X_k + e_k \ \ (k = 1, 2, ..., n),$$

where $e_k = y_k - \hat{y}_k$ is called the *k*th **residual**. The residual presents error in the fit of the model to the *k*th observation, $y_k$.

Many application of regression analysis contain situations that there exist more than one predictor variable in. A regression model which includes more than one predictor variable is called a *Multiple Linear Regression Model* and it can be represented as follows [57, 62]:

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n + \varepsilon. \tag{2.1.6}$$

The method of least squares may be used to estimate the unknown regression parameters in the equation (2.1.6).

### 2.1.2 Non-Linear Regression Models

If there is at least one nonlinear parameter in a model, this model is called as *A Non-Linear Model*. This means that, in a nonlinear model at least one derivative with respect to a parameter must include that parameter. Some examples for nonlinear regression models are as follows [74]:

$$\begin{aligned} Y(t) &= e^{at+bt^2}, \\ Y(t) &= at + e^{-bt}. \end{aligned} \tag{2.1.7}$$

Some examples for nonlinear functions are: exponential functions, logarithmic functions, trigonometric functions, power functions, Gaussian function, and Lorentzian curves. Some functions, such as the exponential or logarithmic functions are assumed to be linear because they can be transformed. Here, when transformed, standard linear regression may be performed but should be employed with caution [74].

#### 2.1.2.1 Non-Linear Growth Models

Those models which define the growth behavior over time are used in many areas. In the field of population biology, growth occurs in organisms, plants, animals, etc. [71]. The type of model which is needed in a specific situation relies on the type of growth that occurs. Generally, growth models are mechanistic in nature, rather than empirical.

*i) Malthus Model:*

The Malthusian model depends on two key ingredients: an agricultural production function that applies the fixed factor land and an income-population feedback where the population growth rate is an increasing function of income per capita. Consider the aggregate production function of the form [70, 74]:

$$Y_t = (A_t X)^{\alpha} N_t^{1-\alpha}, \tag{2.1.8}$$

where $Y_t$ denotes output in period $t$, $A_t$ is productivity, $X$ is the fixed amount of land, and $N_t$ is the size of the population. Dividing $Y_t$ by $N_t$ on both sides, we may see that the income per capita $y_t = \frac{Y_t}{N_t}$ is given by

$$y_t = \left( \frac{A_t X}{N_t} \right)^{\alpha}. \tag{2.1.9}$$

The equation implies that income per capita is an increasing function of productivity, but a decreasing function of population: when the size of the population increases, there is less land for each person to work with, which lowers income per capita.

*ii) Monomolecular Model:*

The monomolecular model defines the progress of a growth situation in which it is believed that the rate of growth at any time is proportional to the resources yet to be achieved [71],

$$\frac{dN}{dt} = r(K - N).$$

Here, $K$ is a carrying size of the system. Integrating both sides of the equation we get:

$$N(t) = K - (K - N_0)\exp(-rt),$$

where $N(0) = N_0$.

*iii) Logistic Model:*

The logistic model is described by the differential equation [71]

$$\frac{dN}{dt} = rN = r_0 N(1 - \frac{N}{K}),$$

which has the following solution:

$$N_t = \frac{N_0 K}{N_0 + (K - N_0)\exp(-r_0 t)}. \tag{2.1.11}$$

*iv) Gompertz Model:*

The gompertz model is described by the differential equation as follows [71]:

$$\frac{dN}{dt} = rN \log_e(K / N).$$

Integration of this equation yields

$$N(t) = K \exp(\log_e(N_0 / K)\exp(-rt)). \tag{2.1.12}$$

### 2.1.2.2 Fitting of Non-Linear Models

The models above are posed deterministically but this is unrealistic. Because of this, adding an error term on the right hand side and making an appropriate assumption about them, we replace these deterministic models by statistical models. This brings us a *Nonlinear Statistical Model*. In non-linear case, parameter estimates can also be constructed by the method of *LS* like linear regression. Minimization of RSS produces normal equations which are nonlinear in the parameters. It is not possible to solve nonlinear equations exactly. For this reason, the next alternative is to obtain approximate analytic solutions by using iterative procedures. For this approximate solution, three main methods are [70]:

i) *Linearization method,*

ii) *Steepest Descent method,*

iii) *Levenberg-Marquardt's method.*

The linearization method applies the results of least squares theory in a succession of stages but, neither this method nor the steepest descent method is ideal. The linearization method is converge very rapidly provided the vicinity of the true parameter values are reached, but if initial trial values are too far removed, convergence may not occur at all whereas the steepest descent method is able to converge on true parameter values even though initial trial values are far from the true parameter values [70]. However, this convergence tends to the very slow at the later stages of the iterative process.

The most widely applied method of computing non-linear least squares estimators is *Levenberg-Marquardt's* one. This method presents a compromise between the other two methods and combines successfully the best features of both and avoids their serious disadvantages. It is good in the sense that it almost always converges and does not 'slow down' at the latter part of the iterative process. The model is [70]

$$y_i = f(x_i, \theta) + \varepsilon_i \quad (i{=}1{,}2{,}...{,}n). \tag{2.1.13}$$

Here, $y_i$ is the $i^{th}$ observation of the dependent variable, $x_i$ is $i^{th}$ independent variable; $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_n)^T$ are parameters, $\varepsilon_i$; the error terms are independent and follow $N(0, \sigma^2)$ distribution. The residual sum of squares is given by [70]:

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - f(x_i, \boldsymbol{\theta}))^2, \tag{2.1.14}$$

where $\boldsymbol{\theta}_0 = (\theta_{10}, \theta_{20}, ..., \theta_{n0})^T$ is the vector of initial parameter values. The algorithm for constructing successive estimates is represented as follows:

$$(\boldsymbol{H} + \tau \boldsymbol{I})(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_1) = \boldsymbol{g},$$

where

$$\boldsymbol{g} = \left. \frac{\partial S(\theta)}{\partial \theta} \right|_{\theta = \theta_0}, \ \boldsymbol{H} = \left. \frac{\partial^2 S(\theta)}{\partial \theta \partial \theta^T} \right|_{\theta = \theta_0}. \tag{2.1.15}$$

Here, $\boldsymbol{I}$ is the identity matrix and $\tau$ is a suitable multiplier.

### 2.1.3 Generalized Linear Models

*Generalized Linear Models* (*GLMs*) are used in many application of prediction, in regression and classification. GLM, mathematical extensions of linear model, does not force data into unnatural scale and so, allows for non-linearity and non-constant variance structures in the data. If the normality and constant variance assumptions are not satisfied, this approach can be applied [40]. In GLM, it can be assumed that data are from several families of probability distributions, involving the normal, binomial, Poisson, negative binomial, or gamma distribution.

By using a link function, GLM makes it possible to search linear and nonlinear relationships between the mean of the response variable and the linear combination of the explanatory variables [40]. The mean value of a dependent variable rely on a linear predictor through a nonlinear link function and allows the response variable *Y*; its probability distribution to be any member of an exponential family of distributions that has the basic structure

$$\mu_k = g(\varsigma_k) = g(\boldsymbol{X}_k^T \boldsymbol{\alpha}), \tag{2.1.16}$$

where $\mu_k = E(Y_k)$ for $k = 1, 2, ..., n.$

Here, $g$ is the smooth link function, $n$ is the number of observation, $\boldsymbol{X}_k^T$ is the *k*th row of the model matrix $\boldsymbol{X}$, and $\boldsymbol{\alpha}$ is the vector of unknown parameters.

GLM generally makes the distribution assumptions which the response variable is independent and may have any distribution from an exponential density family (e.g. Gaussian, Poisson or binomial). The form of this model represented as follows [40, 92]:

$$f_\phi(y) = \exp\left( \frac{y\phi - b(\phi)}{a(\theta)} + c(y, \theta) \right). \tag{2.1.17}$$

Here, *b*, *a*, *c* are arbitrary functions, $\theta$ is an arbitrary, so-called scale parameter and $\phi$ is known as the canonical parameter of the distribution.

### 2.1.4 Generalized Partial Linear Models

A particular semiparametric model of interest is the *Generalized Partial Linear Model* (*GPLM*) that extends the GLM in that the usual parametric terms are enlarged by a nonparametric component. The GPLM model is given by [61, 83]

$$E(Y \,|\, \boldsymbol{X}, \boldsymbol{R}) = G(\boldsymbol{X}^T \boldsymbol{\alpha} + \beta(\boldsymbol{R})), \tag{2.1.18}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)^T$ is a finite dimensional parameter and $\beta(\cdot)$ is a smooth function which we try to estimate by *B*-splines. Here, $\boldsymbol{X}$ is an *n*-variable random vector that typically covers discrete covariables, and $\boldsymbol{R}$ is a *q*-variate random vector of continuous covariables to be modeled in a nonparametric way.

To estimate the GPLM by semiparametric maximum likelihood, we suppose that the first two conditional moments of *Y* may be specified as [61]

$$E(Y \,|\, \boldsymbol{X}, \boldsymbol{R}) = \mu = G(\boldsymbol{X}^T \boldsymbol{\alpha} + \beta(\boldsymbol{R})),$$

$$Var(Y \mid \boldsymbol{X}, \boldsymbol{R}) = \sigma^2 V(\mu),$$

and denotes by the log-likelihood function *L*. Here, maximization of *L*, which is shown as the composite form $L(\theta(\boldsymbol{\alpha}, \beta))$ to emphasize the roles of predictors, parameters, and of the unknown curve, is no longer suitable as a method of estimation. This leads to overfitting in the absence of any constraints on $\boldsymbol{\alpha}$. Indeed, it renders the parameters $\boldsymbol{\alpha}$ unidentifiable. However, if we place weak constraints on the form of $\beta$ by supposing that it is smooth, progress can be possible by maximizing instead a penalized version of log-likelihood. Therefore, we maximize the penalized log likelihood [61, 83]

$$\ell(\varsigma, y) := L(\theta(\boldsymbol{\alpha}, \beta)) - \frac{1}{2}\tau \int_a^b (\beta''(t))^2 \, dt,$$

where $H(\mu) := \varsigma(\boldsymbol{X}, \boldsymbol{R}) = \boldsymbol{X}^T \boldsymbol{\alpha} + \beta(\boldsymbol{R})$ and $G := H^{-1}$ is a link function that links the mean of the response variable to the predictors.

Here, $\ell$ presents the log-likelihood of the linear predictor and the second term is the penalizing part, and $\tau$ is a smoothing parameter. This parameter controls the trade-off between accuracy of the data fitting and its smoothness or complexity [21]. By smoothing, it is wanted to guarantee that the estimation is sufficiently robust with respect to noise in data and other forms of perturbation.

### 2.1.5 Nonparametric Regression Models

Nonparametric regression analysis traces the dependence of a response variable, *y*, on one or several predictors, $x_{ij}$ (*i*=1,2,…,*p*; *j*=1,2,…,*n*), without specifying in advance the function which relates the predictors to the response [33]:

$$E(y_k) = f(x_{1k}, x_2, ..., x_{pk}). \tag{2.1.19}$$

It is supposed that the conditional variance of *y*, Var $(y_k \mid x_{1k}, x_2, ..., x_{nk})$ is a constant, and that the conditional distribution of *y* is normal.

Nonparametric regression is differentiated from linear regression, in which the function relating the mean of *y* to the $x_i$ is linear in the parameters [33],

$$E(y_k) = \alpha_0 + \alpha_1 x_{1k} + \alpha_2 x_{2k} + ... + \alpha_n x_{nk} \tag{2.1.20}$$

and from traditional nonlinear regression, in which the function relating the mean of *y* to the $x_i$, though nonlinear in its parameters, is specified clearly,

$$E(y_k) = f(x_{1k}, x_{2k}, ..., x_{nk}; \alpha_1, \alpha_2, ..., \alpha_l). \tag{2.1.21}$$

In traditional regression analysis, whereas the purpose of classical parametric regression is to estimate the model parameters $\alpha$, the aim of the nonparametric regression is to directly estimate the regression function, *f*.

Global parametric modeling methods are relatively easy to improve and elucidate. However, they have a limited flexibility and work well only in cases where the true underlying relationship is close to the hypothesized function in the model [94]. In order to overcome the disadvantages of the global parametric approach, nonparametric models are developed locally over specific subregions of the data. Therefore, the data is analyzed for optimum number of subregions, and a simple function is optimally fit to the realization in each subregion.

Local regression with many predictors proceeds as follows, for instance. The fit $\hat{y}_0 = \hat{f}(\boldsymbol{x}_0)$ is wanted at the critical point $\boldsymbol{x}_0 = (x_{10}, x_{20}, ..., x_{n0})^T$ in the predictor space and the distances $D(\boldsymbol{x}_k, \boldsymbol{x}_0)$ are needed between the observations on the predictors and the critical point. If the predictors are on the same scale (as, for example, if they establish coordinates on a map), measuring distance is simple. Otherwise, some sort of standardization or generalized distance metric is necessitated [33]. Once distances are described, weighted polynomial fits in many predictors proceed much as in the bivariate case. Some kinds of spline estimators may also be generalized to higher dimensions.

The easiest use of nonparametric regression consists in smoothing scatterplots. Three widely applied methods of nonparametric regression are kernel estimation, local-polynomial regression that is a generalization of kernel estimation, and smoothing splines [33].

The generalization of nonparametric regression to many predictors is mathematically straightforward. However, it is often problematic in practice. First, multivariate data are affected by the so-called *curse of dimensionality*: Multidimensional spaces grow exponentially sparser with the number of dimensions, requiring very large samples to estimate nonparametric regression models with several predictors [33]. Second, it is difficult to visualize a regression surface in more than three dimensions (that is, for more than two predictors) though slicing the surface may be of some help.

Additive regression models are an alternative to unconstrained nonparametric regression with many predictors. This regression model has following form [33]:

$$E(y_k) = \gamma + f_1(x_{1k}) + f_2(x_{2k}) + \ldots + f_n(x_{nk}), \tag{2.1.22}$$

where $f_j$ are smooth partial-regression functions, estimated with smoothing splines or by local regression. Additive Model (AM) may be extended in two directions:

1. To include interactions among specific predictors; for instance,

$$E(y_k) = \gamma + f_1(x_{1k}) + f_{23}(x_{2k}, x_{3k}), \tag{2.1.23}$$

which is not as general as the unseparated model $E(y_k) = \gamma + f(x_{1k}, x_{2k}, x_{3k})$.

2. To include linear terms, as in the model

$$E(y_k) = \gamma + \alpha_1 x_{1k} + f_2(x_{2k}). \tag{2.1.24}$$

Such semiparametric models are particularly useful for containing dummy regressors or other contrasts derived from categorical predictors. There exist some other models such as projection-pursuit regression, Classification and Regression Trees (CART) and MARS. In MARS, functions are multiplicative nature and nonsmooth.

The main issue in nonparametric regression is the selection of smoothing parameters such as the span in kernel and local-polynomial regression or the roughness penalty in smoothing-spline regression or equivalent degrees of freedom for any of those [33]. The statistical balance is between variance and bias, and some methods such as Cross-Validation (CV) try to choose smoothing parameters to minimize estimated mean-square error, e.g., the sum of squared bias and variance.

### 2.1.6 Generalized Additive Models

*Generalized Additive Models* (*GAMs*) are semi-parametric extension forms of the class of GLMs. They differ from GLMs in that additive predictors take the place of linear predictors [40]. GAMs belong to modern techniques from statistical learning, and can be used in many fields of predictions. For identifying and characterizing nonlinear regression effects, they give more flexible statistical methods. GAMs are parameterized like GLMs, but some predictors may be modeled nonparametrically in addition to linear and polynomial terms for other predictors. The probability distribution of the response variable should still be specified, and in this respect, GAMs are parametric [40]. In this sense they are more suitable named semi-parametric models.

A crucial step in employing GAMs is to choose the suitable level of the *smoother* for a predictor. This is best achieved by specifying the level of smoothing applying the concept o effective degrees of freedom. A reasonable balance should be continued between the total number of observations and the total number of degrees of freedom used when fitting the model [40]. Having *j* covariates $x_1, x_2, ..., x_j$, comprised by the *j*-tuple $\boldsymbol{x} = (x_1, x_2, ..., x_j)^T$, and a response $y$ to the input $\boldsymbol{x}$ is assumed to have exponential family density $g_y(y, \beta, \vartheta)$ with the mean $\mu = E(y | x_1, x_2, ..., x_j)$ linked to the predictors through a link function

$$G = \frac{P_r(y-1|x)}{P_r(y-1|x)},$$

where $\beta$ is called the natural parameter and $\vartheta$ is the dispersion parameter. Link functions can be *logit* link function, the probit link function and identity link function. In a regression setting, GAMs have the form [40]

$$\varsigma(\boldsymbol{x}) = G(\mu) = \alpha_0 + \sum_{i=1}^{j} f_i(x_i), \tag{2.1.25}$$

where the functions $f_i$ are nonparametric and $\eta := (\alpha_0, f_1, f_2, ..., f_j)^T$ is the unknown entire parameter vector to be estimated. The including of $\alpha_0$ as an average outcome allows assuming $E(f_i(x_i)) = 0 \;\; (i = 1, 2, ..., j)$ [41].

## 2.2 Tikhonov Regularization

A problem is defined as *ill-posed problem* if a solution is not existing or not unique or if it is not stable under perturbation on data - that is, if an arbitrarily small perturbation of the data can cause an arbitrarily large perturbation of the solution [37]. TR is the most common and well-known form to make these problems regular and stable. For statistics, it is also known as *ride regression.*

TR method searches the regularized solution as a minimizer of a weighted combination of the residual norm and a side constraint [46]. The regularization parameter controls the weight given to the minimization of the side constraint. Therefore, the quality of the regularized solution is controlled by the regularization parameter. An optimal regularization parameter should fairly balance between the size of the residual error and the stabilizing of the approximate solution [46]. A suitable value of the regularization parameter is considered and computed when the norm of the error in the data or the norm of the solution of the error-free problem are available.

The regularization parameter brings the optimal rate of convergence for the approximations, which are generated by the application of TR to ill-posed equations [63]. However, when we derive rates of convergence, we must make assumptions about the nature of the stabilization (i.e., the choice of the semi norm in the TR) and the regularity imposed on the solution. In fact, there is a **trade-off** between stabilization and regularity in terms of the rate of convergence.

### 2.2.1 L-curve criterion

The *L-curve criterion* is a practical method for choosing regularization parameter when data are noisy. The method is based on the plot of the norm of the regularized solution versus the norm of the corresponding residual [38]. The idea of the L-curve criterion is to select a regularization parameter related to the characteristic *L-shaped corner* of the graph. The *corner* shows where the curve is closest to the origin and where the curvature is maximal. However, when it is plotted in linear scale, it is difficult to inspect the features of the L-curve because of the large range of values for the two norms. The features become easier to inspect when the curve is plotted in the double logarithmic scale [38]. Therefore, in many cases it is better to analyze the L-curve in the log-log scale. Figure 2.2 shows a typical L-curve.

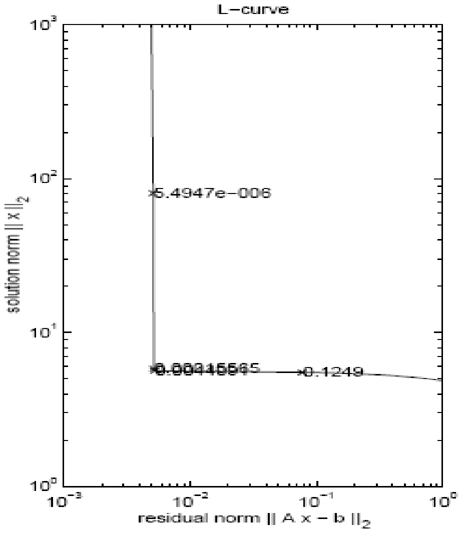


Figure 2.2 A typical L-curve for TR [39].

For TR, the L-curve is important in the analysis of discrete ill-posed problems. The L-curve shows how the regularized solution changes as the regularization parameter changes. The corner of the L-curve corresponds to a good balance between the minimization of the sizes, and the corresponding regularization parameter is a good one, because a distinct L-shaped corner of the L-curve is located exactly where the solution changes, from being dominated by the regularization errors to being dominated by right-hand side errors [46].

### 2.2.2 Choosing a Good Solution

Tikhonov solution can be expressed easily in terms of the *singular value decomposition* (SVD) of the coefficient matrix $\boldsymbol{A}$ of regarded linear systems of equations

$$\boldsymbol{Ax}-\boldsymbol{b},$$

where $\boldsymbol{A}$ is an ill-conditioned $(N\times m)$ matrix. The standard approach to approximately solve this system of equations is known as (*linear*) LS. It seeks to minimize the residual $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2^2$. There can be infinitely many solutions for a general linear LS problem. If it is considered that the data contain noise, in that situation, the data points cannot be fitted exactly because of noise. It becomes evident that there can be many solutions, which can adequately fit the data in the sense that the Euclidean distance $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ is smallest. The discrepancy principle [4] can be used to regularize the solution of a discrete ill-posed problem based on the assumption that a reasonable level for $\alpha=\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ is known.

Different kinds of TR are represented as minimization problems. Under the discrepancy principle, all solutions with $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2\leq\alpha$ are considered, and we select the one that minimizes the norm of $\boldsymbol{x}$:

$$\begin{aligned}&\underset{\boldsymbol{x}}{\text{minimize}} && \|\boldsymbol{x}\|_2\\ &\text{subject to} && \|\boldsymbol{Ax}-\boldsymbol{b}\|_2\leq\alpha,\end{aligned}\tag{2.2.1}$$

or we minimize the norm of residual vector under some tolerance with respect to the norm of $\boldsymbol{x}$:

$$\begin{aligned}&\underset{\boldsymbol{x}}{\text{minimize}} && \|\boldsymbol{Ax}-\boldsymbol{b}\|_2\\ &\text{subject to} && \|\boldsymbol{x}\|_2\leq\beta.\end{aligned}\tag{2.2.2}$$

In the first optimization problem (2.2.1), any important nonzero feature that appears in the regularized solution increases $\|\boldsymbol{x}\|_2$. However, these features exist in the solution because they are necessary to fit the data. Therefore, the minimum of $\|\boldsymbol{x}\|_2$ guarantee that unimportant features should be removed in the regularized solution. As $\alpha$ increases, the set of feasible models expands, and the minimum value of $\|\boldsymbol{x}\|_2$ decreases.

In the second optimization problem (2.2.2), it is wanted to choose the minimum norm solution among those parameter vectors, which adequately fit the data, because any important nonzero feature that appears in the regularized solution must not be ignored to fit the data, and unimportant data must be removed by the regularization. As $\beta$ decreases, the set of all feasible solutions becomes smaller, and the minimum value of increases.

There is also a third option which is considered a dampened LS problem:

$$\underset{\boldsymbol{x}}{\text{minimize}} \; \|\boldsymbol{Ax}-\boldsymbol{b}\|_2^2+\lambda^2\|\boldsymbol{x}\|_2^2, \tag{2.2.3}$$

arising when the method of Lagrange multipliers is applied to problem (2.2.2). Here, $\lambda$ is the tradeoff parameter between the first and the second part. The problems (2.2.1), (2.2.2) and (2.2.3) have the same solution for some appropriate choice of the values $\alpha,\beta$ and $\lambda$ [4].

To solve different kinds of TR problem discussed above, we use “SVD” to have a solution that minimizes the objective function including $\|\boldsymbol{x}\|_2$. However, in many cases, it is preferred to achieve a solution that minimizes some other measure of ***x***, such as the norm of first- or second-order derivatives. These derivatives are, in approximative sense, given by first- or second-order difference quotients of ***x***, considered as a function that is evaluated at the points *k* and *k*+1. These difference quotients approximate first- and second-order derivate; altogether, they are comprised by products ***Lx*** of ***x*** with matrices ***L***. These matrices represent the discrete differential operators of first- and second-order, respectively [4]. Hereby, the optimization problem is the following form:

$$\underset{\boldsymbol{x}}{\text{minimize}} \; \|\boldsymbol{Ax}-\boldsymbol{b}\|_2^2+\lambda^2\|\boldsymbol{Lx}\|_2^2 \; . \tag{2.2.4}$$

The optimization problem of (2.2.4) turns into the optimization problem of (2.23) when ***L=I***. Then, it is called **zeroth order** TR, which is a special case of (2.2.4).

Generally, (2.2.4) consists of high order TR problems. Although zeroth-order TR is solved by using SVD, to solve higher-order TR, **generalized SVD** is used. In many situations, to obtain a solution which minimizes some other measure $\boldsymbol{x}$, the norm of the first or second derivative is preferred. In **first-order** TR, the damped least- squares problem (2.2.4) is solved by using the $\boldsymbol{L}$ matrix:

$$\boldsymbol{L} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Here, $\boldsymbol{Lx}$ is the finite difference approximation to the first derivative of $\boldsymbol{x}$ and minimizing $\|\boldsymbol{Lx}\|_2$ penalizes solutions that are in a first-order derivative sense.

In **second-order** TR, the damped least-squares problem (2.2.4) is solved using the $\boldsymbol{L}$ matrix:

$$\boldsymbol{L} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}.$$

Here, $\boldsymbol{Lx}$ is the finite difference approximation proportional to the second derivative of $\boldsymbol{x}$, and minimizing $\|\boldsymbol{Lx}\|_2$ penalizes solutions that are in a second-order derivative sense. However, a different type of $\boldsymbol{L}$ (cf. Subsection 3.3) is used for our work. For all of these problems and matrices, *MATLAB Regularization Toolbox* can be used [39].

## 2.3 Multi-objective Optimization Problems

In general optimization problems, there is a single objective function and the aim is to find a solution which optimizes the objective function value subject to some constraints by using single-objective optimization method. Nevertheless, most real-world problems have several objectives, and decisions must be made by regarding these objective functions at the same time [77]. When an optimization problem includes more than one objective function, this problem is called as *Multi-Objective Optimization* (*MOO*) problem that has the task of finding one or more optimum solutions [24].

If optimization problems contain multiple objectives, we cannot use single-objective optimization methods. In fact, different objectives are commonly conflicting with each other. Therefore, a solution which performs well in one objective cannot do as good as in the other objectives [24]. There exist several solutions that do not perform suitably in all objectives. It is not clear which of these solutions are better until the decision maker computes them. A MOO problem can be written as following form [77]:

$$\text{minimize } \boldsymbol{Ax} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_p(\boldsymbol{x}))^T \text{ such that } \boldsymbol{x} \in X, \tag{2.3.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a feasible solution and $X$ is the set of all feasible solutions. In this problem, there are $p$ objective functions to be minimized and $\boldsymbol{A}$ is a ( $p \times n$ )-matrix. The $q$th row of $\boldsymbol{A}$ corresponds to the $q$th objective function, $f_q(\boldsymbol{x})$.

The point $\boldsymbol{y} = (y_1, y_2, ..., y_p)^T \in \mathbb{R}^p$ such that $\boldsymbol{y} = \boldsymbol{Ax}$ is the outcome of the solution $\boldsymbol{x} \in X$. The set $X$ is called decision space, and $Y = \{\boldsymbol{y} \in \mathbb{R}^p \mid \boldsymbol{y} = \boldsymbol{Ax}, \boldsymbol{x} \in X\}$ is called the objective (criterion) space. A point $\boldsymbol{x}$ is called to *dominate point* $\boldsymbol{x}'$ if and only if the corresponding $y_q \leq y_q{}'$ for all $q$ and $y_q < y_q{}'$ for at least one $q$. If there is no $\boldsymbol{x}' \in X$ such that $\boldsymbol{x}'$ dominates $\boldsymbol{x}$, then $\boldsymbol{x}$ is called *non-dominated* or *efficient*. The complete set of non-dominated solutions is also known as the pareto-optimal set.

## 2.4 Conic Optimization Problems

***i)* Primal Conic Problem (CP)**

A general primal conic optimization problem is a problem in the conic form

$$\begin{aligned} &\text{minimize } \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to } \boldsymbol{Ax} = \boldsymbol{b},\ \boldsymbol{x} \in K, \end{aligned} \tag{2.4.1}$$

where $K$ is a closed, pointed, non-empty, convex cone. In fact we assume that $K$ is product of the following cones [8]:

a) The *nonnegative orthant* $\mathbb{R}_+^m$ in $\mathbb{R}^n$.

b) The *Lorentz* (or *second order*, or *ice-cream*) *cone*:

$$L^m := \left\{ \boldsymbol{x} = (x_1, x_2, ..., x_m)^T \in \mathbb{R}^m \mid x_m \geq \sqrt{x_1^2 + x_2^2 + ... + x_{m-1}^2} \right\} \ (m \geq 2).$$

c) The *positive semidefinite cone*:

$$S_+^m := \left\{ \boldsymbol{A} \in \mathbb{R}^{m \times m} \;\middle|\; \boldsymbol{A} = \boldsymbol{A}^T,\ \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^m \right\}.$$

***ii)* Dual Conic Problem (CD)**

A general primal conic optimization problem is a problem in the conic form

$$\begin{aligned} &\text{maximize } \boldsymbol{b}^T \boldsymbol{y} \\ &\text{subject to } \boldsymbol{c} - \boldsymbol{A}^T \boldsymbol{y} \in K. \end{aligned} \tag{2.4.2}$$

(CD) is called the dual of (CP), the constraint is called *linear matrix inequality* (*LMI*) [8].

***iii)* Duality Theorem for Conic Optimization Problem**

(CP)
$$\begin{aligned} &\text{minimize } \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in K, \end{aligned}$$

(CD)
$$\begin{aligned} &\text{maximize } \boldsymbol{b}^T \boldsymbol{y} \\ &\text{subject to } \boldsymbol{c} - \boldsymbol{A}^T \boldsymbol{y} \in K. \end{aligned}$$

For conic problem, the properties of the duality are as follows [8]:

**1.** The value of the dual objective at every dual feasible solution is the value of the primal objective at every primal feasible solution (***weak duality***).

**2.** The following two properties are equivalent to each other:

(***i***) The primal is strictly feasible and below bounded,
(***ii***) The dual is solvable.

**3.** The following two properties are equivalent to each other:

(***iii***) The dual is strictly feasible and below bounded,
(***iv***) The primal is solvable.

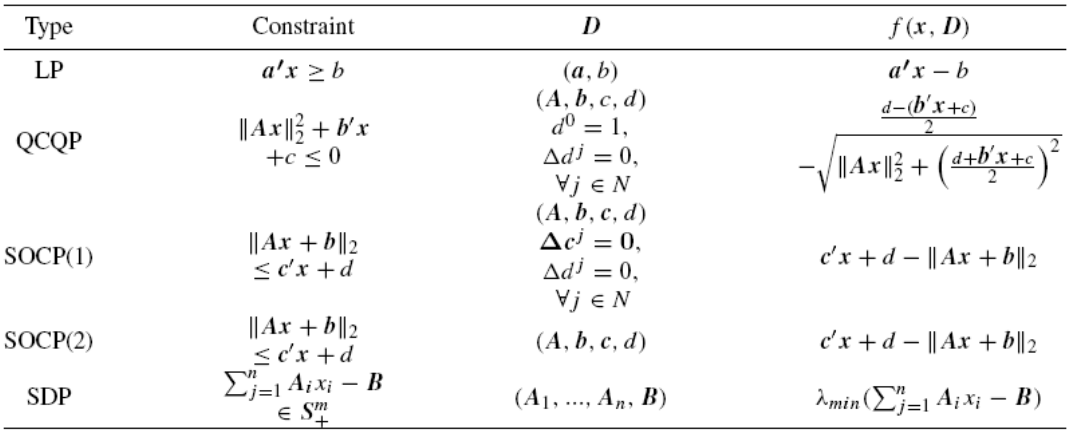
**Strong duality**: whenever (*i*) ≡ (*ii*) or (*iii*) ≡ (*iv*) is the case, the optimal values in the primal and the dual problems are equal to each other (strong duality):

Opt(CP) = Opt(CD).

**4.** The duality is symmetric: the problem dual to the dual is equivalent to the primal.

There are different conic optimization problems that are considered and coped with. Table 2.1 shows the function $f(x,\ D)$ for different conic optimization problems as follows:

Table 2.1 The function $f(x, D)$ for different conic optimization problems (for explanations cf. [14]).



| Type | Constraint | $D$ | $f(x, D)$ |
|---|---|---|---|
| LP | $a'x \geq b$ | $(a, b)$ | $a'x - b$ |
| QCQP | $\|Ax\|_2^2 + b'x$<br>$+c \leq 0$ | $(A, b, c, d)$<br>$d^0 = 1,$<br>$\Delta d^j = 0,$<br>$\forall j \in N$ | $\frac{d-(b'x+c)}{2}$<br>$-\sqrt{\|Ax\|_2^2 + \left(\frac{d+b'x+c}{2}\right)^2}$ |
| SOCP(1) | $\|Ax + b\|_2$<br>$\leq c'x + d$ | $(A, b, c, d)$<br>$\Delta c^j = 0,$<br>$\Delta d^j = 0,$<br>$\forall j \in N$ | $c'x + d - \|Ax + b\|_2$ |
| SOCP(2) | $\|Ax + b\|_2$<br>$\leq c'x + d$ | $(A, b, c, d)$ | $c'x + d - \|Ax + b\|_2$ |
| SDP | $\sum_{j=1}^{n} A_i x_i - B$<br>$\in S_+^m$ | $(A_1, ..., A_n, B)$ | $\lambda_{min}(\sum_{j=1}^{n} A_i x_i - B)$ |

In all these cases, these conic optimization problems can be solved efficiently by *an interior-point method*. Here, to find solution for conic optimization problem, MOSEK [58], SeDuMi [79], SDPT3 [85] can be used as a solver.

### 2.4.2 Solution Method for Conic Optimization Problems

Convex optimization problems such as semidefinite programming, geometric programming and, in particular, CQP problems are very important in data mining and classical *polynomial time algorithms* can be applied to solve these kinds of problems. However, these algorithms have some disadvantages because they use local information on the objective function and the constraints. Therefore, *Interior Point Methods* (*IPMs*) [66], firstly introduced by Karmarkar [42], are employed to solve "well-structured" convex problems like CQP problems.

In recent years, there has been comprehensive research into interior-point methods for linear optimization. One result of this research is the development of a primal-dual interior-point algorithm [44, 59] that is highly efficient both in theory and in practice [2, 53]. Therefore, some authors have studied how to generalize this algorithm to other problems. An important work in this direction is the paper of Nesterov and Todd [65] which presents that the primal-dual algorithm maintains its theoretical efficiency when the nonnegativity constraints are replaced by a convex cone as long as the cone is homogeneous and self-dual, or in the terminology of Nesterov and Todd, a self-scaled cone [3]. It has subsequently been indicated by Güler [36] that the only interesting cones having this property are direct products of $\mathbb{R}_+$, the quadratic cone and the cone of positive semi-definite matrices. For our work, we will mainly focus on conic quadratic optimization and an algorithm for this class of problems.

For conic quadratic optimization, some authors have already studied algorithms. In particular, Tuschiya [84], Monteiro and Tuschiya [60] have studied the complexity of different variants of the primal-dual algorithm. Schmieta and Alizadeh [73] have presented that many of the polynomial algorithms developed for semi-definite optimization may immediately be translated to polynomial algorithms for conic quadratic optimization [3]. Sturm [79] reports that his code SeDuMi may solve conic quadratic and semi-definite optimization problems. We take into account an optimization problem given by [3, 8]

$$\underset{x}{\text{minimize}}\ \ \boldsymbol{c}^T\boldsymbol{x}, \quad \text{where}\ \boldsymbol{x} \in \wp \subseteq \mathbb{R}^n.$$

Here, IPMs base on the interior point of the feasible set $\wp$. We suppose that this feasible set is closed and convex. An *interior penalty function (barrier)* $F(\boldsymbol{x})$ is selected, well defined, which is smooth and strongly convex, in the interior of $\wp$ and "blowing up" as a sequence from the interior int $\wp$ approaches a boundary point of $\wp$:

$$\boldsymbol{x}_r \in \operatorname{int} \wp\ (n \in \mathbb{N}_0),\ \lim_{r\to\infty} \boldsymbol{x}_r \in \partial\wp \quad \Rightarrow \quad F(\boldsymbol{x}_r) \to \infty\ (r \to \infty).$$

Now, we take into account one parametric family of functions generated by our objective and interior *penalty function* $F_p(\boldsymbol{x}) := p\boldsymbol{c}^T + F(\boldsymbol{x})$: int $\wp \to \mathbb{R}$. The *penalty parameter* $p$ is supposed to be nonnegative. Under mild regularity assumptions [3],

- every function $F_p(\cdot)$ attains its minimum over the interior of $\wp$, the minimizers $x_*(p)$ being unique;
- the central path $x_*(p)$ is a smooth curve, and all its limiting points (as $p \to \infty$), belong to the set of optimal solution of above optimization problem.

These algorithms have the advantage of applying the structure of the problem, of allowing better complexity bounds and exhibiting a much better practical performance. In the so-called *primal-dual* IPMs, both the primal and the dual problems and their variables are considered, the joint optimality conditions perturbed, parametrically solved and followed towards a solution along a *central path*.

## 2.5 Conic Quadratic Programming

Conic quadratic optimization is the problem of minimizing a linear objective function subject to the intersection of an affine set and the direct product of quadratic cones of the form

$$\left\{ \boldsymbol{x} \in \mathbb{R}^{m+1} \;\middle|\; {x_{m+1}}^2 \geq \sum_{j=1}^{m} x_j^2 \right\}.$$

The quadratic cone is also known as the second-order (Lorentz or ice-cream) cone. Many optimization problems may be created in this form. Some examples are linear, convex quadratic and convex quadratically constrained quadratic optimization. Other examples are the problem of minimizing the sum of norms and robust linear optimization [3]. Various applications of conic quadratic optimization are presented in [8, 51].

A *conic optimization problem* can be written as in (2.4.1):

$$\begin{aligned} &\text{minimize} \;\; \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \; \boldsymbol{x} \in K, \end{aligned} \tag{2.5.1}$$

associated with a cone *K* given as a direct product of *n* cones, each of them being either a semidefinite or a second-order (Lorentz) cone:

$$K := S_+^{m_1} \times \ldots \times S_+^{m_p} \times L^{m_p - 1} \times \ldots \times L^{m_n} \subseteq E := S_+^{m_1} \times \ldots \times S_+^{m_p} \times \mathbb{R}^{m_{p-1}} \times \ldots \times \mathbb{R}^{m_n}.$$

A *CQP* problem is a conic problem which can be presented as follows [8]:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \;\; \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to} \;\; \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \geq_K \boldsymbol{0}, \end{aligned} \tag{2.5.2}$$

for which the cone *K* is the direct product of several ice-cream cones. In case of CQP, there are no semidefinite factors $S_+^{m_i}$. Therefore, *K* can be represented in the following way:

$$K = L^{m_1} \times \ldots \times L^{m_r} \subseteq E.$$

In general, a CQP problem is an optimization problem with linear objective function and finitely many "ice-cream constraints"

$$\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i \geq_{L^{m_i}} \boldsymbol{0} \quad (i = 1, 2, ..., r).$$

Therefore, a CQP problem can be written as [8]:

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to } \ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \geq_{L^{m_i}} \boldsymbol{0} \quad (i = 1, 2, ..., r). \end{aligned} \qquad (2.5.3)$$

If we subdivide the data matrix, [ $\boldsymbol{A}_i, \boldsymbol{b}_i$ ], as follows:

$$[\boldsymbol{A}_i, \boldsymbol{b}_i] = \begin{bmatrix} \boldsymbol{D}_i & \boldsymbol{d}_i \\ \boldsymbol{p}_i^T & q_i \end{bmatrix},$$

where $\boldsymbol{D}_i$ is of the size ( $m_i - 1$) × dim $x$, the problem can be written as follows:

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad \boldsymbol{c}^T\boldsymbol{x}, \\ &\text{subject to} \quad \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\|_2 \leq \boldsymbol{p}_i^T\boldsymbol{x} - q_i \quad (i = 1, 2, ..., r). \end{aligned} \qquad (2.5.4)$$

This is the most explicit form that is used. In this form, $\boldsymbol{D}_i$ are matrices of the same row dimensions as $\boldsymbol{x}$, $\boldsymbol{d}_i$ are vectors of the same dimensions as the column dimensions of the matrices $\boldsymbol{D}_i$, $\boldsymbol{p}_i$ are vectors of the same dimensions as $\boldsymbol{x}$, and $q_i$ are real numbers [8].

## 2.6 Robust Optimization

In recent years, optimization has become a leading methodology in many fields. In these fields such as engineering, finance and control design, most applications assume complete knowledge of the data of the optimization problem. In other words, it is assumed that to develop a model, the input data are known exactly and equal to some nominal values. However, solutions to optimization problems may present a remarkable sensitivity to perturbations in the parameters of the problem, thus, often a computed solution is highly infeasible, suboptimal, or both. Therefore, optimization affected by parameter uncertainty is a focus of the mathematical programming community and a need to tackle uncertain data arises to develop models when optimization results are combined within real-life applications [13, 15].

There are some sources for the uncertainty: the data of the problem are not exactly known or may not be exactly measured, or the exact solution of the problem may not be implemented because of inherent inaccuracy of the devices [17]. The data uncertainty results in uncertain constraints and objective function. There are two principal methods, which have been proposed to deal with data uncertainty over years. These are *stochastic programming* [19, 43, 69] and *robust optimization* [10, 11].

RO has gained a lot of concentration both from a theoretical and practical point of view as a modeling framework for immunizing against parametric uncertainties in mathematical optimization. It is a modeling methodology to process optimization problems in which the data are uncertain and is only known to belong to some uncertainty set. Robust optimization purposes to find an optimal or near optimal solution that is feasible for every possible realization of the uncertain data [10, 90].

In the early 1970s, Soyster [76] was one of the first researchers to investigate explicit approaches to RO. This short note focused on robust linear optimization in the case where the column vectors of the constraint matrix were constrained to belong to ellipsoidal uncertainty sets. He suggested a linear optimization model to create a solution that was feasible for all input data such that each uncertain input data could take any value from an interval. However, this approach tended to find solutions that were over-conservative. Then Falk [32] followed this a few years later with more work on *inexact linear programs.* However, the optimization community was relatively quiet on the issue of robustness until the work of Ben-Tal and Nemirovski [5, 6, 7] and El Ghaoui et al. [27, 28] in the late 1990s. They coped with the over-conservatism of robust solutions by allowing the uncertainty ellipsoid sets for the data and suggested efficient algorithm to solve convex optimization problems under data uncertainty [15].

The robust optimization approach makes the optimization model robust regarding constraint violations by solving *robust counterparts* of these problems within prespecified uncertainty sets for the uncertain parameters [31]. These counterparts are solved for the *worst-case* realization of those uncertain parameters based on appropriately determined uncertainty sets for the random parameters. For a visualization see Figure 2.3:

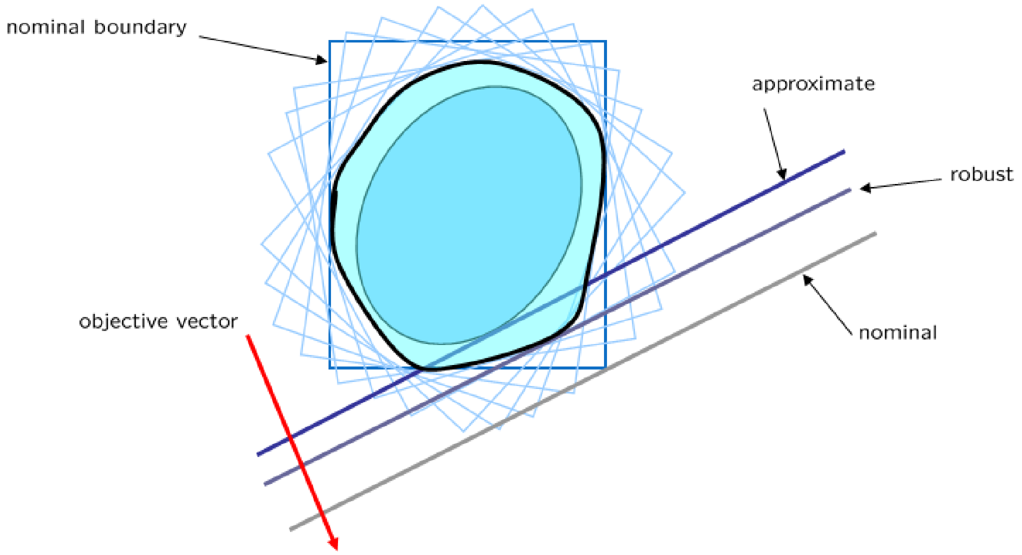


Figure 2.3 Approximating a robust solution [29].

In the figure above, the dark blue rectangle is the feasible set for optimization problem which has a solution of nominal values. When we have small perturbations in the data, our nominal solution jumps to other points. The light blue rectangles are the perturbed feasible sets. Solutions of these optimization problems can show sensitivity to perturbation in parameters of the problem. Therefore, we may need robust solutions. The blue area in the figure is the intersection for all scenarios and the feasible set for the robust optimization problem that has robust solution. But, this area is not a regular geometrical one. Since the robust counterpart can be much harder than original problem, we may need to approximate it. The robust optimization problem can be solved efficiently when we have a special shape. The regular set in the figure is the ellipsoid set and gives us the approximation of a robust solution.

### 2.6.1 The Approach

The general optimization problem under uncertainty is defined as follows:

$$\begin{aligned} &\text{maximize } \boldsymbol{\alpha}^T \boldsymbol{x} \\ &\text{subject to} \quad f_i(\boldsymbol{x},\ \boldsymbol{D}_i) \geq 0 \quad (i \in I), \\ &\qquad\qquad\quad \boldsymbol{x} \in X, \end{aligned} \tag{2.6.1}$$

where $f_i(\boldsymbol{x}, \boldsymbol{D}_i)$ $(i \in I)$ are given functions, $X$ is a given set and $\boldsymbol{D}_i$ $(i \in I)$ is the vector of random coefficient. Ben-Tal and Nemirovski [5, 6, 7] and independently by El Ghaoui et al. [27, 28] have taken a significant step forward for developing theory for robust optimization. They propose to solve the following robust optimization problem:

$$\begin{aligned} &\text{minimize } \max \ \boldsymbol{\alpha}^T \boldsymbol{x} \\ &\text{subject to} \quad f_i(\boldsymbol{x},\ \boldsymbol{D}_i) \geq 0, \ \ \forall\ (i \in I),\ \forall\ \boldsymbol{D}_i \in U_i\ , \\ &\qquad\qquad \boldsymbol{x} \in X, \end{aligned} \tag{2.6.2}$$

where $U_i$ $(i \in I)$ are given uncertainty sets. The motivation for solving problem (2.6.2) is to find a solution $\boldsymbol{x}^* \in X$ that "immunizes" problem (2.6.1) against parameter uncertainty. In the robust optimization literature, it is standard to consider uncertainty sets of the form

$$U = \{(\boldsymbol{A},\boldsymbol{b}) = (\boldsymbol{A}^0,\boldsymbol{b}^0) + \sum_{k=1}^{K} u^k (\boldsymbol{A}^k,\boldsymbol{b}^k) \ \big|\ \boldsymbol{u} \in Z\},$$

where the set $Z$ determines what type of uncertainty set we have. These sets may be:

convex combination of scenarios: $Z = \{\boldsymbol{u} \ \big|\ -1 \leq u_i \leq 1 \ \ (i = 1, 2, ..., k)\}$,

box uncertainty set: $Z = \{\boldsymbol{u} \ \big|\ \boldsymbol{u} \geq \boldsymbol{0}, e^T \boldsymbol{u} \leq 1\}$, and

ellipsoid uncertainty set: $Z = \{\boldsymbol{u} \ \big|\ \boldsymbol{u}^T \boldsymbol{u} \leq 1\}$.

### 2.6.2 Robust Optimization using Mathematical Programming

#### 2.6.2.1 Linear Programming (LP)

An uncertain LP constraint is of the form $\boldsymbol{\alpha}^T \boldsymbol{x} \geq \boldsymbol{b}$, for which $\boldsymbol{\alpha}$ and $\boldsymbol{b}$ are subject to uncertainty. If the corresponding uncertainty set *U* is a polyhedron, then the robust counterpart is also an LP (see Ben-Tal and Nemirovski [6, 7] and Bertsimas and Sim [12, 13]). On the other hand, if *U* is ellipsoidal, the robust counterpart is an SOCP. For linear programming, there exist probabilistic guarantees for feasibility available ([6, 7] and [12, 13]) under reasonable probabilistic assumptions on data variation [75].

#### 2.6.2.2 Quadratic Constrained Quadratic Programming (QCQP)

An uncertain QCQP constraint is of the form $\|\boldsymbol{A}\boldsymbol{x}\|_2^2 + \boldsymbol{b}^T \boldsymbol{x} + c \leq 0$, where ***A***, ***b*** and *c* are subject to data uncertainty. When the uncertainty set *U* is a simple ellipsoid, then the robust counterpart is an SDP. However, when the set is polyhedral, the robust counterpart is a NP-hard (Ben-Tal and Nemirovski [6, 7]). Here, there are no available probabilistic bounds [14, 75].

#### 2.6.2.3 Second-Order Cone Programming (SOCP)

In this class of problems, we are concerned with an uncertain SOCP constraint which is of the form $\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}\|_2 \leq \boldsymbol{c}^T\boldsymbol{x}+d$, where ***A***, ***b***, ***c*** and *d* are subject to data uncertainty. The robust counterpart is a Semidefinite Programming (SDP) when ***A***, ***b*** are in an ellipsoidal uncertainty set $U_1$ and ***c***, *d* are in another ellipsoidal set $U_2$. On the other hand, when ***A***, ***b***, ***c***, *d* vary together in a common ellipsoidal set, the problem has unknown complexity. Nemirovski [64] suggested a tractable approximation in the form of an SDP if ***c*** and *d* are deterministic and represented probabilistic guarantees in this case. However, there exist no available probability bounds to address the problem if ***c*** and *d* are stochastic [14].

#### 2.6.2.4 Semidefinite Programming (SDP)

An uncertain SDP constraint is of the form $\sum_{j=1}^{n}\boldsymbol{A}_j x_j \geq \boldsymbol{b}$, where $\boldsymbol{A}_j\ (j=1,2,...,n)$ and ***b*** are subject to data uncertainty. The robust counterpart is *NP*-hard for ellipsoidal uncertainty sets. Nemirovski [64] suggested a tractable approximation in the form of an SDP and presented probabilistic guarantees in this case [14, 75].

#### 2.6.2.5 Conic Programming

Here, we deal with an uncertain Conic Programming constraint which is of the form $\sum_{j=1}^{n}\boldsymbol{A}_j x_j \geq_K \boldsymbol{b}$, where $\boldsymbol{A}_j\ (j=1,2,...,n)$ and ***b*** are subject to data uncertainty. The cone *K* is closed, pointed and with a nonempty interior. There are no results available considering tractability and probabilistic guarantees in this case. In this work, we concentrate on conic optimization problems [14].

### 2.6.3 Robust Conic Optimization Problem

For all (or most) possible realizations of the data, the solution must satisfy the actual constraints despite of the data uncertainty. Such a solution is called a *robust feasible solution*. The problem of finding an optimal robust solution is called the *robust counterpart* of the original problem. In fact, it is the problem of minimizing the objective function over the set of robust feasible solutions. An uncertain conic problem has the following form [14]:

$$\begin{aligned} &\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \;\; \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to} \;\; \boldsymbol{A}_i\boldsymbol{x}+\boldsymbol{b}_i \in K_i \;\; (i=1,2,...,N), \end{aligned} \tag{2.6.3}$$

where $K_i$ $(i = 1,2,\ldots,N)$ are closed, pointed, non-empty, convex cones, and ***A***, ***b***, ***c*** are subject to data uncertainty.

It is necessary that the robust counterpart is *computationally tractable*, meaning solvable in polynomial time with respect to the problem size for robust optimization which is an applicable methodology for real life large scale problems [14]. Tractability of the robust counterpart depends on the original optimization problem and the uncertainty set which is regarded. In many cases, if the robust counterpart is not tractable, it is an NP-hard problem. Therefore, the need arises to find an approximation to the robust counterpart and to estimate its quality.

### 2.6.4 Robust Approximation

#### 2.6.4.1 Stochastic Robust Approximation

An approximation problem is taken into account with basic objective $\|\boldsymbol{Ax}-\boldsymbol{b}\|$, which has the form of a $\ell_1,\ell_2$ or $\ell_\infty$ norm, and regarding some uncertainty or possible variation in the data matrix ***A*** but it may be extended to overcome the case where both ***A*** and ***b*** have uncertainty [18]. For the variance in ***A***, some statistical models are also considered. It is assumed that ***A*** is random variable which takes values in $\mathbb{R}^{m\times n}$ with mean $\bar{\boldsymbol{A}}$. Therefore, ***A*** may be described as

$$\boldsymbol{A}=\bar{\boldsymbol{A}}+\boldsymbol{U}.$$

Here, ***U*** which defines its statistical variation is a random matrix with zero mean and $\bar{A}$ is the average value of ***A***. As the objective, the *expected value* of $\|\boldsymbol{Ax}-\boldsymbol{b}\|$ is [18]

$$\text{minimize} \;\; E\|\boldsymbol{Ax}-\boldsymbol{b}\|. \tag{2.6.4}$$

This problem is the *stochastic robust approximation problem* and always a convex optimization problem. However, it is generally not tractable because it is very difficult to evaluate the objective or its derivatives in most cases. As a simple case, the problem (2.6.4) may be solved when we assume that ***A*** has only a finite number of values, that is,

$$\text{prob}(\boldsymbol{A}=A_i)=p_i \qquad (i=1,2,...,k).$$

Here, $\boldsymbol{A}_i \in \mathbb{R}^{n\times m}$, $\boldsymbol{p} \geq 0$ and $\mathbf{1}^T \boldsymbol{p} = 1$ where $\mathbf{1} = (1,1,...,1)^T$. So the problem (2.6.4) has the following form:

$$\text{minimize } p_1 \left\|\boldsymbol{A}_1\boldsymbol{x}-\boldsymbol{b}\right\| + p_2 \left\|\boldsymbol{A}_2\boldsymbol{x}-\boldsymbol{b}\right\| + ... + p_k \left\|\boldsymbol{A}_k\boldsymbol{x}-\boldsymbol{b}\right\|. \qquad (2.6.5)$$

It is generally called a *sum-of-norms problem* and can be denoted as [18]

$$\begin{aligned} &\text{minimize } \ \boldsymbol{p}^T\boldsymbol{t} \\ &\text{subject to } \left\|\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}\right\| \leq t_i \quad (i = 1,2,...,k), \end{aligned} \qquad (2.6.6)$$

where the variables are $x \in \mathbb{R}^n$ and $t \in \mathbb{R}^k$. When the norm $\|\cdot\|$ is the $\ell_1$- or $\ell_\infty$- norm, the sum-of-norms problem is an LP. On the other hand, when the norm is the Euclidean norm, this problem can be represented as an SOCP.

On the statistical robust approximation problem, some variations are tractable. For example, let us consider the statistical robust least-squares problem [18]

$$\text{minimize } E\left\|\boldsymbol{Ax}-\boldsymbol{b}\right\|_2^2,$$

where the norm is Euclidean norm. Therefore the objective function may be expressed as

$$\begin{aligned} E\left\|\boldsymbol{Ax}-\boldsymbol{b}\right\|_2^2 &= E(\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b}+\boldsymbol{Ux})^T(\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b}+\boldsymbol{Ux}) \\ &= (\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b})^T(\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b}) + E\boldsymbol{x}^T\boldsymbol{U}^T\boldsymbol{Ux} \\ &= \left\|\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b}\right\|_2^2 + \boldsymbol{x}^T\boldsymbol{Px}, \end{aligned}$$

where $\boldsymbol{P} = \boldsymbol{U}^T\boldsymbol{U}$. Consequently, with solution $\boldsymbol{x} = (\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}}+\boldsymbol{P})^{-1}\bar{\boldsymbol{A}}^T\boldsymbol{b}$, this statistical robust approximation problem has the following form of a *regularized* least-squares problem:

$$\text{minimize } \left\|\bar{\boldsymbol{A}}\boldsymbol{x}-\boldsymbol{b}\right\|_2^2 + \left\|\boldsymbol{P}^{1/2}\boldsymbol{x}\right\|_2^2. \qquad (2.6.7)$$

This observation gives us another interpretation of the Tikhonov regularized least-squares problem (2.1.3), as a robust least-squares problem, regarding possible variation in the matrix $\boldsymbol{A}$. The solution of the Tikhonov regularized least-squares problem minimizes $E\left\|(\boldsymbol{A}+\boldsymbol{U})\boldsymbol{x}-\boldsymbol{b}\right\|^2$, where $U_{ij}$ (*i*=1,2,…,*m*; *j*=1,2,…,*n* ) are zero mean, uncorrelated random variables with variance are $\delta$ and $\boldsymbol{A}$ is deterministic [18].

### 2.6.4.2 Worst-Case Robust Approximation

Using *worst-case approach* and based on a set, it is possible to model the variation in the matrix ***A***. The uncertainty is defined by a set of possible values for ***A*** [18]:

$$\boldsymbol{A} \in Z \subseteq \mathbb{R}^{mxn},$$

which is assumed nonempty and bounded. The associated *worst-case error* of a candidate approximate solution $\boldsymbol{x} \in \mathbb{R}^n$ is described as

$$e_{wc}(\boldsymbol{x}) = \sup\left\{\|\boldsymbol{Ax} - \boldsymbol{b}\| \mid \boldsymbol{A} \in Z\right\}, \tag{2.6.8}$$

that is always a convex function of ***x***. The aim of *worst-case robust approximation problem* is to minimize the worst-case error:

$$\text{minimize} \ \ \sup\left\{\|\boldsymbol{Ax} - \boldsymbol{b}\| \mid \boldsymbol{A} \in Z\right\}, \tag{2.6.9}$$

where the variable is ***x***, the problem data are ***b*** and the set *Z*. If *Z* is the singleton, *Z*= {***A***}, the robust approximation problem (2.6.9) can be represented as a basic norm approximation problem [18]. The robust approximation problem is a convex optimization problem. However, its tractability depends on the description of the uncertainty set *Z* and the norm used.

The robust approximation problem (2.6.9) exists in many contexts and applications. In an estimation case, the set *Z* gives the uncertainty in the linear relation between the vector to be estimated and the measurement vector [18]. On the one hand, the noise term ***v*** in the model $\boldsymbol{y} = \boldsymbol{Ax} + \boldsymbol{v}$ is called *additive noise* or *additive error*, because it is added to the measurement $\boldsymbol{Ax}$. On the other hand, the variation in ***A*** is called a *multiplicative error*, because it multiplies the variable ***x***.

In an optimal design case, the variation may describe uncertainty of the linear equations that relate the design variables ***x*** to the results vector $\boldsymbol{Ax}$. Then, the robust approximation problem (2.6.9) is represented as the robust design problem: find design variables ***x*** which minimize the worst possible mismatch between $\boldsymbol{Ax}$ and ***b***, over all possible values of ***A*** [18].

### 2.6.4.3 Uncertainty sets

Robust optimization needs problems to remain feasible for any values of the uncertain parameters within the uncertainty set which is specified for problems [31]. These uncertainty sets are typically based on statistical estimates and probabilistic guarantees on the solution. If the uncertainty set has a special shape, then the robust optimization problem can be solved efficiently [14]. These special shapes for uncertainty sets may be *ellipsoidal* or *polyhedral*.

### 2.6.4.3.1 Polyhedral Uncertainty

In polyhedral uncertainty, we have $Z=\{\boldsymbol{A}_1,\boldsymbol{A}_2...,\boldsymbol{A}_k\}$, and the robust approximation problem is given by [18]

$$\text{minimize} \;\; \max_{i=1,2,...,k} \; \|\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}\|. \tag{2.6.10}$$

The problem above is equivalent to the robust approximation problem with the *polyhedral set* $Z=\text{conv}\{\boldsymbol{A}_1,\boldsymbol{A}_2,...,\boldsymbol{A}_k\}$:

$$\text{minimize} \;\; \sup\,\{\,\|\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}\| \;|\, \boldsymbol{A} \in \text{conv}\{\boldsymbol{A}_1,\boldsymbol{A}_2,...,\boldsymbol{A}_k\}\}\,.$$

The problem can be put in epigraph form as [18]

$$\begin{aligned}&\text{minimize} \;\; t\\ &\text{subject to} \;\; \|\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}\|\le t \;\; (i=1,2,...,k).\end{aligned} \tag{2.6.11}$$

This can be solved in a variety of ways, depending on the norm used. When the norm is the Euclidean norm, it is an SOCP. When the norm is the $l_1-$ or the $l_\infty-$norm, it is an LP. Since we deal with CQP (SOCP), we will use the Euclidean norm for this study.

**Epigraph Problem Form:**

To define the problem of finding an *x* which minimize $f_0(\boldsymbol{x})$ for all $\boldsymbol{x}$ that satisfy the conditions $f_k(\boldsymbol{x})\le 0 \;\; (k=1,2,...,m)$ and $g_k(\boldsymbol{x})=0 \;\; (k=1,2,...,n)$, the notation of standard problem is of the following form [18]:

$$\begin{aligned}&\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} && f_0(\boldsymbol{x}),\\ &\text{subject to} && f_k(\boldsymbol{x})\le 0 \;\; (k=1,2,...,m),\\ & && g_k(\boldsymbol{x})=0 \;\;\; (k=1,2,...,n).\end{aligned} \tag{2.6.12}$$

The *epigraph form* of the standard problem is the following problem:

$$\begin{aligned} &\text{minimize} && t \\ &\text{subject to} && f_0(\boldsymbol{x}) - t \le 0, \\ &&& f_k(\boldsymbol{x}) \le 0 \quad (k = 1, 2, ..., m), \\ &&& g_k(\boldsymbol{x}) = 0 \quad (k = 1, 2, ..., n), \end{aligned} \tag{2.6.13}$$

with variables $t \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^n$. The standard problem and its epigraph form is equal to each other. The epigraph form problem may be explained geometrically in Figure 2.4:

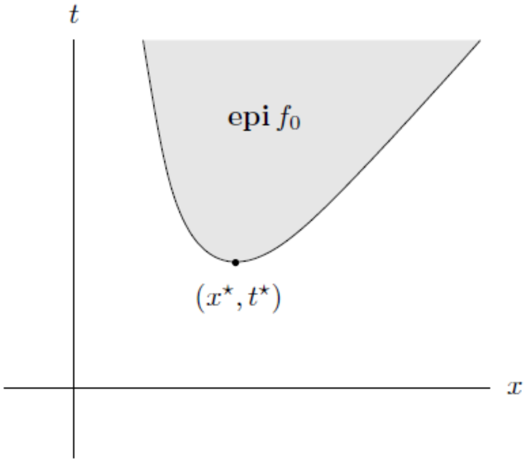


Figure 2.4 Geometric explanation of epigraph form problem [18].

#### 2.6.4.3.2 Norm Bound Error

Here, the uncertainty set $Z$ is a norm ball, $Z = \{\bar{\boldsymbol{A}} + \boldsymbol{U} \mid \|\boldsymbol{U}\| \le \alpha\}$, where $\|\cdot\|$ is a norm on $\mathbb{R}^{m\times n}$. In this case, we have the following form

$$e_{wc}(\boldsymbol{x}) = \sup\{\|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b} + \boldsymbol{U}\boldsymbol{x}\| \mid \|\boldsymbol{U}\| \le \alpha\}, \tag{2.6.14}$$

which should be carefully explained because the norm $\|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b} + \boldsymbol{U}\boldsymbol{x}\|$ is applied to measure the size of the residual and it is on $\mathbb{R}^m$ while, the second norm $\|\boldsymbol{U}\|$ is used to define the norm ball $Z$ and it is on $\mathbb{R}^{m\times n}$ [18]. This expression for $e_{wc}(\boldsymbol{x})$ may be simplified in many cases. For example, the *Euclidean norm* on $\mathbb{R}^n$ and the associated *induced norm* on $\mathbb{R}^{m\times n}$ can be used, e.g., the maximum singular value. When $\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b} \ne 0$ and $\boldsymbol{x} \ne 0$, for $e_{wc}(\boldsymbol{x})$ in the expression, the supremum is reached for $\boldsymbol{U} = \alpha u v^T$, with [18]

$$\boldsymbol{u} = \frac{\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}}{\|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}\|_2} \quad \text{and} \quad \boldsymbol{v} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}. \tag{2.6.15}$$

Then, the resulting worst-case error is given by

$$e_{wc}(\boldsymbol{x}) = \|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}\|_2 + \alpha\|\boldsymbol{x}\|_2. \tag{2.6.16}$$

It is easily confirmed that this expression is also valid when $\boldsymbol{x}$ or $\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}$ is zero. Now, the robust approximation problem (2.6.9) is

$$\text{minimize} \ \ \|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}\|_2 + \alpha\|\boldsymbol{x}\|_2. \tag{2.6.17}$$

This problem is a regularized norm problem and is solved as the SOCP [18]

$$\begin{aligned} &\text{minimize} \quad t_1 + at_2 \\ &\text{subject to} \ \ \|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}\|_2 \le t_1, \quad \|\boldsymbol{x}\|_2 \le t_2. \end{aligned} \tag{2.6.18}$$

Here, the solution of this problem is the same as the solution of the regularized least-squares problem. Therefore, another form of the regularized least-squares problem can be represented as a worst-case robust approximation problem [18]

$$\text{minimize} \ \ \|\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2 \tag{2.6.19}$$

for some value of the regularization parameter $\lambda$.

#### 2.6.4.3.3 Ellipsoidal Uncertainty

The variation in $\boldsymbol{A}$ may be defined by giving an ellipsoid of possible values for each row [18]:

$$U = \left\{ [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_m]^T \mid \boldsymbol{a}_i \in Z_i \ \ (i = 1, 2, ..., m) \right\},$$

where

$$Z_i = \left\{ \bar{\boldsymbol{a}}_i + \boldsymbol{P}_i\boldsymbol{u} \mid \|\boldsymbol{u}\|_2 \le 1 \right\} \quad (i = 1, 2, ..., m).$$

For each column index values, the matrix $\boldsymbol{P}_i \in \mathbb{R}^{n\times n}$ represent the variation in $\boldsymbol{a}_i$. When the variation in $\boldsymbol{a}_i$ is limited to a subspace, $\boldsymbol{P}_i$ is allowed to have a nontrivial nullspace in order to model the situation. As an extreme case, $\boldsymbol{P}_i = 0$ is taken when $\boldsymbol{a}_i$ has no uncertainty. With this ellipsoidal uncertainty type, an explicit expression for the worst-case magnitude of each residual can be described as [18]

$$\begin{aligned}\sup_{\boldsymbol{a}_i \in Z_i} \left|\boldsymbol{a}_i^T \boldsymbol{x} - \boldsymbol{b}_i\right| &= \sup\left\{\left|\bar{\boldsymbol{a}}_i^T \boldsymbol{x} - b_i + (\boldsymbol{P}_i \boldsymbol{u})^T \boldsymbol{x}\right| \;\middle|\; \|\boldsymbol{u}\|_2 \le 1\right\} \\ &= \left|\bar{\boldsymbol{a}}_i^T \boldsymbol{x} - b_i\right| + \left\|\boldsymbol{P}_i^T \boldsymbol{x}\right\|_2 .\end{aligned}$$

Several robust approximations can be solved problems by using this result. For example, the robust $\ell_2$-norm approximation problem

$$\text{minimize } \; e_{wc}(\boldsymbol{x}) = \sup\left\{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \;\middle|\; \boldsymbol{a}_i \in Z_i \quad (i = 1, 2, ..., m)\right\} \tag{2.6.20}$$

can be written as an SOCP. For the worst-case error, an explicit expression is represented as the following form:

$$e_{wc}(\boldsymbol{x}) = \left(\sum_{i=1}^{m}\left(\sup_{a_i \in Z_i} \left|\boldsymbol{a}_i^T \boldsymbol{x} - b_i\right|\right)^2\right)^{1/2} = \left(\sum_{i=1}^{m}\left(\left|\bar{\boldsymbol{a}}_i^T \boldsymbol{x} - b_i\right| + \left\|\boldsymbol{P}_i^T \boldsymbol{x}\right\|_2\right)^2\right)^{1/2}, \tag{2.6.21}$$

where $\bar{\boldsymbol{a}}_i$ is a vector where "$\sup\limits_{\boldsymbol{a}_i \in Z_i}$" is attained.

To minimize $e_{wc}(\boldsymbol{x})$, the problem

$$\begin{aligned}&\text{minimize } \; \|\boldsymbol{t}\|_2 \\ &\text{subject to } \; \left|\bar{\boldsymbol{a}}^T \boldsymbol{x} - b_i\right| + \left\|\boldsymbol{P}_i^T \boldsymbol{x}\right\|_2 \le t_i \quad (i = 1, 2, ..., m)\end{aligned}$$

can be solved where we introduced new variables $t_1, t_2, ..., t_m$. This problem may be formulated as follows where $\boldsymbol{t} = (t_1, t_2, ..., t_m)^T$ [18]:

$$\begin{aligned}&\text{minimize } \; \|t\|_2 \\ &\text{subject to } \; \bar{\boldsymbol{a}}_i^T \boldsymbol{x} - b_i + \left\|\boldsymbol{P}_i^T \boldsymbol{x}\right\|_2 \le t_i \quad (i = 1, 2, ..., m), \\ &\qquad\qquad\quad \bar{\boldsymbol{a}}_i^T \boldsymbol{x} + b_i + \left\|\boldsymbol{P}_i^T \boldsymbol{x}\right\|_2 \le t_i \quad (i = 1, 2, ..., m).\end{aligned} \tag{2.6.22}$$

This problem is an SOCP when put in epigraph form.

The main operations are needed to cope with ellipsoidal uncertainty such as affine-linear transformations, sums and intersections (fusions) of ellipsoids [48]. The family of ellipsoids in $\mathbb{R}^n$ is closed with respect to affine-linear transformations. However, the sum and the intersection of ellipsoids are not generally ellipsoidal. Because of this, both should be approximated by ellipsoidal sets. An ellipsoid in $\mathbb{R}^n$ is parameterized as [48]

$$Z(\boldsymbol{c},\boldsymbol{P}) = \{\boldsymbol{P}^{1/2}\boldsymbol{u}+\boldsymbol{c} \mid \ \|\boldsymbol{u}\|_2 \leq 1\}, \qquad (2.6.23)$$

where center $\boldsymbol{c} \in \mathbb{R}^n$, symmetric non-negative definite configuration matrix $\boldsymbol{P} \in \mathbb{R}^{nxn}$ and $\boldsymbol{P}^{1/2}$ is any square root satisfying $\boldsymbol{P}^{1/2}(\boldsymbol{P}^{1/2})^T = \boldsymbol{P}$.

The eigenvectors of $\boldsymbol{P}$ point in the directions of principal semiaxes of $\boldsymbol{Z}$. The lengths of the semiaxes of the ellipsoid $Z(\boldsymbol{c},\boldsymbol{P})$ are given by $\sqrt{\lambda_i}$, where $\lambda_i$ are the eigenvalues of $\boldsymbol{P}$ for $i$=1,2,…, $p$ [48]. The volume of the ellipsoid $Z(\boldsymbol{c},\boldsymbol{P})$ is given by vol $Z(\boldsymbol{c},\boldsymbol{P})$ $=V_p\sqrt{\det(\boldsymbol{P})}$,

$$V_p = \begin{cases} \dfrac{\pi^{p/2}}{(p/2)!} & \text{, for even } p, \\[2ex] \dfrac{2^p \pi^{(p-1)/2}((p-1)/2)!}{p!} & \text{, for odd } p. \end{cases} \qquad (2.6.24)$$

***i*) Affine Transformation**

The family of ellipsoids is closed with respect to *affine transformations*. Given an ellipsoid $Z(\boldsymbol{c},\boldsymbol{P}) \subset \mathbb{R}^n$, a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$, it is found that $\boldsymbol{A}\,Z(\boldsymbol{c},\boldsymbol{P})+$ $\boldsymbol{b}$ = $Z(\boldsymbol{Ac}+\boldsymbol{b},\boldsymbol{APA}^T)$ [48]. Therefore, ellipsoids are preserved under affine transformation. The affine transformation is called a *projection* when the rows of $\boldsymbol{A}$ are linearly independent, which means $m \leq n$, and $\boldsymbol{b}$ = 0 [50].

***ii*) Sums of two Ellipsoids**

Given two non-degenerate ellipsoids $Z_1 = Z(\boldsymbol{c}_1,\boldsymbol{P}_1)$ and $Z_2 = Z(\boldsymbol{c}_2,\boldsymbol{P}_2)$, their *geometric sum* $Z_1+Z_2 = \{\boldsymbol{D}_1+\boldsymbol{D}_2 \mid \boldsymbol{D}_1 \in Z_1, \boldsymbol{D}_2 \in Z_2\}$ is not usually an ellipsoid but, it may be tightly approximated by parameterized families of external ellipsoids. The range of values of $Z_1+Z_2$ is included in the ellipsoid [48]

$$Z_1 \oplus Z_2 := Z(\boldsymbol{c}_1 + \boldsymbol{c}_2, \boldsymbol{P}(s)), \tag{2.6.25}$$

where, for all, $s > 0$,

$$\boldsymbol{P}(s) := (1 + s^{-1})\boldsymbol{P}_1 + (1 + s)\boldsymbol{P}_2. \tag{2.6.26}$$

An additional condition has to be fulfilled for a *minimal* and *unique* external ellipsoidal approximation. The value of $s$ is generally selected to minimize either the trace or the determinant of $\boldsymbol{P}(s)$. When it is chosen

$$s := \frac{(Tr\,\boldsymbol{P}_1)^{1/2}}{(Tr\,\boldsymbol{P}_2)^{1/2}},$$

this value describes the ellipsoid including the sum that has minimal trace, or, sum of squares of semiaxes [48]. It is considered that the minimum trace calculation may be applied in case of degenerate ellipsoids [30, 49, 50].

### *iii*) Sums of *K* Ellipsoids

Given $K$ bounded ellipsoids of $\mathbb{R}^n$, $Z_k = Z(\boldsymbol{c}_k, \boldsymbol{P}_k)$ ($k$ = 1,2,…,$K$). The notion of the minimal trace ellipsoid is adapted from [26], and represented as the outer ellipsoidal approximation $Z(\sigma, R) = \oplus_{k=1}^{K} Z_k$ including the sum $S = \sum_{k=1}^{K} Z_k$ of ellipsoids, which is determined by [48]

$$\sigma := \sum_{k=1}^{K} c_k$$

and

$$R := \left(\sum_{k=1}^{K} \sqrt{Tr\boldsymbol{P}_k}\right)\left(\sum_{k=1}^{K} \frac{\boldsymbol{P}_k}{\sqrt{Tr\boldsymbol{P}_k}}\right).$$

### *iv*) Intersection of Ellipsoids

When the intersection of two ellipsoids is commonly not an ellipsoid, this set is replaced by the outer ellipsoidal approximation of minimal volume. The notion of *fusion* of ellipsoids is adapted from [72]. Given two non-degenerate ellipsoids $Z_1 = Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z_2 = Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$ in $\mathbb{R}^n$ with $Z(\boldsymbol{c}_1, \boldsymbol{P}_1) \cap Z(\boldsymbol{c}_2, \boldsymbol{P}_2) \neq 0$, an ellipsoid is described as follows [48]:

$$Z_\lambda(\boldsymbol{c}_0, \boldsymbol{P}_0) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \lambda(\boldsymbol{x} - \boldsymbol{c}_1)^T \boldsymbol{P}_1^{-1}(\boldsymbol{x} - \boldsymbol{c}_1) + (1-\lambda)(\boldsymbol{x} - \boldsymbol{c}_2)^T \boldsymbol{P}_2^{-1} \le 1\},$$

where $\lambda \in [0,1]$.

The ellipsoid $Z_\lambda(\boldsymbol{c}_0, \boldsymbol{P}_0)$ is the same $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$ for $\lambda = 1$ and $\lambda = 0$, respectively. To define a tight external ellipsoidal approximation $Z_\lambda(\boldsymbol{c}_0 \boldsymbol{P}_0)$ of the intersection of $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$, we represent as [48]

$$\boldsymbol{T} := \lambda \boldsymbol{P}_1^{-1} + (1-\lambda)\boldsymbol{P}_2^{-1}$$

and

$$\tau := 1 - \lambda(1-\lambda)\ (\boldsymbol{c}_2 - \boldsymbol{c}_1)^T \boldsymbol{P}_2^{-1}\boldsymbol{T}^{-1}\boldsymbol{P}_1^{-1}\ (\boldsymbol{c}_2 - \boldsymbol{c}_1).$$

The ellipsoid $Z_\lambda(\boldsymbol{c}_0, \boldsymbol{P}_0)$ is given by the center

$$\boldsymbol{c}_0 = \boldsymbol{T}^{-1}(\lambda \boldsymbol{P}_1^{-1}\boldsymbol{c}_1 + (1-\lambda)\boldsymbol{P}_2^{-1}\boldsymbol{c}_2)$$

and configuration matrix

$$\boldsymbol{P}_0 = \tau \boldsymbol{T}^{-1}.$$

The *fusion* of $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$, whose intersection is a nonempty bounded region, is determined as the ellipsoid $Z_\lambda(\boldsymbol{c}_0 \boldsymbol{P}_0)$ for the value $\lambda \in [0,1]$ that minimizes its volume [72]. If $Z(\boldsymbol{c}_1, \boldsymbol{P}_1) \subset Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$, the fusion of $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$ is $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$. Similarly, if $Z(\boldsymbol{c}_2, \boldsymbol{P}_2) \subset Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$, the fusion of $Z(\boldsymbol{c}_1, \boldsymbol{P}_1)$ and $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$ is $Z(\boldsymbol{c}_2, \boldsymbol{P}_2)$. Otherwise, it is $Z_\lambda(\boldsymbol{c}_0 \boldsymbol{P}_0)$ described as above where $\lambda$ is the only root in (0,1) of the following polynomial of degree $2p - 1$ [48]:

$$\begin{aligned} &\tau(\det \boldsymbol{T})\ \mathrm{Tr}\ (\mathrm{co}(\boldsymbol{T})\ (\boldsymbol{P}_1^{-1} - \boldsymbol{P}_2^{-1})) - p(\det \boldsymbol{T})^2 \\ &\times (2\boldsymbol{c}_0^T \boldsymbol{P}_1^{-1}\boldsymbol{c}_1 - 2\boldsymbol{c}_0^T \boldsymbol{P}_2^{-1}\boldsymbol{c}_2 + \boldsymbol{c}_0^T(\boldsymbol{P}_2^{-1} - \boldsymbol{P}_1^{-1})\boldsymbol{c}_0 - \boldsymbol{c}_1^T \boldsymbol{P}_1^{-1}\boldsymbol{c}_1 + \boldsymbol{c}_2^T \boldsymbol{P}_2^{-1}\boldsymbol{c}_2) = 0, \end{aligned}$$

where $\mathrm{co}(\chi)$ is the matrix of *cofactors* of $\chi$. Since $\chi^{-1} = \mathrm{co}(\chi)/\det(\chi)$, this polynomial is represented as [48]

$$\begin{aligned} &\tau\ (\det \boldsymbol{T})^2 \mathrm{Tr}\ (\boldsymbol{T}^{-1}(\boldsymbol{P}_1^{-1} - \boldsymbol{P}_2^{-1})) - p(\det \boldsymbol{T})^2 \\ &\times (2\boldsymbol{c}_0^T \boldsymbol{P}_1^{-1}\boldsymbol{c}_1 - 2\boldsymbol{c}_0^T \boldsymbol{P}_2^{-1}\boldsymbol{c}_2 + \boldsymbol{c}_0^T(\boldsymbol{P}_2^{-1} - \boldsymbol{P}_1^{-1})\boldsymbol{c}_0 - \boldsymbol{c}_1^T \boldsymbol{P}_1^{-1}\boldsymbol{c}_1 + \boldsymbol{c}_2^T \boldsymbol{P}_2^{-1}\boldsymbol{c}_2) = 0. \end{aligned}$$

It is pointed out that it is also possible to determine an inner ellipsoidal approximation. The method of finding the internal ellipsoidal approximation of the intersection of two ellipsoids is defined in [87].

#### 2.6.4.3.4 Norm Bounded Error with Linear Structure

As a generalization of the norm bound description $Z = \left\{ \bar{\boldsymbol{A}} + \boldsymbol{U} \mid \|\boldsymbol{U}\| \le \alpha \right\}$, $Z$ may be described as the image of a norm ball under an affine transformation [18]:

$$Z = \left\{ \bar{\boldsymbol{A}} + u_1\boldsymbol{A}_1 + u_2\boldsymbol{A}_2 + \ldots + u_p\boldsymbol{A}_p \right\}.$$

Here, $\|\cdot\|$ is a norm on $\mathbb{R}^p$, and the $p$ + 1 matrices $\bar{\boldsymbol{A}}, \boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_p \in \mathbb{R}^{mxn}$ are given. The worst-case error can be represented as

$$\begin{aligned} e_{wc} &= \sup_{\|\boldsymbol{u}\| \le 1} \left\| (\bar{\boldsymbol{A}} + u_1\boldsymbol{A}_1 + u_2\boldsymbol{A}_2 + \cdots + u_p\boldsymbol{A}_p)\boldsymbol{x} - \boldsymbol{b} \right\| \\ &= \sup_{\|\boldsymbol{u}\| \le 1} \left\| \boldsymbol{P}(\boldsymbol{x})\boldsymbol{u} + q(\boldsymbol{x}) \right\|, \end{aligned}$$

where $\boldsymbol{P}$ and $q$ are defined as $\boldsymbol{P} := \left[ \boldsymbol{A}_1\boldsymbol{x} \mid \boldsymbol{A}_2\boldsymbol{x} \mid \ldots \mid \boldsymbol{A}_p\boldsymbol{x} \right] \in \mathbb{R}^{m \times p}$ and $q(\boldsymbol{x}) := \bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b} \in \mathbb{R}^m$.

As an example, the *robust Chebyshev approximation problem* is regarded as being of the following form:

$$\text{minimize } e_{wc} = \sup_{\|\boldsymbol{u}\|_\infty \le 1} \left\| (\boldsymbol{A} + u_1\boldsymbol{A}_1 + u_2\boldsymbol{A}_2 + \ldots + u_p\boldsymbol{A}_p)\boldsymbol{x} - \boldsymbol{b} \right\|_\infty . \qquad (2.6.27)$$

In this case, for the worst-case error, an explicit expression may be derived. Then, $p_i(\boldsymbol{x})^T$ is denoted the $i$th row of $\boldsymbol{P}(\boldsymbol{x})$. So this problem can be represented as

$$\begin{aligned} e_{wc} = \sup_{\|\boldsymbol{u}\|_\infty \le 1} \left\| P(x)u + q(x) \right\|_\infty &= \max_{i=1,2,\ldots,m} \sup_{\|\boldsymbol{u}\|_\infty \le 1} \left| p_i(\boldsymbol{x})\boldsymbol{u} + q_i(\boldsymbol{x}) \right| \\ &= \max_{i=1,2,\ldots,m} \left\| p_i(\boldsymbol{x})\boldsymbol{u} \right\|_1 + \left| q_i(\boldsymbol{x}) \right|. \end{aligned}$$

As another example, the *robust LS problem* is taken into consideration as follows [18]:

$$\text{minimize } e_{wc}(\boldsymbol{x}) = \sup_{\|\boldsymbol{u}\|_2 \le 1} \left\| (\boldsymbol{A} + u_1\boldsymbol{A}_1 + u_2\boldsymbol{A}_2 + \ldots + u_p\boldsymbol{A}_p)x - \boldsymbol{b} \right\|_2 . \qquad (2.6.28)$$

Here, to compute $e_{wc}$, Lagrange duality is employed. With $\boldsymbol{u}$ as variable, the worst-case error $e_{wc}(\boldsymbol{x})$ is the square root of the optimal value of the nonconvex quadratic optimization problem

$$\begin{array}{ll} \text{maximize} & \|P(\boldsymbol{x})\boldsymbol{u} - q(\boldsymbol{x})\|_2^2 \\ \text{subject to} & \boldsymbol{u}^T\boldsymbol{u} \le 1. \end{array} \tag{2.6.29}$$

The Lagrange dual of this problem can be described as SDP [18]

$$\begin{array}{ll} \underset{\boldsymbol{x},t,\lambda}{\text{minimize}} & t+\lambda \\ \text{subject to} & \begin{pmatrix} I & P(\boldsymbol{x}) & q(\boldsymbol{x}) \\ P(\boldsymbol{x})^T & \lambda I & 0 \\ q(\boldsymbol{x}) & 0 & t \end{pmatrix} \succeq 0, \end{array} \tag{2.6.30}$$

with variables *t*, $\lambda \in \mathbb{R}$. Furthermore, strong duality holds for this pair of primal and dual problems. This means that, for any fixed ***x***, $e_{wc}(\boldsymbol{x})^2$ can be evaluated by solving the SDP with variables *t* and $\lambda$. Optimizing jointly over ***x***, *t* and $\lambda$ is equivalent to minimizing $e_{wc}(\boldsymbol{x})^2$. It is concluded that the robust least-squares problem can be solved as SDP with ***x***, $\lambda$, *t* as variables.

## 2.7 Optimization Software

It is important to make different between optimization solvers (optimizer) and optimization modeling languages [31]. An optimization solver is software which carries out numerical routines to find the optimal solution of an optimization problem. Optimization modeling languages are appeared as user-friendly platforms that let the user to specify optimization problems. AMPL and GAMS are two popular modeling languages. They communicate with a diversified amount of solvers. Also, there exist a number of languages which give modeling interfaces for particular types of optimization problems or solvers [31]. For example, YALMIP let Matlab users to preprocess SDPs and SOCPs. Then, these are passed to semidefinite solvers like SDPT3 and SeDuMi.

SDPT3 [85] and SeDuMi [79] can overcome linear constraints, quasi-convex-quadratic constraints and positive semidefinite constraints. Two of them use a primal-dual interior points method indicated as the *centering-predictors-correctors method,* and may exploit sparse matrix structure, making them very efficient [78]. For these semidefinite programming solvers, creating the inputs may be very time consuming, and can require substantial background in optimization modeling. YALMIP [52] and PROF[1] which are obtained as layers on top of these solvers in Matlab let for intuitive formulation of SDPs and SOCPs, and help the user retrieve the results from the solvers very easily [31].

MOSEK is also a useful optimizer for linear, quadratic, and convex quadratically constrained optimization problems well-known for speed and numerical stability [31]. It provides solvers for the optimization problems which have the types of the linear, conic quadratic (CQ), convex quadratic, general convex and mixed integer. MOSEK optimization tools also consist of interfaces which make it easy to apply the functionality of MOSEK from programming languages such as C, C++, MATLAB Toolbox, Java, NET, and Python [58].

MOSEK has some technical benefits. It is an optimization tool to solve large-scale mathematical optimization problems and, the problem size is only limited by the available memory. MOSEK has an interior-point optimizer with basis identification and it is well known owing to its excellent speed and stability [58]. The software uses problem sparsity and structure automatically to achieve the best possible efficiency.

MOSEK also has both primal and dual simplex optimizers for LP and corrects sensitivity analysis for linear problems. It has an efficient presolver to decrease problem size before optimization. It can cope with primal and dual infeasible problems in a systematic way [58]. Furthermore, MOSEK consists of tools for infeasibility diagnosis and repair and, it may read and write industry standard formats such as MPS, LP and XML.

[1] Currently being developed by Melvyn Sim at the National University of Singapore, http://www.nus.edu.sg/.

# CHAPTER 3

# METHODS

## 3.1 Multivariate Adaptive Regression Splines Method

*Multivariate Adaptive Regression Splines (MARS),* which is introduced by Friedman in 1991 [35], is a powerful adaptive regression method to estimate general functions of high dimensional arguments. Since the choice of basis functions is specific to the problem, MARS is an adaptive approach.

MARS builds a flexible model to high-dimensional nonparametric regression and shows a great promise for fitting nonlinear multivariate functions. This nonparametric regression approach makes no specific assumption about the underlying functional relationship between the dependent and independent variables. MARS searches variables one by one using a fast but intensive search procedure. Additionally, it also searches for interactions between independent variables, allowing any degree of interaction to be regarded as long as the model can better fit the data.

### 3.1.1 The Procedure

MARS generates an additive model in two-stage process. These are forward stage and backward stage [35]. In the **forward stage**, MARS finds which *basis functions* (*BFs*) are added to the model by using a fast searching algorithm and construct a possibly large model that overfit the data set. The process stops when the model reaches the maximum basis function $M_{\max}$ which is a specific value for users. However, this model has BFs that contribute most and least to the overall performance together. Thus, the model is more complex and includes many incorrect terms in the forward stage. In the **backward stage**, the overfit model is pruned to reduce the complexity of the model while supporting the overall performance with considering the fit to the data. In this stage, the BFs that contribute to smallest increase in the *residual sum of squares* (*RSS*) are removed from the model at each stage and at the end an optimally estimated model is produced [41].

MARS uses expansions of piecewise linear basis functions created by data set. The form of BFs is $[x-t]_+$ and $[t-x]_+$, and they are:

$$[x-t]_+ = \begin{cases} x-t, & \text{if } x>t \\ 0, & \text{otherwise} \end{cases} \quad , \quad [t-x]_+ = \begin{cases} t-x, & \text{if } x<t \\ 0, & \text{otherwise} \end{cases},$$

where $t$ is a univariate knot obtained from the data set. These two functions are called *truncated functions*. The following Figure 3.1 shows BF pairs for $t$ = 0.5 as an example:

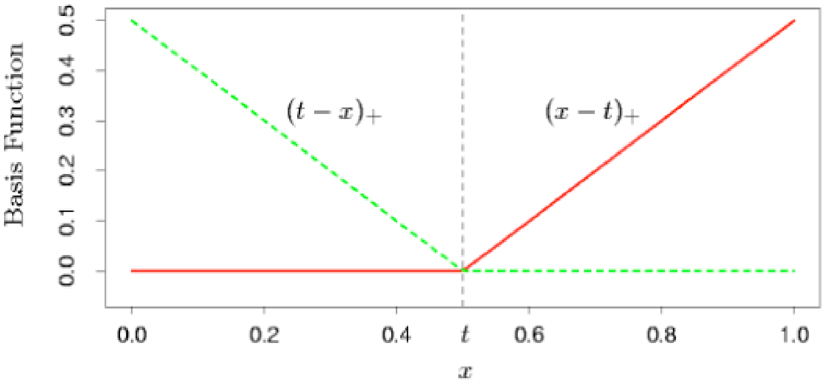


Figure 3.1 The BFs used by MARS for $t$ = 0.5 [41].

Each function is piecewise linear, with a knot at the value $t$, and it is called a *reflected pair*. The aim is to form reflected pairs for each input $X_j$ ( $j$ = 1,2,3,…,$p$) with $p$-dimensional knots at each observed value $x_{ij}$ of that input ($i$ = 1,2,3,…,$N$). Thus, the collection of BFs is written by a set of $\boldsymbol{B}$ defined as

$$B := \{[X_j - t]_+ \; , [t - X_j]_+ \mid t \;\in\; \{x_{1j}, x_{2j}, x_{3j}, ..., x_{Nj}\}, \, j = 1, 2, 3, ..., p\} \; ,$$

where $N$ is the number of observations, $p$ is the dimension of the input space. There are *2Np* BFs if all of the input values are distinct.

In the forward stage of MARS, the model that fits the data is built by using BFs from the set $B$ and their product. So, the model has the form

$$Y = E(\alpha_0 + \sum_{m=1}^{M} \alpha_m T_m(\boldsymbol{X}) \mid X = \boldsymbol{X}) + \varepsilon, \tag{3.1.1}$$

where $\boldsymbol{X} = (X_1, X_2, ..., X_p)^T$. Here, $\varepsilon$ is uncorrelated random error term which is assumed to have a normal distribution with zero mean and unknown constant variance, *M* is the set of BFs in the current model, $\psi_m(\boldsymbol{X})$ are BFs from the set *B* or products of two or more such functions, and $\alpha$ are the unknown coefficients for the constant 1 (*m* = 0) or for the *m*th BF. The form of the *m*th BF is as follows:

$$T_m(\boldsymbol{X}) = \prod_{k=1}^{K_m} [s_{km}.(x_{v(k,m)} - t_{km})]_+ , \tag{3.1.2}$$

where $K_m$ is the number of truncated linear functions multiplied in the *m*th BF, $x_{v(k,m)}$ is the input variable corresponding to the *k*th truncated linear function in the *m*th BF, $t_{km}$ is the knot value corresponding to the variable $x_{v(k,m)}$ and $s_{km} = \pm 1$. A *lack-of-fit criterion* is used to compare the possible BFs.

To generate the model, the MARS forward stepwise algorithm starts with the constant function $T_0(\boldsymbol{X}) = 1$ to estimate $\alpha_0$, and all functions in the set *B* are candidate functions. Possible forms of the BFs $T_m(\boldsymbol{X})$ are [47]:

- 1,
- $x_k$,
- $[x_k - t_i]_+$ ,
- $x_k x_l$,
- $[x_k - t_i]_+ x_l$ and
- $[x_k - t_i]_+ [x_l - t_j]_+$.

For each BF, input variables cannot be same in the MARS algorithm. Therefore, the BFs above use different input variables, $x_k$ and $x_l$, and their knots, $t_i$ and $t_j$. At each stage, with one of the reflected pair in the BFs set *B*, all products of a function $T_m(\boldsymbol{X})$ in the model set are regarded as a new function pair and added the term to the model set. That term which produces the largest decrease in training error has the following form:

$$\alpha_{M+1} T_k(X) \cdot [X_j - t]_+ + \alpha_{M+2} T_k(X) \cdot [t - X_j]_+ . \tag{3.1.3}$$

Here, $\alpha_{M+1}$ and $\alpha_{M+2}$ are coefficients and they are estimated by LS, along with all other $M+1$ coefficient in the model. Then the "winning" products are added to the model and the process stops when the model set reaches some present maximum number of terms. For example, the following BFs are possible candidates [47]:

- 1,
- $x_k$,
- $[x_k - t_i]_+$, if $x_k$ is already in the model,
- $x_k x_l$, if $x_k$ and $x_l$ are already in the model,
- $[x_k - t_i]_+ x_l$, if $x_k x_l$ and $[x_k - t_i]_+$ are already basis functions,
- $[x_k - t_i]_+ [x_l - t_j]_+$, if $[x_k - t_i]_+ x_l$ and $[x_l - t_j]_+ x_k$ are already in the model.

At the end of this forward stepwise process, a large model of the form is obtained. This model typically *overfits* the data, and so a backward deletion procedure is applied.

The backward stepwise algorithm removes the terms that contribute the smallest increase in the residual squared error from the model at each stage, and this iterative procedure continues until an optimal number of effective terms are present in the final model [41]. So, an estimated best model $\hat{f}_\mu$ of each number of terms $\mu$ is produced at the end of this process. In the MARS model, *generalized cross-validation* (*GCV*) is used to find the optimal number of terms $\mu$. Also, it shows the **lack of fit** when using MARS. The *GCV criterion* defined by Friedman [35] is as follows:

$$LOF(\hat{f}_\mu) \ = \ GCV(\mu) := \frac{\sum_{i=1}^{N} (y_i - \hat{f}_\mu(\boldsymbol{x}_i))^2}{(1 - M(\mu)/N)^2}, \qquad (3.1.4)$$

where $M(\mu)$ is the effective number of parameter in the model, and $N$ is the number of sample observations.

The effective number of parameters $M(\mu)$ has different representation. It is represented by $M(\mu) = r + cK$, where $r$ is the number of linearly independent BFs in the model and $K$ is the number of knots selected in the forward process, $c$ is a cost for basis-function optimization and generally chosen as $c = 3$ ($c = 2$ is used when the model is additive). A larger $M(\mu)$ creates a smaller model with less basis functions, whereas a smaller $M(\mu)$ creates a larger model with more basis functions [41].

### 3.1.2 Pros and Cons of MARS

MARS offers a number of advantages. MARS is capable of identifying a relatively small number of predictor variables which are complex transformations of initial variables [25]. It also enables to discover nonlinearities that can exist in the relationship between response and predictor variables.

Conventional statistical methods such as regression can handle interactions terms, but they requires trying many combinations of the variables in the data set. Therefore, they can be computationally infeasible. MARS automatically looks for suitable interactions between independent variables, which makes it in particular preferable whenever there is a large number of interacting variables. MARS identifies interactions, and also produces graphs that help visualize and understand interactions [41].

However, MARS has some limitations which the analyst must keep in mind. Firstly, MARS is a sophisticated technique, requiring a relatively steep learning curve to use the methodology and interpret the results. Secondly, it is not yet widely available in the main statistical packages and has to be purchased separately. Thirdly, as in other modeling techniques, overfitting in MARS must be taken into consideration [25]. In MARS methodology, a very exhaustive search is led to identify nonlinearities and interactions. Therefore, it has a risk of overfitting the data, but it is possible to protect against overfitting such as setting a lower maximum number of BFs and a higher “cost” per knot [35].

### 3.1.3 MARS vs. Other Algorithms

MARS algorithm is based on a modified recursive partitioning methodology [35]. It is an extension of *Classification and Regression Tree* (*CART*) [16]. It is mainly similar to decision tree (DT) techniques such as CART on the partitioning of intervals, where two symmetric BFs are constructed at the knot location. However, MARS is normally not presented in decision tree techniques. While CART uses indicator functions causing the lack of continuity that affects the model accuracy, MARS uses piecewise linear functions that are continuous, and it produces a continuous model that provides a more effective way to model nonlinearities [23].

MARS is a flexible regression technique which applies a modified recursive partitioning strategy for simplifying high-dimensional nonparametric problems. *Recursive Partitioning Regression* (*RPR*) is a powerful method nonparametric modeling approach but, it has some shortcomings such as discontinuity at the subregion boundaries. MARS method handles these restrictions to increase accuracy [94].

The *Artificial Neural Network* (*ANN*) can be characterized as nonlinear, nonparametric function estimation techniques, and shows a great promise for fitting general nonlinear multivariate functions like MARS. However, it has restrictions like a long training process, interpretation difficulties of the model and application in some problems. MARS has also the capability to overcome these problems and it is in most cases both more accurate and much faster than neural networks [23]. Furthermore, if we compare MARS with other modeling techniques such as *multivariate linear regression* (*MLR*), *regression tree* (*RT*), *support vector machine* (*SVM*), MARS provide us better prediction accuracy [95].

MARS is compared to a number of parametric and nonparametric approximations routine in terms of its accuracy, efficiency, robustness, model transparency, and simplicity due to its inception [22]. MARS is chosen from among other better-known methodologies because it is more interpretable than most recursive partitioning, neural and adaptive strategies. It distinguishes well between signal and noise variables. MARS' computation times are competitive for low-dimensional problems and fast for medium- to high-dimensional problems, and it has the best prediction success when it is compared to linear models, principal component regression, and CART based on efficiency, accuracy, and implementation. MARS performs well for predictive modeling of continuous outcomes [22].

On the other hand, MARS is highly sensitive to both sample size and design of the experiment type. This is one main reason for our study of robust CMARS. Also, MARS generally under performs other approximation techniques when it is used with small data sets, although it performs best with medium or large data sets when it is implemented with controlled experimentation [22].

### 3.1.4 MARS Software

For this study, the MARS models are fitted by using MARS (Version 3, Salford Systems, San Diego, Calif., USA). MARS allows the user to set control parameters to explore different models and create the **best** model. The maximum number of knots is defined by trial and error; the maximum number of interactions may be more than the degree of two (2-way interaction). The MARS package which is developed by Salford Systems is available at [20]. It is well-designed software which performs MARS technique with a friendly graphical user interface.

The penalty on added variables causes MARS to prefer reusing of variables already in the model over adding new variables. While the penalty is increased, MARS makes a tender to obtain new knots in existing variables or generates interaction terms in including existing variables [55]. The minimum number of observations between knots is very useful for continuous variables but not discrete ones. By default, MARS allows obtaining a knot at every observed data value and this default allows the MARS regression to change slope or direction anywhere and as often as the data dictate [55].

MARS Software has a different type of speed parameter to search model. This search speed parameter can be set one to five and its default value is four [55]. For real-world problems it is advised to use four as a search speed parameter, but the use of search speed parameter of three or five does not change the models. After setting all the parameters correctly, MARS creates the final model in a rather short time.

There exist many result evaluations provided for MARS. In fact, there are $R^2$, *Mean Square Error* (*MSE*), *ANOVA*, *f*-value, *t*-value, *p*-value, *RSS*, variable importance measurement assessed by observing the decrease in performance when one is removed from the model, etc.. Furthermore, various result clarifications are also available: *the final model includes a number of specific BFs, gain and lift charts, curve and surface plots*, etc. [55]. Additionally, a previously created model can be applied to a new data set. Consequently, the MARS package is regarded as very powerful as it takes in various preferences, criteria, control parameters and constraints for the user.

## 3.2 Conic Multivariate Adaptive Regression Spline (CMARS)

### 3.2.1 MARS Method Revisited by Tikhonov Regularization

MARS (cf. Section 3.1) is a method to estimate general functions of high dimensional arguments that are given sparse data [35]. It has an increasing number of applications in many areas of science, economy and technology. At the same time it is a research challenge, especially, by means of using continuous optimization theory. We mostly refer to a regression formulation, but also classification will be addressed. The finitely many data underlying can base on different types of experiments, questionnaires, records or a preprocessing of information by clustering, etc.

MARS is an adaptive procedure since the selection of BFs is data-based and specific to the problem at hand. This algorithm is a nonparametric regression procedure that makes no specific assumption about the underlying functional relationship between the dependent and independent variables. A special advantage of MARS lies in its ability to estimate the contributions of the BFs so that both the additive and the interactive effects of the predictors are allowed to determine the response variable.

For this model, an algorithm was proposed by Friedman in 1991 [35] as a flexible approach to high dimensional nonparametric regression, based on a modified recursive partitioning methodology. The above explanations have been given in detail in the previous section. In this section, we explain a modified version of MARS called *Conic Multivariate Adaptive Regression Splines* (CMARS). Here, "C" also means convex and continuous. For our explanations on CMARS, it is preferred the following notation for the piecewise linear BFs:

$$c^{+}(\tilde{x},\tau) = \left[+(\tilde{x}-\tau)\right]_{+},\ c^{-}(\tilde{x},\tau) = \left[-(\tilde{x}-\tau)\right]_{+}, \tag{3.2.1}$$

where $\left[q\right]_{+} := \max\ \{0,q\}$ and $\tau$ is an univariate knot ($x,\tau \in \mathbb{R}$). Each function is piecewise linear, with a knot at the value $\tau$ , and it is called a *reflected pair*. For a visualization see Figure 3.2:

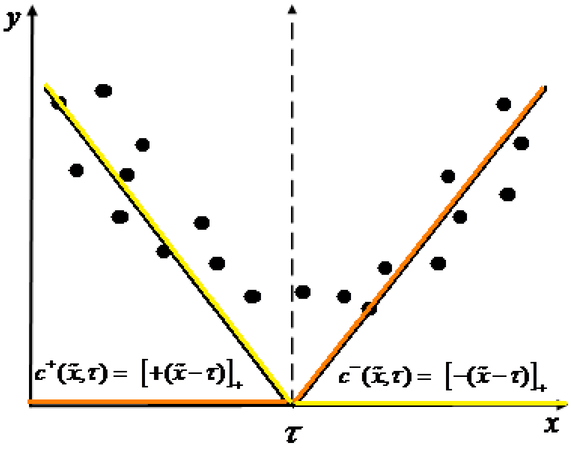


Figure 3.2 Basic elements in the regression with MARS [88].

In this figure, the points demonstrate the data $(\tilde{\boldsymbol{x}}_i,\ \tilde{y}_i)\ (i = 1,2,...,N)$ comprised of a *p*-dimensional input specification of the variable *x* and the corresponding one-dimensional response which specify the variable *y*. To be more precise, the variable *x* in Figure 3.2 is one of the coordinates of **x**. Let us take into account the following general model on the relation between input and response:

$$Y = f(\tilde{\boldsymbol{X}}) + \varepsilon, \tag{3.2.2}$$

where *Y* is a response variable, $\tilde{\boldsymbol{X}} = (\tilde{X}_1,\tilde{X}_2,...,\tilde{X}_p)^T$ is a vector of predictor random variables and $\varepsilon$ is an additive stochastic component which is supposed to have zero mean and finite variance. The aim is to build reflected pairs for each input $\tilde{X}_j\ (j = 1,2,...,p)$ with *p*-dimensional knots $\boldsymbol{\tau}_i = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,p})^T$ at or just nearby each input data vectors $\tilde{\boldsymbol{x}}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, ..., \tilde{x}_{i,p})^T$ of that input ($i$ = 1, 2,...,$N$). Such a nearby placement means a slight modification made in this study [88]. Here, we can without loss of generality suppose that $\tau_{i,j} \neq \tilde{x}_{i,j}$ for all $i$ and $j$, to prevent from nondifferentiability in our optimization problem later on. This assumption is also stated into Figure 3.2. We could even select the knots $\tau_{i,j}$ more far away from the input values $\tilde{x}_{i,j}$ if any such a position promises a better data fitting. After these preparations for CMARS, the set of BFs is:

$$S := \left\{[\tilde{X}_j - \tau]_+,\ [\tau - \tilde{X}_j]_+ \ \Big|\ \tau \in \left\{x_{1,j}, x_{2,j}, ..., x_{N,j}\right\},\ j \in \{1, 2, ..., p\}\right\}.$$

When all of the input values are distinct, there exist $2Np$ BFs altogether. Therefore, we may represent $f(\tilde{\boldsymbol{X}})$ by a linear combination which is successively constructed by the set $S$ and with the intercept $\alpha_0$ such that (3.2.2) takes the following form:

$$Y = \alpha_0 + \sum_{m=1}^{M} \alpha_m \psi_m(\tilde{\boldsymbol{X}}) + \varepsilon. \tag{3.2.3}$$

Here, $\psi_m$ ($m$ = 1, 2,...,$M$) presents a BF from $S$ or products of two or more such functions, $\psi_m$ is taken from a set of $M$ linearly independent basis elements, and $\alpha_m$ is the unknown coefficient for the $m$th BF ($m$ = 1, 2,...,$M$) but, $m$ equals to zero for the constant 1. A set of eligible knots $\tau_{i,j}$ is given separately for each input variable dimension and is selected to approximately coincide with the input levels represented in the data. Interaction BFs are produced by multiplying an existing BF with a truncated linear function including a new variable. In this case, both the existing BF and the newly created interaction BF are employed in the MARS approximation [80]. Provided the observations represented by the data $(\tilde{\boldsymbol{x}}_i,\ \tilde{y}_i)$ $(i = 1, 2, ..., N)$ the form of the $m$th BF is as follows:

$$\psi_m(\tilde{\boldsymbol{x}}) := \prod_{j=1}^{K_m} [s_{\kappa_j^m} . (x_{\kappa_j^m} - \tau_{\kappa_j^m})]_+, \tag{3.2.4}$$

where $K_m$ is the number of truncated linear functions multiplied in the $m$th basis function, $x_{\kappa_j^m}$ is the input variable corresponding to the $k$th truncated linear function in the $m$th basis function, $\tau_{\kappa_j^m}$ is the knot value corresponding to the variable $x_{\kappa_j^m}$ and $s_{\kappa_j^m}$ is the selected sign +1 or -1. The search of new BFs may be limited to interactions of a maximum order. For example, if only up to three-factor interactions are allowed, and then $K_m \leq 3$ could be restricted in.

### 3.2.2 The Penalized Residual Sum of Squares Problem

The *Penalized Residual Sum of Square* (*PRSS*) with $M_{\max}$ BFs is accumulated in the *forward* stepwise algorithm of MARS. To estimate the function $f(\boldsymbol{x})$, it is not employed the backward stepwise algorithm of MARS. At its place, as an alternative [80], penalty terms are used in addition to the *least-squares estimation* (LSE) to control the lack of fit from the viewpoint of the *complexity* and *stability*. It is not needed to run the backward stepwise algorithm of MARS. For the MARS model, PRSS has the following form:

$$PRSS := \sum_{i=1}^{N}(y_i - f(\tilde{\boldsymbol{x}}_i))^2 + \sum_{m=1}^{M_{\max}} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \alpha_m^2 [D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m)]^2 d\boldsymbol{t}^m , \qquad (3.2.5)$$

where $V(m) := \{ \kappa_j^m \mid j = 1,2,..., K_m \}$ is the variable set associated with the *m*th basis function $\psi_m$, $\boldsymbol{t}^m = (t_{m_1}, t_{m_2}, ..., t_{m_{K_m}})^T$ presents the vector of variables that contribute to the *m*th basis function $\psi_m$. The parameters $\phi_m \geq 0$ are in the role of *penalty parameters* (*m* = 1,2,..., $M_{\max}$). The integral of the first-order derivatives measure the *flatness* of the model functions whereas the integrals of the second-order derivatives measure the *instability* and *complexity* inscribed into the model (via the model functions) [41,82]. Moreover,

$$D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m) := (\partial\psi_m / \partial^{\theta_1} t_r{}^m \, \partial^{\theta_2} t_s{}^m)(\boldsymbol{t}^m)$$

for $\boldsymbol{\theta}^T = (\theta_1, \theta_2)$, $|\boldsymbol{\theta}| := \theta_1 + \theta_2$; $\theta_1, \theta_2 \in \{0,1\}$. We note that in any case where $\theta_i$ =2, the derivative $D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m)$ vanishes, and by addressing indices *r* < *s*, it has been applied Schwarz's Theorem. In order not to overload the exposition, a slight flaw in the notation is still accepted since in case of $|\boldsymbol{\theta}|$=1 and $K_m$ >1 the integral terms become mentioned $K_m$ times by the pair *r* < *s*. By redefining, $\phi_m$ by $\phi_m / K_m$, this little deficiency could be easily corrected. It is used the integral symbol "$\int$" as a dummy in the sense of $\int_{Q^m}$, where $Q^m$ is some appropriately large dimensional parallelpipe ( $K_m$ -dimensional) where the integration takes place. The integrals and entire optimization problems are well defined because all the regarded derivatives of any function $\psi_m$ exist except on a set of measure zero [81].

The optimization problem which we use bases on the ***tradeoff*** between both *accuracy*, i.e., a small sum of error squares, and *not too high a complexity*. This tradeoff is established through the penalty parameters $\phi_m$. In this study, tradeoff is tackled by means of penalty methods, such as regularization techniques [4].

If we consider the representations (3.2.2) and (3.2.3) in (3.2.4), then the objective function (3.2.4) will be of the following form [80]:

$$\begin{aligned} PRSS = &\sum_{i=1}^{N} \left( \tilde{y}_i - \alpha_0 - \sum_{m=1}^{M} \alpha_m \psi_m(\tilde{\boldsymbol{x}}_i^m) - \sum_{m=M+1}^{M_{\max}} \alpha_m \psi_m(\tilde{\boldsymbol{x}}_i^m) \right)^2 \\ &+ \sum_{m=1}^{M_{\max}} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \alpha_m^2 [D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m)]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.2.6}$$

where $\tilde{\boldsymbol{x}}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, ..., \tilde{x}_{i,p})^T$ denotes any of the input vectors and $\tilde{\boldsymbol{x}}_i^m = (\tilde{x}_{i,v(1)}^m, \tilde{x}_{i,v(2)}^m, ...,$ $\tilde{x}_{i,v(K_m)}^m)^T$ stands for the corresponding projection vectors of $\tilde{x}_i$ onto those coordinates that contribute to the *m*th basis function $\psi_m$ (related with the *i*th output $\tilde{y}_i$).

It is noted that the second-order derivatives of the piecewise linear functions $\psi_m$ ($m$=1,2,...,$M$) and, thus, the penalty terms related are vanishing. The representation of PRSS may be rearranged as follows:

$$PRSS = \sum_{i=1}^{N} \left( y_i - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_i) \right)^2 + \sum_{m=1}^{M_{\max}} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int_{Q^m} \alpha_m^2 \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m, \tag{3.2.7}$$

where $\boldsymbol{\psi}(\tilde{\boldsymbol{b}}_i) := \left(1, \psi_1(\tilde{\boldsymbol{x}}_i^1), ..., \psi_M(\tilde{\boldsymbol{x}}_i^M), \psi_{M+1}(\tilde{\boldsymbol{x}}_i^{M+1}), ..., \psi_m(\tilde{\boldsymbol{x}}_i^{M_{\max}})\right)^T$ and $\boldsymbol{\alpha} := (\alpha_0, \alpha_1, ..., \alpha_{M_{\max}})^T$ with the point $\tilde{\boldsymbol{b}}_i := \left( \tilde{\boldsymbol{x}}_i^1, \tilde{\boldsymbol{x}}_i^2, ..., \tilde{\boldsymbol{x}}_i^M, \tilde{\boldsymbol{x}}_i^{M+1}, \tilde{\boldsymbol{x}}_i^{M+2}, ..., \tilde{\boldsymbol{x}}_i^{M_{\max}} \right)^T$. A discretized form is used to approximate the multi-dimensional integrals $\int_{Q^m} \alpha_m^2 \left[ D_{r,s}^{\boldsymbol{\theta}} \boldsymbol{\psi}(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m$ [80]. For this aim, data points $(\tilde{\boldsymbol{x}}_k, \tilde{y}_k)$ ($k$= 1,2,...,$N$) with $\tilde{\boldsymbol{x}}_k \in \mathbb{R}^n$ are given. In a natural way, these input data

$\tilde{\boldsymbol{x}}_k = (\tilde{x}_{k,1}, \tilde{x}_{k,2}, ..., \tilde{x}_{k,p})^T$ create a subdivision of any sufficiently large parallelpipe $Q$ of $\mathbb{R}^n$ that includes each of them as elements. Here $Q$ is represented as a parallelpipe that encompasses all our input data in the following form [80, 93]:

$$Q = \left[a_1, b_1\right] \times \left[a_2, b_2\right] \times ... \times \left[a_p, b_p\right] = \prod_{j=1}^{p} Q_j \ ,$$

where $Q_j = [a_j, b_j]$, $a_j \leq \tilde{x}_{k,j} \leq b_j$ ($j$=1,2,...,$p$; $k$=1,2,...,$N$).

We can assume $a_j < \tilde{x}_{k,j} < b_j$ without loss of generality. For all $k$ , we reorder the coordinates of the input data points: $\tilde{x}_{k_1^j, j} \leq \tilde{x}_{k_2^j, j} \leq ... \leq \tilde{x}_{k_N^j, j}$ , where $k_\mu^j$=1,2,...,$N$ ($\mu$ =1,2,...,$N$; $j$=1,2,...,$p$), and $\tilde{x}_{k_\mu^j, j}$ is $j$th component of $\tilde{x}_{k_\mu^j}$ , the $k_\mu^j$ input vector after reordering. We can suppose $\tilde{x}_{k_\mu^j, j} \neq \tilde{x}_{k_\lambda^j, j}$ for all $\mu, \lambda$ =1,2,...,$N$ with $\mu \neq \lambda$ ; i.e., $\tilde{x}_{k_1^j, j} < \tilde{x}_{k_2^j, j} < ... < \tilde{x}_{k_N^j, j}$ ( $j$=1,2,...,$p$). If "=" is attained for some coordinate, we obtain subparallelpipes of a lower dimension in the following integration process. When we can denote $\tilde{x}_{k_0^j, j} := a_j$, $k_0^j := b_j$ and $k_{N+1}^j := N+1$; then,

$$Q = \bigcup_{\mu^j=0}^{N} \prod_{j=1}^{p} \left[ \tilde{x}_{k_{\mu^j}^j, j}, \tilde{x}_{k_{\mu^j+1}^j, j} \right],$$

and, when we use this idea to our case, we may write the discretized form as follows:

$$\int_{Q^m} \alpha_m^2 \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m \ \approx \sum_{(\mu^{\nu(n)})_{n\in\{0,1,2,...,K_m\}} \in\{0,1,2,...,N+1\}^{K_m}} \alpha_m^2 .$$
$$\left[ D_{r,s}^{\boldsymbol{\theta}} \psi_m (\tilde{x}_{k_{\mu^{\nu(1,m)}}^{\nu(1,m)}, \nu(1,m)}, \tilde{x}_{k_{\mu^{\nu(2,m)}}^{\nu(2,m)}, \nu(2,m)}, ..., \tilde{x}_{k_{\mu^{\nu(K_m,m)}}^{\nu(K_m,m)}, \nu(K_m,m)}) \right]^2 \cdot \prod_{n=1}^{K_m} \left( \tilde{x}_{k_{\mu^{\nu(n,m)}}^{\nu(n,m)}+1, \nu(n,m)} - \tilde{x}_{k_{\mu^{\nu(n,m)}}^{\nu(n,m)}, \nu(n,m)} \right).$$

When this discretization is applied, *PRSS* can be approximated in the following way [81]:

$$PRSS = \sum_{i=1}^{N} \left( y_i - \boldsymbol{\alpha}^T \psi(\tilde{\boldsymbol{b}}_i) \right)^2 + \sum_{m=1}^{M_{\max}} \phi_m \alpha_m^2 \sum_{i=1}^{(N+1)^{K_m}} \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_m(\hat{x}_i^m) \right]^2 \right) \Delta\hat{x}_i^m \ , \qquad (3.2.8)$$

where $\hat{\boldsymbol{x}}_i^m = \left( \tilde{x}_{k_{\mu^{\nu(1,m)}}^{\nu(1,m)}, \nu(1,m)}, \tilde{x}_{k_{\mu^{\nu(2,m)}}^{\nu(2,m)}, \nu(2,m)}, ..., \tilde{x}_{k_{\mu^{\nu(K_m,m)}}^{\nu(K_m,m)}, \nu(K_m,m)} \right)$, and

$$\Delta\hat{\boldsymbol{x}}_i^m = \prod_{n=1}^{K_m} \left( \tilde{x}_{k_{\mu^{\nu(n,m)}}^{\nu(n,m)}+1, \nu(n,m)} - \tilde{x}_{k_{\mu^{\nu(n,m)}}^{\nu(n,m)}, \nu(n,m)} \right).$$

For a short representation, we may rewrite the approximate relation (3.2.6) as follows:

$$PRSS \approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 + \sum_{m=1}^{M_{\max}} \phi_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \alpha_m^2 \ , \tag{3.2.9}$$

where

$$L_{im} := \left[ \left( \sum_{\substack{|\theta|=1 \\ \theta=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \left[ D^{\theta}_{r,s} \psi_m(\hat{\boldsymbol{x}}^m_i) \right]^2 \right) \Delta \hat{\boldsymbol{x}}^m_i \right]^{1/2} .$$

Here, $\boldsymbol{\psi}(\tilde{\boldsymbol{b}}) = (\boldsymbol{\psi}(\tilde{\boldsymbol{b}}_1), \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_2), ..., \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_N))^T$ is an $(N \times (M_{\max}+1))$-matrix and $\sum_{i=1}^{N} \left( y_i - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_i) \right)^2 = \left( \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_i)\boldsymbol{\alpha} \right)^T \left( \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}}_i)\boldsymbol{\alpha} \right) = \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 .$

### 3.2.3 Tikhonov Regularization Applied

Now, we deal with the problem PRSS as a *TR problem*. For this aim, the formula (3.2.5) is regarded again and arranged as [80, 81]:

$$\begin{aligned} PRSS &\approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 + \sum_{m=1}^{M_{\max}} \phi_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \alpha_m^2 \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 + \phi_1 \left\| L_1 \alpha_1 \right\|_2^2 + \phi_2 \left\| L_2 \alpha_2 \right\|_2^2 + ... + \phi_{M_{\max}} \left\| L_{M_{\max}} \alpha_{M_{\max}} \right\|_2^2 , \end{aligned} \tag{3.2.10}$$

where $\boldsymbol{L}_m := \left( L_{1m}, L_{2m}, ..., L_{(N+1)^{K_m} m} \right)^T$ $(m = 1, 2, ..., M_{\max})$. There is a finite sequence of the *tradeoff* or *penalty* parameters $\boldsymbol{\phi} = (\phi_1, \phi_2, ..., \phi_{M_{\max}})^T$ such that this equation is not yet a TR problem with a single such parameter. Because of this, we make a uniform penalization by taking the same $\phi$ for each derivative term. Then, our approximation of PRSS may be rearranged as follows:

$$PRSS \approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 + \phi \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 , \tag{3.2.11}$$

where $\boldsymbol{L}$ is an $((M_{\max}+1) \times (M_{\max}+1)$-diagonal matrix with first column $\boldsymbol{L}_0 = \boldsymbol{0}_{N+1)^{K_m}}$ and the other columns being the vectors $\boldsymbol{L}_m$ introduced above. Moreover, $\boldsymbol{\alpha}$ is an $((M_{\max}+1) \times 1)$-parameter vector to be estimated through the data points. Then, our *PRRS* problem looks as a classical *TR problem* (2.2.4) with $\phi > 0$ , $\phi = \lambda^2$ for some $\phi \in \mathbb{R}$ .

### 3.2.4 An Alternative for Tikhonov Regularization Problem with Conic Quadratic Programming

#### 3.2.4.1 Construction of the Conic Quadratic Programming Problem

Let us evaluate the TR problem (3.2.11) with CQP that is a continuous optimization technique. PRSS may be easily formulated as a CQP problem. In fact, based on an approximate selection of a bound $\tilde{M}$ we state the following problem [81, 82]:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} - \boldsymbol{y} \right\|_2^2 \\ &\text{subject to} \quad \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 \le \tilde{M}. \end{aligned} \tag{3.2.12}$$

We underline that this choice of $\tilde{M}$ must be the outcome of a careful learning process, with the help of model-free or model-based methods [4]. In (3.2.12), we have the LS objective function $\left\| \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} - \boldsymbol{y} \right\|_2^2$ and the inequality constraint function $-\left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 + \tilde{M}$. Now, we equivalently write our optimization problem as follows [80, 82]:

$$\begin{aligned} &\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \quad t, \\ &\text{subject to} \quad \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} - \boldsymbol{y} \right\|_2^2 \le t^2, \quad t \ge 0, \\ &\qquad\qquad \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 \le \tilde{M}, \end{aligned} \tag{3.2.13}$$

or, equivalently again,

$$\begin{aligned} &\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \quad t, \\ &\text{subject to} \quad \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{b}})\boldsymbol{\alpha} - \boldsymbol{y} \right\|_2 \le t \ , \\ &\qquad\qquad \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2 \le \sqrt{\tilde{M}}. \end{aligned} \tag{3.2.14}$$

We apply modern methods of *continuous optimization techniques*, especially; from CQP where following the basic notation is used [82]

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad \mathrm{c}^T \boldsymbol{x}, \\ &\text{subject to} \quad \left\| \boldsymbol{D}_i \boldsymbol{x} - \boldsymbol{d}_i \right\| \le \boldsymbol{p}_i^T \boldsymbol{x} - q_i \qquad (i=1,2,...,k). \end{aligned}$$

Indeed, we can see that our optimization problem is such a CQP program with

$\boldsymbol{c} = (1, \mathbf{0}^T_{M_{\max}+1})^T$ , $x = (t, \boldsymbol{\alpha}^T)^T$, $\boldsymbol{D}_1 = (\mathbf{0}_N, \boldsymbol{\psi}(\tilde{\boldsymbol{b}}))$, $\boldsymbol{d}_1 = \boldsymbol{y}$, $\boldsymbol{p}_1 = (1, 0, ...., 0)^T$, $q_1 = 0$,

$\boldsymbol{D}_2 = (\mathbf{0}_{M_{\max}+1}, \boldsymbol{L})$, $\boldsymbol{d}_2 = \mathbf{0}_{M_{\max}+1}$, $\boldsymbol{p}_2 = \mathbf{0}_{M_{\max}+2}$ and $q_2 = -\sqrt{\tilde{M}}$.

To represent the optimality condition for this problem, we firstly reformulate the problem (3.2.14) as follows [80, 81]:

$$\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \quad t,$$

such that

$$\boldsymbol{\chi} := \begin{pmatrix} \mathbf{0}_N & \psi(\tilde{\boldsymbol{b}}) \\ 1 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} -\boldsymbol{y} \\ 0 \end{pmatrix},$$

$$\boldsymbol{\eta} := \begin{pmatrix} \mathbf{0}_{M_{\max}+1} & \boldsymbol{L} \\ 0 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{M_{\max}+1} \\ \sqrt{\tilde{M}} \end{pmatrix},$$

$$\boldsymbol{\chi} \in L^{N+1}, \ \boldsymbol{\eta} \in L^{M_{\max}+2}.$$

The dual problem to the latter primal one is given by

$$\text{maximize} \ \ (\boldsymbol{y}^T, 0)\boldsymbol{\omega}_1 + (\mathbf{0}^T_{M_{\max}+1}, \text{-}\sqrt{\tilde{M}})\boldsymbol{\omega}_2$$

$$\text{such that} \begin{pmatrix} \mathbf{0}^T_N & 1 \\ \boldsymbol{\psi}(\tilde{\boldsymbol{b}})^T & \mathbf{0}_{M_{\max}+1} \end{pmatrix} \omega_1 + \begin{pmatrix} \mathbf{0}^T_{M_{\max}+1} & 0 \\ \boldsymbol{L}^T & \mathbf{0}_{M_{\max}+1} \end{pmatrix} \omega_2 = \begin{pmatrix} 1 \\ \mathbf{0}_{M_{\max}+1} \end{pmatrix},$$

$$\boldsymbol{\omega}_1 \in L^{N+1}, \ \boldsymbol{\omega}_2 \in L^{N+2}.$$

Furthermore, $(t, \boldsymbol{\alpha}, \boldsymbol{\chi}, \boldsymbol{\eta}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is a *primal dual optimal solution* if and only if [80, 81]

$$\boldsymbol{\chi} := \begin{pmatrix} \mathbf{0}_N & \boldsymbol{\psi}(\tilde{\boldsymbol{b}}) \\ 1 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} -\boldsymbol{y} \\ 0 \end{pmatrix},$$

$$\boldsymbol{\eta} := \begin{pmatrix} \mathbf{0}_{M_{\max}+1} & \boldsymbol{L} \\ 0 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{M_{\max}+1} \\ \sqrt{\tilde{M}} \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{0}_N^T & 1 \\ \boldsymbol{\psi}(\bar{\boldsymbol{b}})^T & \mathbf{0}_{M_{\max}+1} \end{pmatrix} \boldsymbol{\omega}_1 + \begin{pmatrix} \mathbf{0}_{M_{\max}+1}^T & 0 \\ \boldsymbol{L}^T & \mathbf{0}_{M_{\max}+1} \end{pmatrix} \boldsymbol{\omega}_2 = \begin{pmatrix} 1 \\ \mathbf{0}_{M_{\max}+1} \end{pmatrix},$$

$$\boldsymbol{\omega}_1^T \boldsymbol{\chi} = 0, \quad \boldsymbol{\omega}_2^T \boldsymbol{\eta} = 0,$$
$$\boldsymbol{\omega}_1 \in L^{N+1}, \ \boldsymbol{\omega}_2 \in L^{M_{\max}+2},$$
$$\boldsymbol{\chi} \in L^{N+1}, \ \boldsymbol{\eta} \in L^{M_{\max}+2},$$

where $L^{N+1}$, $L^{M_{\max}+2}$ are the $(N+1)$- and $(M_{\max}+2)$-dimensional ice-cream (*or second-order, or Lorentz*) cones.

# CHAPTER 4

# THEORETICAL FOUNDATIONS and AN APPLICATION

## 4.1 Robust Conic Multivariate Adaptive Regression Splines Method

### 4.1.1 Introduction

CMARS is an alternative method to a well-known regression tool, called MARS, from data mining and estimation theory (cf. Chapter 3). This method is based on a PRSS for MARS as a TR problem. CMARS treated this problem by a continuous optimization technique, called CQP, which permits to use interior point methods [16]. CMARS is particularly powerful in handling complex and heterogeneous data (for more details see Chapter 3). Heterogeneous data includes both discrete and continuous variables. In this chapter, we include into CMARS the existence of uncertainty about the scenarios. Indeed, data include noise in both output and input variables. Therefore, solutions of the optimization problem may reveal a remarkable sensitivity to perturbations in the parameters of the problem. The data uncertainty results in uncertain constraints and objective function. To overcome this difficulty, we refine our CMARS algorithm by important *RO* technique proposed to cope with data uncertainty (we refer to Section 2.6 for more details).

RO is a modeling methodology to process optimization problems in which the data are uncertain and are only known to belong to some uncertainty set, except of outliers. The purpose of RO is to find an optimal or near optimal solution which is feasible for every possible realization of the uncertain scenario [10, 11, 15]. In Subsection 4.1.3, we introduce a robustification of CMARS with robust optimization under uncertainty sets which have special shapes.

### 4.1.2 CMARS Model with Noisy Input Data

Although all of the real-world data (both input and output variables) can include noise, data are assumed to contain fixed variables for MARS and CMARS methods. However, in this chapter, we assume that not only output data but also input data include noise and we refine our CMARS algorithm using different notations.

For CMARS, the large model that has the maximum number of BFs, $M_{\max}$, is created by Salford MARS [54]. The following general model is considered to represent the relation between the input variables and the response:

$$Y = f(\underbrace{\breve{\boldsymbol{X}}}_{\text{noisy data}}) + \varepsilon, \tag{4.1.1}$$

where $Y$ is the response variable, $\breve{\boldsymbol{X}} = (\breve{X}_1, \breve{X}_2, ..., \breve{X}_p)^T$ is a vector of predictor variables, and $\varepsilon$ is an additive stochastic component which is assumed to have zero mean and finite variance. The aim is to build reflected pairs for each input variable $\breve{X}_j$ $(j = 1, 2, ..., p)$ with $p$-dimensional knots $\boldsymbol{\tau}_i = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,p})^T$ ($i$ = 1, 2,…,$N$) at or just nearby each input data vectors. Moreover, $\breve{X}_j$ are assumed to be normally distributed random variables. Here, the following general model is considered for each input $\breve{X}_j$:

$$\breve{X}_j = \bar{X} + \xi_j{}^{2} \qquad (j = 1, 2, ..., p). \tag{4.1.2}$$

So, the multiplicative form of the $m$th BF can be written as

$$\psi_m(\breve{\boldsymbol{x}}_i) := \prod_{j=1}^{K_m} (\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m})_{\pm} \ \text{ for } i = 1, 2, ..., N, \tag{4.1.3}$$

where $K_m$ is the number of truncated linear functions multiplied in the $m$th BF (Section 3.2). Then, for the CMARS model, PRSS will have the following representation:

$$\begin{aligned} PRSS \ := & \sum_{i=1}^{N} (\breve{y}_i - f(\breve{\boldsymbol{x}}_{\boldsymbol{i}}))^2 \\ & + \sum_{m=1}^{M_{\max}} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T = (\theta_1, \theta_2)}}^{2} \sum_{\substack{r<s \\ r,s \in V(m)}} \int \alpha_m^2 [D_{r,s}^{\theta} \psi_m(\boldsymbol{t}^m)]^2 d\boldsymbol{t}^m, \end{aligned} \tag{4.1.4}$$

[2] For the random variable $\xi_j$ $(j = 1, 2, ..., p),$ we do not make a special distribution assumption. However, in the case of ellipsoidal uncertainty which we introduce, but not fully work out in this thesis (since we focus more on polyhedral uncertainty), we usually suppose a normal distribution.

After using the discretization to approximate the multi-dimensional integrals $\int \alpha_m^2 \left[ D_{r,s}^{\theta} \psi_m(t^m) \right]^2 dt^m$ [89], our PRSS will be as follows:

$$PRSS \approx \left\| \breve{\boldsymbol{y}} - \boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 + \phi \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 . \tag{4.1.5}$$

Here, *PRSS* problem looks like a classical TR problem with $\phi \geq 0$, i.e., $\phi = \lambda^2$ for some $\lambda \in \mathbb{R}$. Then, it can be coped with the CQP (refer to Section 3.2 for more detail).

### 4.1.3 Robustification of CMARS

#### 4.1.3.1 Introduction

CMARS models depend on the parameters. Small perturbations in data may give different model parameters. This may cause unstable solutions. In CMARS, the aim is to reduce the estimation error while keeping efficiency as high as possible. In order to achieve this aim, we use some approaches like scenario optimization, robust counterpart and usage of more robust estimators [90, 91]. By using robustification in CMARS, we try to decrease the estimation variance.

In CMARS model, to reduce the complexity of the regression method MARS, which especially means sensitivity with respect to noise in the data, we do a penalization in the form of TR and studied it as a CQP problem. Regularization from CMARS is already some kind of robustification, however, in our study, we additionally robustify CMARS with the help of Robust Optimization approach (Section 2.6), which is some kind of regularization in the input *and* output domain. Therefore, we have some changes in the part of $\left\| \breve{\boldsymbol{y}} - \boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2$, when we do our robustification of CMARS for both the input and output data by including uncertainty with the help of Robust Optimization. We, however, need not any change in the integration function of complexity part of PRSS model (see Subsection 4.1.2). Therefore, the part of $\left\| \boldsymbol{L}\alpha \right\|_2^2$ is the same as in CMARS.

The robust optimization approach is based on making the optimization models robust regarding constraint violations by solving *robust counterparts* of these problems within prespecified uncertain sets for the uncertain parameters [31]. Since the worst-case scenarios are defined in ways, which do *not* lead to overly conservative formulations, robust counterparts are worst-case formulations of the original problem in terms of deviations of the parameters from their nominal values [31]. Robust counterparts are solved for the worst-case realization of the uncertain parameters based on suitably uncertainty sets predetermined for the random uncertain parameters.

#### 4.1.3.2 Selecting The Shape of Uncertainty Sets

Robust optimization needs problems to remain feasible for any values of the uncertain parameters within the uncertainty set which is specified for problems. Uncertainty set is based on statistical estimates and probabilistic guarantees on the solution [31]. If it has a special shape, then the robust optimization problem may be solved efficiently. Special shapes for uncertainty set can be *ellipsoidal* or *polyhedral* (we refer to Subsubsection 2.6.4.3, for more details).

When ellipsoidal uncertainty sets are employed, robustification is more successful than employing of polyhedral uncertainty sets [17]. Nevertheless, using ellipsoidal uncertainty sets exhibits an increase in complexity for optimization problems. In fact, robust linear optimization problems (LPs) become CQP problems, robust CQPs become semidefinite optimization problems (SDPs) and robust SDPs become NP-hard to solve under ellipsoidal uncertainty sets. In contrast, robust LPs remain LPs, robust CQPs remains CQPs and robust SDPs remain SDPs under polyhedral uncertainty sets [5]. In this thesis, we study our robust CQP (SCOP) and we shall find out that it remains CQP. Therefore, we will guarantee polyhedral uncertainty sets by an interval concept for input and output data in our model; our robust CQP (SCOP) will be traced back directly as a standard conic quadratic program. Therefore, we only focus on *polyhedral uncertainty* with different uncertain scenarios.

### 4.1.4 CMARS Model with Uncertainty

To make a robustification of CMARS, we employ robust optimization on BFs in the model which are created by MARS, and we assume that the input and output variables of our CMARS model are represented by random variables (Section 4.1.2). They lead us to *uncertainty sets*, which are assumed to contain *confidence intervals* (*CIs*).

To employ robust optimization on the CMARS model that has these BFs, we incorporate a "*perturbation"* (*uncertainty*) into the real input data $\breve{\boldsymbol{x}}_i$ in each dimension, and into the output data $y_i$. The perturbation of input and output data will be represented by CIs.

When we consider that we have *p*-dimensional input data, each input data vectors $\breve{\boldsymbol{x}}_i = (\breve{x}_{i,1}, \breve{x}_{i,2}, ..., \breve{x}_{i,p})^T$ is represented as $\breve{\breve{\boldsymbol{x}}}_i = (\breve{\breve{x}}_{i,1}, \breve{\breve{x}}_{i,2}, ..., \breve{\breve{x}}_{i,p})^T$, including the perturbation $\boldsymbol{\Delta}_i = (\Delta_{i,1}, \Delta_{i,2}, ..., \Delta_{i,p})^T$ $(i=1,2,...,N)$. Since $\breve{x}$ can be outliers and perturbations of outliers are not meaningful, for our problem we, instead, refer to $\bar{x}$, the average (mean) of the input data $\breve{x}$, as the value wherever we use $\breve{x}$. Here, $\mathbf{\Delta}_i$ is the element of $U_1$ which is the uncertainty set for our input data. So our new values of piecewise linear BFs are shown in the following:

$$\breve{x}_{ij} \to \breve{\breve{x}}_{ij} \quad ; \qquad \breve{\breve{x}}_{ij} = \bar{x}_j + \Delta_{ij}, \ \left|\Delta_{ij}\right| \le \rho_{ij} \ (j = 1, 2, ..., p; \ i{=}1, 2, ..., N). \tag{4.1.6}$$

Here, $\bar{x}_j$ is the mean of the vector $\breve{\boldsymbol{x}}_j$ $(j = 1, 2, ..., p)$ and

$$\bar{x}_j := \frac{1}{N}\sum_{l=1}^{N} \breve{x}_{lj}. \tag{4.1.7}$$

The amount of perturbation in each dimension is restricted by $\rho_{ij}$ which is the semilength of the CI for input data.

When we incorporate a "*perturbation*" (*uncertainty*) into output data, our output data vector $\breve{\boldsymbol{y}} = (\breve{y}_1, \breve{y}_2, ..., \breve{y}_N)^T$ is represented as $\breve{\breve{\boldsymbol{y}}} = (\breve{\breve{y}}_1, \breve{\breve{y}}_2, ..., \breve{\breve{y}}_N)^T$, including the perturbation $\mathbf{H} = (\mathrm{H}_1, \mathrm{H}_2, ..., \mathrm{H}_N)^T$. Since $\breve{y}$ can be outliers and since perturbations of outliers are not meaningful, for our problem, we, instead, refer to $\bar{y}$, the average (mean) of the output data $\breve{y}$, as the value wherever we use $\breve{y}$. Here, we restrict vector $\mathbf{H}$ to be elements of $U_2$ which is the uncertainty set for our output data. So, our new output values can be represented as follows:

$$\breve{y}_i \to \breve{\breve{y}}_i \quad ; \qquad \breve{\breve{y}}_i = \bar{y} + \mathrm{H}_i, \ \left|\mathrm{H}_i\right| \le \nu_i \ (i{=}1, 2, ..., N). \tag{4.1.9}$$

Here, $\bar{y}$ is the mean of the vector $\breve{\boldsymbol{y}}$ and

$$\bar{y} := \frac{1}{N}\sum_{l=1}^{N} \breve{y}_l \ . \tag{4.1.10}$$

The amount of perturbation in each dimension is restricted by $\nu_i$ which is the semilength of the CI for the output variables.

MARS employs expansions of piecewise linear BFs based on the new data set that implies uncertainty. We prefer the following notation for the piecewise linear BFs:

$$c^{+}(\breve{\breve{x}}, \tau) = \left[+(\breve{\breve{x}} - \tau)\right]_+, \ c^{-}(\breve{\breve{x}}, \tau) = \left[-(\breve{\breve{x}} - \tau)\right]_+ \tag{4.1.11}$$

where, $[q]_+ = \max\ \{0, q\}$, $[q]_- = \max\ \{0, -q\}$, and $\tau$ is a univariate knot. Incorporating the uncertainty sets $U_1 \subseteq \mathbb{R}^{N \times M_{\max}}$, and $U_2 \subseteq \mathbb{R}^N$, defined in Section 4.1.6, into the data $(\breve{\breve{\boldsymbol{x}}}_i, \breve{\breve{y}}_i)$ $(i = 1, 2, ..., N)$, the multiplicative form of the $m$th BF can be represented as:

$$\psi_m(\breve{\breve{\boldsymbol{x}}}_i) := \prod_{j=1}^{K_m} [\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_\pm \quad \text{for } i = 1, 2, ..., N. \tag{4.1.12}$$

Here, we can write and estimate $[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_+$ and $[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_-$ in the subsequent ways. Firstly, for the positive part, we get:

$$\begin{aligned}
[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{i\kappa_j^m}]_+ &= \max\{0, \breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}\} = \max\{0, (\overline{x}_{\kappa_j^m} + \Delta_{i\kappa_j^m}) - \tau_{\kappa_j^m}\} \\
&\leq \max\{0, \overline{x}_{\kappa_j^m} - \tau_{\kappa_j^m}\} + \max\{0, \Delta_{i\kappa_j^m}\} \\
&= \max\{0, \frac{1}{N}\sum_{l=1}^{N} \breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m}\} + \max\{0, \Delta_{i\kappa_j^m}\} \\
&= \max\{0, \frac{1}{N}\sum_{l=1}^{N} (\breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m})\} + \max\{0, \Delta_{i\kappa_j^m}\} \\
&\leq \frac{1}{N}\sum_{l=1}^{N} \max\{0, \breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m}\} + \max\{0, \Delta_{i\kappa_j^m}\}.
\end{aligned}$$

We can evaluate $\frac{1}{N}\sum_{l=1}^{N} \max\{0, \breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m}\}$ as follows with the control variable $\mathrm{A}^+_{i\kappa_j^m}$ :

$$\frac{1}{N}\sum_{l=1}^{N} \max\{0, \breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m}\} \leq \mathrm{A}^+_{i\kappa_j^m} + \max\{0, \breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}\}.$$

When we use the idea above, we can state:

$$\begin{aligned}
\frac{1}{N}\sum_{l=1}^{N} \max\{0, \breve{x}_{l\kappa_j^m} - \tau_{\kappa_j^m}\} + \max\{0, \Delta_{i\kappa_j^m}\} &\leq \mathrm{A}^+_{i\kappa_j^m} + \max\{0, \breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}\} + \max\{0, \Delta_{i\kappa_j^m}\} \\
&= [\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_+ + (\Delta_{i\kappa_j^m})_+ + \mathrm{A}^+_{i\kappa_j^m}.
\end{aligned}$$

So, $[\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_+$ can be estimated in the following form:

$$[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_+ \leq [\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_+ + (\Delta_{i\kappa_j^m})_+ + \mathrm{A}^+_{i\kappa_j^m}. \tag{4.1.13}$$

Similarly, for the negative part, we obtain:

$$\begin{aligned}
[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_- &= \max\{0, \tau_{\kappa_j^m} - \breve{\breve{x}}_{i\kappa_j^m}\} = \max\{0, \tau_{\kappa_j^m} - (\overline{x}_{\kappa_j^m} + \Delta_{i\kappa_j^m})\} \\
&\leq \max\{0, \tau_{\kappa_j^m} - \overline{x}_{\kappa_j^m}\} + \max\{0, -\Delta_{\kappa_j^m}\} \\
&= \max\{0, \tau_{\kappa_j^m} - \frac{1}{N}\sum_{l=1}^{N} \breve{x}_{l\kappa_j^m}\} + \max\{0, -\Delta_{i\kappa_j^m}\} \\
&= \max\{0, \frac{1}{N}\sum_{l=1}^{N} (\tau_{\kappa_j^m} - \breve{x}_{l\kappa_j^m})\} + \max\{0, -\Delta_{i\kappa_j^m}\} \\
&\leq \frac{1}{N}\sum_{l=1}^{N} \max\{0, \tau_{\kappa_j^m} - \breve{x}_{l\kappa_j^m}\} + \max\{0, -\Delta_{i\kappa_j^m}\}.
\end{aligned}$$

We can evaluate $\frac{1}{N}\sum_{l=1}^{N} \max\{0, \tau_{\kappa_j^m} - \breve{x}_{l\kappa_j^m}\}$ as follows with the control variable $\mathrm{A}^-_{i\kappa_j^m}$ :

$$\frac{1}{N}\sum_{l=1}^{N} \max\{0, \tau_{\kappa_j^m} - \breve{x}_{l\kappa_j^m}\} \leq \mathrm{A}^-_{i\kappa_j^m} + \max\{0, \tau_{\kappa_j^m} - \breve{x}_{i\kappa_j^m}\}.$$

When we use the idea stated above, we can understand:

$$\begin{aligned}
\frac{1}{N}\sum_{l=1}^{N} \max\{0, \tau_{\kappa_j^m} - \breve{x}_{l\kappa_j^m}\} + \max\{0, -\Delta_{i\kappa_j^m}\} &\leq \mathrm{A}^-_{i\kappa_j^m} + \max\{0, \tau_{\kappa_j^m} - \breve{x}_{i\kappa_j^m}\} + \max\{0, -\Delta_{i\kappa_j^m}\} \\
&= [\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_- + (\Delta_{i\kappa_j^m})_- + \mathrm{A}^-_{i\kappa_j^m}.
\end{aligned}$$

Therefore, $[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_-$ can be estimated in the following form:

$$[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_- \leq [\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}\}]_- + (\Delta_{i\kappa_j^m})_- + \mathrm{A}^-_{i\kappa_j^m}. \qquad (4.1.14)$$

When we combine (4.1.13) and (4.1.14), we can write:

$$[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_\pm \leq [\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_\pm + [\Delta_{i\kappa_j^m} + (\pm\mathrm{A}_{i\kappa_j^m})]_\pm . \qquad (4.1.15)$$

Here, $\mathrm{A}_{i\kappa_j^m} := \max\{\mathrm{A}^+_{i\kappa_j^m}, \mathrm{A}^-_{i\kappa_j^m}\}$ is regarded and applied as a *control variable*. Since the value of this control variable directly affects the size of our uncertainty set $U_1$ and our uncertainty sets are unknown but bounded, the value of $\mathrm{A}_{i\kappa_j^m}$ is restricted by $\gamma_{i\kappa_j^m}$. When we consider the *conservative* (*risk averse*) case, "worst case" for the value of $\mathrm{A}_{i\kappa_j^m}$, it will be equal to $\gamma_{i\kappa_j^m}$.

However, when the absolute value of our uncertainty set is very high, it may take too much time to find a solution or we may not find any meaningful solution for our problems. Therefore, to select the value of $\mathrm{A}_{i\kappa_j^m}$, we may consider the *risk friendly* case.

To obtain our $\psi_m(\breve{\boldsymbol{x}}_i)$ and $\psi_m(\breve{\breve{\boldsymbol{x}}}_i)$, we can employ (4.1.15) in the following form, where all the signs "+" and "-" belongs to each other, respectively, for $i = 1, 2, ..., N$:

$$\begin{aligned}
\underbrace{\prod_{j=1}^{K_m}[\breve{\breve{x}}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_\pm}_{=:\psi_m(\breve{\breve{\boldsymbol{x}}}_i)} &\leq \prod_{j=1}^{K_m}[(\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m})_\pm + ((\pm\mathrm{A}_{i\kappa_j^m}) + \Delta_{i\kappa_j^m}]_\pm \\
&= \sum_{A\subseteq\{1,...K_m,\}} \prod_{a\in A}[\breve{x}_{ia} - \tau_a]_\pm \prod_{b\in\{1,...K_m,\}/A} [(\pm\mathrm{A}_{ib}) + \Delta_{ib}]_\pm \\
&\leq \underbrace{\prod_{j=1}^{K_m}[\breve{x}_{i\kappa_j^m} - \tau_{\kappa_j^m}]_\pm}_{\psi_m(\breve{\boldsymbol{x}}_i)} + \sum_{A\underset{\neq}{\subseteq}\{1,...K_m,\}} \prod_{a\in A}[\breve{x}_{ia} - \tau_a]_\pm \prod_{b\in\{1,...K_m,\}/A} [(\pm\mathrm{A}_{ib}) + \Delta_{ib}]_\pm \ .
\end{aligned}$$

Here, we can obtain the form below with symmetry:

$$\left.\begin{aligned}
\psi_m(\breve{\breve{\boldsymbol{x}}}_i) - \psi_m(\breve{\boldsymbol{x}}_i) \leq \hat{u}_{im} \\
\psi_m(\breve{\boldsymbol{x}}_i) - \psi_m(\breve{\breve{\boldsymbol{x}}}_i) \leq \hat{\hat{u}}_{im}
\end{aligned}\right\} \Rightarrow \left|\psi_m(\breve{\breve{\boldsymbol{x}}}_i) - \psi_m(\breve{\boldsymbol{x}}_i)\right| \leq \max\{\hat{u}_{im}, \hat{\hat{u}}_{im}\}.$$

So $|u_{im}|$ can be estimated in this way:

$$\begin{aligned}
\breve{\breve{\psi}}_m \to \breve{\psi}_m \quad ; \quad |u_m| = \left|\breve{\breve{\psi}}_m - \breve{\psi}_m\right| &\leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \prod_{a\in A}|\breve{x}_{ia} - \tau_a| \prod_{b\in\{1,...,K_m\}/A} |\pm\mathrm{A}_{ib} + \Delta_{ib}| \\
&\leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \mathrm{B}_i^{|A|-1} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,...,K_m\}/A} (\gamma_{ib} + \rho_{ib}),
\end{aligned} \tag{4.1.16}$$

where for all $i = 1, 2, ..., N$,

$$\begin{aligned}
|u_{im}| &\leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \prod_{a\in A}\underbrace{|\breve{x}_{ia} - \tau_a|}_{\leq D_{ia}\rho_{ia}} \prod_{b\in\{1,...,K_m\}/A} \underbrace{\left|\pm\mathrm{A}_{i\kappa_m^j} + \Delta_{i\kappa_m^j}\right|}_{\leq\gamma_{ib}+\rho_{ib}} \leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \prod_{a\in A}\mathrm{B}_{ia}\rho_{ia} \prod_{b\in\{1,...,K_m\}/A} (\gamma_{ib} + \rho_{ib}) \\
&\leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \prod_{a\in A} \underbrace{\mathrm{B}_{ia}}_{\leq\mathrm{B}_i} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,...,K_m\}/A} (\gamma_{ib} + \rho_{ib}) \leq \sum_{A\underset{\neq}{\subseteq}\{1,...,K_m\}} \mathrm{B}_i^{|A|-1} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,...,K_m\}/A} (\gamma_{ib} + \rho_{ib}).
\end{aligned}$$

Here, $\mathrm{B}_i := \max\{\mathrm{B}_{ia} \mid a = 1, 2, ..., K_m - 1\}$ is also interpreted and employed as a *control variable*. Normally, $\mathrm{B}_i$ is equal to two, but in the case of an outlier it will be bigger than two.

The *control variable* $\mathrm{B}_i$ will change depending on outliers in the input data. When our input data $\breve{X}_j$ have no outliers, we say that all variables in the input data are in the confidence interval whose length is $2\rho$. So the value of $\mathrm{B}_i$ will be two for all variables in the input data. On the other hand, when our input data $\breve{X}_j$ have outliers, we say that some of the variables in the input data are out of the CI. For the variables in the input data that are in the CI, the value of $\mathrm{B}_i$ will again be two, but for the variables in the input data, which are out of the confidence interval, the value of this control variable will be different from two. For this case, we will have to select different values for $\mathrm{B}_i$. When we consider the conservative case, we do not want to ignore any outlier. Therefore the values of $\mathrm{B}_i$ may be very large for some variables in the input data, and the absolute values of our uncertainty set may be very high because of the values of this control variable. If the absolute value of our uncertainty set is very high, it may take too much time to find a solution or we may not find any meaningful solution for our problem at all. Consequently, instead of the conservative case, we may consider a more risk friendly case to select the values of $\mathrm{B}_i$ for the outlier case. For a visualization, see Figure 4.1:

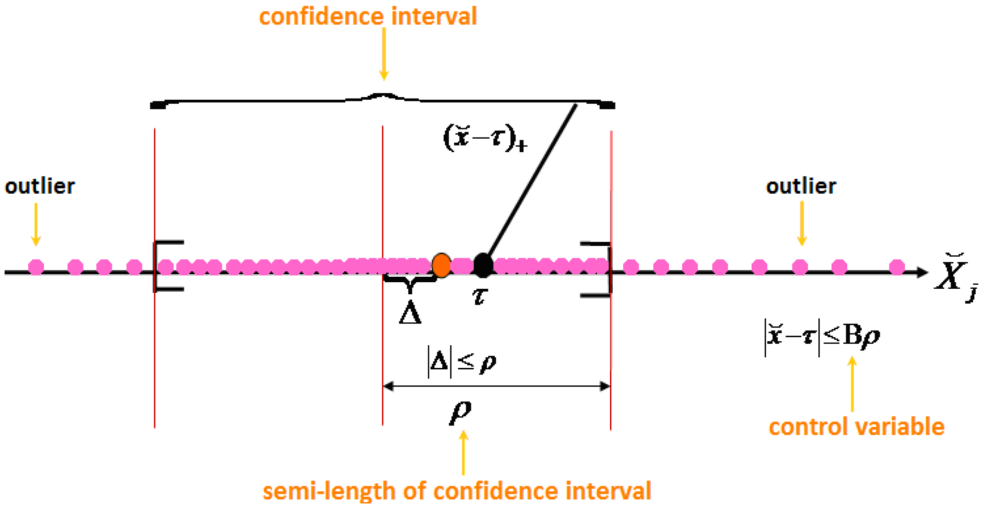


Figure 4.1 The confidence intervals of perturbation $\Delta$ and $|\breve{x}-\tau|$.

For the MARS model with uncertainty, PRSS has the following representation:

$$PRSS \;:=\sum_{i=1}^{N}(\breve{y}_i - f(\breve{\boldsymbol{x}}_i))^2+\sum_{m=1}^{M_{\max}}\phi_m \sum_{\substack{|\boldsymbol{\theta}|=1\\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2}\;\sum_{\substack{r<s\\ r,s\in V(m)}}\int \alpha_m^2[D^{\theta}_{r,s}\psi_m(\boldsymbol{t}^m)]^2\, d\boldsymbol{t}^m \;. \qquad (4.1.17)$$

Then the discretized form is used to approximate the multi-dimensional integrals $\int\limits_{Q^m} \alpha_m^2 \left[ D_{r,s}^{\theta} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m$ (see Subsection 3.2.2) and our PRSS with uncertainty can be represented as follows:

$$PRSS \approx \left\| \breve{\breve{\boldsymbol{y}}} - \boldsymbol{\psi}(\breve{\breve{\boldsymbol{b}}})\boldsymbol{\alpha} \right\|_2^2 + \phi \left\| \boldsymbol{L\alpha} \right\|_2^2 . \tag{4.1.18}$$

As we mentioned before (see Subsubsection 4.1.3.1), there is no difference for the second (complexity) part of our PRSS model after we incorporate a *"perturbation" (uncertainty)* into the real input data $\breve{\boldsymbol{x}}_i$ in any dimension and into the output data $y_i$, because we need not make any change for the function in the multi-dimensional integrals $\int\limits_{Q^m} \alpha_m^2 \left[ D_{r,s}^{\theta} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m$ .

## 4.1.5 Ellipsoidal Uncertainty and Robust Counterpart for CMARS Model

To study the robustness problem, we assume that the given model uncertainty is given by a family of matrices $\boldsymbol{\psi}(\breve{\breve{\boldsymbol{b}}}) = \boldsymbol{\psi}\left(\breve{\boldsymbol{b}}\right) + \boldsymbol{U}$ and vectors $\breve{\breve{\boldsymbol{y}}} = \breve{\boldsymbol{y}} + \boldsymbol{v}$, where $\boldsymbol{U} \in U_1$ and $\boldsymbol{v} \in U_2$ are unknown but bounded sets and $\rho, \nu \geq 0$ are the semi-lengths of our confidence sets, respectively.

Based on $U_1$ and $U_2$ are *ellipsoidal*, the robust counterpart is defined by

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \ \max_{\substack{\boldsymbol{U} \in U_1 \\ \boldsymbol{v} \in U_2}} \left\| (\boldsymbol{\psi}(\breve{\boldsymbol{b}}) + \boldsymbol{U})\boldsymbol{\alpha} - (\breve{\boldsymbol{y}} + \boldsymbol{v}) \right\|_2^2 + \phi \left\| \boldsymbol{L\alpha} \right\|_2^2 , \tag{4.1.19}$$

where

$$\boldsymbol{U} \in U_1 = \left\{ \boldsymbol{P}^{1/2}\boldsymbol{u} \ \middle| \ \boldsymbol{u} \in \mathbb{R}^{N \cdot M_{\max}}, \ \left\| \boldsymbol{u} \right\|_2 \leq \rho \right\}, \tag{4.1.20}$$

$$\boldsymbol{v} \in U_2 = \left\{ \boldsymbol{Q}^{1/2}\boldsymbol{v}' \ \middle| \ \boldsymbol{v}' \in \mathbb{R}^{N}, \ \left\| \boldsymbol{v}' \right\|_2 \leq \nu \right\}. \tag{4.1.21}$$

Here, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetric nonnegative configuration matrix and $\left\| \cdot \right\|_2$ is the Euclidean norm. Furthermore, $\boldsymbol{U} \in U_1$ and $\boldsymbol{v} \in U_2$ are defined by

$$\boldsymbol{U}=\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1M_{\max}} \\ u_{21} & u_{22} & \dots & u_{2M_{\max}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \dots & u_{NM_{\max}} \end{bmatrix} \text{ and } \boldsymbol{v}=\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}. \tag{4.1.22}$$

As we stated equations in (4.1.20) and (4.1.21), we have to use the vector form in the typical ellipsoidal sets. However, for our first uncertainty set $U_1$, we have the matrices form of $\boldsymbol{U} \in U_1$. To overcome this difficulty, we have three alternatives:

**1.** We can represent our matrix $\boldsymbol{U}$ as a vector $\boldsymbol{u}$. We have a $(N \times M_{\max})$-matrix $\boldsymbol{U}=(u_{ij})_{\substack{i=1,2,\dots,N \\ j=1,2,\dots,M_{\max}}}$ and we can write it as a vector $\boldsymbol{u}=(u_k)_{k=1,2,\dots,N\cdot M_{\max}}$, where $u_k := u_{ij}$ with $k$=$i$+($j$-1)$N$. Therefore, our matrix $\boldsymbol{U}$ can be canonically represented as a vector $\boldsymbol{u}=(u_1,u_2,\dots,u_{N\cdot M_{\max}})^T$ by successively aligning the columns of $\boldsymbol{U}$.

**2.** We can obtain an ellipsoidal set for each column vector in the matrix $\boldsymbol{U}$ and we have $M_{\max}$ different ellipsoidal sets. Then, we can use the sum of these $M_{\max}$ ellipsoidal sets [14].

**3.** We can use a different form of ellipsoidal set, which is defined in [91], as follows:

$$\boldsymbol{U} \in U_1 = \left\{ \boldsymbol{U}' \in \mathbb{R}^{NxM_{\max}} \;\middle|\; \frac{N\text{-}1}{2} \left\| \boldsymbol{P}^{-\frac{1}{2}}(\boldsymbol{U}'-\hat{\boldsymbol{U}}')\boldsymbol{P}^{-\frac{1}{2}} \right\|_{tr} \le \rho \right\}. \tag{4.1.23}$$

Here, $\|\boldsymbol{A}\|_{tr}$ is the trace norm defined as $\|\boldsymbol{A}\|_{tr}^2 = \text{trace}(\boldsymbol{A}^T\boldsymbol{A})$, where trace$(\boldsymbol{B}) :=$ $\sum_{i=1}^{n} b_{ii}$ $\;(\boldsymbol{B}=(b_{ij})_{i,j=1,2,\dots,n})$.

If we select the first alternative explained above, based on the uncertainty sets $U_1$ and $U_2$ from the equations (4.1.20) and (4.1.21), our robust counterpart (4.1.19) is redefined as

$$\underset{\alpha}{\text{minimize}} \max_{\substack{\|\boldsymbol{u}\|_2 \le \rho \\ \|\boldsymbol{v}'\|_2 \le \nu}} \left\| \breve{\boldsymbol{y}} + \boldsymbol{Q}^{1/2}\boldsymbol{v}' - (\boldsymbol{\psi}(\breve{\boldsymbol{b}}) + \boldsymbol{P}^{1/2}\boldsymbol{u})\boldsymbol{\alpha} \right\|_2^2 + \phi \|\boldsymbol{L\alpha}\|_2^2. \tag{4.1.24}$$

When we apply Cauchy–Schwarz inequality in its form for both vectors and matrices, we can estimate $\left\|(\breve{\boldsymbol{y}}+\boldsymbol{Q}^{1/2}\boldsymbol{v}')-(\boldsymbol{\psi}(\breve{\boldsymbol{b}})+\boldsymbol{P}^{1/2}\boldsymbol{u})\boldsymbol{\alpha}\right\|_2^2$ in the following form:

$$\left\|\breve{\breve{\boldsymbol{y}}}-\boldsymbol{\psi}(\breve{\breve{\boldsymbol{b}}})\boldsymbol{\alpha}\right\|_2^2 = \left\|(\breve{\boldsymbol{y}}+\boldsymbol{Q}^{1/2}\boldsymbol{v}')-(\boldsymbol{\psi}(\breve{\boldsymbol{b}})+\boldsymbol{P}^{1/2}\boldsymbol{u})\boldsymbol{\alpha}\right\|_2^2$$

$$\leq \left\|\breve{\boldsymbol{y}}-(\boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha}\right\|_2^2+\left\|\boldsymbol{P}^{1/2}\boldsymbol{\alpha}\right\|_2^2\underbrace{\left\|\boldsymbol{u}\right\|_2^2}_{\leq\rho^2}+\left\|\boldsymbol{Q}^{1/2}\right\|_2^2\underbrace{\left\|\boldsymbol{v}'\right\|_2^2}_{\leq\nu^2}\leq\left\|(\boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha}-\breve{\boldsymbol{y}}\right\|_2^2+\rho^2\left\|\boldsymbol{P}^{1/2}\boldsymbol{\alpha}\right\|_2^2+\nu^2\left\|\boldsymbol{Q}^{1/2}\right\|_2^2.$$

Therefore, our robust counterpart is finally defined as

$$\underset{\boldsymbol{\alpha}}{\text{minimize}}\ \left\|\breve{\boldsymbol{y}}-\boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha}\right\|_2^2+\rho^2\left\|\boldsymbol{P}^{1/2}\boldsymbol{\alpha}\right\|_2^2+\lambda\left\|\boldsymbol{L}\boldsymbol{\alpha}\right\|_2^2+\nu^2\left\|\boldsymbol{Q}^{1/2}\right\|_2^2. \tag{4.1.25}$$

The complexity of optimization problems increase if an ellipsoidal uncertainty set is employed. In fact, our robust CQPs become SDPs under ellipsoidal uncertainty sets [6] (Section 2.6.2). However, since we continue to study our optimization problem as a CQP problem, in this study, we use the form of *polyhedral sets* as our uncertainty sets.

### 4.1.6 Polyhedral Uncertainty and Robust Counterpart for CMARS Model

In order not to increase the complexity of our regarded optimization problem (see Subsubsection 4.1.3.2), we choose the uncertainty sets $U_1$ and $U_2$ as *polyhedral* for input and output data in our model to study our robustness problem. Based on these sets $U_1$ and $U_2$, the robust counterpart is defined as follows:

$$\underset{\boldsymbol{\alpha}}{\text{minimize}}\ \max_{\substack{\boldsymbol{W}\in U_1\\ \boldsymbol{z}\in U_2}}\ \left\|\boldsymbol{W}\boldsymbol{\alpha}-\boldsymbol{z}\right\|_2^2+\phi\left\|\boldsymbol{L}\boldsymbol{\alpha}\right\|_2^2. \tag{4.1.26}$$

Here, $U_1$ is a polytope with $2^{N\times M_{\max}}$ vertices $\boldsymbol{W}^1,\boldsymbol{W}^2,...,\boldsymbol{W}^{2^{N\times M_{\max}}}$. It is not exactly known, but belongs to a convex bounded uncertain domain $U_1$ given by

$$U_1=\left\{\sum_{j=1}^{2^{N\times M_{\max}}}\delta_j\boldsymbol{W}^j\ \middle|\ \delta_j\geq 0\ (j\in\{1,2,...,2^{N\times M_{\max}}\}),\ \sum_{j=1}^{2^{N\times M_{\max}}}\delta_j=1\right\}, \tag{4.1.27}$$

where $U_1 = \text{conv}\{\boldsymbol{W}^1, \boldsymbol{W}^2 \ldots, \boldsymbol{W}^{2^{N\times M_{\max}}}\}$ is the convex hull.

Furthermore, $U_2$ is a polytope with $2^N$ vertices $\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^{2^N}$. It is not exactly known, but belongs to a bounded uncertain domain $U_2$ given by

$$U_2 = \left\{ \sum_{i=1}^{2^N} \varphi_i \boldsymbol{z}^i \;\middle|\; \varphi_i \ge 0 \; (i \in \{1, 2, \ldots, 2^N\}), \; \sum_{i=1}^{2^N} \varphi_i = 1 \right\}, \qquad (4.1.28)$$

where $U_2 = \text{conv}\{\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^{2^N}\}$ is the convex hull.

Any uncertainty sets $U_1$ and $U_2$ can be represented as a convex combination of vertices $\boldsymbol{W}^j$ ($j$=1,2,…,$2^{N\times M_{\max}}$) and $\boldsymbol{z}^i$ ($i$=1,…,$2^N$) of the polytope. The entries are found to have become intervals. Therefore, our matrix $\boldsymbol{W}$ and vector $\boldsymbol{z}$ with uncertainty are lying in the Cartesian product of intervals that are parallelpipes. To give an easy illustration, the Cartesian product of intervals in general and, especially, for three entries, can be represented by Figure 4.2.

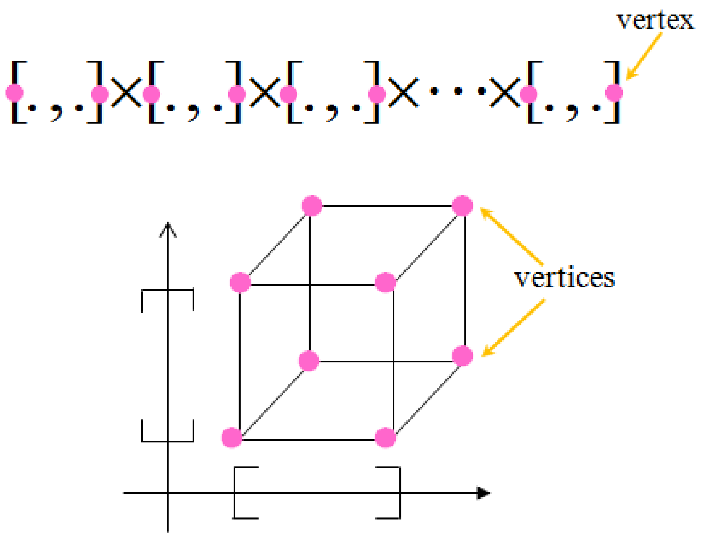


Figure 4.2 Cartesian product of intervals for 3 entries.

Here, our matrix $\boldsymbol{W}$ is represented as a vector with uncertainty which generates a parallelpipe $C$. We have a $(N \times M_{\max})$-matrix $\boldsymbol{W} = (u_{ij})_{\substack{i=1,2,\ldots,N \\ j=1,2,\ldots,M_{\max}}}$ and we can write it as a vector

$\boldsymbol{t} = (t_k)_{k=1,2,...,N\times M_{\max}}$ , where $t_k := u_{ij}$ with $k$=$i$+($j$-1)$N$. So, our matrix $\boldsymbol{W}$ can be canonically represented as a vector $\boldsymbol{t}_k = (t_1, t_2, ..., t_{N\times M_{\max}})^T$ by successively aligning the columns of $\boldsymbol{W}$.

The input data matrix represented as a vector ***t*** with uncertainty which generates a parallelpipe *C*. Let $C$ be a parallelpipe that encompasses entries of input data; we represent it by

$$C = [\breve{x}_1^l, \breve{x}_1^u] \times [\breve{x}_2^l, \breve{x}_2^u] \times .... \times [\breve{x}_{N\times M_{\max}}^l, \breve{x}_{N\times M_{\max}}^u] = \prod_{i=1}^{N\times M_{\max}} C_i, \tag{4.1.29}$$

where $C_i =: [\breve{x}_i^l, \breve{x}_i^u]$, $\breve{x}_i^l \le \breve{x}_i \le \breve{x}_i^u$; $\breve{x}_i^l$ is the lower bound and $\breve{x}_i^u$ is the upper bound of the confidence intervals in the input dimension ($i$=$1, 2, ..., N\times M_{\max}$).

The output data vector includes entries $\breve{\boldsymbol{y}}_j = (\breve{y}_1, \breve{y}_2, ..., \breve{y}_N)^T$ with uncertainty which generates a parallelpipe *D*. Let $D$ be a parallelpipe that encompasses entries of output data; we represent it by

$$D = [\breve{y}_1^l, \breve{y}_1^u] \times [\breve{y}_2^l, \breve{y}_2^u] \times .... \times [\breve{y}_N^l, \breve{y}_N^u] = \prod_{j=1}^{N} D_j, \tag{4.1.30}$$

where $D_j =: [\breve{y}_j^l, \breve{y}_j^u]$, $\breve{y}_j^l \le \breve{y}_j \le \breve{y}_j^u$ , $\breve{y}_j^l$ is the lower bound and $\breve{y}_j^u$ is the upper bound of the confidence intervals in the output dimension ( $j$=$1, 2, ..., N$ ). Here, the symbols " × " and " ∏ " are used for Cartesian product.

### 4.1.7 Robust CQP with Polyhedral Uncertainty

For our CMARS model, the optimization problem is written as follows:

$$\begin{aligned} &\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \ t, \\ &\text{subject to } \left\|\boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha} - \breve{\boldsymbol{y}}\right\|_2 \le t, \\ &\qquad\qquad \left\|\boldsymbol{L}\boldsymbol{\alpha}\right\|_2 \le \sqrt{\breve{M}}. \end{aligned} \tag{4.1.31}$$

When *polyhedral* uncertainty is used for our CMARS model, based on uncertainty sets $U_1$ and $U_2$ from the equations (4.1.27) and (4.1.28), the robust counterpart is defined by:

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \max_{\substack{\boldsymbol{W} \in U_1 \\ \boldsymbol{z} \in U_2}} \; \left\| \boldsymbol{W}\boldsymbol{\alpha} - \boldsymbol{z} \right\|_2^2 + \phi \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2^2 . \tag{4.1.32}$$

So, robust CQP for our optimization problem is represented in the following form:

$$\begin{aligned} &\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \quad t, \\ &\text{subject to} \quad \left\| \boldsymbol{W}\boldsymbol{\alpha} - \boldsymbol{z} \right\|_2 \le t \quad \forall \quad \underset{= \sum_{j=1}^{M_{\max}} \delta_j \boldsymbol{W}^j}{\boldsymbol{W}} \in U_1, \quad \underset{= \sum_{i=1}^{N} \varphi_i \boldsymbol{z}^i}{\boldsymbol{z}} \in U_2, \\ &\qquad\qquad \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\|_2 \le \sqrt{\breve{M}} . \end{aligned} \tag{4.1.33}$$

If $U_1$ and $U_2$ are polytopes which are described by their vertices:

$$\begin{aligned} U_1 &= \text{conv}\left\{ \boldsymbol{W}^1, \boldsymbol{W}^2, ..., \boldsymbol{W}^{M_{\max}} \right\}, \\ U_2 &= \text{conv}\left\{ \boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^N \right\}, \end{aligned} \tag{4.1.34}$$

then, our robust CQP can be equivalently represented as a standard conic quadratic program (Section 2.6.3):

$$\begin{aligned} &\underset{t,\boldsymbol{\alpha}}{\text{minimize}} \quad t, \\ &\text{subject to} \quad \left\| \boldsymbol{W}^j \boldsymbol{\alpha} \text{ - } \boldsymbol{z}^i \right\|_2 \le t \quad (i = 1, 2, ..., N \;;\; j = 1, 2, ..., M_{\max}), \\ &\qquad\qquad \left\| \boldsymbol{L}\boldsymbol{\alpha} \right\| \le \sqrt{\breve{M}} . \end{aligned} \tag{4.1.35}$$

Let us use modern methods of *continuous optimization techniques*, especially, from CQP where the basic notation is used [5]

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \;\; \mathbf{c}^T \boldsymbol{x}, \\ &\text{subject to} \;\; \left\| \boldsymbol{D}_i \boldsymbol{x} - \boldsymbol{d}_i \right\| \le \boldsymbol{p}_i^T \boldsymbol{x} - q_i \qquad (i=1,2,...,k). \end{aligned} \tag{4.1.36}$$

In fact, we see that our optimization problem is such a CQP program with

$$\begin{aligned} &\boldsymbol{c} = (1, \mathbf{0}_{M_{\max}+1}^T)^T, \; x = (t, \boldsymbol{\alpha}^T)^T, \; \boldsymbol{D}_1 = (\mathbf{0}_N, \boldsymbol{W}^j), \; \boldsymbol{d}_1 = \boldsymbol{z}^i, \; \boldsymbol{p}_1 = (1, 0, ..., 0)^T, \; q_1 = 0, \\ &\boldsymbol{D}_2 = (\mathbf{0}_{M_{\max}+1}, \boldsymbol{L}), \boldsymbol{d}_2 = \mathbf{0}_{M_{\max}+1}, \; \boldsymbol{p}_2 = \mathbf{0}_{M_{\max}+2} \text{ and } q_2 = -\sqrt{\breve{M}} . \end{aligned} \tag{4.1.37}$$

In order to write the optimality conditions for this problem, we reformulate the problem (4.1.35) as follows:

$$\underset{t,\boldsymbol{\alpha}}{\text{minimize}}\ t,$$

such that

$$\chi^{i,j} := \begin{pmatrix} \mathbf{0}_N & \boldsymbol{W}^j \\ 1 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} -\boldsymbol{z}^i \\ 0 \end{pmatrix},$$

$$\boldsymbol{\eta} := \begin{pmatrix} \mathbf{0}_{M_{\max}+1} & \boldsymbol{L} \\ 0 & \mathbf{0}^T_{M_{\max}+1} \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{M_{\max}+1} \\ \sqrt{\breve{M}} \end{pmatrix}, \tag{4.1.38}$$

$$\chi^{i,j} \in L^{N+1},\ \boldsymbol{\eta} \in L^{M_{\max}+2},$$

where $L^{N+1}$, $L^{M_{\max}+2}$ are the $(N+1)$- and $(M_{\max}+2)$-dimensional ice-cream (*or second-order, or Lorentz*) cones (refer to Subsubsection 3.2.4.1 for more details).

## 4.2 Numerical Example of Robust CMARS

### 4.2.1 Description of Data Set

Knowledge of components which are responsible for the quality-defining aroma and flavour characteristics in wines is highly desirable to let progress in controlling and improving wine quality [34]. To suitably understand the chemical compounds in wine which give desirable sensory characteristics, information considering both the chemical nature and the sensory properties of a wine, or of those components in the wine, is required. The intelligent combination of these two different types of information is a powerful strategy which can give large gains in knowledge.

A major aim of research efforts in trying to distinguish components of importance to wine flavour from a practical is to establish sensory-instrumental correlations [34]. When these correlations are adequately robust, in that a practical guide can be obtained considering the quality of a wine given its composition, the industry may benefit from the definition of such correlations.

The data set, which is used for explaining the implementation of RCMARS algorithm, is taken from Minitab package program [56] as a sample of *wine quality data*. In these data set, for the quality-determining aroma, we can only use three chemical components as predictor variables $(x_1, x_2, x_3)$ because input data and output data are assumed to be normally distributed in this study. These chemical components are *cadmium* (Cd), *chrome* (Cr) and *calcium* (Ca) and they have 20 observations. Here $\boldsymbol{x}$ is written as a *generic* variable in the corresponding space $\mathbb{R}^n$ ($n \in \{1,2,3\}$). Later on $\boldsymbol{x}$ will substituted by $\boldsymbol{t}^1, \boldsymbol{t}^2, ..., \boldsymbol{t}^5$.

### 4.2.2 Obtaining Large Model from MARS Program

In order to implement RCMARS algorithm, first, the MARS model is constructed by using the Salford MARS version 3 [54], and then, the maximum number of BFs ($M_{\max}$) and the highest degree of interactions are determined by trial and error. In our example, $M_{\max}$ is assigned to be five and the highest degree of interaction is assigned to be two. Then the largest model, which is constructed in the forward MARS algorithm by the software, includes the BFs as follows:

$$\begin{aligned}
\psi_1(\boldsymbol{x}) &= \max\{0, x_3 + 0.0459)\},\\
\psi_2(\boldsymbol{x}) &= \max\{0, -0.0459 - x_3\}\\
\psi_3(\boldsymbol{x}) &= \max\{0, x_2 + 0.1396\},\\
\psi_4(\boldsymbol{x}) &= \max\{0, -0.1396 - x_2)\},\\
\psi_5(\boldsymbol{x}) &= \max\{0, x_1 + 1.6729\} \cdot \max\{0, -0.0459 - x_3\}.
\end{aligned}$$

Here, $\psi_1$ and $\psi_2$ are the standard BFs and reflected (mirror image) BFs for the predictor variable $x_3$, respectively (presented in Figure 4.3). Similarly, $\psi_3$ and $\psi_4$ are the standard BFs and the reflected BFs for the predictor variable $x_2$, respectively (given in Figure 4.4).

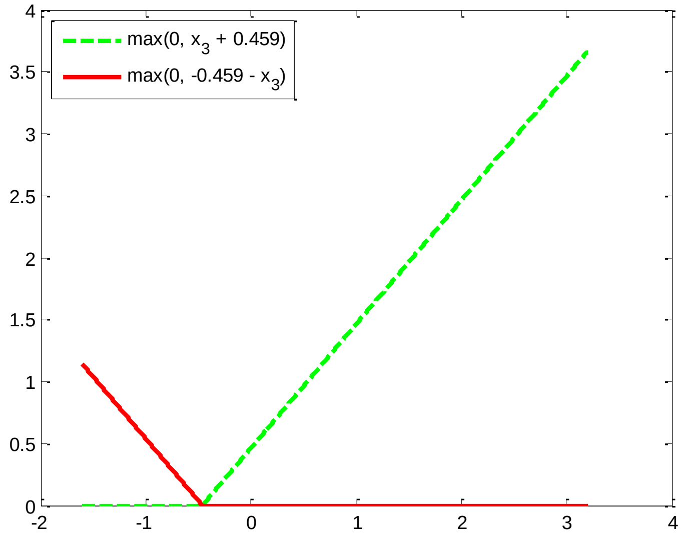


Figure 4.3 The graphical illustration of $\psi_1$ and $\psi_2$.

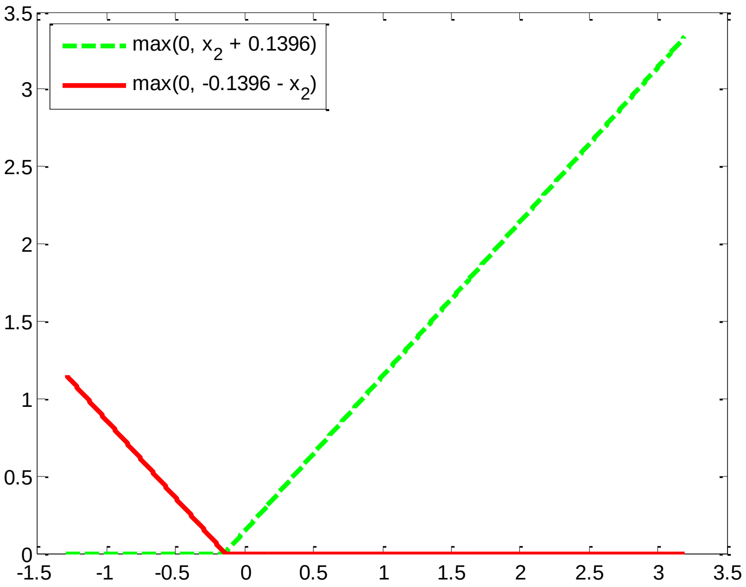


Figure 4.4 The graphical illustration of $\psi_3$ and $\psi_4$.

On the other hand, BF $\psi_5$ uses the function $\psi_2$ to express the interaction between the predictor variables $x_1$ and $x_3$ (presented in Figure 4.5).

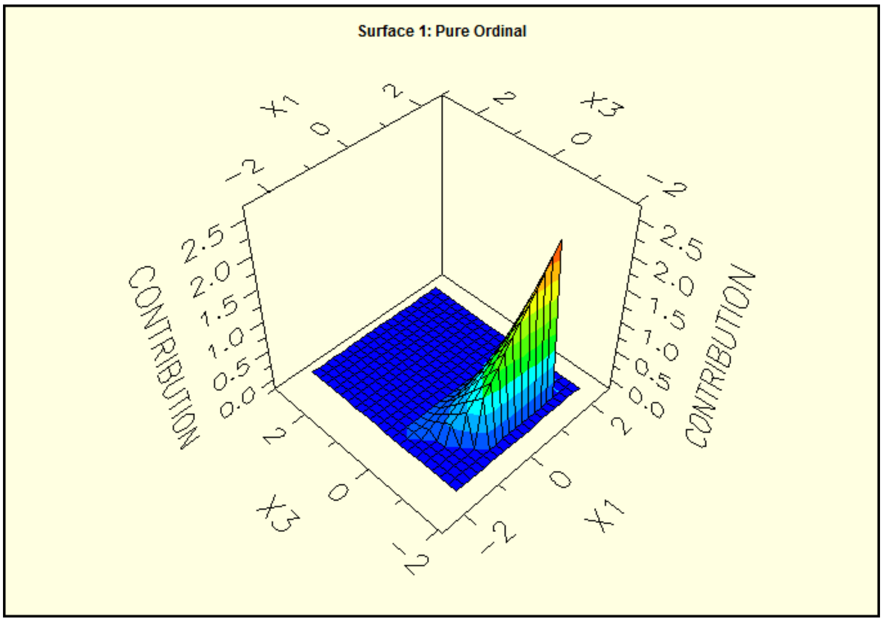


Figure 4.5 The graphical illustration of $\psi_5$ which has interactions between the predictor variables $x_1$ and $x_3$ [54].

In order to prevent our optimization problem from nondifferentiability (Subsection 3.3.2), we select the knot values different from data points. However, these knot values very close to data points. For each BF, the knot values are selected below:

$$\psi_1 \text{ and } \psi_2 \ : \left.\begin{array}{r} \tau_{10,3} = -0.0459 \\ \breve{x}_{10,3} = -0.04586 \end{array}\right\} \Rightarrow \tau_{10,3} \neq \breve{x}_{10,3}.$$

$$\psi_3 \text{ and } \psi_4 \ : \left.\begin{array}{r} \tau_{4,2} = -0.1396 \\ \breve{x}_{4,2} = -0.13961 \end{array}\right\} \Rightarrow \tau_{4,2} \neq \breve{x}_{4,2}.$$

$$\psi_5 \ : \left.\begin{array}{r} \tau_{1,1} = -1.6729 \\ \breve{x}_{1,1} = -1.67292 \end{array}\right\} \Rightarrow \tau_{1,1} \neq \breve{x}_{1,1}, \quad \left.\begin{array}{r} \tau_{10,3} = -0.0459 \\ \breve{x}_{10,3} = -0.04586 \end{array}\right\} \Rightarrow \tau_{10,3} \neq \breve{x}_{10,3}.$$

Then, the BFs of the form the equation (3.2.4), which we obtain for our numerical example, can be written as follows:

For $\psi_1 : \ K_1 = 1, \ x_{\kappa_1^1} = x_3, \ \tau_{\kappa_1^1} = -0.0459, \ s_{\kappa_1^1} = +1,$

$$\Rightarrow \ \psi_1(\boldsymbol{t}^1) = \prod_{j=1}^{K_1} [s_{\kappa_j^1}.(x_{\kappa_j^1} - \tau_{\kappa_j^1})]_+ = \prod_{j=1}^{1} [s_{\kappa_1^1}.(x_{\kappa_1^1} - \tau_{\kappa_1^1})]_+ = [(+1).(x_3 - (-0.0459))]_+,$$

For $\psi_2 : K_2 = 1, \ x_{\kappa_1^2} = x_3, \ \tau_{\kappa_1^2} = -0.0459, \ \ s_{\kappa_1^1} = -1,$

$$\Rightarrow \ \psi_2(\boldsymbol{t}^2) = \prod_{j=1}^{K_2} [s_{\kappa_j^2}.(x_{\kappa_j^2} - \tau_{\kappa_j^2})]_+ = \prod_{j=1}^{1} [s_{\kappa_1^2}.(x_{\kappa_1^2} - \tau_{\kappa_1^2})]_+ = [(\text{-}1).(x_3 - (-0.0459))]_+,$$

For $\psi_3 : K_3 = 1, \ x_{\kappa_1^3} = x_2, \ \tau_{\kappa_1^3} = -0.1396, \ \ s_{\kappa_1^3} = +1,$

$$\Rightarrow \ \psi_3(\boldsymbol{t}^3) = \prod_{j=1}^{K_3} [s_{\kappa_j^3}.(x_{\kappa_j^3} - \tau_{\kappa_j^3})]_+ = \prod_{j=1}^{1} [s_{\kappa_1^3}.(x_{\kappa_1^3} - \tau_{\kappa_1^3})]_+ = [(+1).(x_2 - (-0.1396))]_+,$$

For $\psi_4 : K_4 = 1, \ x_{\kappa_1^4} = x_2, \ \tau_{\kappa_1^4} = -0.1396, \ s_{\kappa_1^4} = -1,$

$$\Rightarrow \ \psi_4(\boldsymbol{t}^4) = \prod_{j=1}^{K_4} [s_{\kappa_j^4}.(x_{\kappa_j^4} - \tau_{\kappa_j^4})]_+ = \prod_{j=1}^{1} [s_{\kappa_1^4}.(x_{\kappa_1^4} - \tau_{\kappa_1^4})]_+ = [(\text{-}1).(x_2 - (-0.1396))]_+,$$

For $\psi_5 : K_5 = 2,\ x_{\kappa_1^5} = x_1,\ x_{\kappa_2^5} = x_3,\ \tau_{\kappa_1^5} = -1.6729,\ \tau_{\kappa_2^5} = -0.0459,\ s_{\kappa_1^5} = +1\ ,\ s_{\kappa_2^5} = -1,$

$$\Rightarrow\ \psi_5(\boldsymbol{t}^5) = \prod_{j=1}^{K_5}[s_{\kappa_j^5}.(x_{\kappa_j^5} - \tau_{\kappa_j^5})]_+ = \prod_{j=1}^{2}[s_{\kappa_1^5}.(x_{\kappa_1^5} - \tau_{\kappa_1^5})]_+.[s_{\kappa_2^5}.(x_{\kappa_2^5} - \tau_{\kappa_2^5})]$$

$$=[(+1).(x_1 - (-1.6729))]_+.[(\text{-}1).(x_3 - (-0.0459))]_+.$$

So, for our numerical example, the large model (3.2.3) becomes

$$Y = \alpha_0 + \sum_{m=1}^{M} \alpha_m \psi_m(\boldsymbol{x}) + \varepsilon$$

$$= \alpha_0 + \alpha_1\psi_1(\boldsymbol{x}) + \alpha_2\psi_2(\boldsymbol{x}) + \alpha_3\psi_3(\boldsymbol{x}) + \alpha_4\psi_4(\boldsymbol{x}) + \alpha_5\psi_5(\boldsymbol{x}) + \varepsilon$$

$$= \alpha_0 + \alpha_1 \max\{0, x_3 + 0.0459)\} + \alpha_2 \max\{0, -0.0459 - x_3\} + \alpha_3 \max\{0, x_2 + 0.1396\}$$

$$+ \alpha_4 \max\{0, -0.1396 - x_2)\} + \alpha_5 \max\{0, x_1 + 1.6729\}.\max\{0, -0.0459 - x_3\} + \varepsilon.$$

### 4.2.3 Evaluating Accuracy and Complexity of PRSS Form

For this numeric example, we can write the PRSS objective function in the equation (3.2.7) as follows:

$$PRSS = \sum_{i=1}^{20}\left(y_i - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\breve{\boldsymbol{b}}_i)\right)^2 \sum_{m=1}^{5} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int_{Q^m} \alpha_m^2 \left[D_{r,s}^{\boldsymbol{\theta}} \psi_m(\boldsymbol{t}^m)\right]^2 dt^m$$

$$= \sum_{i=1}^{20}\left(y_i - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\breve{\boldsymbol{b}}_i)\right)^2 + \phi_1 \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(1)}} \int_{Q^1} \alpha_1^2 \left[D_{r,s}^{\boldsymbol{\theta}} \psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1 \right)$$

$$+ \phi_2 \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(2)}} \int_{Q^2} \alpha_2^2 \left[D_{r,s}^{\boldsymbol{\theta}} \psi_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2 \right)$$

$$+ \phi_3 \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(3)}} \int_{Q^3} \alpha_3^2 \left[D_{r,s}^{\boldsymbol{\theta}} \psi_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3 \right)$$

$$+\phi_4\left(\sum_{\substack{|\boldsymbol{\theta}|=1\\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2}\ \sum_{\substack{r<s\\ r,s\in V(4)}}\ \int_{Q^4}\alpha_4^2\left[D_{r,s}^{\boldsymbol{\theta}}\psi_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4\right)$$

$$+\phi_5\left(\sum_{\substack{|\boldsymbol{\theta}|=1\\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2}\ \sum_{\substack{r<s\\ r,s\in V(5)}}\ \int_{Q^5}\alpha_5^2\left[D_{r,s}^{\boldsymbol{\theta}}\psi_5(\boldsymbol{t}^5)\right]^2 d\boldsymbol{t}^5\right).$$

Here, all of the evaluations for the notations $V(m)$ and $\boldsymbol{t}^m$ ($m$=1,2,…,5) in the equation are stated below:

$$V(1)=\{\kappa_j^1=1\}=\{3\},\ \boldsymbol{t}^1=(t_1^1)^T=(x_3)^T,$$
$$V(2)=\{\kappa_j^2=1\}=\{3\},\ \boldsymbol{t}^2=(t_1^2)^T=(x_3)^T,$$
$$V(3)=\{\kappa_j^1=1\}=\{2\},\ \boldsymbol{t}^3=(t_1^3)^T=(x_2)^T,$$
$$V(4)=\{\kappa_j^4=1\}=\{2\},\ \boldsymbol{t}^4=(t_1^4)^T=(x_2)^T,$$
$$V(5)=\{\kappa_j^5=1,2\}=\{1,3\},\ \boldsymbol{t}^5=(t_1^5,t_2^5)^T=(x_1,x_3)^T.$$

Besides, the derivatives for the BFs $D_{r,s}^{\boldsymbol{\theta}}\psi_m(\boldsymbol{t}^m)$ ($m$=1, 2,…,5) are given below. For our first BF $\psi_1(\boldsymbol{t}^1)=\max\{0,x_3+0.0459)\}$, $r=s=3$ because of no interaction. Therefore, our indicated first- and second-order derivatives of $\psi_1$ are

$$|\boldsymbol{\theta}|=1:\quad D_3^1\psi_1(\boldsymbol{t}^1)=\frac{\partial\psi_1}{\partial t_1^1}(\boldsymbol{t}^1)=\frac{\partial\psi_1}{\partial x_3}(x_3)=\begin{cases}1, & \text{if } x_3>-0.0459,\\ 0, & \text{if } x_3\le-0.0459,\end{cases}$$

$$|\boldsymbol{\theta}|=2:\quad D_3^2\psi_1(\boldsymbol{t}^1)=\frac{\partial^2\psi_1}{\partial t_1^1\partial t_1^1}(\boldsymbol{t}^1)=\frac{\partial^2\psi_1}{\partial x_3\partial x_3}(x_3)=0\ \text{ for all } x_3,$$

and the sum of them can be written as follows

$$\sum_{\substack{|\boldsymbol{\theta}|=1\\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2}\ \sum_{\substack{r<s\\ r,s\in V(1)}}\left[D_{r,s}^{\boldsymbol{\theta}}\psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1.$$

For our second BF $\psi_2(\boldsymbol{t}^2)=\max\{0,-0.0459-x_3\}$, $r=s=3$ due to no interaction. Consequently, our indicated first- and second-order derivatives of $\psi_2$ are

$$|\boldsymbol{\theta}| = 1: \quad D_3^1 \psi_2(\boldsymbol{t}^2) = \frac{\partial \psi_2}{\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial \psi_2}{\partial x_3}(x_3) = \begin{cases} -1, & \text{if } x_3 < -0.0459, \\ 0, & \text{if } x_3 \geq -0.0459, \end{cases}$$

$$|\boldsymbol{\theta}| = 2: \quad D_3^2 \psi_2(\boldsymbol{t}^2) = \frac{\partial^2 \psi_2}{\partial t_1^2 \partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial^2 \psi_2}{\partial x_3 \partial x_3}(x_3) = 0 \ \text{ for all } x_3,$$

and the sum of them is equal to following equation

$$\sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(2)}} \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_2(\boldsymbol{t}^2) \right]^2 d\boldsymbol{t}^2.$$

For our third BF $\psi_3(\boldsymbol{t}^3) = \max\{0, x_2 + 0.1396\}$, there is no interaction; so $r = s = 2$, and our indicated first- and second-order derivatives of $\psi_3$ are

$$|\boldsymbol{\theta}| = 1: \quad D_2^1 \psi_3(\boldsymbol{t}^3) = \frac{\partial \psi_3}{\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial \psi_3}{\partial x_2}(x_2) = \begin{cases} 1, & \text{if } x_2 > -0.1396, \\ 0, & \text{if } x_2 \leq -0.1396, \end{cases}$$

$$|\boldsymbol{\theta}| = 2: \quad D_2^2 \psi_3(\boldsymbol{t}^3) = \frac{\partial^2 \psi_3}{\partial t_1^3 \partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial^2 \psi_3}{\partial x_2 \partial x_2}(x_2) = 0 \ \text{ for all } x_2,$$

and the sum of them can be written below

$$\sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(3)}} \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_3(\boldsymbol{t}^3) \right]^2 d\boldsymbol{t}^3.$$

For our fourth BF $\psi_4(\boldsymbol{t}^4) = \max\{0, -0.1396 - x_2)\}$, since there is no interaction, $r = s = 2$. Therefore, our indicated first- and second-order derivatives of $\psi_4$ are

$$|\boldsymbol{\theta}| = 1: \quad D_2^1 \psi_4(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial t_1^4}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_2}(x_2) = \begin{cases} -1, & \text{if } x_2 < -0.1396, \\ 0, & \text{if } x_2 \geq -0.1396, \end{cases}$$

$$|\boldsymbol{\theta}| = 2: \quad D_2^2 \psi_4(\boldsymbol{t}^4) = \frac{\partial^2 \psi_4}{\partial t_1^4 \partial t_1^4}(\boldsymbol{t}^4) = \frac{\partial^2 \psi_4}{\partial x_2 \partial x_2}(x_2) = 0 \ \text{ for all } x_2,$$

and the sum of them is equal to following form

$$\sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(4)}} \left[ D_{r,s}^{\boldsymbol{\theta}} \psi_4(\boldsymbol{t}^4) \right]^2 d\boldsymbol{t}^4.$$

On the other hand, for our fifth BF $\psi_5(\boldsymbol{t}^5) = \max\{0, x_1 + 1.6729\} \cdot \max\{0, -0.0459 - x_2\}$, there is an interaction between predictors $x_1$ and $x_2$. Consequently, $r < s \Rightarrow r = 1$ and $s = 2$, and our indicated first- and second-order derivatives of $\psi_5$ are

$$|\boldsymbol{\theta}| = 1: \; D^1_{1,2}\psi_5(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial t_1^5}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_1}(x_1, x_2) = \begin{cases} \max\{0, -0.0459 - x_2\}, & \text{if } x_1 > -1.6729, \\ 0, & \text{if } x_1 \leq -1.6729, \end{cases}$$

$$D^1_{1,2}\psi_5(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial t_1^5}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_2}(x_1, x_2) = \begin{cases} -\max\{0, x_1 + 1.6729\}, & \text{if } x_2 < -0.0459, \\ 0, & \text{if } x_2 \geq -0.0459, \end{cases}$$

$$|\boldsymbol{\theta}| = 2: \; D^2_{1,2}\psi_5(\boldsymbol{t}^5) = \frac{\partial^2 \psi_5}{\partial t_1^5 \partial t_2^5}(\boldsymbol{t}^5) = \frac{\partial^5 \psi_5}{\partial x_1 \partial x_2}(x_1, x_2) = \begin{cases} -1, & \text{for all } x_2 < -0.0459, \\ 0, & \text{for all } x_2 \geq -0.0459, \end{cases}$$

and the sum of them is equal to

$$\sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T = (\theta_1, \theta_2)}}^{2} \; \sum_{\substack{r<s \\ r,s \in V(5)}} \left[ D^{\boldsymbol{\theta}}_{r,s}\psi_5(\boldsymbol{t}^5) \right]^2 d\boldsymbol{t}^5.$$

If $\phi_1 = \phi_2 = \phi_3 = \phi_4 = \phi_5 := \phi$, then the TR application put the PRSS function and so, our PRSS equation can be written as follows:

$$PRSS \approx \overbrace{\left\| \boldsymbol{y} - \boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2}^{=\boldsymbol{Accuracy}} + \overbrace{\phi \left\| L\boldsymbol{\alpha} \right\|_2^2}^{\approx \boldsymbol{Complexity}} .$$

Here, the first part of the TR and that of the PRSS function are equal to each other. However, the second parts of them are approximately equal. These parts are stated subsequently:

***Accuracy:*** $\left\| \boldsymbol{y} - \boldsymbol{\psi}(\breve{\boldsymbol{b}})\boldsymbol{\alpha} \right\|_2^2 = \left( \boldsymbol{y} - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\breve{\boldsymbol{b}}) \right)^T \left( \boldsymbol{y} - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\breve{\boldsymbol{b}}) \right) = \sum_{i=1}^{N} \left( y_i - \boldsymbol{\alpha}^T \boldsymbol{\psi}(\breve{\boldsymbol{b}}_i) \right)^2,$

***Complexity*****:** $\phi \left\| L\boldsymbol{\alpha} \right\|_2^2 \approx \sum_{m=1}^{2} \phi_m \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T = (\theta_1, \theta_2)}}^{2} \; \sum_{\substack{r<s \\ r,s \in V(m)}} \int_{Q^m} \alpha_m^2 \left[ D^{\boldsymbol{\theta}}_{r,s}\psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m.$

For illustration, the ***accuracy*** part is expanded:

$$
\begin{aligned}
\sum_{i=1}^{20}\left(y_i-\boldsymbol{\alpha}^T\boldsymbol{\psi}(\breve{\boldsymbol{b}}_i)\right)^2 = {} & (-1.48931-\alpha_0-(\max\{0.24644-(-0.0459)\})\alpha_1 \\
& -(\max\{0,-0.0459-0.24644\})\alpha_2 \\
& -(\max\{0,-0.49499-(-0.1396)\})\alpha_3 \\
& -(\max\{0,-0.1396-(-0.49499)\})\alpha_4 \\
& -(\max\{0,-1.67292-(-1.6729)\})\cdot(\max\{0,-0.0459-0.24644\})\alpha_5)^2 \\
& +(-0.56113-\alpha_0-(\max\{0,-0.03074-(-0.0459)\})\alpha_1 \\
& -(\max\{0,-0.0459-(-0.03074)\})\alpha_2 \\
& -(\max\{0,-0.03808-(-0.1396)\})\alpha_3 \\
& -(\max\{0,-0.1396-(-0.03808)\})\alpha_4 \\
& -(\max\{0,0.70001-(-1.6729)\})\cdot(\max\{0,-0.0459-(-0.03074)\})\alpha_5)^2 \\
& \vdots \\
& +(-1.12647-\alpha_0-(\max\{0,-0.56998-(-0.0459)\})\alpha_1 \\
& -(\max\{0,-0.0459-(-0.56998)\})\alpha_2 \\
& -(\max\{0,0.31730-(-0.1396)\})\alpha_3 \\
& -(\max\{0,-0.1396-0.31730\})\alpha_4 \\
& -(\max\{0,0.13051-(-1.6729)\})\cdot(\max\{0,-0.0459-(-0.56998)\})\alpha_5)^2.
\end{aligned}
$$

When computing the maximum function, the ***Accuracy*** term (*RSS*) becomes:

$$
\begin{aligned}
\sum_{i=1}^{20}\left(y_i-\boldsymbol{\alpha}^T\boldsymbol{\psi}(\breve{\boldsymbol{b}}_i)\right)^2 = {} & \left(-1.48931-\alpha_0-0.29234\alpha_1-0.35539\alpha_4\right)^2 \\
& +\left(-0.56113-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& \vdots \\
& +\left(1.12647-\alpha_0-0.52408\alpha_2-0.45690\alpha_3-0.94514\alpha_5\right)^2.
\end{aligned}
$$

The whole form of RSS can be seen in Appendix A.

The multi-dimensional integral in the ***complexity*** part of the equation (3.2.7) turn into the form of the equation (3.2.8) after discretization, and finally, the discretized form is indicated by ***L*** which is given in equation (3.2.11). To apply this discretization, first we sort the data set used for our numerical example, and then, we slightly decrease the first value of each predictor variable and slightly increase the last value of each predictor variable (see Subsection 3.2.2 for more details). Therefore, after adding two new observations to each of the predictor variables, the number of observations of our data set increases from 20 to 22, and we obtain a new data set. For each predictor variables, the new observations are as follows:

For $x_1$: The first discretization value of $x_1$ becomes $\breve{x}_{1,1} = 1.6477$ and the last discretization value of $x_1$ becomes $\breve{x}_{22,1} = 2.4379$.

For $x_2$: The first discretization value of $x_2$ becomes $\breve{x}_{1,2} = 1.7627$ and the last discretization value of $x_2$ becomes $\breve{x}_{22,2} = 2.5496$.

For $x_3$: The first discretization value of $x_3$ becomes $\breve{x}_{1,3} = 1.7729$ and the last discretization value of $x_3$ becomes $\breve{x}_{22,3} = 1.8441$.

The values $L_m$ ($m$=1,2,...5) corresponding to BFs $\psi_1, \psi_2, ..., \psi_5$ are calculated as

$$L_1 = \left( \sum_{i=1}^{(21)^{\kappa_1}} \left[ \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(1)}} \left[ D^{\boldsymbol{\theta}}_{r,s} (\max\{0, x_3 + 0.0459\}) \right]^2 \right) \left( \breve{x}_{k^{\kappa_1^1}_{\mu^{\kappa_1}}+1,\kappa_1^1} - \breve{x}_{k^{\kappa_1^1}_{\mu^{\kappa_1}},\kappa_1^1} \right) \right] \right)^{1/2} = 1.5760,$$

$$L_2 = \left( \sum_{i=1}^{(21)^{\kappa_2}} \left[ \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(2)}} \left[ D^{\boldsymbol{\theta}}_{r,s} (\max\{0, -0.0459 - x_3\}) \right]^2 \right) \left( \breve{x}_{k^{\kappa_1^2}_{\mu^{\kappa_1}}+1,\kappa_1^2} - \breve{x}_{k^{\kappa_1^2}_{\mu^{\kappa_1}},\kappa_1^2} \right) \right] \right)^{1/2} = 1.2656,$$

$$L_3 = \left( \sum_{i=1}^{(21)^{\kappa_3}} \left[ \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(3)}} \left[ D^{\boldsymbol{\theta}}_{r,s} (\max\{0, x_2 + 0.1396\}) \right]^2 \right) \left( \breve{x}_{k^{\kappa_1^3}_{\mu^{\kappa_1}}+1,\kappa_1^3} - \breve{x}_{k^{\kappa_1^3}_{\mu^{\kappa_1}},\kappa_1^3} \right) \right] \right)^{1/2} = 1.6086,$$

$$L_4 = \left( \sum_{i=1}^{(21)^{\kappa_4}} \left[ \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(4)}} \left[ D^{\boldsymbol{\theta}}_{r,s} (\max\{0, -0.1396 - x_2)\}) \right]^2 \right) \left( \breve{x}_{k^{\kappa_1^4}_{\mu^{\kappa_1}}+1,\kappa_1^4} - \breve{x}_{k^{\kappa_1^4}_{\mu^{\kappa_1}},\kappa_1^4} \right) \right] \right)^{1/2} = 1.3132,$$

$$L_5 = \left( \sum_{i=1}^{(21)^{\kappa_5}} \left[ \left( \sum_{\substack{|\boldsymbol{\theta}|=1 \\ \boldsymbol{\theta}^T=(\theta_1,\theta_2)}}^{2} \sum_{\substack{r<s \\ r,s\in V(5)}} \left[ D^{\boldsymbol{\theta}}_{r,s} \psi_{_5} (\boldsymbol{t}^5) \right]^2 \right) \left( \breve{x}_{k^{\kappa_1^5}_{\mu^{\kappa_1}}+1,\kappa_1^5} - \breve{x}_{k^{\kappa_1^5}_{\mu^{\kappa_1}},\kappa_1^5} \right) \right] \right)^{1/2} = 0.8166,$$

where, $\psi_{_5}(\boldsymbol{t}^5) = \max\{0, x_1 + 1.6729\} \cdot \max\{0, -0.0459 - x_2\}$.

As a result, the $\boldsymbol{L}$ matrix becomes a $(6\times 6)$ - diagonal matrix and the first column elements of $\boldsymbol{L}$ are all zero. The diagonal elements of this matrix are $L_m$ $(m{=}1{,}2{,}...{,}5)$ as given below:

$$\boldsymbol{L}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5760 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.2656 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.6086 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.3132 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8166 \end{bmatrix}.$$

From the equation (4.1.18),

$$\boldsymbol{L\theta}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5760 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.2656 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.6086 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.3132 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8166 \end{bmatrix}\cdot\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix}=\begin{bmatrix} 0 \\ 1.5760\cdot\theta_1 \\ 1.2656\cdot\theta_2 \\ 1.6086\cdot\theta_3 \\ 1.3132\cdot\theta_4 \\ 0.8166\cdot\theta_5 \end{bmatrix},$$

and $\left\|\boldsymbol{L\theta}\right\|_2^2$ is the squared norm of $\boldsymbol{L\theta}$ which is

$$\left\|\boldsymbol{L\theta}\right\|_2^2=(1.5760\cdot\theta_1)^2+(1.2656\cdot\theta_2)^2+(1.6086\cdot\theta_3)^2+(1.3132\cdot\theta_4)^2+(0.8166\cdot\theta_5)^2.$$

### 4.2.4 Calculating Uncertainty Values for Both Input and Output Data Under Polyhedral Uncertainty Sets

After we obtain ***accuracy*** and ***complexity*** terms, to employ the robust optimization technique on the CMARS model, we incorporate a perturbation (uncertainty) into the real input data in each dimension and into the output data. For this purpose, the uncertainty from (4.1.16) is evaluated for all input and output values which are represented by *CIs* and the uncertainty matrices and vectors based on *polyhedral uncertainty sets* are obtained by using (4.1.27) and (4.1.28). After we transform the variables into the standard normal distribution, the CI is constructed to be (-3, 3). The uncertainty values for each BF are represented below:

$$\left|\boldsymbol{u}_{mi}\right| \le \sum_{\substack{A\subseteq\{1,\dots,K_m\}\\ \neq}} \prod_{a\in A} \underbrace{\left|\breve{x}_{ia}-\tau_a\right|}_{\le D_{ia}\rho_{ia}} \prod_{b\in\{1,\dots,K_m\}/A} \underbrace{\left(\left|\pm \mathrm{A}_{i\kappa_m^j} + \Delta_{i\kappa_m^j}\right|\right)}_{\le \gamma_{ib}+\rho_{ib}} \le \sum_{\substack{A\subseteq\{1,\dots,K_m\}\\ \neq}} \prod_{a\in A} \mathrm{B}_{ia}\rho_{ia} \prod_{b\in\{1,\dots,K_m\}/A} (\gamma_{ib}+\rho_{ib})$$

$$\le \sum_{\substack{A\subseteq\{1,\dots,K_m\}\\ \neq}} \prod_{a\in A} \underbrace{\mathrm{B}_{ia}}_{\le \mathrm{B}_i} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,\dots,K_m\}/A} (\gamma_{ib}+\rho_{ib}) \le \sum_{\substack{A\subseteq\{1,\dots,K_m\}\\ \neq}} \mathrm{B}_i^{|A|-1} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,\dots,K_m\}/A} (\gamma_{ib}+\rho_{ib}).$$

From worst case approach, for each observation, we use the following equation to obtain the uncertainty vectors $\boldsymbol{u}_{mi}$ $(m=1,2,\dots,5;\ i=1,2,\dots,20)$:

$$\left|\boldsymbol{u}_{mi}\right| = \left|\psi_m(\breve{\breve{\boldsymbol{x}}}_i) - \psi_m(\breve{\boldsymbol{x}}_i)\right| = \sum_{\substack{A\subseteq\{1,\dots,K_m\}\\ \neq}} \mathrm{B}_i^{|A|-1} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,\dots,K_m\}/A} (\gamma_{ib}+\rho_{ib}).$$

Therefore, for our example, the uncertainty values for $\psi_1$ are represented as:

$$\left|\boldsymbol{u}_{1i}\right| = \left|\psi_1(\breve{\breve{\boldsymbol{x}}}_i)_i - \psi_1(\breve{\boldsymbol{x}}_i)_i\right| = \sum_{\substack{A\subseteq\{1,\dots,K_1\}\\ \neq}} \mathrm{B}_i^{|A|-1} \prod_{a\in A}\rho_{ia} \prod_{b\in\{1,\dots,K_1\}/A} (\gamma_{ib}+\rho_{ib}),$$

where

$$i=1:\ \ u_{11} = \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_1^{|A|-1} \prod_{a\in A}\rho_{1a} \prod_{b\in\{1\}/A} (\gamma_{1b}+\rho_{1b}) = \sum_{\{\}} \underbrace{\mathrm{B}_1^{1-1} \prod_{a\in\{\}}\rho_{1a}}_{1} \underbrace{\prod_{b\in\{1\}} (\gamma_{1b}+\rho_{1b})}_{3.2623} = 3.2623,$$

$$i=2:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_2^{|A|-1} \prod_{a\in A}\rho_{2a} \prod_{b\in\{1\}/A} (\gamma_{2b}+\rho_{2b}) = 3.4009,$$

$$i=3:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_3^{|A|-1} \prod_{a\in A}\rho_{3a} \prod_{b\in\{1\}/A} (\gamma_{3b}+\rho_{3b}) = 2.9977,$$

$$i=4:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_4^{|A|-1} \prod_{a\in A}\rho_{4a} \prod_{b\in\{1\}/A} (\gamma_{4b}+\rho_{4b}) = 2.9322,$$

$$i=5:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_5^{|A|-1} \prod_{a\in A}\rho_{5a} \prod_{b\in\{1\}/A} (\gamma_{5b}+\rho_{5b}) = 3.1590,$$

$$i=6:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_6^{|A|-1} \prod_{a\in A}\rho_{6a} \prod_{b\in\{1\}/A} (\gamma_{6b}+\rho_{6b}) = 2.4686,$$

$$i=7:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_7^{|A|-1} \prod_{a\in A}\rho_{7a} \prod_{b\in\{1\}/A} (\gamma_{7b}+\rho_{7b}) = 2.2166,$$

$$i=8:\ \ \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_8^{|A|-1} \prod_{a\in A}\rho_{8a} \prod_{b\in\{1\}/A} (\gamma_{8b}+\rho_{8b}) = 2.9751,$$

$$i=10:\quad \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_{10}^{|A|-1}\prod_{a\in A}\rho_{10a}\prod_{b\in\{1\}/A}(\gamma_{10b}+\rho_{10b})=3.4085,$$

$$i=13:\quad \sum_{\substack{A\subseteq\{1\}\\ \neq}} \mathrm{B}_{13}^{|A|-1}\prod_{a\in A}\rho_{13a}\prod_{b\in\{1\}/A}(\gamma_{13b}+\rho_{13b})=3.1792.$$

Here, the value of $\sum_{\substack{A\subseteq\{1,\ldots,K_1\}\\ \neq}} \mathrm{B}_{i}^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_1\}/A}(\gamma_{ib}+\rho_{ib})$ vanishes for all other *i* values.

For $\psi_2$, the uncertainty values are represented as:

$$|u_{2i}|=\left|\psi_2(\breve{\breve{\boldsymbol{x}}}_i)-\psi_2(\breve{\boldsymbol{x}}_i)\right|=\sum_{\substack{A\subseteq\{1,\ldots,K_2\}\\ \neq}} \mathrm{B}_{i}^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_2\}/A}(\gamma_{ib}+\rho_{ib}),$$

where

$$i=9:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{9}^{|A|-1}\prod_{a\in A}\rho_{9a}\prod_{b\in\{2\}/A}(\gamma_{9b}+\rho_{9b})=\sum_{\{\}}\mathrm{B}_{9}^{1-1}\underbrace{\prod_{a\in\{\}}\rho_{9a}}_{1}\underbrace{\prod_{b\in\{2\}}(\gamma_{9b}+\rho_{9b})}_{3.2492}=3.2492,$$

$$i=11:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{11}^{|A|-1}\prod_{a\in A}\rho_{11a}\prod_{b\in\{2\}/A}(\gamma_{11b}+\rho_{11b})=3.1913,$$

$$i=12:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{12}^{|A|-1}\prod_{a\in A}\rho_{12a}\prod_{b\in\{2\}/A}(\gamma_{12b}+\rho_{12b})=3.1509,$$

$$i=14:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{14}^{|A|-1}\prod_{a\in A}\rho_{14a}\prod_{b\in\{2\}/A}(\gamma_{14b}+\rho_{14b})=2.6117,$$

$$i=15:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{15}^{|A|-1}\prod_{a\in A}\rho_{15a}\prod_{b\in\{2\}/A}(\gamma_{15b}+\rho_{15b})=2.8637,$$

$$i=16:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{16}^{|A|-1}\prod_{a\in A}\rho_{16a}\prod_{b\in\{2\}/A}(\gamma_{16b}+\rho_{16b})=3.0955,$$

$$i=17:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{17}^{|A|-1}\prod_{a\in A}\rho_{17a}\prod_{b\in\{2\}/A}(\gamma_{17b}+\rho_{17b})=2.6923,$$

$$i=18:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{18}^{|A|-1}\prod_{a\in A}\rho_{18a}\prod_{b\in\{2\}/A}(\gamma_{18b}+\rho_{18b})=2.9317,$$

$$i=19:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{19}^{|A|-1}\prod_{a\in A}\rho_{19a}\prod_{b\in\{2\}/A}(\gamma_{19b}+\rho_{19b})=3.1131,$$

$$i=20:\quad \sum_{\substack{A\subseteq\{2\}\\ \neq}} \mathrm{B}_{20}^{|A|-1}\prod_{a\in A}\rho_{20a}\prod_{b\in\{2\}/A}(\gamma_{20b}+\rho_{20b})=3.1005.$$

Here, the value of $\sum_{\substack{A\subseteq\{1,\ldots,K_2\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_2\}/A}(\gamma_{ib}+\rho_{ib})$ is zero for all other *i* values.

For $\psi_3$, the uncertainty values are represented as:

$$\left|u_{3i}\right| = \left|\psi_3(\breve{\breve{\boldsymbol{x}}}_i)-\psi_3(\breve{\boldsymbol{x}}_i)\right| = \sum_{\substack{A\subseteq\{1,\ldots,K_3\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_3\}/A}(\gamma_{ib}+\rho_{ib}),$$

where,

$$i=2:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_2^{|A|-1}\prod_{a\in A}\rho_{2a}\prod_{b\in\{3\}/A}(\gamma_{2b}+\rho_{2b}) = \sum_{\{\}}\underbrace{\mathrm{B}_2^{1-1}\prod_{a\in\{\}}\rho_{2a}}_{1}\ \underbrace{\prod_{b\in\{3\}}(\gamma_{2b}+\rho_{2b})}_{3.4513} = 3.4513,$$

$$i=5:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_5^{|A|-1}\prod_{a\in A}\rho_{5a}\prod_{b\in\{3\}/A}(\gamma_{5b}+\rho_{5b}) = 2.9182,$$

$$i=7:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_7^{|A|-1}\prod_{a\in A}\rho_{7a}\prod_{b\in\{3\}/A}(\gamma_{7b}+\rho_{7b}) = 2.8421,$$

$$i=8:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_8^{|A|-1}\prod_{a\in A}\rho_{8a}\prod_{b\in\{3\}/A}(\gamma_{8b}+\rho_{8b}) = 3.2990,$$

$$i=12:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_{12}^{|A|-1}\prod_{a\in A}\rho_{12a}\prod_{b\in\{3\}/A}(\gamma_{12b}+\rho_{12b}) = 2.2075,$$

$$i=13:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_{13}^{|A|-1}\prod_{a\in A}\rho_{13a}\prod_{b\in\{3\}/A}(\gamma_{13b}+\rho_{13b}) = 3.3751,$$

$$i=15:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_{15}^{|A|-1}\prod_{a\in A}\rho_{15a}\prod_{b\in\{3\}/A}(\gamma_{15b}+\rho_{15b}) = 2.8167,$$

$$i=16:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_{16}^{|A|-1}\prod_{a\in A}\rho_{16a}\prod_{b\in\{3\}/A}(\gamma_{16b}+\rho_{16b}) = 2.8167,$$

$$i=20:\ \sum_{\substack{A\subseteq\{3\}\\ \neq}} \mathrm{B}_{20}^{|A|-1}\prod_{a\in A}\rho_{20a}\prod_{b\in\{3\}/A}(\gamma_{20b}+\rho_{20b}) = 3.2736.$$

Here, the value of $\sum_{\substack{A\subseteq\{1,\ldots,K_3\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_3\}/A}(\gamma_{ib}+\rho_{ib})$ vanishes for all other *i* values.

For $\psi_4$, the uncertainty values are represented as:

$$\left|u_{4i}\right| = \left|\psi_4(\breve{\breve{\boldsymbol{x}}}_i)-\psi_4(\breve{\boldsymbol{x}}_i)\right| = \sum_{\substack{A\subseteq\{1,\ldots,K_4\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\ldots,K_4\}/A}(\gamma_{ib}+\rho_{ib}),$$

where

$$i=1:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_1^{|A|-1}\prod_{a\in A}\rho_{1a}\prod_{b\in\{4\}/A}(\gamma_{1b}+\rho_{1b})=\sum_{\{\}}\mathrm{B}_1^{1-1}\underbrace{\prod_{a\in\{\}}\rho_{1a}}_{1}\underbrace{\prod_{b\in\{4\}}(\gamma_{1b}+\rho_{1b})}_{3.1062}=3.1062,$$

$$i=3:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_3^{|A|-1}\prod_{a\in A}\rho_{3a}\prod_{b\in\{4\}/A}(\gamma_{3b}+\rho_{3b})=3.2585,$$

$$i=4:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_4^{|A|-1}\prod_{a\in A}\rho_{4a}\prod_{b\in\{4\}/A}(\gamma_{4b}+\rho_{4b})=3.2838,$$

$$i=6:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_6^{|A|-1}\prod_{a\in A}\rho_{6a}\prod_{b\in\{4\}/A}(\gamma_{6b}+\rho_{6b})=2.8523,$$

$$i=9:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_9^{|A|-1}\prod_{a\in A}\rho_{9a}\prod_{b\in\{4\}/A}(\gamma_{9b}+\rho_{9b})=2.9285,$$

$$i=10:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{10}^{|A|-1}\prod_{a\in A}\rho_{10a}\prod_{b\in\{4\}/A}(\gamma_{10b}+\rho_{10b})=3.1315,$$

$$i=11:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{11}^{|A|-1}\prod_{a\in A}\rho_{11a}\prod_{b\in\{4\}/A}(\gamma_{11b}+\rho_{11b})=3.0808,$$

$$i=14:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{14}^{|A|-1}\prod_{a\in A}\rho_{14a}\prod_{b\in\{4\}/A}(\gamma_{14b}+\rho_{14b})=3.0300,$$

$$i=17:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{17}^{|A|-1}\prod_{a\in A}\rho_{17a}\prod_{b\in\{4\}/A}(\gamma_{17b}+\rho_{17b})=2.9031,$$

$$i=18:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{18}^{|A|-1}\prod_{a\in A}\rho_{18a}\prod_{b\in\{4\}/A}(\gamma_{18b}+\rho_{18b})=2.9031,$$

$$i=19:\quad \sum_{\substack{A\subseteq\{4\}\\ \neq}} \mathrm{B}_{19}^{|A|-1}\prod_{a\in A}\rho_{19a}\prod_{b\in\{4\}/A}(\gamma_{19b}+\rho_{19b})=2.5223.$$

Here, the value of $\sum_{\substack{A\subseteq\{1,\dots,K_4\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\dots,K_4\}/A}(\gamma_{ib}+\rho_{ib})$ is zero for all other $i$ values.

For $\psi_5$, the uncertainty values are represented as:

$$|u_5|=\left|\psi_5(\breve{\breve{\mathbf{x}}}_i)-\psi_5(\breve{\mathbf{x}}_i)\right|=\sum_{\substack{A\subseteq\{1,\dots,K_5\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\dots,K_5\}/A}(\gamma_{ib}+\rho_{ib}),$$

where

$$i=9: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_9^{|A|-1}\prod_{a\in A}\rho_{9a}\prod_{b\in\{2,5\}/A}(\gamma_{9b}+\rho_{9b}) = \sum \mathrm{B}_9^{1-1}\prod_{a\in A}\rho_{9a}\prod_{b\in\{2,5\}/A}(\gamma_{9b}+\rho_{9b}) = 54.5087,$$

$$i=11: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{11}^{|A|-1}\prod_{a\in A}\rho_{11a}\prod_{b\in\{2,5\}/A}(\gamma_{11b}+\rho_{11b}) = 47.3986,$$

$$i=12: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{12}^{|A|-1}\prod_{a\in A}\rho_{12a}\prod_{b\in\{2,5\}/A}(\gamma_{12b}+\rho_{12b}) = 45.7299,$$

$$i=14: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{14}^{|A|-1}\prod_{a\in A}\rho_{14a}\prod_{b\in\{2,5\}/A}(\gamma_{14b}+\rho_{14b}) = 36.4180,$$

$$i=15: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{15}^{|A|-1}\prod_{a\in A}\rho_{15a}\prod_{b\in\{2,5\}/A}(\gamma_{15b}+\rho_{15b}) = 36.4337,$$

$$i=16: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{16}^{|A|-1}\prod_{a\in A}\rho_{16a}\prod_{b\in\{2,5\}/A}(\gamma_{16b}+\rho_{16b}) = 41.1340,$$

$$i=17: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{17}^{|A|-1}\prod_{a\in A}\rho_{17a}\prod_{b\in\{2,5\}/A}(\gamma_{17b}+\rho_{17b}) = 45.7592,$$

$$i=18: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{18}^{|A|-1}\prod_{a\in A}\rho_{18a}\prod_{b\in\{2,5\}/A}(\gamma_{18b}+\rho_{18b}) = 49.0705,$$

$$i=19: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{19}^{|A|-1}\prod_{a\in A}\rho_{19a}\prod_{b\in\{2,5\}/A}(\gamma_{19b}+\rho_{19b}) = 47.7710,$$

$$i=20: \quad \sum_{\substack{A\subseteq\{2,5\}\\ \neq}} \mathrm{B}_{20}^{|A|-1}\prod_{a\in A}\rho_{20a}\prod_{b\in\{2,5\}/A}(\gamma_{20b}+\rho_{20b}) = 45.7116.$$

Here, the value of $\sum_{\substack{A\subseteq\{1,\dots,K_5\}\\ \neq}} \mathrm{B}_i^{|A|-1}\prod_{a\in A}\rho_{ia}\prod_{b\in\{1,\dots,K_5\}/A}(\gamma_{ib}+\rho_{ib})$ vanishes for other *i* values.

Then, we can write our uncertainty matrix for input data as follows:

$$\boldsymbol{U} = \begin{bmatrix} |u_{11}| & |u_{12}| & \cdots & |u_{15}| \\ |u_{21}| & |u_{22}| & \cdots & |u_{25}| \\ \vdots & \vdots & \ddots & \vdots \\ |u_{201}| & |u_{202}| & \cdots & |u_{205}| \end{bmatrix} = \begin{bmatrix} [3.5246,-3.5246] & 0 & \cdots & 0 \\ [3.8018,-3.8018] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & [3.2011,-3.2011] & \cdots & [46.4190,-46.4190] \end{bmatrix}.$$

After we incorporate an uncertainty for each input value, our basis function’s matrices can be represented as the following forms:

$$\boldsymbol{W}_{upper} = \boldsymbol{\psi}\left(\breve{\boldsymbol{b}}\right) + \boldsymbol{U}_{upper} = \begin{bmatrix} 1 & 3.8170 & \cdots & 0 \\ 1 & 3.8170 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & \vdots & \cdots & 47.3642 \end{bmatrix},$$

$$\boldsymbol{W}_{lower} = \boldsymbol{\psi}\left(\breve{\boldsymbol{b}}\right) + \boldsymbol{U}_{lower} = \begin{bmatrix} 1 & -3.2323 & \cdots & 0 \\ 1 & -3.7866 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & -45.4739 \end{bmatrix}.$$

For output data, the uncertainty vector and the vectors with uncertainty are represented below:

$$\boldsymbol{v} = \begin{bmatrix} |v_1| \\ |v_2| \\ \vdots \\ |v_{20}| \end{bmatrix} = \begin{bmatrix} [3,-3] \\ [3,-3] \\ \vdots \\ [3,-3] \end{bmatrix}, \quad \boldsymbol{z}_{upper} = \breve{\boldsymbol{y}} + \boldsymbol{v}_{upper} = \begin{bmatrix} 1.5107 \\ 2.4389 \\ \vdots \\ 4.1265 \end{bmatrix}, \quad \boldsymbol{z}_{lower} = \breve{\boldsymbol{y}} + \boldsymbol{v}_{lower} = \begin{bmatrix} -4.4893 \\ -3.5611 \\ \vdots \\ -1.8735 \end{bmatrix}.$$

Whole form of $\boldsymbol{U}, \boldsymbol{v}, \boldsymbol{W}_{upper}, \boldsymbol{W}_{lower}, \boldsymbol{z}_{upper}$ and $\boldsymbol{z}_{lower}$ can be seen in Appendix B.

### 4.2.5 Obtaining Weak RCMARS Models Using Combinatorial Approach

As we mentioned in the previous section, PRSS is the TR problem and we can easily formulate PRSS as a CQP problem. Moreover, using our robust optimization approach for a robustification of CMARS, we incorporate a perturbation (uncertainty) into the real input data, $\breve{\boldsymbol{x}}_i \ (i = 1, 2, ..., N),$ in each dimension and into the output data, $\boldsymbol{y}.$ For this aim, the uncertainty matrices and vectors based on polyhedral uncertainty sets are obtained by applying (4.1.27) and (4.1.28). Then, using the equation (4.1.16), uncertainty is evaluated for all input and output values which are represented by CIs. The boundaries of CIs are assumed to be (-3, 3) after the variables are transformed into the standard normal distribution.

For our example, the uncertainty matrix for input data has a huge size, and we do not have enough computer capacity to solve our problem for this uncertainty matrix. Indeed, we have a *tradeoff* between tractability and robustification. To overcome this difficulty, in this example, we formulate PRSS as a CQP problem for each sample value (observation) using the *combinatorial* approach, which we call *weak robustification*. As a result, we obtain 20 different *weak RCMARS* (*WRCMARS*) models, and solve them by using MOSEK program [58].

Based on polyhedral uncertainty sets, to solve our problem, we use the vertices. In order to find these vertices, we need to apply Cartesian product for each interval of input data in the observations (refer to Subsection 4.1.6 for more detail). Hence, our WRCMARS models have different structures depending on the number of entries (BFs) which the observations have. In our example, we have two or three entries in our sample values. For instance, in our 20 different models, we present two models which have two entries and three entries respectively.

We can show the first observation's WRCMARS model, which has two entries, in the following form:

$$
\begin{aligned}
&\underset{t,\boldsymbol{\alpha}}{\text{minimize}}\ t,\\
&\text{subject to}
\end{aligned}
$$

$$
\begin{array}{ll}
1.51069\ -\alpha_0 - 3.81696\alpha_1 - 3.56769\alpha_4 & =\beta_1,\\
2.43887\ -\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & =\beta_2,\\
\vdots & \vdots\\
4.12647\ -\alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 & =\beta_{20},\\
\\
1.51069\ -\alpha_0 - 3.81696\alpha_1 - 3.56769\alpha_4 & =\beta_{21},\\
2.43887\ -\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & =\beta_{22},\\
\vdots & \vdots\\
4.12647\ -\alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 & =\beta_{40},\\
\\
1.51069\ -\alpha_0 + 3.23229\alpha_1 - 3.56769\alpha_4 & =\beta_{41},\\
2.43887\ -\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & =\beta_{42},\\
\vdots & \vdots\\
4.12647\ -\alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 & =\beta_{60},\\
\\
1.51069\ -\alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 & =\beta_{61},\\
2.43887\ -\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & =\beta_{62},\\
\vdots & \vdots\\
4.12647\ -\alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 & =\beta_{80},
\end{array}
$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 3.81696\alpha_1 - 3.56769\alpha_4 &= \beta_{81},\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{82},\\
\vdots \qquad\qquad\qquad\qquad & \vdots\\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{100},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 3.81696\alpha_1 + 2.85691\alpha_4 &= \beta_{101},\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{102},\\
\vdots \qquad\qquad\qquad\qquad & \vdots\\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{120},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 + 3.23229\alpha_1 - 3.56769\alpha_4 &= \beta_{121},\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{122},\\
\vdots \qquad\qquad\qquad\qquad & \vdots\\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{140},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 &= \beta_{141},\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{142},\\
\vdots \qquad\qquad\qquad\qquad & \vdots\\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{160},
\end{aligned}$$

$$\begin{aligned}
1.54717\alpha_1 &= \beta_{161},\\
\vdots \qquad & \vdots\\
0.81663\alpha_5 &= \beta_{165},
\end{aligned}$$

$$\begin{aligned}
(\beta_1^2 + \beta_2^2 + \ldots + \beta_{20}^2)^{1/2} &\le t,\\
(\beta_{21}^2 + \beta_{22}^2 + \ldots + \beta_{40}^2)^{1/2} &\le t,\\
\vdots \qquad\qquad &\\
(\beta_{121}^2 + \beta_{122}^2 + \ldots + \beta_{140}^2)^{1/2} &\le t,\\
(\beta_{141}^2 + \beta_{142}^2 + \ldots + \beta_{160}^2)^{1/2} &\le t,
\end{aligned}$$

$$(\beta_{161} + \beta_{162} + \beta_{163} + \beta_{164} + \beta_{165})^{1/2} \le (\breve{M})^{1/2}.$$

For solving this problem, we transform it into the MOSEK format above. The whole form of MOSEK model can be seen in Appendix C. For this transformation, we attribute new unknown variables in the linear notations in these nine cones. By this way, we simplify the notations in the cones and write them as constraints. Therefore, for our first sample, our problem includes one hundred seventy three linear constraints and nine quadratic cones. To try to obtain MOSEK model for first sample value in the interval uncertainty case, only the first values $\beta_1, \beta_{21}, \beta_{41}, \beta_{61}, \beta_{81}, \beta_{101}, \beta_{121}$ and $\beta_{141}$ in the cones are different for the first eight cones. The other values remain the same.

Similarly, we can represent the last observation's WRCMARS model, which has three entries, in following form:

$$\begin{aligned}
&\underset{t,\boldsymbol{\alpha}}{\text{minimize}}\ t,\\
&\text{subject to}
\end{aligned}$$

$$\begin{aligned}
&1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 && = \beta_1,\\
&2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 && = \beta_2,\\
&\vdots && \vdots\\
&4.12647\ -\alpha_0 -\ 3.72516\alpha_2 -\ 4.00407\alpha_3 -\ 47.36416\alpha_5 && = \beta_{20},
\end{aligned}$$

$$\begin{aligned}
&1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 && = \beta_{21},\\
&2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 && = \beta_{22},\\
&\vdots && \vdots\\
&4.12647\ -\alpha_0 -\ 3.72516\alpha_2 -\ 4.00407\alpha_3 + 45.47389\alpha_5 && = \beta_{40},
\end{aligned}$$

$$\begin{aligned}
&1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 && = \beta_{41},\\
&2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 && = \beta_{42},\\
&\vdots && \vdots\\
&4.12647\ -\alpha_0 -\ 3.72516\alpha_2 + 3.09027\alpha_3 -\ 47.36416\alpha_5 && = \beta_{60},
\end{aligned}$$

$$\begin{aligned}
&1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 && = \beta_{61},\\
&2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 && = \beta_{62},\\
&\vdots && \vdots\\
&4.12647\ -\alpha_0 -\ 3.72516\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 && = \beta_{80},
\end{aligned}$$

$$\begin{aligned}
&1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 && = \beta_{81},\\
&2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 && = \beta_{82},\\
&\vdots && \vdots\\
&4.12647\ -\alpha_0 + 2.67700\alpha_2 -\ 4.00407\alpha_3 -\ 47.36416\alpha_5 && = \beta_{100},
\end{aligned}$$

$$
\begin{aligned}
&1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{101},\\
&2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{102},\\
&\vdots && \vdots\\
&4.12647 - \alpha_0 + 2.67700\alpha_2 - 4.00407\alpha_3 + 45.47389\alpha_5 && = \beta_{120},
\end{aligned}
$$

$$
\begin{aligned}
&1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{121},\\
&2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{122},\\
&\vdots && \vdots\\
&4.12647 - \alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 && = \beta_{140},
\end{aligned}
$$

$$
\begin{aligned}
&1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{141},\\
&2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{142},\\
&\vdots && \vdots\\
&4.12647 - \alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 && = \beta_{160},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{161},\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{162},\\
&\vdots && \vdots\\
&-1.87353 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 - 47.36416\alpha_{55} && = \beta_{180},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{181},\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{182},\\
&\vdots && \vdots\\
&-1.87353 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 + 45.47389\alpha_5 && = \beta_{200},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{201},\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{202},\\
&\vdots && \vdots\\
&-1.87353 - \alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 && = \beta_{220},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 && = \beta_{221},\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 && = \beta_{222},\\
&\vdots && \vdots\\
&-1.87353 - \alpha_0 + 2.67700\alpha_2 - 4.00407\alpha_3 - 47.36416\alpha_5 && = \beta_{240},
\end{aligned}
$$

$$\begin{aligned}
-4.48931 &-\alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{241},\\
-3.56113 &-\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{242},\\
&\vdots &&\vdots\\
-1.87353 &-\alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 &&= \beta_{260},
\end{aligned}$$

$$\begin{aligned}
-4.48931 &-\alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{261},\\
-3.56113 &-\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{262},\\
&\vdots &&\vdots\\
-1.87353 &-\alpha_0 + 2.67700\alpha_2 - 4.00407\alpha_3 + 45.47389\alpha_5 &&= \beta_{280},
\end{aligned}$$

$$\begin{aligned}
-4.48931 &-\alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{281},\\
-3.56113 &-\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{282},\\
&\vdots &&\vdots\\
-1.87353 &-\alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 &&= \beta_{300},
\end{aligned}$$

$$\begin{aligned}
-4.48931 &-\alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{301},\\
-3.56113 &-\alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{302},\\
&\vdots &&\vdots\\
-1.87353 &-\alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 &&= \beta_{320},
\end{aligned}$$

$$\begin{aligned}
1.54717\alpha_1 &= \beta_{321},\\
\vdots \quad &\quad \vdots\\
0.81663\alpha_5 &= \beta_{325},
\end{aligned}$$

$$\begin{aligned}
(\beta_1^2 + \beta_2^2 + \ldots + \beta_{20}^2)^{1/2} &\le t,\\
(\beta_{21}^2 + \beta_{22}^2 + \ldots + \beta_{40}^2)^{1/2} &\le t,\\
\vdots\quad&\\
(\beta_{281}^2 + \beta_{282}^2 + \ldots + \beta_{300}^2)^{1/2} &\le t,\\
(\beta_{301}^2 + \beta_{302}^2 + \ldots + \beta_{320}^2)^{1/2} &\le t,
\end{aligned}$$

$$(\beta_{321} + \beta_{322} + \beta_{323} + \beta_{324} + \beta_{325})^{1/2} \le (\breve{M})^{1/2}.$$

For solving this problem, we transform it into the MOSEK format above. The whole form of MOSEK model can be seen in Appendix C. For this transformation, we attribute new unknown variables in the linear notations in these nine cones. By this way, we simplify the notations in the cones and write them as constraints. Consequently, for our last sample, our problem includes three hundred forty one linear constraints and seventeen quadratic cones. To try to obtain

MOSEK model for last sample value in the interval uncertainty case, only the last values $\beta_{20}, \beta_{40}, \beta_{60}, \beta_{80}, \beta_{100}, \beta_{120}, \beta_{140}, \beta_{160}, \beta_{180}, \beta_{200}, \beta_{220}, \beta_{240}, \beta_{260}, \beta_{280}, \beta_{300}$ and $\beta_{320}$ in the cones are different for the first sixteen cones. The other values remain the same.

We write this formulation for each value of our sample ($N$=20) and solve them separately by using MOSEK program [58]. After we obtain MOSEK results and find the $t$ values for all auxiliary problems, using the *worst-case* approach, we select the solution which has the *maximum t* value. Then we continue with our calculations using the parameter values $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ and $\alpha_5$ that we find from the auxiliary problem which has the highest $t$ value.

### 4.2.6 Sensitivity to The Changes in the Confidence Interval Limits of RCMARS

In order to represent sensitivity to the changes in the CI limits of the input data and output data and find suitable interval limit for us, we obtain different uncertainty matrices, ***U***, for the input data and different uncertainty vectors, ***v***, for the output data as the form of (4.1.22) by using four different intervals which are given by the pairs ±3, ±3E-6, ±3E-7, and as a special case, mid-point value of our interval (i.e., zero length interval), in which case it reduces to CMARS model. We calculate our parameters with 16 different uncertainty scenarios using these values under polyhedral uncertainty sets. All of the parameter estimates as well as model accuracies for different uncertainty scenarios are shown in Table 4.1 - 4.4. We note here that we defined the values $\sqrt{\breve{M}}$ by a model-free method. When we apply the $\sqrt{\breve{M}}$ values in our RCMARS code and solve by using MOSEK, we use the $\sqrt{\breve{M}}$ value which has the minimum value of PRSS in the equation (4.1.18).

The accuracy measures evaluated based on Average Absolute Error (AAE), Root Mean Squared Error (RMSE) and Correlation Coefficient (r). These performance measures and their general notations are as follows;

$N$: a number of observations,<br>
$p$: a number of terms in the model,<br>
$y_i$: an $i$th observed response value,<br>
$\hat{y}_i$: an $i$th fitted response,<br>
$y$ : the actual response variables,<br>
$\bar{y}$ : the mean of actual values,

$\hat{y}$ : the predicted response variables,

$\overline{\hat{y}}$ : a mean of the predicted response variables,

$s(y)^2$ : the standard deviations of actual response variable,

$s(\overline{y})^2$ : the standard deviations of predicted response variable.

**Comparison Measures:**

AAE measures the average magnitude of error. The smaller AAE, the better it is [86]. The formula of AAE in the following form:

$$AAE := \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|.$$

RMSE is a measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. The smaller RMSE, the better it is [86]. A model formula is

$$\text{RMSE:=}\sqrt{\frac{1}{N-p-1}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2},$$

Correlation coefficient is a measure of how well linear association between the predicted and the actual response values [86]. The formula is represented as

$$r := \frac{\dfrac{\sum_{i=1}^{n}(y-\overline{y})(\hat{y}-\overline{\hat{y}})}{(n-1)}}{\sqrt{s(y)^2 s(\overline{y})^2}} \text{ such that } -1 \le r \le +1.$$

Table 4.1 Parameter estimates and the model performances I

| ***v*** | **±3** | | | |
|---|---|---|---|---|
| ***U*** | **±3** | **±3e-6** | **±3e-7** | **zero** |
| **$\alpha_0$** | 0.1230 | -0.0634 | -0.0773 | -0.3732 |
| **$\alpha_1$** | -0.3131 | -0.0526 | -0.0577 | 0.0274 |
| **$\alpha_2$** | 0.0000 | 0.2596 | 0.3141 | 0.1136 |
| **$\alpha_3$** | 0.0109 | -0.0029 | -0.0044 | -0.0700 |
| **$\alpha_4$** | 0.0000 | -0.0206 | -0.0315 | -0.0657 |
| **$\alpha_5$** | 0.0000 | -0.0021 | -0.0016 | 0.5238 |
| **AAE** | 0.7822 | 0.7241 | 0.7109 | 0.4885 |
| **RMSE** | 1.1814 | 1.1063 | 1.0862 | 0.7888 |
| **r** | 0.2124 | 0.6516 | 0.6617 | 0.7648 |

Table 4.2 Parameter estimates and the model performances II

| *v* | ±3e-6 | | | |
|---|---|---|---|---|
| ***U*** | **±3** | **±3e-6** | **±3e-7** | **zero** |
| $\alpha_0$ | 0.1230 | -0.0654 | -0.0815 | -0.3733 |
| $\alpha_1$ | -0.3133 | -0.0528 | -0.0592 | 0.0274 |
| $\alpha_2$ | 0.0000 | 0.2592 | 0.3297 | 0.1136 |
| $\alpha_3$ | 0.0110 | -0.0033 | -0.0046 | -0.0700 |
| $\alpha_4$ | 0.0000 | -0.0179 | -0.0337 | -0.0656 |
| $\alpha_5$ | 0.0000 | 0.0001 | -0.0018 | 0.5238 |
| **AAE** | 0.7822 | 0.7232 | 0.7080 | 0.4885 |
| **RMSE** | 1.1814 | 1.1043 | 1.0809 | 0.7888 |
| **r** | 0.2124 | 0.6536 | 0.6631 | 0.7648 |

Table 4.3 Parameter estimates and the model performances III

| *v* | ±3e-7 | | | |
|---|---|---|---|---|
| ***U*** | **±3** | **±3e-6** | **±3e-7** | **zero** |
| $\alpha_0$ | 0.1230 | -0.0597 | -0.0838 | -0.3733 |
| $\alpha_1$ | -0.3133 | -0.0513 | -0.0600 | 0.0274 |
| $\alpha_2$ | 0.0000 | 0.2441 | 0.3375 | 0.1136 |
| $\alpha_3$ | 0.0110 | -0.0023 | -0.0045 | -0.0700 |
| $\alpha_4$ | 0.0000 | -0.0150 | -0.0347 | -0.0656 |
| $\alpha_5$ | 0.0000 | -0.0031 | -0.0017 | 0.5238 |
| **AAE** | 0.7822 | 0.7285 | 0.7065 | 0.4885 |
| **RMSE** | 1.1814 | 1.1130 | 1.0781 | 0.7888 |
| **r** | 0.2124 | 0.6443 | 0.6638 | 0.7648 |

Table 4.4 Parameter estimates and the model performances IV

| *v* | zero | | | |
|---|---|---|---|---|
| ***U*** | **±3** | **±3e-6** | **±3e-7** | **zero** |
| $\alpha_0$ | 0.1230 | -0.0017 | -0.0676 | -0.3733 |
| $\alpha_1$ | -0.3133 | -0.0021 | -0.0543 | 0.0274 |
| $\alpha_2$ | 0.0000 | 0.0074 | 0.2751 | 0.1136 |
| $\alpha_3$ | 0.0110 | 0.0000 | -0.0029 | -0.0700 |
| $\alpha_4$ | 0.0000 | -0.0001 | -0.0239 | -0.0656 |
| $\alpha_5$ | 0.0000 | -0.0001 | -0.0016 | 0.5238 |
| **AAE** | 0.7822 | 0.7842 | 0.7200 | 0.4885 |
| **RMSE** | 1.1814 | 1.2057 | 1.1001 | 0.7888 |
| **r** | 0.2124 | 0.6191 | 0.6553 | 0.7648 |

The above results indicate that solutions obtained are sensitive to the limits of CIs. We obtain better performance results when the lengths of CIs are narrow. Moreover, when we use the mid-point of our interval values for both input and output data, which is the certain data case, we obtain the same parameter estimates, and thus, the same model performances compared with CMARS. This reveals that CMARS is a special case of RCMARS. In addition, according to the results, the solutions are more sensitive to the changes in the CI limits of the input data than the output data.

MOSEK uses an interior-point optimizer as a default for the CQP problem. The interior-point optimizer is an implementation of the homogeneous and self-dual algorithm. The values $\sqrt{\breve{M}}$ in our example are defined by a model-free (train and error) method. When we access the $\sqrt{\breve{M}}$ values in our RCMARS code and solve by using MOSEK, RCMARS provides us several solutions, each of them based on the five BFs.

In addition, as we expected, CMARS produces more accurate models than RCMARS. This is mainly due to the randomness involved in the input-output variables. However, we also expect to see that the variation of the parameter estimates and hence the variation of accuracy measures will be much less than that of CMARS. In order to reveal this expectation, a simulation study is conducted and the results are presented in the following section.

# CHAPTER 4

# SIMULATION STUDY

In previous chapters, we represented and investigated scientifically MARS, CMARS and RCMARS in detail. In this chapter, we compare these methods using 20 different data sets created by Monte Carlo simulation based on variation of the parameter estimates.

For this simulation study, we assume that similar to RCMARS, MARS and its modified method CMARS have random and normally distributed input and output data which are derived from data set we use for our numerical example (see Section 4.5 for more details) although they are nonparametric and have input data that are assumed to contain fixed variables. So we created 20 different random data sets to apply simulation for these algorithms. On the other hand, for RCMARS, 20 different interval values are determined for our example in Section 4.5 and hence, under polyhedral uncertainty sets, 20 different uncertainty scenarios are obtained by using these values. Then, 20 different data sets to apply simulation for RCMARS algorithm are also constructed with these uncertainty scenarios.

For each data set, parameter values are calculated for MARS, CMARS and RCMARS. To find these parameter estimates, for MARS application, Salford Systems [54] is applied while a MATLAB code is written and MOSEK software is used to solve the CQP problem for CMARS and RCMARS. MARS construct the best model by applying forward and backward stepwise algorithm. In this case, we obtain different models by using Salford MARS. Then, with minimum GCV, the best model is chosen among the obtained models having different numbers of BFs and interaction term. Therefore, 20 best models are generated for MARS. To obtain CMARS models, we use the BFs of the large models of MARS which is produced by the forward step-wise algorithm with 20 different data sets when the best models are obtained for MARS and we select $\sqrt{\breve{M}}$ for our CMARS code. Conversely, we use the BFs of the large model of MARS which is obtained by the forward step-wise algorithm with our numerical example in Section 4.5 and choose $\sqrt{\breve{M}}$ for our RCMARS code.

We have determined the values $\sqrt{\breve{M}}$ by a model-free method and thus CMARS and RCMARS algorithm provides us many different models. However, in this simulation, when we apply the

$\sqrt{\breve{M}}$ values in our CMARS and RCMARS code and solve using MOSEK, we select the $\sqrt{\breve{M}}$ value which has the minimum value of PRSS in the equation (4.1.18). Moreover, in order to see variation of model performance with parameter estimates, estimation errors of simulation models are evaluated based on AAE, RMSE and r. All of the parameter estimates and model performance values are shown in Appendix D for these three algorithms.

As we mentioned before, MARS and CMARS models depend on the parameters. Small perturbations in data may give different model parameters and this can cause unstable solutions. In this simulation study, MARS obtained models which have different numbers of BFs and also it obtained models which have interaction or no interaction. Because of this, we receive different model parameters for some of our data sets. When we investigate the tables in Appendix D, we can easily see that solutions of the optimization problem may reveal a remarkable sensitivity to perturbations in the parameters of the problem.

In CMARS, the aim is to reduce the estimation error while keeping efficiency as high as possible. CMARS also robustifies in view of the response variable *y*, not in view of the input variable *x*. However, in RCMARS, our aim is to decrease the estimation variance by implying full robustification in CMARS even though the estimation errors of RCMARS is higher than that of CMARS when we incorporate perturbation (uncertainty) into the real input data and output data based on polyhedral uncertainty set. In portfolio optimization and risk management, reduction of the variance is often based the *diversification effect* [45]. Since it is not easy to illustrate the reduction of estimation variance in RCMARS statistically, we represent this reduction graphically. For this simulation study, the results for the variance of parameter estimates can be seen in Figure 4.6-4.8.

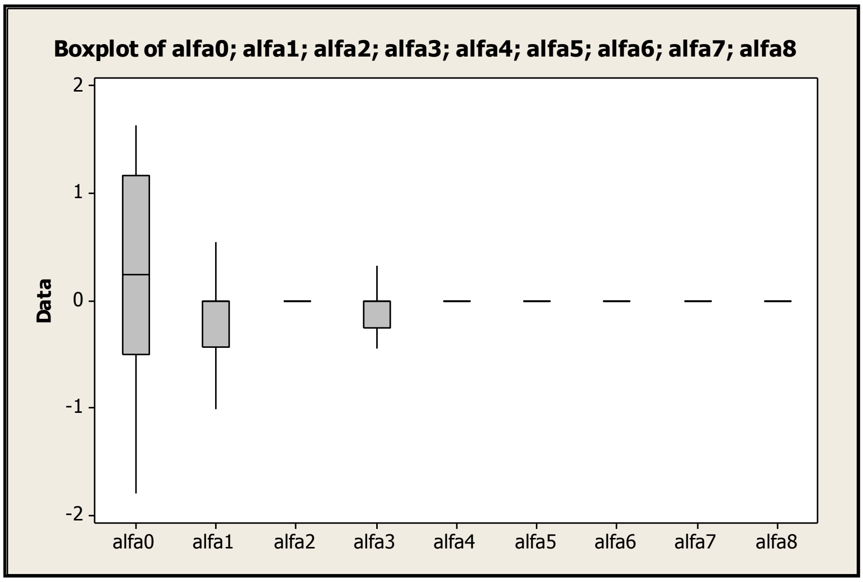


Figure 4.6 The graphical representation for the variance of parameter estimates of MARS

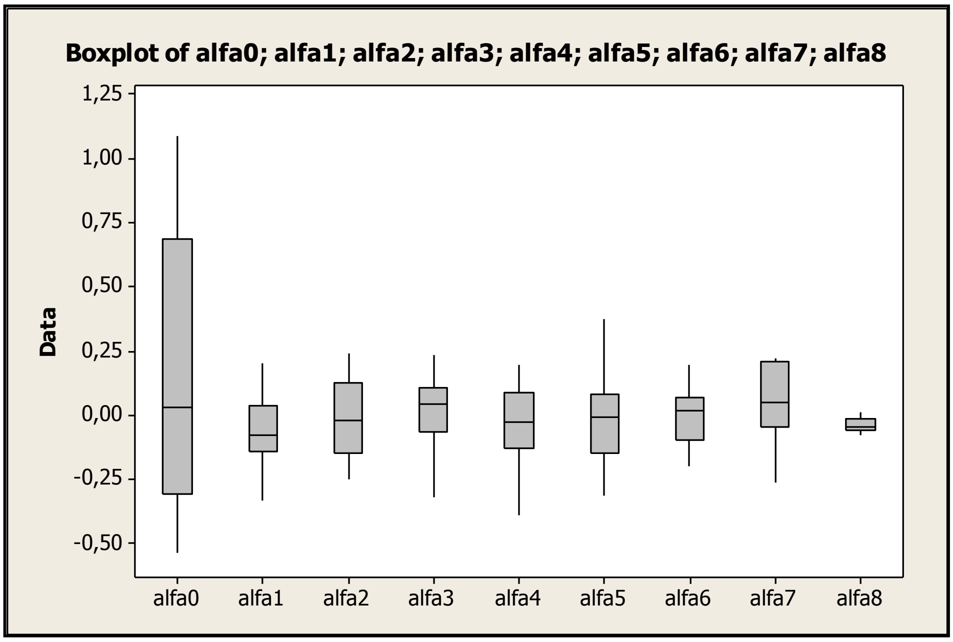


Figure 4.7 The graphical representation for the variance of parameter estimates of CMARS

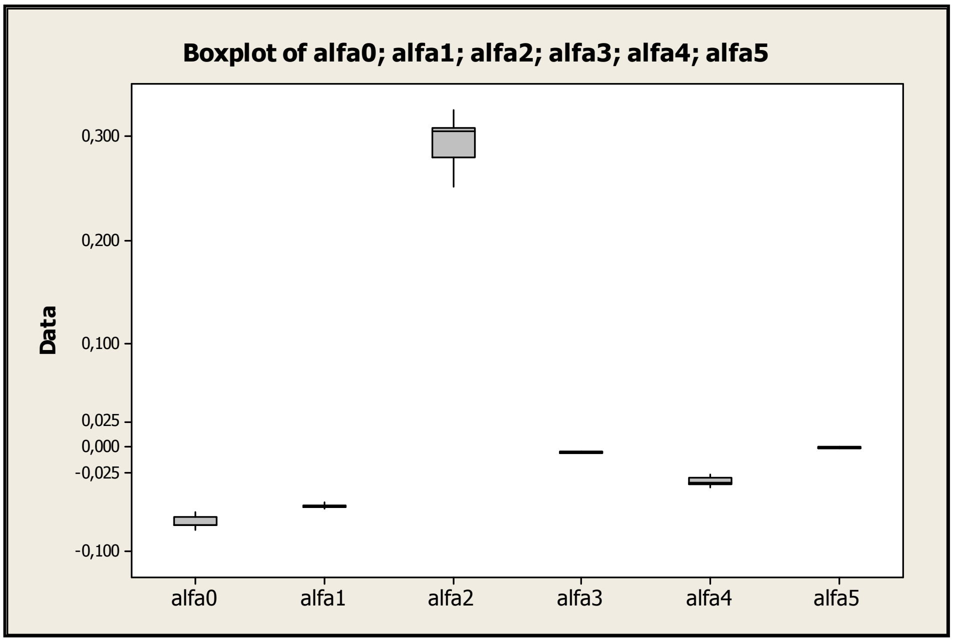


Figure 4.8 The graphical representation for the variance of parameter estimates of RCMARS

As we see in the figures above, the variability of the model parameter estimates of the RCMARS is considerably less than that of MARS and much less than that of CMARS. For this simulation study, the variance of model performance can be seen in Figure 4.9-4.11 for three methods.

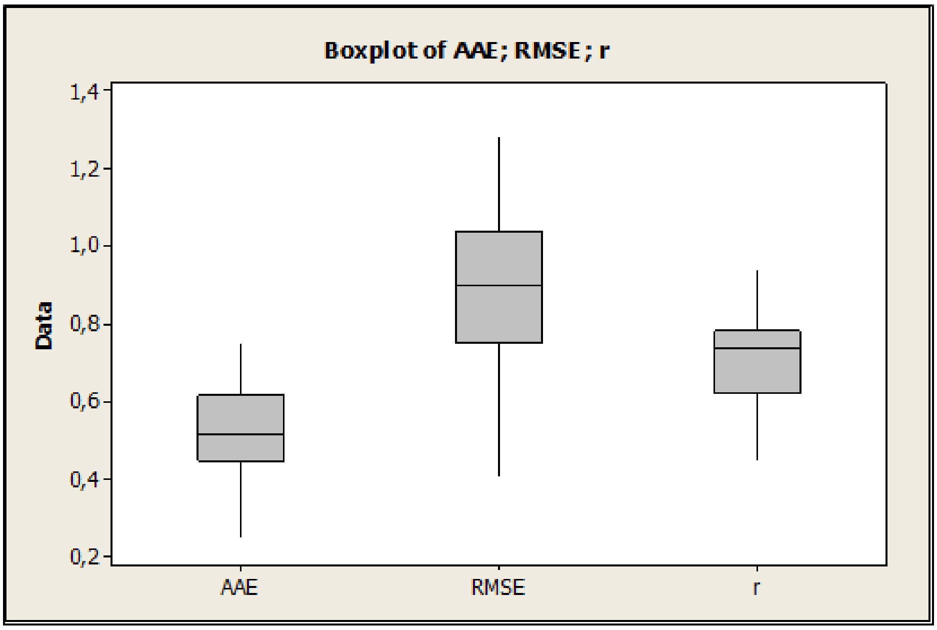


Figure 4.9 The graphical representation for the variance of model performance of MARS

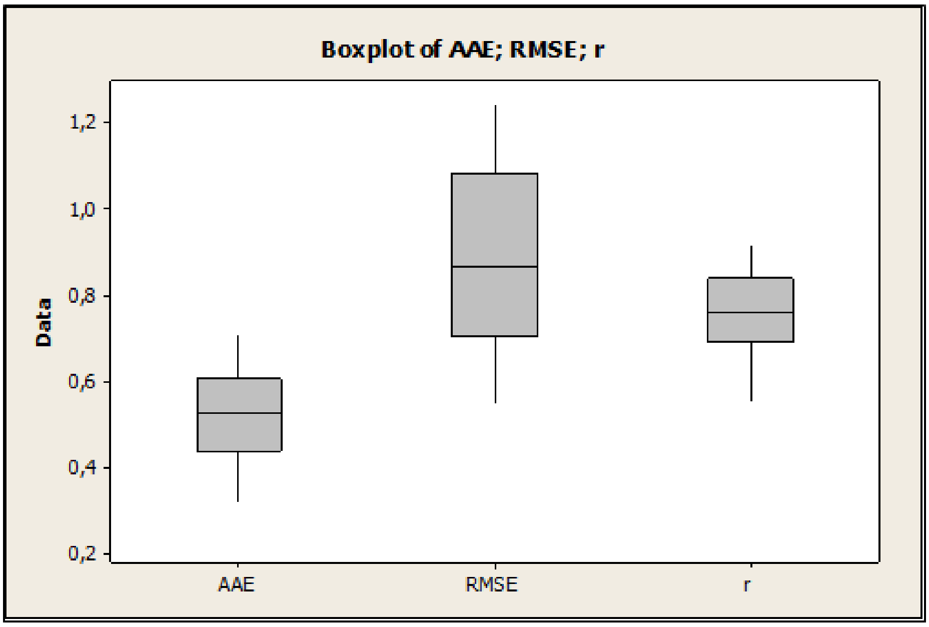


Figure 4.10 The graphical representation for the variance of model performance of CMARS

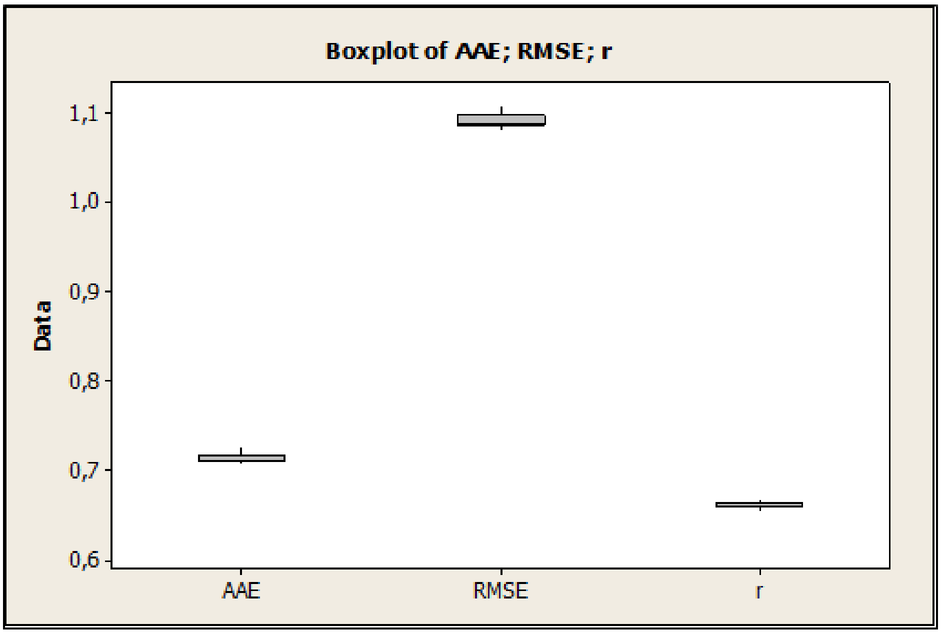


Figure 4.11 The graphical representation for the variance of model performance of RCMARS

As we see by the figures above, similar to variability of the model parameter estimates, the variability of model performance of the RCMARS is considerably less than that of MARS and much less than that of CMARS.

From MARS to CMARS, more BFs are coming into play and the boxes are becoming relatively big. This does not come by surprise since we select BFs and their number differently for CMARS, compared to MARS, and CMARS implies a higher complexity, so that it was expected by us that the variance of the solution estimator would become larger.

From CMARS to RCMARS, the number of BFs diminishes strongly, and the size of the boxes becomes smaller again. This is not surprising either: the robustification rules out some BFs which are created by MARS differently from our original model. Such to say, their coefficients are insignificant for RCMARS and the variances of the parameter estimators became less very strongly.

We also see that the estimators partially changed their values, very much as expressed by positions in the plot. This reflects the fact which we expected that RCMARS is related with a smaller accuracy than CMARS.

Altogether, this shows we are confronted with very heterogeneous outcomes, but all of them being explainable, and that we are on a way of gradually understanding deeper and developing

RCMARS as a powerful tool and real alternative. By heterogeneous outcomes we mean, on the one hand, with pros and cons outcomes regarding criteria such as accuracy, stability, and robustness, especially, a small variance of the estimators. On the other hand, our results are heterogeneous in the sense of partially incomparable: CMARS and RCMARS may have a different number of coefficients (BFs) compared with MARS.

# CHAPTER 5

# CONCLUSION and FUTURE STUDIES

In previous works, the importance and benefit of CQP in this area became already demonstrated. There, the complexity of the regression method MARS, which especially means sensitivity with respect to noise in the data, was penalized in the form of TR, which became expressed and studied as a CQP problem. This was leading to the new method CMARS; it is more model-based and employs continuous, actually, well-structured convex optimization which enables the use of IPMs and their codes such as MOSEK. In this study, we generalize the regression problem by including uncertainty in the model, especially, in the input data, too.

CMARS, developed as an alternative method to MARS, is powerful in handling complex and heterogeneous data. However, for the methods of MARS and CMARS, data are assumed to contain fixed variables. In fact, data include noise in both output and input variables. Consequently, our regarded optimization problem's solutions can show a remarkable sensitivity to perturbations in the parameters of the problem. In this study, we include the existence of uncertainty in the future scenarios into CMARS and robustify it with robust optimization which is coped with data uncertainty. This is some kind of regularization in the input and output domain. It incorporates various kinds of noise and perturbations into the programming problem. This robustification of CQP with RO is compared with previous contributions that based on TR, and with the traditional MARS method.

In this study, we first briefly review the theory and methods of RCMARS, a newly developed method for modeling uncertain data. We run the code for the data that include uncertainties, and then evaluate the results with respect to accuracy and efficiency. Then the results of the sensitivity analysis on the parameter estimates, and thus, the model performances are presented. As expected, CMARS produces more accurate results than RCMARS. As the CIs on the variables become narrower, the performance results approaches those of CMARS.

We solve our optimal problem and robustify CMARS by using our continuous RO approach but, also a combinatorial approach, which is the weakly robust case, to handle uncertainties that may exist in data and to make our rich approach feasible. In this way, we aim to decrease the estimation variance. For this goal, we use 20 different data sets in simulation study and represented our results graphically. Although RCMARS results in less accurate models than CMARS; its models have much less variability in parameter estimates and thus in accuracy measures as expected.

CMARS with its yet given use of MARS is *hybrid* in the sense that it employs a discrete part on optimization and decision and a continuous one. The discrete part comes from the Salford MARS with its selection of a finite number of BFs; the continuous part comes from the use of CQP together with the IPMs of MOSEK. That makes, in our example, a greater number of BFs to appear for the CMARS application and a smaller number for RCMARS. In this respect, the variances of the parameters which are regarded insignificant (i.e., dropped) in RCMARS seems moved into the variances of the remaining parameters that are treated significant (i.e., remaining in the RCMARS model). Herewith, there was a move of risk between CMARS and RCMARS. This explanation lets the advantages and disadvantages of CMARS and RCMARS discussed above appear more relative.

We are confronted with very heterogeneous outcomes and with certain technical incompatibilities because of different continues-discrete characters. However, we are on a way of gradually understanding better and developing RCMARS as a powerful tool and real alternative. But, we may not expect to see the power of RCMARS at this stage because of the aforementioned yet hybrid character of our methods. In fact, the more we can by future works make all of the parts of RCMARS and of the other methods prepared continuous (real-valued), the more we can see our positive improvements.

In this study, although we have small data set for our numerical example, our uncertainty matrix for input data has a huge size, and we do not have enough computer capacity to solve our problem for this uncertainty matrix. Indeed, we have a *tradeoff* between tractability and robustification. To overcome this difficulty, in this example, we obtain different *WRCMARS* model for each sample value (observation) applying combinatorial approach, and solve them by using MOSEK program. As a future study, we will discuss about how we can obtain a more robust model using different methods and about what further research will consist of in this respect.

For now, we work on only wine-quality data with small size. We could not find suitable data sets easily to apply our RCMARS algorithm because our assumption on normally distribution data. As a future work, we will develop the method further by considering other distributional assumptions rather than normal distribution for the data. We will also use robust estimators to construct CIs for our data. We will study on real-world application data in some areas e.g., financial sector, quality management, manufacturing and biotechnology.

We work on this Robust CMARS (RCMARS) in terms of both polyhedral and ellipsoidal uncertainty; however, in this study, we focus on polyhedral uncertainty. This brings us back to CQP naturally. In our future study, we are also going to work on RCMARS under ellipsoidal uncertainty. This can be increase our model complexity but will give us a more robust approximation and prediction than polyhedral uncertainty.

# REFERENCES

[1] Anakli, Z., A Comparison of Data Mining Methods for Prediction and Classification Types of Quality Problems. MSc Thesis, METU, 2009.

[2] Andersen, K.D., *Minimizing a Sum of Norms (Large Scale solutions of symmetric positive definite linear systems)*, PhD thesis, Odense University, 1995.

[3] Andersen, E.D., Roos, C., and Terlaky, T., On implementing a primal-dual interior-point method for conic quadratic optimization, Math. Program., Ser. B 95:249–277, 2003.

[4] Aster, R.C., Borchers B. and Thurber, C., *Parameter Estimation and Inverse Problems*, Academic Press, 2004.

[5] Ben-Tal, A., and Nemirovski, A., Robust convex optimization, Math. Oper. Res. 23, 769–805, 1998

[6] Ben-Tal, A., and Nemirovski, A., Robust solutions to uncertain Linear Programs, Operations Research Letters, 25, 1:1-13, 1999

[7] Ben-Tal, A., and Nemirovski, A., Robust solutions of Linear Programming problems contaminated with uncertain data, Math. Progr. 88:411–424, 2000.

[8] Ben-Tal, A., and Nemirovski, A., *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPR-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

[9] Ben-Tal, A., El-Ghaoui, L., and Nemirovski, A., Robust semidefinite programming, In: Saigal, R., Vandenberghe, L., and Wolkowicz H., (eds.), Semidefinite programming and applications, Kluwer Academic Publishers, 2000.

[10] Ben-Tal, A. and Nemirovski A., Robust optimization – methodology and applications, Mathematical Programming, 92- 3:453-480. 2002.

[11] Ben-Tal, A., El-Ghaoui, L., and Nemirovski, A., *Robust Optimization*, Princeton University Press, 2009.

[12] Bertsimas, D., and Sim, M., Robust discrete optimization and network flows, Math. Progr., 98:49–71, 2003.

[13] Bertsimas, D., and Sim, M., Price of robustness, Oper. Res. 52 (1):35–53, 2004.

[14] Bertsimas, D., and Sim, M., Tractable approximations to robust conic optimization problems, Math. Program., Ser. B 107:5–36, 2006.

[15] Bertsimas, D., Brown, D.B., and Caramanis, C., Theory and applications of robust optimization, Technical report, University of Texas at Austin, Austin, TX, 2007.

[16] Breiman, L., Friedman J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Belmont, CA: Wadsworth Int. Group, 1984.

[17] Boni, O., *Robust Solutions of Conic Quadratic Problems*, PhD Thesis, Technion, Israeli Institute of Technology, IE&M Faculty, 2007.

[18] Boyd, S., and Vanderberghe L., *Convex Optimization*, Cambridge University Press, 2004.

[19] Chen, X., Sim, M., and Sun, P., A Robust Optimization Perspective of Stochastic Programming, Working Paper, National University of Singapore Business School, 2005.

[20] Copyright StatSoft, Inc., *Multivariate Adaptive Regression Splines*, http://www.statsoft.com/textbook/stmars.html (accessed 05 Sep. 2008).

[21] Craven, P., and Wahba, G., Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation, in: Numerische Mathematik, 31, 1979.

[22] Crino, S., and Brown, D.E., Global optimization with multivariate adaptive regression splines, IEEE Transactions on Systems Man and Cybernetics Part b- cybernetics, 37, 2: 333-340, 2007.

[23] De Veaux, R.D., Psichogios, D.C., and Ungar, L.H., A comparison of two non- parametric schemes: MARS and Neural Networks, Computers in Chemical Engineering, 17:819-837, 1993.

[24] Deb, K., *Multi Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, 2001.

[25] Deichmann, J., Eshghi, A., Haughton, D., Sayek, S., and Teebagy, N., Application of multiple adaptive regression splines (MARS) in direct response modeling, Journal of Direct Marketing, 16, 4:15-27, 2002.

[26] Durieu, P., Walter, E., and Polyak, B., *Multi-input multi-output ellipsoidal state bounding*, J. Optim. Theory Appl., 111, 2:273–303, 2001.

[27] El-Ghaoui L. and Lebret, H., Robust solutions to least-square problems to uncertain data matrices, SIAM J. Matrix Anal. Appl. 18:1035–1064, 1997.

[28] El-Ghaoui, L., Oustry, F., and Lebret, H., Robust solutions to uncertain semidefinite programs, SIAM J. Optim. 9:33–52, 1998.

[29] El-Ghaoui, L., Robust Optimization and Applications, IMA Tutorial, 2003.

[30] Elishakoff, I., Whys and hows in uncertainty modelling*:* Probability, Fuzziness and Anti-Optimization, 1999.

[31] Fabozzi F.J., Kolm P.N., Pachamanova D.A., and Focardi S.M., *Robust Portfolio Optimization and Management*, Wiley Finance, 2007.

[32] Falk, J.E., Exact solutions of inexact linear programs, Operations Research, 24:783-787, 1976.

[33] Fox, J., *Nonparametric Regression*, in: B. Everitt and D. Howell, eds. Encyclopedia of Statistics in the Behavioral Sciences. London: Wiley, 2005.

[34] Francis, I.L., Newton, J.L., Determining wine aroma from compositional data, Australian Journal of Grape and Wine Research 11:114–126, 2005.

[35] Friedman, J.H., Multivariate adaptive regression splines, The Annals of Statistics, 19, 1:1-141, 1991.

[36] Güler O., Barrier functions in interior-point methods, Math. Oper. Res., 21:860–885, 1996.

[37] Hadamard, J., *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, 1923.

[38] Hansen, P.C. and O'Leary D. P., The use of the L-curve in the regularization of discrete ill-posed problems, SIAM J. Sci. Comput., 14, 6:1487-1503, 1993.

[39] Hansen, P.C., Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems, Numer. Algorithms, 6 (I–II):1–35, 1994.

[40] Hastie, T.J., and Tibshirani, R.J., *Generalized Additive Models*. Chapman and Hall, London, 1990.

[41] Hastie, T., Tibshirani, R., and Friedman, J.H., *The Element of Statistical Learning*, Springer Verlag, New York, 2001.

[42] Karmakar, N., A new polynomial-time algorithm for linear programming, Combinatorica, 4:373-395, 1984.

[43] Kibzun, A., and Kan, Y., Stochastic Programming Problems with Probability and Quantile Functions, Wiley, 1996.

[44] Kojima, M., Mizuno, S., and Yoshise, A., A primal-dual interior point algorithm for linear programming, In N. Megiddo, editor, Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, pages 29–47. Springer Verlag, Berlin, 1989.

[45] Korn, R. and Korn, E., *Option Pricing and Portfolio Optimization: Modern Methods of Financial Mathematics (Graduate Studies in Mathematics)*, American Mathematical Society, 2001.

[46] Krawczyk-stando, D., and Rudnicki, M., Regularization parameter selection in discrete Ill–posed problems —The use of the U–curve, Int. J. Appl. Math. Comput. Sci., 17, 2:157-164, 2007.

[47] Kriner, M., *Survival Analysis with Multivariate Adaptive Regression Splines*, 2007. Dissertation, LMU Mnchen: Faculty of Mathematics, Computer Science and Statistics.

[48] Kropat, E., Weber, G.-W., and Sekar, P.C., Regulatory networks under ellıpsoidal uncertainty – Optimızation theory and dynamical systems, SIAM, 2009.

[49] Kurzhanski, A.B., and Valyi, I., Ellipsoidal calculus for estimation and control, Birkhauser, 1997.

[50] Kurzhanski, A.A., and Varaiya, P., *Ellipsoidal Toolbox Manual*, EECS Department, University of California, Berkeley, 2008.

[51] Lobo, M.S., Vanderberghe, L., Boyd, S., and Lebret, H., *Applications of second-order cone programming*, Linear Algebra Appl.,193–228, 1998.

[52] Löfberg J., YALMIP: A Toolbox for Modeling and Optimization in MATLAB, 2004. http://users.isy.liu.se/johanl/yalmip.php.

[53] Lustig, I.J., Marsten, R.E., and Shanno, D.F., Interior point methods for linear programming: Computational state of the art. ORSA J. on Comput., 6(1):1–15, 1994.

[54] MARS from Salford Systems, http://www.salfordsystems.com/mars/phb (accessed 05 Sep. 2008).

[55] *MARS User Guide*, San Diego, CA: Salford Systems, 2001.

[56] Minitab package program, http://www.minitab.com/mars/phb (accessed 05 Sep. 2008).

[57] Montgomery D.C. and Runger G.C., Applied Statistics and Probability for Engineers, New York: John Wiley and Sons, 2007.

[58] MOSEK, A *very powerful commercial software for CQP*, http://www.mosek.com (accessed 05 Sep. 2008).

[59] Monteiro, R.D.C., and Adler, I., Interior path following primal-dual algorithms, Part I: Linear programming. Math. Programming, 44:27–41, 1989.

[60] Monteiro, R.D.C., and Tsuchiya T., Polynomial convergence of primal-dual algorithms for the second order cone program based on the MZ-family of directions, Math. Programming, 88(1):61–83, 2000.

[61] Müller, M., Estimation and testing in generalized partial linear models- a comparative study, Statistics and Computing, 11:29-309, 2001.

[62] Myers R.H., and Montgomery D.C., *Response surface methodology: Process and Product Optimization Using Designed Experiments*, Wiley Series in Probability and Statistics, Second edition, John Wiley & Sons Inc., New York: Wiley, 2008.

[63] Nair, M.T., Hegland, M. and Anderssen R.S., The trade-off between regularity and stability in Tikhonov regularization, Mathematics of Computation, 66, 217:193-206, 1997.

[64] Nemirovski, A., On tractable approximations of randomly perturbed convex constraints, In: Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii, USA, 2419–2422, 2003.

[65] Nesterov, Y., and Todd M. J., Self-scaled barriers and interior-point methods for convex programming, Math. Oper. Res., 22(1):1–42, 1997.

[66] Nesterov, Y.E., and Nemirovskii, A.S., Interior Point Methods in Convex Programming, SIAM, 1993.

[67] Özmen, A., Weber, G-W., Batmaz, I., The new robust CMARS (RCMARS) method, preprint at Institute of Applied Mathematics, METU, ISI Proceedings of 24th MEC-EurOPT 2010–Continuous Optimization and Information-Based Technologies in the Financial Sector, Izmir, Turkey, June 23-26, 2010, 362-368; ISBN 978-9955-28-598-4.

[68] Özmen, A., Weber, G-W., Batmaz, I. and Kropat E., RCMARS: Robustification of CMARS with different scenarios under polyhedral uncertainty set, preprint at IAM, METU, to appear in the 3rd Conference on Nonlinear Science and Complexity (NSC 3rd), Ankara, Turkey, July 28-31, 2010.

[69] Popescu, I., Robust mean-covariance solutions for stochastic optimization, Operations Research, 55 (1): 98-112, 2007.

[70] Prajneshu, *Cautionary note About Nonlinear Models in Fisheries*, Ind. J. Fisheries, 38: 231-33, 1991.

[71] Prajneshu, *A Nonlinear statistical Model for Aphid Population Growth*, Jour. Ind. Soc. Ag. Statistics, 51:73-80, 1998.

[72] Ros, L., Sabater, A., and Thomas, F., An ellipsoidal calculus based on propagation and fusion, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 32, 4:430–442, 2002.

[73] Schmieta S. H., and Alizadeh, F., Associative algebras, symmetric cones and polynomial time interior point algorithms, Technical Report RRR 17-98, RUTCOR, Rutgers Center for Operations Research, P.O. Box 5062, New Brunswick, New Jersey, 1998.

[74] Seber, G.F. and Wild C.J. *Nonlinear Regression*, John Wiley and Sons, 1989.

[75] Sim, M., *Robust Optimization*, PhD Thesis, Massachusetts Institute of Technology, Cambridge MA, 2004.

[76] Soyster, A.L., Convex programming with set-inclusive constraints and applications to inexact linear programming, Operations Research, 21:1154-1157, 1973.

[77] Steuer R.E., Multiple Criteria Optimisation: Theory, Computation and Application, New York: John Wiley and Sons, NY, 1986.

[78] Sturm, J.F., Primal-Dual Interior Point Approach to Semidefinite Programming, Vol.156 of Tinbergen Institute Research Series, Thesis Publishers, The Netherlands, 1997.

[79] Sturm J., Using SeDuMi 1.02 a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software Research, 11-12:625-653, 1999.

[80] Taylan, P., Weber, G.-W., and Yerlikaya, F., Continuous optimization applied in MARS for modern applications in finance, science and technology, in the ISI Proceedings of 20th

Mini-EURO Conference Continuous Optimization and Knowledge-Based Technologies, Neringa, Lithuania, 317-322, 2008.

[81] Taylan, P., and Weber, G.-W., Multivariate adaptive regression spline and continuous optimization for modern applications in science, economy and technology, preprint at IAM, METU, 2007.

[82] Taylan, P., Weber, G.-W., and Beck, A., New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology, Journal Optimization 56, 5–6:1–24, 2007.

[83] Taylan, P., Weber, G.-W., Liu L., and Yerlikaya-Özkurt F., On foundations of parameter estimation for Generalized Partial Linear Models with B-Splines and Continuous Optimization to appear in journal Computers and Mathematics with Applications.

[84] Tsuchiya, T., A polynomial primal-dual path-following algorithm for second-order cone programming, Technical report, The Institute of Statistical Mathematics, Tokyo, Japan, 1997.

[85] Tütüncü, R. H., Toh, K.C., and Todd, M.J., Solving semidefinite-quadratic-linear programs using SDPT3, Mathematical Programming Ser. B, 95:189– 217, 2003.

[86] Upton G., and Cook I., *The Dictionary of Statistics*, Oxford University Press Inc., New York, 2008.

[87] Vazhentsev, A.Y., On Internal ellipsoidal approximations for problems of control synthesis with bounded coordinates, J. Comput. System Sci. Int., 39, 3, 399, 2000.

[88] Weber, G.W., Taylan, P., Sezer, D., Köksal, G., Batmaz, I., Yerlikaya, F., Özöğür, S., Shawe-Taylor, J., Özbudak, F., and Akyıldız, E., New Pathways of Research at IAM of METU and Collaboration Proposed - MARS - SVM with Infinitely Many Kernels, Coding Theory and Cryptography Indicated, seminar presentation, distributed at Technion, Israel Institute of Technology, Haifa, Israel, January 20-25, 2008.

[89] Weber, G.-W., Batmaz, I., Köksal G., Taylan P., and Yerlikaya F., CMARS: A New Contribution to Nonparametric Regression with Multivariate Adaptive Regression Splines Supported by Continuous Optimisation, preprint at IAM, METU, submitted for publication, 2009.

[90] Werner, R., Cascading: an adjusted exchange method for robust conic programming, CEJOR, 16:179–189, 2008.

[91] Werner, R., Consistency of robust portfolio estimates, Optimization in Finance, Coimbra, 2007.

[92] Wood, S.N., Generalized additive models, Texts in Statistical Science Series, An Introduction with R, Chapman & Hall/CRC, Boca Raton, FL, 2006.

[93] Yerlikaya, F., *A New Contribution to Nonlinear Robust Regression and Classification with Mars and Its Applications to Data Mining for Quality Control in Manufacturing*, Master Thesis, METU, 2008.

[94] Zareipour, H., Bhattacharya, K., and Canizares, C.A., Forecasting the hourly Ontario energy price by multivariate adaptive regression splines, IEEE, Power Engineering Society General Meeting, 2006.

[95] Zhou, Y., and Leung, H., Predicting object-oriented software maintainability using multivariate adaptive regression splines, Journal of Systems and Software, 80-8:1349-1361, 2007.

# APPENDIX A

## RSS MODEL in THE NUMERICAL EXAMPLE

The model RSS is addressed in Section 4.2.2 On ***accuracy*** (RSS), the model represented as:

$$
\begin{aligned}
\sum_{i=1}^{20}\left(y_i-\boldsymbol{\alpha}^T\boldsymbol{\psi}(\breve{\boldsymbol{b}}_i)\right)^2 = & \left(-1.48931-\alpha_0-0.29234\alpha_1-0.35539\alpha_4\right)^2 \\
& +\left(-0.56113-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(-0,98303-\alpha_0-0,82150\alpha_1-0,05078\alpha_4\right)^2 \\
& +\left(-0,98303-\alpha_0-0,95253\alpha_1-0,00001\alpha_3\right)^2 \\
& +\left(0,45143-\alpha_0-0,49896\alpha_1-1,16766\alpha_3\right)^2 \\
& +\left(-0,39237-\alpha_0-1,87982\alpha_1-0,86307\alpha_4\right)^2 \\
& +\left(-0,22361-\alpha_0-2,38379\alpha_1-1,31996\alpha_3\right)^2 \\
& +\left(0,19829-\alpha_0-0,86686\alpha_1-0,40613\alpha_3\right)^2 \\
& +\left(-0,64551-\alpha_0-0,22674\alpha_2-0,71077\alpha_4-0,04304\alpha_5\right)^2 \\
& +\left(-0,64551-\alpha_0-0,00004\alpha_1-0,30462\alpha_4\right)^2 \\
& +\left(0,02953-\alpha_0-0,34266\alpha_2-0,40616\alpha_4-0,04304\alpha_5\right)^2 \\
& +\left(-1,48931-\alpha_0-0,42329\alpha_2-2,58916\alpha_3-0,80354\alpha_5\right)^2 \\
& +\left(0,70457-\alpha_0-0,45865\alpha_1-0,25383\alpha_3\right)^2 \\
& +\left(2,22341-\alpha_0-1,50177\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(1,71713-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(0,36705-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(1,04209-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(-0,05485-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(-0,39237-\alpha_0-0.01516\alpha_1-0.10152\alpha_3\right)^2 \\
& +\left(1.12647-\alpha_0-0.52408\alpha_2-0.45690\alpha_3-0.94514\alpha_5\right)^2 .
\end{aligned}
$$

# APPENDIX B

## UNCERTAINT MATRICES and VECTORS

The matrices $\boldsymbol{U}, \boldsymbol{W}_{upper}, \boldsymbol{W}_{lower}$ and the vectors $\boldsymbol{v}, \boldsymbol{z}_{upper}, \boldsymbol{z}_{lower}$ which are addressed in Section 4.2.3 are of the following form:

$$\boldsymbol{U} = \begin{bmatrix} |u_{11}| & |u_{12}| & |u_{13}| & |u_{14}| & |u_{15}| \\ |u_{21}| & |u_{22}| & |u_{23}| & |u_{24}| & |u_{25}| \\ |u_{31}| & |u_{32}| & |u_{33}| & |u_{34}| & |u_{35}| \\ |u_{41}| & |u_{42}| & |u_{43}| & |u_{44}| & |u_{45}| \\ |u_{51}| & |u_{52}| & |u_{53}| & |u_{54}| & |u_{55}| \\ |u_{61}| & |u_{62}| & |u_{63}| & |u_{64}| & |u_{65}| \\ |u_{71}| & |u_{72}| & |u_{73}| & |u_{74}| & |u_{75}| \\ |u_{81}| & |u_{82}| & |u_{83}| & |u_{84}| & |u_{85}| \\ |u_{91}| & |u_{92}| & |u_{93}| & |u_{94}| & |u_{95}| \\ |u_{101}| & |u_{102}| & |u_{103}| & |u_{104}| & |u_{105}| \\ |u_{111}| & |u_{112}| & |u_{113}| & |u_{114}| & |u_{115}| \\ |u_{121}| & |u_{122}| & |u_{123}| & |u_{124}| & |u_{125}| \\ |u_{131}| & |u_{132}| & |u_{133}| & |u_{134}| & |u_{135}| \\ |u_{141}| & |u_{142}| & |u_{143}| & |u_{144}| & |u_{145}| \\ |u_{151}| & |u_{152}| & |u_{153}| & |u_{154}| & |u_{155}| \\ |u_{161}| & |u_{162}| & |u_{163}| & |u_{164}| & |u_{165}| \\ |u_{171}| & |u_{172}| & |u_{173}| & |u_{174}| & |u_{175}| \\ |u_{181}| & |u_{182}| & |u_{183}| & |u_{184}| & |u_{185}| \\ |u_{191}| & |u_{192}| & |u_{193}| & |u_{194}| & |u_{195}| \\ |u_{201}| & |u_{202}| & |u_{203}| & |u_{204}| & |u_{205}| \end{bmatrix} =$$

$$\begin{bmatrix}
[3.5246,-3.5246] & 0 & 0 & [3.2123,-3.2123] & 0 \\
[3.8018,-3.8018] & 0 & [3.9025,-3.9025] & 0 & 0 \\
[2.9955,-2.9955] & 0 & 0 & [3.5169,-3.5169] & 0 \\
[2.8644,-2.8644] & 0 & 0 & [3.5677,-3.5677] & 0 \\
[3.3180,-3.3180] & 0 & [2.8364,-2,8364] & 0 & 0 \\
[1.9371,-1.9371] & 0 & 0 & [2.7046,-2.7046] & 0 \\
[1.4332,-1.4332] & 0 & [2.6841,-2.6841] & 0 & 0 \\
[2.9501,-2.9501] & 0 & [3.5979,3.5979] & 0 & 0 \\
0 & [3.4984,-3.4984] & 0 & [2.8569,-2.8569] & [64.4090,-64.4090] \\
[3.8169,-3.8169] & 0 & 0 & [3.2631,-3.2631] & 0 \\
0 & [3.3825,-3.3825] & 0 & [3.1615,-3.1615] & [49.8253,-49.8253] \\
0 & [3.3019,-3.3019] & [1.4149,-1.4149] & 0 & [46.4390,-46.4390] \\
[3.3583,-3.3583] & 0 & [3.7502,3.7502] & 0 & 0 \\
0 & [2.2234,-2.2234] & 0 & [3.0600,-3.0600] & [28.2948,-28.2948] \\
0 & [2.7274,-2.7274] & [2.6333,-2.6333] & 0 & [28.0932,-28.0932] \\
0 & [3.1910,-3.1910] & [2.6333,-2.6333] & 0 & [37.1687,-37.1687] \\
0 & [2.3847,-2.3847] & 0 & [2.8062,-2.8062] & [46.2685,-46.2685] \\
0 & [2.8634,-2.8634] & 0 & [2.8062,-2.8062] & [53.0693,-53.0693] \\
0 & [3.2263,-3.2263] & 0 & [2.0446,-2.0446] & [50.5855,-50.5855] \\
0 & [3.2011,-3.2011] & [3.5472,-3.5472] & 0 & [46.4190,-46.4190]
\end{bmatrix}.$$

$$\boldsymbol{W}_{upper} = \boldsymbol{\psi}\left(\breve{\boldsymbol{b}}\right) + \boldsymbol{U}_{upper} = \begin{bmatrix} 1 & 3.8170 & 0 & 0 & 3.5677 & 0 \\ 1 & 3.8170 & 0 & 4.0041 & 0 & 0 \\ 1 & 3.8170 & 0 & 0 & 3.5677 & 0 \\ 1 & 3.8170 & 0 & 0 & 3.5677 & 0 \\ 1 & 3.8170 & 0 & 4.0041 & 0 & 0 \\ 1 & 3.8170 & 0 & 0 & 3.5677 & 0 \\ 1 & 3.8170 & 0 & 4.0041 & 0 & 0 \\ 1 & 3.8170 & 0 & 4.0041 & 0 & 0 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 64.4521 \\ 1 & 3.8170 & 0 & 0 & 3.5677 & 0 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 50.3782 \\ 1 & 0 & 3.7252 & 4.0041 & 0 & 47.2425 \\ 1 & 3.8170 & 0 & 4.0041 & 0 & 0 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 32.7137 \\ 1 & 0 & 3.7252 & 4.0041 & 0 & 31.5027 \\ 1 & 0 & 3.7252 & 4.0041 & 0 & 38.6644 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 47.5409 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 53.6828 \\ 1 & 0 & 3.7252 & 0 & 3.5677 & 51.2721 \\ 1 & 0 & 3.7252 & 4.0041 & 0 & 47.3642 \end{bmatrix},$$

$$\boldsymbol{W}_{lower}=\boldsymbol{\psi}\left(\breve{\boldsymbol{b}}\right)+\boldsymbol{U}_{lower}=\begin{bmatrix}1 & -3.2323 & 0 & 0 & -2{,}8569 & 0\\ 1 & -3.7866 & 0 & -3{,}8010 & 0 & 0\\ 1 & -2{,}1740 & 0 & 0 & -3{,}4661 & 0\\ 1 & -1{,}9119 & 0 & 0 & -3{,}5677 & 0\\ 1 & -2{,}8190 & 0 & -1{,}6688 & 0 & 0\\ 1 & -0{,}0573 & 0 & 0 & -1{,}8416 & 0\\ 1 & 0{,}9506 & 0 & -1{,}3641 & 0 & 0\\ 1 & -2{,}0833 & 0 & -3{,}1918 & 0 & 0\\ 1 & 0 & -3{,}2717 & 0 & -2{,}1462 & -64{,}3660\\ 1 & -3{,}8169 & 0 & 0 & -2{,}9584 & 0\\ 1 & 0 & -3{,}0399 & 0 & -2{,}7554 & -49{,}2724\\ 1 & 0 & -2{,}8786 & 1{,}1743 & 0 & -45{,}6354\\ 1 & -2{,}8997 & 0 & -3{,}4964 & 0 & 0\\ 1 & 0 & -0{,}7216 & 0 & -2{,}5523 & -23{,}8760\\ 1 & 0 & -1{,}7295 & -1{,}2626 & 0 & -24{,}6837\\ 1 & 0 & -2{,}6568 & -1{,}2626 & 0 & -35{,}6730\\ 1 & 0 & -1{,}0442 & 0 & -2{,}0446 & -44{,}9962\\ 1 & 0 & -2{,}0017 & 0 & -2{,}0446 & -52{,}4559\\ 1 & 0 & -2{,}7274 & 0 & -0{,}5216 & -49{,}8989\\ 1 & 0 & -2{,}6770 & -3{,}0903 & 0 & -45{,}4739\end{bmatrix}.$$

$$\boldsymbol{v} = \begin{bmatrix} |v_1| \\ |v_2| \\ |v_3| \\ |v_4| \\ |v_5| \\ |v_6| \\ |v_7| \\ |v_8| \\ |v_9| \\ |v_{10}| \\ |v_{11}| \\ |v_{12}| \\ |v_{13}| \\ |v_{14}| \\ |v_{15}| \\ |v_{16}| \\ |v_{17}| \\ |v_{18}| \\ |v_{19}| \\ |v_{20}| \end{bmatrix} = \begin{bmatrix} [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \\ [3,-3] \end{bmatrix}, \quad \boldsymbol{z}_{upper} = \breve{\boldsymbol{y}} + \boldsymbol{v}_{upper} = \begin{bmatrix} 1.5107 \\ 2.4389 \\ 2.0170 \\ 2.0170 \\ 3.4514 \\ 2.6076 \\ 2.7764 \\ 3.1983 \\ 2.3545 \\ 2.3545 \\ 3.0295 \\ 1.5107 \\ 3.7046 \\ 5.2234 \\ 4.7171 \\ 3.3671 \\ 4.0421 \\ 2.9452 \\ 2.6076 \\ 4.1265 \end{bmatrix}, \quad \boldsymbol{z}_{lower} = \breve{\boldsymbol{y}} + \boldsymbol{v}_{lower} = \begin{bmatrix} -4.4893 \\ -3.5611 \\ -3.9830 \\ -3.9830 \\ -2.5486 \\ -3.3924 \\ -3.2236 \\ -2.8017 \\ -3.6455 \\ -3.6455 \\ -2.9705 \\ -4.4893 \\ -2.2954 \\ -0.7766 \\ -1.2829 \\ -2.6329 \\ -1.9579 \\ -3.0548 \\ -3.3924 \\ -1.8735 \end{bmatrix}.$$

# APPENDIX C

# MOSEK MODELS for FIRST and LAST OBSERVATIONS

## 1. MOSEK Model for First Observation

After incorporating uncertainty into input and output data and turn it into MOSEK format, our RCMARS the model represented as following form:

$$\underset{t,\boldsymbol{\alpha}}{\text{minimize}}\ t,$$

subject to

$$
\begin{aligned}
1.51069\ -\alpha_0 -\ 3.81696\alpha_1 -\ 3.56769\alpha_4 &= \beta_1,\\
2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 &= \beta_2,\\
2.01697\ -\alpha_0 -\ 0.82150\alpha_1 -\ 0.05078\alpha_4 &= \beta_3\\
2.01697\ -\alpha_0 -\ 0.95253\alpha_1 -\ 0.00001\alpha_4 &= \beta_4\\
3.45143\ -\alpha_0 -\ 0.49896\alpha_1 -\ 1.16766\alpha_3 &= \beta_5\\
2.60763\ -\alpha_0 -\ 1.87982\alpha_1 -\ 0.86307\alpha_4 &= \beta_6\\
2.77639\ -\alpha_0 -\ 2.38379\alpha_1 -\ 1.31996\alpha_3 &= \beta_7\\
3.19829\ -\alpha_0 -\ 0.86686\alpha_1 -\ 0.40613\alpha_3 &= \beta_8\\
2.35449\ -\alpha_0 -\ 0.22674\alpha_2 -\ 0.71077\alpha_4 -\ 0.04304\alpha_5 &= \beta_9\\
2.35449\ -\alpha_0 -\ 0.00004\alpha_1 -\ 0.30462\alpha_4 &= \beta_{10}\\
3.02953\ -\alpha_0 -\ 0.34266\alpha_2 -\ 0.40616\alpha_4 -\ 0.55290\alpha_5 &= \beta_{11}\\
1.51069\ -\alpha_0 -\ 0.42329\alpha_2 -\ 2.58916\alpha_3 -\ 0.80354\alpha_5 &= \beta_{12}\\
3.70457\ -\alpha_0 -\ 0.45865\alpha_1 -\ 0.25383\alpha_3 &= \beta_{13}\\
5.22341\ -\alpha_0 -\ 1.50177\alpha_2 -\ 0.50769\alpha_4 -\ 4.41884\alpha_5 &= \beta_{14}\\
4.71713\ -\alpha_0 -\ 0.99781\alpha_2 -\ 1.37073\alpha_3 -\ 3.40951\alpha_5 &= \beta_{15}\\
3.36705\ -\alpha_0 -\ 0.53416\alpha_2 -\ 1.37073\alpha_3 -\ 1.49567\alpha_5 &= \beta_{16}\\
4.04209\ -\alpha_0 -\ 1.34050\alpha_2 -\ 0.76154\alpha_4 -\ 1.27235\alpha_5 &= \beta_{17}\\
2.94515\ -\alpha_0 -\ 0.86174\alpha_2 -\ 0.76154\alpha_4 -\ 0.61344\alpha_5 &= \beta_{18}\\
2.60763\ -\alpha_0 -\ 0.49888\alpha_2 -\ 1.52306\alpha_4 -\ 0.68661\alpha_5 &= \beta_{19}\\
4.12647\ -\alpha_0 -\ 0.52408\alpha_2 -\ 0.45690\alpha_3 -\ 0.94514\alpha_5 &= \beta_{20},
\end{aligned}
$$

$$
\begin{aligned}
1.51069 - \alpha_0 - 3.81696\alpha_1 + 2.85691\alpha_4 &= \beta_{21} \\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{22} \\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{23} \\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{24} \\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{25} \\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{26} \\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{27} \\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{28} \\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{29} \\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{30} \\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{31} \\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{32} \\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{33} \\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{34} \\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{35} \\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{36} \\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{37} \\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{38} \\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{39} \\
4.12647 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{40},
\end{aligned}
$$

$$
\begin{aligned}
1.51069 - \alpha_0 + 3.23229\alpha_1 - 3.56769\alpha_4 &= \beta_{41} \\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{42} \\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{43} \\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{44} \\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{45} \\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{46} \\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{47} \\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{48} \\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{49} \\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{50} \\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{51} \\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{52} \\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{53} \\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{54} \\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{55}
\end{aligned}
$$

$$\begin{aligned}
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{56} \\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{57} \\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{58} \\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{59} \\
4.12647 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{60},
\end{aligned}$$

$$\begin{aligned}
1.51069 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 &= \beta_{61} \\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{62} \\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{63} \\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{64} \\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{65} \\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{66} \\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{67} \\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{68} \\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{69} \\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{70} \\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{71} \\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{72} \\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{73} \\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{74} \\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{75} \\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{76} \\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{77} \\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{78} \\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{79} \\
4.12647 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{80},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 &= \beta_{81} \\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{82} \\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{83} \\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{84} \\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{85} \\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{86} \\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{87} \\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{88} \\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{89} \\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{90}
\end{aligned}$$

$$-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 = \beta_{91}$$
$$-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 = \beta_{92}$$
$$-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 = \beta_{93}$$
$$-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 = \beta_{94}$$
$$-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 = \beta_{95}$$
$$-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 = \beta_{96}$$
$$-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 = \beta_{97}$$
$$-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 = \beta_{98}$$
$$-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 = \beta_{99}$$
$$-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 = \beta_{100},$$

$$-4.48931 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 = \beta_{101}$$
$$-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 = \beta_{102}$$
$$-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 = \beta_{103}$$
$$-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 = \beta_{104}$$
$$-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 = \beta_{105}$$
$$-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 = \beta_{106}$$
$$-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 = \beta_{107}$$
$$-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 = \beta_{108}$$
$$-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 = \beta_{109}$$
$$-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 = \beta_{110}$$
$$-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 = \beta_{111}$$
$$-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 = \beta_{112}$$
$$-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 = \beta_{113}$$
$$-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 = \beta_{114}$$
$$-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 = \beta_{115}$$
$$-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 = \beta_{116}$$
$$-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 = \beta_{117}$$
$$-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 = \beta_{118}$$
$$-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 = \beta_{119}$$
$$-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 = \beta_{120},$$

$$-4.48931 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 = \beta_{121}$$
$$-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 = \beta_{122}$$
$$-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 = \beta_{123}$$
$$-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 = \beta_{124}$$
$$-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 = \beta_{125}$$

$$\begin{aligned}
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{126} \\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{127} \\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{128} \\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{129} \\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{130} \\
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{131} \\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{132} \\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{133} \\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{134} \\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{135} \\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{136} \\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{137} \\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{138} \\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{139} \\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{140},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 + 3.23229\alpha_1 + 2.85691\alpha_4 &= \beta_{141} \\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{142} \\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{143} \\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{144} \\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{145} \\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{146} \\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{147} \\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{148} \\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{149} \\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{150} \\
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{151} \\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{152} \\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{153} \\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{154} \\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{155} \\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{156} \\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{157} \\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{158} \\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{159} \\
-1.87353 - \alpha_0 - 0.52408\alpha_2 - 0.45690\alpha_3 - 0.94514\alpha_5 &= \beta_{160},
\end{aligned}$$

$$\begin{aligned}
1.5471\alpha_1 &= \beta_{161},\\
1.2656\alpha_2 &= \beta_{162}\\
1.6086\alpha_3 &= \beta_{163}\\
1.3132\alpha_4 &= \beta_{164}\\
0.8166\alpha_5 &= \beta_{165},
\end{aligned}$$

$$\begin{aligned}
&(\beta_1^2+\beta_2^2+\beta_3^2+\beta_4^2+\beta_5^2+\beta_6^2+\beta_7^2+\beta_8^2+\beta_9^2+\beta_{10}^2\\
&+\beta_{11}^2+\beta_{12}^2+\beta_{13}^2+\beta_{14}^2+\beta_{15}^2+\beta_{16}^2+\beta_{17}^2+\beta_{18}^2+\beta_{19}^2+\beta_{20}^2)^{1/2} \qquad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{21}^2+\beta_{22}^2+\beta_{23}^2+\beta_{24}^2+\beta_{25}^2+\beta_{26}^2+\beta_{27}^2+\beta_{28}^2+\beta_{29}^2+\beta_{30}^2\\
&+\beta_{31}^2+\beta_{32}^2+\beta_{33}^2+\beta_{34}^2+\beta_{35}^2+\beta_{36}^2+\beta_{37}^2+\beta_{38}^2+\beta_{39}^2+\beta_{40}^2)^{1/2} \qquad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{41}^2+\beta_{42}^2+\beta_{43}^2+\beta_{44}^2+\beta_{45}^2+\beta_{46}^2+\beta_{47}^2+\beta_{48}^2+\beta_{49}^2+\beta_{50}^2\\
&+\beta_{51}^2+\beta_{52}^2+\beta_{53}^2+\beta_{54}^2+\beta_{55}^2+\beta_{56}^2+\beta_{57}^2+\beta_{58}^2+\beta_{59}^2+\beta_{60}^2)^{1/2} \qquad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{61}^2+\beta_{62}^2+\beta_{63}^2+\beta_{64}^2+\beta_{65}^2+\beta_{66}^2+\beta_{67}^2+\beta_{68}^2+\beta_{69}^2+\beta_{70}^2\\
&+\beta_{71}^2+\beta_{72}^2+\beta_{73}^2+\beta_{74}^2+\beta_{75}^2+\beta_{76}^2+\beta_{77}^2+\beta_{78}^2+\beta_{79}^2+\beta_{80}^2)^{1/2} \qquad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{81}^2+\beta_{82}^2+\beta_{83}^2+\beta_{84}^2+\beta_{85}^2+\beta_{86}^2+\beta_{87}^2+\beta_{88}^2+\beta_{89}^2+\beta_{90}^2\\
&+\beta_{91}^2+\beta_{92}^2+\beta_{93}^2+\beta_{94}^2+\beta_{95}^2+\beta_{96}^2+\beta_{97}^2+\beta_{98}^2+\beta_{99}^2+\beta_{100}^2)^{1/2} \qquad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{101}^2+\beta_{102}^2+\beta_{103}^2+\beta_{104}^2+\beta_{105}^2+\beta_{106}^2+\beta_{107}^2+\beta_{108}^2+\beta_{109}^2+\beta_{110}^2\\
&+\beta_{111}^2+\beta_{112}^2+\beta_{113}^2+\beta_{114}^2+\beta_{115}^2+\beta_{116}^2+\beta_{117}^2+\beta_{118}^2+\beta_{119}^2+\beta_{120}^2)^{1/2} \quad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{121}^2+\beta_{122}^2+\beta_{123}^2+\beta_{124}^2+\beta_{125}^2+\beta_{126}^2+\beta_{127}^2+\beta_{128}^2+\beta_{129}^2+\beta_{130}^2\\
&+\beta_{131}^2+\beta_{132}^2+\beta_{133}^2+\beta_{134}^2+\beta_{135}^2+\beta_{136}^2+\beta_{137}^2+\beta_{138}^2+\beta_{139}^2+\beta_{140}^2)^{1/2} \quad \le t,
\end{aligned}$$

$$\begin{aligned}
&(\beta_{141}^2+\beta_{142}^2+\beta_{143}^2+\beta_{144}^2+\beta_{145}^2+\beta_{146}^2+\beta_{147}^2+\beta_{148}^2+\beta_{149}^2+\beta_{150}^2\\
&+\beta_{151}^2+\beta_{152}^2+\beta_{153}^2+\beta_{154}^2+\beta_{155}^2+\beta_{156}^2+\beta_{157}^2+\beta_{158}^2+\beta_{159}^2+\beta_{160}^2)^{1/2} \quad \le t,
\end{aligned}$$

$$(\beta_{161}+\beta_{162}+\beta_{163}+\beta_{164}+\beta_{165})^{1/2} \quad \le (\breve{M})^{1/2}.$$

## 2. MOSEK Model for Last Observation

After incorporating uncertainty into input and output data and turn it into MOSEK format, our RCMARS the model represented as following form:

$$\begin{array}{l} \underset{t,\boldsymbol{\alpha}}{\text{minimize}}\ t, \\ \text{subject to} \end{array}$$

$$\begin{array}{ll}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 & = \beta_1, \\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & = \beta_2, \\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 & = \beta_3 \\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 & = \beta_4 \\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 & = \beta_5 \\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 & = \beta_6 \\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 & = \beta_7 \\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 & = \beta_8 \\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 & = \beta_9 \\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 & = \beta_{10} \\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 & = \beta_{11} \\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 & = \beta_{12} \\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 & = \beta_{13} \\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 & = \beta_{14} \\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 & = \beta_{15} \\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 & = \beta_{16} \\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 & = \beta_{17} \\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 & = \beta_{18} \\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 & = \beta_{19} \\
4.12647 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 - 47.36416\alpha_5 & = \beta_{20},
\end{array}$$

$$\begin{array}{ll}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 & = \beta_{21} \\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 & = \beta_{22} \\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 & = \beta_{23} \\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 & = \beta_{24} \\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 & = \beta_{25}
\end{array}$$

$$\begin{aligned}
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{26}\\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{27}\\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{28}\\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{29}\\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{30}\\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{31}\\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{32}\\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{33}\\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{34}\\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{35}\\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{36}\\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{37}\\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{38}\\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{39}\\
4.12647 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 + 45.47389\alpha_5 &= \beta_{40},
\end{aligned}$$

$$\begin{aligned}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{41},\\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{42}\\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{43}\\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{44}\\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{45}\\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{46}\\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{47}\\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{48}\\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{49}\\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{50}\\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{51}\\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{52}\\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{53}\\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{54}\\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{55}\\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{56}\\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{57}\\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{58}\\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{59}\\
4.12647 - \alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 &= \beta_{60},
\end{aligned}$$

$$\begin{aligned}
&1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{61}\\
&2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{62}\\
&2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &&= \beta_{63}\\
&2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &&= \beta_{64}\\
&3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &&= \beta_{65}\\
&2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &&= \beta_{66}\\
&2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &&= \beta_{67}\\
&3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &&= \beta_{68}\\
&2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &&= \beta_{69}\\
&2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &&= \beta_{70}\\
&3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &&= \beta_{71}\\
&1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &&= \beta_{72}\\
&3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &&= \beta_{73}\\
&5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &&= \beta_{74}\\
&4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &&= \beta_{75}\\
&3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &&= \beta_{76}\\
&4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &&= \beta_{77}\\
&2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &&= \beta_{78}\\
&2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &&= \beta_{79}\\
&4.12647 - \alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 &&= \beta_{80},
\end{aligned}$$

$$\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{81}\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{82}\\
&-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &&= \beta_{83}\\
&-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &&= \beta_{84}\\
&-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &&= \beta_{85}\\
&-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &&= \beta_{86}\\
&-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &&= \beta_{87}\\
&-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &&= \beta_{88}\\
&-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &&= \beta_{89}\\
&-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &&= \beta_{90}\\
&-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &&= \beta_{91}\\
&-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &&= \beta_{92}\\
&-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &&= \beta_{93}\\
&-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &&= \beta_{94}\\
&-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &&= \beta_{95}
\end{aligned}$$

$$
\begin{aligned}
&-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &&= \beta_{96}\\
&-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &&= \beta_{97}\\
&-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &&= \beta_{98}\\
&-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &&= \beta_{99}\\
&-1.87353 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 - 47.36416\alpha_5 &&= \beta_{100},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{101}\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{102}\\
&-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &&= \beta_{103}\\
&-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &&= \beta_{104}\\
&-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &&= \beta_{105}\\
&-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &&= \beta_{106}\\
&-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &&= \beta_{107}\\
&-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &&= \beta_{108}\\
&-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &&= \beta_{109}\\
&-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &&= \beta_{110}\\
&-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &&= \beta_{111}\\
&-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &&= \beta_{112}\\
&-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &&= \beta_{113}\\
&-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &&= \beta_{114}\\
&-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &&= \beta_{115}\\
&-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &&= \beta_{116}\\
&-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &&= \beta_{117}\\
&-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &&= \beta_{118}\\
&-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &&= \beta_{119}\\
&-1.87353 - \alpha_0 - 3.72516\alpha_2 - 4.00407\alpha_3 + 45,47389\alpha_5 &&= \beta_{120},
\end{aligned}
$$

$$
\begin{aligned}
&-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &&= \beta_{121}\\
&-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &&= \beta_{122}\\
&-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &&= \beta_{123}\\
&-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &&= \beta_{124}\\
&-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &&= \beta_{125}\\
&-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &&= \beta_{126}\\
&-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &&= \beta_{127}\\
&-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &&= \beta_{128}\\
&-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &&= \beta_{129}\\
&-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &&= \beta_{130}
\end{aligned}
$$

$$
\begin{aligned}
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{131}\\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{132}\\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{133}\\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{134}\\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{135}\\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{136}\\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{137}\\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{138}\\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{139}\\
-1.87353 - \alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 &= \beta_{140},
\end{aligned}
$$

$$
\begin{aligned}
-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{141}\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{142}\\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{143}\\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{144}\\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{145}\\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{146}\\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{147}\\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{148}\\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{149}\\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{150}\\
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{151}\\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{152}\\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{153}\\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{154}\\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{155}\\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{156}\\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{157}\\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{158}\\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{159}\\
-1.87353 - \alpha_0 - 3.72516\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 &= \beta_{160},
\end{aligned}
$$

$$
\begin{aligned}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{161},\\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{162},\\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{163}\\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{164}\\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{165}
\end{aligned}
$$

$$\begin{array}{ll}
2.60763\ -\alpha_0 -\ 1.87982\alpha_1 -\ 0.86307\alpha_4 & =\beta_{166}\\
2.77639\ -\alpha_0 -\ 2.38379\alpha_1 -\ 1.31996\alpha_3 & =\beta_{167}\\
3.19829\ -\alpha_0 -\ 0.86686\alpha_1 -\ 0.40613\alpha_3 & =\beta_{168}\\
2.35449\ -\alpha_0 -\ 0.22674\alpha_2 -\ 0.71077\alpha_4 -\ 0.04304\alpha_5 & =\beta_{169}\\
2.35449\ -\alpha_0 -\ 0.00004\alpha_1 -\ 0.30462\alpha_4 & =\beta_{170}\\
3.02953\ -\alpha_0 -\ 0.34266\alpha_2 -\ 0.40616\alpha_4 -\ 0.55290\alpha_5 & =\beta_{171}\\
1.51069\ -\alpha_0 -\ 0.42329\alpha_2 -\ 2.58916\alpha_3 -\ 0.80354\alpha_5 & =\beta_{172}\\
3.70457\ -\alpha_0 -\ 0.45865\alpha_1 -\ 0.25383\alpha_3 & =\beta_{173}\\
5.22341\ -\alpha_0 -\ 1.50177\alpha_2 -\ 0.50769\alpha_4 -\ 4.41884\alpha_5 & =\beta_{174}\\
4.71713\ -\alpha_0 -\ 0.99781\alpha_2 -\ 1.37073\alpha_3 -\ 3.40951\alpha_5 & =\beta_{175}\\
3.36705\ -\alpha_0 -\ 0.53416\alpha_2 -\ 1.37073\alpha_3 -\ 1.49567\alpha_5 & =\beta_{176}\\
4.04209\ -\alpha_0 -\ 1.34050\alpha_2 -\ 0.76154\alpha_4 -\ 1.27235\alpha_5 & =\beta_{177}\\
2.94515\ -\alpha_0 -\ 0.86174\alpha_2 -\ 0.76154\alpha_4 -\ 0.61344\alpha_5 & =\beta_{178}\\
2.60763\ -\alpha_0 -\ 0.49888\alpha_2 -\ 1.52306\alpha_4 -\ 0.68661\alpha_5 & =\beta_{179}\\
4.12647\ -\alpha_0 +\ 2.67700\alpha_2 -\ 4.00407\alpha_3 -\ 47.36416\alpha_5 & =\beta_{180},
\end{array}$$

$$\begin{array}{ll}
1.51069\ -\alpha_0 -\ 0.29234\alpha_1 -\ 0.35539\alpha_4 & =\beta_{181}\\
2.43887\ -\alpha_0 -\ 0.01516\alpha_1 -\ 0.10152\alpha_3 & =\beta_{182}\\
2.01697\ -\alpha_0 -\ 0.82150\alpha_1 -\ 0.05078\alpha_4 & =\beta_{183}\\
2.01697\ -\alpha_0 -\ 0.95253\alpha_1 -\ 0.00001\alpha_4 & =\beta_{184}\\
3.45143\ -\alpha_0 -\ 0.49896\alpha_1 -\ 1.16766\alpha_3 & =\beta_{185}\\
2.60763\ -\alpha_0 -\ 1.87982\alpha_1 -\ 0.86307\alpha_4 & =\beta_{186}\\
2.77639\ -\alpha_0 -\ 2.38379\alpha_1 -\ 1.31996\alpha_3 & =\beta_{187}\\
3.19829\ -\alpha_0 -\ 0.86686\alpha_1 -\ 0.40613\alpha_3 & =\beta_{188}\\
2.35449\ -\alpha_0 -\ 0.22674\alpha_2 -\ 0.71077\alpha_4 -\ 0.04304\alpha_5 & =\beta_{189}\\
2.35449\ -\alpha_0 -\ 0.00004\alpha_1 -\ 0.30462\alpha_4 & =\beta_{190}\\
3.02953\ -\alpha_0 -\ 0.34266\alpha_2 -\ 0.40616\alpha_4 -\ 0.55290\alpha_5 & =\beta_{191}\\
1.51069\ -\alpha_0 -\ 0.42329\alpha_2 -\ 2.58916\alpha_3 -\ 0.80354\alpha_5 & =\beta_{192}\\
3.70457\ -\alpha_0 -\ 0.45865\alpha_1 -\ 0.25383\alpha_3 & =\beta_{193}\\
5.22341\ -\alpha_0 -\ 1.50177\alpha_2 -\ 0.50769\alpha_4 -\ 4.41884\alpha_5 & =\beta_{194}\\
4.71713\ -\alpha_0 -\ 0.99781\alpha_2 -\ 1.37073\alpha_3 -\ 3.40951\alpha_5 & =\beta_{195}\\
3.36705\ -\alpha_0 -\ 0.53416\alpha_2 -\ 1.37073\alpha_3 -\ 1.49567\alpha_5 & =\beta_{196}\\
4.04209\ -\alpha_0 -\ 1.34050\alpha_2 -\ 0.76154\alpha_4 -\ 1.27235\alpha_5 & =\beta_{197}\\
2.94515\ -\alpha_0 -\ 0.86174\alpha_2 -\ 0.76154\alpha_4 -\ 0.61344\alpha_5 & =\beta_{198}\\
2.60763\ -\alpha_0 -\ 0.49888\alpha_2 -\ 1.52306\alpha_4 -\ 0.68661\alpha_5 & =\beta_{199}\\
4.12647\ -\alpha_0 +\ 2.67700\alpha_2 -\ 4.00407\alpha_3 +\ 45,47389\alpha_5 & =\beta_{200},
\end{array}$$

$$
\begin{aligned}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{201}\\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{202}\\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{203}\\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{204}\\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{205}\\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{206}\\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{207}\\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{208}\\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{209}\\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{210}\\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{211}\\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{212}\\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{213}\\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{214}\\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{215}\\
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{216}\\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{217}\\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{218}\\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{219}\\
4.12647 - \alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 &= \beta_{220},
\end{aligned}
$$

$$
\begin{aligned}
1.51069 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{221}\\
2.43887 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{222}\\
2.01697 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{223}\\
2.01697 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{224}\\
3.45143 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{225}\\
2.60763 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{226}\\
2.77639 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{227}\\
3.19829 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{228}\\
2.35449 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{229}\\
2.35449 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{230}\\
3.02953 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{231}\\
1.51069 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{232}\\
3.70457 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{233}\\
5.22341 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{234}\\
4.71713 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{235}
\end{aligned}
$$

$$\begin{aligned}
3.36705 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{236}\\
4.04209 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{237}\\
2.94515 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{238}\\
2.60763 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{239}\\
4.12647 - \alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 + 45.47389\alpha_5 &= \beta_{240},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{241}\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{242}\\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{243}\\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{244}\\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{245}\\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{246}\\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{247}\\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{248}\\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{249}\\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{250}\\
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{251}\\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{252}\\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{253}\\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{254}\\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{255}\\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{256}\\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{257}\\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{258}\\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{259}\\
-1.87353 - \alpha_0 + 2.67700\alpha_2 - 4.00407\alpha_3 - 47.36416\alpha_5 &= \beta_{260},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{261}\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{262}\\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{263}\\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{264}\\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{265}\\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{266}\\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{267}\\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{268}\\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{269}\\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{270}
\end{aligned}$$

$$\begin{aligned}
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{271}\\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{272}\\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{273}\\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{274}\\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{275}\\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{276}\\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{277}\\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{278}\\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{279}\\
-1.87353 - \alpha_0 + 2.67700\alpha_2 - 4.00407\alpha_3 + 45.47389\alpha_5 &= \beta_{280},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{281}\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{282}\\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{283}\\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{284}\\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{285}\\
-3.39237 - \alpha_0 - 1.87982\alpha_1 - 0.86307\alpha_4 &= \beta_{286}\\
-3.22361 - \alpha_0 - 2.38379\alpha_1 - 1.31996\alpha_3 &= \beta_{287}\\
-2.80171 - \alpha_0 - 0.86686\alpha_1 - 0.40613\alpha_3 &= \beta_{288}\\
-3.64551 - \alpha_0 - 0.22674\alpha_2 - 0.71077\alpha_4 - 0.04304\alpha_5 &= \beta_{289}\\
-3.64551 - \alpha_0 - 0.00004\alpha_1 - 0.30462\alpha_4 &= \beta_{290}\\
-2.97047 - \alpha_0 - 0.34266\alpha_2 - 0.40616\alpha_4 - 0.55290\alpha_5 &= \beta_{291}\\
-4.48931 - \alpha_0 - 0.42329\alpha_2 - 2.58916\alpha_3 - 0.80354\alpha_5 &= \beta_{292}\\
-2.29543 - \alpha_0 - 0.45865\alpha_1 - 0.25383\alpha_3 &= \beta_{293}\\
-0.77659 - \alpha_0 - 1.50177\alpha_2 - 0.50769\alpha_4 - 4.41884\alpha_5 &= \beta_{294}\\
-1.28287 - \alpha_0 - 0.99781\alpha_2 - 1.37073\alpha_3 - 3.40951\alpha_5 &= \beta_{295}\\
-2.63295 - \alpha_0 - 0.53416\alpha_2 - 1.37073\alpha_3 - 1.49567\alpha_5 &= \beta_{296}\\
-1.95791 - \alpha_0 - 1.34050\alpha_2 - 0.76154\alpha_4 - 1.27235\alpha_5 &= \beta_{297}\\
-3.05485 - \alpha_0 - 0.86174\alpha_2 - 0.76154\alpha_4 - 0.61344\alpha_5 &= \beta_{298}\\
-3.39237 - \alpha_0 - 0.49888\alpha_2 - 1.52306\alpha_4 - 0.68661\alpha_5 &= \beta_{299}\\
-1.87353 - \alpha_0 + 2.67700\alpha_2 + 3.09027\alpha_3 - 47.36416\alpha_5 &= \beta_{300},
\end{aligned}$$

$$\begin{aligned}
-4.48931 - \alpha_0 - 0.29234\alpha_1 - 0.35539\alpha_4 &= \beta_{301}\\
-3.56113 - \alpha_0 - 0.01516\alpha_1 - 0.10152\alpha_3 &= \beta_{302}\\
-3.98303 - \alpha_0 - 0.82150\alpha_1 - 0.05078\alpha_4 &= \beta_{303}\\
-3.98303 - \alpha_0 - 0.95253\alpha_1 - 0.00001\alpha_4 &= \beta_{304}\\
-2.54857 - \alpha_0 - 0.49896\alpha_1 - 1.16766\alpha_3 &= \beta_{305}
\end{aligned}$$

$$
\begin{aligned}
&-3.39237-\alpha_0-1.87982\alpha_1-0.86307\alpha_4 &&=\beta_{306}\\
&-3.22361-\alpha_0-2.38379\alpha_1-1.31996\alpha_3 &&=\beta_{307}\\
&-2.80171-\alpha_0-0.86686\alpha_1-0.40613\alpha_3 &&=\beta_{308}\\
&-3.64551-\alpha_0-0.22674\alpha_2-0.71077\alpha_4-0.04304\alpha_5 &&=\beta_{309}\\
&-3.64551-\alpha_0-0.00004\alpha_1-0.30462\alpha_4 &&=\beta_{310}\\
&-2.97047-\alpha_0-0.34266\alpha_2-0.40616\alpha_4-0.55290\alpha_5 &&=\beta_{311}\\
&-4.48931-\alpha_0-0.42329\alpha_2-2.58916\alpha_3-0.80354\alpha_5 &&=\beta_{312}\\
&-2.29543-\alpha_0-0.45865\alpha_1-0.25383\alpha_3 &&=\beta_{313}\\
&-0.77659-\alpha_0-1.50177\alpha_2-0.50769\alpha_4-4.41884\alpha_5 &&=\beta_{314}\\
&-1.28287-\alpha_0-0.99781\alpha_2-1.37073\alpha_3-3.40951\alpha_5 &&=\beta_{315}\\
&-2.63295-\alpha_0-0.53416\alpha_2-1.37073\alpha_3-1.49567\alpha_5 &&=\beta_{316}\\
&-1.95791-\alpha_0-1.34050\alpha_2-0.76154\alpha_4-1.27235\alpha_5 &&=\beta_{317}\\
&-3.05485-\alpha_0-0.86174\alpha_2-0.76154\alpha_4-0.61344\alpha_5 &&=\beta_{318}\\
&-3.39237-\alpha_0-0.49888\alpha_2-1.52306\alpha_4-0.68661\alpha_5 &&=\beta_{319}\\
&-1.87353-\alpha_0+2.67700\alpha_2+3.09027\alpha_3+45.47389\alpha_5 &&=\beta_{320},
\end{aligned}
$$

$$
\begin{aligned}
1.5471\alpha_1 &= \beta_{321},\\
1.2656\alpha_2 &= \beta_{322}\\
1.6086\alpha_3 &= \beta_{323}\\
1.3132\alpha_4 &= \beta_{324}\\
0.8166\alpha_5 &= \beta_{325},
\end{aligned}
$$

$$
\begin{aligned}
&(\beta_1^2+\beta_2^2+\beta_3^2+\beta_4^2+\beta_5^2+\beta_6^2+\beta_7^2+\beta_8^2+\beta_9^2+\beta_{10}^2\\
&+\beta_{11}^2+\beta_{12}^2+\beta_{13}^2+\beta_{14}^2+\beta_{15}^2+\beta_{16}^2+\beta_{17}^2+\beta_{18}^2+\beta_{19}^2+\beta_{20}^2)^{1/2} \quad \le t,
\end{aligned}
$$

$$
\begin{aligned}
&(\beta_{21}^2+\beta_{22}^2+\beta_{23}^2+\beta_{24}^2+\beta_{25}^2+\beta_{26}^2+\beta_{27}^2+\beta_{28}^2+\beta_{29}^2+\beta_{30}^2\\
&+\beta_{31}^2+\beta_{32}^2+\beta_{33}^2+\beta_{34}^2+\beta_{35}^2+\beta_{36}^2+\beta_{37}^2+\beta_{38}^2+\beta_{39}^2+\beta_{40}^2)^{1/2} \quad \le t,
\end{aligned}
$$

$$
\begin{aligned}
&(\beta_{41}^2+\beta_{42}^2+\beta_{43}^2+\beta_{44}^2+\beta_{45}^2+\beta_{46}^2+\beta_{47}^2+\beta_{48}^2+\beta_{49}^2+\beta_{50}^2\\
&+\beta_{51}^2+\beta_{52}^2+\beta_{53}^2+\beta_{54}^2+\beta_{55}^2+\beta_{56}^2+\beta_{57}^2+\beta_{58}^2+\beta_{59}^2+\beta_{60}^2)^{1/2} \quad \le t,
\end{aligned}
$$

$$
\begin{aligned}
&(\beta_{61}^2+\beta_{62}^2+\beta_{63}^2+\beta_{64}^2+\beta_{65}^2+\beta_{66}^2+\beta_{67}^2+\beta_{68}^2+\beta_{69}^2+\beta_{70}^2\\
&+\beta_{71}^2+\beta_{72}^2+\beta_{73}^2+\beta_{74}^2+\beta_{75}^2+\beta_{76}^2+\beta_{77}^2+\beta_{78}^2+\beta_{79}^2+\beta_{80}^2)^{1/2} \quad \le t,
\end{aligned}
$$

$$
\begin{aligned}
&(\beta_{81}^2+\beta_{82}^2+\beta_{83}^2+\beta_{84}^2+\beta_{85}^2+\beta_{86}^2+\beta_{87}^2+\beta_{88}^2+\beta_{89}^2+\beta_{90}^2\\
&+\beta_{91}^2+\beta_{92}^2+\beta_{93}^2+\beta_{94}^2+\beta_{95}^2+\beta_{96}^2+\beta_{97}^2+\beta_{98}^2+\beta_{99}^2+\beta_{100}^2)^{1/2} \quad \le t,
\end{aligned}
$$

$$\begin{aligned}&(\beta_{101}^2+\beta_{102}^2+\beta_{103}^2+\beta_{104}^2+\beta_{105}^2+\beta_{106}^2+\beta_{107}^2+\beta_{108}^2+\beta_{109}^2+\beta_{110}^2\\&+\beta_{111}^2+\beta_{112}^2+\beta_{113}^2+\beta_{114}^2+\beta_{115}^2+\beta_{116}^2+\beta_{117}^2+\beta_{118}^2+\beta_{119}^2+\beta_{120}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{121}^2+\beta_{122}^2+\beta_{123}^2+\beta_{124}^2+\beta_{125}^2+\beta_{126}^2+\beta_{127}^2+\beta_{128}^2+\beta_{129}^2+\beta_{130}^2\\&+\beta_{131}^2+\beta_{132}^2+\beta_{133}^2+\beta_{134}^2+\beta_{135}^2+\beta_{136}^2+\beta_{137}^2+\beta_{138}^2+\beta_{139}^2+\beta_{140}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{141}^2+\beta_{142}^2+\beta_{143}^2+\beta_{144}^2+\beta_{145}^2+\beta_{146}^2+\beta_{147}^2+\beta_{148}^2+\beta_{149}^2+\beta_{150}^2\\&+\beta_{151}^2+\beta_{152}^2+\beta_{153}^2+\beta_{154}^2+\beta_{155}^2+\beta_{156}^2+\beta_{157}^2+\beta_{158}^2+\beta_{159}^2+\beta_{160}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{161}^2+\beta_{162}^2+\beta_{163}^2+\beta_{164}^2+\beta_{165}^2+\beta_{166}^2+\beta_{167}^2+\beta_{168}^2+\beta_{169}^2+\beta_{170}^2\\&+\beta_{171}^2+\beta_{172}^2+\beta_{173}^2+\beta_{174}^2+\beta_{175}^2+\beta_{176}^2+\beta_{177}^2+\beta_{178}^2+\beta_{179}^2+\beta_{180}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{181}^2+\beta_{182}^2+\beta_{183}^2+\beta_{184}^2+\beta_{185}^2+\beta_{186}^2+\beta_{187}^2+\beta_{188}^2+\beta_{189}^2+\beta_{190}^2\\&+\beta_{191}^2+\beta_{192}^2+\beta_{193}^2+\beta_{194}^2+\beta_{195}^2+\beta_{196}^2+\beta_{197}^2+\beta_{198}^2+\beta_{199}^2+\beta_{200}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{201}^2+\beta_{202}^2+\beta_{203}^2+\beta_{204}^2+\beta_{205}^2+\beta_{206}^2+\beta_{207}^2+\beta_{208}^2+\beta_{209}^2+\beta_{210}^2\\&+\beta_{211}^2+\beta_{212}^2+\beta_{213}^2+\beta_{214}^2+\beta_{215}^2+\beta_{216}^2+\beta_{217}^2+\beta_{218}^2+\beta_{219}^2+\beta_{220}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{221}^2+\beta_{222}^2+\beta_{223}^2+\beta_{224}^2+\beta_{225}^2+\beta_{226}^2+\beta_{227}^2+\beta_{228}^2+\beta_{229}^2+\beta_{230}^2\\&+\beta_{231}^2+\beta_{232}^2+\beta_{233}^2+\beta_{234}^2+\beta_{235}^2+\beta_{236}^2+\beta_{237}^2+\beta_{238}^2+\beta_{239}^2+\beta_{240}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{241}^2+\beta_{242}^2+\beta_{243}^2+\beta_{244}^2+\beta_{245}^2+\beta_{246}^2+\beta_{247}^2+\beta_{248}^2+\beta_{249}^2+\beta_{250}^2\\&+\beta_{251}^2+\beta_{252}^2+\beta_{253}^2+\beta_{254}^2+\beta_{255}^2+\beta_{256}^2+\beta_{257}^2+\beta_{258}^2+\beta_{259}^2+\beta_{260}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{261}^2+\beta_{262}^2+\beta_{263}^2+\beta_{264}^2+\beta_{265}^2+\beta_{266}^2+\beta_{267}^2+\beta_{268}^2+\beta_{269}^2+\beta_{270}^2\\&+\beta_{271}^2+\beta_{272}^2+\beta_{273}^2+\beta_{274}^2+\beta_{275}^2+\beta_{276}^2+\beta_{277}^2+\beta_{278}^2+\beta_{279}^2+\beta_{280}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{281}^2+\beta_{282}^2+\beta_{283}^2+\beta_{284}^2+\beta_{285}^2+\beta_{286}^2+\beta_{287}^2+\beta_{288}^2+\beta_{289}^2+\beta_{290}^2\\&+\beta_{291}^2+\beta_{292}^2+\beta_{293}^2+\beta_{294}^2+\beta_{295}^2+\beta_{296}^2+\beta_{297}^2+\beta_{298}^2+\beta_{299}^2+\beta_{300}^2)^{1/2}\quad\le t,\end{aligned}$$

$$\begin{aligned}&(\beta_{301}^2+\beta_{302}^2+\beta_{303}^2+\beta_{304}^2+\beta_{305}^2+\beta_{306}^2+\beta_{307}^2+\beta_{308}^2+\beta_{309}^2+\beta_{310}^2\\&+\beta_{311}^2+\beta_{312}^2+\beta_{313}^2+\beta_{314}^2+\beta_{315}^2+\beta_{316}^2+\beta_{317}^2+\beta_{318}^2+\beta_{319}^2+\beta_{320}^2)^{1/2}\quad\le t,\end{aligned}$$

$$(\beta_{321}+\beta_{322}+\beta_{323}+\beta_{324}+\beta_{325})^{1/2}\quad\le(\breve{M})^{1/2}.$$

# APPENDIX D

# RESULTS for SIMULATION STUDY

## 1. Simulation Result for MARS

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | MARS | MARS | MARS | MARS | MARS | MARS | MARS | MARS | | MARS | MARS |
| α0 | -0.3970 | -0.5168 | 0.4308 | 1.0917 | -1.1979 | -0.4285 | -1.7981 | -0.6498 | α0 | 1.6127 | 1.6113 |
| α1 | 0.9485 | 0.0000 | -1.0045 | -0.5712 | 0.0000 | 0.0000 | 0.6546 | 0.5567 | α1 | -0.7479 | -0.6205 |
| α2 | 0.0000 | 1.3427 | 0.0000 | 0.0000 | 1.3737 | 1.1185 | 0.0000 | -0.7419 | α2 | 0.0000 | 0.0000 |
| α3 | 0.0000 | 0.0000 | 0.0000 | -0.9557 | 0.3263 | 0.0000 | 0.0000 | 0.0000 | α3 | 0.0000 | 0.0000 |
| α4 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | α4 | 0.0000 | 0.0000 |
| α5 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | α5 | 0.0000 | 0.0000 |
| α6 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | α6 | 0.0000 | 0.0000 |
| α7 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | α7 | 0.0000 | 0.0000 |
| α8 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | AAE | 0.5334 | 0.5444 |
| AAE | 0.6651 | 0.6721 | 0.5919 | 0.7515 | 0.3798 | 0.6255 | 0.5305 | 0.4894 | RMSE | 0.8724 | 1.0307 |
| RMSE | 1.1249 | 1.0743 | 1.0312 | 1.2862 | 0.6394 | 1.0853 | 1.0420 | 0.9702 | r | 0.7479 | 0.6205 |
| r | 0.5780 | 0.6265 | 0.6636 | 0.4488 | 0.8859 | 0.6165 | 0.6546 | 0.7103 | | | |

| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| | MARS | MARS | MARS | MARS | MARS | MARS | MARS | MARS | MARS | MARS |
| α0 | 0.8903 | -0.2782 | 0.1750 | 1.6404 | 0.3220 | 1.4521 | -0.5314 | 1.1906 | 0.8094 | -0.3352 |
| α1 | 0.0000 | 0.0000 | 0.0000 | -0.6180 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| α2 | -0.2574 | 0.0000 | -0.4768 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | -0.3270 | 0.0000 | 0.0000 |
| α3 | 0.0000 | 0.3359 | 0.8175 | 0.0000 | -0.3290 | -0.4462 | 0.0000 | 0.0000 | -0.8244 | -0.3290 |
| α4 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | -0.2679 | 0.9012 | 0.0000 | -0.4452 | 0.0000 |
| α5 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | -0.5988 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| AAE | 0.5037 | 0.4323 | 0.4932 | 0.6327 | 0.5033 | 0.5001 | 0.5319 | 0.4210 | 0.4124 | 0.2484 |
| RMSE | 0.7544 | 0.7537 | 0.7605 | 0.9505 | 0.8248 | 0.9274 | 0.7767 | 0.6179 | 0.6240 | 0.4058 |
| r | 0.7814 | 0.7819 | 0.7773 | 0.6180 | 0.7312 | 0.7522 | 0.7663 | 0.8595 | 0.8565 | 0.9420 |

## 2. Simulation Result for CMARS

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | | CMARS | CMARS |
| **α0** | -0.1534 | -0.1925 | -0.1803 | 0.7089 | -0.5074 | 0.0020 | -0.3890 | -0.2002 | **α0** | 1.0237 | 0.0699 |
| **α1** | 0.3361 | -0.1350 | -0.3305 | -0.2036 | -0.1093 | 0.0524 | 0.2028 | 0.1294 | **α1** | -0.4353 | -0.1837 |
| **α2** | -0.0032 | 0.1496 | 0.1094 | 0.1336 | 0.7672 | 0.2428 | -0.0360 | -0.2481 | **α2** | -0.1246 | 0.0718 |
| **α3** | 0.0702 | 0.0611 | 0.0337 | 0.0028 | 0.1273 | -0.0407 | 0.1401 | 0.0155 | **α3** | -0.0731 | 0.1296 |
| **α4** | -0.1073 | 0.0649 | 0.0830 | -0.1383 | 0.0051 | -0.0822 | -0.2328 | 0.0323 | **α4** | 0.1691 | 0.0912 |
| **α5** | 0.1393 | 0.0480 | -0.0163 | -0.2969 | -0.1925 | -0.0089 | 0.0348 | -0.0800 | **α5** | 0.0133 | -0.0183 |
| **α6** | 0.2003 | -0.0886 | 0.0557 | 0.0364 | 0.0482 | 0.0074 | -0.0230 | -0.1112 | **α6** | 0.1140 | -0.1951 |
| **α7** | 0.0627 | 0.0400 | 0.2093 | -0.2624 | 0.0447 | -0.0259 | -0.0918 | 0.2199 | **α7** | 0.0749 | 0.2215 |
| **α8** | -0.0500 | -0.0623 | 0.0136 | -0.0071 | -0.0498 | -0.0412 | -0.0363 | -0.0755 | **AAE** | 0.4680 | 0.5600 |
| **AAE** | 0.6072 | 0.6886 | 0.6436 | 0.5971 | 0.4778 | 0.7083 | 0.5993 | 0.5306 | **RMSE** | 0.8518 | 0.9999 |
| **RMSE** | 1.0881 | 1.2078 | 1.1343 | 1.0094 | 0.7892 | 1.2417 | 1.0701 | 1.0904 | **r** | 0.8092 | 0.7005 |
| **r** | 0.6885 | 0.5548 | 0.6805 | 0.7329 | 0.8649 | 0.7252 | 0.6942 | 0.6905 | | | |

| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS | CMARS |
| **α0** | 0.6211 | -0.5328 | 0.6114 | 0.7956 | 0.4417 | 1.0921 | -0.3387 | 1.0282 | 0.3416 | -0.5090 |
| **α1** | -0.1322 | -0.0045 | -0.0643 | -0.1439 | 0.0123 | -0.0811 | -0.0008 | -0.1059 | -0.0274 | 0.1751 |
| **α2** | -0.1602 | 0.0555 | -0.1267 | -0.0862 | -0.1530 | -0.1745 | 0.0473 | -0.2361 | -0.0320 | 0.9914 |
| **α3** | 0.0983 | 0.2400 | 0.1148 | 0.0874 | -0.3316 | -0.3152 | -0.1708 | 0.0272 | -0.2767 | 0.0514 |
| **α4** | -0.0850 | 0.1977 | -0.1235 | 0.1020 | 0.0057 | -0.2304 | 0.6135 | -0.1299 | -0.3854 | -0.0604 |
| **α5** | 0.0983 | -0.0069 | -0.1998 | -0.1254 | -0.1547 | -0.3147 | 0.0017 | 0.0917 | 0.3745 | 0.3669 |
| **AAE** | 0.5012 | 0.4298 | 0.5944 | 0.6379 | 0.4746 | 0.3221 | 0.5270 | 0.3937 | 0.4225 | 0.3295 |
| **RMSE** | 0.7344 | 0.6959 | 0.8756 | 0.9468 | 0.7354 | 0.5715 | 0.7652 | 0.5527 | 0.6860 | 0.6195 |
| **r** | 0.8059 | 0.8282 | 0.7190 | 0.6618 | 0.8011 | 0.8963 | 0.7936 | 0.8949 | 0.8444 | 0.9169 |

## 3. Simulation Result for RCMARS

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS |
| α0 | -0.0550 | -0.0666 | -0.0674 | -0.0676 | -0.0624 | -0.0676 | -0.0677 | -0.0682 | -0.0743 | -0.0744 |
| α1 | -0.0830 | -0.0590 | -0.0575 | -0.0565 | -0.0529 | -0.0547 | -0.0545 | -0.0535 | -0.0566 | -0.0565 |
| α2 | 0.2517 | 0.2788 | 0.2804 | 0.2811 | 0.2584 | 0.2788 | 0.2797 | 0.2817 | 0.3059 | 0.3059 |
| α3 | -0.0020 | -0.0047 | -0.0047 | -0.0046 | -0.0038 | -0.0044 | -0.0044 | -0.0050 | -0.0047 | -0.0048 |
| α4 | 0.0025 | -0.0268 | -0.0284 | -0.0292 | -0.0261 | -0.0300 | -0.0304 | -0.0317 | -0.0347 | -0.0348 |
| α5 | -0.0031 | 0.0001 | 0.0002 | -0.0001 | 0.0003 | 0.0003 | 0.0000 | 0.0000 | -0.0007 | -0.0006 |
| AAE | 0.7264 | 0.7173 | 0.7170 | 0.7171 | 0.7223 | 0.7176 | 0.7175 | 0.7173 | 0.7119 | 0.7119 |
| RMSE | 1.1080 | 1.0966 | 1.0959 | 1.0961 | 1.1041 | 1.0967 | 1.0966 | 1.0960 | 1.0880 | 1.0879 |
| r | 0.6004 | 0.6557 | 0.6586 | 0.6596 | 0.6588 | 0.6621 | 0.6622 | 0.6642 | 0.6648 | 0.6650 |

| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS | RCMARS |
| α0 | -0.0745 | -0.0745 | -0.0745 | -0.0746 | -0.0765 | -0.0795 | -0.0755 | -0.0765 | -0.0785 | -0.0702 |
| α1 | -0.0565 | -0.0565 | -0.0564 | -0.0564 | -0.0571 | -0.0580 | -0.0569 | -0.0572 | -0.0578 | -0.0554 |
| α2 | 0.3059 | 0.3059 | 0.3059 | 0.3059 | 0.3137 | 0.3254 | 0.3098 | 0.3141 | 0.3219 | 0.2903 |
| α3 | -0.0048 | -0.0048 | -0.0048 | -0.0048 | -0.0050 | -0.0054 | -0.0049 | -0.0050 | -0.0053 | -0.0041 |
| α4 | -0.0349 | -0.0349 | -0.0350 | -0.0350 | -0.0365 | -0.0388 | -0.0355 | -0.0364 | -0.0379 | -0.0313 |
| α5 | -0.0005 | -0.0004 | -0.0004 | -0.0003 | -0.0003 | -0.0003 | -0.0005 | -0.0005 | -0.0005 | -0.0011 |
| AAE | 0.7118 | 0.7118 | 0.7117 | 0.7117 | 0.7099 | 0.7073 | 0.7109 | 0.7099 | 0.7081 | 0.7158 |
| RMSE | 1.0878 | 1.0877 | 1.0877 | 1.0876 | 1.0849 | 1.0807 | 1.0864 | 1.0849 | 1.0822 | 1.0940 |
| r | 0.6653 | 0.6654 | 0.6655 | 0.6655 | 0.6666 | 0.6682 | 0.6656 | 0.6662 | 0.6673 | 0.6614 |