Department of Mathematics and Statistics

A STUDY OF OPTIMIZATION PROBLEMS INVOLVING STOCHASTIC SYSTEMS WITH JUMPS

Chunmin Liu

This thesis is presented for the Degree of
Doctor of Philosophy of
Curtin University of Technology

August 2008

Declaration

The work presented in this thesis is my own work and all references are duly acknowledged.

This work has not been submitted, in whole or in part, in respect of any academic award at Curtin University of Technology or elsewhere.

Chunmin Liu

11 August, 2008

Acknowledgments

I sincerely thank my supervisor, Professor Kok Lay Teo, for supporting me over my doctoral studies. I have benefited greatly from his ideas, encouragement and overall guidance. I also thank my co-supervisor, Dr Quanxi Shao, for spending much of his time to share his knowledge during my doctoral studies. I am also grateful to Dr Wei Rong Lee for sharing his ideas during our various discussions. I would also like to thank Associate Professor Yong Hong Wu for his friendly and effective support for all activites associated with my PhD studies. He has been a wonderful Postgraduate Course Coordinator.

I have also benefited from many members in Professor Kok Lay Teo's research group. In particular, I would like to acknowledge Honglei Xu, Siew-Fang Woon, Dr Changzhi Wu, Dr Zhiguo Feng and Ryan Loxton for their friendship and for many stimulating discussions and fruitful collaboration. I am grateful to Dr Darish Nadri, Dr Yongju Liu, Binghui Li, Qun Lin, Kojo Szeto and Hamid Ghaframfor for their friendship and assistance.

My doctoral study in Department of Mathematics and Statistics at Curtin University of Technology has been a thoroughly enjoyable experience.

Abstract

The optimization problems involving stochastic systems are often encountered in financial systems, networks design and routing, supply-chain management, actuarial science, telecommunications systems, statistical pattern recognition analysis associated with electronic commerce and medical diagnosis.

This thesis aims to develop computational methods for solving three optimization problems, where their dynamical systems are described by three different classes of stochastic systems with jumps.

In Chapter 1, a brief review on optimization problems involving stochastic systems with jumps is given. It is then followed by the introduction of three optimization problems, where their dynamical systems are described by three different classes of stochastic systems with jumps. These three stochastic optimization problems will be studied in detail in Chapters 2, 3 and 4, respectively. The literature reviews on optimization problems involving these three stochastic systems with jumps are presented in the last three sections of each of Chapters 2, 3 and 4, respectively.

In Chapter 2, an optimization problem involving nonparametric regression with jump points is considered. A two-stage method is proposed for nonparametric regression with jump points. In the first stage, we identify the rough locations of all the possible jump points of the unknown regression function. In the second stage, we map the yet to be decided jump points into pre-assigned fixed points. In this way, the time domain is divided into several sections. Then the spline function is used to approximate each section of the unknown regression function. These approximation problems are formulated and subsequently solved as optimization problems. The inverse time scaling transformation is then carried out, giving rise to an approximation to the nonparametric regression with jump points. For illustration, several examples are solved by using this method. The result obtained are highly satisfactory.

In Chapter 3, the optimization problem involving nonparametric regression with jump curves is studied. A two-stage method is presented to construct an approximating surface with jump location curve from a set of observed data which are corrupted with noise. In the first stage, we detect an estimate of the jump location curve in a surface. In the second stage, we shift the jump location curve into a row pixels or column pixels. The shifted region is then divided into two disjoint subregions by the jump location row pixels. These subregions are expanded to two overlapping expanded subregions, each of which includes the jump location row pixels. We calculate artificial values at these newly added pixels by using the observed data and then approximate the surface on each expanded subregions in which the artificial values at the pixels in the jump location row pixels for each expanded subregion. The curve with minimal distance between the two surfaces is chosen as the curve dividing the region. Subsequently, two nonoverlapping tensor product cubic spline surfaces are obtained. Then, by carrying out the inverse space scaling transformation, the two fitted smooth surfaces in the original space are obtained. For illustration, a numerical example is solved using the method proposed.

In Chapter 4, a class of stochastic optimal parameter selection problems described by linear Ito stochastic differential equations with state jumps subject to probabilistic constraints on the state is considered, where the times at which the jumps occurred as well as their heights are decision variables. We show that this constrained stochastic impulsive optimal parameter selection problem is equivalent to a deterministic impulsive optimal parameter selection problem subject to continuous state inequality constraints, where the times at which the jumps occurred as well as their heights remain as decision variables. Then we show that this constrained deterministic impulsive optimal parameter selection problem can be transformed into an equivalent constrained deterministic impulsive optimal parameter selection problem with fixed jump times. We approximate the continuous state inequality constraints by a sequence of canonical inequality constraints. This leads to a sequence of approximate deterministic impulsive optimal parameter selection problems subject to canonical inequality constraints. For each of these approximate problems, we derive the gradient formulas of the cost function and the constraint functions. On this basis, an efficient computational method is developed. For illustration, a numerical example is solved.

Finally, Chapter 5 contains some concluding remarks and suggestions for future studies.

List of Publications

(1) Wu C. Z., Liu C. M., Teo K. L. and Shao Q. X., A New Two-stage Method for Nonparametric Regression with Jump Points. In Chaos Control for Circuits and Systems: A Practical Approach, Nonlinear Sciences Series of World Scientific Publishing, to appear in 2008.

(2) Liu C. M., Feng Z. G. and Teo K. L., On a class of Stochastic Impulsive Optimal Parameter Selection Problems, International Journal of Innovative Computing, Information and Control, accepted.

(3) Liu C. M., Wu C. Z.,Teo K. L. and Shao Q. X., A Two-stage Method for Surface Spline Regression with Jump Curves, submitted.

Conference Presentation

(1) Teo K. L., Liu C. M. and Feng Z. G., A Computational Approach to a class of Stochastic Impulsive Optimal Parameter Selection Problems, The 16th International Conference on Finite or Infinite Dimensional Complex Analysis and Applications (16th ICFIDCAA), 28 July- 1 August,2008, Dongguk University, Gyeongju, Korea.

(2) Liu C. M., Wu C. Z. and Teo K. L., A New Two-stage Method for Nonparametric Regression with Jump Points, The 4th Sino-Japanese Optimization Meeting (4th SJOM 2008), 27 August - 31 August, 2008 in National Center for Theoretical Sciences, National Cheng-Kung University, Tainan, Taiwan.

Contents

List of Figures

Chapter 1

An Overview

1.1 A Brief Description

Stochastic systems are systems which are subject to random or stochastic disturbances. The optimization problems involving stochastic systems are dealing with optimal decision making for complex systems involving uncertain data. There are many real world applications in areas such as economics, networks design and routing, supply-chain management, actuarial science, telecommunication systems, statistical pattern recognition analysis associated with electronic commerce and medical diagnosis.

This thesis aims to study three classes of optimization problems, where their dynamics are described by three different classes of stochastic systems with jump phenomena. They are: (i) nonparametric regression with jump points; (ii) nonparametric regression with jump curves; and (iii) stochastic optimal parameter selection problems with state jumps.

1.2 The Optimization Problems Involving Stochastic Systems with Jumps

Nonsmooth and abrupt change phenomena are often observed in environmental systems.

In (Holling, 1985), a review on discontinuous, imperfectly reversible change in ecological systems is presented. On the other hand, the focus of the review reported in (Brooks, 1985) is on socio-technical systems.

In (Ermoliev and Norkin, 1997), a class of stochastic optimization problems involving complex stochastic systems is considered, where the behaviors of the systems could be nonsmooth and could also exhibit abrupt changes. Nonsmooth and discontinuous behaviours are often encountered due to changes in structure of the system. For example, discontinuity phenomena occurred in manufacturing systems, communication networks, and neural nets could be due to the presence of discrete variables in the governing dynamical systems. In environmental systems, abrupt changes could partly be caused by nature.

1.3 Three Optimization Problems Involving Stochastic Systems

1.3.1 Nonparametric regression with jump points

Statistical modelling generally assumes smoothness and continuity of the phenomena of interest. However, some phenomena may experience sudden or sharp changes. For example, groundwater levels may undergo drastic changes in very short durations of time (Shao and Campbell, 2002) due to sudden changes in environment, such as land clearing. In financial portfolio management, the amount of stocks of a particular investor can be viewed as experiencing a jump when he/she purchases or sells his/her stocks. These sudden changes are reflected as jumps in visual display. Without considering these jumps, we may make serious errors in drawing inference about the process under study. It is clearly important to estimate both the number of jumps and their locations and magnitudes. Problems related to regression with jump points have been addressed and investigated for more than two decades. In (McDonald and Owen, 1986), a family of smoothing algorithms that can produce discontinuous output is proposed. The proposed smoother can be used for smoothing with edge detection. It is applied to study sea surface temperature data, where discontinuities are caused by the changes in ocean currents. In (Yin, 1988), an algorithm is proposed to estimate the number, locations and magnitudes of jumps. In this algorithm, the right-sided and left-sided moving averages of the observation at a given point are calculated. The differences between these two values are then used to estimate the number, locations and magnitudes of the jumps.

In (Lombard, 1988), it is demonstrated that Fourier analysis is a useful supplementary tool for calculating the cumulative sum (CUSUM) - a diagnostic tool for detecting change points in the data. This idea is applied to construct sample distribution functions in change-point problems. The technique is used in the simulation study of three data sets appeared in the statistical literature.

In (Lee, 1990), a class of nonparametric regressions with change points is considered. The focuses are on the following points: (i) detecting the locations of discontinuities; (ii) classifying discontinuities by their degrees; and (iii) estimating the sizes of discontinuities. These tasks are achieved from sampled (spare and noisy) data. The algorithm for detecting the jumps and their magnitudes is developed based on pairing patterns.

In (Muller, 1992), an estimator is proposed for detecting locations and sizes of discontinuities or change-points. To detect the location and its magnitude of a discontinuity, the proposed estimator is developed based on a comparison of the left one-sided and right one-sided kernel smoothers. This estimator has been applied to detect discontinuities in financial derivatives based on annual flow volume data.

In (Wu and Chu, 1993), the idea of kernel-type estimators is used to obtain an estimator for the detection of the locations of jump points and their magnitudes of a regression. Precise kernel-type formulas are derived. The almost sure convergence property and the limiting distribution of this kernel-type estimator are obtained. The limiting distribution is used to test the number of jump points and asymptotic confidence intervals for the magnitudes of the jump points of the regression. The choices of kernel functions and bandwidths for setting up these kernel-type estimators are discussed. Simulation studies show that the asymptotic properties hold for reasonable sample sizes.

In (Wang, 1995), a method is proposed to detect the jumps and sharp cusps of an observed function which is interrupted with noise. This method is developed by making use of the large absolute values across fine scale levels of the wavelet transformation of the data. It has been used to detect the jump points and their magnitudes in the simulation study of some simulated examples as well as an example using a set of stock market return data.

In (Muller and Song, 1997), a regression, which is everywhere smooth except at a point, is considered. This point is a discontinuity point and it has only one-sided limit. A two-stage estimation scheme for the detection of the jump locations is proposed, where asymptotic properties of this estimator are also studied.

Most of the estimators mentioned above are of the kernel type. The key idea is to investigate the difference between the estimators of the left- and right-hand side limits at an observation point of the unknown regression function. On this basis, the locations and magnitudes of the jumps can be estimated by using the maximum likelihood argument. However, the overall fittings obtained using these methods are not very satisfactory at around the jump points and at around the end points.

The spline function approximation methods have also been applied for detecting the jump points. In (Koo, 1997), a knot-merging procedure is proposed to estimate a smooth regression function with a finite number of discontinuities at unknown locations by a linear spline approximation with multiple knots. This procedure includes the least squares method, stepwise knot addition, stepwise basis deletion, and knot-merging, while the Baye's information criterion is used to select the final model. The proposed method can detect the jumps of a regression function, and it has been used in the simulation study of real data. In (Miyata and Shen, 2003), a function estimation procedure is proposed for implementation in nonparametric regression. This estimation procedure is based on free-knot splines and an associated algorithm. In contrast to conventional splines with knots confined to distinct design points, the free-knot splines allow the selection of knot numbers, the replacement of knots at any location and placing repeated knots at the same location. This flexibility substantially improves the representation power of splines, leading to an adaptive spline estimator for functions with inhomogeneous smoothness and/or with discontinuities. However, since a large number of spline functions is being used, it is important to devise efficient schemes for knot selection. The existing knot selection schemes, such as the stepwise selection, are not suitable for the purpose due to the knot confounding difficulty. In this paper, a new knot selection scheme is proposed using an evolutionary Monte Carlo algorithm and an adaptive model selection criterion. The evolutionary algorithm locates the optimal knots accurately, while the adaptive model selection strategy guards against the selection error in searching through a large candidate knot space. The performance of the procedure is examined and illustrated via simulations. The usefulness of the procedure is illustrated by an application to an actual data set.

In (Julious, 2001), a two-line model is introduced for parameter estimation of a regression when the location of the change points are known, where an F-test is used to detect a change in the regression coefficient. An algorithm is then proposed for parameter estimation for cases when the change points are unknown. It is observed that for the case when the locations of the change points are not known, the F-test does not conform to its expected parametric distribution. Nonparametric bootstrap methods are thus proposed as a way of overcoming the problems encountered. For illustration, a physiology example is solved using the proposed method, where the regression change represents the change from aerobic to anaerobic energy production.

In (Chu and Marron, 1991), discussions on the differences between two estimators are given, where the weights of one of the estimator are chosen by direct kernel evaluation, while the weights of the other are chosen by convolution of the kernel. It was found that the performances of these estimators are nearly the same for situations, where the design points are fixed and equally spaced. However, the performances of these two estimators are quite different for situations, where the design points are not equally spaced or are randomly chosen.

In (Gijbels, Peter and Kneip, 1999), a two-step method is suggested for a nonparametric regression. The first step is to use a kernel-type diagnostic test to estimate the jump points, while the second step is to construct a smooth curve based on local least-squares.

It is well known that smoothing splines provide nice curves which smooth discrete, noisy data. In (Craven and Wahba, 1979), a practical, effective method for estimating the optimum amount of smoothing from the data is proposed, where the derivatives are estimated from the data by differentiating the resulting (nearly) optimally smoothed spline.

In (Akaike,1974), the history of the development of statistical hypothesis testing in time series analysis is briefly reviewed, where it is noted that the hypothesis testing procedure is not adequately defined as the procedure for statistical model identification. The classical maximum likelihood estimation procedure is reviewed. The Akaike information criterion (AIC) is introduced, which is defined by

AIC = (-2)log-(maximum likelihood) + 2(number of independently adjusted parameters within the model).

On this basis, a new minimum information theoretical criterion (AIC) estimate (MAICE), which is defined by the model and the maximum likelihood estimates of the parameters, is introduced.

When there are several competing models, the MAICE gives the minimum of AIC. MAICE provides a versatile procedure for statistical model identification. It is free from ambiguities that exist when the conventional hypothesis testing procedure is used. The effectiveness of MAICE in time series analysis is demonstrated with some numerical examples.

In (Tong and Lim, 1980), the AIC is proposed for model selection in a threshold autoregression. It is shown that the class of threshold autoregressive models is able to capture the notion of the limit cycle of a nonlinear system. This is important, as the limit cycle plays the main role of the cyclical data modelling. The threshold models are then applied to the study using real data. From the study, it is seen that this new class of models has good potential in the analysis of cyclical data. Simulation results presented in the paper demonstrate that this new class of models exhibits some nonlinear vibrations.

The AIC (Akaike, 1974) and a bias-corrected version, AIC_C (Hurvich and Tsai, 1989; Hurvich and Tsai, 1991) are two methods for the selection of regression and autoregressive models. Both criteria may be viewed as estimators of the expected Kullback-Leibler information. In (Hurvich and Tsai, 1991), the bias of AIC and AIC_C for both the normal linear regression and autoregressive candidate models is investigated in the case of underfitting (i.e. none of the candidate models includes the true model). A simulation study in which the true model is an infinite-order autoregression indicates that AIC_C provides substantially better model selections than AIC, even in the case involving moderate sample sizes.

In (Shao and Campbell, 2002), a method is proposed for the study of the trends in groundwater levels. These trends have been monitored by Agriculture Western Australia throughout the South-West of Western Australia for 10 years. The method is a constrained segmented regression with unknown number of change points. The model parameters are estimated by the multiple linear regression, while a modified AIC is used for model selection. A simulation study has carried out on some typical examples, demonstrating the performance of the proposed method.

1.3.2 Nonparametric regression with jump curves

Surface fitting has many important applications in meteorology, geology and image processing. For example, meteorologists are interested in data fitting for equi-temperature surfaces in sky or in ocean. Geologists reconstruct the mine surfaces according to mineral samples. However, we often encounter situations, where the curves to be fitted are discontinuous. When the discontinuous locations on a surface are curves, they are called jump location curves (JLCs). The jump location curves are not known precisely as the observation data are corrupted with noise. An important example in image processing is the images with step discontinuities (or step edges) at the outlines of the objects. The edges often contain much of the image information. The image structures of the edges are very important for the understanding of the surface structures (Gonzalez and Woods, 1992).

For situations where the JLCs are known, the design space can be separated into several regions. One can then achieve data fitting in each of these regions as usual (Muller and Song 1994).

There are a number of methods available in the statistics literature for detecting the JLCs. In (Korostelev and Tsybakov, 1993), piecewise polynomials are proposed to be used for approximating the JLCs. The maximum likelihood is used to estimate the polynomial coefficients.

In (O'Sullivan and Qian, 1994), an optimization approach is proposed for the estimation of a simple closed curve, which describes the boundary of an object of an image. The objective function of the optimization problem measures the normalized image contrast between inside and outside of a boundary. Numerical methods are developed to implement the approach, and simulation studies are performed to quantify statistical performance characteristics. Two sets of simulations are carried out. The first set models emission computed tomography (ECT) images, while the second set deals with images with a locally coherent noise pattern. In both cases, it is observed that the error characteristics are encouraging. The approach works well in the medical imaging field.

In order to detect the jump location curves, the following method is proposed in (Hall and Raimondo, 1997), where a straight line is thrown randomly into a plane, within which a regular grid is inscribed. Each grid vertex that lies above the line is coloured in black, while each below it is coloured in white. Then, the line is removed and an attempt is made to reconstruct the image from the pattern of vertex colours in an $m \times m$ section of the grid. It is shown that for any $\varepsilon > 0$ the line can be approximated to within the order $m^{-1}(\log m)(\log \log m)^{1+\varepsilon}$, with probability one, and that there is no deterministic subsequence along which the best achievable rate is better than $(m \log m)^{-1}(\log \log m)^{-1-\varepsilon}$with positive probability. These results are not valid for the case when $\varepsilon = 0$. By virtue of these results, near-optimal local linear approximations to general smooth boundaries are developed.

In (Qiu and Yandell, 1997), the problem of locating the jumps in regression surfaces is considered, where it is assumed that the jump surfaces and the noise are mild enough. For a given design point, a least squares plane is fitted in a neighbourhood of this design point. The gradient vector of this plane contains both the continuous and jump information of the regression surface at this design point. The continuous information is removed by using two neighbouring least squares planes along the direction of that gradient vector. In this way, the jump information is extracted. A jump detection algorithm can thus be constructed for which the computational requirement is $O(NK)$, where N is the sample size and k is the window width of the neighbourhood. This property makes it possible to handle large data sets.

In (Wang, 1998), a method based on wavelet theory is proposed to estimate jump and sharp cusp curves of a function in the plane. The asymptotic theory for the method is established. The validity of this asymptotic theory is verified through simulation study. The proposed method is near-optimal and it can be implemented by fast algorithms. The method has been applied to the simulation study of a real image.

In (Qiu, 1997), the question on the estimation of bivariate jump regression functions is considered. Based on differences of two one-sided kernel smoothers and a rotation transformation, an almost sure consistent estimator of the jump location curve is proposed. This estimator is first developed for an ideal case when the jump location curve has an explicit function form. Then, it is generalized to a more general case when the explicit function form does not exist.

Several jump-preserving surface fitting methods are suggested in (Chu, Glad, Godtliebsen and Marron, 1998) and (Qiu, 1998). In (Chu, Glad, Godtliebsen and Marron, 1998), it was argued that classical smoothers cannot avoid blurred image on the sharp edges. The requirement of moderately large smooth stretches for the edge-preserving smoothers is discussed. This is called sigma filtering.

In (Qiu, 1998), a three-stage procedure is proposed for constructing discontinuous regression surfaces (DRSs) from noisy data. The computational requirement of the method is simple, and this method provides a general model under mild assumptions. In the first stage, a jump detection criterion is introduced to detect possible jump positions (jump candidate points). In the second stage, a local principal component (PC) line through the jump candidate points in a neighbourhood of a design point is constructed. This PC line provides a first-order approximation to the true jump location curve (JLC). In the third stage, observations on the same side of the line as the given point are combined. Then, a weighted average procedure is used to fit the surface at the point. For the case when there are no jump candidate points in the neighbourhood, all observations in that neighbourhood are used in the surface fitting. On the other hand, for the case when the centre of the neighbourhood is on a jump location curve, only those observations on one side of the line are used. In this way, the blurring phenomenon can be avoided around the jump locations. This method requires $O(N(k^*)^2)$ computation, where N is the sample size and k^* is the window width.

Edge detection in image processing is essentially an application of jump detection in regression surfaces. In most text books on image processing, the designs of edge detectors are based on gradient estimation. In other words, these detectors make use of properties associated with first-order derivatives of the image intensity function. The first-order derivatives are large or infinite at edge pixels (Gonzalez and Woods 1992) and (Marr and Hildreth, 1980).

In (Marr and Hildreth, 1980), an edge detection theory is presented in two parts. (1) Intensity changes, which could occur over a wide range of scales in any natural image, are detected for different scales separately. An appropriate filter, which is applicable to any given scale, is obtained. The filter is a two-dimensional Gaussian operator. It is shown that the filter does not need to be orientation-dependent under some mild conditions. The way to detect intensity changes for a given scale is to find the zero values of the Laplacian of the convolution of the Gaussian operator with the image. Then, the discovered intensity changes for each channel are represented by the oriented primitives called zero-crossing segments. Images Intensity changes on surface discontinuities, reflectance or illumination boundaries all have spatially localized property. Thus, the zero-crossing segments of different channels are dependent.

In (Canny, 1986), optimal filtering techniques are employed for the design of edge detectors. Other edge detectors are constructed based on various techniques, such as random field models (Geman and Geman 1984), surface fitting (Haralick, 1984), anisotropic diffusion (Perona and Malik, 1990), local smoothing and hypothesis testing (Qiu and Bhandarkar, 1996), residual analysis (Chen, Lee and Pavlidis 1991), global cost minimization using hill-climbing search (Tan, Gelfand and Delp, 1989), simulated annealing (Tan, Gelfand and Delp, 1991), and the genetic algorithm (Bhandarkar, Zhang and Potter, 1994).

In (Canny, 1986), a computational edge detection method is proposed, where a comprehensive set of goals for the computation of edge points is clearly defined. These goals match precisely the desired behaviours of the detector with minimum assumptions on the form of the solution. Detection criterion and localisation criterion are defined for a class of edges, where these criteria are described by appropriate functions in terms of impulse response. A third criterion is introduced for the detector which is restricted to cases where there is only one response to a single edge. Detectors are constructed for several common image features, including step edges, by using existing optimization techniques. Focusing on step edges, it is found that there is a natural uncertainty principle between detection and localisation performance, where detection and localisation are the two main goals. On the basis of this principle, a single operator shape, which is optimal at any scale, is derived. This optimal detector has a single approximate implementation, where edges are marked at maxima in gradient magnitude in a Gaussian smoothed image. By using operators of several widths, this simple detector is generalised to deal with cases involving different signal to noise ratios in the image. A general method is presented for feature synthesis of images involving fine to coarse integration of information from operators at different scales. It is observed that the step edge detector performs considerably better when the operator point spread function is extended along the edge. This detection scheme makes use of several elongated operators at each point, while the directional operator outputs are integrated with the gradient maximum detector.

In (Haralick, 1984), a step edge detection is implemented by using the facet model. In the facet model, any analysis that uses the pixel values in some neighbourhood has its final authoritative interpretation relative to the underlying gray tone intensity surface. These neighbourhood pixel values are observed noisy samples. An edge will occur in a pixel if and only if there is some point in the pixel's area which has a negatively sloped zero crossing of the second directional derivative. This second directional derivative is taken in the direction of a nonzero gradient at the pixel's centre. Whether or not a pixel should be marked as a step edge pixel is determined by its underlying gray tone intensity surface, which is estimated by the pixels in its neighbourhood.

The scale-space technique involves generating coarser resolution images by the convolution of original image with a Gaussian kernel. It is difficult to obtain accurately the locations of the edges at coarse scales. In (Perona and Malik, 1990), a new definition of scale-space is introduced, and a class of algorithms that realize it by using a diffusion process. These algorithms produce images that remain sharp on the region boundaries. Thus, the proposed method supplies a high quality edge detector through exploiting global information.

In (Qiu, 1996), the question on the detection and localization of step edges and roof edges is considered. Based on local smoothing and statistical hypothesis testing, an edge detection technique is proposed, where the hypothesis testing procedures for detection and localization of step edges and roof edges are formulated.

In (Chen, Lee and Pavlidis, 1991), a residual analysis is presented, where the moving average filters of regularization are used to obtain the residual images. The strength of the correlation is used to eliminate noise, weak edges and so forth.

In (Tan, Gelfand and Delp, 1989), an edge detection is presented as a minimization problem. The concept of an edge is introduced in terms of accurate localization, thinness, continuity, and length. The mathematical formulation of a comparative cost function is given for evaluating edges based on the description. The function involves information from both image data and local edge structure. Edges are detected by a proposed heuristic iterative search algorithm for minimizing the comparative cost function. An efficient method for computing the cost function is developed.

In (Tan, Gelfand and Delp, 1991), a mathematical formulation of a cost function to evaluate the quality of edge configurations is given. It is expressed as a linear sum of weighted cost factors. The characteristics of the minimum cost configurations are analysed for edge detection based on the choices of weights. A simulated annealing method is used to minimize the cost function.

In (Bhandarkar, Zhang and Potter, 1994), a genetic algorithm is proposed as a cost minimization technique for edge detection. The edge detection problem is formulated as the problem of choosing a minimum cost edge configuration, where edge configurations are regarded as two-dimensional chromosomes with fitness values inversely proportional to their costs. The crossover operator and the mutation operator are then constructed for the two-dimensional chromosomal representation. The designed knowledge-augmented mutation operator exploits knowledge of the local edge structure in the edge image. It is observed that this genetic algorithm is robust against noise. Its rate of convergence is fast and the quality of the final edge image obtained is good.

Most jump detection methods mentioned above for searching the JLCs are based on maximization/minimization procedures. They are computationally very expensive, as seen in (Muller and Song, 1994; Qiu, 1997) where the maximization procedures in the kernel-type methods are used. In (Korostelev and Tsybakov, 1993), the maximum likelihood technique is used to compute the estimators for the piecewise polynomial coefficients. It is also computationally intensive.

In (Qiu, 2002), a procedure is proposed for detecting jump location curves of a regression from a set of noisy data surfaces based on kernel local smoothing techniques.

In (Stark, 1992), the least square curve fitting method is used. The least square method is one of the curve-fitting methods. The least square curve-fitting method assume that the fitted curve with known data points is an n degree polynomial.

The sum of square of errors between estimated magnitudes and measured magnitudes is the total error square. The least square curve-fitting problem is easy to be solved, where the first order derivatives of the total error square with respect to coefficients of the estimated polynomial are set to be zero. This gives rise to a system of linear equations in coefficients, which can be easily solved by using any existing equation solver. The polynomial with these obtained coefficients is the required fitted curve.

The Lagrangian extrapolation formula can be found in (Rade and Westergren, 1998; Tam, Kurbatskii, 2000). In practice, discrete values are often obtained from sampling or experiment. These discrete values actually represent data points on a continuous function. If the obtained data points are accurate enough, one may wish to find intermediate values of the continuous function on which these data points are located. The process of finding intermediate values is called interpolation. If the obtained data points are not accurate, a continuous function is constructed to fit these data points such that the error is minimized. The process of fitting data points by a continuous curve is called curve-fitting. Interpolation could be regarded as a special curve fitting, in which the function must exactly pass through the obtained data points. The Lagrange interpolation polynomial is the simplest polynomial for implementing interpolation.

The approximation method that uses tensor product splines is a more effective method than other surface approximation techniques (Boor, 2001; Dierckx, 1993) for constructing approximate surfaces of functions with several variables.

In the area of computer aided design, curve and surface fitting is an important step in the design of optimal shapes for many industrial products, such as cars, ships, airplane, propeller blades and so forth. It is also an important tool for describing the physical phenomena arise in geological and medical applications. In (Ferguson, 1986), methods for constructing curves and surfaces are developed. Particular focuses are on the modelling of curves and surfaces using spline curves and tensor product spline surfaces. The paper also points out that many problems in this area can be dealt with by using B-spline representation and then solved by numerical optimization and linear algebra techniques.

Tensor product splines are used in (Boehm, Farin, and Kahman, 1984) for surface modeling. In (Potra and Liu. 2007), the class of tensor product cubic splines is considered for the optimization process for two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) image alignment. They use a combined forward-inverse tensor product cubic splines to align very large collections of gel images.

A project at the University of Texas at Austin makes use of non-uniform rational tensor product cubic splines and rational B-splines trimming curves to achieve the visualization and modeling of large molecules, which is then used as a supporting tool for the synthetic drug design and structural reasoning applications. The rational parameterization allows the representation for the contribution of each atom to a macromolecular structure surface(http://ccvweb.csres.utexas.edu/ccv/projects/

angstrom/sd/shape.php).

In (Wahba, 1990; Duchon, 1977), the formula for the penalized least square method is derived for the construction of the smoothing splines fitting. The penalized least squares method is a frequently used method in the smoothing spline fitting theory for data involving noise. This method is a minimization problem, where a function, which consists of a least-square-like term plus a penalty term, is to be minimized.

A review on curve and surface fitting methods, which include those using tensor product splines proposed in the computer aided design literature, is reported in (Ferguson, 1986).

Although tensor product splines have been used extensively for interpolation and surface fitting, they have not been used for fitting the surfaces with jumps.

In (Muller, and Song, 1994), a maximin estimator is proposed for estimating the location and the magnitude of a discontinuity of a smooth multidimensional regression. The maximin estimator is able to select the boundary on which the minimal estimated directional difference amongst all points in the neighbourhood about this boundary is minimized. The proposed maximin estimator is obtained based on the differences of kernel on strong consistency of the boundary estimators and the corresponding rates of convergence.

1.3.3 Stochastic optimal control problems with state jumps

In (Teo and Jennings, 1999), a general class of optimal control problems is considered, where the cost function is minimized subject to canonical inequality constrains. The classical control parameterization technique partitions the time (planning) horizon into several subintervals. The partition points are called the switching times. Then, piecewise constant or piecewise linear functions with possible discontinuities only at these pre-fixed switching times are used to approximate the control functions. Thus, approximate optimal control problems involving piecewise constant or piecewise linear control functions with pre-fixed switching time points are obtained. These approximate optimal control problems can be regarded as respective optimal parameter selection problems. It is intuitively clear that the accuracy of the approximation depends on the number of the switching time points. However, a larger number of switching time points will give rise to a larger number of decision variables. Thus, to obtain an accurate approximation to the optimal control, the number of decision variables may need to be very large. Consequently, the dimension of the corresponding optimization problem may become too large to be solved effectively. Intuitively, the situation could be much improved if the switching time points are also regarded as decision variables. This intuition is, in fact, valid. However, the gradients of the cost function with respect to these switching points turn out to be discontinuous. Thus, any gradient-based optimization technique based on these gradients will not be effective. To overcome this deficiency, a time scaling transform, which is called the control parameterization enhancing transform (CPET)(Lee, Teo, Rehbock and Jennings, 1997; Teo, Jennings, Lee and Rehbock,1999), is introduced to map these variable switching time points into pre-fixed knots via the introduction of an additional differential equation. With this time scaling transform, the approximate optimal control problems involving piecewise constant or piecewise linear control functions with varying switching time points are converted into equivalent optimal control problems involving piecewise constant or piecewise linear control functions with pre-fixed switching time points. Thus, the transformed problems can be regarded as standard optimal parameter selection problems. Each of which is solvable by any existing gradient-based technique.

In (Friedman, 2006) and (Oksendal, 2003) fundamental theory on Ito stochastic differential equations (SDE) driven by Wiener processes and counting processes (for example Poisson processes) and their many important applications are reported. A stochastic differential equation (SDE) is a differential equation of which at least one term is a stochastic process so that the solution of a stochastic differential equation is also a stochastic process. The SDE is a powerful mathematical tool which has many real world applications in areas ranging from engineering to economics.

In (Aberkane, Ponsart and Sauter, 2006), the problem of static output feedback H-infinity control of a continuous time Active Fault Tolerant Control Systems with Markovian Parameters (AFTCSMP), which is interrupted by Wiener process, is investigated. First, the output feedback stochastic stabilization of the AFTCSMP with Wiener process subject to multiple failure processes is considered. The necessary and sufficient conditions, which are expressed in terms of a nonlinear matrix inequality feasibility problem, for the internal exponential stability in the mean square sense are obtained. To eliminate the nonlinearities, an LMI relaxation scheme is presented. In this way, a tractable sufficient condition is obtained. Based on this result, the output feedback H-infinity control of a continuous time Active Fault Tolerant Control Systems with Markovian Parameters (AFTCSMP) with Wiener process, which is expressed in terms of an LMI optimization problem, is studied by using the convex programming approach. Efficient numerical algorithms are thus developed for solving these problems.

In (Situ, 2005), a class of optimal control problems described by linear Ito stochastic differential equations driven by counting processes is considered and studied.

In (Teo, Ahmed 1974), a class of stochastic optimal control problems is considered, where the dynamical system is described by Ito stochastic differential equations driven by Wiener processes. It is shown that this class of stochastic optimal control problems is equivalent to a class of optimal control problems involving linear parabolic partial differential equations. However, numerical solution methods available in the literature (see, for example, (Huang, 2000) and (Wang, 2003)) for solving such deterministic optimal control problems with dynamics being described by partial differential equations are only applicable for small dimensional problems.

(Huang, Wang and Teo, 2000) proposed an approach to numerically solve the Hamilton-Jacobi-Bellman (HJB) equation governing a class of optimal feedback control problems. The approach is referred to the viscosity approximation to the HJB equation. They perturbed the first order HJB equation by adding a diffusion term with a small singular perturbation parameter. The time and spatial variables of the resulting equation can be discretized respectively by the Modified Method of Characteristic (MMOC) and a finite scheme. The usefulness of the method is demonstrated through solving numerical examples.

In (Wang, Jennings and Teo, 2003), a finite volume method is proposed to solve Hamilton- Jacobi-Bellman (HJB) equations which is obtained from a class of optimal feedback control problems. The method is based on a finite volume discretization in state space coupled with an upwind finite difference technique, and an implicit backward Euler finite differencing in time. This method is absolutely stable. It is shown that the system matrix of the resulting discrete equation is an M-matrix (Berman and Plemmons, 1994). Numerical experiments are carried out to test the effectiveness of the proposed method on problems with up to three states and two control variables. The results of the numerical experiments show that the method yields accurate approximate solutions.

The optimal parameter selection problems occur in many dynamic optimization models where the controls are restricted to be constant functions of time. Examples of these can be found in a number of parameter identification problems, controller parameter design problems, as well as economic and industrial management type problems. Furthermore, it plays a fundamental role in the numerical computation of optimal control problems. To be more specific, after the control parameterization (see (Teo, Goh and Wong, 1991)), all optimal control problems essentially are reduced to optimal parameter selection problems. Thus, the solvability of optimal parameter selection problem is crucial for generating numerical solution methods to many complex optimal control problems.

In (Ahmed and Georgenas, 1973) and (Teo, Ahmed 1974), respective necessary conditions for optimality are derived for deterministic and stochastic optimal parameter selection problems. Computational methods for solving deterministic optimal parameter selection problems are reported in (Dolezal, 1981), (Reid and Teo, 1980) and (Teo and Goh, 1989), where the system dynamic is described by ordinary differential equations in (Dolezal, 1981), and (Teo and Goh, 1989), while the system dynamic is described by parabolic partial differential equation in (Reid, 1980).

As a historical note, a pointwise (in time) maximum priciple was developed for non-singular optimal parameter selection problems in(Pontryagin, Boltyanskii, Gamkrelidze and Mishchenko, 1962). This result was, however, found to contain some errors in (Hofer and Sagirow, 1968). A correct version of the result may be found in (Ahmed and Georgenas, 1973) and (Boltyanskii, 1971).

In (Dolezal, 1981), necessary optimality conditions for are obtained for optimal parameter selection problems, where the dynamic system is described by ordinary differential equations with general boundary conditions.

In (Reid and Teo, 1980), an optimal parameter selection problem, where the dynamic system is described by parabolic partial ordinary differential equation with first boundary condition, is considered. The gradient formula of the cost function with respect to the parameter is derived. On the basis, a numerical solution method is proposed. In this numerical solution method, it is assumed that the solution of the parabolic partial ordinary differential equation is computable. This is, of course, only true for problems with small dimension.

In (Teo and Goh, 1989), a unified computational approach to the numerical solution of a class of combined optimal parameter selection and optimal control problems with general constraints is proposed. This approach is based on the control-parameterization technique, and is supported by regions convergent analysis. In this paper, it is also shown that several different classes of unconventional optimal control problems can be transformed into special cases of the problem. Four numerical examples are shown using the proposed approach.

Optimal filtering problems and optimal fusion problems can be formulated as specific stochastic optimal control problems. These problems have been extensively studied in the literature, see for example, (Basin, Sanchez and Martinez-Zuniga, 2007), (Feng, Teo, and Ahmed, 2006), (Tanikawa, 2006) and (Takeuchi, 2008). In particular, an optimal fusion problem is considered in (Feng, Teo, and Ahmed, 2006), where the measurement data are obtained from multiple sensors. It is shown that this optimal fusion problem is equivalent to a deterministic optimal control problem. Optimal filtering problems with multiple sensors are also considered in (Feng, Teo and Rehbock, 2008) and (Feng, Teo and Zhao, 2005).

In (Basin, Sanchez and Martinez-Zuniga 2007), the optimal filtering problem for linear systems described by stochastic Ito differential equations with multiple state and observation delays are considered, where the emphases are on the study of optimal estimate, error variance, and various error covariances. The paper deals with the most general case of multiple delays in both state and observation equations, which are allowed to be different from each other. In the general case, the determination of the filter gain matrix can be obtained through solving an infinite set of equations, which is clearly an impossible task. However, in a particular case of equal or commensurable delays in the observation and state equations, it suffices to solve a finite set of filtering equations. Through an example, the performance of the designed optimal filter for linear systems with state and observation delays is verified against the best Kalman-Bucy filter obtained for linear systems without delays.

In (Feng, Teo, Ahmed, Zhao and Yan, 2006), the question of optimal fusion of sensor data for Kalman filtering is considered. The basic problem is to design a linear filter whose output provides an unbiased minimum variance estimate of a signal process whose noisy measurements from multiple sensors are available for input to the filter. The problem is to assign weights to each of the sources (sensor data) dynamically so as to minimize estimation errors. The problem is formualted as an optimal control problem where the weight given to each of the sensor data is considered as one of the control variables satisfying certain constraints. There are as many controls as there are sensors. Using a time scaling transform, an efficient method is developed for determining the optimal fusion strategy. Some numerical results are presented for illustration.

In (Tanikawa, 2006), the fixed-point smoothing problem for discrete-time linear stochastic systems with unknown disturbances is considered. A new smoothing algorithm is obtained from the optimal filter with disturbance decoupling property for these systems. The algorithm is robust with respect to the unknown disturbances. In the absence of the unknown inputs, the algorithm reduces to the standard optimal smoother derived from the Kalman filter. Thus, it is clear that the new smoothing algorithm is a natural extension of the standard optimal smoother to the linear systems with unknown diaturbances.

In (Takeuchi, 2008), the concern is on an optimization problem of the gain matrix which is used in the linear observation for the Kalman filter. The optimal output feedback, which minimizes the power of the linearly encoded signal, is given by the least-squares estimate of the linear term. Then, the channel outputs become the innovations process which has the same structure as the model of parallel Gaussian channels with output feedback. The remaining task is the optimal selection of the gain matrix in the observation such that the estimation error is minimized.

In (Feng, Teo and Zhao, 2005), the question of sensor scheduling in discrete time is considered, where the basic problem is to design a linear filter such that its output provides an unbiased minimum variance estimate of a signal process. It is assumed that the noisy measurements of this signal process from multiple sensors are available for input to the filter. One source (sensor data) is to be selected dynamically so as to minimize estimation errors. This problem is formulated as an optimal control problem. For the calculation of the optimal scheduling strategy, a branch and bound method is developed based on the positive semi-definite property of the error covariance matrix.

In (Feng, Teo and Rebbock, 2008), a general class of optimal sensor scheduling problems in discrete time is considered, where there are sensors available for acquiring data so as to estimate the needed but unknown signal. It is assumed that only one of the sensors can be turned on at any moment and that different weights can be assigned to different sensors. This problem is formulated as a discrete time deterministic optimal control problem involving both discrete and continuous valued controls. A computational method is developed for solving this discrete time deterministic optimal control problem based on a branch and bound method in conjunction with a gradient-based method. The branch and bound method is used to determine the optimal schedule of sensors, where a sequence of lower bound dynamic systems is introduced so as to provide effective lower bounds for the construction of the branching rules. Each of the branches is an optimal weight vector assignment problem and a gradient-based method is developed for solving this optimal control problem. For illustration, two numerical examples are solved.

In (Goh and Teo, 1990), a class of optimal parameter selection problems, where the dynamics is described by a linear Ito stochastic differential equation with some controllable decision parameters appearing nonlinearly in the system. The aim is to minimize the expected value of a cost function subject to some probabilistic constraints on the state. It is first shown that this problem is equivalent to a deterministic optimal parameter selection problem subject to continuous state inequality constraints. The continuous state inequality constraints are then transformed into equivalent equality constraints using a constraint transcription given in (Teo and Goh, 1987). The paper then goes on to suggest that any standard gradient-based constrained optimization technique can be used to solve the transformed problem. However, as pointed out in Remark 6.6.5 of (Teo and Goh, 1991), the equality constraints obtained by this constraint transcription fail to satisfy any constraint qualification. Thus, the transformed problem cannot be solved effectively by using any standard gradient-based constrained optimization technique. In particular, the constraint violation cannot be avoided in the numerical computation if this constraint transcription is used.

In view of the serious deficiency of the constraint transcription introduced in (Teo and Goh, 1987), a computational procedure is proposed in (Jennings and Teo, 1990) to solve a class of functional inequality constrained optimization problems. In this paper, a novel constraint transcription is introduced to transform the functional inequality constraints into non-smooth equality constraints. Then, a local smoothing function is constructed to approximate these equality constraint functions as a sequence of conventional inequality constraints. These inequality constraints do not always fail to satisfy constraint qualification as for case of the constraint transcription introduced in (Teo and Goh, 1987). In this way, a sequence of conventional approximate optimization problems subject to inequality constraints is obtained. Each of these approximate problems can be considered as a conventional optimization problem, and hence is solvable by any standard gradient-based optimization technique. There are two parameters involved in the proposed approximation. One is to control the smoothness of the approximation, while the other controls the accuracy of the solution obtained. The main result of the paper ensures that, for any given smoothness parameter, if the accuracy parameter is chosen sufficiently small, then any feasible solution of the conventional approximate inequality constraints will satisfy the original functional inequality constraints.

Chapter 2

Nonparametric regression with jump points

2.1 Introduction

In this chapter, a two-stage method is presented for nonparametric regression with jump points. After the rough locations of all the possible jump points are identified by using an existing efficient kernel method, a smoothing spline function is used to approximate each segment of the regression function. A time scaling transformation is derived so as to map the yet to be decided jump points and the spline knots into fixed points. In this way, each of these approximate problems can be solved as an optimization problem by many existing optimization techniques. The method is applied to several examples. The results obtained show that the method is highly efficient.

The main reference for this chapter is (Wu and Chu, 1993; Craven and Wahba, 1979; Lee, Teo, Rehbock and Jennings, 1997; McQuarrie and Tsai, 1998; Schwarz, 1978; Shao and Campbell, 2002; Wu, Liu, Teo and Shao, to appear in 2008.).

2.2 Notations and Methodology

The following notations are used in this chapter. Let X be the response variable with respect to the variable t. For a given set of observations $\{X_{i,}\ t_i\}$, $i = 1,\ 2,\ \cdots,\ n$, the regression function is written as:

$$x_i = m\left(t_i\right) + \varepsilon_i, \quad i = 1, \cdots, n, \tag{2.2.1}$$

where m is an unknown function defined on $[0, T]$, $\{\varepsilon_i\}$, $i = 1, \cdots, n$, are independent and identically distributed $N(0, \sigma^2)$ normal random variables. For each $i = 1, ..., n$, ε_i represents the variation of x_i around $m\left(t_i\right)$. Without loss of generality, we can let $T = 1$. Otherwise, a re-scaling of T can be used. To form a segmented regression, we write $m(t)$ as:

$$m(t) = m_l(t), \text{ for } \tau_l{-}1 < t \leq \tau_l,\ l = 1, \cdots, N, \tag{2.2.2}$$

where $\tau_0 = 0$ and $\tau_N = 1$, where τ_i, $i = 1, \cdots, N-1$, are jump points. This setting covers the case of the change of mean values (step changes) as a special case by choosing $m_l(t)$ as constants. Note that the form of the regression $m\left(t\right)$ is usually unknown, and hence is nonparametric. It is well-known that spline functions are effective for approximating nonparametric regressions if they are smooth. However, the smoothing spline does not work well directly for regressions with jump points, since the smoothing spline is rather sensitive to the location of jump points. Some difficulties in jump point estimation have been demonstrated in (Julious, 2001) for linear regression with only one jump point. Intuitively, the jump points should be identified before applying the smoothing spline to each of the individual segments defined by the jump points. This is the motivation behind our proposed two-stage method.

The proposed method consists of two steps. First, we should locate all potential locations of jump points. Then, we use a time scaling transformation to transform the potential jump points and spline knots into pre-fixed points. By doing this, the parameter estimation using least squares becomes an optimization problem, which can be solved by many efficient optimization techniques, such as the sequential quadratic approximation optimization method. Finally, the modified Akaike's information criterion is used to determine which potential jump points are real jump points. We then obtain the final model.

2.3 Estimation for Potential Jump points

In statistics, the kernel method is an efficient and powerful statistical tool to detect jump points. Therefore, it is used in our two-stage method to find potential jumps. We assume that the following conditions are satisfied.

$(A1)$. The number of the observation points is sufficiently large to detect all the possible jump points. That is, the number of jump points is very small in comparison with the number of observation points and the number of observation points between consecutive jump points is sufficiently large for spline fitting.

$(A2)$. The jump points should be located in the interval $[\delta, 1-\delta]$, where δ is some small positive constant. That is, there is no jump point in the neighborhoods of the boundary points on which they become undetectable.

$(A3)$. There is at most one jump point between t_i and t_{i+1}, $i = 0, 1, \cdots, n$, where $t_0 = 0$ and $t_{n+1} = 1$.

$(A4)$. The regression errors ε_i, $i = 0, \cdots, n$, are independent and identically distributed (*i.i.d*) normal random variables with mean 0 and variance $\sigma^2 < \infty$.

Let K be the kernel function with the bandwidth h. There are two popular methods to construct kernel estimators, depending on the choice of weights by either direct kernel evaluation or the convolution of the kernel with a histogram obtained from the data. Each estimator has several important advantages and disadvantages.

A thorough review on this subject can be found in the paper by (Chu and Marron, 1991). In (Gijbels, et al, 1999), the first method is used to construct an estimator to detect the vicinity of the jump points. In our two-stage method, the second method is used to identify rough positions of the jump points.

For the kernel function K and the bandwidth h, the Gasser-Muller estimator is defined as:

$$\hat{m}(t) = \sum_{i=1}^{n} x_i \int_{s_{i-1}}^{s_i} K_h(t-\tau)\, d\tau, \text{ for } t \in (0,1), \tag{2.3.1}$$

where $K_h(\cdot) = h^{-1}K(\cdot/h)$, $s_0 = 0$, $s_i = (t_i + t_{i+1})/2$, $i = 2, \cdots, n-1$, and $s_n = 1$. Let $\hat{m}_1(t)$ and $\hat{m}_2(t)$ be the two Gasser-Muller estimators obtained with the kernel functions K_1 and K_2, respectively, using the same bandwidth h. Let

$$J(t) = \hat{m}_1(t) - \hat{m}_2(t). \tag{2.3.2}$$

To continue, we need to analyze the value of $|J(t)|$.

If $m(t)$ has no jump point, then, under the usual regularity conditions, $\hat{m}_1(t)$ and $\hat{m}_2(t)$ are uniformly strongly consistent estimators of $m(t)$. Thus, the variation of $|J(t)|$ would not be obvious. If $m(t)$ has a jump point, then the value of $|J(t)|$ would have an obvious change around the jump point. In Figure 2.1, we give an example where $m(t)$ has a jump at $t = 1/2$. Write

$$m(t) = \varphi(t) + \psi(t) \tag{2.3.3}$$

where $\varphi(t)$ is the continuous part of $m(t)$, while $\psi(t)$ is a step function to characterize the jumps of $m(t)$. The magnitude of $|J(t)|$ corresponding to $\varphi(t)$ is of small order. However, the magnitude corresponding to $\psi(t)$ is symmetric and convex downward in the neighborhood of the jump point if the kernel functions K_1 and K_2 are chosen such that

$$K_1(t) = K_2(-t). \tag{2.3.4}$$

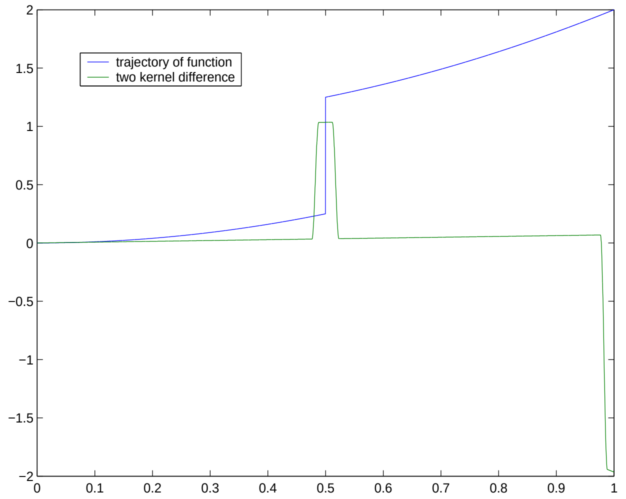

Figure 2.1: the magnitudes of $|J(t)|$ and the regression function $m(t)$.

Furthermore, if K_1, K_2 have compact support in $[-1,1]$, then the widths of the neighborhoods mentioned above are no more than $2h$. Based on the above discussions, we can give a numerical procedure to estimate rough locations (and hence, number) of the jump points.

Algorithm 2.3.1 (Kernel method to detect the potential jump points)**.**

Step 1: *Choose h such that $h = O(n^{-1/3})$.*

Step 2: *Choose a nonnegative function $K(t)$ with a compact supported in $[-1,1]$ and is such that $\int_0^1 K \neq \int_{-1}^0 K$.*

Step 3: *Let $K_1(t) = K(t)$ and $K_2(t) = K(-t)$. Calculate $J(t)$ by using (2.3.1) and (2.3.2), where h is given in Step 1.*

Step 4: *Find all the points, which correspond to local maxima of $|J(t)|$.*

The points obtained by Algorithm 2.3.1 are considered as potential jump points due to Step 4. In the next section, we will use a time scaling transformation to find the accurate positions of these jump points.

2.4 Segmented Regression with Constraints

Suppose that $\{\tau_1, \tau_2, \cdots, \tau_{N-1}\}$ is a set of potential jump points (obtained from the kernel method crudely). Then, the interval $[0, 1]$ has been partitioned into N subintervals $[\tau_{l-1}, \tau_l]$, $l = 1, \cdots, N$. For each $l = 1, \cdots, N$, let the observation points contained in the subinterval $[\tau_{l-1}, \tau_l]$ be denoted by $t_{l,1,} \cdots, t_{l,N_l}$ and let $x_{l,1,} \cdots, x_{l,N_l}$ be the corresponding observations. The regression function is denoted by $m_l(t)$ in the subinterval $(\tau_{l-1}, \tau_l]$, where $l = 1, \cdots, N$, $\tau_0 = 0$ and $\tau_N = T$. Now, we use the segmented regression to fit the segment $m_l(t)\, for$ $l = 1, \cdots, N$.

The most widely used approach to curve fitting is least squares. If we place no restrictions on the residual sum of squares, this method is, in fact, an interpolation which may exhibit fluctuation. To avoid this, we append a smoothness requirement in the cost function. In this chapter, we will introduce the cubic spline for fitting the segment $m_l(t)$.

A general cubic spline basis is defined as

$$\left\{1, t, t^2, t^3, (t-\theta_1)_+^3, \cdots, (t-\theta_K)_+^3\right\}. \tag{2.4.1}$$

where $(t-\theta_k)_+ = \max\{0, t-\theta_k\}$, θ_k, $k = 1, \cdots, K$, are the knot points. Since the smoothness requirement is incorporated in the cost, we need the estimator $\hat{m}_l(t)$ of $m_l(t)$ to be continuously differentiable at the knot θ_k, $k \in \{1, \cdots, K\}$. Write

$$\hat{m}_l(t) = a_{l,1} + \sum_{k=1}^{3} a_{l,k+1} t^k + \sum_{k=1}^{i_l} a_{l,k+4}(t-\theta_{l,k})_+^3, \tag{2.4.2}$$

where $\theta_{l,1}, \cdots, \theta_{l,i_l}$, are the i_l pre-fixed knots contained in the $l-th$ segment, $a_{l,1}, \cdots, a_{l,i_l+4}$, are the coefficients. Note that $(t-\theta_k)_+ = \max\{0, t-\theta_k\}$ and $(t-\theta_k)_+^3$ is a polynomial of degree three. Thus $(t-\theta_k)_+^3$ is twice continuously differentiable. Therefore, there are no further restrictions to be imposed on $\hat{m}_l(t)$.

Define the cost function as:

$$J(\boldsymbol{\tau}, \mathbf{a}) = \sum_{l=1}^{N} \sum_{i=1}^{N_l} \left(\hat{m}_l(t_{l,i}) - x_{l,i}\right)^2 + \lambda \sum_{i=1}^{N} \int_{\tau_{i-1}}^{\tau_i} \left(\hat{m}_i''(t)\right)^2 dt, \tag{2.4.3}$$

where $\tau = [\tau_1, \tau_2, \cdots, \tau_{N-1}]^\intercal$, $\mathbf{a} = [a_{1,1}, a_{1,2}, \cdots, a_{1,i_1+4}, \cdots, a_{N,i_N+4}]^\intercal$, while λ is the smoothness penalty parameter.

Our objective is to find a $(\tau, \mathbf{a})$ such that (2.4.3) is minimized subject to the following constraints

$$t_{l,N_l} \leq \tau_l \leq t_{l+1,1}, l = 1, ..., N-1. \tag{2.4.4}$$

For this optimization problem, the estimates of the jump locations and the optimal regression coefficients are obtained simultaneously. Let this problem be referred to as Problem $(2.P)$.

For the penalized parameter λ, it could be chosen it by the generalized cross-validation method (Craven and Wahba, 1979). Here it will be chosen interactively.

To solve Problem (2.P), we need to compute the cost. We note that the cost (2.4.3) is composed of two parts. Since the first part is only related to the coefficient vector $\mathbf{a}$, it is easily computed. The second part is the sum of some integrals with the limits of the integrations being related to the jump points. The computation of this part of the cost is difficult. Clearly, its gradient will be even harder to compute. To overcome this difficulty, we will introduce a time scaling transform. By this transform, the jump points and the spline knots are all mapped into some pre-fixed points.

2.5 A Time Scaling Transformation Method

We suppose that $\tau_1, \cdots, \tau_{N-1}$, are $N-1$ variable times in the time interval $(0, T)$. A time scaling transform is introduced such that the variable times $\tau_1, \cdots, \tau_N$, are sequentially transformed into pre-fixed times $\xi_1, \cdots, \xi_N$ in a new time scale. This transform was called the control parameterization enhancing transform in (Lee, Teo, Rehbock and Jennings, 1997). It was introduced to overcome the numerical difficulties in the computation of optimal control problems. This time scaling transform is defined by

$$\frac{dt\left(s\right)}{ds} = v\left(s\right), \tag{2.5.1}$$

with initial condition

$$t\left(0\right) = 0, \tag{2.5.2}$$

where $v\left(s\right)$ satisfies the following conditions:

- $v\left(s\right) \geq 0$ for all s;
- $v\left(s\right)$ is piecewise constant on the interval $\left(\xi_{i-1}, \xi_i\right]$;
- $$\int_{\xi_{i-1}}^{\xi_i} v\left(s\right) ds = \tau_i - \tau_{i-1},\ i = 0, 1, \cdots, N, \tag{2.5.3}$$
  where $\tau_0 = 0$, $\tau_N = T$, $\xi_0 = 0.$ and $\xi_N = 1$.

Now we apply this transform to our problem such that the jump points

$$\tau_1, \tau_2, \cdots, \tau_{N-1}$$

are sequentially mapped into the fixed points

$$\xi_1, \cdots, \xi_{N-1}$$

and the spline knots

$$\theta_{l,1}, \theta_{l,2} \cdots, \theta_{l,i_l}$$

are sequentially mapped into

$$l + \frac{1}{i_l + 1}, l + \frac{2}{i_l + 1}, \cdots, l + \frac{i_l}{i_l + 1},$$

where $l = 1, \cdots, N$.

To achieve it, we choose

$$v(s) = \sum_{k=1}^{N} \sum_{j=1}^{i_k+1} \xi_{k,j} \varkappa_{\left[k+\frac{j-1}{i_k+1},\, k+\frac{j}{i_k+1}\right.} \quad ,$$

where $\varkappa_I$ is the indictator function defined by

$$\varkappa_I(s) = \begin{cases} 1, \text{ if } s \in I \\ 0, \text{ otherwise,} \end{cases}$$

$\xi_{k,j}$, $k = 1, \cdots, N$; $j = 1, \cdots, i_k$, satisfy the following conditions:

$$\xi_{k,j} \geq 0;\ k = 1, \cdots, N,\ j = 1, \cdots, i_k + 1; \tag{2.5.4}$$

$$\sum_{k=1}^{N} \sum_{j=1}^{i_k+1} \xi_{k,j} = T; \tag{2.5.5}$$

$$\xi_{k,j} = (i_k + 1)(\theta_{k,j} - \theta_{k,j-1}),\ k = 1, \cdots, N;\ j = 2, \cdots, i_k; \tag{2.5.6}$$

$$\frac{1}{i_k + 1} \xi_{k,i_k+1} + \frac{1}{i_{k+1} + 1} \xi_{k+1,1} = \theta_{k+1,1} - \theta_{k,i_k},\ k = 1, \cdots, N. \tag{2.5.7}$$

By this transform, we obtain

$$\sum_{i=1}^{N} \int_{\tau_{i-1}}^{\tau_i} \left(\hat{m}_l''(t)\right)^2 dt = \sum_{i=1}^{N} \int_{i-1}^{i} \left(\hat{m}_i''(t(s))\right)^2 v(s)\, ds. \tag{2.5.8}$$

Thus cost function (2.4.3) is equivalent to the transformed cost function given below.

$$\bar{J}(\boldsymbol{\xi}, \mathbf{a}) = \sum_{l=1}^{N} \sum_{i=1}^{N_l} \left(\hat{m}_l(t_{l,i}) - x_{l,i}\right)^2 + \lambda \sum_{i=1}^{N} \int_{i-1}^{i} \left(\hat{m}_i''(t(s))\right)^2 v(s)\, ds, \tag{2.5.9}$$

where $\boldsymbol{\xi} = [\xi_{1,1}, \xi_{1,2}, \cdots, \xi_{1,i_1+1}, \cdots, \xi_N, i_{N+1}]^\intercal$. We have the following optimization problem.

Problem $(2.\hat{P})$**.** $\min \bar{J}(\boldsymbol{\xi}, \mathbf{a})$ *subject to (2.5.4), (2.5.5), (2.5.6) and (2.5.7).*

The conclusions of the following theorem are clear.

Theorem 2.5.1. *Problem (2.P) is equivalent to Problem* $(2.\hat{P})$ *in the sense that* $(\boldsymbol{\tau}, \mathbf{a})$ *is the optimal solution of Problem (2.P) if and only if* $(\boldsymbol{\xi}, \mathbf{a})$ *is an optimal solution of Problem* $(2.\hat{P})$*. Furthermore, they have the same optimal cost value.*

2.6 Model Selection

Note that the kernel method presented in Section 2.3 tries to detect all the potential jump points. The spline approximation method outlined in Section 2.4 can be used in conjunction with the time scaling transform introduced in Section 2.5 to fit a smooth curve to each regression segment between successive jump points so as to achieve optimum fitting of the unknown regression with jumps. Suppose that P is the number of candidates jump points. Then, the number P is required to be chosen appropriately. Several methods for choosing it have been suggested in the literature. We propose to use the Akaike information criterion (AIC) (Akaike, 1974) for the purpose of model selection. In (Tong, 1980), AIC is applied for model selection in a threshold autoregression. In (Hurvich and Tsai, 1989), a modified Akaike information criterion AIC$_c$ is used in the case of small samples. It is forced that the AIC$_c$ dramatically reduces the bias and hence leads to the improvement of the model selection. The AIC$_c$ penalizes the RSS by a function of the number of free parameters. In (McQuarrie and Tsai, 1998) the AIC$_c$ is defined as

$$AIC_c = \ln \frac{RSS}{N} + \frac{N+P}{N-P-2}, \tag{2.6.1}$$

where N is the number of observations, P is the number of free parameters in the model and RSS (residual sum of squares) is the first part of the cost (2.4.3). Note that AIC tends to overestimate the number of parameters. Thus, we will use Bayesian information criterion (BIC) (Schwarz, 1978; McQuarrie and Tsai, 1998) as a criterion instead of AIC when the number of observation points is large enough, where

$$BIC = N \ln \frac{RSS}{N} + P \ln(N) \tag{2.6.2}$$

In our simulation study, for those examples, where the data are generated from a mathematical function, BIC will be used as the criterion as the data can be collected as much as we need. For the real data, we will use AIC$_c$ as the criterion since the observation points are limited.

Algorithm 2.6.1.

Step 1: *Set $\boldsymbol{\tau} = \{\tau_1, \tau_2, \cdots, \tau_{N-1}\}$, which is a set of candidate jump points. Let Γ be the updated solution set and initially Γ is chosen as an empty set. $AIC_c = 1000$.*

Step 2: *Choose candidate jump points $\{\tau_i\}_{i=1}^{m} \subset \boldsymbol{\tau}$ and for given spline knots $\boldsymbol{\theta}$, use the time scaling transform (2.5.1) to convert the corresponding Problem $(2.P)$ into its equivalent Problem $(2.\hat{P})$. Solve Problem $(2.\hat{P})$ and evaluate the corresponding AIC_c' $($or $BIC)$. Let Γ' denote the current solution set.*

Step 3: *If $AIC_c' \leq AIC_c$, then set $\Gamma = \Gamma'$, $AIC_c = AIC_c'$ and goto Step 2. If there is no possible candidate jump points, then stop.*

2.7 Reverse Transform

Since the time scaling transform given in Section 2.5 is a one-to-one mapping, we can find the reverse transform to map the estimated jump points back to the original axis, t axis. The reverse transform is given by

$$\tau_i = \sum_{j=1}^{i} \xi_j, \ i = 1, \cdots, N-1. \tag{2.7.1}$$

2.8 Numerical Examples

To test the performance of our proposed method, some numerical examples are presented here. All calculations are carried out within the Matlab environment.

For the simulated study, the data were generated by the following equation:

$$x_i = m(t_i) + \varepsilon_i, \quad i = 1, \cdots, n \tag{2.8.1}$$

with $t_i = i/n$ and ε_i sampled randomly from a normal distribution with standard deviation σ_ε. In the next numerical examples, we choose $\lambda = 0.1$.

Example 2.8.1. Let $n = 200$, $\sigma_\varepsilon = 0.2$ and the data is produced from (2.8.1) with

$$m(t) = 2 - 2\left|t - 0.26\right|^{1/5} 1\left(t \le 0.26\right) - 2\left|t - 0.26\right|^{3/5} 1\left(t > 0.26\right) + 1\left(t \ge 0.78\right). \tag{2.8.2}$$

First, we use Algorithm 2.3.1 to detect rough locations of the jump points. It will be good if the locations of these jump points are given. Let the spline knots vector be $[0.1, 0.15, 0.3, 0.5, 0.6]^\intercal$, and let $\lambda = 0.01$. We use BIC as criterion. The obtained results are: 0.25225, 0.78178 which are the two jump points. The corresponding BIC obtained is -4.1056. The result is depicted in Figure 2.2. We do the same simulation for 100 times. The results are depicted in Figure 2.3.

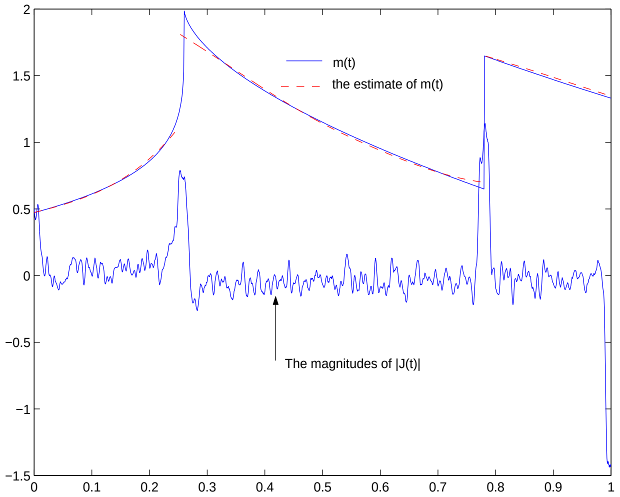

Figure 2.2: Results obtained for Example 2.8.1 with $\lambda = 0.01$.

Now, we choose $\lambda = 10$, and depict the obtained result in Figure 2.4. From Figure 2.4, we can see that if λ is large enough, then the fitting becomes linear fitting. Thus, the parameter λ controls the gradient change rate of the splines.

Example 2.8.2. We now apply our method to a real example, which contains the wa-

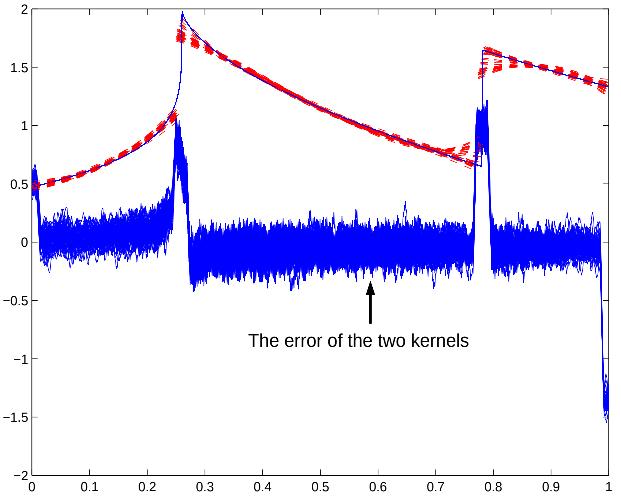

Figure 2.3: Results obtained (100 times simulation) for Example 2.8.1 with $\lambda = 0.01$.

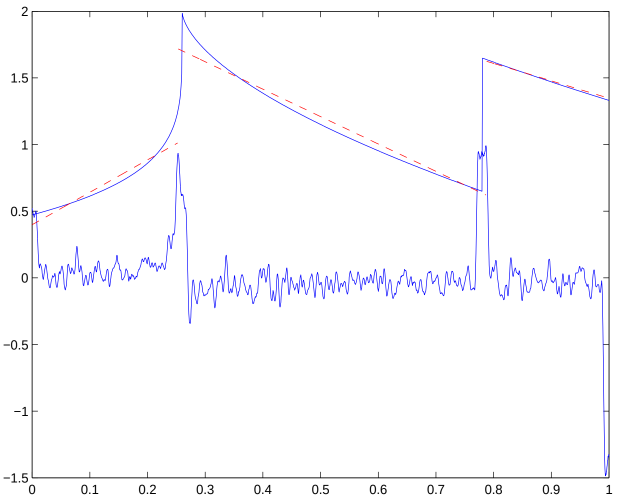

Figure 2.4: Results obtained for Example 2.8.1 with $\lambda = 10$.

ter levels (depths) in boreholes monitored irregularly over time. The data are taken from the database of Agriculture of Western Australia and have been analyzed in (Shao and Campbell, 2002). There are 49 observations. The observation points as well as its jumps estimated by Algorithm 2.3.1 is depicted in Figure 2.5. We can see that the kernel method cannot present an good jump estimator since the number of observation points is too little from the figure. Thus, we have to consider all the potential jump points and then use the AIC$_C$ as the criterion to choose the best model. In the process of fitting, we suppose that all of the jumps are positive since the level of the groundwater does not experience drop instantly. We re-scale the time to the new interval $[0, 1]$. The final model obtained 3 jump points: $0.1050, 0.2727, 0.4554$. AIC$_C = -4.3148$. The obtained results are depicted in Figure 2.6.

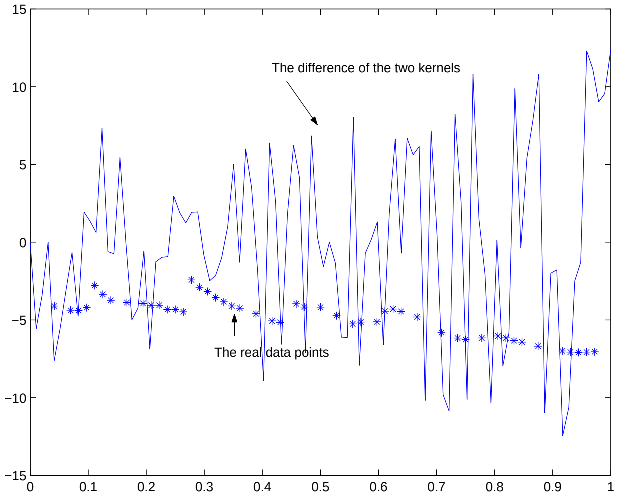

Figure 2.5: Observation points and the estimations of the jump points of Example 2.8.2.

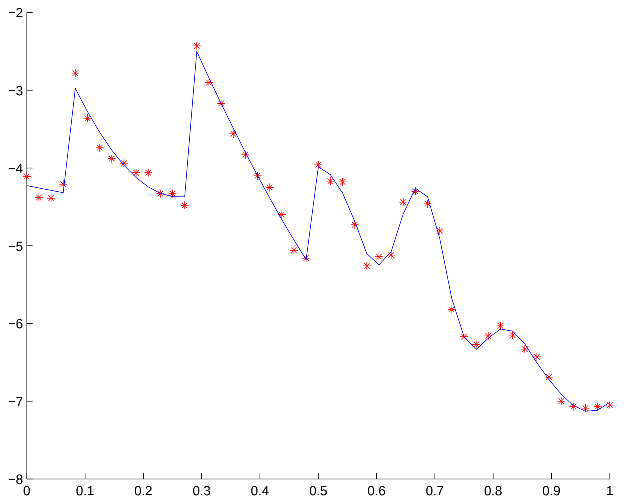

Figure 2.6: The fitting curve for Example 2.8.2.

Chapter 3

Nonparametric regression with jump curves

3.1 Introduction

A two-stage method is proposed to construct an approximating surface with jump location curve from a set of observed data which are corrupted with noise. In the first stage, a nonparametric kernel method is used to detect an estimate of the jump location curve in a surface. In the second stage, we introduce a space scaling transform to shift the jump location curve into a row pixels or column pixels, which we assume, without loss of generality, is a row pixels. The shifted region is then divided into two disjoint subregions by the jump location row pixels. These subregions are expanded to two overlapping expanded subregions, each of which includes the jump location row pixels. The Lagrangian interpolation is used to calculate artificial values at these newly added pixels by using the observed data. Then, the tensor product cubic splines are constructed using the least square fitting method to approximate the surface on each expanded subregions in which the artificial values at the pixels in the jump location row pixels for each expanded subregion created by using the Lagrangian interpolation are used in the fitting process with a penalty on the difference between the two surfaces across the jump location row pixels. The inclusion of these artificial values assimilates the use of curvature information of the tensor product cubic splines during the fitting process. The curve with minimal distance between the two surfaces is chosen as the curve dividing the region. Subsequently, two nonoverlapping tensor product cubic spline surfaces are obtained. Now, by carrying out the inverse space transformation, the two fitted smooth surfaces in the original space are obtained. For illustration, a numerical example is solved using the method proposed.

The main reference for this chapter is (Qiu, 2002; Rade and Westergren, 1998; Tam, Kurbatskii, 2000; Boor, 2001; Dierckx, 1993; Wahba, 1990; Duchon, 1977; Liu, Wu, Teo and Shao submitted.).

3.2 Statement of Problem

Consider a real-valued surface function $f(x, y)$ defined on $\Omega = [0, 1] \times [0, 1]$. We assume that it contains a unique jump location curve, $y = J(x)$ or $x = J(y)$. The jump location curve divides the region Ω into two connected regions Ω_1 and Ω_2 such that $\Omega_1 \cap \Omega_2 = \varnothing$ and $\Omega_1 \cup \Omega_2 = \Omega$.

Let the surface function $f(x, y)$ be written as:

$$f(x, y) = g(x, y) + h(x, y)I(x, y), \tag{3.2.1a}$$

where

$$I(x, y) = \begin{cases} 1, & x \geq J(y) \quad \text{or} \quad y \geq J(x) \\ 0, & \qquad \text{otherwise,} \end{cases} \tag{3.2.1b}$$

while $g(x, y)$ and $h(x, y)$ are continuously differentiable functions on Ω for which $h(x, y)$ denotes the jump magnitude.

Let

$$\{(x_i, y_i) = (i/n_1, i/n_1) : i = 1, 2, ..., n_1\} \tag{3.2.2}$$

be the set of the observation points in Ω. Suppose that a set of observation data is obtained from the regression function (3.2.1a) and (3.2.1b), which is corrupted by a Gaussian noise, at the set of the observation points defined by (3.2.2). More specifically, we consider the regression function defined by (3.2.1a) and (3.2.1b), which is interrupted by a Gaussian noise. We assume that the observation data (x_i, y_i, z_i), $i = 1, \cdots, n$, are collected at the observation points (x_i, y_i), $i = 1, \cdots, n$, as follows.

$$z_i = f(x_i, y_i) + \varepsilon_i, \quad i = 1, ..., n; \quad x_i, y_i \in [0, 1], \tag{3.2.3}$$

where $\{z_i\}$ are the estimated magnitudes of the surface collected from the noisy observation at the observation points $\{(x_i, y_i)\} \subset \Omega$, where $\Omega \subset R^2$. The bivariate regression function $f(x, y)$ is continuous everywhere except at the JLC, and $\{\varepsilon_i\}$ are independently identically distributed Gaussian noise with mean 0 and variance σ^2.

3.3 The Jump Detection Procedure

In this section, we shall review a method which can be used to detect a rough location of the JLC. This method, which is due to (Qiu, 2002), uses two estimators $M_n^{(1)}(x, y)$, $M_n^{(2)}(x, y)$ and the threshold U_n to detect jumps, where

$$M_n^{(1)}(x, y) = \frac{1}{nh_np_n} \sum_{i=1}^{n} z_i [K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) - K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n})], \tag{3.3.1}$$

$$M_n^{(2)}(x, y) = \frac{1}{nh_np_n} \sum_{i=1}^{n} z_i [K_2(\frac{y_i - y}{h_n}, \frac{x_i - x}{p_n}) - K_1(\frac{y_i - y}{h_n}, \frac{x_i - x}{p_n})], \tag{3.3.2}$$

and

$$U_n = \sqrt{\frac{\sigma^2\chi^2_{1,\alpha_n/2}}{nh_np_n}[\int_{-1/2}^{1/2}\int_0^1 (K_2(x,y))^2dxdy + \int_{-1/2}^{1/2}\int_{-1}^0 (K_1(x,y))^2dxdy].} \tag{3.3.3}$$

Here,

$$K_2(x,y) = 12/11(1-x^2)(12/11(1-(y-0.5)^2)I_{[-1/2,1/2]\times[0,1]}(x,y), \tag{3.3.4a}$$

$$K_1(x,y) = K_2(x,-y), \tag{3.3.4b}$$

$h_n = k_1/n_1$ and $p_n = k_2/n_1$, where k_1 and k_2 are two positive odd integer numbers.

Clearly, the sample size is $n = (n_1)^2$. Furthermore, we note that the estimators $M_n^{(1)}(x,y)$ and $M_n^{(2)}(x,y)$ are used to detect jumps which are parallel to the x-axis and the y-axis, respectively. For each $j = 1, 2$, let

$$\hat{D}_j = \{(x_i, y_i) : M_n^{(j)}(x_i, y_i) > U_n\}$$

be the set of points which is an estimate of the JLC.

The following is an algorithm for finding an estimate of the JLCs.

Algorithm 3.3.1.

Step 1: Calculate the quantities of $M_n^{(1)}(x_i, y_i)$ and $M_n^{(2)}(x_i, y_i)$.

Step 2: Calculate quantities U_n.

Step 3: Compare $M_n^{(1)}(x_i, y_i)$ with U_n for all points (x_i, y_i).

Step 4: Compare $M_n^{(2)}(x_i, y_i)$ with U_n for all points (x_i, y_i).

Step 5: If $M_n^{(1)}(x_i, y_i) > U_n$, add (x_i, y_i) up to set $\hat{D}_1$.

Step 6: If $M_n^{(2)}(x_i, y_i) > U_n$, add (x_i, y_i) up to set $\hat{D}_2$.

From Algorithm 3.3.1, we collect all those points (x_i, y_i) in each of $\hat{D}_j$, $j = 1, 2$. From which, we can determine easily either $\hat{D}_1$ (i.e., the jumps are parallel to the $x-$axis) or $\hat{D}_2$ (i.e., the jumps are parallel to the $y-$axis) is to be used as an estimate of the JLCs.

Two modification procedures are reported in (Qiu, 2002) for making the detected JLC thinner by deleting scattered jump candidates.

3.4 Space Scaling Transformation

Let the observation points be denoted by $\{x_i,\ y_j\}$ and let the corresponding values of the surface at these observation points be denoted by $z_{i,j}$, $i = 1, ..., n_1$; $j = 1, \cdots, n_2$. Suppose that the observation points are regularly distributed. That is,

$$x_i = x_0 + ih_1,\ i = 1, \cdots, n_1,$$

and

$$y_j = y_0 + jh_2,\ j = 1, \cdots, n_2,$$

where $x_0 = y_0 = 0$.

From Algorithm 3.3.1, we obtain an estimate of the JLC. Without any loss of generality, we assume that the estimate obtained is parallel to the $x-$axis (*i.e.*, $\hat{D}_1$). It divides the space Ω into two connected but disjoint regions Ω_1 and Ω_2. Clearly, $\hat{D}_1$ is irregular in shape, and therefore Ω_1 and Ω_2 are also irregular. Thus, it is hard to fit this surface by the existing available method. To overcome this difficulty, we will introduce a space scaling transform which maps the jump curve into a parallel line. By this transformation, the two original irregular subregions Ω_1 and Ω_2 are mapped into two new subregions and the jump curve is located in the boundary lines of these two new subregions.

Since we assume that the JLC is parallel to the x-axis, we use $y = \hat{J}(x)$ to denote the detected JLC. Let $\min\limits_{y} \hat{D}_1$ and $\max\limits_{y} \hat{D}_1$ denote the lower and upper bounds of the coordinate y of the estimated JLC, respectively. Clearly, we can find an integer $\hat{n}$ such that

$$\min_{y} \hat{D}_1 + \hat{n}h_2 \leq \max_{y} \hat{D}_1 \leq \min_{y} \hat{D}_1 + (\hat{n}+1)\, h_2.$$

Thus,

$$\min_{y} \hat{D}_1 \leq y = \hat{J}(x) \leq \min_{y} \hat{D}_1 + (\hat{n}+1)\, h_2.$$

Let $y = \hat{J}_s(x)$ denote the shifted JLC. For each fixed x_i, where $i = 1, ..., n_1$, we shift the column pixels $f\,(x_i, y)\,,\; i = 1, ..., n_1$, such that the JLC is located between the two parallel lines

$$y = \min_{y} \hat{D}_1$$

and

$$y = \min_{y} \hat{D}_1 + h_2.$$

That is,

$$\min_{y} \hat{D}_1 \leq y = \hat{J}_s(x) \leq \min_{y} \hat{D}_1 + h_2.$$

By this transformation, the original region has been divided into two subregions G_1 and G_2, where

$$G_1 = \{(x, y) : 0 \leq x \leq 1, \;\; -(\hat{n}+1)\, h_2 \leq y \leq \min_{y} \hat{D}_1\}$$

and

$$G_2 = \{(x, y) : 0 \leq x \leq 1, \;\; \min_{y} \hat{D}_1 + h_2 \leq y \leq 1\}.$$

If the jump location curve is parallel to the y-axis, we use $x = \bar{J}(y)$ to denote the detected JLC. In a similar way, we can define $\min\limits_{x} \hat{D}_2$, $\max_x \hat{D}_2$ and the corresponding integer $\bar{n}$ such that

$$\min_x \hat{D}_2 + \bar{n}h_1 \leq \max_x \hat{D}_2 \leq \min_x \hat{D}_2 + (\bar{n}+1)\, h_1.$$

Thus,

$$\min_x \hat{D}_2 \leq x = \bar{J}(y) \leq \min_x \hat{D}_2 + (\bar{n}+1)\, h_1.$$

Let $x = \bar{J}_s(y)$ denote the shifted JLC. We shift the row pixels $f\,(x, y_i)$ such that the shifted JLC is located between the two parallel lines

$$x = \min_x \hat{D}_2$$

and

$$x = \min_x \hat{D}_2 + h_1.$$

That is,

$$\min_x \hat{D}_2 \leq x = \bar{J}_s(y) \leq \min_x \hat{D}_2 + h_1.$$

The row pixels divide the region into two disjoint rectangles G_3 and G_4, where

$$G_3 = \{(x,y) : -\,(\bar{n}+1)\, h_1 \leq x \leq \min_x \hat{D}_2,\ 0 \leq y \leq 1\}$$

and

$$G_4 = \{(x,y) : \min_x \hat{D}_2 + h_1 \leq x \leq 1,\ 0 \leq y \leq 1\}.$$

Let

$$\theta_1 = \min_y \hat{D}_1.$$

We now introduce a space scaling transform with which the shifted JLC is located between the lines $y = \theta_1$ and $y = \theta_1 + h_2$. It is presented in the following as a theorem.

Theorem 3.4.1. *Let the detected JLC be denoted by $y = \hat{J}(x)$ and let the column pixels be shifted by a one-to-one mapping defined below.*

$$\widetilde{x} = x, \tag{3.4.1a}$$

$$\widetilde{y} = y - (\hat{y}_i - \min_y \hat{D}_1), \textit{for } x \in [x_i, x_{i+1}) \tag{3.4.1b}$$

where $\hat{y}_i$ is chosen such that

$$\hat{y}_i - h_2 \leq \hat{J}(x_i) \leq \hat{y}_i.$$

Then, after the one-to-one mapping, the shifted JLC becomes the row pixels given by

$$[0, 1] \times [\min_y \hat{D}_1, \min_y \hat{D}_1 + h_2].$$

Let

$$\theta_2 = \min_x \hat{D}_2.$$

Similarly, the space scaling transform with which the shifted JLC is located between the lines $x = \theta_2$ and $x = \theta_2 + h_2$ is presented in the following as a theorem.

Theorem 3.4.2. *Let the detected JLC be denoted by $x = \bar{J}(y)$ and let the row pixels be shifted by a one-to-one mapping defined below.*

$$\widetilde{x} = x - (\bar{x}_i - \min_x \hat{D}_2), \ \textit{ for } y \in [y_i, y_{i+1}) \tag{3.4.2a}$$

$$\widetilde{y} = y, \tag{3.4.2b}$$

where $\bar{x}_i$ is chosen such that

$$\bar{x}_i - h_1 \leq \bar{J}(y_i) \leq \bar{x}_i.$$

Then, after the one-to-one mapping, the JLC becomes the column pixels given by

$$[\min_x \hat{D}_2, \min_x \hat{D}_2 + h_1] \times [0, 1].$$

In any practical situation, the estimate of the JLC is either parallel to the $x-$axis or to the $y-$axis. Thus, either Theorem 3.4.1 or Theorem 3.4.2 is used. Without any loss of generality, we consider the situation, where Theorem 3.4.1 is to be applied. Then, the original space after shifting is divided into two disjoint subregions, denoted by G_1 and G_2. On each of these subregions, the surface is smooth, and hence can be approximated by product splines using surface fitting methods. In this chapter, we shall develop a new surface fitting method using a least square criterion, where the estimation of the JLC forms part of the surface fitting process of the two continuous surfaces. This is achieved by formulating this surface fitting problem as an optimization problem, and hence it is solvable by any efficient optimization techniques, such as quasi-Newton methods or conjugate gradient method. Before using the optimization method, the subregions G_1 and G_2 are expanded to $\tilde{G}_1$ and $\tilde{G}_2$, respectively, where

$$\tilde{G}_1 = \{(x, y) : 0 \leq x \leq 1, \ -(\hat{n}+1)\, h_2 \leq y \leq \min_y \hat{D}_1 + h_2\}$$

and

$$\tilde{G}_2 = \{(x, y) : 0 \leq x \leq 1, \ \min_y \hat{D}_1 \leq y \leq 1\}.$$

When we consider the surface fitting on the expanded subregions $\tilde{G}_1$ and $\tilde{G}_2$, we shall use the Lagrangian extrapolation method to generate some artificial observation values in the overlapping area of $\tilde{G}_1$ and $\tilde{G}_2$ in next section.

If the estimate of the JLC is parallel to the $y-$axis (*i.e.*, Theorem 3.4.2 is to be applied), the original space, after shifting, is divided by two disjoint subregions G_3 and G_4. Similarly, we obtain the two corresponding expanded subregions $\tilde{G}_3$ and $\tilde{G}_4$, as well as the corresponding optimization problem.

3.5 Lagrangian extrapolation

In this section, we shall show how the required artificial points in the overlapping area of $\tilde{G}_1$ and $\tilde{G}_2$ are to be obtained by using the cubic interpolation polynomial method. The introduction of these artificial points is to ensure that the resulting product spline approximating surface obtained is achieved as if the curvature of the product splines are having been incorporated in the surface fitting process.

Consider the situation where the estimated JLC is parallel to $x-$axis. Suppose that the space scaling transform has been applied, leading to the two disjoint subregions G_1 and G_2 and that these two subregions have been expanded to $\tilde{G}_1$ and $\tilde{G}_2$, respectively. These two expanded subregions are overlapping in the boundary of the estimated JLC.

Let the three points closest to the estimated JLC in G_1 be denoted by

$$\left(x_i, \bar{y}_{i,1}^-\right), \quad \left(x_i, \bar{y}_{i,2}^-\right), \text{ and } \quad \left(x_i, \bar{y}_{i,3}^-\right).$$

Similarly, let the three points closest to the estimated JLC in G_2 be denoted by

$$\left(x_i, \bar{y}_{i,1}^+\right), \quad \left(x_i, \bar{y}_{i,2}^+\right), \text{ and } \left(x_i, \bar{y}_{i,3}^+\right).$$

We will use the observed values at

$$\left(x_i, \bar{y}_{i,1}^-\right), \quad \left(x_i, \bar{y}_{i,2}^-\right), \text{ and } \left(x_i, \bar{y}_{i,3}^-\right)$$

to produce an artificial value at $\left(x_i, \bar{y}_{i,1}^+\right)$. This can be done as follows.

Let

$$f\left(x_i, y\right) = \sum_{k=1}^{3} c_{i,k} Q\left(\frac{y - y_k}{h_2} - k\right). \tag{3.5.1}$$

We substitute the left of (3.5.1) by the corresponding values of $\left(x_i, \bar{y}_{i,j}^-\right)$ and the y in the right of (3.5.1) by $\bar{y}_{i,j}^-$, $i = 1, ..., n_1$; $j = 1, 2, 3$. Then, we obtain a system of linear equations as follows.

$$\begin{cases} f\left(x_i, \bar{y}_{i,1}^{-}\right) = \sum_{k=1}^{3} c_{i,k} Q\left(\frac{\bar{y}_{i,1}^{-} - y_k}{h_2} - k\right), \\ f\left(x_i, \bar{y}_{i,2}^{-}\right) = \sum_{k=1}^{3} c_{i,k} Q\left(\frac{\bar{y}_{i,2}^{-} - y_k}{h_2} - k\right), \\ f\left(x_i, \bar{y}_{i,3}^{-}\right) = \sum_{k=1}^{3} c_{i,k} Q\left(\frac{\bar{y}_{i,3}^{-} - y_k}{h_2} - k\right). \end{cases} \tag{3.5.2}$$

where $i = 1, ..., n_1$.

By solving the system of linear equations in (3.5.2), we obtain the values of the coefficients $c_{i,k}$ (Anton and Rorres, 2000). With these coefficient values $c_{i,k}$, we obtain the artificial values at the points $\left(x_i, \bar{y}_{i,1}^{+}\right)$ based on the observed values at $\left(x_i, \bar{y}_{i,1}^{-}\right)$, $\left(x_i, \bar{y}_{i,2}^{-}\right)$, and $\left(x_i, \bar{y}_{i,3}^{-}\right)$. In a similar way, we obtain artificial values at other added points.

Similar discussion applies to the disjoint subregions G_3 and G_4 and their expanded subregions $\tilde{G}_3$ and $\tilde{G}_4$.

3.6 Surface fitting

We shall use the penalized least square method to construct two disjoint approximating product spline surfaces. This is achieved by formulating the least square surface fitting by product splines as an optimization problem. The cubic spline is given by

$$Q(x) = Q_3(x) = \begin{cases} 0, & |x| \geq 2, \\ \frac{1}{2}|x|^3 - x^2 + \frac{2}{3}, & |x| \leq 1, \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + \frac{4}{3}, & 1 < |x| < 2. \end{cases}$$

Suppose that the estimated JLC is parallel to $x-$axis. Define

$$\varphi\left(x, y\right) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} a_{i,j} Q\left(\frac{x - x_i}{h_1} - i\right) Q\left(\frac{y - y_j}{h_2} - j\right).$$

and

$$\psi\left(x,y\right)=\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}b_{i,j}Q\left(\frac{x-x_i}{h_1}-i\right)Q\left(\frac{y-y_j}{h_2}-j\right).$$

where k_1 and k_2 are two integers, and $a_{i,j}$, $b_{i,j}$, $i=1,...k_1$; $j=1,...,k_2$, are the coefficients of the two product cubic splines, respectively. The smoothness of a surface $z=g\left(x,y\right)$ in the region Ω is measured by the following smoothness function.

$$J_s=\iint\limits_{\Omega}\left(g_{xx}^2+2g_{xy}^2+g_{yy}^2\right)dxdy.$$

Denote $\boldsymbol{\rho}=[\rho_1,\ldots,\rho_{n_1}]^{\mathsf{T}}$. Let $\tilde{\Omega}_1=[0,1]\times[-\left(\hat{n}+1\right)h_2,\boldsymbol{\rho}]$ and let $\tilde{\Omega}_2=[0,1]\times[\boldsymbol{\rho},1]$, where $\theta_1\leq\boldsymbol{\rho}\leq\theta_1+h_2$. Then, the problem of fitting such a surface can be formulated as the following optimization problem

$$\begin{aligned}\min J\left(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\rho}\right)=&\sum_{(x_i,y_j)\in\tilde{G}_1}\left(\varphi\left(x_i,y_j\right)-z_{ij}\right)^2+\sum_{(x_i,y_j)\in\tilde{G}_2}\left(\psi\left(x_i,y_j\right)-z_{ij}\right)^2\\&+\lambda\iint\limits_{\tilde{\Omega}_1}\left(\varphi_{xx}^2(x,y)+2\varphi_{xy}^2(x,y)+\varphi_{yy}^2(x,y)\right)dxdy\\&+\lambda\iint\limits_{\tilde{\Omega}_2}\left(\psi_{xx}^2(x,y)+2\psi_{xy}^2(x,y)+\psi_{yy}^2(x,y)\right)dxdy,\end{aligned}\tag{3.6.1}$$

where $\boldsymbol{a}=[a_{1,1},...,a_{1,k_1},a_{2,1},...,a_{k_1,k_2}]^{\mathsf{T}}$ and $\boldsymbol{b}=[b_{1,1},...,b_{1,k_1},b_{2,1},...,b_{k_1,k_2}]^{\mathsf{T}}$ are the coefficients vectors and λ is a penalty constant. For easy reference, let this problem be referred to as Problem (3.P).

To solve this optimization problem, we let

$$\begin{cases}\partial J/\partial\boldsymbol{a}=0,\\\partial J/\partial\boldsymbol{b}=0,\\\partial J/\partial\boldsymbol{\rho}=0,\end{cases}\tag{3.6.2}$$

where the gradients can be derived as follows:

$$\begin{aligned}\frac{\partial J}{\partial a_{kl}} &= \sum_{(x_i,y_j)\in\tilde{G}_1} 2\left(\varphi\left(x_i,y_j\right)-z_{i,j}\right)\frac{\partial\varphi\left(x_i,y_j\right)}{\partial a_{kl}} \\ &\quad + \lambda\iint\limits_{\tilde{\Omega}_1}\left(\frac{\partial\varphi^2_{xx}(x,y)}{\partial a_{kl}} + 2\frac{\partial\varphi^2_{xy}(x,y)}{\partial a_{kl}} + \frac{\partial\varphi^2_{yy}(x,y)}{\partial a_{kl}}\right)dxdy \\ &= \sum_{(x_i,y_j)\in\tilde{G}_1} 2\left(\varphi\left(x_i,y_j\right)-z_{i,j}\right)Q\left(\frac{x-x_k}{h_1}-k\right)Q\left(\frac{y-y_l}{h_2}-l\right) \\ &\quad + \lambda\iint\limits_{\tilde{\Omega}_1}\left(\frac{\partial\varphi^2_{xx}(x,y)}{\partial a_{kl}} + 2\frac{\partial\varphi^2_{xy}(x,y)}{\partial a_{kl}} + \frac{\partial\varphi^2_{yy}(x,y)}{\partial a_{kl}}\right)dxdy, \end{aligned} \tag{3.6.3}$$

where

$$\varphi_{xx}(x,y) = \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q''(\frac{x-x_i}{h_1}-i)Q(\frac{y-y_j}{h_2}-j), \tag{3.6.4}$$

$$\begin{aligned}\frac{\partial\varphi^2_{xx}(x,y)}{\partial a_{kl}} =& 2\left(\sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q''(\frac{x-x_i}{h_1}-i)Q(\frac{y-y_j}{h_2}-j)\right) \\ &\cdot Q''(\frac{x-x_k}{h_1}-k)Q(\frac{y-y_l}{h_2}-l),\end{aligned} \tag{3.6.5}$$

$$\varphi_{xy}(x,y) = \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q'(\frac{x-x_i}{h_1}-i)Q'(\frac{y-y_j}{h_2}-j), \tag{3.6.6}$$

$$\begin{aligned}\frac{\partial\varphi^2_{xy}(x,y)}{\partial a_{kl}} =& 2\left(\sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q'(\frac{x-x_i}{h_1}-i)Q'(\frac{y-y_j}{h_2}-j)\right) \\ &\cdot Q'(\frac{x-x_k}{h_1}-k)Q'(\frac{y-y_l}{h_2}-l),\end{aligned} \tag{3.6.7}$$

$$\varphi_{yy}(x,y) = \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q(\frac{x-x_i}{h_1}-i)Q''(\frac{y-y_j}{h_2}-j), \tag{3.6.8}$$

$$\begin{aligned}\frac{\partial\varphi^2_{yy}(x,y)}{\partial a_{kl}} =& 2\left(\sum_{i=1}^{k_1}\sum_{j=1}^{k_2} a_{i,j}Q(\frac{x-x_i}{h_1}-i)Q''(\frac{y-y_j}{h_2}-j)\right) \\ &\cdot Q(\frac{x-x_k}{h_1}-k)Q''(\frac{y-y_l}{h_2}-l),\end{aligned} \tag{3.6.9}$$

and $\partial J/\partial b_{kl}$ is obtained similarly to $\partial J/\partial a_{kl}$, and

$$\begin{aligned}\frac{\partial J}{\partial \rho_k} =& \frac{\partial}{\partial \rho_k}\left(\lambda \sum_{i=1}^{n_1} \int_{(i-1)h_1}^{ih_1} dx \int_{-(\hat{n}+1)h_2}^{\rho_i} \left(\varphi_{xx}^2(x,y) + 2\varphi_{xy}^2(x,y) + \varphi_{yy}^2(x,y)\right) dy\right) \\ &+ \frac{\partial}{\partial \rho_k}\left(\lambda \sum_{i=1}^{n_1} \int_{(i-1)h_1}^{ih_1} dx \int_{\rho_i}^{1} \left(\psi_{xx}^2(x,y) + 2\psi_{xy}^2(x,y) + \psi_{yy}^2(x,y)\right) dy\right) \\ =& \lambda \int_{(k-1)h_1}^{kh_1} \left(\varphi_{xx}^2(x,\rho_k) + 2\varphi_{xy}^2(x,\rho_k) + \varphi_{yy}^2(x,\rho_k)\right) dx \\ &- \lambda \int_{(k-1)h_1}^{kh_1} \left(\psi_{xx}^2(x,\rho_k) + 2\psi_{xy}^2(x,\rho_k) + \psi_{yy}^2(x,\rho_k)\right) dx. \end{aligned} \tag{3.6.10}$$

Then, we can find out $\boldsymbol{a}^*$, $\boldsymbol{b}^*$, and $\boldsymbol{\rho}^*$.

For the case when the estimate JLC is parallel to $y-$axis, we can deal with it in a similar way as follows.

Denote $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{n_1}]^\intercal$. Let $\tilde{\Omega}_3 = [-\left(\bar{n}+1\right)h_1, \boldsymbol{\beta}] \times [0,1]$ and let $\tilde{\Omega}_4 = [\boldsymbol{\beta}, 1] \times [0,1]$, where $\theta_2 \leq \boldsymbol{\beta} \leq \theta_2 + h_1$. Then, the problem of fitting such a surface can be formulated as the optimization problem which is similar to (3.6.1). Thus, we can obtain the optimal values of the coefficients vectors and $\boldsymbol{\beta}$. Let them be determined as $\boldsymbol{c}^*$, $\boldsymbol{d}^*$, and $\boldsymbol{\beta}^*$.

3.7 Inverse Space Transformation

Since the space scaling transformation is a one-to-one mapping, we can find the inverse transformation to map the estimated JLC and the two approximating surfaces on the two shifted subregions, say G_1 and G_2, back to the original space according to either Theorem 3.4.1 or Theorem 3.4.2. They are stated in the following as two separate theorems.

Theorem 3.7.1. *Consider the situation where the estimated JLC is parallel to x-axis. Suppose that the estimated JLC and the two approximating tensor product cubic spline surfaces have been constructed on the shifted disjoint subregions G_1*

and G_2 using the approach presented in earlier sections. Then, these estimated JLC and the two approximating product cubic spline surfaces are mapped back to the original space by the following inverse transformation.

$$\begin{cases} x = \widetilde{x}, \\ y = \widetilde{y} + \hat{y}_i - \min\limits_{y} \hat{D}_1. \end{cases} \tag{3.7.1}$$

Theorem 3.7.2. *Consider the situation where the estimated JLC is parallel to y-axis. Suppose that the estimated JLC and the two approximating tensor product cubic spline surfaces have been constructed on the shifted disjoint subregions G_3 and G_{42} using the approach presented in earlier sections. Then, these estimated JLC and the two approximating tensor product cubic spline surfaces are mapped back to the original space by the following inverse transformation.*

$$\begin{cases} x = \widetilde{x} + \bar{x}_i - \min\limits_{x} \hat{D}_2, \\ y = \widetilde{y}. \end{cases} \tag{3.7.2}$$

3.8 Numerical Example

In this section, we shall construct approximating tensor product cubic spline surface to the surface of the regression function given by

$$\begin{aligned} f(x,y) = 0.25(1-x)y + [(1+0.2\sin(2\pi x))]I_y \\ \geq 0.6\sin(0.2x+0.4) + \varepsilon_{ij}. \end{aligned} \tag{3.8.1}$$

where $(x,y) \in [0,1] \times [0,1]$ and $i,j \in \{1,2,..n_1\}$. All calculations are carried out within the Matlab environment.

Suppose that the observation data are obtained from (3.8.1) which is corrupted by a Gaussian noise, $\varepsilon_i \sim N(0,0.3^2)$, at the points $(x_i,y_i) = (i/n_1, i/n_1)$, $\forall\ i = 1,2...n$. The regression surface has a unique JLC

$$\phi(x) = 0.6\sin(0.2x+0.4)$$

and the jump magnitude is

$$1 + 0.2\sin(2\pi x).$$

We define the following two Epanechnikov kernel functions, $K_1(x,y)$ and $K_2(x,y)$(http://www.quantlet.com/mdstat/scripts/mm3/pdf/mm3pdf.pdf).

$$K_2(x,y) = 12/11(1-x^2)(12/11(1-(y-0.5)^2)I_{[-1/2,1/2]\times[0,1]},$$

and

$$K_1(x,y) = K_2(x,-y).$$

We choose $n_1 = 100$. The sample size is $n = (n_1)^2$. $h_n = k_1/n_1$ and $p_n = k_2/n_1$, where k_1 and k_2 are two positive odd integer numbers.

The regression function $f(x,y)$ in (3.8.1) with $n_1 = 100$ is depicted in Figure 3.1. The observations, the regression function $f(x,y)$ with noise $\varepsilon_i \sim N(0, 0.3^2)$. The figure of $M_n^{(1)}(x,y)$ in (3.3.1) for $f(x,y)$ in (3.8.1) is displayed in Figure 3.2. If we remove the noise in (3.8.1), we obtain $f(x,y) - \varepsilon_i$. After we perform the space scaling transformation for $f(x,y) - \varepsilon_i$ and the surface fitting by tensor product cubic splines for the two continuous surfaces combined with the row pixels or the column pixels that the jump location curve is shifted into, we show the two overlapping subregions in Figure 3.3. After we process $f(x,y)$ in (3.8.1) according to procedures given in Section 3.3, 3.4, 3.5 and 3.6 , we obtain Figure 3.4.

After we process $f(x,y)$ in (3.8.1) according to procedures given in Section 3.3, 3.4, 3.5 and 3.6 , we obtain Figure 3.4.

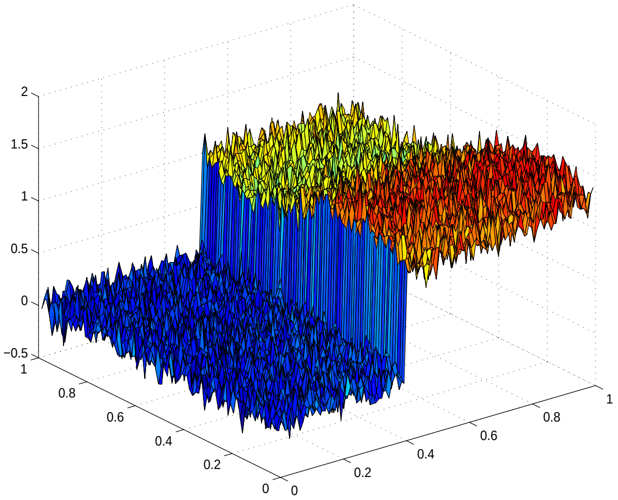

Figure 3.1: The regression function $f(x, y)$ in (3.8.1) with $n_1 = 100$.

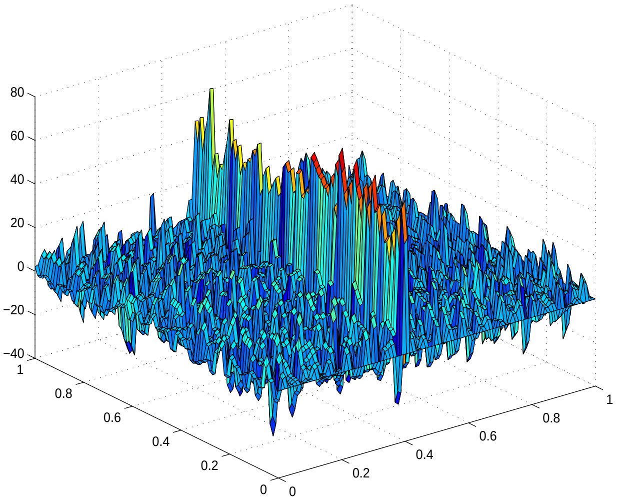

Figure 3.2: The figure of $M_n^{(1)}(x, y)$ for $f(x, y)$.

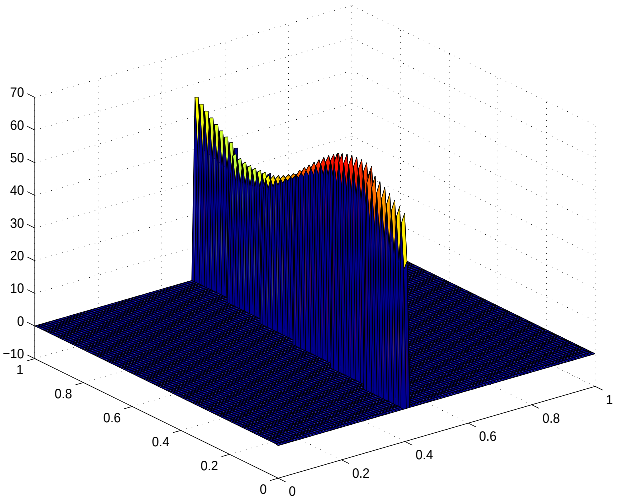

Figure 3.3: The obtained two subregions after processing $f(x, y) - \varepsilon_i$ according to the procedures given in Section 3.4, 3.5, and 3.6.

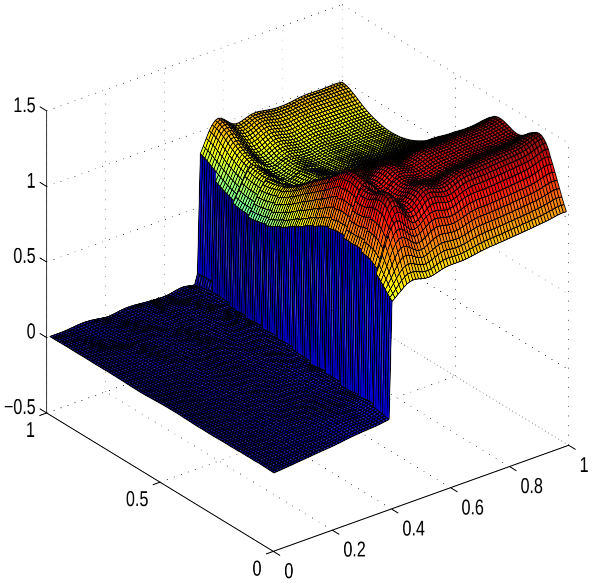

Figure 3.4: The resultant figure after processing $f(x, y)$ according to the procedures given in Section 3.3, 3.4, 3.5 and 3.6.

Chapter 4

Stochastic parameter selection problems with state jumps

4.1 Introduction

This chapter considers a class of stochastic optimal parameter selection problems described by linear Ito stochastic differential equations with state jumps subject to probabilistic constraints on the state, where the times at which the jumps occurred as well as their heights are decision variables. We show that this constrained stochastic impulsive optimal parameter selection problem is equivalent to a deterministic impulsive optimal parameter selection problem subject to continuous state inequality constraints, where the times at which the jumps occurred as well as their heights remain as decision variables. Then, by introducing a time scaling transform (Lee, Teo, Rehbock and Jennings, 1997), we show that this constrained deterministic impulsive optimal parameter selection problem is transformed into an equivalent constrained deterministic impulsive optimal parameter selection problem with fixed jump times. A constraint transcription technique (Jennings and Teo, 1990) is then used to approximate the continuous state inequality constraints by a sequence of canonical inequality constraints. This leads to a sequence of approximate deterministic impulsive optimal parameter selection problems subject to canonical inequality constraints. For each of these approximate problems, we derive the gradient formulas of the cost function and the constraint functions. On this basis, an efficient computational method is developed.

The main reference for this chapter is (Teo, Goh and Wong, 1991; Goh and Teo, 1990; Teo and Ahmed, 1974; Lee, Teo, Rehbock and Jennings, 1997; Jennings and Teo, 1990; Liu, Feng and Teo, to appear in 2008.).

4.2 Statement of Problem

Consider an impulsive dynamical system described by linear Ito stochastic differential equations defined on a fixed time interval $(0, T]$:

$$d\boldsymbol{\xi}(t) = \boldsymbol{A}(t,\boldsymbol{\delta})\boldsymbol{\xi}(t)dt + \boldsymbol{B}(t,\boldsymbol{\delta})dt + \boldsymbol{D}(t,\boldsymbol{\delta})d\boldsymbol{w}(t) \qquad (4.2.1.a)$$

$$\boldsymbol{\xi}(0) = \boldsymbol{\xi}^0 \qquad (4.2.1.b)$$

$$\boldsymbol{\xi}(\tau_i^+) = \boldsymbol{J}^i\boldsymbol{\xi}(\tau_i^-) + \boldsymbol{\Delta}_i + \boldsymbol{\gamma}^i, \quad i = 1,\ldots,m. \qquad (4.2.1.c)$$

Here, $\boldsymbol{\xi}(t) = [\xi_1(t),\ldots,\xi_n(t)]^\intercal \in \mathbb{R}^n$ is the state vector; $\boldsymbol{\delta} = [\delta_1,\ldots,\delta_r]^\intercal \in \mathbb{R}^r$ is the system parameter vector; $\boldsymbol{\xi}^0 = [\xi_1^0,\ldots,\xi_n^0]^\intercal \in \mathbb{R}^n$ is the initial state vector which is Gaussian with mean $\boldsymbol{\mu}^0$ and covariance matrix $\boldsymbol{\Psi}^0$; and $\boldsymbol{w}(t) = [w_1(t),\ldots,w_d(t)]^\intercal \in \mathbb{R}^d$ is a Wiener process with mean $\mathbf{0}$ and covariance matrix given by

$$\mathcal{E}\{\boldsymbol{w}(t_1)(\boldsymbol{w}(t_2))^\intercal\} = \int_0^{\min\{t_1,t_2\}} \boldsymbol{I}ds,$$

where $\mathcal{E}\{\cdot\}$ denotes the mathematical expectation and $\boldsymbol{I}$ is the identity matrix. Equation (4.2.1.c) are conditions on the state jumps, where $\boldsymbol{J}^i \in \mathbb{R}^{n\times n}$, $i = 1,\ldots,m,$ are given coefficient matrices, $\tau_1,\ldots,\tau_m,$ are the time points at which the state jumps are occurred, $\boldsymbol{\Delta}_i$, $i = 1,\ldots,m,$ are Gaussian vectors with mean $\mathbf{0}$ and covariance matrices $\boldsymbol{K}^i$, $i = 1,\ldots,m,$ and $\boldsymbol{\gamma}^i = [\gamma_1^i,\ldots,\gamma_n^i]^\intercal$, $i = 1,\ldots,m,$ are the magnitude vectors of the jumps. Let $\boldsymbol{\tau} = [\tau_1,\ldots,\tau_m]^\intercal$.

We assume that the following conditions are satisfied.

(i). $\boldsymbol{A}(t, \boldsymbol{\delta}) \in \mathbb{R}^{n \times n}$, $\boldsymbol{B}(t, \boldsymbol{\delta}) \in \mathbb{R}^{n}$ and $\boldsymbol{D}(t, \boldsymbol{\delta}) \in \mathbb{R}^{n \times d}$ are continuously differentiable with respect to all their arguments.

(ii). The Wiener process $\boldsymbol{w}(t)$ and the random vectors $\boldsymbol{\xi}^0$, $\boldsymbol{\Delta}_i$, $i = 1, \dots, m$, are statistically mutually independent.

The probabilistic state constraints given below arise naturally when the state is required to stay within a given acceptable region with a given degree of confidence for all $t \in [0, T]$.

$$Prob\{a_k \leq (\boldsymbol{c}^k)^{\intercal} \boldsymbol{\xi}(t) \leq b_k\} > \alpha_k, \ \forall t \in [0, T], \ k = 1, \dots, N, \tag{4.2.2}$$

where $\boldsymbol{c}^k$, $k = 1, \dots, N$, are $n-$vectors, and a_k, b_k, α_k, $k = 1, \dots, N$, are real constants.

Define a compact set

$$\boldsymbol{\Omega} = \{\boldsymbol{\delta} \in \mathbb{R}^r : h_j(\boldsymbol{\delta}) \leq 0, \ \ j = 1, \dots, M\}, \tag{4.2.3}$$

where h_j, $j = 1, \dots, M$, are continuously differentiable functions of the parameter $\boldsymbol{\delta}$. Let $\boldsymbol{h} = [h_1, \dots, h_M]^{\intercal}$.

For the jump time vector $\boldsymbol{\tau} = [\tau_1, ..., \tau_m]^{\intercal}$, it is assumed, without loss of generality, that

$$0 < \tau_1 < \cdots < \tau_m < T. \tag{4.2.4}$$

Let $\boldsymbol{\mathcal{T}}$ be the set of all those $\boldsymbol{\tau} = [\tau_1, \dots, \tau_m]^{\intercal}$ which satisfy (4.2.4). For brevity in notation, we denote $\tau_0 = 0$ and $\tau_{m+1} = T$.

Let $\boldsymbol{\Gamma}$ be the set of all those magnitude vectors $\boldsymbol{\gamma} = [(\boldsymbol{\gamma}^1)^{\intercal}, \dots, (\boldsymbol{\gamma}^m)^{\intercal}]^{\intercal}$ such that

$$\underline{\gamma}_j^i \leq \gamma_j^i \leq \overline{\gamma}_j^i, i = 1, \dots, m; \ j = 1, \dots, n. \tag{4.2.5}$$

An element $(\boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\gamma}) \in \boldsymbol{\Omega} \times \boldsymbol{\mathcal{T}} \times \boldsymbol{\Gamma}$ is said to be a feasible parameter vector if it satisfies the probabilistic state constraints (4.2.2). Let $\boldsymbol{\mathcal{D}}$ be the class of all such feasible parameter vectors. We may now state our problem formally as follows.

Problem (4.P1)**.** *Given the dynamical system (4.2.1), find a feasible parameter vector $(\boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\gamma}) \in \boldsymbol{\mathcal{D}}$, such that the cost function*

$$\begin{aligned} g_0(\boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\gamma}) &= \varphi(\boldsymbol{\gamma}) + \mathcal{E}\{(\boldsymbol{\xi}(T))^{\mathsf{T}} \boldsymbol{S}_2(\boldsymbol{\delta})\boldsymbol{\xi}(T) + (\boldsymbol{S}_1(\boldsymbol{\delta}))^{\mathsf{T}}\boldsymbol{\xi}(T) + \boldsymbol{S}_0(\boldsymbol{\delta}) \\ &\quad + \sum_{i=1}^{m+1} \int_{\tau_{i-1}}^{\tau_i} [(\boldsymbol{\xi}(t))^{\mathsf{T}} \boldsymbol{Q}_2(t, \boldsymbol{\delta})\boldsymbol{\xi}(t) + (\boldsymbol{Q}_1(t, \boldsymbol{\delta})^{\mathsf{T}}\boldsymbol{\xi}(t) + \boldsymbol{Q}_0(t, \boldsymbol{\delta})]dt\} \end{aligned} \tag{4.2.6}$$

is minimized, where $\varphi(\boldsymbol{\gamma})$ is a penalty term to prevent high jumps, and $\boldsymbol{S}_2(\boldsymbol{\delta}) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{Q}_2(t, \boldsymbol{\delta}) \in \mathbb{R}^{n\times n}$ are positive semi-definite matrices which are continuously differentiable with respect to their respective arguments, while $\boldsymbol{S}_1(\boldsymbol{\delta})$ and $\boldsymbol{Q}_1(t, \boldsymbol{\delta})$ (respectively, $\boldsymbol{S}_0(\boldsymbol{\delta})$ and $\boldsymbol{Q}_0(t, \boldsymbol{\delta})$) are $n-$vector valued functions (respectively, real-valued functions) which are also continuously differentiable with respect to their respective arguments.

Problem (4.P1) is a stochastic impulsive optimal parameter selection problem with probabilistic constraints. We shall show that it is equivalent to a deterministic optimal parameter selection problem subject to continuous state inequality constraints. A numerical computational method will be developed for solving this equivalent constrained deterministic optimal parameter selection problem. This is to be done in several stages as detailed below.

4.3 Deterministic Transformation

In this section, we shall show that the stochastic impulsive optimal parameter selection problem (4.P1) can be transformed into a deterministic impulsive optimal parameter selection problem.

For each $\boldsymbol{\delta}$, it is clear from (4.2.1) that the solution of system (4.2.1), for $t \in (\tau_{i-1}, \tau_i)$ with $i = 1, \ldots, m$, is given by

$$\begin{aligned}\boldsymbol{\xi}(t) = \boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta})\boldsymbol{\xi}(\tau_{i-1}^{+}) + \int_{\tau_{i-1}}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{B}(s, \boldsymbol{\delta})ds \\ + \int_{\tau_{i-1}}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{D}(s, \boldsymbol{\delta})d\boldsymbol{w}(s),\end{aligned} \tag{4.3.1}$$

where $\boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta}) \in \mathbb{R}^{n\times n}$ is the principal solution matrix of the homogeneous system:

$$\frac{\partial \boldsymbol{\Phi}(t, s)}{\partial t} = \boldsymbol{A}(t, \boldsymbol{\delta})\boldsymbol{\Phi}(t, s), \quad t > s \tag{4.3.2a}$$

$$\boldsymbol{\Phi}(s, s) = \boldsymbol{I}. \tag{4.3.2b}$$

Theorem 4.3.1. *The process $\{\boldsymbol{\xi}(t) : t \geq 0\}$ is a Gaussian process with mean and covariance matrix given, for* t $\in (\tau_{i-1}, \tau_i)$ *with* $i = 1, ..., m$, *by*

$$\begin{aligned}\boldsymbol{\mu}(t) &= \mathcal{E}\{\boldsymbol{\xi}(t)\} \\ &= \boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta})\boldsymbol{\mu}(\tau_{i-1}^{+}) + \int_{\tau_{i-1}}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{B}(s, \boldsymbol{\delta})ds,\end{aligned} \tag{4.3.3}$$

and

$$\begin{aligned}\boldsymbol{\Psi}(t) =& \boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta})\boldsymbol{\Psi}(\tau_{i-1}^{+})(\boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta}))^{\intercal} \\ &+ \int_{\tau_{i-1}}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{D}(\tau, \boldsymbol{\delta})(\boldsymbol{D}(s, \boldsymbol{\delta}))^{\intercal}(\boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta}))^{\intercal}ds,\end{aligned} \tag{4.3.4}$$

respectively. Here, at t $= \tau_i$ *with* $i = 1, ..., m$, *the mean and the covariance matrix of* $\boldsymbol{\xi}(\tau_i^{+})$ *are*

$$\boldsymbol{\mu}(\tau_i^{+}) = \boldsymbol{J}^i\boldsymbol{\mu}(\tau_i^{-}) + \boldsymbol{\gamma}^i, \tag{4.3.5}$$

and

$$\boldsymbol{\Psi}(\tau_i^{+}) = \boldsymbol{J}^i\boldsymbol{\Psi}(\tau_i^{-})(\boldsymbol{J}^i)^{\intercal} + \boldsymbol{K}^i, \tag{4.3.6}$$

respectively.

Proof: Since $\boldsymbol{\xi}^0$ is a Gaussian vector and the linear transformation of a Gaussian is Gaussian, it follows from (4.3.1) with $i = 1$ that $\{\boldsymbol{\xi}(t) : 0 \leq t \leq \tau_1\}$ is a Gaussian process. Its mean and covariance matrix are given, for $t \in (0, \tau_1)$, by

$$\begin{aligned}\boldsymbol{\mu}(t) = \mathcal{E}\{\boldsymbol{\xi}(t)\} &= \boldsymbol{\Phi}(t, 0 \mid \boldsymbol{\delta})\mathcal{E}\{\boldsymbol{\xi}^0\} + \int_0^t \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{B}(s, \boldsymbol{\delta})ds \\ &= \boldsymbol{\Phi}(t, 0 \mid \boldsymbol{\delta})\boldsymbol{\mu}^0 + \int_0^t \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{B}(s, \boldsymbol{\delta})ds \qquad (4.3.7)\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\Psi}(t) =&\boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta})\boldsymbol{\Psi}(\tau_{i-1}^+)(\boldsymbol{\Phi}(t, \tau_{i-1} \mid \boldsymbol{\delta}))^\intercal \\ &+ \int_0^t \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{D}(\tau, \boldsymbol{\delta})(\boldsymbol{D}(s, \boldsymbol{\delta}))^\intercal(\boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta}))^\intercal ds, \qquad (4.3.8)\end{aligned}$$

respectively.

Note that $\boldsymbol{\xi}(\tau_1^-)$ is a Gaussian vector. Since the linear transformation of Gaussian is Gaussian, it follows from (4.2.1c) with $i = 1$ that $\boldsymbol{\xi}(\tau_1^+)$ is Gaussian with the mean and covariance matrix given by

$$\boldsymbol{\mu}(\tau_1^+) = \mathcal{E}\{\boldsymbol{\xi}(\tau_1^+)\} = \mathcal{E}\{\boldsymbol{J}^1\boldsymbol{\xi}(\tau_1^-)\} + \boldsymbol{\gamma}^1 = \boldsymbol{J}^1\boldsymbol{\mu}(\tau_1^-) + \boldsymbol{\gamma}^1, \qquad (4.3.9)$$

and

$$\begin{aligned}\boldsymbol{\Psi}(\tau_1^+) &= \mathcal{E}\{[\boldsymbol{\xi}(\tau_1^+) - \boldsymbol{\mu}(\tau_1^+)][\boldsymbol{\xi}(\tau_1^+) - \boldsymbol{\mu}(\tau_1^+)]^\intercal\} \\ &= \mathcal{E}\{[\boldsymbol{J}^1\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{J}^1\boldsymbol{\mu}(\tau_1^-) + \boldsymbol{\Delta}_1][\boldsymbol{J}^1\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{J}^1\boldsymbol{\mu}(t_1^-) + \boldsymbol{\Delta}_1]^\intercal\} \\ &= \mathcal{E}\{[\boldsymbol{J}^1\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{J}^1\boldsymbol{\mu}(\tau_1^-)][\boldsymbol{J}^1\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{J}^1\boldsymbol{\mu}(\tau_1^-)]^\intercal\} + \mathcal{E}\{\boldsymbol{\Delta}_1\boldsymbol{\Delta}_1^\intercal\} \\ &= \boldsymbol{J}^1\mathcal{E}\{[\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{\mu}(\tau_1^-)][\boldsymbol{\xi}(\tau_1^-) - \boldsymbol{\mu}(\tau_1^-)]^\intercal\}(\boldsymbol{J}^1)^\intercal + \boldsymbol{K}^1 \\ &= \boldsymbol{J}^1\boldsymbol{\Psi}(\tau_1^-)(\boldsymbol{J}^1)^\intercal + \boldsymbol{K}^1. \qquad (4.3.10)\end{aligned}$$

respectively.

By the same token, we can show that for $i = 2$, the process $\{\boldsymbol{\xi}(t) : t \in [\tau_1, \tau_2]\}$ is a Gaussian process with the mean and covariance matrix given, for $t \in (\tau_1, \tau_2)$, by

$$\boldsymbol{\mu}(t) = \boldsymbol{\Phi}(t, \tau_1 \mid \boldsymbol{\delta})\boldsymbol{\mu}(\tau_1^+) + \int_{\tau_1}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{B}(s, \boldsymbol{\delta})ds \tag{4.3.11}$$

and

$$\begin{aligned}\boldsymbol{\Psi}(t) =& \boldsymbol{\Phi}(t, \tau_1 \mid \boldsymbol{\delta})\boldsymbol{\Psi}(\tau_1^+)(\boldsymbol{\Phi}(t, \tau_1 \mid \boldsymbol{\delta}))^{\intercal} \\ &+ \int_{\tau_1}^{t} \boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta})\boldsymbol{D}(\tau, \boldsymbol{\delta})(\boldsymbol{D}(s, \boldsymbol{\delta}))^{\intercal}(\boldsymbol{\Phi}(t, s \mid \boldsymbol{\delta}))^{\intercal}ds,\end{aligned} \tag{4.3.12}$$

respectively.

At $t = \tau_2$, it follows from (4.2.1c) with $i = 2$ that the mean and the covariance matrix of $\boldsymbol{\xi}(\tau_2^+)$ are

$$\boldsymbol{\mu}(\tau_2^+) = \mathcal{E}\{\boldsymbol{\xi}(\tau_2^+)\} = \boldsymbol{J}^2\boldsymbol{\mu}(\tau_2^-) + \boldsymbol{\gamma}^2, \tag{4.3.13}$$

and

$$\begin{aligned}\boldsymbol{\Psi}(\tau_2^+) &= \mathcal{E}\{[\boldsymbol{\xi}(\tau_2^+) - \boldsymbol{\mu}(\tau_2^+)][\boldsymbol{\xi}(\tau_2^+) - \boldsymbol{\mu}(\tau_2^+)]^{\intercal}\} \\ &= \boldsymbol{J}^2\boldsymbol{\Psi}(\tau_2^-)(\boldsymbol{J}^2)^{\intercal} + \boldsymbol{K}^2,\end{aligned} \tag{4.3.14}$$

respectively.

The process can be repeated for $i = 3, .., m$. This completes the proof.

From (4.3.7), it follows that, for $t \in (\tau_{i-1}, \tau_i)$ with $i = 1, 2, ..., m$, $\boldsymbol{\mu}(t)$ is the solution of the following system of differential equations:

$$d\boldsymbol{\mu}(t)/dt = \boldsymbol{A}(t, \boldsymbol{\delta})\boldsymbol{\mu}(t) + \boldsymbol{B}(t, \boldsymbol{\delta}) \tag{4.3.15a}$$

with initial condition

$$\boldsymbol{\mu}(0) = \boldsymbol{\mu}^0, \tag{4.3.15b}$$

and jump conditions

$$\boldsymbol{\mu}(\tau_i^+) = \boldsymbol{J}^i\boldsymbol{\mu}(\tau_i^-) + \boldsymbol{\gamma}^i. \tag{4.3.15c}$$

Similarly, it follows from (4.3.8) that, for $t \in (\tau_{i-1}, \tau_i)$ with $i = 1, 2, ..., m$, $\boldsymbol{\Psi}(t \mid \boldsymbol{\delta})$ is the solution of the following matrix differential equation:

$$d\boldsymbol{\Psi}(\boldsymbol{t})/dt = \boldsymbol{A}(t, \boldsymbol{\delta})\boldsymbol{\Psi}(t) + \boldsymbol{\Psi}^{\intercal}(t)\boldsymbol{A}(t, \boldsymbol{\delta}) + \boldsymbol{D}(t, \boldsymbol{\delta})(\boldsymbol{D}(t, \boldsymbol{\delta}))^{\intercal} \qquad (4.3.16a)$$

with initial condition

$$\boldsymbol{\Psi}(0) = \boldsymbol{\Psi}^0, \qquad (4.3.16b)$$

and jump conditions

$$\boldsymbol{\Psi}(\tau_i^+) = \boldsymbol{J}^i\boldsymbol{\Psi}(\tau_i^-)(\boldsymbol{J}^i)^{\intercal} + \boldsymbol{K}^i, \quad i = 1, \dots, m. \qquad (4.3.16c)$$

Consider the cost function (2.6). Since $\mathcal{E}\{\boldsymbol{\xi}(t)\boldsymbol{\xi}^{\intercal}(t)\} = \boldsymbol{\Psi}(t) + \boldsymbol{\mu}(t)(\boldsymbol{\mu}(t))^{\intercal}$, it follows that

$$\begin{aligned} g_0(\boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\gamma}) = \varphi(\boldsymbol{\gamma}) + \boldsymbol{S}_2(\boldsymbol{\delta})[\boldsymbol{\Psi}(T) + \boldsymbol{\mu}(T)(\boldsymbol{\mu}(T))^{\intercal}] + \boldsymbol{S}_1^{\intercal}(\boldsymbol{\delta})\boldsymbol{\mu}(T) + \boldsymbol{S}_0(\boldsymbol{\delta}) \\ + \sum_{i=1}^{m+1} \int_{\tau_{i-1}}^{\tau_i} \{\text{trace}[\boldsymbol{Q}_2(t, \boldsymbol{\delta})(\boldsymbol{\Psi}(t) + \boldsymbol{\mu}(t)(\boldsymbol{\mu}(t))^{\intercal})] \\ + \boldsymbol{Q}_1(t, \boldsymbol{\delta})^{\intercal}\boldsymbol{\mu}(t) + \boldsymbol{Q}_0(t, \boldsymbol{\delta})\}dt. \quad (4.3.17) \end{aligned}$$

We now consider the probabilistic state constraint (4.2.2). Since $\boldsymbol{\xi}(t)$ is a Gaussian vector with mean $\boldsymbol{\mu}(t)$ and covariance $\boldsymbol{\Psi}(t)$, it is clear that for each $k = 1, \dots, N$, the scalar product $(\boldsymbol{c}^k)^{\intercal}\boldsymbol{\xi}(t)$ is a Gaussian variable with the mean $(\boldsymbol{c}^k)^{\intercal}\boldsymbol{\mu}(t)$ and covariance $(\boldsymbol{c}^k)^{\intercal}\boldsymbol{\Psi}(t)\boldsymbol{c}^k$. Thus, for each $k = 1, \dots, N$, (4.2.2) is equivalent to

$$\begin{aligned} &q_k(t, \boldsymbol{\mu}(t), \boldsymbol{\Psi}(t)) \\ =&\alpha_k - \int_{a_k}^{b_k} \frac{1}{(2\pi(\boldsymbol{c}^k)^{\intercal}\boldsymbol{\Psi}(t)\boldsymbol{c}^k)^{1/2}} \exp\{\frac{-(y - (\boldsymbol{c}^k)^{\intercal}\boldsymbol{\mu}(t))^2}{2(\boldsymbol{c}^k)^{\intercal}\boldsymbol{\Psi}(t)\boldsymbol{c}^k}\}dy \le 0, \qquad (4.3.18) \end{aligned}$$

for all $t \in [0, T]$. These constraints are continuous state inequality constraints.

Now, we have transformed the stochastic optimal parameter selection problem into a deterministic optimal parameter selection problem defined as follows.

Problem (4.P2)**.** *Given the dynamical system (4.3.15a)-(4.3.15c) and (4.3.16a)-(4.3.16c), and the continuous state inequality constraints (4.3.18), find a feasible parameter* $(\boldsymbol{\delta}, \boldsymbol{\tau}, \gamma) \in \boldsymbol{\Omega} \times \boldsymbol{\mathcal{T}} \times \boldsymbol{\Gamma}$*, such that the cost function (4.3.17) is minimized.*

We now summarize the results obtained so far below as a theorem.

Theorem 4.3.2. *Problem (4.P1) is equivalent to Problem (4.P2).*

4.4 Time Scaling Transformation

Problem (4.P2) is a deterministic impulsive optimal parameter selection problem subject to continuous state inequality constraints, where the jump times are decision variables to be determined optimally. This will encounter difficulty in numerical calculation when the jump times of the impulsive optimal parameter selection problem are varying. In this section, we will use a time scaling transform reported in (Lee, Teo, Rehbock and Jennings, 1997) to map these variable jump times into fixed knots in a new time scale.

We consider a new time variable s which varies from 0 to $m+1$. We re-scale $t \in [0, T]$ into $s \in [0, m+1]$. The transformation from $t \in [0, T]$ to $s \in [0, m+1]$ is defined by the differential equation

$$dt(s)/ds = \upsilon(s) = \sum_{i=1}^{m+1} \upsilon_i \chi_{[i-1,i]}(t) \tag{4.4.1a}$$

$$t(0) = 0, \tag{4.4.1b}$$

where $\upsilon_i = \tau_i - \tau_{i-1}$. Let $\boldsymbol{\Upsilon}$ be the set of all those $\boldsymbol{\upsilon} = [\upsilon_1, \ldots, \upsilon_{m+1}]^\intercal \in \mathbb{R}^{m+1}$ such that

$$\upsilon_i \geq 0, \ i = 1, \ldots, m+1. \tag{4.4.2}$$

Obviously, the following constraint must also be satisfied.

$$\sum_{i=1}^{m+1} \upsilon_i = T. \tag{4.4.3}$$

Denote $\hat{\boldsymbol{\mu}}(s) = \boldsymbol{\mu}(t(s))$, and $\hat{\boldsymbol{\Psi}}(s) = \boldsymbol{\Psi}(t(s))$. Then, (4.3.15) and (4.3.16) are transformed into

$$d\hat{\boldsymbol{\mu}}(s)/ds = v(s)[\boldsymbol{A}(t(s),\boldsymbol{\delta})\hat{\boldsymbol{\mu}}(s) + \boldsymbol{B}(t(s),\boldsymbol{\delta})] \tag{4.4.4a}$$

$$\hat{\boldsymbol{\mu}}(0) = \boldsymbol{\mu}^0 \tag{4.4.4b}$$

$$\hat{\boldsymbol{\mu}}(i^+) = J^i\hat{\boldsymbol{\mu}}(i^-) + \boldsymbol{\gamma}^i, \quad i = 1, \ldots, m, \tag{4.4.4c}$$

and

$$\begin{aligned} d\hat{\boldsymbol{\Psi}}(s)/ds =& v(s)[\boldsymbol{A}(t(s),\boldsymbol{\delta})\hat{\boldsymbol{\Psi}}(s) + \hat{\boldsymbol{\Psi}}^{\mathsf{T}}(s)\boldsymbol{A}(t(s),\boldsymbol{\delta}) \\ &+ \boldsymbol{D}(t(s),\boldsymbol{\delta})(\boldsymbol{D}(t(s),\boldsymbol{\delta}))^{\mathsf{T}}] \end{aligned} \tag{4.4.5a}$$

$$\hat{\boldsymbol{\Psi}}(0) = \boldsymbol{\Psi}^0. \tag{4.4.5b}$$

$$\hat{\boldsymbol{\Psi}}(i^+) = \boldsymbol{J}^i\hat{\boldsymbol{\Psi}}(i^-)(\boldsymbol{J}^i)^{\mathsf{T}} + \boldsymbol{K}^i, \quad i = 1, \ldots, m. \tag{4.4.5c}$$

The cost function (4.3.17) is transformed into

$$\begin{aligned} \hat{g}_0(\boldsymbol{\delta},\boldsymbol{v},\boldsymbol{\gamma}) =& \hat{\Phi}_0(\hat{\boldsymbol{\mu}}(m+1),\hat{\boldsymbol{\Psi}}(m+1),\boldsymbol{\delta},\boldsymbol{\gamma}) \\ &+ \sum_{i=1}^{m+1}\int_{i-1}^{i}\hat{\mathcal{L}}_0(t(s),\hat{\boldsymbol{\mu}}(s),\hat{\boldsymbol{\Psi}}(s),\boldsymbol{\delta},\boldsymbol{v},\boldsymbol{\gamma})ds, \end{aligned} \tag{4.4.6}$$

where

$$\begin{aligned} \hat{\Phi}_0(\hat{\boldsymbol{\mu}}(m+1),\hat{\boldsymbol{\Psi}}(m+1),\boldsymbol{\delta},\boldsymbol{\gamma}) =& \varphi(\gamma) + \boldsymbol{S}_1^{\mathsf{T}}(\boldsymbol{\delta})\hat{\boldsymbol{\mu}}(m+1) + \boldsymbol{S}_0(\boldsymbol{\delta}) \\ &+ \text{trace}\{\boldsymbol{S}_2(\boldsymbol{\delta})[\hat{\boldsymbol{\Psi}}(m+1) + \hat{\boldsymbol{\mu}}(m+1)(\hat{\boldsymbol{\mu}}(m+1))^{\mathsf{T}}]\} \end{aligned}$$

and

$$\begin{aligned} \hat{\mathcal{L}}_0(t(s),\hat{\boldsymbol{\mu}}(s),\hat{\boldsymbol{\Psi}}(s),\boldsymbol{\delta},\boldsymbol{v},\gamma) =& v_i\{\text{trace}[\boldsymbol{Q}_2(t(s),\boldsymbol{\delta})(\hat{\boldsymbol{\Psi}}(s) + \hat{\boldsymbol{\mu}}(s)(\hat{\boldsymbol{\mu}}(s))^{\mathsf{T}})] \\ &+ \boldsymbol{Q}_1(t(s),\boldsymbol{\delta})^{\mathsf{T}}\hat{\boldsymbol{\mu}}(s) + \boldsymbol{Q}_0(t(s),\boldsymbol{\delta})\}, \quad i \in (i-1,i). \end{aligned}$$

For the continuous state inequality constraints (4.3.18), they are transformed into

$$\begin{aligned}&\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s))\\ =&\alpha_k - \int_{a_k}^{b_k} \frac{1}{(2\pi(\boldsymbol{c}^k)^\intercal \hat{\boldsymbol{\Psi}}(s)\boldsymbol{c}^k)^{1/2}} \exp\{\frac{-(y-(\boldsymbol{c}^k)^\intercal \hat{\boldsymbol{\mu}}(s))^2}{2(\boldsymbol{c}^k)^\intercal \hat{\boldsymbol{\Psi}}(s)\boldsymbol{c}^k}\}dy \leq 0,\end{aligned} \tag{4.4.7}$$

for all $s \in [0, m+1]$, where $k = 1, \ldots, N$.

Then, after this time scaling transformation, Problem (4.P2) is equivalent to

Problem (4.P3)**.** *Given the dynamical system (4.4.1), (4.4.4) and (4.4.5), find a feasible parameter from* $(\boldsymbol{\delta}, \boldsymbol{\upsilon}, \gamma) \in \boldsymbol{\Omega} \times \boldsymbol{\Upsilon} \times \boldsymbol{\Gamma}$ *such that the cost function (4.4.6) is minimized subject to the constraints (4.4.3) and the continuous state inequality constraints (4.4.7).*

4.5 Constraint Transcription

From the continuous state inequality constraints (4.4.7), we see that these inequality constraints are to be satisfied for all $s \in [0, m+1]$. They are extremely difficult to deal with directly. We shall use a constraint transcription technique introduced in (Jennings and Teo, 1990) to approximate these continuous state inequality constraints.

Let $\boldsymbol{\sigma} = (\boldsymbol{\delta}, \boldsymbol{\upsilon}, \boldsymbol{\gamma})$. For each $k = 1, \ldots, N$, the continuous state inequality constraint (4.4.7) is equivalent to

$$G_k(\boldsymbol{\sigma}) = \sum_{i=1}^{m+1} \int_{i-1}^{i} \max\{\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s \mid \boldsymbol{\sigma})), 0\} ds = 0. \tag{4.5.1}$$

Then, Problem (4.P3) is equivalent to

Problem (4.P4)**.** *Problem (4.P3) with the continuous state inequality constraints (4.4.7) replaced by their respective equality constraints (4.5.1).*

However, the equality constraints (4.5.1) are non-differentiable. For each $k = 1, \ldots, N$, we use the constraint transcription method to construct a smoothing function $\hat{\mathcal{L}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s))$ to approximate the non-smooth function

$$\max\{\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)), 0\}$$

in (4.5.1) as follows.

$$\begin{aligned} &\hat{\mathcal{L}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)) \\ =&\begin{cases} 0, & \text{if } \hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)) < -\varepsilon, \\ (\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)) + \varepsilon)^2/4\varepsilon, & \text{if } -\varepsilon \leq \hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)) \leq \varepsilon, \\ \hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)), & \text{if } \hat{q}_k(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s)) > \varepsilon. \end{cases} \end{aligned} \tag{4.5.2}$$

For any $\varepsilon > 0$, $\hat{\mathcal{L}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s), \hat{\boldsymbol{\Psi}}(s))$, $k = 1, \ldots, N$, are continuously differentiable and they do not always fail to satisfy the constraint qualifications (see Chapter 3 of (Teo, Goh and Wong , 1991)).

For each $k = 1, \ldots, N$, define

$$G_{k,\varepsilon}(\boldsymbol{\sigma}) = \sum_{i=1}^{m+1} \int_{i-1}^{i} \hat{\mathcal{L}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma}))ds. \tag{4.5.3}$$

Problem (4.P4) with (4.5.1) replaced by

$$G_{k,\varepsilon}(\boldsymbol{\sigma}) = 0 \tag{4.5.4}$$

is denoted as Problem (4.P4(ε)).

To relate the solutions of Problem (4.P4) and Problem (4.P4(ε)) as $\varepsilon \longrightarrow 0$, we need some assumptions as

(A1) $\mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}} \neq \emptyset$, where

$$\begin{aligned} &\mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}} \\ =&\{\boldsymbol{\sigma} \in \boldsymbol{\Omega} \times \boldsymbol{\Upsilon} \times \boldsymbol{\Gamma} : \max_{s \in [0,m+1]} \hat{q}_k(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma})) < 0,\ k = 1, \ldots, N\}. \end{aligned}$$

(A2) For any parameter vector $\boldsymbol{\sigma}$ in $\boldsymbol{\Omega} \times \boldsymbol{\Upsilon} \times \boldsymbol{\Gamma}$, there exists a parameter vector $\bar{\boldsymbol{\sigma}} \in \mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}}$ such that

$$\alpha\bar{\boldsymbol{\sigma}} + (1-\alpha)\boldsymbol{\sigma} \in \mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}} \quad \text{for all} \;\; \alpha \in (0,1].$$

Then, we have the convergent result given in the following theorem.

Theorem 4.5.1. *Suppose that $\boldsymbol{\sigma}^*$ is an optimal solution of Problem (4.P4) and $\boldsymbol{\sigma}^*_\varepsilon$ is an optimal solution of Problem (4.P4(ε)). Then,*

$$\lim_{\varepsilon\to 0} \hat{g}_0(\boldsymbol{\sigma}^*_\varepsilon) = \hat{g}_0(\boldsymbol{\sigma}^*).$$

Proof: By assumption (A1) and (A2), there exists a $\bar{\boldsymbol{\sigma}} \in \mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}}$ such that

$$\boldsymbol{\sigma}^\alpha = \alpha\bar{\boldsymbol{\sigma}} + (1-\alpha)\boldsymbol{\sigma}^* \in \mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}}, \;\; \forall\alpha \in (0,1].$$

For any $\varepsilon_1 > 0, \exists$ an $\alpha_1 \in (0,1)$ such that

$$\hat{g}_0(\boldsymbol{\sigma}^*) \leq \hat{g}_0(\boldsymbol{\sigma}^\alpha) \leq \hat{g}_0(\boldsymbol{\sigma}^*) + \varepsilon_1, \; \forall\alpha \in (0,\alpha_1).$$

Choose $\alpha_2 = \alpha_1/2$. Then, it is clear that $\boldsymbol{\sigma}^{\alpha_2} \in \mathring{\boldsymbol{\Omega}} \times \mathring{\boldsymbol{\Upsilon}} \times \mathring{\boldsymbol{\Gamma}}$. Thus, there exists a $\varepsilon_2 > 0$ such that

$$\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}^{\alpha_2}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma}^{\alpha_2})) < -\varepsilon_2, \quad \forall s \in [0, m+1], \;\; k = 1, \ldots, N.$$

If we choose $\varepsilon = \varepsilon_2$, then $G_{k,\varepsilon}(\boldsymbol{\sigma}^{\alpha_2}) = 0, \;\; k = 1, \ldots, N$. Hence, $\boldsymbol{\sigma}^{\alpha_2}$ is an available parameter vector for Problem (4.P4(ε)). Thus, it follows that

$$\hat{g}_0(\boldsymbol{\sigma}^*_\varepsilon) \leq \hat{g}_0(\boldsymbol{\sigma}^{\alpha_2}).$$

Then, we have

$$\hat{g}_0(\boldsymbol{\sigma}^*) \leq \hat{g}_0(\boldsymbol{\sigma}^*_\varepsilon) \leq \hat{g}_0(\boldsymbol{\sigma}^*) + \varepsilon_1.$$

Letting $\varepsilon \to 0$ and noting that $\varepsilon_1 > 0$ is arbitrary, the conclusion of the theorem follows.

Since the equality constraints (4.5.4) fail to satisfy any constraint qualification, Problem (4.P4(ε)) cannot be solved efficiently by any gradient-based optimization technique. Thus, we shall introduce a further approximation as defined below.

Problem (4.P4(ε, β))**.** *Given dynamical system (4.4.1), (4.4.4) and (4.4.5), find a feasible parameter from* $(\delta, \sigma, \gamma) \in \mathbf{\Omega} \times \mathbf{\Upsilon} \times \mathbf{\Gamma}$ *such that the cost function (4.4.6) is minimized subject to the constraints*

$$-\beta + G_{k,\varepsilon}(\boldsymbol{\sigma}) \leq 0, \; k = 1, \ldots, N. \tag{4.5.5}$$

The following theorem is crucially important, justifying the significance and usefulness of the approximate problem (4.P4(ε, β)).

Theorem 4.5.2. *For any* $\varepsilon > 0$*, there exists a* $\beta(\varepsilon) > 0$ *such that for all* β*,* $0 < \beta < \beta(\varepsilon)$*, if a parameter vector* $\boldsymbol{\sigma}$ *is such that the inequality constraints (4.5.5) are satisfied, then the constraint (4.5.1) are also satisfied.*

Proof: Since $\mathbf{\Omega} \times \mathbf{\Upsilon} \times \mathbf{\Gamma}$ is compact, it follows from (4.4.4) and (4.4.5) that for any $s \in [0, m+1]$ and any $\boldsymbol{\sigma} \in \mathbf{\Omega} \times \mathbf{\Upsilon} \times \mathbf{\Gamma}$, $\hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma})$ and $\hat{\mathbf{\Psi}}(s|\boldsymbol{\sigma})$ is bounded. Then, for each $k = 1, \ldots, N$, and for any $s \in [0, m+1]$, $\boldsymbol{\sigma} \in \mathbf{\Omega} \times \mathbf{\Upsilon} \times \mathbf{\Gamma}$,

$$\frac{d\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\mathbf{\Psi}}(s \mid \boldsymbol{\sigma}))}{ds}$$

is bounded. That is, there exists a positive constant η_k such that, for all $\boldsymbol{\sigma} \in \mathbf{\Omega} \times \mathbf{\Upsilon} \times \mathbf{\Gamma}$,

$$\left| \frac{d\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\mathbf{\Psi}}(s \mid \boldsymbol{\sigma}))}{ds} \right| \leq \eta_k, \quad a.e. \;\; s \in [0, m+1].$$

Next, for any $\varepsilon > 0$, define

$$\rho_{k,\varepsilon} = \frac{\varepsilon}{16}\text{min}\{m+1, \frac{\varepsilon}{2\eta_k}\}$$

Assume that there exists a $\boldsymbol{\sigma} \in \boldsymbol{\Omega} \times \boldsymbol{\Upsilon} \times \boldsymbol{\Gamma}$ such that

$$-\beta + G_{k,\varepsilon}(\boldsymbol{\sigma}) \leq 0$$

for any β such that $0 < \beta < \rho_{k,\varepsilon}$ but

$$G_k(\boldsymbol{\sigma}) > 0.$$

Since $\hat{q}_k$ is a continuous function of s almost everywhere in $[0, m+1]$, it implies that there exists a $\bar{s} \in [0, m+1]$ such that

$$\hat{q}_k(\bar{s}, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s \mid \boldsymbol{\sigma})) > 0.$$

Again by continuity, there exists an interval $I_k \subset [0, m+1]$ containing $\bar{s}$ such that

$$\hat{q}_k(s, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s \mid \boldsymbol{\sigma})) > -\varepsilon/2, \qquad \forall s \in I_k.$$

Thus, the length $|I_k|$ of the interval I_k must satisfy

$$|I_k| \geq \min\{m+1, \frac{\varepsilon}{2\eta_k}\}.$$

From the definition of $G_{k,\varepsilon}(\boldsymbol{\delta}, \boldsymbol{\upsilon}, \gamma)$ and the fact that $\hat{\boldsymbol{\mathcal{L}}}_{k,\varepsilon}$ is nonnegative, we have

$$\begin{aligned}
0 \geq & -\beta + G_{k,\varepsilon}(\boldsymbol{\sigma}) = -\beta + \sum_{i=1}^{m+1} \int_{i-1}^{i} \hat{\boldsymbol{\mathcal{L}}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma}))ds \\
\geq & -\beta + \int_{I_k} \hat{\boldsymbol{\mathcal{L}}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma}))ds \\
\geq & -\beta + \min_{s \in I_k}\{\hat{\boldsymbol{\mathcal{L}}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s|\boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s|\boldsymbol{\sigma}))ds\}\, |I_k| \\
\geq & -\beta + \min\{m+1, \frac{\varepsilon}{2\eta_k}\} * \varepsilon/16 = -\beta + \rho_{k,\varepsilon}.
\end{aligned}$$

This is a contradiction, because $\beta < \rho_{k,\epsilon}$. Thus, the proof is complete.

Now, the equality constraints (4.5.1) are approximated by

$$\hat{G}_{k,\varepsilon,\beta}(\boldsymbol{\sigma}) = \beta + \sum_{i=1}^{m+1} \int_{i-1}^{i} \hat{\mathcal{L}}_{k,\varepsilon}(s, \hat{\boldsymbol{\mu}}(s \mid \boldsymbol{\sigma}), \hat{\boldsymbol{\Psi}}(s \mid \boldsymbol{\sigma}))ds \leq 0, \tag{4.5.6}$$

where $k = 1, \ldots, N$, $\beta > 0$ is the parameter for adjusting the feasibility of the solution, while $\varepsilon > 0$ is the parameter for adjusting the accuracy of the solution.

For each $\varepsilon > 0$ and $\beta > 0$, Problem (4.P4(ε, β)) is an optimal parameter selection problem subject to canonical inequality constraints, where $\hat{\boldsymbol{\Psi}}(s)$ is determined by a system of differential equations in matrix form. We shall re-define the variables of the systems of differential equations (4.2.1a) and (4.2.1c) and rewrite these systems together as a system of standard ordinary differential equations in vector form.

Let $\boldsymbol{x}(s)$ be the vector formed by $t(s)$, $\hat{\boldsymbol{\mu}}(s)$ and the independent components of the matrix $\hat{\boldsymbol{\Psi}}(s)$, *i.e.*,

$$\boldsymbol{x}(s) = [t(s), \hat{\boldsymbol{\mu}}^{\mathsf{T}}(s), \hat{\psi}_{11}(s), \ldots, \hat{\psi}_{1n}(s), \hat{\psi}_{22}(s), \ldots, \hat{\psi}_{2n}(s), \ldots, \hat{\psi}_{nn}(s)]^{\mathsf{T}}. \tag{4.5.7}$$

Let $\boldsymbol{f}$ be the corresponding vector obtained from the right hand sides of (4.4.1a), (4.4.4a) and (4.4.5a). Furthermore, let Φ_0, $\mathcal{L}_0$ and $\mathcal{L}_{i,\varepsilon}$, $i = 1, ...N$, be obtained from $\hat{\Phi}_0$, $\hat{\mathcal{L}}_0$ and $\hat{\mathcal{L}}_{k,\varepsilon}$, $k = 1, ...N$, respectively, with $t(s)$, $\hat{\boldsymbol{\mu}}(s)$ and $\hat{\boldsymbol{\Psi}}(s)$ appropriately replaced by $\boldsymbol{x}(s)$.

Then, for each ε and β, Problem (4.P4(ε, β)) is equivalent to

Problem (4.P5(ε, β))**.** *Given the dynamical system*

$$d\boldsymbol{x}(s)/ds = \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}) \tag{4.5.8a}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}^0 \tag{4.5.8b}$$

$$\boldsymbol{x}(i^+) = \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma}), \quad i = 1, \ldots, m, \tag{4.5.8c}$$

where $\boldsymbol{x}^0$ *and* $\boldsymbol{\psi}^i$ *are obtained from (4.4.1b), (4.4.4b), (4.4.5b) and (4.4.4c), (4.4.5c), respectively, find a feasible parameter* $\boldsymbol{\sigma} \in \boldsymbol{\Omega} \times \boldsymbol{\Upsilon} \times \boldsymbol{\Gamma}$*, such that the cost function*

$$\hat{g}_0(\boldsymbol{\sigma}) = \Phi_0(\boldsymbol{x}(m+1 \mid \boldsymbol{\sigma}), \boldsymbol{\sigma}) + \sum_{j=1}^{m+1} \int_{j-1}^{j} \mathcal{L}_0(s, \boldsymbol{x}(s \mid \boldsymbol{\sigma}), \boldsymbol{\sigma}) ds, \qquad (4.5.9)$$

is minimized subject to the constraints (4.4.3) and

$$\hat{G}_{k,\varepsilon,\beta}(\boldsymbol{\sigma}) = \beta + \sum_{j=1}^{m+1} \int_{j-1}^{j} \mathcal{L}_{k,\varepsilon}(s, \boldsymbol{x}(s \mid \boldsymbol{\sigma}), \boldsymbol{\sigma}) ds \leq 0, \quad k = 1, \ldots, N. \quad (4.5.10)$$

To solve Problem (4.P5(ε, β)) as a mathematical programming problem, we need the gradients of cost function and constraint functions. They can be obtained by using similar idea as that given for Theorem 5.2.1 of (Teo, Goh and Wong, 1991). Details of these gradients are presented below in the following two theorems.

Theorem 4.5.3. *The gradient of the cost function (4.5.8) with respect to* $\boldsymbol{\sigma}$ *are given by*

$$\nabla_{\boldsymbol{\sigma}} \hat{g}_0(\boldsymbol{\sigma}) = \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} + \sum_{i=1}^{m} (\boldsymbol{\lambda}^0(i^+))^{\mathsf{T}} \frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} + \sum_{j=1}^{m+1} \int_{j-1}^{j} \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} ds, \qquad (4.5.11)$$

where the Hamiltonian H_0 *is defined by*

$$H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = \mathcal{L}_0(s, \boldsymbol{x}(s), \boldsymbol{\sigma}) + (\boldsymbol{\lambda}(s))^{\mathsf{T}} \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}), \qquad (4.5.12)$$

and $\boldsymbol{\lambda}^0(s)$ *is the co-state determined by the following differential equations*

$$\frac{d\boldsymbol{\lambda}(s)}{ds} = -\left[\frac{\partial H_0(s, \boldsymbol{x}(s), \boldsymbol{\lambda}(s), \boldsymbol{\sigma})}{\partial \boldsymbol{x}}\right]^{\mathsf{T}}, \qquad (4.5.13a)$$

with terminal conditions:

$$\boldsymbol{\lambda}(m+1) = \left[\frac{\partial \boldsymbol{\Phi}_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \boldsymbol{x}}\right]^{\mathsf{T}}, \qquad (4.5.13b)$$

and jump conditions:

$$\boldsymbol{\lambda}(i^-) = \left[\frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \boldsymbol{x}}\right]^{\mathsf{T}} \boldsymbol{\lambda}(i^+). \qquad (4.5.13c)$$

Proof: Differentiating (4.5.8a) with respect to $\boldsymbol{\sigma}$ yields:

$$\frac{d}{ds}\left\{\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k}\right\} = \frac{\partial \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma})}{\partial \boldsymbol{x}(s)} \frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k} + \frac{\partial \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma})}{\partial \sigma_k}. \tag{4.5.14a}$$

The initial condition for (4.5.14a) is

$$\frac{\partial \boldsymbol{x}(0)}{\partial \sigma_k} = 0, \tag{4.5.14b}$$

and the jump conditions are

$$\frac{\partial \boldsymbol{x}(i^+)}{\partial \sigma_k} = \frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \boldsymbol{x}(i^-)} \frac{\partial \boldsymbol{x}(i^-)}{\partial \sigma_k} + \frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \sigma_k}, \quad i = 1, \ldots, m. \tag{4.5.14c}$$

Recall that the cost function is given by

$$\begin{aligned} \hat{g}_0(\boldsymbol{\sigma}) =& \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma}) \\ &+ \sum_{j=1}^{m+1} \int_{j-1}^{j} \left\{ H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma}) - (\boldsymbol{\lambda}^0(s))^{\intercal} \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}) \right\} ds. \end{aligned} \tag{4.5.15}$$

Taking the gradient of $\hat{g}_0(\boldsymbol{\sigma})$ with respect to $\boldsymbol{\sigma}$, we obtain

$$\begin{aligned} \frac{\partial \hat{g}_0(\boldsymbol{\sigma})}{\partial \sigma_k} =& \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \boldsymbol{x}(m+1)} \frac{\partial \boldsymbol{x}(m+1)}{\partial \sigma_k} + \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \sigma_k} \\ &+ \sum_{j=1}^{m+1} \int_{j-1}^{j} \left\{ \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \boldsymbol{x}(s)} \frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k} + \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \sigma_k} \right. \\ &+ \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \boldsymbol{\lambda}^0(s)} \frac{\partial \boldsymbol{\lambda}^0(s)}{\partial \sigma_k} - \frac{\partial (\boldsymbol{\lambda}^0(s))^{\intercal}}{\partial \sigma_k} \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}) \\ &\left. -(\boldsymbol{\lambda}^0(s))^{\intercal} \left(\frac{\partial \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma})}{\partial \boldsymbol{x}(s)} \frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k} + \frac{\partial \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma})}{\partial \sigma_k} \right) \right\} ds. \end{aligned} \tag{4.5.16}$$

By (4.5.12) and (4.5.14a), we can show that

$$\begin{aligned} \frac{\partial \hat{g}_0(\boldsymbol{\sigma})}{\partial \sigma_k} =& \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \boldsymbol{x}(m+1)} \frac{\partial \boldsymbol{x}(m+1)}{\partial \sigma_k} + \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \sigma_k} \\ &+ \sum_{j=1}^{m+1} \int_{j-1}^{j} \left\{ \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \boldsymbol{x}(s)} \frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k} + \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \sigma_k} \right. \\ &\left. -(\boldsymbol{\lambda}^0(s))^{\intercal} \frac{d}{ds} \left(\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k} \right) \right\} ds. \end{aligned} \tag{4.5.17}$$

Applying integration by parts to the last term of (4.5.17), it follows that

$$\begin{aligned}\frac{\partial \hat{g}_0(\boldsymbol{\sigma})}{\partial \sigma_k} = & \frac{\partial \Phi_0(\boldsymbol{x}(m+1),\boldsymbol{\sigma})}{\partial \boldsymbol{x}(m+1)}\frac{\partial \boldsymbol{x}(m+1)}{\partial \sigma_k} + \frac{\partial \Phi_0(\boldsymbol{x}(m+1),\boldsymbol{\sigma})}{\partial \sigma_k} \\ & + \sum_{j=1}^{m+1}\int_{j-1}^{j}\left\{\left(\frac{\partial H_0(s,\boldsymbol{x},\boldsymbol{\lambda}^0,\boldsymbol{\sigma})}{\partial \boldsymbol{x}(s)} + \left(\frac{d\boldsymbol{\lambda}^0(s)}{ds}\right)^{\intercal}\right)\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k}\right. \\ & \left. + \frac{\partial H_0(s,\boldsymbol{x},\boldsymbol{\lambda}^0,\boldsymbol{\sigma})}{\partial \sigma_k}\right\} ds - \sum_{j=1}^{m+1}(\boldsymbol{\lambda}^0(s))^{\intercal}\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k}\Bigg|_{s=(j-1)^+}^{s=j^-}. \qquad (4.5.18)\end{aligned}$$

The second summation appearing in (4.5.18) can be rewritten as

$$\begin{aligned}\sum_{j=1}^{m+1}(\boldsymbol{\lambda}^0(s))^{\intercal}\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k}\Bigg|_{s=(j-1)^+}^{s=j^-} = & \sum_{j=1}^{m}[(\boldsymbol{\lambda}^0(j^-))^{\intercal}\frac{\partial \boldsymbol{x}(j^-)}{\partial \sigma_k} - (\boldsymbol{\lambda}^0(j^+))^{\intercal}\frac{\partial \boldsymbol{x}(j^+)}{\partial \sigma_k}] \\ & -(\boldsymbol{\lambda}^0(0))^{\intercal}\frac{\partial \boldsymbol{x}(0)}{\partial \sigma_k} + (\boldsymbol{\lambda}^0(m+1))^{\intercal}\frac{\partial \boldsymbol{x}(m+1)}{\partial \sigma_k}. \qquad (4.5.19)\end{aligned}$$

By (4.5.13c) and (4.5.14c), the summation in (4.5.19) can be simplified as

$$\begin{aligned}& \sum_{j=1}^{m}[(\boldsymbol{\lambda}^0(j^-))^{\intercal}\frac{\partial \boldsymbol{x}(j^-)}{\partial \sigma_k} - (\boldsymbol{\lambda}^0(j^+))^{\intercal}\frac{\partial \boldsymbol{x}(j^+)}{\partial \sigma_k}] \\ = & \sum_{j=1}^{m}\left[(\boldsymbol{\lambda}^0(j^+))^{\intercal}\frac{\partial \boldsymbol{\psi}^j(\boldsymbol{x}(j^-),\boldsymbol{\sigma})}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}(j^-)}{\partial \sigma_k}\right. \\ & \left. -(\boldsymbol{\lambda}^0(j^+))^{\intercal}\left(\frac{\partial \boldsymbol{\psi}^j(\boldsymbol{x}(j^-),\boldsymbol{\sigma})}{\partial \boldsymbol{x}(j^-)}\frac{\partial \boldsymbol{x}(j^-)}{\partial \sigma_k} + \frac{\partial \boldsymbol{\psi}^j(\boldsymbol{x}(j^-),\boldsymbol{\sigma})}{\partial \sigma_k}\right)\right] \\ = & -\sum_{j=1}^{m}(\boldsymbol{\lambda}^0(j^+))^{\intercal}\frac{\partial \boldsymbol{\psi}^j(\boldsymbol{x}(j^-),\boldsymbol{\sigma})}{\partial \sigma_k}. \qquad (4.5.20)\end{aligned}$$

Then, by (4.5.20), (4.5.13b) and (4.5.14b), (4.5.19) can be simplified as

$$\begin{aligned}\sum_{j=1}^{m+1}(\boldsymbol{\lambda}^0(s))^{\intercal}\frac{\partial \boldsymbol{x}(s)}{\partial \sigma_k}\Bigg|_{s=(j-1)^+}^{s=j^-} = & -\sum_{j=1}^{m}(\boldsymbol{\lambda}^0(j^+))^{\intercal}\frac{\partial \boldsymbol{\psi}^j(\boldsymbol{x}(j^-),\boldsymbol{\sigma})}{\partial \sigma_k} \\ & + \frac{\partial \Phi_0(\boldsymbol{x}(m+1),\boldsymbol{\sigma})}{\partial \boldsymbol{x}(m+1)}\frac{\partial \boldsymbol{x}(m+1)}{\partial \sigma_k}. \qquad (4.5.21)\end{aligned}$$

Substituting (4.5.21) to (4.5.18) and applying (4.5.13a), we obtain

$$\begin{aligned}\frac{\partial \hat{g}_0(\boldsymbol{\sigma})}{\partial \sigma_k} = \frac{\partial \Phi_0(\boldsymbol{x}(m+1), \boldsymbol{\sigma})}{\partial \sigma_k} &+ \sum_{i=1}^{m} (\boldsymbol{\lambda}^0(i^+))^\mathsf{T} \frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \sigma_k} \\ &+ \sum_{j=1}^{m+1} \int_{j-1}^{j} \frac{\partial H_0(s, \boldsymbol{x}, \boldsymbol{\lambda}^0, \boldsymbol{\sigma})}{\partial \sigma_k} ds, \qquad (4.5.22)\end{aligned}$$

This completes the proof.

We can also obtain the gradients of the constraint functions (4.5.10) in a similar way.

Theorem 4.5.4. *For each $k = 1, \ldots, N$, the gradient of the constraint function (4.5.7) with respect to $\boldsymbol{\sigma}$ is given by*

$$\nabla_{\boldsymbol{\sigma}} \hat{G}_k(\boldsymbol{\sigma}) = \sum_{i=1}^{m} (\boldsymbol{\lambda}^k(i^+))^\mathsf{T} \frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} + \sum_{i=1}^{m+1} \int_{i-1}^{i} \frac{\partial H_k(s, \boldsymbol{x}, \boldsymbol{\lambda}^k, \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} ds, \qquad (4.5.23)$$

where the Hamiltonian H_k is defined by

$$H_k(s, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma}) \;= \mathcal{L}_{k,\varepsilon}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}) + (\boldsymbol{\lambda}(s))^\mathsf{T} \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{\sigma}), \qquad (4.5.24)$$

and $\boldsymbol{\lambda}^k(s)$ is the co-state determined by the following system of differential equations

$$\frac{d\boldsymbol{\lambda}(s)}{ds} = -\left[\frac{\partial H_k(s, \boldsymbol{x}(s), \boldsymbol{\lambda}(s), \boldsymbol{\sigma})}{\partial \boldsymbol{x}}\right]^\mathsf{T}, \qquad (4.5.25a)$$

with terminal conditions:

$$\boldsymbol{\lambda}(m+1) = \mathbf{0}, \qquad (4.5.25b)$$

and jump conditions:

$$\boldsymbol{\lambda}(i^-) = \left[\frac{\partial \boldsymbol{\psi}^i(\boldsymbol{x}(i^-), \boldsymbol{\sigma})}{\partial \boldsymbol{x}}\right]^\mathsf{T} \boldsymbol{\lambda}(i^+). \qquad (4.5.25c)$$

The proof is similar to that given for Theorem 4.5.3.

4.6 Numerical Example

In this section, we solve an example using our proposed methods as follows. Consider the dynamic system defined on $(0, 1]$ with the coefficients given by

$$\boldsymbol{A} = \begin{pmatrix} 0.6+\delta & \delta \\ 1.5 & -0.5-\delta \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 2+\sin(t) \\ -2-\cos(t) \end{pmatrix}, \boldsymbol{D} = \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}.$$

The mean and the covariance matrix of the initial state are

$$\boldsymbol{\mu}^0 = \begin{pmatrix} -0.1 \\ -0.5 \end{pmatrix}, \boldsymbol{\Psi}^0 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.2 \end{pmatrix}.$$

The coefficients of two jump functions are

$$\boldsymbol{J}^i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \boldsymbol{K}^i = \begin{pmatrix} 0.36 & 0 \\ 0 & 0.36 \end{pmatrix}, \quad \forall i = 1, 2.$$

The coefficients of the cost function are

$$\boldsymbol{S}_2 = \begin{pmatrix} 2.5 & 0 \\ 0 & 2.5 \end{pmatrix}, \boldsymbol{S}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{S}_0 = 0,$$

$$\boldsymbol{Q}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \boldsymbol{Q}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{Q}_0 = 0,$$

$$\varphi(\boldsymbol{\gamma}) = \sum_{i=1}^{2} \frac{1}{2} (\boldsymbol{\gamma}^i)^\intercal \boldsymbol{\gamma}^i.$$

There is one probabilistic state constraint with the coefficients given by

$$\boldsymbol{c}^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, a = -3, b = 3, \alpha = 0.9.$$

We apply the solution procedure presented in previous sections to solve this example, where the corresponding version of the Problem (4.P5(ε, β)) is solved using the optimal control software package MISER3.3 (Jennings, Fisher, Teo and Goh, http://www.cado.uwa.edu.au/miser), which was implemented in FORTRAN 77. The optimal parameter obtained is $\delta^* = -0.692009$. The optimal cost function value obtained is $g_0^* = 11.2821699$. The first jump appears at time $\tau_1^* = 0.28796$ with the corresponding magnitude vector $\boldsymbol{\gamma}^{1,*} = [-1.01096, 1.40555]^\intercal$. The second jump appears at time $\tau_2^* = 0.69132$ with the corresponding magnitude vector $\boldsymbol{\gamma}^{2,*} = [-1.43826, 0.756040]^\intercal$.

The optimal trajectories of the mean and covariance processes are illustrated in Figure 4.1 and Figure 4.2. For the simulation of $\xi(t)$, we have obtained 500 samples in Matlab. The results are given in Figure 4.3 and Figure 4.4.

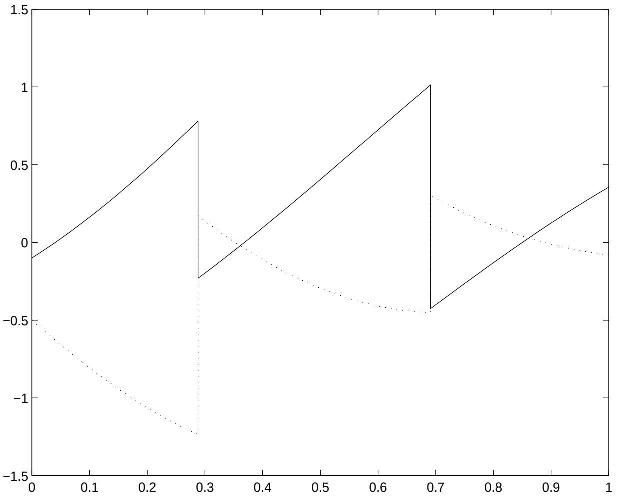

Figure 4.1: Solid line: $\mu_1(t)$; dotted line: $\mu_2(t)$.

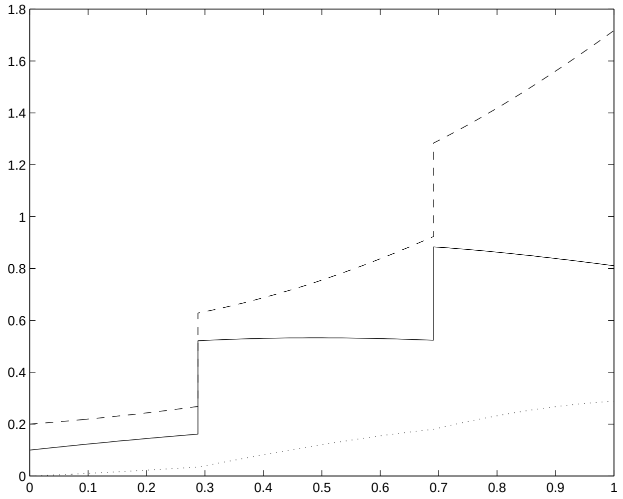

Figure 4.2: Solid line: $\Psi_{11}(t)$; dashed line: $\Psi_{22}(t)$; dotted line: $\Psi_{12}(t)$.

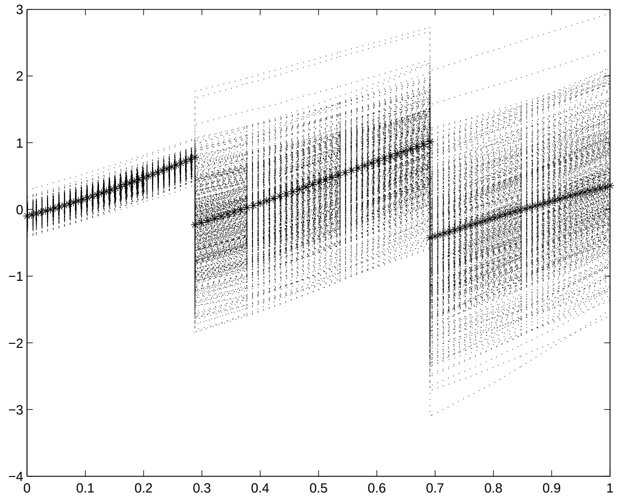

Figure 4.3: * line: $\mu_1(t)$; dotted line: 500 samples of $\xi_1(t)$.

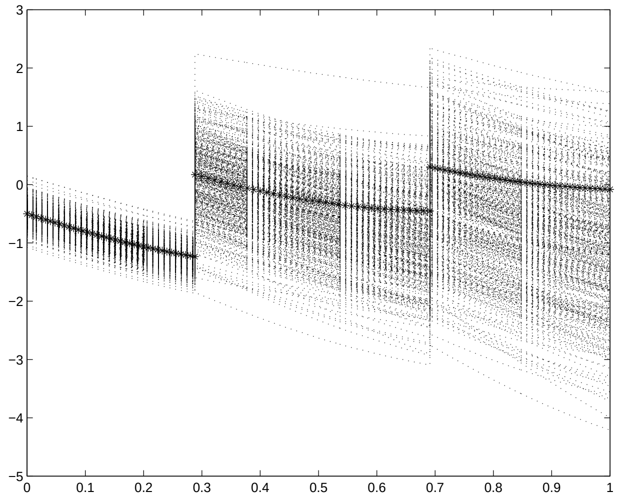

Figure 4.4: * line: $\mu_2(t)$; dotted line: 500 samples of $\xi_2(t)$.

Chapter 5

Conclusions and Suggestions for Future Studies

For nonparametric regression with jump points, we proposed a new two-stage method for solving spline approximating regression problem with jump points. First, we detect the rough locations of jump points based on the kernel method. Then, we introduce a time scaling transform to reformulate our regression problem as a nonlinear optimization problem which is easy to solve using any gradient-base method. Also, some numerical results are presented and the result obtained showed that our proposed method is efficient.

For nonparametric regression with jump curve, a two-stage method was proposed to construct an approximating surface with jump location curve from a set of observed data which are corrupted with noise. This method is developed based on a nonparametric kernel method, a space scaling transform, tensor product cubic splines approximation using least square error criterion, and inverse space transformation. A numerical example was solved using the method proposed.

For stochastic optimal control problems with state jumps, we considered a class of stochastic optimal parameter selection problems involving an impulsive dynamical system subject to probabilistic constraints on the state. We have shown that this stochastic optimal impulsive parameter selection problem with probabilistic constraints is equivalent to a deterministic impulsive optimal parameter selection problem with continuous state inequality constraints. A numerical method was developed for solving this equivalent constrained deterministic impulsive optimal parameter selection problem. From the numerical study through solving a numerical example, we see that the solution method is effective.

The following are some future research projects directly related to the current research.

1. Projects related to Chapter 2 are:

    (i) Application of the results obtained in Chapter 2 to the study of time series analysis in economics;

    (ii) Extension of the results obtained in Chapter 2 to problems subject to constraints arising from various practical specifications;

    (iii) Further statistical analysis so as to gain further understanding of the properties of the break points. This knowledge will be useful for constructing a more efficient method to detect the unknown break points more accurately.

2. Projects related to Chapter 3 are:

    (i) Application of the results obtained in Chapter 3 to the study of image processing;

    (ii) Further statistical analysis so as to gain further understanding of the properties of the jump curves. This knowledge will be useful for constructing a more efficient method to determine the jump curves more accurately.

(iii) Extension of the results obtained in Chapter 3 to cases involving multiple jump curves.

3. Projects related to Chapter 4 are:

   (i) Extension of the results obtained in Chapter 4 to the case involving discrete time system;

   (ii) Extension of the results obtained in Chapter 4 to the case involving impulsive system subject to Poisson disturbances;

   (iii) Extension of the results obtained in Chapter 4 to the case involving control in the problem specification.

Bibliography

[1] Aberkane S., Ponsart J. C. and Sauter D., Output feedback H-infinity control of a class of stochastic hybrid systems with wiener process via convex analysis, International Journal of Innovative Computing, Information and Control, 2(6): 1179-1196, 2006.

[2] Akaike H., A new look at the statistical model identification, IEEE Transaction on Automatic Control, 19: 716-723, 1974.

[3] Ahmed N. U. and Georgenas N. D., On optimal parameter selection, IEEE Transactions on Automatic Control, 18: 313-314, 1973.

[4] Anton H. and Rorres C., Elementary linear algebra : applications version, Wiley, New York, 2000.

[5] Basin M., Sanchez E. and Martinez-Zuniga R., Optimal linear filtering for systems with multiple state and observation delays, International Journal of Innovative Computing, Information and Control, 3(5): 1309-1320, 2007.

[6] Berman A. and Plemmons J. R. Nonnegative Matrices in the Mathematical Sciences, Philadelphia : Society for Industrial and Applied Mathematics, 1994.

[7] Bhandarkar S. M., Zhang Y., and Potter W. D., An edge detection technique using genetic algorithm-based optimization, Pattern Recognition 27: 1159-1180, 1994.

[8] Boehm W., Farin G., and Kahman J., A survey of curve and surface methods in CAGD, Computer Aided Geometric Design, 1: 1-60, 1984.

[9] Boltyanskii V. G., Mathematical methods of optimal control. Holt, Rinehart and Winston, New York , 1971

[10] Brooks H., The topology of surprises in technology, institutions and development, in: Clark, W.C., Munn , R.E., Sustainable development of the biosphere, Cambridge University Press, Cambridge, UK, 1985.

[11] Canny J., A Computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8: 679-698, 1986.

[12] Carl de Boor., A Practical Guide to Splines, Springer-Verlag, New York Inc., 2001.

[13] Chen M. H., Lee D., and Pavlidis T., Residual analysis for feature detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13: 30-40, 1991.

[14] Chu C. K. and Marron J. S., Choosing a kernel regression estimators, Statistical Science, 6:404-419, 1991.

[15] Chu C. K., Glad I. K., Godtliebsen F., and Marron J. S., Edge-preserving smoothers for image processing, Journal of the American Statistical Association 93: 526-556. 1998.

[16] Craven P. and Wahba G. Smoothing noisy data with spline functions, Numerical Mathematics, 31: 377-403, 1979.

[17] Dierckx P., Curve and surface fitting with splines, Oxford, England, Clarendon, 1993.

[18] Dolezal J., On the solution of optimal control problems involving parameters and general boundary conditions, Kybernetika, 17:71-81, 1981.

[19] Duchon J., Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, Constructive Theory of Functions of Several Variables, Springer-Verlag, Berlin-Heidelberg, 1977.

[20] Ermoliev Y. M. and Norkin V. I., On nonsmooth and discontinuous problems of stochastic systems optimization, European Journal of Operational Research, 101(2): 230-244, 1997.

[21] Feng Z. G., Teo K. L. and Zhao Y., Branch and bound method for sensor scheduling in discrete time, Journal of Industrial and Management Optimization, 1(4): 499-512, 2005.

[22] Feng Z. G., Teo K. L., Ahmed N. U., Zhao Y. and Yan W. Y., Optimal fusion of sensor data for Kalman filtering, Discrete and Continuous Dynamical Systems, 14: 483-503, 2006.

[23] Feng Z. G., K. L. Teo and Rehbock V., Hybrid method for a general optimal sensor scheduling problem in discrete time, Automatica, appear in 2008.

[24] Ferguson D. R., Construction of curves and surfaces using numerical optimization techniques, Computer Aided Design 18(1): 15-21, 1986.

[25] Friedman A., Stochastic differential equations and applications, Dover Publications, 2006.

[26] Gijbels I., Peter H. and Kneip A., On the estimation of jump points in smooth curves, Annales Institue of Statistics and Mathematics, 51: 231-251, 1999.

[27] Goh C. J. and Teo K. L., On constrained stochastic optimal parameter selection problems. Bulletin of the Australian Mathematical Society, 41: 393-405, 1990.

[28] Gonzalez R. C., and Woods R. E., Digital image processing, Addison-Wesley Publishing Company, Inc., 1992.

[29] Hall P. and Raimondo M., Approximating a line thrown at Random onto a grid, The Annals of Applied Probability, 7: 648-665, 1997.

[30] Haralick R. M., Digital step edges from zero crossing of second directional derivatives, IEEE Transactions on Pattern Analysis and Machine Intelligence, 6: 58-68, 1984.

[31] Hofer E. and Sagirow, P. Optimal systems depending on parameters, AIAA Journal, 953-953, 1968.

[32] Holling C. S., Resistance of ecosystems: local surprise and global change, in: Clark, W.C., Munn , R.E., Sustainable development of the biosphere, Cambridge University Press, Cambridge, UK, 1985.

[33] http://ccvweb.csres.utexas.edu/ccv/projects/angstrom/sd/shape.php

[34] http://www.quantlet.com/mdstat/scripts/mm3/pdf/mm3pdf.pdf

[35] Huang C. S., Wang S. and Teo K. L., Solving Hamilton-Jacobi-Bellman equations by a modified method of characteristics, Nonlinear Analysis, Theory, Methods and Applications, 40:279-293, 2000.

[36] Hurvich C. M. and Tsai C.L., Regression and time series model selection in small samples, Biometrika 76: 297-307, 1989.

[37] Hurvich C. M. and Tsai C. L., Bias of the corrected AIC criterion for underfitted regression and time series models, Biometrika, 78: 499-509, 1991.

[38] Jennings L. S. and Teo K. L., A computational algorithm for functional inequality constrained optimization problems, Automatica, 26: 371-375, 1990.

[39] Jennings L. S., Fisher M. E., Teo K. L. and Goh C. J., MISER3 Optimal Control Software Version 3.0 Theory and User Manual, Website http://www.cado.uwa.edu.au/miser.

[40] Julious S., Inference and estimation in a change-point regression problem, Journal of Royal Statistical Society, Series D, 50: 51-61, 2001.

[41] Koo J. Y., Spline estimation of discontinuous regression functions, Journal of Computational and Grahpical Statistics, 6: 266-284, 1997.

[42] Korostelev A. P., and Tsybakov A. B., Minimax theory of image reconstruction, Lecture Notes in Statistics, Vol. 82, Springer, New York, 1993.

[43] Lee D., Coping with discontinuities in computer vision: their detection, classification, and measurement, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12: 321-344, 1990.

[44] Lee H. W. J., Teo K. L., Rehbock V. and Jennings L.S., Control parameterization enhancing transform technique for time optimal control problems, Dynamic Systems and Applications, 6: 243-261, 1997.

[45] Liu C. M., Feng Z. G. and Teo K. L., On a class of stochastic impulsive optimal parameter selection problems, International Journal of Innovative Computing, Information and Control, to appear in 2008.

[46] Liu C. M., Wu C. Z. and Teo K. L., A two-stage method for surface spline regression with Jump Curves, submitted.

[47] Lombard F., Detecting change points by Fourier analysis, Technomerics, 30: 305-310, 1988.

[48] Marr D., and Hildreth E., Theory of edge detection, Proceedings of the Royal Society of London 207: 187-217, 1980.

[49] McDonald A. and Owen A. L., Smoothing with Split Linear Fits, Technometrics, 28: 195-208, 1986.

[50] McQuarrie, A. D. R., and Tsai, C.-L., Regression and Time Series Model Selection, Singapore River Edge, N.J., World Scientific,1998.

[51] Miyata S. and Shen X., Adaptive free-knot splines, Journal of Computational and Grahpical Statistics, 12: 197-213, 2003.

[52] Muller H. G., Change-points in nonparametric regression analysis, Annales Statistics, 20: 737-761, 1992.

[53] Muller H. G. and Song K. S., Maximin estimation of multidimensional boundaries, Journal of the Multivariate Analysis, 50: 265-281, 1994.

[54] Muller H. G. and Song K., Two-stage change-point estimators in smooth regression models, Statistics and Probability Letters, 34: 323-335, 1997.

[55] Oksendal B., Stochastic Differential Equations: An Introduction with Applications, Springer-Verlag, New York, 2003.

[56] O'Sullivan, F. and Qian, M., A Regularized contrast statistic for object boundary estimation-implementation and statistical evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16: 561-570, 1994.

[57] Perona P., and Malik J., Scale space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12: 629-639, 1990

[58] Pontryagin L. S., Boltyanskii V.G., Gamkrelidze R. V. and Mishchenko E. F., The mathematical theory of optimal processes, Wiley Interscience, New York, 1962.

[59] Potra F. A. and Liu X., Protein image alignment via tensor product cubic splines, Optimization Methods and Software, 22(1): 155-168, 2007.

[60] Qiu P., and Bhandarkar S. M., An Edge detection technique using local smoothing and statistical hypothesis testing, Pattern Recognition Letters, 17: 849-872, 1996.

[61] Qiu, P. Nonparametric estimation of jump surface, Sankhya, Series A, 59: 268-294, 1997.

[62] Qiu, P. and Yandell B., Jump detection in regression Surfaces, Journal of Computational and Graphical Statistics, 6: 332-354, 1997.

[63] Qiu P., Discontinuous regression surfaces fitting, The Annals of Statistics, 26: 2218-2245, 1998.

[64] Qiu P., A Nonparametric procedure to detect jumps in regression surfaces, Journal of Computational and Graphical Statistics, 11(4): 799-822, 2002.

[65] Rade L. and Westergren B., Mathematics handbook, Studentlitteratur and Chartwell-Bratt, 1988.

[66] Reid D. W. and Teo K. L., Optimal parameter selection of parabolic systems, Mathematics of Operations Research, 5: 467-474, 1980.

[67] Schwarz, G. Estimating the dimension of a model, The Annals of Statistics. 6(2):461-464, 1978.

[68] Situ R., Theory of stochastic differential equations with jumps and applications: Mathematical and Analytical Techniques with Applications to Engineering, Springer, New York, 2005.

[69] Shao Q. and Campbell N., Modelling trends in groundwater levels by segmneted regression with constraints, Australian and New Zeland Journal of Statistics, 44: 129-141, 2002.

[70] Stark P. A., Introduction to numerical methods, Macmillan USA, 1992.

[71] Takeuchi Y., Optimazation of linear observations for the stationary Kalman filter based on a generalized water filling theorem, International Journal of Innovative Computing, Information and Control, 4(1): 211-230, 2008.

[72] Tam C. K. W., Kurbatskii K. A., A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes, Journal of Computational Physics, 157(2): 588-617, 2000.

[73] Tan H. L., Gelfand S. B. and Delp E. J., A comparative cost function approach to edge detection, IEEE Transaction on Systems, Man, and Cybernetics, 19: 1337-1349, 1989.

[74] Tan H. L., Gelfand S. B., and Delp E. J., A cost minimization approach to edge detection using simulated annealing, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14: 3-18, 1991.

[75] Tanikawa A., On a smoother for descrete-time linear stochastic systems with unknown disturbances, International Journal of Innovative Computing, Information and Control, 2(5): 907-916, 2006.

[76] Teo K. L. and Ahmed N. U., Optimal feedback control for a class of stochastic systems, International Journal of Systems Science, 5: 357-365, 1974.

[77] Teo K. L. and Goh C. J., A Computational method for combined optimal parameter selection and optimal control problems with general constraints, Journal of the Australian Mathematical Society Series B., 30: 350-364, 1989.

[78] Teo K. L., Goh C. J. and Wong K. H., A Unified Computational Approach to Optimal Control Problems, Longman Scientific and Technical, Wiley, New York, 1991.

[79] Teo K. L., Jennings L.S., Lee H.W.J. and Rehbock V., The Control prametrization enhancing transform for constrained optimal control problems, Journal of Australia Mathematical Society, Series B, 40: 314-335, 1999.

[80] Tong H. and Lim K. S., Threshold autoregression, limit cycles and cyclical data, with discussions, Journal of Royal Statistical Society, B 42: 245-292, 1980.

[81] Wahba G., Spline models for observational data, Philadelphia: Society for Industrial and Applied Mathematics, 1990.

[82] Wang S., Jennings L. S. and Teo K. L., Numerical solution of Hamilton-Jacobi-Bellman equations by an upwind finite volume method, Journal of Global Optimization, 27: 177-192, 2003.

[83] Wang. Y., Jump and sharp cusp detection by wavelets, Biometrika, 82: 385-397, 1995.

[84] Wang Y., Change curve estimation via wavelets, Journal of the American Statistical Association, 93: 163-172, 1998.

[85] Wu C. Z., Liu C. M., Teo K. L. and Shao Q. X., A New two-stage method for nonparametric regression with jump points. In Chaos Control for Circuits and Systems: A Practical Approach, Nonlinear Sciences Series of World Scientific Publishing, to appear in 2008.

[86] Wu J. S. and Chu C. K., Kernel-type estimators of jump points and values of a regression function, Annales Statistics, 21: 1545-1566, 1993.

[87] Yin Y. Q., Detection of the number, locations and magnitudes of jumps, Communications in Statistics-Stochastic Models, 4: 445-455, 1988.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.