Josif A. Boguslavskiy
Mark Borodovsky
Editor

Dynamic Systems Models

New Methods of Parameter and State Estimation

Springer

Dynamic Systems Models

Josif A. Boguslavskiy

Dynamic Systems Models

New Methods of Parameter and State Estimation

Mark Borodovsky
Editor

Springer

Josif A. Boguslavskiy (deceased)
State Scientific Research Institute
of Automated Systems
Moscow
Russia

ISBN 978-3-319-04035-6 ISBN 978-3-319-04036-3 (eBook)
DOI 10.1007/978-3-319-04036-3
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014930189

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Introduction

Many applications are reduced to a numerical solution to the problem of parameter estimation. Parameters determine how a model of the real dynamic system functions. Examples of such unknown (or not exactly known) parameters may be the initial conditions and the coefficients of the differential equations of the model.

Let the observation vector $Y, \| y_0, y_1, \ldots, y_k, \ldots \|$, be recorded as a function of the discrete-time points $t_0, t_1, \ldots, t_k, \ldots$ These vectors depend on θ—the vector of unknown parameters. An unknown θ vector occurs, for example, in inverse and boundary problems, as well as when the structure of a mathematical model of a physical phenomenon or a system is known from the general laws of natural science, but the parameters of the model are not exactly known. A posteriori data determine the functional dependence of $y_k = F(t_k, \theta)$. Typically, such a relationship is not explicitly defined.

The traditional method of solving the problem of estimation is based on the numerical determination of the global extremum for the decision function $J(\ldots)$, which depends on vectors of observations and vectors of estimation: $J(y_0, y_1, \ldots, y_k, \ldots, \hat{\theta}(y_0, y_1, \ldots))$.

The choice of estimation method determines the type of decision function $J(\ldots)$. Thus, if the nonlinear least squares method applies, then the decision function $J(\ldots)$ is the sum of squares of differences between the components of the actual observations and the components of the calculated observations, which correspond to some approximation to the actual observations. In problems of estimating the parameters of statistical systems, the decision function $J(\ldots)$ is often tasked with appointing a probabilistic likelihood function, which is the approach to the occurrence probability vector's actual observations for the dynamical system and its initial conditions.

The traditional definition of vector estimates—delivering a global extremum of the decision function $J(\ldots)$—produced an iterative process based on some version of Newton-Raphson gradient methods. The process of computing has difficulties that arise because of the existence of multiple relative extrema of the decision function $J(\ldots)$; convergence of computing is achieved only when "guessing" a

good first approximation. At each step of the calculation process, one needs linearization functions that represent a dynamic system requiring the existence of derivatives and the "good" of the local structure functions. There is no information about the expected accuracy of estimation.

The general case to achieve global optimization of the decision function $J(\ldots)$ in the form of a probabilistic likelihood function does not guarantee small estimation errors (the fundamental monograph [1, Chap. 1] states, "Contrary to assert many textbooks estimation method maximum likelihood is not universally gut procedure and shall not be dogmatic").

The iteration process requires choosing the vector of first approximation, while the sequence linearization of the decision function requires calculating the sequence of matrices required by the inverse Jacobian of the decision function. Thus, another vector estimation affects all subsequent vector approximations at each iteration step.

The goal of this book is to present a new method of estimation [2, 3] that does not involve the difficulties mentioned so far in calculations. The method does not require global optimization for the sum of squared residuals, which is used in iterations to select plausible values for the components of assessments.

The proposed method does not require global optimization of the decision function. The sequence of the vectors of approximations is replaced by a sequence a priori and a posteriori of the regions of the parameter vector. Linearization and the definition of the derivative method are not required, and new vector estimation has little influence on all the vector approximations in each iteration step.

The structure of this book can be divided into two parts.

The first part of the book contains Chaps. 1 and 2, which form the theoretical foundations of the new method of estimation. Chapters 3–9 make up the second part of the book; they provide numerous examples of effective parameter estimation in nonlinear situations for the problems of the analysis of observations and the synthesis of optimal control.

The main lemma is presented in Chap. 1. The lemma defines the construction of the proposed algorithm-estimator, which is designated MPA, for the multipolynomial approximation algorithm, in Chap. 9.

Let W vectors be arbitrary given functions in the elements of Y random vectors of possible observations. The moments of discrete time $t_0, t_1, \ldots$ are input to the algorithm-estimator, which receives values $w_0, w_1, \ldots$—components of one of the realizations of the random vector W. The possible implementations of Y depend on the possible realizations of the random parameter vector θ and match some points of Ω_Y. Then the possible implementations of W correspond to the points of Ω_W.

The following form is an expression for the estimation vector of θ, which is optimal in the mean square sense:

$$\hat{\theta}(w, E(\theta), E(W), Q, L, W)^o = E(\theta) + \Lambda^o(w - E(W)), \tag{0.1}$$

where

$$\Lambda^o Q = L,$$

w is a realization for a component of W, corresponding to some moment of discrete time,

$E(\theta), E(W), Q, L$ are the a priori first and second statistical moments for a joint distribution of random θ, W, respectively.

The algorithm-estimator uses the values $w_0, w_1, \ldots$. The algorithm builds a vector estimating $\widehat{\theta}(\ldots)$, linear in $w_0, w_1, \ldots$ and optimal in the mean square. The a priori data constructing an algorithm are the first and second statistical moments of the random vectors θ, W.

The model of a dynamic system generates the emergence of the information on the joint distribution of vectors θ, W, sufficient to determine the first and second statistical moments. The moments are determined using the Monte Carlo method or by numerically determining the appropriate integrals via a modified trapezoid method.

The basic lemma formulates sufficient conditions for a semidefinite matrix of the difference between the estimation error covariance matrix for an arbitrary algorithm-estimator and that for our proposed algorithm-estimator, which depends on some weight matrix.

The proposed algorithm-estimator should be regarded as the best algorithm-estimator since its estimation error covariance matrix is "no more" than a covariance matrix generated by any other estimator. The weight matrix Λ^o satisfies a matrix linear algebraic equation. The solution of the equation exists to meet the relevant conditions of regularity. The decision of the equation does not require determining the inverse matrix, and the recursion procedure is implemented. This statement follows from the principle of the decomposition of observations, which is presented in Chap. 1.

In Chap. 1, we show that for linear discrete dynamical systems, the proposed algorithm-estimator is similar to the standard Kalman filter for the solution to the recurrent filtration problem. The algorithm-estimator uses the similarity of the Kalman filter algorithm to smooth in mean square problems of interpolation and prediction.

In Chap. 2, we believe that the components of W are linear combinations of products as great as m degrees of the components of Y-terms of the form

$$y_{i_1}^{a_1} \cdots y_{i_m}^{a_m}. \tag{0.2}$$

All m terms $a_1, \ldots, a_m$ for the sum of powers of the products are nonnegative solutions. The integer inequalities $a_1 + \ldots \imath + a_m \leq d$, where the sum of m variables of degree does not exceed a given integer d. The vector W forms a countable sequence of polynomials in $y_0, y_1, \ldots$ for $m \to \infty, d \to \infty$.

In Chap. 2, the counting sequence is called the base.

Further, we believe that the next fundamental basic supposition is true: In this problem, the Weierstrass–Stone theorem is applicable and components of a vector of the conditional expectations of the parameters (vector $E(\theta|Y)$) can be arbitrarily small errors of approximation of linear combinations of polynomials of the sequence of W polynomials that is constructed. The approximation is optimal in the quadratic mean. It implements the algorithm-estimator from Chap. 1. The linear combinations of members of the basic sequence, which the algorithm delivered in Chap. 1, define arbitrarily small errors of the estimate components of vector $E(\theta|Y)$. Chapter 2 provides a fragment of the modified trapezoid method. The algorithm and the modified trapezoid method's program are used to calculate the values of multidimensional integrals required in determining the statistical moments.

Chapter 3 considers the multipolynomial approximations of a set of continuous function $F(Y_N), N$ variables via the algorithm-estimator introduced in Chap. 1, if the integer d represents the maximum sum of degrees given by the approximating multipolynomials.

Let's suppose $\theta = F(Y_N)$, where Y_N is a random vector, given by a distribution on the compact set Ω_Y. Then the algorithm-estimator realizes the approximate representation:

$$\hat{\theta} = \sum_{a_1+\ldots+a_N \le d} \lambda(a_1, \ldots, a_m), y_1^{a_1} \cdots y_N^{a_N}. \tag{0.3}$$

The magnitudes $\lambda(a_1, \ldots, a_N)$ depend on $F(\ldots, \Omega_Y, d, N)$ and on the statistical moments for random Y_N, θ.

Let's note that in calculus a function of real variables is frequently represented as a segment of a power series. The series is a many-dimensional segment of some Taylor series. But the construction of the series demands the definition of derivatives of the corresponding order. Area convergence of the series and errors of approximation are established separately; it is complicated to review this.

The algorithm-estimator from Chap. 1 is free of these difficulties. The definition of derivatives is absent, a series converges in regular intervals on a compact set of definitions of the function $F(Y_N)$, and the variance of random approximation errors is calculated.

Examples of applications of multipolynomial approximations include the following:

1. Detection of the multipolynomial term: In the set sequence, it is necessary to discover the multipolynomial that possesses the maximum sum of degrees of multipolynomial representations. The algorithm-estimator solves the task: The necessary term of the sequence corresponds to close to a zero variance of errors of approximation.

2. Approximation errors by restrictions on derivatives: For an equation that models the dynamic system, which has restrictions on the magnitudes of derivatives, the influence of nonlinearity on the approximation of the components of a state vector needs to consider how the entry conditions function. Linear approximation errors are appreciable and sharply decrease for a squared approximation.
3. Polynomial approximation for optimal terminal control: Let the dynamic model of the system be

$$\dot{X} = \varphi(X, u(t)), \tag{0.4}$$

where X is the state vector, and $u(t)$ is a scalar control, continuous on $[0, T]$. We need to find the optimal control $|u(t)^o| < 1$:

$$u(t)^o = argmax_{|u(t)| \le 1}(F(X(T)), \tag{0.5}$$

where $F(\ldots)$ is a given function and $max(F(X(T))$ is a terminal criterion for optimization.

Let $u(t)$ be Bernstein's polynomial of degree $n-1$:

$$u(t) = \sum_{k=0}^{n-1} \theta_{k+1} C_{n-1}^k (t/T)^k (1 - (t/T)^{n-1-k}). \tag{0.6}$$

If $|\theta_1| \le 1, \ldots, |\theta_n| \le 1$, then $|u(t)| < 1$.

Solution of the terminal optimization arises in the numerical solution of the sequence of routine tasks of nonlinear programming, which makes the maximum terminal optimization criterion. Under the constraints $|\theta_k| < 1, k = 1, \ldots, n$, the sequence of the optimal parameters $\theta_1^o, \ldots, \theta_n^o$ is determined, performing terminal condition (3.1) for the function $u(t)^o$.

Solution of the terminal optimization becomes much simpler if the model is linear. Let the model have the form

$$\dot{X} = AX + Bu, \tag{0.7}$$

where A and B are the same matrices. If $X(0) = 0$, and $u(t)$ is as defined in Eq. (0.6), then a correct equality is

$$X(T, n) = \sum_{k=0}^{n-1} \theta_{k-1} x_k(T, n),$$

where a vector $x_k(T, n)$ is determined by numerical integration of (3.3) on segment $[0, T]$ and are $u_i(t) = 0, i = 0, \ldots, k-1, k+1, \ldots, n$, $u_k(t) = C_{n-1}^k (t/T)^k (1 - t/T)^{n-1-k}, 0 < t < T$. Then the problem of terminal optimization goes into a simple problem of determining an extreme for the function of the parameters $\theta_1, \ldots, \theta_n$ under the constraints $|\theta_k| < 1$.

In Chap. 4, we consider the use of multipolynomial approximation to determine the feedback vector-functions. There are many applications of situations where, for instance, we are given a domain of unknown parameters, $\Theta \in \Omega_\theta \in R^n$, and the observation vector Y is of the form $Y = F(\theta), Y \in R^N$. We need to find the inverse function: $F^{-1}(Y) = \theta$. Typically, Ω_θ is a box in R^n. If the area of Ω_θ is not small, inclusion of the nonlinearity function $F(\ldots)$ requires a fairly large number of members in the representation Eq. (0.1), which indicates a sufficiently large number d in Eq. (0.1). There are difficulties with calculations because of the large condition number of certain matrices. A radical way to reduce the number d is a member of iterations for the computation process. The a priori region Ω_θ is divided into a region that is less than the a priori area. At each subarea, multipolynomial approximations implemented autonomous and a small number of d. For the next iteration-selected subregion, which corresponds to a small vector discrepancy. We present a method to use radical solutions for the under-and overdetermined systems of linear algebraic equations, the systems of nonlinear algebraic equations, on the right sides of which are integer powers of the components of the vector Y or trigonometric and differentiable functions of the "module" type.

The numerical solution of the boundary value problem for the system of nonlinear differential equations is an example of a situation where, in the equation for the inverse function $Y = F(\theta)$, the function $F(\ldots)$ is defined implicitly and by numerical integration of the equations of motion.

Let the model of order $2n$ be of the form

$$dx/dt = f(x, t),$$

where $x_1(0) = \theta_1, \ldots, x_n(0) = \theta_n, x_1(T) = Y_1, \ldots, x_n(T) = Y_n, \ldots, x_{2n} = Y_{2n}$. If $f(x, t)$ is a linear function, then $\theta_i = (\hat{\theta})_i$ and the numerical solution of the problem is trivial. Let $F(x, t)$ be a nonlinear function: $f(x, t) = Ax - 0.01\, \dot{I}_{2n} \dot{x} * ^3$, A - $2n \times 2n$ matrix, I is the identity matrix $2n \times 2n$, and $x*^3$ a vector, whose components are cubes of the components of the vector x.

In Chap. 5, we present a method to design an organized search multipolynomial approximation for the solution of several applications of parameter estimation with nonlinear dynamic systems. We split the a priori existence of a box of parameters into a set of parallelepipeds with small edge lengths. By successive search, we define a small enough box for which the value of the decision function takes the extreme value. Proof of the method was carried out via the following:

- parameter estimation of the Van der Pol equation according to Bellman [4];
- identification of parameters of nonlinear systems for which the following are uncertain: parameters, forces, proportional movement, third degree of movement, damping parameter of speed, setting the dry friction force;
- smoothing and filtering for model integration with nonlinear feedback;
- identification of speed characteristic of the integration c nonlinearity type a gap;

- identification of parameter tracking system with a relay drive and a hysteresis loop;
- estimation of the principal moments of inertia of the solid.

In Chap. 6, the nonlinear filtering is implemented if the navigational parameters of a moving object should be based on limited information—without measuring the distance—and on noise. Information without the noise generates nonlinear differential equations, satisfied by the trigonometric functions for angles determining the orientation of the beam in the visual space. The end of the visual ray is a point on the object for which the nonlinear filtering algorithm assesses the current state of the vector components.

Option 1: Autonomous navigation

The property is a celestial body (such as satellites), and a visual beam is generated by the optical electronic system on the surface of Earth. It is sometimes necessary to define the navigation settings without measuring the distance; for instance, this occurs if the equipment onboard the satellite radio fails. The input to the algorithm-estimator is the sum of the results of an original observation sequence of pairs of angles defining the orientation of the visual ray.

Option 2: Autonomous navigation

The property is a brilliant point (BP) on the surface of Earth. This point and the visual ray are generated by the radiation of optical electronic systems onboard the aircraft. The situation arises where it is necessary to determine—without an onboard range-finder—the navigation parameters of the aircraft's movement relative to the BP. This scenario may occur if, for example, it is necessary to make an emergency landing in LA not far from BP in the Arctic or in a jungle or need to lose weight on the ground is not prepared in BP. The input to the algorithm-estimator is similar to that in option 1. The algorithm for option 2 is characterized by having not only the current navigation parameters, but also errors and two orientation angles in the vector in the local vertical coordinate system board.

The simulation showed that the estimation errors are small for all estimated values after a few iterations.

Chapter 7 described the use of the algorithm-estimator to estimate the parameters of stochastic systems with different structures.

In Sect. 7.1, we consider the estimation of parameters of hidden Markov models (HMMs). In recent years, these models have been used in the statistical analysis of biological sequences of nucleotides (DNA) and sequences of speech. The sequences are derived from experimental data.

In Sect. 7.7 of Chap. 7, we believe there is a random process with a finite number of states and continuous time. As is known, the state probabilities satisfy Kolmogorov's ordinary differential equation, for which the a priori data is the intensity of the states. We consider the situation with some unknown intensity that is estimated from observations.

In Sect. 7.8 of Chap. 7, we believe it is necessary to assess the current state of the Markov process if they are observed with additive errors. The statement of the problem is close to the formulation of the problem in Chap. 9 in [5].

The monograph offers no constructive solution to the problem, as requires consideration of the infinite-dimensional system of stochastic differential equations.

In Sect. 7.9 of Chap. 7, we describe the principles underlying a multipolynomial approximation algorithm, and we use the algorithm to estimate the parameters of control time series generated by STGARCH and MGARCH models (in the BEKK specification). These models are used in nonlinear problems of financial mathematics.

Usually, problems like those in Sect. 7.1–Sect. 7.8 use variants of the method of the maximum likelihood function for which the criticism in [1, 6] is given above.

The simulation in Chap. 7 showed that the estimation of parameters of stochastic systems successfully implemented the proposed algorithm-estimator. Thus, HMM parameter estimation via $d = 3, m(d, N) = 968, N = 16$ (the approximations of the algorithm-estimator use linear combinations of polynomials of degree 1–3 at number addends components are 968) showed that almost all 45 relative errors of solution of the inverse problem are smaller than 0.1.

Chapter 8 outlines the methodology of designing an optimal control for a limited linear dynamic system. Vector control sends the vector of the initial state of the system to a given vector of the final state of the system in the shortest time. Pontryagin's optimality principle is used to determine the necessary and sufficient conditions for vector control. Management is optimally by speed of acting.

The equations of motion of the system are complemented by the differential equations for the vector of conjugate variables, whose initial conditions are some of the trigonometric functions vector of unknown parameters. These features provide equal one length of the initial vector conjugate variables. Specified conditions for the initial and final states of the system of vectors and for the initial conditions of the vector of conjugate variables determine the two-point boundary value problem of synthesis of optimal control. The problem is solved by iterations for the joint operation of the simple search method and the polynomial approximation method.

The approved technique is to design time-optimal control to solve the problem of synthesis of applied fast control of thrusters to correct Sputnik orbit and the satellite's position in orbit. The solution of the latter problem is necessary for docking satellites. The simulation showed that time-optimal control of two thrusters several times reduces correction of the satellite's orbit or time correction of the satellite in orbit. Each component has a diagonal component of the vector $x(t)$ in the third degree. If $n = 3, T = 3, -10 \le \theta_i \le 10, i = 1, 2, 3, d = 7, m(d, N) = 119$, the relative error of the estimate of each component of the parameter vector does not exceed 5×10^{-2}. Each discrepancy is reduced by a factor of more than 10^5 if the a priori domain Ω_θ is reduced by a factor of 10.

Chapter 9 presents the theory and application techniques to correct the results nominal (a priori) data of the aerodynamic coefficients of aircraft; these data are determined by experiments with model aircraft in a wind tunnel. The nominal data

are a priori bank dimensionless characteristics of the aerodynamic forces and moments; the data are determined with some errors. Therefore, the actual characteristics of the aircraft movements (the angles of attack and sliding overload angles of pitch, roll, heading) are measured equipment LA do not coincide with the design characteristics of the movement, which are determined by numerical integration of the equations of flight dynamics of the aircraft. The coefficients of these equations are chosen from a bank of nominal data. Real and design characteristics of the aircraft movements are input to the algorithm-estimator, which corrects the nominal data by estimating the components of the vector of unknown parameters, to yield the vector of experimental errors of nominal data.

Practice calculations to implement the methodology have detected a significant effect on the accuracy of the estimation of random errors from aircraft equipment, the statistical characteristics of which are unknown. Therefore, a regularization procedure is necessary; it will significantly reduce the impact of the errors mentioned. The procedure consists of adding the data from the aircraft equipment random process similar to white noise, whose intensity is selected experimentally. Note that this procedure was referred to earlier in the theory of artificial neural networks.

We use the proposed method to estimate the characteristics of a number of complex nonlinear dynamic systems. In some situations, the method requires the development of appropriate **private receptions computing**. Therefore, the book explains several options for sequential calculation steps. The options are similar to each other, but their details are not the same.

This monograph is an presentation of a novel method for solving inverse problems. They arise at of parameters estimation for time series, data which collected from simulate of real experiments. These time series might be generated by measuring the dynamics of aircraft in flight, might be a function of a hidden Markov model used in bioinformatics or speech recognition, might be at analyzer the dynamics of supply price from the nonlinear models of financial mathematics. The monograph demonstrates the use of algorithms based on polynomial approximation which have more weaker requirements to dynamic models than popular iterative methods. Specifically, they do not require a first approximation for goal function and they allow have non-differentiable elements in the vector functions being approximated. The text covers all the points necessary for the understanding and use the method of polynomial approximation for which the mathematical fundamentals is represented. Inputs of algorithms are data received at mathematical modelling or real experiments—for instance aeroplane flight dynamics or biological sequence analysis. The technical material is illustrated by the use of worked examples and methods for training the algorithms are included.

The monograph “Dynamic Systems Models: New Method ....” provides researchers in aeronautics engineering, bioinformatics and financial mathematics (as well as computer scientists interested in any of these fields) with a reliable and effective numerical method for nonlinear estimation and solving boundary problems.

It will also be of interest to academic researchers and students investigate inverse problems and their solution.

The author thanks the GosNIIAS manual at work for the conditions that led to the creation of the book.

The book is written with the financial support of the Russian Foundation for Basic Research.

References

1. Zaks S (1975) Theory of statistical inferences (World). Wiley, New York
2. Boguslavskiy JA (1994) Recurrent algorithm of optimum estimation. Papers of the Russian Academy of Sciences, No. 4
3. Boguslavskiy JA (1996) Bayesian estimations of nonlinear regression and related questions. Bulletin of the Russian Academy of Sciences. Theory and control systems, No. 4
4. Wiener N (1949) Extrapolation, interpolation, and smoothing of stationary time series. Wiley, New York
5. Boguslavskiy JA, Borodovskiy M. Yu (1998) On detection of states generated by hidden Markov model, Bulletin of the Russian Academy of Sciences. Theory and control systems, No. 3
6. Rubin R et al (1998) Biological sequence analysis. Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge

Chapter 1
Linear Estimators of a Random Parameter Vector

1.1 Linear Estimator, Optimal in the Root-Mean-Square Sense

The method of polynomial approximation is based on the following components: (1) the formation of a vector W characteristic of basic supervision from components of the vector Y_N of actual observations and from polynomial functions of these components; (2) a Bayes approach based on assignment of an a priori stochastic measure to a vector θ of unknown parameters and to a vector Y_N of basic observations. In the polynomial approximation algorithm, assignment of a stochastic measure permits one to use the well-known method of constructing the linear estimator, which is optimal in the root-mean-square sense. The method was laid out in fundamental works by A. Kolmogorov, N. Wiener, and V. Pugachev.

We emphasize that the symbol W is the total set of all possible basic observations determined by a mathematical model and, in particular, a private set of observations, which occurred at a given time interval.

In what follows, we will walk through the steps of the above-mentioned method and present its development in certain directions [1–4].

Let a vector $Y_N \in R^N$ be a random vector of scalar observations, and let $\theta \in R^q$ be a random vector of estimated parameters. If every observation is a vector, then $N = N_1 \times n$, where N_1 is the number of vectors of observations of dimension $n \times 1$.

We believe that a priori statistical data about θ and Y_N exist; these are the first and central second statistical moments of the components of vectors θ, Y_N, represented by the vectors and matrices

$E(\theta)$, $E(Y_N)$,

$$C_0 = E\big((\theta - E(\theta))(\theta - E(\theta))^T\big),$$

$$Q = E\big((Y_N - E(Y_N))(Y_N - E(Y_n))^T\big),$$

$$L = E\big((\theta - E(\theta))(Y_N - E(Y_V))^T\big).$$

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_1,

Let the vector Y_N be fixed. It is well known that the cited a priori data permit us to construct a vector $\hat{\theta}(Y_N, N)^o \in R^q$ to estimate the vector θ. The estimator will be linear in relation to Y_N and optimal in the root-mean-square sense among all vectors $\hat{\theta}(Y_N, N)$, which are linear in relation to Y_N.

We will represent any above-mentioned vector $\hat{\theta}(Y_N, N)$ by the formula

$$\hat{\theta}(Y_N, N) = z + \Lambda(Y_N - E(Y_N)), \tag{1.1}$$

where z is an arbitrary vector of dimensionality $q \times 1$, and Λ is an arbitrary matrix of dimensionality $q \times N$. In (1.1) $E(Y_N)$ is found by averaging the total set of the model observations, and the letter Y_N means a particular set of observations generated by the model or a real experiment for the specified period of time.

Let's assume that the matrices C^o and C are estimation error covariance matrices if $\hat{\theta}(Y_N, N)^o$ and $\hat{\theta}(Y_N, N)$ are estimation vectors:

$$C^o = E((\hat{\theta}(Y_N, N)^o - \theta)(\hat{\theta}(Y_N, N)^o - \theta)^T)$$

and

$$C = E((\hat{\theta}(Y_N, N) - \theta)(\hat{\theta}(Y_N, N) - \theta)^T).$$

For the linear optimal estimation vector $\hat{\theta}(Y_N, N)^o$, the matrix inequality is true:

$$C^o \quad \leq \quad C. \tag{1.2}$$

Lemma 1.1

$$\hat{\theta}(Y_N, N)^o = E(\theta) + \Lambda^o(Y_N - E(Y_N)), \tag{1.3}$$

where

$$\Lambda^o Q = L. \tag{1.4}$$

Proof From Eqs. (1.1) and (1.3), we will find the expressions for error estimation vectors; after averaging these expressions, we will find the matrices C, C^o. Then the identity will be true:

$$C = C^o + (\Lambda - \Lambda^o)Q(\Lambda - \Lambda^o)^T + (\Lambda^o Q - L)(\Lambda - \Lambda^o)^T$$

$$+ (\Lambda - \Lambda^o)(\Lambda^o Q - L)^T + (z - E(\theta))(z - E(\theta))^T. \tag{1.5}$$

The second and sixth matrix terms in Eq. (1.5) are nonnegatively defined matrices. Hence, the equality to zero of the third and fourth matrix terms in Eq. (1.5) for any matrices Λ serves as a sufficient condition for the validity of Eq. (1.2). Lemma 1.1 is proved. □

Consequence 1. If $\Lambda = \Lambda^o$, then

$$C^o = C_0 - \Lambda^o L^T. \tag{1.6}$$

Consequence 2. Let the covariance matrix Q of the random vector Y be nonsingular (all components of Y_N are linearly independent). Then from Eq. (1.4), we have

$$\Lambda^o = LQ^{-1}, \tag{1.7}$$

$$\begin{gathered} C^o = C_0 - LQ^{-1}L^T, \\ \sigma_i^2 = \sigma_i(0)^2 - l_i Q^{-1} l_i^T, \end{gathered} \tag{1.8}$$

where σ_i^2 is a dispersion of errors of estimating the ith component of the vector θ, $\sigma(0)^2$ is an a priori dispersion of this component, and the ith diagonal element of the a priori matrix C_0, l_i is the ith row of the matrix L. Hence, $\sigma_i^2 \leq \sigma_i^2(0)$.

Consequence 3. Let the covariance matrix Q be singular because some components of the random vector Y_N are linear combinations of other components:

$$Y_N^T = \| Y_1^T \cdot (AY_1)^T \cdot Y_2^T \|.$$

Here the random vectors Y_1 and Y_2, having dimensionality $r_1 \times 1$ and $r_2 \times 1$ accordingly, are linearly independent, the matrix A has dimensionality $r_3 \times r_1$, $r_1 + r_2 + r_3 = N$, and the matrix L is divided into blocks

$$L = \| L_1 \cdot (L_1 A^T) \cdot L_2 \|.$$

Direct verification shows that in this case the solution of matrix Eq. (1.4) exists and looks as follows:

$$\Lambda^o = \| L_1 V_{1,1} + L_2 V_{2,1} \cdot 0_{q,r3} \cdot L_1 V_{2,1}^T + L_2 V_{1,1} \|,$$

where $0_{q,r3}$ is a matrix of dimensionality $q \times r3$ whose elements are all equal to zero, $V_{1,1}$ and $V_{2,1}$ are the top left and bottom left blocks of the matrix, respectively, and $V_{2,2}$ is the bottom right block in this matrix, which is the reverse of the matrix of covariances of the random vector $\| Y_1^T \cdot Y_2 T \|)^T \|$.

The formulas, similar to the aforesaid, are also true for the general case, in which the random and linearly independent components and the components that are linear combinations of the previous components alternate.

Consequence 4. The linear optimal estimation vector $\hat{\theta}(Y, N)^o$ is unique. In fact, let there exist estimation vectors—unequal to one another—of form (1.3):

$$\hat{\theta}_1(Y_N, N)^o = E(\theta) + \Lambda_1^o (Y_N - E(Y_N))$$

and

$$\hat{\theta}_2(Y_N, N)^o = E(\theta) + \Lambda_2^o(Y_N - E(Y_N)).$$

Then

$$(\hat{\theta}_1(Y_N, N)^o - \hat{\theta}_2(Y_N, N)^o)^T) = (\Lambda_1^o - \Lambda_2^o)Q(\Lambda_1^o - \Lambda_2^o)^T$$

$$= (L - L)(\Lambda_1^o - \Lambda_2^o)^T = 0.$$

Hence, the vectors $\hat{\theta}_1(Y_N, N)^o$ and $\hat{\theta}_2(Y_N, N)^o$ coincide almost everywhere on elements of the probabilistic space for any weight matrices Λ^o, satisfying Eq. (1.4). The vector of linear optimal estimation is unique.

Consequence 5. Let the vector Y_1 be composed of the first p components of the vector Y_N and be of the form $Y_1 = B\theta$, where B is a nonsingular matrix of dimensionality $p \times p$. Then the optimal (and accurate) estimator of the vector θ will be defined by the equality as follows:

$$\hat{\theta}(Y_N, N)^o = B^{-1}Y_1.$$

This estimator does not depend on the other components of the vector Y_N. Hence, the first q column-vectors of the matrix Λ^o should form a square matrix B^{-1}, and the other elements of Λ^o should be equal to zero. As the vector θ is estimated without errors, then all elements of the error covariance matrix of the estimator C^o, defined by Eq. (1.8), should be equal to zero.

The matrix C^o in Eq. (1.8) is a difference of two matrices whose elements are of the same order of size if the mean errors of estimation are small. Practice with calculations revealed that using Eq. (1.8) often results in a significant loss of accuracy because the calculated diagonal elements of C^o—being dispersions of estimation errors—turn out to be negative.

From the above, it follows that formulas for the weight matrix Λ^o and the error covariance matrix C^o of the optimal linear estimator are reasonable to transform, and so the structure of elements of these matrices should be explicit from them if the formula $Y_1 = B\theta$ is true.

Then we assume that the matrix Q^{-1} exists. Let's suppose that

$$Y_N^T = \|Y_1^T \cdot h^T\|,$$

where Y_1 and h are vectors of dimensionality $q \times 1$ and $(N - q) \times 1$. Let's suppose that an additive part, which is linear in relation to θ, has been extracted from the vector Y_1, and hence there exists a representation $Y_1 = B\theta + v$, where B is a nonsingular matrix of dimensionality $q \times q$, v—a random p-dimensional vector. Then the symmetric matrix Q can be divided into blocks:

$$Q = \begin{pmatrix} P \cdot R^T \\ R \cdot Q(h) \end{pmatrix},$$

where the matrices P, R, $Q(h)$ are of dimensionality $q \times q$, $(N-q) \times q$, $(N-q) \times (N-q)$ accordingly.

The following representations of these matrixes are true:

$$P = E((Y_1 - E(Y_1))(Y_1 - E(Y_1))^T = BC_0B^T + BL(v) + L(v)^T B^T + Q(v),$$

$$Q(h) = E((h - E(h))(h - E(h))^T),$$

$$R = E((Y_1 - E(Y_1))(h - E(h))^T = BL(h) + L(v, h),$$

where

$$\begin{aligned} Q(v) =& E((v - E(v))(h - E(h))^T), \\ L(\theta, v) =& E((\theta - E(\theta))(v - E(v))^T), \\ L(\theta, h) =& E(\theta - E(\theta))(h - E(h))^T), \\ L(v, h) =& E((v - E(v))(h - E(h))^T). \end{aligned}$$

The matrix Q^{-1} can be represented by blocks P_1, Q_1, R_1 that are similar in terms of their dimensionality and position:

$$Q^{-1} = \begin{pmatrix} P_1 \cdot R_1^T \\ R_1 \cdot Q_1 \end{pmatrix}.$$

From the identity

$$QQ^{-1} = I_q,$$

where I_q is an identity matrix of dimensionality $q \times q$, we find the equalities $PP_1 + RR_1^T = I_q$, $PR_1 + RQ_1 = 0$, $R^T R_1 + Q(h)Q_1 = I_{N-q}$. Furthermore,

$$L = \|L_1 \cdot L(\theta, h)\|,$$

where

$$L_1 = E((\theta - E(\theta))(Y_1 - E(Y_1))^T) = C_0B^T + L(\theta, v).$$

Using relationships between blocks of matrixes Q and Q^{-1}, we will find after a number of transformations that

$$\begin{aligned} \Lambda^o = B^{-1} \| & I_q - (L(\theta, v)^T B^T + Q(v))P_1 - L(v, h)R_1^T \\ & - (L(v, h)^T B^T + Q(v))R_1 + L(v, h)Q_1 \|, \end{aligned} \tag{1.9}$$

$$C^o = B^{-1}(-L(\theta, v)^T + ((L(\theta, v)^T B^T + Q(v)P_1$$

$$+ L(v, h)R_1^T)(BC_0 + L(\theta, v)^T)$$

$$+ ((L(\theta, v)^T B^T + Q(v))R_1 + L(v, h)Q_1)L(\theta, h)^T. \quad (1.10)$$

But if $v = \text{const}$, then the matrices $L(\theta, v)$, $Q(v)$, $L(v, h)$ are composed of elements that are equal to zero. Then from Eqs. (1.9), (1.10), it follows that for $v = \text{const}$,

$$\Lambda^o = \|B^{-1}, \quad 0_{N-q}\|, \quad C^o = 0, \quad \hat{\theta}(Y_N, N) = \theta,$$

and hence the qualitative requirements for the formulas for the matrices Λ^o, C^o—formulated above—are held.

1.2 Vector Measure of Nonlinearity of Vector Y_1 in Relation to Vector θ

According to what we saw in Sect. 1.1, it is suitable to select from the first components of the vector Y_N an additive part that would linearly depend on the vector θ. Then, using the received Eqs. (1.9), (1.10), one can reduce the possible calculation errors that arise when one subtracts matrices or vectors with component values close to one another. For this it is enough to compose—from all N components of the vector Y_N—all possible q-dimensional vectors and by enumeration assign Y_1 and the corresponding matrix B, so that the vector of the random difference $v = Y_1 - B\theta$ would be minimal in the sense of some average norm.

Let's plan a possible way of rationally choosing the matrix B if the vector Y_1 has already been composed of, for example, the first n components of the vector Y. To solve the problem at hand, it is enough to construct a linear, optimal in the root-mean-square-sense, estimator $\hat{y}_1(\theta, q)$ of vector Y_1, using a linear function from the vector θ, and to assume the weight matrix *Lambda* to be equal to the required matrix B.

From formulas of forms (1.4) and (1.5), it follows that

$$\hat{Y}_1^o = E(Y_1) + \Lambda(\theta - E(\theta)),$$

where

$$\Lambda = LC_0^{-1}, L = E((Y_1 - E(Y_1))(\theta - E(\theta))^T).$$

From here, it follows that

$$B = LC_0^{-1}.$$

As a mean norm of the estimation error, represented by the difference in $v = Y_1 - B\theta$, one can take any matrix norm from the estimation error covariance matrix $C(v)$, defined by the formula

$$C(v) = E((Y_1 - E(Y_1))(Y_1 - E(Y_1))^T).$$

The greater the variances of errors of estimating vector Y_1 (by means of a linear combination of components of vector θ) are, represented by diagonal elements of matrix $C(v)$, then the more essential, on average, is the nonlinear dependence of vector Y_1 from vector θ. The vector, composed of diagonal elements of $C(v)$, can be called a vector measure of nonlinearity of vector Y_N relative to vector θ, statistically connected with it. If, for some Y_1, this measure is equal to zero, then vector Y_1 is a linear function of the vector of parameters, B, and is estimated without errors.

Generalizing the preceding statement, we will notice that it is possible to similarly define a vector measure of the "square-law characteristic", and so on.

1.3 Decomposition of Path of Observations to the Recurrence Algorithm

Let's consider an algorithm that constructs a vector of estimators $\hat{\theta}(Y_N, N)^\circ$, which permits us to find this vector without calculating elements of the reverse matrix Q^{-1}. The algorithm is based on serially decomposing a vector of observations and will be used in the following in solving problems of nonlinear smoothing (interpolation), filtration, and prediction (extrapolation).

Let's divide vector Y_N into subvectors Y_1 and Y_* of dimensions k and $N - k$:

$$Y_N^T = \| Y_1^T \cdot Y_*^T \|$$

and denote the corresponding blocks of a priori matrices Q, L as follows:

$$Q_1 = E((Y_1 - E(Y_1))(Y_1 - E(Y_1))^T),$$

$$Q_* = E((Y_* - E(Y_*))((Y_* - E(Y_*))^T),$$

$$Q_{*1} = E((Y_* - E(Y_*))((Y_1 - E(Y_1))^T),$$

$$Q = \begin{pmatrix} Q_1 \cdot Q_{*1}^T \\ Q_{*1} \cdot Q_* \end{pmatrix},$$

$$L_1 = E((\theta - E(\theta))(Y_1 - E(Y_1))^T),$$

$$L_* = E((\theta - E(\theta))(Y_* - E(Y_*)),$$

$$L = \|L_1 \cdot L_*\|.$$

Let's assume that a random vector Y_1 is composed of k linearly independent components and, hence, $Q_1 > 0$. After measuring the vector Y_1, we—from formulas of forms (1.3), (1.7)—will find estimators of the vectors θ and Y_*, linear in relation to Y_1 and optimal in the root-mean-square sense:

$$\hat{\theta}(Y_1, k)^o = E(\theta) + L_1 Q_1^{-1}(Y_1 - E(Y_1)),$$

$$\hat{Y_{N*}}(Y_1, k)^o = E(Y_*) + Q_{*1} Q_1^{-1}(Y_1 - E(Y_1)).$$

The following formulas are true (these can be checked directly):

$$Q^* = E((Y_* - \hat{Y}_*(Y_1, k)^o)(Y_* - \hat{Y}_*(Y_1, k)^o)^T = Q_* - Q_{*1} Q_1^{-1} Q_{*1}^T,$$

$$L^* = E((\theta - \hat{\theta}(Y_1, k)^o)(Y_* - \hat{Y}_*(Y_1, k)^o)^T) = L_* - L_1 Q_1^{-1} Q_{*1}^T,$$

$$C^* = E((\theta - \hat{\theta}(W_1, k)^o)(\theta - \hat{\theta}(Y_1, k)^o)^T = C_0 - L_1 Q_1^{-1} L_1^T.$$

Before using the vector Y_* to construct a new estimator of vector θ—linear and optimal in the root-mean-square sense—we will call the above obtained vectors $\hat{\theta}(Y_1, k)^o$, $\hat{Y}_*(Y_1, k)^o$, and the matrices Q^*, L^*, C^* new a priori data about the vectors θ and Y_*. We will emphasize that the new a priori data are not the first two a posteriori moments of vectors θ, Y_*, found after fixing the random vector Y_1.

In essence, these data are the first two statistical moments of the vectors θ, Y^*, whose information arises after using the linear and optimal in the root-mean-square sense estimator—of the vector Y_1.

Let the random vector Y_* be fixed. Then the vector $\hat{\theta}(Y_*, N - k)^o$ of the vector θ's estimator (being linear in relation to Y_*, optimal in the root-mean-square sense and considering new a priori data) will be of the form

$$\hat{\theta}(Y_*, N - k) = \hat{\theta}(Y_1, k)^o + \Lambda_*^o(Y_* - \hat{Y}_*(Y_1, k)^o), \tag{3.1}$$

where Λ_*^o satisfies the matrix equation

$$\Lambda_*^o Q^* = L^*.$$

Such a matrix also exists when matrix Q is a singular matrix (consequence 3 of Lemma 1.1).

Lemma 3.1 *The following equality is true:*

$$\hat{\theta}(Y_N, N)^o = \hat{\theta}(Y_*, N-k)^o. \tag{3.2}$$

Proof One may directly check that the matrix

$$\|(L_1 - \Lambda_*^o Q_{*1} Q_1^{-1} \cdot \Lambda_*)\|$$

satisfies matrix Eq. (1.4); from Eq. (1.3), we will receive another form of writing the linear optimal vector of estimators $\hat{\theta}(Y, N)^o$:

$$\begin{aligned}\hat{\theta}(Y_N, N)^o = E(\theta) + (L_1 - \Lambda_*^o Q_{*1} Q_1^{-1}(Y_1 - E(Y_1))) \\ + \Lambda_*^o(Y_* - E(Y_*)).\end{aligned} \tag{3.3}$$

Substituting in Eq. (3.3)'s expressions for the vectors $\hat{\theta}(Y_1, k)^o$, $\hat{Y}_*(Y_1, k)^o$, we check the validity of Eq. (3.2). Lemma 3.1 has been proved. □

So, upon decomposing a vector of observations, we see that there are two steps to construct an estimation vector. In step 1, one constructs the estimation vectors $\hat{\theta}(Y_1, k)^o$, $\hat{Y}_*(Y_1, k)^o$—linear in relation to Y_1—and the corresponding matrices of covariances. Elements of these vectors and matrices serve as new a priori data before step 2.

In step 2, one constructs the vector $\hat{\theta}(Y_*, N-k)^o$ of estimation vector θ, linear in relation to Y_*. This vector is coincident with the estimation vector $\hat{\theta}(Y_N, N)$ defined by Eq. (1.3), linear in relation to Y and optimal in the root-mean-square sense.

Lemma 3.2 *We will denote residual vectors as*

$$\varepsilon_1 = Y_1 - E(Y_1), \varepsilon_* = Y_\varepsilon - \hat{Y}_*(Y_1, k)^o.$$

The residual vectors $\varepsilon_1, \varepsilon_*$ *have the property of a repeating sequence:*

$$E(\varepsilon_1 \varepsilon_*^T) = 0.$$

Proof Equation (3.4) follows from the given formula for the vector $\hat{Y}_*(Y_1, k)^o$ (see above). Lemma 3.2 has been proved. □

1.4 Recurrent Form of Algorithm for Estimator Vector

Let $Q > 0$. Then direct application of Eqs. (1.4), (1.7) would require inversion of the matrix Q. But all components of the random vector W are linearly independent, and hence, in the formulas of the previous section,

$$Q^* > 0, \Lambda_* = L^*(Q^*)^{-1}.$$

Then, using the decomposition process permits us—upon constructing the vector $\hat{\theta}(Y_N, N)^o$—to invert only those matrices Q_1, Q^* whose dimension is less than that of the matrix Q.

Let Y_1 be a scalar; then the matrix Q_1 is also a scalar, and step 1 of the decomposition process does not require matrix inversion. Repeating step 1 m times and using in each step the new a priori data about the first and second statistical moments, we will find an estimation vector $\hat{\theta}(Y_N, N)^o$ without inversion of matrices. To arrange a recurrent process of calculations with a singular matrix Q, we will prove the following lemma.

Lemma 4.1 *Some components of a random vector Y_* are linear combination components of the vector Y_1 when and only when the corresponding diagonal elements of the matrix Q^* are equal to zero.*

Proof Let

$$y_j = h^T Y_1,$$

where y_j is the jth component of vector Y_*, and h is a nonrandom vector. But

$$h^T Q_1 h = (Q_{*1} Q_1^{-1} Q_{*1}^T)_j, h^T Q_1 h = (Q_*)_j,$$

where the bottom index j marks the jth diagonal element of the corresponding matrices. From here we will obtain

$$(Q^*)_j = 0.$$

The controverse is true: let the variance $(Q^*)_j$ of the random variable $E(Y_*)_j + (Q_{*1} Q_1^{-1} (Y_1 - E(Y_1))_j)$ be equal to zero. But then this random variable is equal to zero almost everywhere. Lemma 4.1 has been proved. □

Let Y_1 be composed of one component and let the matrix Q be singular, but with its first diagonal element positive. If some diagonal elements of the matrix Q^* are equal to zero, then, as follows from Lemma 4.1, the corresponding elements of the vector Y_* linearly depend on Y_1.

Let's exclude these components from the composition of Y_* and exclude the corresponding rows and columns from the matrices Q_*, Q_{*1}, L_*. Then, after step 1, we will obtain a new vector W and new (used in step 2) a priori statistical data, which we, as before, denote as L, Q, C. Then, we regard the new vector's first component as the value W_1, and so on.

The procedure just described defines the algorithm of the same computational procedures, which—as a result of no more than an m-fold application—builds a sequence of scalars Y_1, vectors Y_*, $\hat{Y}_*(Y_1, 1)$, and matrices of the form Q^*, L^*. Before the last procedure, the next vector Y is equal to Y_1 (it is composed of one component), and matrices Q^*, L^* have dimensions 1×1 and $n \times 1$. After the last procedure,

$$\hat{\theta}(Y_*, 1)^o = \hat{\theta}(Y_N, N)^o, \quad C^* = C^o.$$

If $Q > 0$, then in the sequence of m matrices Q^*, all diagonal elements are greater than zero, and the computational procedure is used m times.

Let's represent a formalized description of the recurrent algorithm for $Q > 0$, following from the principle of decomposition of an observation sequence. Let's assume that $y_1, y_2, \ldots, y_N$ are components of the vector Y_N.

The computational process is composed of N consecutive steps. At each step, on the basis of new, updated a priori data, a new estimation of the vector of parameters, θ, is performed; also, the forecast is implemented, which is to estimate the rest of the observation vector. The estimation error covariance matrix, reached at this step, is simultaneously calculated. During the last (Nth) step, there is no forecast, and the last refinement of the estimation vector θ occurs.

For step k of the calculation process, we accept the following notations:

- The vector V_k, of dimensionality $(q + N - k) \times 1$, is composed of n components of the vector θ and $N - k$ components $y_{k+1}, \ldots, y_N$.
- The vector $\hat{V}_k(y_k)$, of dimensionality $(q + N - k) \times 1$, is a linear and optimal in the root-mean-square-sense estimator of the vector V_k after observing the component w_k and all the previous components.
- The scalar $z_{y_{k+1}}(y_k)$ is the $(q + 1)$th component of the vector $\hat{V}_k(y_k)$ (the estimator of component y_{k+i} after observing component y_k and all previous components).
- The vector $(\hat{V}_k(y_k)^1)$, of dimensionality $(q + N - k - 1) \times 1$, is a vector obtained from $\hat{V}_k(y_k)$ by eliminating component $z_{y_{k+1}}(y_k)$ of this vector.
- The matrix $Q_k = E((V_k - \hat{V}_k(y_k))(V_k - \hat{V}_k(y_k))^T)$, of dimensionality $(n + N - k) \times (q + N - k)$, is the estimation error covariance matrix of the vector V_k after observing the component y_k and all previous components.
- The scalar q_k is the $(q + 1)$th diagonal element of the matrix Q_k (estimation error variance of component w_{k+i} after observing component w_k and all previous components).
- The matrix Q_k^1, of dimensionality $(q + N - k - 1) \times (q + N - k - 1)$, is the matrix obtained from the matrix Q_k by eliminating the $(q + 1)$th row vector and the $(q + 1)$th column-vector.
- The vector l_k, of dimensionality $(q + N - k - 1) \times 1$, is the $(q + 1)$th column-vector of matrix Q_k after eliminating the $(q + 1)$th component. The recurrent algorithm is composed of N steps of calculations, in the process of which the vectors $V_1, V_2, \ldots, V_N = \theta$ are consequently estimated by linear and optimal in the root-mean-square-sense functions of components $y_1, y_2, \ldots, y_N$. At step k, the recurrent algorithm's formulas are of the form

$$\hat{V}_{k+1}(y_{k+1}) = \hat{V}_k(y_k)^1 + q_k^{-1} l_k (y_{k+1} - z_{y_{k+1}}(y_k)), \tag{4.1}$$

$$Q_{k+1} = Q_k^1 - q_k^{-1} l_k l_k^T, \tag{4.2}$$

where $k = 0, \ldots, N-1$,

$$V_o^T = \|\theta^T \cdot Y_N^T\|, \hat{V}_0(y_0)^T = E\|\theta^T \cdot Y^T\|, z_{y_1}(y_0) = E(y_1),$$

$$Q_0 = \begin{pmatrix} C_0 & L \\ L^T & Q \end{pmatrix}.$$

At $k = N$, the recurrent algorithm determines the vector

$$\hat{\theta}(Y_N, N) = \hat{V}_N(y_N)$$

of the last estimation vector θ (after observing the last component y_N) and the estimation error covariance matrix.

$$C^o = Q_N$$

is the estimation error covariance matrix.

Let $Y(k)$ be the vector of dimensionality $k \times 1$, composed of the first k components of the vector Y.

The vector $\hat{\theta}(Y(k))$ of dimensionality $q \times 1$, composed of the first q components of the vector $\hat{V}_k(y_k)$, is the linear and optimal in the root-mean-square-sense estimator of the vector θ after observing $Y(k)$, $\hat{\theta}(0) = E(\theta)$.

In matrix Q_k, the top left block C_k of dimensionality $q \times q$ is an error covariance matrix of the estimation vector θ after observing $Y(k)$.

Let $l(k)$ be a vector composed of the k first components of the vector l_k. Then, the formula, representing evolution of the covariance matrix C_k versus the number k of watched components W, will be of the form

$$C_k = C_0 - q_1^{-1} l(1) l(1)^T - \cdots - q_{k-1}^{-1} 1k - 1l(k-1)^T. \tag{4.3}$$

Let

$$\varepsilon_1 = y_1 - z_{y1}(y_0), \ldots, \varepsilon_k = y_k - z_{y_k}(y_{k-1}).$$

Then, from Eq. (3.4), it follows that values of the residuals $\varepsilon_1, \ldots, \varepsilon_k, \ldots$ form an updating sequence of uncorrelated random variables:

$$E(\varepsilon_k \varepsilon_j) = \delta_{k,j} q_k. \tag{4.4}$$

By definition, we have

$$l(k) = E((\theta - \hat{\theta})(Y(k)))\varepsilon_k,$$

and from Eq. (4.1), it follows that

$$\hat{\theta}(Y(k)) = E(\theta) + q_1^{-1} l(1)\varepsilon_1 + \ldots + q_{k-1}^{-1} l(k-1)\varepsilon_{k-1}. \tag{4.5}$$

Subtracting the vector θ from both parts of Eq. (4.5), multiplying by ε_k, and considering Eq. (4.4), we will find, after averaging, a different expression for the vector $l(k)$:

$$l(k) = E((\theta - E(\theta))\varepsilon_k), \tag{4.6}$$

This relationship is essentially used later, upon solving the problem of linear optimal interpolation.

So, at $Q > 0$, the recurrent algorithm implements m steps of the calculation process, during which one constructs a sequence of optimal estimators of the vector parameters θ, linear in relation to components of the vector W. This sequence has the corresponding sequence of estimation error covariance matrices with decreasing diagonal elements.

The recurrent algorithm starts to function once the multidimensional integrals or analytical expressions have determined the a priori first and second statistical moments for θ, Y_N: the vectors $E(\theta)$ and $E(Y_N)$, and the matrices L, Q, and C_0.

Let $Q \geq 0$. Some of the components of the vector Y can be linearly dependent on the component located above. If, after step k, some diagonal elements of the matrix Q_k are equal to zero, then we exclude the column-vectors and row-vectors containing these elements, and from vector Y's composition we exclude the corresponding components, which—according to Lemma 4.1—are linearly dependent on the previous components of this vector. As a result, the number of steps of the recurrent algorithm will become less than m and will be coincident with the value rank Q.

From Eqs. (4.1) and (4.2), it follows that the recurrent algorithm evidently does not require inversion of matrices, but it does include the procedure of a linear, optimal in the root-mean-square-sense forecast of the observation result vector whose dimensionality decreases to 1 beginning with $N - 1$.

From (4.2), it is evident that the covariance matrix Q_k is a difference of two matrices. Due to calculation errors, the matrix may no longer be nonnegatively defined, an indication that will become evident with negative diagonal elements. One can eliminate such an effect of calculation errors.

The structure of Eq. (4.2) coincides with that of the formulas used in numerically solving a system of linear algebraic equations by the method of elimination. Hence, in order to increase the accuracy of calculating the elements of matrices Q_k, one can use some known computational methods, except partial ordering of the main element.

1.5 Problem of Optimal Linear Filtration

Let's consider some problems to be relevant to the estimation—in discrete time—of current state vectors of a linear dynamic system disturbed with discrete white noise. The observations linearly depend on state vectors and contain terms of the type of discrete white noise. Similar problems have been investigated in a large number of publications. However, one often supposes that random vectors of an initial state and

noises have normal distributions (a known theorem of normal correlation [1, 6] is used, or theorems of the theory of orthogonal random processes [7]).

In what follows, we will show that solutions of the above-mentioned problems are delivered by a special case of a recurrent form of the algorithm of linear optimal estimation. Let a linear dynamic system and observations be of the form

$$x_k = a_{k-1}x_{k-1} + \eta_{k-1}, \quad y_k = h_k x_k + \xi_k, \tag{5.1}$$

where a_k is a matrix of dimensionality $q \times q$, the vector x_k of dimensionality $q \times 1$ is a vector indicating the state of the dynamic system at instant k, y_k is a scalar of observations at instant x_k,

$$E(\eta_{k-1}) = 0, E(\xi_k) = 0,$$

$$E((x_0 - E(x_0))(x_0 - E(x_0))^T) = C_0,$$

$$E(\xi_i \xi_k) = \delta_{i,k}\sigma_k^2 \quad (\sigma_k \geq 0),$$

and

$$E(\eta_i \eta_k^T) = \delta_{i,k}\psi_k, E(\eta_i \xi_k) = \delta_{i+1,k}V_k.$$

By results of linear observations in noises, one often labels the problem of optimally (in the root-mean-square sense) estimating the current vector of the state of a linear system, disturbed by white noises, as the "problem of linear filtration". In what follows, this problem is considered a problem of Bayes estimation with a vector of the observations, subsequently equal to the vectors $Y_1, \ldots, Y_k, \ldots$, where Y_k is a vector of observations with components $y_1, \ldots, y_{k-1}, y_k \ldots$.

From the preceding sections (in which the recurrent process of decomposing observations was outlined), it follows that at instant $k-1$ (after observing that random components of the vector Y_{k-1} as the first and second a priori statistical moments of the random vector x_{k-1} serve vectors of the Bayes linear estimator),

$$E_{k-1}(x_{k-1}) = \hat{x}_{k-1}(Y_{k-1})$$

and the estimation error covariance matrix of the vector

$$E((x_{k-1} - E_{k-1}(x_{k-1}))(x_{k-1} - E_{k-1}(x_{k-1}))^T) = C_{x_{k-1}}.$$

Let's find formulas for the given vector and matrix.

At instant $k-1$, the a priori first and second statistical moments have the following appearance:

(1) for random vector x_k,

$$E_{k-1}(x_k) = \hat{x}_k(Y_{k-1}) = a_{k-1}\hat{x}_{k-1}(Y_{k-1}), \tag{5.2}$$

$$E((x_k - E_{k-1}(x_k))(x_k - E_{k-1}(x_k))^T) = a_{k-1}C_{x_{k-1}}a_{k-1}^T + \psi_{k-1}; \quad (5.3)$$

(2) for random variable $y_k = h_k a_{k-1}x_{k-1} + h_k\eta_{k-1} + \xi_k$,

$$E_{k-1}(y_k) = \hat{y}_{k-1}(Y_{k-1}) = h_k a_{k-1}\hat{x}_{k-1}(Y_{k-1}), \quad (5.4)$$

$$E((y_k - E_{k-1}(y_k))(y_k - E_{k-1}(y_k))^T) =$$

$$q(k) = h_k a_{k-1}C_{x_{k-1}}a_{k-1}^T h_k^T$$

$$+ h_k\psi_{k-1}h_k^T + \sigma_k^2. \quad (5.5)$$

At instant $k-1$, a statistical connection of the random vector x_k and of the scalar y_k is represented by the relationship

$$E((x_k - \hat{x}_k(Y_{k-1}))(y_k - \hat{y}_k(Y_{k-1}))) = L(k)$$

$$= a_{k-}C_{x_{k-1}}a_{k-1}^T h^T + \psi_{k-1}h_k^T. \quad (5.6)$$

Thus, we have just presented the statistical characteristics of the random state vector x_k and those of the random variable y_k, a priori before instant k.

Let the instant k have occurred and then observe value y_k. Then, from Eqs. (4.1), (4.2), we will obtain expressions for the vector of linear, optimal in the root-mean-square-sense, estimation vector x_k and for the estimation error covariance matrix:

$$\hat{x}_k(Y_k) = \hat{x}_k(Y_{k-1}) + L(k)q(k)^{-1}(y_k - \hat{y}_k(Y_{k-1})), \quad (5.7)$$

$$C_k = C_{k-1} - L(k)q(k)^{-1}L(k)^T, \quad (5.8)$$

$$k = 1, 2, \ldots.$$

The initial conditions take the shape of

$$\hat{x}_0(Y_0) = E(x_0), \quad C_{x_0} = C_0.$$

Equations (5.2)–(5.8) represent the recurrent algorithm of linear filtration. Its structure repeats the algorithm of the standard Kalman filter in discrete time.

In principle, there can be situations when $q(k) = 0$ at some k, for example, if

$$h_k a_{k-1}C_{x_{k-1}}a_{k-1}^T h_k^T + h_k\psi_{k-1}h_k^T + h_k V_k + V_k^T h_k^T = 0, \sigma_k = 0.$$

According to what we stated earlier, this means that the random variable y_k linearly depends on components of the vector Y_{k-1}, carries no new information about the vector x_k, and should be excluded from the full observation vector.

1.6 Problem of Linear Optimal Recurrent Interpolation (Problem of Optimal Smoothing)

Let's consider a problem of linear optimal recurrent interpolation: In the process of observations $y_1, \ldots, y_k, \ldots$ for the linear dynamic system (5.1), it is necessary to define—as a function of k—an optimal, in the root-mean-square-sense, estimation vector of initial conditions x_0 and an estimation error covariance matrix: vector $\hat{x}_0(Y_k)$ and matrix $Cx_0(k)$. We notice that in [6] this problem is solved on the assumption that random vectors of disturbances and of observations have conditional normal distributions. The solution presented in this section is based on the principle of decomposing a sequence of observations and does not require assumptions on the a priori distribution of random vectors.

The vector x_0 ought to be considered as a vector of unknown parameters θ, and one should use the recurrent formulas presented in Sect. 1.4. From Eqs. (4.1) and (5.2)–(5.8), we will obtain the recurrent relationships for the estimation vector $\hat{x}_0(Y_k)$ and the estimation error covariance matrices $C_{x_0}(k)$:

$$\hat{x}_0(Y_k) = \hat{x}_0(Y_{k-1} + L(x_0, y_k)q(k)^{-1}(y_k - \hat{y}_k(Y_{k-1})), \tag{6.1}$$

$$C_{x_0,k} = C_{x_{0,k-1}} - L(x_0, y_k)q(k)^{-1}L(x_0, y_k)^T, \tag{6.2}$$

$$k = 1, 2, \ldots,$$

at the initial conditions

$$\hat{x}_0 = E(x_0), \quad C_{x_0} = C_0.$$

We will find values $q(k), \quad \hat{y}_k(Y_{k-1})$ at functioning—from 1 to k—of the recurrent formulas (5.2)–(5.8) of the Kalman filter algorithm.

To determine the value $L(x0, yk)$, we will take into account the relationship (4.6), which—in the case under consideration—will acquire the form

$$L(x_0, Y_k) = E((x_0 - E(x_0))(y_k - \hat{y}_k(Y_{k-1}))). \tag{6.3}$$

Because the random vector x_0 is not connected statistically with noise vectors η_j, ξ_i, then upon implementing Eq. (6.3), we consider these noises equal to zero and account for only the statistical characteristics of the random vector x_0. From Eq. (6.3), we will find

$$L(x_0, y_k) = (A_k - B_k)h_k^T, \tag{6.4}$$

where

$$A_k = E((x_0 - E(x_0))x_k^T), \quad B_k = E((x_0 - E(x_0))\hat{x}_k(Y_{k-1})^T)$$

are matrices of dimensionality $n \ldots k$ which are defined by the recurrent relationships

$$A_{k+1} = C_0(a_k \dots a_0)^T, \tag{6.5}$$

$$B_{k+1} = (B_k(I_n - q(k))^{-1}L(k)h_k)^T + (A_k q(k)^{-1}(L(k)h_k)^T)a_k^T, \tag{6.6}$$

$$A_1 = C_0 a_0^T, \quad B_1 = 0, \quad k = 1, 2, \dots.$$

Thus, the recurrent relationships (6.1)–(6.6) fully solve the problem of linear optimal interpolation—at the expense of recurrent formulas of the discrete Kalman filter algorithm.

Let's find the test whose implementation points out that the algorithm and program of optimal linear interpolation are correct. We will assume that $P_k = (a_0 \dots a_{k-1})$ and $\psi_0 = \dots = \psi_{k-1} = 0$.

A problem of linear optimal filtration will be solved if we initially solve a problem of optimal linear interpolation and then, using motion equations, "transpose" the found vector by optimally estimating the vector of initial conditions x_0 to instant k. Hence, the validity of the identity

$$P_k C_{x_0,k} P_k^T = C_k \tag{6.7}$$

follows. The satisfaction of Eq. (6.7) serves as a test for the algorithm of optimal linear interpolation.

Example Let's consider a problem of the optimal estimation of unknown initial conditions of three digital blocks, consequently implementing triple discrete integration of harmonious oscillations of a set frequency in the presence of additive random disturbances at the input of the first integrator, and of random errors of observations at the output of the third integrator. The dynamic system and observations are of form (5.1) under the following conditions:

$$x_k = a_k x_{k-1} + u_k + \eta_{k-1}, \quad y_k = h_k x_k + \xi_k,$$

$$a_k = \begin{pmatrix} 1 \cdot \tau \cdot 0 \\ 0 \cdot 1 \cdot \tau \\ 0 \cdot 0 \cdot 1 \end{pmatrix},$$

$$u_k^T = \|0 \cdot 0 \cdot \sin(10k) \quad \tau\|, \quad \eta(k)^T = \|0 \cdot 0 \cdot \eta_3(k)\|, \quad h_k = \|1 \cdot 0 \cdot 0\|,$$

$$C_0(1,1) = C_0(2,2) = C_0(3,3) = 1/3, \sigma_k^2 = (0.05)^2/3,$$

$$\psi_k(3,3) = (0.05)^2/3, V_k = 0, \tau = 1/10, q = 3.$$

To characterize the accuracy of the optimal linear interpolation, we consider the ratio of current and initial root-mean-square deviations (RMSD) of estimation errors $(C_{x_0,k}(i,i)/C_0(i,i))^{1/2}$ for $i = 1, 2, 3$.

Here are the values of these ratios versus k (multiplied by 100):

$k = 5$	5.999520	27.159232	91.212830
$k = 10$	5.461143	20.283446	39.854941
$k = 15$	4.428993	12.284388	18.930264
$k = 20$	3.871042	9.313944	15.066365
$k = 25$	3.677751	8.687775	14.744050
$k = 30$	3.654168	8.662084	14.723541
$k = 50$	3.646723	8.608962	14.661674
$k = 100$	3.646521	8.607919	14.660844

The given data imply that the estimation algorithm quickly passes to a steady state, at which a further increase in the number of observations does not increase the accuracy of the estimation.

It is evident that in the absence of random disturbances, upon input of the first integrator, an increase in the number of observations leads to an increase in the accuracy of the optimal linear interpolation.

References

1. Boguslavskiy JA (1994) Recurrent algorithm of optimum estimation. Papers of the Russian Academy of Sciences, No. 4
2. Boguslavskiy JA (1996) Bayesian estimations of nonlinear regression and related questions. Bulletin of the Russian Academy of Sciences. Theory and control systems, No. 4
3. Boguslavskiy JA (2001) Bayesian method of numerical solution of nonlinear equations. Bulletin of the Russian Academy of Sciences. Theory and control systems, No. 2
4. Boguslavskiy JA (2005) Integrated method of numerical decision of the algebraic equations II. Applied Mathematics and Computation 166(2):324–338
5. Albert A (1977) Regression, pseudo inversion and recurrent valuation. Physmatlit, Moscow
6. Liptser RS, Shiryaev AN (1974) Statistics of stochastic processes. Nauka, Moscow (Science)
7. Davis MHA (1984) Linear valuation and stochastic controls. Physmatlit, Moscow

Chapter 2
Basis of the Method of Polynomial Approximation

2.1 Extension Sets of Observations: The Heuristic Path for Nonlinear Estimation

The estimation algorithm outlined in Chap. 1 can be constructively implemented if some a priori data are known. However, the algorithm does not fully use information from observations, since its operations are linear over the results of observations. Is it possible to increase the accuracy of the estimate, to use more complex, nonlinear operations? A multidimensional version of K. Veyershtrassa's theorem answers this question affirmatively.

We believe that the real functions of many variables $\theta(Y_N)$, which are further defined as approximation representations, are continuous in the closed bounded domain Ω_{Y_N} of the multidimensional space. Fulfillment of this condition [1] allows the use of a multidimensional analog of Weierstrass's theorem (Stone's corollary theorems). The theorem states that for every ε, the following holds:

$$\sup_{Y_N \in \Omega_{Y_N}} |P(Y_N, \varepsilon) - E(\theta|Y_N)| \leq \varepsilon, \tag{1.1}$$

where Ω_N is compact, $E(\theta|Y_N)$ is a continuous function of N components of Y_N, and $P(Y_N, \varepsilon)$ is a polynomial from Y_N (linear combination of powers of the components of Y_N). If this condition is fulfilled, the approximation error ε tends to zero with increasing dimension of the vector Y_N.

Let's define the set $\Omega_{W_{Y_N}}$ polynomial observing $W(Y_N)$, which depends on the primary vector of observations Y_N. The input estimation algorithm is not supposed to be the vectors Y_N and $W(Y_N)$. Inequality (1.1) makes it natural to assign that the degree of the components of Y_N are components of the vectors $W(Y_N)$. The sum of the exponents of all the degrees does not exceed a given integer d. $\Omega_{W_{Y_N}}$ contains a set of Ω_{Y_N} original observations:

$$Y \in \Omega_Y, \Omega_Y \in R^N \in R^N, W \in \Omega_W \in R^{N_1}, N_1 > N. \tag{1.2}$$

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_2,

We will estimate the vector $\theta(W(Y_N))$ via a formula similar to (1.3) of Chap. 1:

$$\hat{\theta}(W(Y_N)) = E(\theta(W(Y_N))) + \Lambda^o(W(Y_N) - E(W(Y_N))), \tag{1.3}$$

where

$$\Lambda^o Q = L.$$

Algorithm (1.3) defines a vector of estimates, the optimal mean-square and linear on the set of degrees of the components of Y_N. This vector is a polynomial of the components of Y_N and the corresponding estimation of the errors' mean cannot have more errors; that delivers a suboptimal polynomial $P(Y_N, \varepsilon)$ of (1.1).

This statement is true because algorithm (1.3) defines the optimal mean-square evaluation on the set of linear combinations of the components of the degrees Y_N and $P(Y_N, 2_N)$ (linear combination of powers of the components of Y_N). However, the linear combinations $P(Y_N, \varepsilon)$ are not optimal in the mean-square and true matrix inequality C (1.3) $\leq$ C (1.1), where C (1.3) and C (1.1) are covariance matrices of the estimation errors, corresponding to the expressions (1.1) and (1.3) for the estimation methods.

The set of elements $W(Y_N)$ expands with increasing d; some of its elements are degrees of component Y_N in the polynomial $P(Y_N, \varepsilon)$. In this case, the estimation error, which corresponds to algorithm (1.3), is at least not greater than the ε in formula (1.1).

Next, Sect. 2.3 represents data on the construction of the vector $W(Y_N)$. The arrangement is such that with an increase in d, estimation errors are reduced and do not exceed ε in (1.1).

A priori data for formula (1.3) are the vector and matrix $E(\theta)$, $E(W(Y_N))$, C_0, Q, L numerically defined in Chap. 1, by replacing the symbol Y_N on the symbol $W(Y_N)$. The elements of the vectors $W(Y_N)$ linearly depend on degrees of the observations; therefore, formula (1.3) corresponds to a nonlinear algorithmic process. The estimation error we obtain when using an extended set of observations will always be less than many original observations Ω_Y.

In Chap. 1, the formula was determined by calculating the estimation error covariance matrix obtained with this nonlinear algorithm and finding the best of their reduction of the nonlinear terms in the function WY_N.

Membership of the vector $W(Y_N)$'s degrees and the plurality of a sufficiently large value of d in principle ensure the achievement of arbitrarily small mean-value estimation errors.

2.2 The Statistical Basis

We assume that parameter vector θ has components $\theta_1, \ldots, \theta_q$ and at fixed vector Y_N belongs to a region $\Omega_{\theta|Y_N} \in R^q$. This region can have a finite or infinite number of points. The latter will be the case, for example, if $\theta = F(Y_N, \xi)$, where ξ is an independent variable that varies in a region.

If the vector Y_N spans the region of Ω_{Y_N} points, then the vector θ spans points of some region Ω_θ.

We suppose that Y_N, θ are random vectors on region $\Omega_{Y_N} \asymp \Omega_{\theta_N|Y} \in R^{N+q}$ and that their joint stochastic measure is

$$p(\theta, Y_N) = p(Y)p(\theta|Y_N),$$

where function $p(\theta|Y_N)$ is the conditional density of probabilities of the random vector θ at the fixed vector Y_N. If the set $\Omega_{\theta|Y_N}$ is composed of points $\theta_1(Y_N), \ldots, \theta_r(Y_N)$, then

$$p(\theta|Y_N) = (\delta(\theta - \theta_1(Y_N)) + \cdots + \delta(\theta - \theta_r(Y_N)))/r,$$

where $\delta(\ldots)$ is a delta function of θ variables. The vector of conditional expectation $E(\theta|Y_N)$ is represented by

$$E(\theta|Y_N) = \int_{\theta \in \Omega_{\theta|Y_N}} \theta p(\theta|Y_N) d\theta. \tag{2.1}$$

Let the vector W be a function of components of the vector $Y_N : W = W(Y_N)$. If the random vector Y_N is fixed, then the algorithm outlined in Chap. 1 delivers an estimator of the vector θ (or a function of this vector) that is linear relative to vector W and optimal in the root-mean-square sense on a class of linear operators.

We will use this algorithm to construct an estimator of the vector $E(\theta|Y_N)$ that will be linear relative to a vector $W(Y_N)$ and optimal in the root-mean-square sense. We can construct an estimator because the joint density of probabilities $p(\theta, Y_N)$ permits us to find the first and second statistical moments of the random vectors $E(\theta|Y_N)$, $W(Y)$ that are necessary for using formulas of Chap. 1. From Eq. (2.1), we will find

$$E(E(\theta|Y_N)) = \int_{Y_N \in \Omega_{Y_N}, \theta \in \Omega_{\theta|Y_N}} \theta p(\theta, Y_N) d\theta dY_N, \tag{2.2}$$

$$E(W(Y_N)) = W(Y_N)p(\theta, Y_N) d\theta dY_n, \tag{2.3}$$

$$L = E((E(\theta|Y_N) - E(E\theta|Y_N)))(W(Y_N) - E(W(Y_N)))^T)$$

$$= \int_{Y_N \in \Omega_{Y_N}, \theta \in \Omega_{\theta|Y_N}} (E(\theta|Y_N) - E(E\theta|Y_N)))(W(Y_N) - E(W(Y_N)))^T p(\theta, Y_N) d\theta dY_N, \tag{2.4}$$

$$Q = E((W - E(W))((W - E(W))^T)$$

$$= \int_{Y \in \Omega_{Y_N}, \theta \in \Omega_{\theta|Y_N}} (W - E(W))(W - E(W))^T p(\theta, Y) d\theta dY. \tag{2.5}$$

We assume that $E(\theta|Y_N)$ is a continuous vector-function Y_N.

Let the components $w_1(Y_N), \ldots, w_m(Y_N)$ of the vector W be the first m components of W. Let's find the optimal in the root-mean-square-sense estimator of the vector $E(\theta|Y_N)$; we'll do it using a linear (relative to W) vector-function of orm (1.3) of Chap. 1. But under this condition, the vector W is a function of Y_N. Hence, additionally, the vector to estimate the conditional expectation—an estimator realized via an optimal in the root-mean-square-sense linear operator over the vector $W(Y_N)$—is denoted as $\hat{E}_{\theta|Y}(Y, m)^o$ and defined by

$$\hat{E}_{\theta|Y_N}(Y_N, m)^o = E(E(\theta|Y_N)) + \Lambda^o(W(Y_N) - E(W(Y_N))), \tag{2.6}$$

where

$$\Lambda^o Q = L. \tag{2.7}$$

2.3 Polynomial Approximation

Now, we believe that elements of the basic sequence are products of integer nonnegative power functions of components of the vector of primary observations Y_N:

$$w_{a_1,\ldots,a_N}(Y_N) = y_1^{a_1} \cdots y_N^{a_N}, \tag{3.1}$$

where the nonnegative integers $a_1, \ldots, a_N$ deliver all integer nonnegative solutions of the inequality $0 \le a_1 + \cdots + a_N \le d, d = 1, 2, \ldots$. For $d \to \infty$, we obtain a countable sequence of basic functions. We will notice that in this case, Stone algebra is a space of polynomial N variables, and Stone's theorem serves as a multidimensional analog of Weierstrass's theorem. For given integers d, N, we will denote the number of elements of the basic sequence as $m(d, N)$.

Lemma 4.1 *The value $m(d, N)$ is defined by the recurrent formula*

$$m(d, N) = m(d-1, N) + (1/d!)(N + d - 1) \cdots N, m(1, N) = N.$$

The formula is proved by induction.

With increasing d, the value $m(d, N)$ quickly increases. For example, if $N = 4$, then

d	1	2	3	4	5	6	7	8
$m(d, N)$	4	14	34	69	125	209	329	494.

The vectorial linear combination of basic functions that at fixed integer d delivers—onto regions $\Omega_Y \times \Omega_{\theta|Y}$—an optimal (in the root-mean-square-sense) estimator of the vector $\hat{E}(\theta|Y)$ of the form

$$\hat{E}(\theta|Y_N)(Y_N,d)^o = E(E(\theta|Y_N)) + \Lambda^o(W(Y_N) - E(W(Y_N))), \tag{3.2}$$

$$\Lambda^o Q = L, \tag{3.2*}$$

or

$$\hat{E}(\theta|Y)(Y,d)^o = \sum_{0\le(a_1+\cdots+a_N)\le d} \lambda(a_1,\ldots,a_N) y_1^{a_1}\cdots y_N^{a_N}, \tag{3.3}$$

where $\ldots, \lambda(a1,\ldots,aN),\ldots$ are the vectorial weight coefficients. We will find these coefficients if, to the right of Eq. (3.2), we substitute components of vectors and matrices from relationships (2.2)–(2.5) as well as components of the vector $W(Y)$ from Eq. (3.1), and set as equal the coefficients before identical products of power functions in Eqs. (3.2) and (3.3).

Vectors $\ldots, \lambda(a_1,\ldots,a_N),\ldots$ should be input to the computer. Then formula (3.2) or (3.3) solves the problem of polynomial approximation without solving matrix equation (3.2*) for any vectors $Y \in \Omega_{Y_N}$.

Equation (3.2) or (3.3) solves the problem of polynomial approximation of a vector of conditional expectation for the random vector of unknown parameters with an error, uniformly small on the given region Ω_Y:

$$\sup_{Y_N\in\Omega_{Y_N}} |E(\theta|Y_N) - \sum_{0\le a_1+\cdots+a_N\le d} \lambda(a_1,\ldots,a_N) y_1^{a_1}\cdots y_N^{a_N}| \to 0, d\to\infty.$$

We emphasize that with increasing number d, the polynomial, approximating vectorial series, contains a vector $E(\theta|Y)$ with arbitrary small root-mean-square error on region Ω_Y, despite having used the simple linear operator over polynomial functions of results of primary observations: Simplicity of the operator represented by its linearity, "is compensated" for by nonlinear (polynomial) functions of results of the primary observations, processed by the linear operator.

Then, by $W(d,k)$ we denote a vector whose components contain all possible products of the form $y_1^{a_1}\cdots y_k^{a_k},\ldots, 0 \le a_1+\cdots+a_k \le d$. We assume that all the components of the vector $W(d,N)$ are linearly independent. Then the numbering of these components can be arbitrary. However, to represent the recurrent form of an algorithm of polynomial approximation, the numbering, defined by recurrent relationships, is reasonable.

The recurrent form of writing the vector $W(d,k)$ is of the form

$$W(d,k)^T = \|W(d,k-1)^T \quad w(d-1,k-1,k-1,y_k)^T\|, \tag{3.5}$$

where

$$w(d-1,k-1,y_k)^T = \|W(d-1,k-1)^T y_k \cdots W(1,k-1)^T y_k^{d-1} \quad W(0,k-1)^T y_k^d\|, \tag{3.6}$$

$$W(0,i) = W(i,0) = 1, i = 0,1,2,\ldots$$

For a given integer d, successive application of relationships (3.5), (3.6) leads to the formulas

$$W(d,1)^T = \|1 y_1 \cdots y_1^{d-1} y_1^d\|,$$

$$W(d,2)^T = \|W(d,1)^T \quad W(d-1,1)^T y_2 \cdots W(1,1)^T y_2^{d-1} \quad y_2^d\|,$$

$$W(d,3)^T = \|W(d)^T \quad W(d-1,2)^T y_3 \cdots W(1,2)^T y_3^{d-1} \, y_3^d\|, \ldots$$

The formulas imply a way of successively numbering the components of vector $W(d,k)$. These components are then denoted as $w_1, w_2, \ldots, w_{m(d,k)}$.

2.4 Calculating Statistical Moments and Choice of Stochastic Measure

The preceding material points to the fact that at the exact calculation of integrals (2.2)–(2.5) and at a great value of an integer d, the presented method of approximation delivers an estimator vector of parameters in the form of a vector of conditional expectation. It is well known that the estimator is optimal in the root-mean-square sense for all vector-functions of a vector of observations. The efficiency of this estimator—as a carrier of information, contained in a vector of observations—depends on the a priori stochastic measure that was chosen.

It is likely that the problem of choosing an optimal stochastic measure can be formulated. However, a similar problem is not further considered here.

Besides the a priori region $\Omega_Y \times \Omega_{\theta|Y_N}$, there are commonly no a priori data for stochastic characteristics of the vectors Y_N, θ. Hence, it is natural to use the heuristic arguments when assigning a stochastic measure. However, the heuristics' role can be reduced if we connect choosing a stochastic measure with the problem of numerically determining integrals (2.2)–(2.5).

Let's consider a method of calculating multidimensional integrals, effectively used below for polynomial approximation in solving applied problems.

Let $\Omega \in R^m$; it is necessary to calculate the integral

$$J = \int_{x \in \Omega} F(x) dx, \tag{4.1}$$

where $F(x^1, \ldots, x^m)$ is a given integrand function, and Ω is a unity cube in R^m: $0 \le x^i \le 1$.

Every cube's edge is divided into r equal segments of length $1/r$, whose ends are vertices r^m of the smaller elementary cubes denoted as $E_1, \ldots, E_k, \ldots, E_{r^m}$. Then,

$$J = \sum_{k=1}^{k=r^m} J_k, \tag{4.2}$$

where

$$J_k = \int_{x \in E_k} F(x)dx. \tag{4.3}$$

To calculate J_k, one generalizes the method of trapezoids to a multidimensional case. Multidimensional linear interpolation is performed, at which the integrand function $F(x)$ is replaced with a multilinear function $F(x)'$. This function is a sum of products of functions, linear in variables $x^1, \ldots, x^m$, and coincides with $F(x)$ in 2^m cube vertices E_k.

We assume that $x_{k_1}, \ldots, x_{k_m}, 1 \le k_1, \ldots, k_m \le r-1$ are coordinates of that cube vertex E_k, whose coordinates have the smallest values of all 2^m cube vertices E_k. Then, coordinates of all 2^m cube vertices can be represented by the expressions

$$x_{k_1}(\alpha_1) = x_{k_1} + \alpha_1/r, \ldots, x_{k_m}(\alpha_m) = x_{k_m} + \alpha_m/r,$$

where quantities $\alpha_1, \ldots, \alpha_m$ assume—independently of one another—the value 0 or 1. Next, $x_{k_i}(1) - x_{k_i}(0) = 1/r, i = 1, \ldots, m$, are the coordinates $x_1, \ldots, x_m$ of a point, belonging to E_k, that satisfy the inequalities $x_{k_i}(0) \le x^i \le x_{k_i}(1)$, $i = 1, \ldots, m$.

Let's define linear functions of these point coordinates by

$$f_0(x^i) = r(x_{k_i}(1) - x^i),\ f_1(x^i) = r(x^i - xk_i(0)).$$

It is clear that $0 \le f_0(x^i) \le f_1(x^i)$, $f_0(x^i) + f_1(x^i) = 1$.

The interpolating function $F(x)'$ is defined by

$$F(x^1, \ldots, x^m)'$$

$$= \sum_{\alpha_1, \ldots, \alpha_m = 0,1} f_{\alpha_1}(x^1) \cdots f_{\alpha_n}(x^n) F(x_{k_1} + \alpha_1/r, \ldots, x_{k_m} + \alpha_m/r). \tag{4.4}$$

Summation in Eq. (4.4) is done over all binary numbers of the form $\alpha_1, \ldots, \alpha_m$, and the term's number is equal to 2^m.

Linear functions $f_0(x^i)$, $f_1(x^i)$ satisfy the identity

$$\sum_{\alpha_1, \ldots, \alpha_m = 0,1} f_{\alpha_1}(x^1) \cdots f_{\alpha_m}(x^m) = 1. \tag{4.5}$$

The identity is proved by induction.

Replacing in Eq. (4.3) the function $F(x)$ with $F(x)'$, after integrating over cube E_k, we will find an approximate expression for the integral J_k:

$$J_k \simeq (1/2r)^m \sum_{\alpha_1, \ldots, \alpha_m = 0,1} F(x_{k_1} + \alpha_1/r, \ldots, x_{k_m} + \alpha_m/r). \tag{4.6}$$

Hence, the approximate value of the integral over every elementary cube is proportional to an arithmetic average of values of the integrand function in vertices of this cube.

Let's consider a situation when every integral J_k is of the form

$$J_k = \int_{x \in E_k} F(x)dx, \tag{4.7}$$

where $F(x)$ is a yet-to-be-determined probability density of the random vector x. The value of J_k from (4.7) will become equal to the right of Eq. (4.6) if the probability density is assumed to be the proportional sum of products of delta-functions:

$$p(x^1, \ldots, x^m) = (1/2r)^m$$

$$\times \sum_{\alpha_1, \ldots, \alpha_m = 0,1} \delta(x_{k_1} + \alpha_1/r - x^1) \cdots \delta(x_{k_m} + \alpha_m/r - x^m). \tag{4.8}$$

For this function, there is a valid normalization condition on the unit cube.

Thus, assigning the probability density to be equal to a sum of products of delta functions delivers an exact value of the corresponding integral. But we get the same integral value if distribution of the random vector x on the unit cube is assumed uniform, and generalization of the method of trapezoids (presented above) is taken as an approximate method of calculating multidimensional integrals.

Hence, there are two possibilities.

1. Assign the probability density as a sum of products of delta functions on Ω and find an exact value of integral (4.7).
2. Assign the probability density as uniform on Ω and find an approximate value of this integral after using a generalized method of trapezoids.

In the latter case, approximate values should also be used; then some of the integrals being calculated can be determined analytically. This means that integrals being calculated are elements of a priori vectors and matrices needed to determine the vector of linear estimators in Chap. 1, optimal in the root-mean-square sense.

It should be emphasized that for a determinate connection of random vectors θ and Y, at the exact calculation of the integrals (2.2)–(2.5) and a sufficiently great value of d, the estimator $\hat{\theta}(Y, d)$ is close to the estimated vector θ and practically does not depend on the chosen probability density $p(Y)$. This statement follows from Eq. (4.8).

Let's assume that the a priori region Ω_Y is a cube in R_N, which is divided into r^N elementary cubes in realizing the generalized method of trapezoids (see above). Let's choose probability density $p(Y)$ as the corresponding sum of products of delta functions. Then Eq. (4.8) (representing, for the method of polynomial approximation, the key convergence of the calculations) will be true for any (including small) number r.

Of course, for small r, it will be necessary to use a greater number d. Hence, the "rough" (for small r) calculating weight coefficients $\lambda(a_1, \ldots, a_N)$ in Eq. (3.3) should be compensated for by a greater number $m(d, N)$ of terms in Eq. (3.3). The choice of rational—according to the criterion of minimum time—calculating numbers r and d is a subject of special consideration.

2.5 Fragment of Program of Modified Method of Trapezoids

An approximate value of a multidimensional integral over a unit cube is equal to a sum of approximate values, to be calculated under Eq. (4.2). However, practical use of this method is not rational, as it requires a huge volume of computing.

In fact, if a vertex of a small parallelepiped E_k is within a unit cube (it is surrounded by small parallelepipeds all around), then, for multiple uses of formulas like Eq. (4.6), one should calculate the value of the function $F(\ldots)$ in this vertex $2n$ times. However, if a vertex coincides with that of the unit cube, then the function value of $F(\ldots)$ is calculated only once.

It is reasonable to design an algorithm that would require calculating the function $F(\ldots)$ only once in every node of the grid covering the unit cube. In such a case, the explicit representation of an approximate integral as a linear combination of the function $F(\ldots)$'s values in nodes is rather difficult, because coefficients of this linear combination depend on the integer r.

We will present the offered algorithm as a fragment of a Pascal program for the case of $r = 5$:

```
J:=0;
for x1:=0 to r do for x2:=0 to r do
for x3:=0 to r do for x4:=0 to r do
for x5:=0 to r do
begin
x[1]:=x1;x[2]:=x22;x[3]:=x3; x[4]:=x4;x[5]:=x5;
nj:=0;
for i:=1 to 5 do if (x[i]=0) or (x[i]=r) then nj:=nj+1;
if nj=0 then mj:=32;
if nj=1 then mj:=16;
if nj=2 then mj:=8;
if nj=3 then mj:=4;
if nj=4 then mj:=2;
if nj=5 then mj:=1;
K:=mj/32;
J:=J+KF(x[1],x[2],x[3],x[4],x[5]);
end;
```

Table 2.1 Values of integers for different values of r

r	$k(1)$	$k(1/2)$	$k(1/4)$	$k(1/8)$	$k(1/16)$	$k(1/32)$
5	1,024	2,560	2,560	1,280	320	32
10	59,049	65,610	29,160	6,480	720	32
15	537,824	384,160	109,760	15,680	1,120	32
20	2,476,099	1,303,210	274,360	28,880	1,520	32

The fragment implies that the algorithm gives an approximate integral J as a linear combination of the function $F(\ldots)$'s values in vertices of small cubes. The coefficients K of this linear combination take the values 1, 1/2, 1/4, 1/8, 1/16, 1/32, multiplied by integers, automatically determined by the algorithm for the modified method of trapezoids.

Table 2.1 gives values of these integers for different values of r.

Reference

1. Timan AF (1960) Approximation theory of real-approximate functions of real variable (Science). Macmillan, New York

Chapter 3
Polynomial Approximation and Optimization of Control

3.1 Introduction

Polynomial approximation of a function with many variables is reasonable because it replaces this function with a transparent differentiated function whose properties are well investigated. In such a case, there should, of course, be a constructive algorithm to define the coefficient of polynomials, and at an increasing power of polynomials, the convergence of a sequence of polynomials to an approximated function that is uniform on its field of definition should be guaranteed. Apparently, the most productive polynomial approximation is in solving optimization problems. For example, let the vector-function parameters be defined by an algorithm of numerical integration of a system of differential equations; it is necessary to define its extremum point. This problem is essentially facilitated if the function of the vector parameters is replaced by a polynomial of the vector parameters with an error uniform on the field of definition and marginally small. A polynomial of the vector of parameters is a smooth function of its arguments, and points of its extremum in a real region are defined by well-known methods of computational mathematics. We notice that these points can be found by a method (explained ahead here) of solving systems of nonlinear algebraic equations if they are stationary points of the approximating polynomial.

As an approximated function is continuous, and approximation errors are uniformly small, then, with the exception of some cases of degeneracy (not interesting for applied problems), the extrema of approximating polynomials should be close to the extrema of the function being approximated.

In the modern theory of approximating functions of real variables, there is a theorem of Weierstrass that proves the existence of a necessary polynomial sequence for a function of one variable. Bernstein's polynomials provide a way for the factual construction of a sequence of polynomials.

For functions with many variables, the validity of a multidimensional analog of Weierstrass's theorem follows from Stone's theorem [1] for a continuous function, set on a compact. Chapter 2 presented a general method for the constructive design of a sequence of approximating polynomials.

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_3,

In Sect. 3.2, we present a method of multidimensional polynomial approximation for a continuous function, satisfying Stone's theorem. The method has an algorithm to construct a sequence of approximating polynomials whose approximation errors uniformly converge to zero when its power is increased. The method does not demand differentiability of the function and can be used in the problems of optimizing dynamic systems' control. We describe an application of the method for approximately solving a similar problem.

We notice that in computational mathematics, one widely uses the representation of a function of a vector of real variables by a segment of the series whose members are proportional to products of integer power functions of this vector's component. Usually, one uses the Taylor multidimensional series as such a series. Basic difficulties with using Taylor series are known: The function should be differentiable the corresponding number of times; the area of convergence of the series is established by special, rather complex, research. In the stated method we present, these difficulties are absent: There is no need for the function's differentiability (the method's algorithm uses only integration operations); the method's area of uniform convergence coincides with the compact on which the function is defined.

3.2 Problem of Polynomial Approximation of a Given Function

Let's say we have a function

$$\theta = F(Y_N), \tag{2.1}$$

where $Y_N \in \Omega_Y \in R^N, \theta \in R, \Omega_{Y_N}$ is a given compact in R^N, $F(Y_N)$ is continuous on Ω_{Y_N}, and $y_1, ..., y_N$ are components of vector Y_N.

It is necessary to find an approximate polynomial representation for $F(Y_N)$:

$$F(y_1, \ldots, y_N) \simeq \sum_{0 \le (a_1 + \ldots + a_N) \le d} \lambda(a_1, \ldots, a_N) y_1^{a_1} \cdots y_N^{a_N}, \tag{2.2}$$

where d is a given integer value and $\lambda(a_1, \ldots, a_N), \ldots$ depend on $F(Y_N), \Omega_{Y_N}, d, N$.

The number $m(d, N)$ of members in the series segment on the right of Eq. (2.2), which is equal to the number of integer nonnegative solutions of the inequality $0 < a_1 + \ldots + a_N \le d$, is defined by the recurrent formula of Lemma 4.1 in Chap. 2.

To use the results of Chap. 2, we assumed that vector Y is random on Ω_Y with density of probabilities $p(Y)$. From (2.1), one sees that in Eqs. (2.2)–(2.5) of Chap. 2, it is necessary to assume

$$\Omega_{\theta|Y_N} = F(Y_N),$$

$$p(\theta, Y_N) d\theta dY = p(Y_N) \delta(\theta - F(Y_N)) d\theta dY.$$

Hence,

$$E(\theta|Y_N) = \theta. \tag{2.3}$$

The identity (2.3) points out that in the considered problem, Eq. (3.3) of Chap. 2, presenting a polynomial estimator of random vector $E(\theta|Y)$ passes to a formula for a polynomial estimator of parameter θ if vector Y and integer value d are given. Let's denote this estimator here as $\hat{\theta}(Y_N, d)$.

The algorithm of function $F(Y_N)$ decomposition to a power series should define weight coefficients $\ldots, \lambda(a_1, \ldots, a_N), \ldots$, so that error $|F(Y) - \hat{\theta}(Y_N, d)|$ of the representation of function $F(Y_N)$ by the right part of Eq. (2.2) converges—uniformly on Ω_{Y_N}—to 0 at increasing d:

$$\sup |F(Y_N) - \hat{\theta}(Y_N, d)| \to 0, d \to \infty. \tag{2.4}$$

Equation (2.4) is a special case of Eq. (3.4) in Chap. 2.

Equations (2.2)–(2.5) of Chap. 2, defining the first and second statistical moments of random vectors $E(\theta|Y_N)$, $W(Y_N)$, will acquire the following form:

$$E(F(Y_N)) = \int_{\Omega_{Y_N}} F(Y_N) p(Y_N) dY, \tag{2.5}$$

$$E(W(Y_N)) = \| \int_{\Omega_{Y_N}} W_i(Y) p(Y_N) dY^T \|, i = 1, \ldots, m(d, N), \tag{2.6}$$

$$L = E((F(Y_N) - E(F(Y_N))W(Y_N)^T =$$

$$\| \int_{\Omega_{Y_N}} (F(Y)) W_i(Y_N) p(Y_N) dY^T \|, i = 1, \ldots, m(d, N), \tag{2.7}$$

$$Q = E((W(Y_N) - E(W)))W(Y_N)^T =$$

$$\| \int_{\Omega_Y} (w_i(Y_N) - E(W_i(Y_N)))W_j(Y_N) p(Y_N) dY^T \|, i, j = 1 \ldots, m(d, N). \tag{2.8}$$

It is expected that choosing a stochastic measure $p(Y_N)$ delivers with random linear independence a component of vector $W(Y_N)$ and, hence, that matrix Q^{-1} exists.

Theorem 2.1 *A polynomial estimator of random variable θ, which is optimal in the root-mean-square sense, is represented by*

$$\hat{\theta}(Y_N, d)^o = E(F(Y_N)) + LQ^{-1}(W(Y_N) - E(W(Y_N))). \tag{2.9}$$

The proof follows from Eq. (2.6) of Chap. 2.

Let's denote by

$$D(d)^o = E((\hat{\theta} - \theta)^2)$$

a variance of errors of the optimal estimator. The consequences of Eq. (2.9) are as follows:

Consequence 1. $D(d)^o = E(F(Y_N) - E(F(Y_N)))^2 - LQ^{-1}L^T$.

Consequence 2. If we have two polynomials $\hat{\theta}(Y_N, d_1)^o$ and $\hat{\theta}(Y, d_2)^o$, where $d_1 < d_2$, then $D(d_2)^o < D(d_1)^o$.

Consequence 3. If $F(Y_N)$ is a polynomial of power d_0, then $D(d_0)^o$ 0 for $d = d_0$, and further increasing d will not change the situation.

Consequence 4. Upon the exact calculation of integrals (2.5)–(2.8) and at a sufficiently great value of d, the value of estimator $\hat{\theta}(Y, d)$ is close to the estimated value θ and practically does not depend on the selected density of probabilities $p(Y)$. This statement follows from identity (2.3) and relationship (2.4).

The square of the mean square of the estimation random error—divided by the function's value—serves as a characteristic of the mean errors of polynomial approximation:

$$\sigma(d) = (E((\hat{\theta}(Y_N, d) - F(Y_N))/F(Y))/^2))^{1/2}.$$

The calculated polynomial approximation has a uniformly small error on Ω_{Y_N}. Hence, the value $\sigma(d)$ serves as an exhaustive characteristic of approximation errors.

An approximated function $F(Y_N)$ can be defined by analysis, by algorithms, or by tables. Function definition should only provide the calculation of integrals (2.5)–(2.8).

The diagram of the principal algorithm of calculating the coefficients of polynomials applies for any approximated function. The time of calculations increases with a rise in value d and with the complexity of the approximated functions.

3.3 Applied Examples

3.3.1 Detection of a Polynomial Function

Let's say we need a computational process that answers the questions of whether the given function is polynomial and, if yes, what are its power and polynomial coefficients?

A software implementation of the presented principal algorithm easily solves the posed problem. We consequently increase the value $d = 1, 2, \ldots$. The algorithm's scheme guarantees proximity to zero of the value $\sigma(d)$ for approximation errors, as soon as d becomes equal to the maximum power of the unknown polynomial. Let's consider an example.

Let $\Omega_{Y_N} : -1 \le y_i \le 1, i = 1, \ldots, 4$, and

$$F(Y) = y_1^5 + y_1 y_2 y_3 y_4 + y_3^3 + y_2 y_4.$$

The calculations were done for $q = 1, \ldots, 5$:

d	1	2	3	4	5
$m(d, n)$	4	14	34	69	125
$a(d)$	69.8	41.8	5.5	4.9	2.8×10^{-13}.

The results of the calculations show that for $d = 5$, the value $\sigma(d)$ abruptly jumps to zero. The algorithm confidently signals that the investigated function is a polynomial of power 5. Sensitivity of the algorithm to small, "not polynomial" terms, to be present in the function's composition, illustrates the calculations' results if a term $0.05\ sin(y1 + y2 + y3 + y4)$ is added to the polynomial being considered:

d	1	2	3	4	5
$m(d, n)$	4	14	34	69	125
$\sigma(d)$	33	46.5	23.8	3.8	0.017.

3.3.2 Approximation Errors for a State Vector of Dynamic Systems

In applications one often uses a representation of functions of many variables by means of a segment of its decomposition into a power series relative to components of a vector of variables that belong to an a priori region Y. Most often, this segment is a linear function of components; sometimes the series segment is a quadratic function, and so on. Determining the limits in which the function is represented by a polynomial of the given power is often a difficult problem if the function is defined by an algorithm of sequential calculations. Let's have, for example, a numerical integration program of a system of differential equations.

For instance, let the analyzed functions be components of a vector $\theta(t)$, satisfying the equation

$$\dot{\theta} = \Phi(\theta),$$

and let vector $y_1, \ldots, y_N$ be a vector of initial conditions: $Y = \theta(0)$. It is necessary to find time intervals t of numerical integration of this equation in which the linear dependence of vector $\theta(t)$ components from components of Y will be approximately true [with a small value $\sigma(d)$], then approximate quadratic dependence, and so on.

The principal algorithm of polynomial approximation—used for the sequence of values t and d—solves this problem.

For a differential equation with the right part $Phi(\theta)$ to be continuous and having continuous partial derivative, vector $\theta(t)$, one needs a continuous function of the vector of the initial conditions

$$\theta(T) = F(Y_N), Y = \theta(0),$$

and the polynomial approximation algorithm, presented above, is also applied.

We emphasize that after the algorithm has made a sufficiently exact approximation and defined the corresponding values $\ldots, \lambda(a_1, \ldots, a_N), \ldots$, a Cauchy problem for any initial conditions from Ω_Y can be solved without a numerical integration program.

Let's consider an example. Say we have a fourth-order nonlinear dynamic system

$$\dot{\theta}_1 = \theta_2,$$

$$\dot{\theta_2} = \varphi(-2\xi\theta_2(1/(\tau_1)) - \theta_1(1/\tau_1^2) + 1),$$

$$\dot{\theta_3} = \theta_4,$$

$$\dot{\theta_4} = \varphi(-2\xi\theta_4(1/(\tau_2)) - (\theta_3 - \theta_1)(1/\tau_2^2) + 1),$$

where $\varphi(a) = a$ if $-gm \le a \le gm$ *and* $\varphi(a) = gm$ if $|a| > gm$.

The represented model of a dynamic system can be interpreted as a serial connection of two dynamic subsystems, each of which is—in a linear region of changing phase coordinates—an oscillatory link with time constants τ_1, τ_2 and with damping decrements ξ_1, ξ_2. The values θ_1 and θ_3 serve as outputs of these subsystems. The second subsystem implements tracking value θ_1 for the value θ_1. Modules of acceleration by engines of every servo-system are limited by the value gm.

For a number of values t, d, we will estimate the accuracy of the polynomial approximation of the functional dependence of the values $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$, $\theta_4(t)$ from the initial conditions y_1, y_2, y_3, y_4.

Let's assume $\Omega_Y : -1 \le y_i \le 1, i = 1, ..., 4, \tau_1 = 1, \xi_1 = 0.5, \tau_2 = 2, \xi_2 = 0.25$.

If the value gm is great, then the dynamic system is linear, and a good accuracy of approximation is reached for $d = 1$. For example, let $g = 10$. Then, for $d = 1, m(d, N) = 4$, the values $\sigma(d, \theta_i), i = 1, \ldots, 4$, have an order 10^{-17}. Then we assume that $gm = 1.95$.

The accuracy data of the approximation, which appear below, were obtained by a Monte Carlo method for 100,000 realizations and at a uniform dispersion of the initial conditions within the region ω. The presented values should be considered as root-mean-square deviations (RMSDs) of linear approximation errors.

Linear approximation: $d = 1, m(d, N) = 4$.

t(s)	0.5	1	1.5	2	2.5	3
$\sigma(d, \theta_1)$	0.01100	0.0110	0.0130	0.0500	0.0250	0.0260
$\sigma(d, \theta_2)$	0.02400	0.0058	0.3800	0.0240	0.0019	0.0550
$\sigma(d, \theta_3)$	0.00063	0.0037	0.0018	0.0027	0.0046	0.0039
$\sigma(d, \theta_4)$	0.00130	0.0045	0.0065	0.0250	0.0076	0.0400

The represented calculation results show that the RMSD of linear approximation errors do not exceed 5 % from the range of changing of initial conditions.

The exception arises for $t = 1.5$ s, when the RMSDs of linear approximation errors of speed θ_2 reach 37 %, apparently because of unique features of transients in the first dynamic subsystem.

For a sharp reduction in the RMSDs of approximation errors, it is necessary to increase the power d of approximating polynomials:

$t = 1.5$ s.

d	1	2	3	4
$m(d, N)$	4	14	34	69
$\sigma(d, \theta_1)$	0.0130	0.00350	0.00260	0.00140
$\sigma(d, \theta_2)$	0.3800	0.10000	0.06600	0.03600
$\sigma(d, \theta_3)$	0.0018	0.00050	0.00031	0.00019
$\sigma(d, \theta_4)$	0.0065	0.00170	0.00220	0.00130

One can see that the quadratic approximation ($d = 2$) has already sharply reduced RMSD values.

3.4 Polynomial Approximation in Control Optimization Problems

A mathematical model of a dynamic system is represented by the equation

$$\dot{\theta} = \varphi(\theta, u), \tag{4.1}$$

where $\theta(t) \in R^q$ is a current system state vector, $u(t) \in R_1$ is a current scalar of control $|u(t)| \leq 1$, $\varphi(\ldots)$ is a given vector function, differentiable by its arguments, and $\theta(0)$ is a given vector of initial conditions.

Let's have one of the sets of possible problems of optimizing a dynamic system: to find a function $u^o(t)$, $|u(t)^o| \leq 1$, where

$$u^o(t) = \arg \max_{|u(t)\leq 1} \Phi(\theta(T)). \tag{4.2}$$

A continuous function $\Phi(\ldots)$ is an optimization criterion for this problem. The value T is given.

Pontryagin's maximum principle [2] delivers necessary conditions that $u^o(t)$ should satisfy. However, in a general case, the realization of conditions is complicated, as it requires solutions of a two-point boundary value problem for Eq. (4.1) and for the equations that a vector of conjugate variables satisfies. Polynomial approximation implements an approximate replacement of the initial optimization problem to a sequence of standard nonlinear programming problems.

Now, we will denote by $\Psi(u(t))$ the value $\Phi(\theta(T))$, obtained at control $u(t)$, $0 \leq t \leq T$.

Let $B(t, y_1, \ldots, y_n)$ be a Bernstein polynomial of power $n-1$:

$$B(t, y_1, \ldots, y_n) = \sum_{k=0}^{n-1} y_{k+1} C_{n-1}^{k} (1/T)^k (1-(1/T))^{n-1}, \tag{4.3}$$

where $y_1, \ldots, y_n$ are constants, and C_{n-1}^{k} is a number of combinations of $n-1$ by k. From (4.3) one sees that $|B(t, y_1, \ldots, y_n)| \le 1$ if $|y_{k+1}| \le 1, k = 0, \ldots, n-1$.

Let's suppose that $|u^o(t)$ is a continuous function. It is known that if

$$y_{k+1} = u^o(((k/n)-1)T), \dot{k} = 0, \ldots, n-1,$$

then constructing the polynomials $B(t, y_1, \ldots, y_n)$ gives proof of Weierstrass's theorem [it asserts that for any continuous functions $u(t)$, defined on the segment $[0, T]$, there is a sequence of polynomials from t, uniformly converging to this function]:

$$\sup_{0 \le t \le T} |u^o(t) - B(t, y_1, \ldots, y_n)| \to 0, n \to \infty. \tag{4.4}$$

Hence, for any ε, there exists an integer $y_{n(\varepsilon)}$:

$$\sup_{0 \le t \le T} |u^o(t) - B(t, y_1, \ldots, y_{n(\to)})| \le \varepsilon. \tag{4.5}$$

From (4.5) and from properties of the functions that are solutions of Eq. (4.1), it follows that for any decreasing sequence of positive numbers,

$$\delta_1 > \delta_2 > \ldots \delta_k > \ldots,$$

a decreasing sequence of positive numbers will be found:

$$\varepsilon_1 > \varepsilon_2 \ldots \varepsilon_k \ldots$$

and the corresponding sequence of integers is

$$n_{\varepsilon_1}, n_{\varepsilon_2}, \ldots, n_{\varepsilon_k, \ldots}, \ldots : \tag{4.6}$$

$$|\Psi(u^o(t)) - \Psi(B(t, y_1, \ldots, y_{n(\varepsilon_k)}))| \le \delta_k. \tag{4.7}$$

In approximating the polynomials $B(t, y_1, \ldots, y_{n(\varepsilon)})$ denoted in (4.7), we find that the values $y_1, \ldots, y_{n(\varepsilon)}$ are equal to the values of the unknown function $u^o(t)$ in some points of the segment $0, T$. This circumstance interferes with the direct construction of a sequence of approximating polynomials.

For a sequence of powers of Bernstein polynomials $1, 2, ::, s, \ldots$, we will solve a sequence of standard nonlinear programming programs: At restrictions

$$|y_i^0| \le 1, i = 1, \ldots, s,$$

we will find a sequence of finite sets of numbers $y_1^o, \ldots, y_s^o$:

$$\Psi(B(t, y_1^0, \ldots, y_s^o)) = \max_{|y_1^o| \le 1, \ldots, |y_s^o| \le 1} \Psi(B(t, y_1, \ldots, y_s)).$$

Theorem 4.2 *In the sequence of finite sets $y_1^o, \ldots, y_s^o$, there is a subsequence corresponding to integers $s_1, s_2, \ldots, s_i, \ldots$. This sequence solves a problem of approximating optimum control $u^o(t)$:*

$$\sup_{0 \le t \le T} |u^o(t) - B(t, y_1^o, \ldots, y_{s_k}^o)| \to 0, k \to \infty. \tag{4.8}$$

Proof The sequence of values $\Psi(B(t, y_1^o, \ldots, y_s^o)), i = 1, 2, \ldots, s, \ldots$, is limited from above by the value $\Psi(u^o(t))$.

Let the ordinal numbers of finite sets of a desired sequence be coincident with integers (4.6):

$$s_1 = n(\varepsilon_1), \ldots, s_k = n(\varepsilon_k), \ldots.$$

Then the inequalities are true:

$$\Psi(U^o(t)) \ge \Psi(B(t, y_1^o, \ldots, y_{n(\varepsilon_k)})^o) \ge \Psi(B(t, y_1, \ldots, y_{n(\varepsilon_k)})). \tag{4.9}$$

From Eqs. (4.7) and (4.9), we will obtain

$$\sup_{0 \le t \le T} |(\Psi(u^o(t)) - \Psi(B(t, y_1^o, \ldots, y_{s_k}^o))| \to 0, k \to \infty. \tag{4.10}$$

□

Bernstein's polynomials, belonging to the subsequence defined above, converge to optimum control $u^o(t)$, which is one of the solutions of the optimization problem (4.2). The theorem has been proved.

What we just presented is based on Weierstrass's theorem, which is formulated for continuous functions of one variable. Let the approximated function $u^o(t)$ have segments of relay control. However, such control can be approximated by a continuous-time function with an error, uniformly small on the interval $[0, T]$. Hence, in the preceding reasoning, the function $u^o(t)$ ought to be considered a continuous-time function that uniformly and marginally differs from the actual optimum control.

Solving the sequence of nonlinear programming problems is complicated by the fact that the functions $B(t, y_1, \ldots, y_{n(\varepsilon)})$ are defined implicitly—only by numerical solutions of Eq. (4.1). However, $\Psi(B(t, y_1, \ldots, y_{n(\varepsilon_k)}))$ is continuous and defined on the compact

$$|y_k| \le 1, k = 1, \ldots, n.$$

This function can be approximately replaced by an obvious polynomial function.

Such a replacement can facilitate solving a sequence of nonlinear programming problems.

3.5 Optimization of Control by a Linear System: Linear and Quadratic Optimality Criteria

Let a mathematical model (4.1) of a dynamic system be of the form

$$\dot{\theta} = A\theta + bu, \tag{5.1}$$

where A, b are a matrix of dimensionality $r \times r$ and a vector of dimensionality $r \times 1$, depending, generally speaking, on t. Furthermore, we assume that

$$-1 \le u(t) \le 1, \theta(0) = 0.$$

For Eq. (5.1), we will numerically solve T times a Cauchy problem under conditions

$$0 \le t \le T, u_k(t) = C_{n-1}^k (t/T)_1^k - (t/T))^{n-1-k}, k = 0, \ldots, n-1.$$

In this problem, n of its solutions deliver y_n vectors $\lambda_i(T, n), i = 1, \ldots, n$, of dimensionality $r \times 1$. If control is equal to $B(t, y_1, \ldots, y_n)$, then for the vector $\theta(T, n)$, delivered by this control, the representation holds:

$$\theta(T, n) = \sum_{i=1}^{n} y_i \lambda_i(T, n). \tag{5.2}$$

Let the optimality criterion in Eq. (4.2) be linear:

$$\Phi(\theta(T)) = \alpha\theta(T, n), \tag{5.3}$$

where α is a unity vector in R^r. Then the sequence of standard nonlinear programming problems, described in Sect. 3.4 and delivering a converging sequence of approximations of optimum control $u^o(t)$:

$$u^o = \arg \max_{u(t) \le 1} \alpha^T y(T, n), \tag{5.4}$$

becomes a sequence of linear programming problems:

$$\max_{|y_i| \le 1, \ldots, |y_n| \le 1} \sum_{i=1}^{n} y_i \alpha^T \lambda_i(T, n),$$

whose solutions are evident:

$$\theta_i^o = sign(\alpha^T \lambda_i(T, n)), i = 1, \ldots, n. \tag{5.5}$$

The sequence of controls $B(t, y_1^o, \ldots, y_n^o)$ solves an optimum control approximation problem at the optimality criterion of the form (5.3).

We notice that dynamic system (5.1)'s set of accessibility [3] is convex in R^r, and problem (5.4) defines this set's supporting function for a given vector a. If vectors a sufficiently densely fill a unity sphere in R^r, then the vectors

$$\theta^o(T, n) = \sum_{i=1}^{n} sign(\alpha^T \lambda_i(T, n))\lambda_i(T, n) \tag{5.6}$$

implement a point-by-point approximation of a region of accessibility of dynamic system (5.1) for instant T.

Let's consider a numerical example:

$$\dot{\theta_1} = \theta_2, \dot{\theta_2} = -2(\vartheta/\tau)\theta_2 - (1/\tau^2)\theta_1 + u,$$

where $\varepsilon = 0.5, \tau = 1, \theta(0) = 0;$. Let's assume $n = 10$:

$$\alpha_1 = sin\varphi, \alpha_2 = cos\varphi, u_n^o, = B(t, x_1^o, \ldots, x_n^o).$$

The convergence of the sequence of solutions of linear programming problems (values $\alpha^T y^o, n = n0, n0 + 1$, are solutions) is characterized by a sequence of relative differences:

$$\Delta_n = (\alpha^T y^o(T, n) - \alpha^T y^o(T, n - 1))/\alpha^T y^o(T, n).$$

If, despite some oscillations, the modules of sequence members decrease, then the sequence $\alpha^T \theta^o(T, n), n = n0, n0 + 1, \ldots,$ converges to value $\alpha\theta^o(T)$. This statement is true if one accounts for relationships (4.7), (4.8), proved earlier.

Let $\varphi = 0.314$. The sequence of values Δ_n (from $n = 6$ to $n = 95$) has the following appearance:

6	7	8	9	10	11	12	13	14
0.0474	0.0785	0.0596	0.0357	0.0295	0.0118	0.0107	0.0256	0.0314
15	16	17	18	19	20	21	22	23
0.0173	0.0164	0.0070	0.0065	0.0068	0.0180	0.0180	0.0121	0.0099
24	25	26	27	28	29	30	31	32
0.0040	0.0016	0.0105	0.0090	0.0064	0.0053	0.0027	0.0022	0.0042
33	34	35	36	37	38	39	40	41
0.0072	0.0044	0.0045	0.0018	0.0023	0.0013	0.0057	0.0037	0.0035
87	88	89	90	91	92	93	94	95
0.0005	0.0001	0.0008	0.0009	0.0008	0.0006	0.0005	0.0003	0.0004.

The data presented imply that from $n = 35, \ldots, 40$, the negative increments in relative difference values in the modulus do not exceed 0.005. Hence, the further increase in powers of the Bernstein polynomials, approximating optimum control, will not increase the value of the optimality criterion in practice.

In the example being considered, numerically solving the Cauchy problem 35–40 times will produce an accessibility region for any number of vectors a, relative to which a supporting function of this region is being built.

Let the criterion of optimality (4.2) be quadratic:

$$\Phi(\theta(T)) = \theta(T)^T H\theta(T), \tag{5.7}$$

where H is a given positively defined matrix. Then the sequence of standard nonlinear programming problems, described in Sect. 3.4 and delivering a converging sequence of approximations of optimum control $u^o(t)$:

$$u^o = arg \max_{|u(t)|\le 1} \theta(T)^T H\theta(T), \tag{5.8}$$

becomes a sequence of quadratic programming problems:

$$\max_{|y_1|\le 1,\ldots,|y_n|\le 1} (\sum_{i=1}^{n} y_i \lambda y_i(T,n))^T H(\sum_{i=1}^{n} y_i \lambda_i(T,n)). \tag{5.9}$$

In an upcoming example, a sequence of these problems was solved using the function *quadprog* from the software package MATLAB® 6.x.

Sequence of controls $B(t, y_1^o, \ldots, y_n^o)$ solves the optimum control approximation problem for the optimality criterion of the form (5.7).

In the dynamic system considered above, optimum control solves a problem

$$u^o = arg \max_{|u(t)|\le 1} (\theta_1(T)^2 + \theta_2(T)2).$$

Hence, optimum control maximizes a distance from a point on the border of the accessibility region before the origin of the coordinates at the instant of $T = 10c$.

The convergence of the sequence of solutions of quadratic programming problems (values $|\theta^o(T,n)|^2 = n_0, n_0 + 1, \ldots$ are solutions) is characterized by a sequence of relative differences

$$\Delta_n = (|\theta^o(T,n)|^2 - |\theta^o(T,n-1)|^2)/|\theta^o(T,n)|^2.$$

This sequence has the following appearance:

6	7	8	9	10	11	12	13	14
0.0424	0.0241	0.0471	0.0209	0.0263	0.0331	0.0098	0.0217	0.0233
15	16	17	18	19	20	21	22	23
0.0060	0.0201	0.0137	0.0058	0.0174	0.0071	0.0021	0.0187	0.0048
24	25	26	27	28	29	30	31	32
0.0071	0.0099	0.0032	0.0080	0.0062	0.0022	0.0086	0.0037	0.0032
33	34	35	36	37	38	39	40	41
0.0069	0.0027	0.0034	0.0054	0.0020	0.0042	0.0036	0.001	0.0051
42	43	44	45	46	48	49	50	
0.0023	0.0009	0.0051	0.0018	0.0020	0.0034	0.0038	0.0024.	

It has been seen that the sequence of Bernstein polynomials $B(t, y_1^o, \ldots, y_n^o)$, approximating the function $u^o(t)$, approximates relay control when $u^o(t) = 1$ or $u^o(t) = -1$, with two instants of switching the control's sign. For all t, the following situations are true: If $y_i^o = \pm 1$ for $1 \le i \le k_1$, then $y_i^o = -1(\pm 1)$ for $k_1 + 1 \le i \le k2$ and $y_i^o = +1$ for $k2 + 1 \le i \le n$. Then, when passing from k_1 to $k_2 + 1$, one always has either that k_1, k_2 did not change (the integer $k_1 - k_2$ increased by 1) or that k_1 or k_2 increased by 1.

So, for example, in the obtained sequence of Bernstein polynomials, we have

n	40	41	42	43	44	45	46	47	48	49	50
$k1$	10	10	10	11	11	11	11	12	12	12	13
$k2$	14	15	16	16	16	17	17	17	17	18	18.

3.6 Approximate Control Optimization for a Nonlinear Dynamic System

Let's consider a general case, when the right side of Eq. (4.1) is a nonlinear function of θ, and an optimization criterion is defined by relationship (4.2). Let's replace $u(t)$ with Bernstein's polynomial $B(t, y_1, \ldots, y_n)$. According to well-known problems value $\Phi(\theta(T))$ is a continuous function from $y_1, \ldots, y_n, |y_i| \le 1$. Then the algorithm of multidimensional polynomial approximation will replace $F(Y(T))$ with a polynomial function of $y_1, \ldots, y_n$ with a uniform small error ε if the value d is large enough:

$$|\Phi(\theta(T)) - \sum_{0 \le a_1 + \ldots + a_n \le d} \lambda(T, a_1, \ldots, a_n) y_1^{a_1} \ldots y_n^{a_n}| \le \varepsilon.$$

If the value ε is sufficiently small, then the solution of the optimization problem for the original criterion differs—on a small value—from the solution of this problem for a polynomial criterion.

Hence, an approximate solution of the optimization problem becomes a standard nonlinear programming problem for a polynomial optimization criterion and for simple restrictions.

Let's consider an example. Let a nonlinear dynamic system look like

$$\dot{\theta_1} = \theta_2,$$

$$\dot{\theta_2} = -2\theta_2(\xi/\tau) - (1/\tau^2)\theta_1 - (\theta_1^3)/100 + u,$$

where $\xi = 0.5, \tau = 1, \theta(0) = 0$.

The following values, dependent on time T, are mean approximation errors of the vector $y(T)$ by a third-order polynomial ($d = 3, m(d, N) = 34$) for $u(t) = B(t, y_1, y_2, y_3, y_4)$:

$T(s)$	3	4	5	6	7	8	9	10	11	12	13	14	15
$\sigma(d, \theta_1) \times 10^4$	0.6	6	9	7	90	2	4	2	20	5	50	80	2
$\sigma(d, \theta_2) \times 10^4$	7	6	5	60	10	5	180	4	8	30	20	20	10.

Apparently, all relative errors of approximation by a third-order polynomial are essentially less than 1.

3.7 Polynomial Approximation with Random Errors

Optimal in the root-mean-square-sense polynomial approximation of conditional expectation—in the presence of random additive errors in independent variables in approximated functions—is carried out by a change of integrand functions in integrals (2.5)–(2.8).

Let's assume

$$\theta = F(y_1 + \xi_1, \ldots, y_N + \xi_N),$$

where $\xi_1, \ldots, \xi_N$ are independent random variables with density of probabilities $p_1(\xi_1), \ldots, p_N(\xi_N)$ on given regions $\Omega_1, \ldots, \Omega_N$.

Then, on the right side of Eq. (2.2) is an estimator $\hat{E}(\theta_| Y)(Y, d)$ for the random variable $E(\theta|Y)$.

To construct an optimal in the root-mean-square-sense polynomial approximation of this value, it is sufficient to add—to Eqs. (2.5)–(2.8)—integration over regions $\Omega_1, \ldots, \Omega_N$, to replace function $F(y_1, \ldots, y_N)$ with function $F(y_1 + \xi_1, \ldots, y_N + \xi_N)$, and to replace product $p(Y)dY$ with a product

$$p(Y)p(\xi_1)\cdots p(\xi_N)d\xi_1 \cdots d\xi_N.$$

3.8 Identification of a "Black Box"

Let's consider a situation when the statistical connection of vectors θ and Y is defined experimentally. Let a vector function

$$\theta = F(Y)$$

of dimensionality $N \times 1$ from N arguments $y_1, \ldots, y_N$ be set by using a computer to form an algorithm of a complicated program; also, there is a region of the vector of arguments: $Y \in \Omega_Y$. The vector Y serves as the program input, and vector θ as its output.

In practice, it might be necessary to have a simpler and more approximate representation of this vector function that allows one to avoid using such a complicated program. As such a representation, we consider the polynomial representation $F(Y)$ in the form of a vector series, whose members are products of vectorial weight coefficients by integer powers of arguments:

$$F(Y) \sim \sum_{0 \le k_1 + \ldots + k_N \le d} \lambda(k_1, \ldots, k_N) y_1^{k_1} \ldots y_N^{k_N}, \tag{8.1}$$

where the number of series members m is equal to an earlier considered integer function integer function $m(d, N)$, such that

$$m(d, N) \to \infty, d \to \infty.$$

In order to solve the formulated problem effectively, we will assume that Y is a random vector for which there has been assigned a distribution function on an a priori region Ω_Y. If there are no a priori data about statistical characteristics of inputs of the program, one ought to consider, apparently, that all random components of vector Y are independent and distributed uniformly.

Under the conditions formulated, the errors of representation $F(Y)$ in Eq. (8.1) will be random. Hence, vectorial weight coefficients $\lambda(k_1, \ldots, k_N)$ should produce—at a prescribed d—the least root-mean-square errors of the representation $F(Y)$ by a polynomial series (8.1).

Let's assume that W is a random vector of dimensionality $m(d, N) \times 1$, composed from integer powers of values $y_1, \ldots, y_N$ at a given integer d. In Sect. 3.7, we showed that the best in the root-mean-square-sense on region Ω_Y polynomial representation of vector-function $F(Y)$ produces a vector

$$\hat{E}(Y, m) = E(\theta) + L(m)Q(m)^{-1}(W - E(W)), \tag{8.2}$$

where

$$L(m) = E(\theta - E(\theta))(W - E(W))^T,$$
$$Q(m) = E((W - E(W))((W - E(W))^T.$$

One can easily find a matrix of covariances $Q(m)$ of random components W, and a matrix $Q(m)^{-1}$, as we are given the distribution of components, and by definition the linear independence of vector W's components is provided. Statistical approximations for vector $E(\theta)$ and for matrix $L(m)$ are to be found by a Monte Carlo method.

Let's implement a series of r independent statistical realizations of random vectors Y and fix in memory r pairs of vectors Y and of corresponding vectors θ, generated by the computer program. Under known formulas of mathematical statistics, we will find a vector $E_r(\theta)$ and a matrix $L_r(m)$, which serve as statistical approximations for the vector $E(\theta)$ and matrix $L(m)$. On the validity of the known conditions, according to the central limit theorem for $r \to \infty$,

$$E_r(\theta) \to E(\theta), L_r(m) \to L(m). \tag{8.3}$$

Vector $E_r(\theta)$ and matrix $L_r(m)$ serve as experimental information carriers about a hidden mechanism of how the computer program functions. The basic theorem of the polynomial approximation method implies that

$$m \to \infty, r \to \infty : max_{Y \in \Omega_Y} |F(Y) - \hat{\theta}(Y, m)| \to 0.$$

Let's suppose we have a static system model whose inputs serve as components of vector Y and whose outputs serve as components of vector θ. The connection of input Y and output θ is performed by an unknown function $\theta = F(Y)$ that permits one to call this static system a "black box". As a result of experimental study of the system, it is necessary to find an analytical or algorithmic representation of vector-function $F(Y)$.

The presented sequence of calculations solves the formulated problem of identification of the "black box".

References

1. Stoun M (1937) Applications of the theory of Boolean rings to general topology trans. Am Math Soc 41:375–481
2. Pontryagin LS et al (1969) Mathematical theory of optimal processes. Physmatgiz, Moscow
3. Chernousko FL (1988) Valuation of phase state of dynamic systems, method of ellipsoids. Physmatlit, Moscow

Chapter 4
Polynomial Approximation Technique Applied to Inverse Vector-Function

4.1 Introduction

Many assessment and control problems in applied mathematics are, in fact, problems of numerically determining the parameter vector, θ, to elucidate the statement of the problem to be fulfilled.

The task of assessment and judgment by observation data often consists of determining the parameter vector to find solutions of maximum likelihood equations.

Suppose we need to know the state vector corresponding to the dynamic equilibrium point of the nonlinear dynamic system that is undergoing stability analysis. In this study, we shall find this vector when the system of nonlinear algebraic equations is solved. The system itself will be determined if we set to zero all time derivatives of the state vector components in the dynamic system equations.

The optimum conditions specified by Pontryagin's maximum principle require the two-point boundary value problem to be solved. The problem is, in fact, a parameter vector determination problem, the parameter vector being an unknown initial data vector for some differential equation system. Both the system state vector and the vector of costate variables satisfy the equation.

In all of the preceding and similar cases, the constructive solution of the problem can be reduced to finding a numerical solution of the equation in the form

$$F(\theta) = Y_N, \tag{1.1}$$

where $\theta \in R^q$ is still the unknown Eq. (1.1) root vector, where the vector is contained in an a priori domain Ω_θ, where the roots of Eq. (1.1) exist; $Y_N \in R^N$ is a vector of predetermined real numbers found, for example, experimentally; moreover, Y_N is allowed to have an additive random component, $F(\theta)$, which is an explicit or implicit vector-function. Next, we shall give an example of the situation connected with the two-point boundary value problem where the above-mentioned vector-function is implicit. Let the model of the dynamic system be defined as a differential equation as follows:

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_4,

$$dX/dt = \Phi(X), \tag{1.2}$$

where X is the state vector of the dynamic system, whereas the order of system (1.2) is an even number. Next, we suppose that half of the $X(0)$ vector components are predefined, while the second (unknown) half of them are represented by components of the vector θ. Next, we take for Y_N a vector composed of one half of the $X(T)$ vector components. Let the vector θ be predefined, and therefore the initial data vector $X(0)$ is predetermined accordingly. Then the vector Y_N is unequivocally determined (as a numerical solution of the Cauchy problem) through numerical integration of Eq. (1.2) over an interval $[0, T]$.

In the outlined situation, we introduce an implicit vector-function $F(\theta)$ to provide a functional connection of two vectors, Y_N and θ. The solution of Eq. (1.1) is afforded by the vector-function $F^{-1}(Y_N)$, which is the inverse of the vector-function $F(\theta)$.

It is defined over an area $\Omega_{Y_N} \in R^N$ that is obtained from the a priori domain Ω_θ using a mapping $F(\theta)$. If the problem is one-dimensional (i.e., $n = N = 1$) and the function $F(\theta)$ comprises an analytical function near the point $\theta = 0$, then the Lagrange expansion exists and determines the inverse function $F^{-1}(Y_N)$ via a power series in Y_N. Serial expansion coefficients depend on the derivatives of $F(\theta)$ at the point $\theta = 0$. The series converges near $Y_N = 0$. Nevertheless, practical use of the Lagrange series is associated with the difficulty of determining its radius of convergence; furthermore, repeated differentiation of the function $F(\theta)$ needed. Note that an option of uniformly convergent approximation with $n = 1$ is afforded by the Bernstein polynomials used in the proof of Weierstrass's theorem. If we approximate the inverse function, however, computational difficulties arise when performing an attempt of uniform markup of the interval Ω_Y.

In contemporary applied mathematics, the only multipurpose approach to solving such problems is the iterative Newton technique in its various implementations. These implementations generally rely on differential properties of the vector-function $F(\theta)$ that are represented by its gradient vectors (see, e.g., [1, 2]). In its simplest form, the method mentioned above determines the iteration process according to the equation

$$z_{i+1} = z_i - (dF(z_i)/dz)^{-1}(F(z_i) - Y_N), \tag{1.3}$$

where z_i is the ith approach vector for $\theta, 0$; $dF(z_i)/dz$ is the partial derivatives matrix of the vector-function $F(\theta)$ components with respect to z_i vector components (Jacobian matrix).

We shall arrive at formula (1.3) if we take a sum of the two first vector members in the power series expansion of the vector-function $F(\theta)$ components about a point θ_i or, alternatively, if we write out a stationary condition for the sum of squares of corresponding differences like $(F(\theta) - Y_N)^T (F(\theta) - Y_N)$ to be minimized.

The computational process converges if $z_i \to \theta$ as i increases.

Notorious difficulties are associated with application of gradient techniques:

- Equation (1.3) is based on linearization of the $F(\theta)$ components near the vector z_i; therefore, what is needed to achieve iteration process convergence is some extent

of proximity between the zero approach vector z_0 and the unknown vector θ to determine roots of Eq. (1.1); it is the choice of the appropriate z_0 that is a central problem to solve, with the involvement of both a priori data and heuristic methods.

- Calculation of the matrix elements $dF(z_i)/dz$ is needed; such calculations often involve challenges. It is especially true in the case of the implicitly defined vector-function $F(\theta)$ or, otherwise, if nondifferentiable functions are present in its explicit definition; for example, partial derivatives may fail to exist at all as in the synthesis of the optimum control problem since in that case the maximum principle affords discontinuous functions of at least some components of the vector of costate variables.
- Equation (1.3) suggests that the iteration process depends on differential (local) characteristics of the vector-function $F(\ldots)$ in some intermediate points z_i (unrelated, in general, to the properties of this vector-function at the point of interest, θ; the calculations could therefore be of poor quality if the matrix $dF(z_i)/dz$ comprises nearly a singular matrix or is characterized by a nonvanishing condition number at z_i.
- Another prerequisite is that the residual minima function $(F(\theta)-Y)^T(F(\theta)-Y_N)$ must be free of complicated topography; such a topography usually arises together with local minima in the proximity of the global minimum point of the function.
- Any substitution of Y_N for another vector requires for organization of a new iteration process with a new zero approach the vector z_0 introduced accordingly; the data previously calculated with the earlier vector Y_N are now useless.

Next, we present the polynomial approximation technique applied to an inverse vector-function, a gradient-free method that is essentially free of the disadvantages inherent to earlier techniques.

Set $N = q$. Next, suppose that Eq. (1.1) defines a univalent mapping (bijection) Ω_θ to Ω_{Y_N}, being a compact space, and that there exists a vector-function $F^{-1}(Y_N)$

$$F^{-1}(Y_N) = F^{-1}(y_1, \ldots, y_N),$$

where $y_1, \ldots, y_N$ are components of the vector Y_N. The vector-function F^{-1} is the inverse of the vector-function F and is a continuous function over a region Ω_{Y_N}. Then, applying the polynomial approximation technique yields an asymptotic representation that is the integral counterpart of the multivariate Taylor series:

$$F^{-1}(y_1, \ldots, y_N) \simeq \sum \lambda(a_1, \ldots, a_N) y_1^{a_1} \cdots y_N^{a_N}, \tag{1.4}$$

where $a_1, \ldots, a_N$ are integer nonnegative solutions of the inequality

$$a_1 + \ldots + a_N \leq d,$$

where d is the prerequisite integer, and $(a_1, \ldots, a_N)$ are real vector weight factors that are defined by the corresponding algorithm, which, in turn, is an operator over Ω_θ, d, F.

If Eq. (1.1) defines a mapping that is not a bijection, and there exist several vector roots of Eq. (1.1), then an arithmetic mean of the roots is to be substituted in the left-hand side of Eq. (1.4).

The basic properties of the series on the right-hand side of Eq. (1.4) are as follows:

- The construction algorithm, being $(a_1, \ldots, a_N)$, only uses operations of the numerical integration over an a priori domain Ω_θ (as a whole or in part); this feature makes it possible to build a series (1.4) with the nondifferentiable vector-function $F(\theta_1, \ldots, \theta_N)$ as well, so the algorithm is essentially robust in terms of local irregularities of $F(\theta)$ at some points within Ω_θ.
- Let us introduce a uniform stochastic measure for vectors θ over Ω_{Y_N}; then there exists a stochastic measure for vectors Y_N over Ω_{Y_N} along with the existence of the nullity vectors' covariance matrix C, the nullity vectors being the vectors of differences between the right- and left-hand sides of Eq. (1.1); the vector series (1.4) converges uniformly over Y_N in the sense that every matrix element of C tends to zero as $d \to \infty$ (of course, assuming that the vector Y_N is free of any random component with the vector θ fixed).
- There is an algorithm to calculate matrix elements of C; the algorithm allows for the evaluation of approximation errors that are about to arise, provided that an integer d, a vector-function $F(\theta)$, and an a priori domain Ω_θ are predetermined.
- Vectors $\lambda(a_1, \ldots, a_N)$ are independent of the components of the vector Y; so an asymptotic representation of the roots could be defined in the case of Eq. (1.1) for any $Y_N \in \Omega_{Y_N}$, without having to use any sophisticated algorithm to calculate vectors $\lambda(a_1, \ldots, a_N)$, provided that the vectors are calculated in advance and are stored in the computer memory.
- Asymptotic properties of the series (1.4) afford an efficient criterion of the fact that no roots of Eq. (1.1) exist within the domain Ω_θ or that for at least some points within the domain the root vectors are not unique; the criterion is that the differences' vector (meaning the differences between the right- and left-hand sides of Eq. (1.4)) does not decrease in its norm as integer d increases.

Next, we will test the polynomial approximation technique applied to an inverse vector-function by considering a number of nontrivial applied problems; the problems were numerically solved using computer routines written in Object Pascal.

4.2 The Problem of Polynomial Approximation of an Inverse Vector-Function

Suppose a vector $\theta \in R^q$ satisfies the algebraic equation

$$F(\theta) = Y_N, \tag{2.1}$$

where $\theta \in \Omega_\theta \in R^N$, $Y \in \Omega_{Y_N|\theta} \in R^N$, $q \leq N$ is a predefined vector, Ω_θ is a predefined still restricted domain where the roots of Eq. (2.1) exist, and $F(\ldots)$ is a

predefined continuous vector-function to establish a mapping

$$\Omega_\theta \to \Omega_{Y_N} \in R^N.$$

Assumption 1: The domain Ω_{Y_N} comprises a compact; there exists a continuous vector-function to establish a mapping $\Omega_{Y_N} \to \Omega_\theta$; here we call the vector-function the "inverse function" and denote it $F^{-1}(Y_N)$.

Assumption 2: Each vector $Y \in \Omega_{Y_N}$ is mapped to a unique vector $\theta = F^{-1}(Y_N)$ satisfying Eq. (2.1):

$$F(F^{-1}(Y_N)) = Y_N.$$

Next, we give a well-known variation of sufficient conditions adequate to guarantee the continuity of the inverse vector-function. Assume that there exist relations

$$F(\theta + \delta(\theta)) = Y_N + \delta(Y_N),$$

where $|\delta(Y_N)|$ is a small quantity, and

$$A(\theta, Y_N)\delta(\theta) \simeq \delta(Y_N),$$

where $A(\theta, Y_N)$ is a matrix. Then $F^{-1}(Y_N)$ is a continuous vector-function if the matrix $A(\theta, Y_Y)$ is nonsingular. The conditions so stated are satisfied if, for instance, there exist linearly independent gradient vectors for the vector-function $F(\theta)$ at every point of the a priori domain Ω_θ.

These conditions, however, are merely sufficient ones. The highly efficient application of the polynomial approximation technique to derive inverse vector-functions in the case of nondifferentiable vector-functions $F(\theta)$ is illustrated next by many examples.

Let $y_1, \ldots, y_N$ be components of the vector Y_N. We want to find a polynomial representation

$$F^{-1}(Y) \simeq \sum_{0 \le a_1 + \ldots + a_N \le d} \lambda(a_1, \ldots, a_N) y_1^{a_1} \ldots y_N^{a_N}, \tag{2.2}$$

where d is a predetermined integer, and $\ldots, (a_1, \ldots, a_N)$ are vectors of the dimension $q \times 1$ dependent on $F(.)$, Ω_θ, d, q, and N_N.

Denote by $\hat{F}^{-1}(Y_N, d)$ the right-hand side of Eq. (2.2). The algorithm to represent the vector-function $F^{-1}(Y_N)$ via polynomial series expansion must define the vector coefficients $\ldots, (a_1, \ldots, a_N)$ so that the error $|F^{-1}(Y_N) - \hat{F}^{-1}(Y_N, d)|$ of the representation of the vector-function $F^{-1}(Y_N)$ by the right-hand side of Eq. (2.2) uniformly converges to 0 over Y as d increases:

$$\sup_{Y_N \in \Omega_{Y_N}} |F^{-1}(Y_N) - \hat{F}^{-1}(Y_N, d)| \to 0, d \to \infty. \tag{2.3}$$

The number of terms in the truncated series on the right-hand side of Eq. (2.2), $m(d, N)$, which is equal to the number of nonnegative solutions of the inequality as follows:

$$0 \leq a_1 + \ldots + a_N \leq d,$$

is further determined by the recursion formula of Lemma 4.1 (see Chap. 2 for details).

No differentiability of the $F(\theta)$ components is required by the conditions of the problem: In particular, the rank of the partial derivative matrix is not a "must-consider" item (nonsingularity of the matrix is to be conventionally stated when root vector existence analysis is performed).

The method under development [3–6] merely uses integration operations over a domain $F(\theta)$ as applied to some functions of the $F(\theta)$ components. Its efficiency, therefore, is independent of the local properties of the vector-function $F(\theta)$, the features usually determining the performance of the conventional methods, which in turn use various gradient-based techniques [2] applied to the vector-function $F(\theta)$. The method does not require an arbitrary choice of the initial guess vector that is a must for both iteration and gradient-based procedures. Instead, we have to assign an a priori (surely, as small as possible) domain, where the roots of Eq. (2.1) definitely exist although, as we will demonstrate at a later stage, the primary a priori domain may be chosen arbitrarily large. In this case, the "reduced" working region will be defined through sequential breakdown (dichotomization) of the initial domain.

To use a Bayesian approach (see Chap. 2 for details), we assume that the vector θ is a random vector over Ω_θ with a stochastic measure $W(\theta)$. Then, according to Eq. (2.1), the variables $y_1, \ldots, y_N, \ldots, y_1^{a1}, \ldots, y_N^{a_N}, \ldots$ will be randomly distributed as well. Denote these variables by $w_1, \ldots, w_{m(d,N)}$ and let them be components of the random vector of dimension $m(d, N) \times 1$. The vector W is a function of

$$W = W(\theta).$$

From (2.1) it follows that the joint probability density of the random vectors θ, Y can be represented by the following relations:

$$p(\theta, Y) = p(\theta)\delta(Y_N - F(\theta)) = p(Y_N)\delta(\theta - F^{-1}(Y_N)).$$

In order to use polynomial approximation of the conditional expectation vector presented in detail in Chap. 2, we value that

$$E(\theta|Y_N) = \int_{\theta \in \Omega_\theta} \theta\delta(\theta - F^{-1}(Y_N))d\theta = F^{-1}(Y_N). \tag{2.4}$$

Equations (2.2)–(2.6) thus afford the polynomial approximation of the inverse function $F^{-1}(Y)$ for the problem of interest.

The first (primary) and second (secondary) statistical moments of the vectors θ, $W(Y)$ are represented by the relations as follows:

$$E(F(\theta)) = \int_{\Omega_\theta} F(\theta)p(\theta)d\theta, \tag{2.5}$$

$$E(W) = E(W(\theta)) = \| \int_{\Omega_\theta} w_i(\theta)p(\theta)d\theta \|, i = 1, \ldots, m(d, N), \tag{2.6}$$

$$L = E((\theta - E(\theta))W(\theta)^T) =$$

$$\| \int_{\Omega_{(\theta)}} (\theta - E(\theta))w_i(\theta)p(\theta)d\theta \|^T, i = 1, \ldots, m(d, N), \tag{2.7}$$

$$Q = E((W(\theta) - E(W(\theta)))W(\theta)^T) =$$

$$\| \int_{\Omega_{(\theta)}} (w_i(\theta)w_j(\theta)p(\theta)d\theta \|^T, i, j = 1, \ldots, m(d, N) \tag{2.8}$$

Theorem 2.1 *Given the predefined integer d, the root-mean-square optimum estimate of the root vector in Eq. (2.1) is given by*

$$\hat{F}^{-1}(Y_N, d)^o = E(\theta) + \Lambda^o(W(Y_N) - E(W(Y_N))), \tag{2.9}$$

where

$$\Lambda^o Q = L. \tag{2.10}$$

The assertion of the theorem is obviously implied by formula (2.6) of Chap. 2.

Conclusion 1:

$$\sup_{Y_N \in \Omega_{Y_N}} |F^{-1}(Y_N) - \hat{F}^{-1}(Y_N, d)^o| \to 0, d \to \infty. \tag{2.11}$$

This follows from Theorem 2.1 of Chap. 2.
Conclusion 2: If Q is a nonsingular matrix, then

$$\Lambda^o = LQ^{-1}. \tag{2.12}$$

In this case, the optimum error estimate covariance matrix, $C(d)^o$, can be represented by the formula

$$C(d)^o = C_0 - LQ^{-1}L^T, \tag{2.13}$$

where

$$C_0 = E((\theta - E(\theta))(\theta - E(\theta))^T).$$

Denote the ith row of the matrix L by l_i. Then from (2.12) we get

$$\sigma_i(d)^2 = \sigma_i(0)^2 - l_i Q^{-1} l_i^T,$$

where $\sigma_i(d)^2$ is the error variance for estimating the ith vector component of θ, and $\sigma_i(0)$ is the a priori dispersion of this component (i.e., the ith diagonal entry of the a priori matrix C_0).

Therefore, if and when the error covariance matrix is a nonsingular matrix ($Q > 0$), we have $\sigma_i(d)^2 < \sigma(0)^2$.

Conclusion 3: Assume $d_1 < d_2$. It follows from the definition of the root-mean-square optimum estimate for given d that

$$C(d_2)^o \leq C(d_1)^o.$$

Let the vectors Y_1, Y_2 have dimensions $N_1 \times 1$, $N_2 \times 1$, respectively, and also $N_1 < N_2 \times 1$ and $Y_1 \in Y_2$. Then, with the current d, we have

$$C(d, N_2)^o \leq C(d, N_1)^o.$$

It now follows that the "more overdetermined" a system of algebraic equations is, the fewer (on average) errors of the root vector component estimate using polynomial approximation technique there are.

Conclusion 4: The formulas (2.9), (2.10) afford an estimate $\hat{F}^{-1}(Y, d)^o$, which is the root-mean-square optimum estimate, for any integer d and for any statistical construction of the random vector W, including any vector to have a singular covariance matrix Q, if only the weight matrix, Λ^o, satisfies the matrix equation (2.10).

This follows from Conclusion 3, the consequence of Lemma 1.1 of Chap. 1.

Conclusion 5: The optimum linear estimate vector as defined by Eq. (2.9) is the unique vector.

This follows from Conclusion 4, the consequence of Lemma 1.1 of Chap. 1.

Conclusion 6: In an exact calculation of integrals (2.5)–(2.8) and with the value of d being large enough, the estimate $\hat{F^{-1}}(Y_N, d)^o$ is close to the "estimate" value, $F^{-1}(Y_N)$, and is essentially independent of the probability density selection, $F(\theta)$.

This assertion follows from the identical equation (2.4) and the relation (2.11).

Note that in the case of general function—in particular, nonlinear and nondifferentiable function $F(\theta)$—the exact solution of the equation is impossible to obtain (2.1).

Let $F_1(\theta), \ldots, F_N(\theta), y_1, \ldots, y_N$ be components of the vectors $F(\theta), Y$, correspondingly. Then, given Y and d, the accuracy of the problem solution will be represented with relative residual values as follows:

$$re_1(Y_N, d) = |y_1 - F_1(\hat{F}_1^{-1}(Y_N, d))|/|y_1|,$$

.....

$$re_N(Y_N, d) = |y_N - F_N(\hat{F}_N^{-1}(Y, d))|/|y_N|.$$

Given d, the formulas to get the estimate of the error variances over a set of all vectors $Y_N \in \Omega_{Y_N}$ are afforded by the Bayesian interpretation, which allows for finding the estimated error random vector covariance matrix. If neither relative residual estimated values nor error variance estimated values $C(d)^o$ (matrix diagonal entries) reduce as d increases, then the method states the following: There are no solutions of Eq. (2.1) over a domain Ω_θ; otherwise, the solution is not unique.

4.3 A Case Where Multiple Root Vectors Exist Along with Partitioning of the a Priori Domain

In the applied problem analysis, a situation where Assumption 2 is not fulfilled—and thus several root vectors of Eq. (2.1) exist over an a priori domain Ω_θ—is commonplace. Suppose $Y \in \Omega_{Y_N}$ and the vectors $\theta_1, \ldots, \theta_k$ are root vectors of Eq. (2.1), each root vector depending on Y_N:

$$\theta_1 = F_1^{-1}(Y_N), \ldots, \theta_k = F_k^{-1}(Y_N).$$

It follows from (2.1) and probability density function normalization requirements that the joint probability density for the random vectors θ, Y and Y can be represented by the equality

$$p(\theta, Y_N) = p(Y_N)p(\theta|Y_N) = p(Y_N)(\delta(\theta_1 - F_1^{Y_N})) + \ldots + \delta(\theta_k - F_k^{Y_N}))/k.$$

Therefore, we arrive at

$$E(\theta|Y_N) = \int_{\Omega_{\theta|Y_N}} \theta \frac{\delta(\theta_1 - F_1^{-1}(Y_N)) + \ldots + \delta(\theta_k - F_k^{-1}))}{k} d\theta =$$

$$\frac{F_1^{-1}(Y_N) + \ldots + F_k^{-1}(Y_N)}{k}.$$

Given integer d, the vector $\hat{F}^{-1}(Y_N, d)$, which is a root-mean-square optimum estimate for the vector $E(\theta|Y_N)$, will be defined by Eqs. (2.9) and (2.10). Then, instead of (2.11), we arrive at

$$\sup_{Y_N \in \Omega_{Y_N}} |\frac{F_1^{-1}(Y_N) + \ldots + F_k^{-1}(Y_N)}{k} - \hat{F}^{-1}(Y_N, d)| \to 0, d \to \infty. \qquad (3.1)$$

Since, under the conditions,

$$F(F_1^{-1}(Y_N)) = Y, \ldots, F(F_k^{-1}(Y_N)) = Y_N,$$

then, in general, we have

$$F((F_1^{-1}(Y_N) + \ldots + F_k^{-1}(Y_N))/k) \neq Y_N.$$

So, given that several root vectors exist within the a priori domain Ω_θ, significant relative residual values $re_1(Y_N, d), \ldots, re_N(Y_N, d)$ will arise, the values not decreasing as the integer d increases. An efficient way to eliminate difficulties associated with the existence of the several root vectors is introduced by the sequential breakdown of the a priori domain Ω_θ into subdomains followed by applying a polynomial approximation technique to each subdomain and calculating the corresponding relative residuals. The breakdown procedure is considered complete when the subdomains are located with relative residuals steadily decreasing as the integer d increases. This stage is further repeated the same way within those subdomains that still contain nondecreasing relative residuals.

4.4 Correctness of the Estimator Algorithm and a Way of Taking Random Observation Items into Account

Suppose $\delta_1, \ldots, \delta_N$ are random component determination errors related to vector Y in Eq. (2.1).

In this case, the formula (2.2) to derive the evaluation vector is as follows:

$$\hat{F}^{-1}(Y, d, \delta) = \sum_{0 \le a_1 + \ldots + a_N \le d} \lambda(a_1, \ldots, a_N)(y_1 + \delta_1)^{a_1} \cdots, (y_N + \delta_N)^{a_N}. \quad (4.1)$$

From (4.1) it follows that the inverse vector-function polynomial approximation operator has the property of correctness [7] since as $\delta_1 \to 0$ and $\delta_N \to 0$, we have

$$|\hat{F}^{-1}(Y_N, d) - \hat{F}^{-1}(Y_N, d, \delta)| \to 0.$$

Having said all the above, however, does not allow for construction errors of the inverse operator itself—these errors arise when determining the weight matrix Λ, the column vectors in being the vectors $\lambda(a_1, \ldots, a_N)$.

The components of the vector-function $F(\theta)$ are linearly independent by condition; therefore, the covariance matrix Q is a positive definite matrix provided that $d > 1$. If $d > 1$, however, and due to errors introduced by numerical evaluation of n-dimensional integrals, the resultant matrix Q may be characterized with small singular values [8] and therefore may have a small condition number. In this case, as is clear from Eq. (2.10), the weight matrix Λ is available only as a matrix of severely noticeable errors.

There are two possible approaches to eliminating the computational problems related to the effect of small singular values: (1) The algorithm inverts the matrix Q and next "strikes out" the rows close to linear dependence. An exemplary approach having this property is the modified Cholesky algorithm, which has been used to solve the most of the applied problems discussed later in this chapter.

(2) A plausible stochastic measure is introduced for random errors $\delta_1, \ldots, \delta_N$ that are inherently taken into account in the construction of the algorithm.

Instead of (2.1), let the following vector equality be true:

$$F(\theta) = Y_N + \xi,$$

where the random vector ξ can be interpreted as an additive noise arising during measurement of the observation vector Y. The random variables $\zeta_1, \ldots, \zeta_N$ are components of the vector ξ.

$$\int_{\Omega} \int_{-a}^{a} \ldots \int_{-a}^{a} dxdydz.$$

We assume them to be statistically independent of θ and uncorrelated, and to be uniformly distributed over an interval [−,]. The random items $\zeta_1, \ldots, \zeta_N$, if present, do not change in any way the performance of the polynomial approximation algorithm provided that the existence of these random variables has been properly taken into account when calculating the first and second statistical moments for the random vector $F(\theta) - \xi$. In this case, the components of vector $E(W)$ and of every resulting matrix $Q(W)$ and $L(W)$ consist of integral terms of the form

$$\int_{\Omega} \int_{-a}^{a} \ldots \int_{-a}^{a} (F_1(\vartheta) - \zeta_1)^{c_1} \ldots$$

$$(F_N(\vartheta) - \zeta_N)^{c_N} \times$$

$$p_\theta(\vartheta) p_{\xi_1}(\zeta) \ldots p_{\xi}(\zeta_N) \times d\vartheta d\zeta_1 \ldots d\zeta_N,$$

where $c_1, \ldots, c_N$ are some integers, and $p_\theta(\vartheta) p_\xi(\zeta_i)$ are probability distribution densities for θ and ζ_i, respectively.

Since $p_\vartheta(zeta_i)$ are constant functions by convention, as mentioned earlier, the terms of the following form become the integral terms upon integration with respect to $\zeta_1, \ldots, \zeta_N$:

$$\int_{\Omega} \frac{(F_1(\vartheta) + a)^{c_1+1} - (F_1(\vartheta) - a)^{c_1+1}}{2a(c_1 + 1)} \times \frac{(F_N(\vartheta) + a)^{c_N+1} - (F_N(\vartheta) - a)^{c_N+1}}{2a(c_N + 1)} p_\theta(\vartheta) d\vartheta. \tag{4.2}$$

If is small, the power differences are appropriately represented by expanding them in a power series.

The integral terms can be calculated using a modification of the trapezoidal rule, exactly as is done if no additive noise is present in experimental observations. Chapter 8 presents a case study for the linear estimation problem using the LS method; the treatment involves algorithmic approaches to eliminating the effect of small singular values.

4.5 Implementations of the Polynomial Approximation Technique Applied to the Inverse Vector-Function

In this section we suppose the a priori distribution of the vector θ over Ω_θ to be uniform, that is, to add obviousness to qualitative considerations given below.

As is illustrated with the following examples, the condition number of the matrix Q increases rapidly as d increases; so the accuracy of calculations to evaluate vector $\lambda(a_1, \ldots, a_N)$ becomes inadequate with large $\sum(a_1 + \ldots + a_N)$. It is thus impossible to get an acceptably accurate solution for Eq. (2.1) as simply as by increasing the number of items in a truncated vector series.

A straightforward way of keeping errors as small as possible with a small number of items $m(d, N)$ in the series of interest is in alternating the form of the a priori domain Ω_θ to get a small a priori domain. Suppose, for instance, the domain is a cube in R^q having an edge s and the center at the origin. A priori [before the vector Y_N in (1.1) is determined] estimated error variations at $W = 0$ are equal to $s^2/12$ due to the uniform a priori distribution of the root vector components. These values are entries of the a priori scalar covariance matrix C_0. Therefore, as follows from formulas of Sect. 2 [see Conclusion 2 from Theorem 2.1, and Eq. (2.13)], if vector W is nonvanishing, then the root vector components' estimated error variations will not exceed $s^2/12$ and, therefore, the smaller the edge of the a priori cube, the smaller the estimate error will be.

These qualitative findings and calculation practice have demonstrated the following procedural step sequence to be optimal when one implements the polynomial approximation technique.

First, we divide the a priori domain Ω_θ into smaller subdomains and, with small integer d (e.g., when $d = 1$ if the estimate vector is a vector linear combination of the vector Y's components), the polynomial approximation algorithm will define the estimate vectors and corresponding residual vectors over each of these subdomains. Several subdomains that have short-length residual vectors may be candidates for the subdomain containing the actual vector θ. Further, we repeat the procedure to calculate the estimate and residual vectors with these several subdomains as before, however with d increased to minimize estimation errors. As practice with calculations has shown, it is the estimate vector characterized by the shortest-length residual vector that typically corresponds to the subdomain containing the target (unknown) root vector of Eq. (2.1).

If, however, a unique root vector of Eq. (2.1) has been shown to exist, then the breakdown procedure may prove to be redundant. In many cases, polynomial approximation of the inverse function $F^{-1}(Y)$ becomes accurate enough upon several iterations and with a small d value, thus avoiding the computational problems that arise when treating an ill-conditioned high-dimension matrix Q.

Let an a priori domain $\Omega_\theta = \Omega_{\theta,0}$ (i.e., the first guessed domain) be a box (parallelepiped) centered around the vector θ_0), a zero-order approximation vector having components Ω_0^i, $i = 1, \ldots, n$, and a_0^i being edge lengths of the box $\Omega_{\theta,0}$ parallel to the coordinate axes.

Further, let the values $(a_0)^2/12$ be the components' a priori variations for the unknown vector θ. These values are diagonal entries of the a priori matrix C_0 in (2.13); the matrix itself is a representation of the unknown root vectors' a priori distribution (dispersion) around θ_0.

The polynomial approximation algorithm defines [over $\Omega_{\theta,0}$] an estimate vector $\hat{F}^{-1}(Y_N, d)_1^o$; we take it as a first-order approximation to find the root vector. From formula (2.13) we find a covariance matrix $C(d)_1$, which characterizes the "post hoc" distribution of the components of the vector θ around the components of the vector $\hat{F}^{-1}(Y_N, d)_1^o$, the first-order approximation vector. As is clear from (2.13), the "post hoc" distribution is narrower than the a priori distribution, due to the fact that the diagonal entries of the matrix $C(d)_1$ are smaller than the diagonal entries of the matrix C_0, provided that Q is a nonsingular matrix.

So, for an unknown root vector θ, the average distance to the first-order approximation vector is shorter than the average distance to the zero-order approximation vector.

The box $\Omega_{\theta,1}$ (the domain where the first-order approximation vector exists) has a vector $\hat{F}^{-1}(Y_N, d)_1^o$ for its center, and its edge lengths are 1i $a_0^i = 6(c_{i,i})^{1/2}$. $i = 1, \ldots, n$, where $c_{i,i}$ is the ith diagonal entry of the covariance matrix $C(d)_1$.

Next, the polynomial approximation algorithm defines over $\Omega_{\theta,1}$ an estimate vector $\hat{F}^{-1}(Y_Y, d)_2^o$; we take it as a second-order approximation to find the root vector. From formula (2.13), we find a covariance matrix $C(d)_2$, which characterizes the new "post hoc" distribution of the components of the vector θ around the components of the vector $\hat{F}^{-1}(Y_N, d)_2^o$, the second-order approximation vector. As is clear from (2.13), the new "post hoc" distribution is narrower than the previous "post hoc" distribution, due to the fact that the diagonal entries of the matrix $C(d)_2$ are smaller than the diagonal entries of the matrix $C(d)_1$, provided that new Q is a nonsingular matrix. So, for an unknown root vector θ, the average distance both to and from the second-order approximation vector is shorter than the average distance to the first-order approximation vector.

A repetitive procedure implemented as described above suggests that the iteration process converges in the mean if only the covariance matrices for the vector W (with predefined integer d) have no singularities at the "post hoc" distribution boxes. This statement is true provided that all "post hoc" distribution spaces are contained within the initial a priori domain.

Note that the computer program developed to implement the method presented above defines the estimate vector by the formula

$$\hat{F}^{-1}(Y_N, d)^o = E(\theta) + LV,$$

where the vector V is a solution of the linear equation

$$QV = W - E(W),$$

and the solution is found using the Cholesky-type algorithm. The original algorithm has been modified so that it works even when running into the "worst-case scenario", that is, when the covariance matrix Q has become a singular matrix due to computational errors.

In the following examples, the a priori domain where the roots exist is a box (parallelepiped) in R_q, and q-variate integrals are calculated using a modification of the trapezoidal rule (see Chap. 2 for details); a dedicated integration procedure has been included as a part of the computer program developed to implement the polynomial approximation method applied to the inverse vector-function.

The algorithm divides the a priori parallelepiped into r equal parts so that the box becomes covered with a grid to partition it into r^q plate-like elements (PE) of volume. Each integral (which is a representation of the vector and matrix components associated with the first and second statistical moments as discussed above) is a sum of rn integrals taken over all PE. For every PE, the integral is evaluated upon completion of the multivariate straight-line interpolation procedure applied to the integrands that are present in integral approximations for the vector and matrix components. Therefore, the evaluated expression of the integral over each PE is a multilinear function at each of 2^q vertices of this PE. Each vertex, however, belongs to several adjacent ("nearest-neighbor") PEs, where the number of the "neighbors" varies from 1 to 2^q. The computer program implementing the modification of the trapezoidal rule requires the calculation of the integrands to be performed only once at each point of the grid covering the a priori parallelepiped, to ensure evaluation of integrals for all "nearest-neighbor" PEs.

Any more complicated shape of the a priori domain is replaced with a parallelepiped to describe the domain. In this case, the integrands in the q-variate integrals are multiplied by the characteristic function equal to unity within the a priori field and equal to zero at all other points of the box.

If the vector-function $F(\theta)$ in (2.1) is an implicit function, the integrand values calculated at the points of the grid covering the a priori domain are determined by numerical integration of the corresponding differential equations. In this case, some variations of the initial conditions, which are the grid points as discussed above, may be excluded to meet the boundary conditions put in for the current state vectors of the dynamic system.

Basically, a simple search method could be considered as an alternative to the present method; here we emphasize that the computational basis of the present method is the approximate determination of n-variate integrals through calculating

the grid point values of the components for the vector-function $F(\theta)$, the grid covering the a priori domain Ω_θ. In case of a simple search method, the a priori domain is also covered with the grid points, then the components of $F(\theta)$ are calculated, and, finally, the point is searched where the inherent vector could approximate the root vector θ for Eq. (2.1). Unfortunately, the number of points within such a grid must be exceedingly large, and it is almost impractical to achieve the desired accuracy in solving Eq. (2.1)—the accuracy that has been demonstrated using the polynomial approximation technique in applications described later in this chapter. In some cases (see Chap. 9 for details), we provide comparative treatment using both a simple search method (to find a rough solution of the problem) and a polynomial approximation technique, to afford the refined, more accurate solution.

4.6 Numerical Solutions of Underdetermined and Overdetermined Systems of Linear Algebraic Solutions

Now we illustrate the accuracy of the solution of (2.1) that is achieved in the numerical evaluation of multivariate integrals by solving a system of linear algebraic equations with $q = 4, 6$ and with Hilbert matrices, which are known to have large condition numbers and an integer-valued inverse matrix. Let (2.1) have the form

$$A\theta = Y_N,$$

where $A(i, j) = 1/(i + j - 1)$; the condition number $Cond(A)$ is $1.5514\ 10^4$ with $q = 4$.

It was shown in Chap. 1 that if the vector-function $F(\theta)$ in (2.1) is a linear function of the vector θ, then the weight matrix Λ does not change in replacing $F(\theta)$ with $F(\theta) + v$, where v is any vector independent of the (still unknown) vector θ. Therefore, for a linear problem, the result of applying the polynomial approximation algorithm is the invariant with respect to the a priori domain Ω_θ form alteration—we assume here that the domain can be of any shape [and, surely, the interval where every component of the root vector (2.1) exists should not degenerate into a point].

Next we define an a priori cube through inequalities $-1 \leq \theta_i \leq 1$, setting all components of the vector Y equal to 1, and introduce a variable to characterize the error in solving the simultaneous linear algebraic equations, which is the square root of the sum of squared residuals.

Since the root vector θ is a linear function of the components of vector Y, the accurate solution is to be available even if we apply linear approximation alone (this case corresponds to $d = 1$ within the polynomial approximation algorithm). In the case $d > 1$, the approximation also includes nonlinear items, which are integer power series of the components of vector Y. If the integrals are accurately calculated, the computer software is free of bugs, and computational tolerances are small to negligible, then—as is shown in Chap. 1—the coefficients preceding nonlinear terms

vanish, and the results of the simultaneous linear system solution are the same with $d = 1$ and $d > 1$. In practice, however, this increases with d, which is due to the computational tolerances and computer software bugs.

These statements are clearly supported with computational results. Four-variate integrals were calculated using a modification of the trapezoidal rule; in so doing, we saw that the Cartesian coordinates of the grid points (here, as before, the grid covers the a priori cube) were the points that arose when we divided the cube edges into kd equal parts. We performed calculations with different k values:

$$d = 1,$$

k	2	4	6	8	10
Δ	1.99×10^{-13}	5.51×10^{-13}	1.92×10^{-12}	$l.52 \times 10^{-12}$	4.77×10^{-12}

The root vector components were integers to the eighth decimal place:

$$\theta_1 = -4\ \theta_2 = 60\ \theta_3 = -180\ \theta_4 = 149.$$

The errors in simultaneous linear algebraic equations' solution achieve their minimum values with $r = 2$, that is, when four-variate integrals are evaluated after the dichotomization of each edge of the four-variate a priori cube into two equal parts. The errors increase with the number of dividing points; presumably, this is due to accumulated computational errors:

$$d = 2,$$

k	2	4	6	8	10
δ	1.00×10^{-7}	2.01×10^{-7}	2.94×10^{-7}	1.26×10^{-7}	1.87×10^{-6}

Thus, the errors in solving the system of linear algebraic equations have increased enormously (by several orders of magnitude) due to maintaining square terms in the approximation of the root vector.

Next, consider a situation when the number of the vector components for Y_N is less than q; in particular, it is $N = 1$. Then we assume that the system of linear equations consists of the first row of the previously considered system. The polynomial approximation algorithm works in this situation as before, with no changes. Let $d = 5$. Then, with $q = 4$, we get $m(d, N) = 125$. What this means is that the root vector satisfying the remaining first row is approximated by a vector series truncated to 125 items, and this approximation is optimized in the mean square. The a priori covariance matrix Q, however, has a dimension of 125×125 and is a singular matrix.

The modified Cholesky algorithm, when applied to the matrix, has consistently eliminated linearly dependent rows and columns and has transformed it into a 6×6

array. The roots of this underdetermined system of linear algebraic equations are as follows:

$$\theta_1 = 0.70350, \theta_2 = 0.36237, \theta_3 = 0.22411, \theta_4 = 0.16246.$$

In this case of the singular linear system, the solution errors are small to negligible: $\Delta = 1.68 \times 10^{-17}$.

If the linear algebraic equations are to be solved simultaneously, then $A\theta = Y_N$, as follows from Chap. 1, and with $q \leq N$, we have

$$\hat{F}^{-1}(Y_N, d) = B^{-1}Y_1 = \theta,$$

where it is a nonsingular $q \times N$ array composed of q rows of the matrix A, and Y_1 is a q-variate vector composed of corresponding components of the vector Y_N.

Next, we illustrate the effect the "extent of underdetermination" has on the solution's accuracy in the case of linear algebraic equations. In particular, we consider a system of six linear equations. Suppose there are mutually independent random errors on the right-hand side of each equation. The errors are evenly distributed over an interval $[-0.0001, 0.0001]$, and the equations are as follows

(let the a priori domain θ be a parallelepiped in R^6):

$$\theta_1 + \theta_2 - \theta_3 + \theta_4 + 9\theta_5 - 2\theta_6 = y_1,$$

$$3\theta_1 + 2\theta_2 - 8\theta_3 - \theta 4 + 11\theta_5 - \theta_6 = y_2,$$

$$4\theta_1 - 8\theta_2 + 9\theta_3 + \theta_4 + 2\theta_5 + 11\theta_6 = y_3,$$

$$2\theta_1 + \theta_2 - \theta_3 - 3\theta_4 + \theta_5 + \theta_6 = y_4,$$

$$-\theta_1 - 2\theta_2 + \theta_3 + 3\theta_4 - 15\theta_5 - \theta_6 = y_5,$$

$$7\theta_1 + \theta_2 - \theta_3 + 7\theta_4 + \theta_5 - \theta_6 = y_6.$$

Let's assume that the a priori domain Ω_θ is a box

$$-500 \leq \theta_1, \theta_2, \theta_3 \leq 500; -0.5 \leq \theta_4, \theta_5, \theta_6 \leq 0.5.$$

It is clear from the a priori data that there are two enormously different a priori areas of existence, for the first and second root triples.

The unknown parameters θ_i are uniformly distributed with σ_i^a errors as follows:

σ_1^a	σ_2^a	σ_3^a	σ_4^a	σ_5^a	σ_6^a
353, 553	353, 553	353, 553	0.353553	0.353553	0.353553.

The system has been solved numerically in accordance with a recursive algorithm as described in Chap. 1, with no explicit definition of the inverse matrix. The system's equations were treated in sequence, from the first to the sixth. The recursive algorithm sequentially determines the optimum (in the mean square) six-variate vectors, that is, the solutions of the first equation, the first and second equations solved simultaneously, etc. In the sixth step, the recursive algorithm finds the solution of the determined system of six linear equations.

Given next are values of $\sigma_i/\sigma_i^a, i = 1, \ldots, 6$, namely, ratios of mean-square deviation (MSD) errors at each step of the computation with reference to the corresponding a priori MSD error. The ratios characterize the progress of the solution refinement (in terms of the progress toward accuracy) for an underdetermined system of linear algebraic equations as a function of the number of equations in the system. σ_i values were determined using a Monte Carlo method (with 100,000 samplings), assuming that the random variables θ are uniformly distributed over an a priori parallelepiped $Omega_\theta$:

σ_1/σ_1^a	σ_2/σ_2^a	σ_3/σ_3^a	σ_4/σ_4^a	σ_5/σ_5^a	σ_6/σ_6^a
$step1$					
0.66758	0.66729	0.66820	0.81589	0.81737	0.81586
$step2$					
0.62273	0.51896	0.10380	0.81585	0.81735	0.81586
$step3$					
0.00523	0.00433	0.00196	0.81585	0.81735	0.81586
$step4$					
0.00102	0.00108	0.00103	0.77904	0.36499	0.76987
$step5$					
0.00078	0.00015	0.00010	0.61678	0.00917	0.53488
$step6$					
6.3×10^{-13}	7.4×10^{-12}	$1.8 \times 10-12$	1.7×10^{-9}	6.3×10^{-10}	4.2×10^{-9}.

It is clear from the presented data that the first three roots of the underdetermined system—the roots distributed more widely—are determined with errors below 1as early as upon completion of three iteration steps. Once the underdetermined system has been transformed into the determined system, all six roots can be determined with minute errors.

A similar consideration carried out for the system of six linear equations using a Hilbert matrix results in the following data:

$$1.48 \times 10^{-5}\ 1.25 \times 10^{-4}\ 1.53 \times 10^{-5}\ 0.601\ 0.657\ 0.614.$$

We see that the last three roots are available with large errors—even though the system is the determined one. If, however, we assume that no accidental additive errors present on the right-hand sides of the equations, then the accuracy increases markedly:

Step 6

$$3.66 \times 10^{-7}\ 4.75 \times 10^{-6}\ 3.12 \times 10^{-7}\ 0.0774\ 0.167\ 0.0976.$$

4.7 Solving Simultaneous Equations with Nonlinearities Expressed by Integer Power Series

Let a component of the vector-function $F(\theta)$ be as follows:

$$F_i(\theta) = \sum \theta_1^{c1_i} \cdots \theta_q^{cq_i}.$$

Then the integrands in n-variate integrals representing components of the vector and matrices $E(W)$, Q, and L are as follows:

$$(\sum \theta_1^{c1_1} \cdots \theta_q^{cq_1})^{\alpha_1} \cdots (\sum \theta_1^{c1_q} \cdots \theta_n^{c_q})^{\alpha_q}.$$

In the above expression, $c_{1_i}, \cdots c_{q_i}, \alpha_1, \cdots \alpha_q$ are some integers. When developing the integer power values of corresponding multivariate polynomials, we obtain the integrands as above in the form of linear combinations of items $\theta_1^{\beta_1} \cdots \theta_q^{\beta_q}$. Upon integration of the integrands between predefined limits, we get accurate vector and matrix component values (vectors and matrices represent a priori statistical moments). Computational practice, however, has demonstrated that, with $d > 2$, the number of items in the linear combinations is extremely large as is the time of computation. That is why later, in all considered problems, we have evaluated the integrals of the functions in the form $F_1(\theta)^{k_1} \cdots F_q(\theta)^{k_q}$ using a modification of the trapezoidal rule. It appears that the time of computation is greatly decreased compared to the time needed to calculate integrals accurately.

Next, we apply the polynomial approximation technique to find the roots of the system of four algebraic equations with four unknowns, with nonlinearities expressed by integer power series. The system of equations is as follows:

$$\theta_1 + 2\theta_1^2 + 4\theta_2 - 0.2\theta_3 - \theta_4 = y_1,$$

$$-2\theta_1 + \theta_2 + 2\theta_2^2 + 0.8\theta_4 = y_2,$$

$$0.7\theta_1^3 + 0.5\theta_2 - \theta_3 + 7\theta_4 = y_3,$$

$$\theta_1 - \theta_2 + \theta_3 - 2\theta_4^2 = y_4.$$

We shall find the $y1$, $y2$, $y3$, $y4$ values if we set on the left-side part of the system:

$$\theta_1 = 1, \theta_2 = 2, \theta_3 = 3, \theta_4 = 4.$$

The a priori domain is limited by

$$-2 \le \theta_1 \le 3, -1 \le \theta_2 \le 4, 1 \le \theta_3 \le 6, 2 \le \theta_4 \le 7.$$

Here are values of r, calculated as

$$r = log_{10}(Cond(Q(m)))$$

and represented versus $m = m(d, N)$:

m	40	80	120	160	200	240	280	320
r	13.25	18.29	20.23	21.98	23.11	25.41	25.36	26.92.

As is clear from the above data, the condition numbers of the matrix $Q(m)$ become larger as m increases. Note that "saturation" of r at large m is presumably due to inherent errors of the $Cond(Q(m))$ function as implemented in the MATLAB® 5.x software. It was demonstrated by calculation that the vector measure components describing the nonlinearity are close to unity for this and related systems of algebraic equations. This implies that in both cases, the systems are considerably nonlinear. Set $m = 494 (d = 8)$, and then $Cond(Q(494)) = 5 \times 10^{32}$. The polynomial approximation algorithm has calculated the following values for the estimate vector and the components of the relative residual vector:

$\hat{F}^{-1}(Y, 494)$	0.9503	1.8869	3.0336	4.0055
$re(i, Y, 494)$	0.3762	0.0793	0.0056	0.0003.

We can see that the first component of the relative residual vector is not so very small. Next, diminish the number of segments within the interval approximating the vector series by setting $m = 310$ and then

$$Cond(Q(310)) = 1.1542 \times 10^{24}.$$

The result is that procedural approximation errors will be increased when approximating the conditional expectation vector, while the $Cond(Q(m))$ value will be decreased by eight orders of magnitude.

In this case, the algorithm has calculated the following values for the estimate vector and the relative residual vector components:

$\hat{F}^{-1}(Y, 310)$	0.9999	1.999	2.9999	3.9999
$re(i, Y, 310)$	3.50×10^{-5}	9.29×10^{-7}	5.05×10^{-7}	8.26×10^{-8}.

The resulting solution can be regarded as almost accurate since the relative residual vector's components are vanishing. So, for the equation of interest, the condition

number decreased by eight orders of magnitude is entirely counterbalanced by the reduced number of segments within the interval approximating the vector series (now 310 instead of 494). This fact enabled us to get out of reducing the a priori root existence domain that is otherwise necessary to get low values of the relative residual vector components.

4.8 Solving Simultaneous Equations with Nonlinearities Expressed by Trigonometric Functions, Exponentials, and Functions Including Modulus

The system of equations including nondifferentiable functions on their left-hand sides is as follows:

$$sin(\theta_1) + 2\theta_1^2 + 0.4Cos(\theta_2 - 0.2\theta_3) - \theta_4 = y_1,$$

$$|2\theta_1) + \theta_2| + 2exp(\theta_2) + 0.8sin(\theta_4) = y_2,$$

$$\theta_1^3 + |0.5\theta_2 + \theta_3| + 7\theta_4 = y_3,$$

$$\theta_1 - sin(\theta_2 + \theta_3) + 2\theta_4^2 = y_4.$$

We shall find the $y1, y2, y3,$ and $y4$ values if we set, on the left-side part of the system,

$$\theta_1 = 1, \theta_2 = 2, \theta_3 = 3, \theta_4 = 4.$$

The a priori domain is limited by

$$-2 \le \theta_1 \le 3, -1 \le \theta_2 \le 4, 1 \le \theta_3 \le 6, 2 \le \theta_4 \le 7.$$

Here are the values of r, calculated as $r = log_{10}(Cond(Q(m))$ and represented versus m; the data are as in Sect. 4.7. Set $m = 494(d = 8)$, and then $Cond(Q(494))$ $= 7 \times 10^3 2$.

In this case, the algorithm has calculated the following values for the estimate vector and the relative residual vector components:

$$\begin{array}{ll} \hat{F}^{-1}(Y, 494) & 0.6006\ 1.8676\ 3.7060\ 3.9247 \\ re(i, Y, 494) & 1.26\quad 0.08\quad 0.01\quad 0.01. \end{array}$$

We can see that the first component of the relative residual vector is not so very small and even larger than in the above example. It has been shown by numerical experiments that the accuracy is not improved as m decreases. It seems likely that the nonlinearities within the system are so large that increased approximation errors

are no longer counterbalanced by the reduced condition number. Now we reduce the a priori parallelepiped by placing in its center the estimate vector we just found and setting its edges equal to 2. In this case, we get

$$-1 + 0.6 \le \theta_1 \le 1 + 0.6, -1 + 1.86 \le \theta_2 \le 1 + 1.86,$$

$$-1 + 3.7 \le \theta_3 \le 1 + 3.7, -1 + 3.9 \le \theta_4 \le 1 + 3.9.$$

Set $m = 209(d = 6)$, and then

$Cond(Q(209)) = 8.1687 \times 10^{21}$.

The algorithm has calculated the following values for the estimate vector and the relative residual vector components:

$$\begin{array}{ll} \hat{F}^{-1}(Y, 209)_i & 1.0087\ 1.9861\ 2.9647\ 4.0039 \\ re(i, Y, 209) & 0.03\ \ 0.01\ \ 0.0006\ 0.0014. \end{array}$$

To further reduce the estimation errors, we further decrease the a priori parallelepiped, where the roots exist by placing in its center the estimate vector we just found and setting its edges equal to 0.2:

$$-0.1 + 1.009 \le \theta_1 \le 0.1 + 1.009, -0.1 + 1.986 \le \theta_2 \le 0.1 + 1.986,$$

$$-0.1 + 2.965 \le \theta_3 leq 0.1 + 2.965, -0.1 + 4.004 \le \theta_4 \le 0.1 + 4.004.$$

Set $m = 69(d = 4)$, and then $Cond(Q(69)) = 3.3329 \times 10^{22}$. In this case, the algorithm has calculated the following values for the estimate vector and the relative residual vector components:

$$\begin{array}{ll} \hat{F}^{-1}(y, 69)_i & 0.9998\ \ \ 1,9977\ \ \ 2.9989\ \ \ 4.0000 \\ re(i, Y, 69) & 1.57 \times 10^{-6}\ 2.23 \times 10^{-3}\ 9.54 \times 10^{-8}\ 4.85 \times 10^{-8}. \end{array}$$

It is seen that the relative residuals became reasonably small. Thus, the algorithm in fact succeeded in finding the almost exact solution of the system with "large-scale" nonlinearities, including nondifferentiable functions with a modulus. Only three iterations are needed.

4.9 Solving a Two-Point Boundary Value Problem for a System of Nonlinear Differential Equations

Applying the polynomial approximation technique to this problem is not difficult; it only requires defining the component values for the vector-function $F(\theta)$ at a number of points within the domain Ω_θ. The vector component values are needed to

calculate n-variate integrals. The vector-function thus can be an implicit expression, for instance, one resulting from the output of some computational procedure. This is a commonplace occurrence when considering some boundary value problems provided that there is a known ordinary differential system along with the method of solving it. Then the components of the vector $F(\theta)$ are calculated by numerical integration of the system at several grid points provided that the grid covers a parallelepiped of the domain Ω_θ. Consider a $2n$-order system:

$$du/dt = f(u, t), \tag{9.1}$$

for which we know the first n components of the initial condition vector: $u_1(0), \ldots, u_n(0)$, and the variables $y_1, \ldots, y_q$, are predetermined and equal to $u_1(T), \ldots, u_q(T)$, respectively, with the moment T fixed. The unknowns are variables $\theta_1, \ldots, \theta_n$, which are initial conditions $u_{q+1}(0), \ldots, u_{2q}(0)$, respectively.

The two-point boundary value problem just formulated is an equivalent of problem (2.1) provided that the vector-function $F(\theta)$ is determined by numerical integration of system (9.1).

If $f(u, t)$ is a linear function of u, then $F(\theta)$ is also a linear function of the vector θ; the two-point boundary value problem is an exactly solvable problem in this case:

$$\hat{F}^{-1}(Y_N, q) = \theta.$$

So we can state that, with the polynomial approximation technique applied, the two-point boundary value problem becomes a trivial problem in terms of computation—for a dynamic system model with linear differential equations.

Of specific interest is solving a two-point boundary value problem for the system of nonlinear differential equations.

An example is the two-point boundary value problem with nonlinear right-hand side (9.1) as follows:

$$f(u, t) = Au - 0.001 I_{2n}.u^3,$$

where: I is a matrix with entries that are coefficients of $(2q)^2$ random number samplings distributed uniformly over an interval $[-0.5, 0.5]$, I_{2q} is a $2q \times 2q$ unit diagonal matrix, and u^3 is a definition of the $2q$-variate vector with components that are third powers of the u components.

Next, set $q = 3$, $T = 3$s. System (9.1) has been integrated using the direct Euler method with a step of 0.1 s:

$$u_1(0) = 0, u_2(0) = 0, u_3(0) = 0, u_1(3) = 1, u_2(3) = -1, u_3(3) = 1.$$

The a priori domain is a cube in R^3 centered at the origin of R^3 and its edge equal to $20 : -10 \le \theta_i \le 10, i = 1, 2, 3$.

Let $m == 119 (d = 7)$. In this case, the algorithm has calculated the following values for the θ estimate vector components and the relative residual vector

components for the solution of the two-point boundary value problem:

$$\begin{array}{ll} \hat{F}^{-1}(Y_N, 119)_i & -1.5885 \quad -0.07617 \quad -0.9467 \\ re(i, Y, 119) & 1.35 \times 10^{-2}\ 7.81 \times 10^{-3}\ 4.80 \times 10^{-2}. \end{array}$$

The accuracy (in terms of the relative residual vector components) is not improved as m increases, due to computational errors. So we apply an iteration procedure; to do so, decrease the a priori parallelepiped where the roots exist by placing in its center the estimate vector just found and setting its edges equal to 1:

$$-1 - 1.5885 \le \theta_1 \le 1 - 1.5885, -1 - 0.07617 \le \theta_2 \le 1 - 0.07617,$$
$$-1 - 0.9467 \le \theta^3 \le 1 - 0.9467.$$

Next, we again set $m = 119(d = 7)$ and get

$$\begin{array}{ll} \hat{F}^{-1}(Y, 119)_i & -1.6210 \quad -0.09908 \quad -0.9962 \\ re(i, Y, 119) & 5.49 \times 10^{-8}\ 1.65 \times 10^{-7}\ 1.18 \times 10^{-7}. \end{array}$$

The resulting small values of the relative residual vector components enable us to state that the two-point boundary value problem has been solved almost exactly; the single iteration was adequate to succeed. Note that 119 vectors $\lambda(_1, \ldots, _n)$ of series (2.2) found in the first iteration step and stored in the computer memory make it possible to determine the first approximation to the solution of the boundary value problem for any vector Y_N with no need to solve equations like (2.10).

4.10 The System of Algebraic Equations with Complex-Valued Roots

In Sect. 4.7, we considered a problem of applying the polynomial approximation technique to calculate the real-valued roots of the simultaneous algebraic equations where the components of the vector-function $F(\theta)$ in (2.1) are polynomials with respect to components of the vector θ. Sometimes, however, it is necessary to find all (both real-valued and complex-valued) roots of Eq. (2.1).

Let θ^k be a component of the vector θ, $1, \ldots, k, \ldots, q$:

$$\theta^k = \theta_1^k exp(i\theta_2^k), i = (-1)^{1/2},$$

$$exp(i\theta_2^k) = cos(\theta_2^k) + isin(\theta_2^k),$$

and further consider all the components of the vector Y_N as real-valued variables.

So we find that Eq. (2.1) resolves itself into two equations of the form

$$F_1(\theta) = Y_N,\ F_2(\theta) = 0,$$

where θ is a $2q \times 1$ vector with its components expressed by $\theta_1^k, \theta_2^k, 1, \ldots, k, \ldots, q$.

The a priori parallelepiped $\Omega_{\theta,1}$, corresponding to the parameters θ_1^k, is determined by inequalities as follows:

$$0 \le \theta_1^1, \ldots, \theta_1^q \le \alpha,$$

where the α value is selected a priori.

The a priori parallelepiped $\Omega_{\theta,2}$ corresponding to the parameters $\theta_2^{(k)}$ is determined by inequalities as follows:

$$0 \le \theta_2^1, \ldots, \theta_2^q \le 2\pi.$$

Similarly to examples discussed above, we now will search approximations to roots applying the polynomial approximation technique upon breaking down the a priori parallelepipeds into a number of smaller parallelepipeds with edges obtained by dividing the edges of the a priori parallelepipeds into some integer parts.

To isolate real-valued roots, we may now define the a priori domains for the variables $\theta_{k,2}$ (all or in part) using inequalities as follows:

$$-\varepsilon \le \theta_2^k \le \varepsilon \tag{10.2}$$

or

$$-\varepsilon + \pi \le \theta_2^k \le \varepsilon + \pi, \tag{10.3}$$

where ε is a small positive number. Next, we give an example of the root calculation procedure.

Example.

$$\theta^5 + 3\theta^4 - \theta^3 + 2\theta^2 = 3\theta = 1 \tag{10.4}$$

From here we get

$$\theta_1^5 cos(5\theta_2) + 3\theta_1^4 cos(4\theta_2) - \theta_1^3 cos(3\theta_2)$$

$$+ 2\theta_1^2 cos(2\theta_2) + 3\theta_1 cos(\theta_2) = 1, \tag{10.5}$$

$$\theta_1^5 sin(5\theta_2) + 3\theta_1^4 sin(4\theta_2) - \theta_1^3 sin(3\theta_2)$$

$$+ 2\theta_1^2 sin(2\theta) + 3\theta_1 sin(\theta_2) = 0. \tag{10.6}$$

Let $\alpha = 5$, and set a manifold of a priori intervals $\Omega_{\theta_1}^j$, $1 \le j \le 50$; here we define the intervals by dividing the interval [0, 5] into 50 congruent segments.

First, we find all real-valued roots of Eq. (10.2). We apply the polynomial approximation technique in a sequence for the intervals $\Omega_{\theta_1}^j$ and then for the intervals (10.2)

and (10.3) with $\varepsilon = 0.0001$. Setting $d = 5$ enabled us to solve Eqs. (10.5) and (10.6) with an error of about 10^{-8} and 10^{-15}, respectively. The resulting three real-valued roots are as follows:

$$\begin{array}{ll} \theta_1: & 0.281193, \quad 0.896830 \ 3.385003 \\ \theta_2: & -2.622590 \times 10^{-17} \ 3.141592 \ 3.141592. \end{array}$$

To determine one of two conjugate complex-valued roots, it is sufficient to set $\Omega_{\theta,2} : 0 \le \theta_2 \le \pi$. Next we bisect the interval $\Omega_{\theta,2}$ into congruent segments, and further apply polynomial approximation over these segments and the intervals $\Omega^j_{\theta_1}$. Thus, we have found a complex-valued root:

$$\theta_1 = 1.082337, \theta_2 = 1.090255.$$

References

1. Karmanov VG (1980) Mathematical programming. Physmatlit, Moscow
2. Polak E (1974) Numerical methods of optimization. Universal Approach, World
3. Boguslavskiy JA (1994) Recurrent algorithm of optimum estimation. Pap Russ Acad Sci 4
4. Boguslavskiy JA (1996) Bayesian estimations of nonlinear regression and related questions. Bulletin of the Russian Academy of Sciences. Theory and Control Systems, No. 4
5. Boguslavskiy JA (2001) Bayesian method of numerical solution of nonlinear equations. Bulletin of the Russian Academy of Sciences. Theory and Control Systems, No. 2
6. Boguslavskiy JA (2005) Integrated method of numerical decision of the algebraic equations II. Appl Math Comput 166(2):324–338
7. Tikhonov AN et al. (1990) Numerical methods for solving incorrect problems. Science
8. Horn RA, Johnson CR (1989) Matrix analysis. Cambridge University Press, Cambridge

Chapter 5
Identification of Parameters of Nonlinear Dynamic Systems; Smoothing, Filtration, Forecasting of State Vectors

5.1 Problem Statement

The development of algorithms to solve the parameter identification problems with nonlinear dynamic systems is very important when one considers numerous fundamental and applied problems. Such algorithms are imperative, for example, in the following instances:

- One has created a plausible mathematical model with unknown parameters for a real dynamic system under study; some of the variables are observable in noise and depend on some components of the current state vector; it is necessary to find the system parameters.
- One is creating an algorithm for adaptive control of a dynamic linear system; the algorithm should contain a computational procedure for estimating the unknown elements of a matrix of linear differential equations of a system model.
- One is developing a nonlinear filtration algorithm for an optimal in the root-mean-square-sense estimate of the current state vector of the nonlinear dynamic system; the algorithm should contain the computational procedure mentioned in the preceding instance.

Then one considers [1] a problem of estimating an unknown parameter vector θ for the mathematical model of the dynamic system of the form

$$\mathrm{d}x/\mathrm{d}t = f(x, \theta, t), \tag{1.1}$$

where $x \in R^m$ is a state vector system; $f(x, \theta, t)$ is a uniform vector-function of its arguments; $\theta \in \Omega_\theta \in R^q$ is a vector of unknown parameters of the dynamic system; and Ω_{thet} is an a priori region (defined in R^q) of the existence of the parameter vector θ.

If we assume that

$$x'^T = \| x^T \quad \theta^T \|^T, \; f'(x', t)^T = \| f(x, \theta, t)^T \quad 0_q^T \|^T,$$

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_5,

then we will find that the identification problem is a special case of a more general problem, as follows: For a dynamic system

$$\mathrm{d}x^{'}/\mathrm{d}t = f^{'}(x^{'}, t),$$

it is necessary to estimate an unknown vector of initial conditions $x(0)^{'}$. If one has found an estimation vector, then the known Cauchy problem will solve the nonlinear filtration problem; that is, it will estimate the current state vector $x(0)^{'}$ of a dynamic system.

Therefore, the technique of identification—outlined later in this chapter—should be regarded as a version of the solution of the nonlinear filtration problem.

The term "mathematical model of a dynamic system" includes, of course, not only formal notation (1.1), but also a method of numerical integration of differential equation (1.1), providing—for all points of region Ω_θ—some accuracy and stability of the computational process, sufficient to simulate a model of the dynamic system. It is expected that in all points of the a priori region, model (1.1) is stable.

As an input for the algorithm to estimate vector θ, one uses a scalar sequence of observation results, $y_1, \ldots, y_N$, where the numbers y_k are of the form

$$y_k = H(k, x_k). \tag{1.2}$$

Here $x_k = x(t_k)$, $y_k = y(t_k)$, $t_1, \ldots, t_k, \ldots, t_N$, is a sequence of observation instants. Next, we assume that variables $y_1, \ldots, y_N$ are components of the vector Y_N.

As an output for the algorithm, one uses a vector to estimate a parameter vector—the vector-function $\hat{\theta}(Y_N) \in R^q$, delivering a vanishing variable (a criterion for evaluation quality)—a positive number $J(\ldots)$, by which a sum of squares of residuals serves

$$J(Y_N, \hat{\theta}) = \sum_{k=1}^{N} (y_k - H(k, x_k(\hat{\theta}(Y_N)))^2, \tag{1.3}$$

where the notation $x_k(\hat{\theta}(Y_N))$ means that on the right side of Eq. (1.1), we assume $\theta = \hat{\theta}(Y_N)$.

A criterion for estimation quality of the form (1.3) is based on

$$\arg \min_{\theta \in \Omega_\theta} J(y_N, \vartheta) = \theta,$$

$$\min_{\vartheta \in \Omega_\theta} J(Y_N, \vartheta) = 0,$$

where $\vartheta \in \Omega_\theta$, and function $J(Y_N, \vartheta(Y, \vartheta))$ is continuous with respect to $\vartheta \in \Omega_\theta$. In the present-day literature (see, for example, [2]), as a universal method of solving the problem of minimizing $J(Y_N, \hat{\theta})$ by choosing a vector-function $\hat{\theta}(Y_N)$

(this problem is commonly called a problem of the nonlinear least-squares method), one regards the development of various versions of the gradient method. However, in the situation being considered, the creation and use of such a method can cause difficulties, including the following:

1. The computation at instants $1, \ldots, k, \ldots, N$ of gradient vectors of the function $J(Y_N, \hat{\theta})$ will require the computation of matrices of private derivatives of the current state vectors with respect to components of the parameter vector that is not implementable with the nondifferentiable—with respect to x—vector-function $f(x, \theta, t)$ in Eq. (1.1).

2. At differentiable $f(x, \theta, t)$ with respect to x, the computation—versus time—of elements of the matrices of private derivatives (see item 1) will require additional numerical integration of the system of differential equations of dimension $n \times m$. This integration should provide very small gradient vectors with a high computational accuracy, which is necessary if one needs to estimate the vector of unknown parameters to a high accuracy.

3. The definition of a global minimum by the gradient method runs into complications if there are local minima of the minimized functions with complex relief ("rifts," "plateaus," etc.). In the neighborhood of a local minimum, the movement about points of the region Ω_θ in the direction of a decreasing function $J(Y_N, \vartheta)$ can correspond to an increase in components of the estimation error vector. The example we consider here shows that the particular identification problem under study has a function—to be minimized—that is precisely of this kind.

Then one considers the nongradient algorithms, which are based only on computation of the values of the minimized functions $J(Y_N, \vartheta)$ or on defining its variables $H(k, x_k(\vartheta))$ on different points of the a priori region Ω_θ, assigned according to some rational principles. Such algorithms are further conditionally called "organized search" algorithms because, in the process of their functioning, one "searches through" various points of the a priori region. The family of similar algorithms also includes a "simple search" algorithm: It uniformly covers the region Ω_θ with a great number of points and computes a set of variables $J(Y_N, \vartheta)$; the estimate vector $\hat{\theta}(Y_N)$ is taken to be equal to the vector for which one of these values is minimum. Alternatives to simple search include an algorithm using the MATLAB® function f_{mins} as well as a polynomial approximation algorithm.

We note that the identification problem, which is reducible to the search for a global minimum of a multivariate function that is only defined implicitly—as a result of the numerical integration of a system of nonlinear differential equations—is a complicated problem of computational mathematics. It partially justifies the heuristic character of the algorithms that appear in Sect. 5.2. This heuristic character is inherent to many numerical methods of solving complex nonlinear problems. Hence, the criteria for proving that computational processes provide the necessary results are not strict, and confidence in their working capacity is only based on positive results from numerically solving a number of actual identification problems.

5.2 Heuristic Schemes of a Simple Search and an Organized Search

Then, for definiteness, we assume that n, a priori region Ω_θ is a cube in R^n whose edge lengths, being parallel to the axes of coordinates, are equal to the unit, and the center coincides with the origin of the coordinates.

In step 1 of the organized search, the a priori cube edges are divided into r equal parts whose ends serve as vertices of r^n cubes * with edge lengths of $1/r$.

Every cube * has an operator computed from it: a vector $g(1,i) \in R^q, i = 1,\ldots,r^q$, and a function $J(Y, g(1,i))$. The vector $g(1,i)$ is a center of symmetry of cube *. In a simple search for the vector, $g((1,i)$ coincides with one of the vertices of cube *. In an organized search for the vector, $g(1,i)$ belongs to an interior point of cube *, which is further selected using a polynomial algorithm for multiapproximations or the MATLAB function f_{mins}.

Then, from r^n vectors $g(1,i)$, the vector $g(1)$, for which this function $J(y, g(1))$ is minimum, is selected; the vector $g(1)$ is the center of symmetry of the region $\Omega_\theta(1)$ of the cube with edges of preset length $a(1) < 1$. The region $\Omega_\theta(1)$ is an a priori region for step 2 of an organized search.

Additional steps are built similarly. After dividing edges of region $\Omega_\theta(k-1)$ into r identical parts, building r^4 subregions (cubes *), and selecting those on which the variable $J(Y, g(k))$ is minimum, we assign vector $g(k)$ as the center of symmetry of subregion $\Omega_\theta(k)$, whose edge lengths are equal to $a(k) < a(k-1)$. The computational process stops at step K, when the variable $J(Y_N, g(K))$ becomes less than the preset border δ. The latter is defined by a series of predesigns and ensures reaching a desired accuracy of estimating the vector of unknown parameters.

At a given structure of operator $g(k,i)$, positive numbers $a(1), a(2), \ldots, a(k), \cdots (1 > a(1) > a(2) > \cdots > a(k) > \cdots)$ are selected after a series of computational experiments. The following heuristic conditions should be fulfilled:

1. The sequence of regions

$$|\Omega_\theta| > |\Omega_\theta(1)| > |\Omega_\theta(2)| > \cdots, > |\Omega_\theta(k)| > \cdots,$$

with decreasing edge lengths $l, a(1), a(2), \ldots, a(k), \ldots$ of generating regions $\Omega_\theta(1), \Omega_\theta(2), \ldots, \Omega_\theta(k), \ldots$, contains the vector θ.

2. Upon increasing k : $J(Y_N, g(k)) \to 0$, $g(k) \to \theta$.

If the values $a(1), \ldots, a(k), \ldots$ and the integer r as selected are too small, then beginning from some step of the computational process variable $J(Y_N, g(k))$ will stop decreasing. Then one should increase the values $a(1), \ldots, a(k), \ldots$ and integer r.

The a priori region Ω_θ should be assigned some "margin" so that the real vector will not be close to its border points. Then the regions $\Omega_\theta(1), \Omega_\theta(2), \ldots., \Omega_\theta(k), \ldots$ will remain cubes. In this case, the closeness to 1 of the numbers $a(1), \ldots, a(k), \ldots$, the corresponding increase in the number of steps, and the uniformity of $J(Y_N, \vartheta)$ ensure the validity of conditions (1) and (2).

Noncompliance with condition (1) will lead to noncompliance with the first of conditions (2), which has been established in the computational process. This fact is a signal to organize a new computational process with increased values of the integer r and of edge lengths $a(1), \ldots, a(k), \ldots$.

5.3 Mathematical Model to Test Algorithms

Comprehensive research and comparison of algorithms are impossible when functions in Eqs. (1.1) and (1.2) have a general form. Hence, in what follows, the algorithms to solve the identification problem are analyzed for an explicit nonlinear dynamic system at $q = 4, m = 2$ and at not differentiable right part of one of the differential equations.

Let x_1 and x_2 be the current coordinates and its derivative with respect to time of a material point that moves in a horizontal plane under the action of (1) an elastic force—described by a nonlinear (linear-cubic) Hooke's law—as well as (2) a velocity damping force and (3) a dry friction force. In the parametric formulas to represent the forces, these parameter values are unknown. We will suppose that the model of the system being considered is of the form

$$\mathrm{d}x_1/\mathrm{d}t = x_2,$$

$$\mathrm{d}x_2/\mathrm{d}t = (1+\theta_1) - 0.2(1+\theta_2) - 0.1(1+\theta_3)\,\mathrm{sign}\,(x_2) - 0.1(1+\theta_4)x_1^3,$$

where the unknown parameters $\theta_1, \theta_2, \theta_3, \theta_4$ define a measure of ignorance of the exact coefficients of acceleration from, respectively: the linear component of Hooke's law, the velocity damping force, the dry friction force, and the cubic component of Hooke's law.

An a priori region Ω_θ is a cube in R^4 with a center at the origin of coordinates and with edges of unit length, which are parallel to the coordinates axes: $-0.5 <= i <= 0.5, i = 1, \ldots, 4$. At instants $1, \ldots, k, \ldots, N$, one observes the object's coordinates x_1, $y(k) = x_1(k)$. We will suppose that $x_1(0) = 1, x_2(0) = 0$ and that in the process of oscillatory damping of the object, there are $N = 200$ observations of value x_1, with an interval of 0.1 s.

Numerical integration was performed by a fourth-order Runge–Kutta method with a constant step of 0.02 s. Here are the actual data on estimation accuracy and on computational time consumption are given for the factual values of the estimated parameters:

$$\theta_1 = 0.32, \quad \theta_2 = -0.22, \quad \theta_3 = -0.018, \quad \theta_4 = -0.35.$$

The existence of a local minimum of function $J(Y_N, \vartheta)$ is illustrated with *Data 1 and Data 2*, where S are values of the function at some vectors ϑ, and four figures under J set the corresponding relative estimation errors if, as an estimate, one regards variables ϑ_i :

$$\delta_i = (\vartheta_i - \theta_i)/\theta_i, i = 1, \ldots, 4.$$

Data 1. $J(\ldots) = 0.0020715$:

$$\delta_1 = 0.0117; \quad \delta_2 = 0.2503; \quad \delta_3 = -2.6027; \quad \delta_4 = -0.0141.$$

Data 2. $J(\ldots) = 0.0016057$:

$$\delta_1 = 0.0148; \quad \delta_2 = 0.2689; \quad \delta_3 = -2.6564; \quad \delta_4 = 0.1979.$$

From the data, it follows that in order to reduce $J(\ldots)$ (passing from data 1 to data 2), one should increase the relative estimation errors; one should move from a point of the global minimum ($\vartheta = \theta$) to a point of a local minimum, increasing all components of the vector $\vartheta - \theta$.

The general criteria for the quality of a search algorithm are as follows: the lowest value S reached at computing function $J(Y_N, \vartheta))$ at V points of region Ω_θ, and the time to be spent for these computations.

Further, these data are consistent with the calculations on Computer low productivity, which intentionally increase several times the actual value calculation time T.

Let one use, for example, the method of simple search; the point coordinates were obtained by dividing the edges of unit cube Ω_θ into 24 equal parts: $V = 244$. Then the lowest value, reached by the simple search method at $V = 244$, is equal to $J(\ldots) = 0.0017874$ at the relative estimation errors, as follows:

$$\delta_1 = 0.0786; \quad \delta_2 = -0.1851; \quad \delta_3 = 2.3333; \quad \delta_4 = 1.5142.$$

The relative estimation errors were large despite the considerable times of computation. In order to reach small (of the order of 0.001) relative estimation errors, the value $J(\ldots)$ should be of the order of 10^{-7}, which—for the direct search method—corresponds to a computation time that cannot be implemented on a low-performance PC.

In addition to the model just presented, we analyzed a nonlinear problem of identifying parameters of an oscillatory link. The parameters were the time constant and decrement of damping, as well as the initial conditions of a dynamic system: an initial coordinate and initial velocity.

In this problem, we estimated four parameters, $\theta_1, \ldots, \theta_4$, defining the motion equation of the oscillatory link:

$$\mathrm{d}x_1/\mathrm{d}t = x_2,$$

$$\mathrm{d}x_2/\mathrm{d}t = (1 + \theta_1)(1/T^2)x_1 - 2(1 + \theta_2)(\xi/T)x_2,$$

$$1 + \theta_3 = x_1(0), \quad 1 + \theta_4 = x_2(0),$$

where $T = 1$ s and $\xi = 0.1$; this rates the values of the time constant and of the damping decrement of the oscillatory link.

The presented problem is—in line with the identification problem—a nonlinear filtration problem, because estimating the system parameters and the initial conditions makes it possible to define a current state vector of a dynamic system for any instant.

The presented data justify organized search methods as an alternative to the direct search method.

5.4 Organized Search with the MATLAB Function f_{mins}

The function $f_{\text{mins}}(f(\vartheta, X))$ is intended to define point $z(X)$ of a local minimum, set by the uniform function $f(\vartheta)$ of several variables in the neighborhood of vector X. The function f_{mins} returns vector $z(X)$ of the local minimum. Additionally, it returns the variable $f(z(X))$ and the number of iterations involved in the computation. Each iteration means computing the minimized function in at least one point of the neighborhood of vector X. We will emphasize that the f_{mins} algorithm uses only values of the minimized functions, computed in points of the neighborhood of X according to the principle outlined in the MATLAB HELP manual and in [3]. One should notice that the accuracy of computing a local minimum is defined by some constants of the f_{mins} algorithm. The results given below correspond to the constants of the standard MATLAB package. Let

$$f(\vartheta) = J(Y_N, \vartheta)$$

and X be a vector of the symmetry center of a region, obtained as a result of region partitioning described above. Then the above-mentioned operator from this region will be assigned a vector of local minimum $z(X, Y_N)$, returned by the function $f_{\text{mins}}(J(Y, \vartheta), X)$. We suppose that the minimized function $J(Y_N, \vartheta)$ in the a priori cube has a finite number of local minima, which all differ from each other, and that the accuracy of f_{mins} is sufficient for the following statement: A given local minimum is not equal to all those found earlier.

The conceptual scheme of the computation is similar to the general one outlined above. If

$$\delta < J(Y_N, g(k)) < J(Y_N, g(k-1)),$$

then the computation process is at step $k + 1$, similar to that described above. However, let

$$J(Y_N, g(k)) > J(Y_N, g(k-1)) > \delta.$$

Then step k is repeated, but already at the partitioning of edges of region $\Omega_\theta(k-1)$ into a number r_1 of equal parts, where $r_1 > r$. From the uniformity of $J(Y_N, \vartheta)$, it follows that there exists r_1 for which

$$\delta < J(Y_N, g(k)) < J(Y_N, g(k-1)).$$

So, one can build a computational process for which values of the local minima $J(Y_N, g(i))$, found—with some small errors—by the f_{mins} function, form a decreasing sequence of positive numbers:

$$J(Y_N, g(1)) > J(Y_N, g(2)) > \cdots > J(Y_N, g(k)) > \cdots$$

Due to having a finite number of local minima, this sequence will necessarily find, at some $k = K$, a global minimum for which

$$J(Y_N, g(K)) < \delta.$$

We suppose that the estimate vector is equal to the vector $g(K)$. As such, it is assumed, of course, that the values $a(1), \ldots, a(k), \ldots$ have been selected such that condition (1) is satisfied (see Sect. *5.2).

Let us build an explicit computational process for the organized search with function f_{mins}. Let $r = 2$; in step 1, the a priori cube is divided into 24 smaller cubes *; we define 24 vectors $z(X_i, Y), i = 1, \ldots, 24$, where the X_i are vectors of these cubes' centers. The computation has pointed to the fact that among 24 numbers $J(Y_N, z(X_i, Y))$, there exist two pairs of about identical numbers, and the remaining numbers significantly differ from each other. All the numbers lie within the range 2.104×10^{-6}–1.825; four components of the vector $g(1)$—for which $J(Y_N, d(1)) = 2.104 \times 10^{-6}$—are as follows: 0.3192, −0.2182, −0.0195, −0.333.

The presented data point to the fact that function $J(Y, \vartheta)$ has at least 14 local minima. The computation time of each vector-function $z(X_i, Y)$ is of the order of 5 min, and the process takes 150 iterations. For this reason, the general computation time in step 1 is $T \sim 80$ min, with the total number of iterations of the order of 2,500.

Before step 2, we assume $a(1) = 0.1$. The computation has shown that 24 numbers $J(Y_N, z(X_i, Y)$, obtained in step 2, can be divided into six groups approximately composing six local minima. Groups 1–4 are composed accordingly from 3, 4, 4, 3 nearly identical numbers for which the values $J(Y, z(X_i, Y))$ lie within the limits 0.0002–0.0005.

The isolated quantities 1.3707×10^{-5} and 4.314×10^{-7} compose groups 5 and 6, respectively; the four components of the vector $g(2)$, for which $J(Y, g(2)) = 4.314 \times 10^{-7}$, are as follows: 0.31980, −0.21974, −0.01834, −0.34697.

The time of computation and the number of iterations in step 2 are about the same as those in step 1. The value $J(Y, g(2))$ is of the order of 10^{-7}. Thus, $g(2)$ can be viewed as an estimate of the vector of unknown parameters: $z(Y_N) = g(2)$. Under

these conditions, the relative estimation errors are as follows:

$$\delta_1 = -0.00062; \quad \delta_2 = 0.0012; \quad \delta_3 = -0.019; \quad \delta_4 = 0.0087.$$

So, after two steps of the organized search process, using the function f_{mins}, we see that the general time of computation is $T \sim 160$ min, the total number of iterations (the number V of operations of computing a minimized function) $J(Y_N, \vartheta)$ is about 5,000, and the relative estimation errors do not exceed two.

5.5 System of Implicit Algebraic Equations

The whole preceding statement was based on an idea of the adequacy of the problems of identifying and defining $\arg\min_{\vartheta \in \Omega_\theta} J(Y_N, \vartheta)$, at whose solution the existence of a set of local minima of the function $J(Y_N, \vartheta)$ caused computational difficulties. The Bayes interpretation (outlined in Chap. 2) has no such disadvantage, as it implies "directly" considering the problem by reducing it to analyzing the problem of defining a q-dimensional root vector of the system of N algebraic equations of the form (2.1) of Chap. 4. The left part of this system—the vector-function $F(\theta)$—is set implicitly, and its components are the N functions $H(k, x_k(\theta)), k = 1, \ldots, N$, where the vectors $x_k(\theta)$ are the results of the numerical solution of the differential equation (1.1) on the discrete-time segments $[0, k]$ at preset vectors $x(0), \theta$. A system of implicit algebraic equations is of the form

$$H(k, x_k(\theta)) = y_k, \quad k = 1, \ldots, N.$$

Equations (2.5)–(2.8) of Chap. 4 imply that integrands of integrals for components of the a priori vectors $E(\theta)$, $E(W(Y))$ and for elements of the matrices L, Q can be found by numerical integration of Eq. (1.1) at a given vector $x(0)$ and parameter vectors ϑ, set by a grid of nodes into which the a priori region Ω_θ is divided. As a result, for the root vector of system (5.1), an estimation vector $\hat{F}^{-1}(Y_N, d)$, optimal in the root-mean-square sense, will be built. The estimation vector is implemented by a linear combination of integer powers from components of the vector of observations Y_N, delivered by a method of polynomial approximation. It is clear that the described sequence of computations contains no conceptual difficulties in attempting to find a global minimum of the criterion of quality (1.3).

However, in applied problems, the dimensionality N of the vector of observations is great; the value N can exceed the value of $m(N, d)$, a dimensionality of the vector $W(Y_N)$, by tens and hundreds. Because of this, some difficulties may occur in computing elements of the matrix Q^{-1} of dimensionality $m(N, d) \times m(N, d)$, necessary to obtain elements of the optimal weight matrix Λ.

These difficulties can be avoided if, for computation of the matrix Q^{-1}, one uses (see Chap. 1) a sequential computational process based on decomposition of the observation vector Y.

5.6 Contraction Operator

The other way to avoid computational difficulties consists of using a contraction operator acting on an observation vector. In this case, the entry of the identification algorithm is not by the N-dimensional vectors $F(\theta)$, Y_N, but by the n-dimensional vectors $F^c(\theta)$, Y_N^c, obtained by applying a **comp operator** to the vectors $F(\theta)$, Y, reducing the dimensionality:

$$F^c(\theta) = \text{comp}(F(\theta)), \quad Y_N^c = \text{comp}(Y_N).$$

Then the parameter vector θ satisfies an algebraic equation such as Eq. (2.1) of Chap. 4:

$$F^c(\theta) = Y_N^c. \tag{6.1}$$

As at $\theta \in \Omega_\theta \in R^q$, $Y_N \in \Omega_{Y_N} \in R^N$, then $Y_N^c \in \Omega_{Y_N}$. Region $\Omega_{Y_N^c}$ is obtained by applying a comp operator to all vectors of region Ω_{Y_N}:

$$\Omega_{Y_N^c} = \text{comp}(\Omega_{Y_N}).$$

The contraction operator should provide the following properties:

1. For the vectors $\theta \in \Omega_\theta$ and Y, whose components are defined by Eqs. (1.1) and (1.2), Eq. (6.1) should be compatible and have a unique root; according to the Bayes interpretation, problem (1.1) can be replaced by the problem of defining a vector of conditional expectation. Because of peculiarities (Chap. 4) of the joint probability density of random vectors θ, Y_N^c, this vector is coincident with the desired parameter vector:

$$E(\theta | Y_N^c) = \theta.$$

2. The vector-function $E(\theta | Y_N^c)$ is a uniform function of Y_N^c.
3. The region $\Omega_{Y_N^c}$ is a compact set.

Next, one considers a situation when the contraction operator is linear and is presented by the matrix G of dimensionality $G_{q \times N}$. Then

$$F^c(\theta) = GF(\theta), \quad Y_N^c = GY_N.$$

The contraction operator reduces the information on parameter θ, used at its estimation. For this reason, the variance of estimation errors with the vector-function $F^c(\theta)$ and with the vector Y^c is at least not less than that of estimate reached in the absence of the contraction operator. This qualitative statement corresponds to a matrix inequality

$$LG^T(GQG^T)^{-1}GL^T < LQ^{-1}L^T. \tag{6.2}$$

The difference in diagonal elements of the right and left matrices in Eq. (6.2) can be used to evaluate the quality of the selected matrix G.

We consider possible versions of the matrix G:

1. *Smoothing contraction operator*. Without loss of generality, we assume that an integer N is proportional to n. The components $g(k, i), k = 1, \ldots, n, i = 1, \ldots, N$, of matrix G are defined by

$$g(k, i) = 1/n, \quad \text{if}((k-1)N/N) + 1 <= i <= kN/n;$$

for the rest of $i : g(k, i) = 0$.

The component of the vector Y_N^c is an arithmetic mean value of the components of the observation vector Y_N beginning from the instant $((k-1)N/q) + 1$ and through instant kN/q. This circumstance defines the smoothing properties of the linear operator, which reduce the influence of additive random errors of observations such as discrete white noise.

2. *Contraction operator for quasilinear sequence of observations.* Let the matrix G be such that at the linear dependence of Y_N on θ, replacing Y for Y_N^c leaves components of the error estimation vector equal to zero when the polynomial approximation algorithm is used.

We find an optimal—in the root-mean-square-sense—estimate $\hat{Y}(\theta, 1)$ of the vector Y by means of a vectorial linear combination of components of the vector $\theta(d = 1)$. From equations of Chap. 1, we obtain

$$\hat{Y_N}(\theta, 1) = E(Y_N) + L_1 Q_1^{-1}(\theta - E(\theta)),$$

where

$$L_1 = E((Y_N - E(Y_N))(\theta - E(\theta))^T,$$

$$Q_1 = E((\theta - E(\theta))(\theta - E(\theta))^T = C_0.$$

For each vector θ, the vector $\hat{Y}(\theta, 1)$ is the closest (in the root-mean-square sense) vector to one of the random N-dimensional vectors, for which the first and second a priori statistical moments are presented by the vector $E(Y_N)$ and by the matrices

$$L = ((|\theta - E(\theta - E(\theta))Y^T), \quad Q = E((Y_N - E((Y_N - E(Y_N))Y_N^T).$$

The vector Y_N^c is thought to be equal to the vector from the a priori region Ω_θ, for which the factual vector Y_N is the closest one to the vector $\hat{Y}(\theta, 1)$—in the sense of the least squares method. As a measure of closeness, we take a value

$$J(\theta) = (Y_N - \hat{Y}_N(\theta, 1))^T (Y_N - \hat{Y}_N(\theta, 1)).$$

Thus,

$$Y_N^c = \arg(\min_{\theta \in R^N}(\theta)).$$

Hence, we obtain

$$Y_N^c = E(\theta) = C_0(LL^T)^{-1}L(Y_N - E(Y_N)).$$

So,

$$E(Y_N^c) = E(\theta),$$

$$L^c = E(\theta - E(\theta))(Y_N^c - E(Y_N^c = C_0,$$

$$Q^c = ((Y_N^c - E(Y_N^c))(Y_N^c - E(Y_N^c))^T) = C_0(LL^T)^{-1}LQL^T(LL^T)^{-1},$$

and then

$$\Lambda^c = C^0 = LL^T(LQL^T)^{-1}LL^TC_0.$$

The matrix LQL^T has dimensionality $q \times q$. After we've used vector Y_N^c, the estimation error covariance matrix looks like

$$C^c = C_0 - L^c(Q^c)^{-1}(L^c)^T = C_0 - LL^TLQL^{T,-1}LL^T.$$

In the process of building itself, a matrix inequality is true:

$$LL^T(LQLT)^{-1}LL^T < LQ^{-1}L^T.$$

We will show that

$$C^c = 0 \quad by \quad Y_N = A\theta,$$

and, hence, at the linear (with respect to θ) sequence of observation results, the replacement of vector Y_N for vector Y_N^c of dimensionality q does not reduce the accuracy of the estimate by the polynomial approximation algorithm. The preceding equations imply that in this case

$$Y_N^c = (A^TA)^{-1}A_N^Y \quad \hat{\theta}(Y_N^c, 1) = \theta.$$

We notice that Y_N^c is coincident with the estimate of the least squares method when the contraction of the observation vector's dimensionality occurs via a transition to a system of normal equations.

3. Optimal linear contraction operator. We will try to define a structure of the optimal matrix G^o of the linear contraction operator, considering variances of estimate errors by the polynomial approximation algorithm at linear ($d = 1$) approximation

of the parameter vector θ by components of the vector Y^c. The matrix G^o of dimensionality $n \times N$, rank $G^o = q$, is optimal if the method's algorithm, using vector $Y_N^c = G^o Y_N$, delivers the estimation error variances of the vector θ, which are the closest to the estimation error variances, optimal in the root-mean-square sense, when employing the vector Y_N.

The matrix inequality (6.2) implies that for the matrix G^o, the diagonal elements of the matrix

$$P = G^{o,T}(G^o Q G^{o,T})^{-1} C^o$$

should reach a maximum. But for P, there is a true representation,

$$P = U/(\lambda_1, \ldots, \lambda_N) U^T,$$

where U is an orthogonal matrix, $\lambda_1, \lambda_2, \ldots, \lambda_N$, of nonnegative eigenvalues of the matrix P, not all of which are equal to zero. Therefore, G^o should reach a maximum value of $x^T P x$, where x is an arbitrary vector, normalized by equality $|P| = 1$. But then $g_{i,j}$ are elements of the matrix G^o, which should satisfy a necessary condition of extremum

$$\partial P / \partial g_{i,j} = 0. \tag{6.3}$$

Hence, we obtain

$$G^{oT}(G^o Q G^{oT})^{-1} I(i, j)(I_N - Q Q G^{oT}(G^o Q G^{oT})^{-1} G^o)$$

$$+ (G^{oT}(G^o Q G^{oT})^{-1} I(i, j)(I_N - Q Q G^{oT}(G^o Q G^{oT})^{-1} G^o))^T = 0,$$

where $I(i, j)$ is a matrix of dimensionality $n \times N$ for which the element, belonging to the ith row and jth column, is equal to unity; the remaining elements are equal to zero.

After matrix transformations, we obtain

$$Q G^{oT}(G^o Q G^{oT})^{-1} G^o = I_N. \tag{6.4}$$

If there exists a matrix G^o, satisfying Eq. (6.4), then at $d = 1$, the polynomial approximation algorithm delivers identical estimation error variances for the vectors Y^c and Y, and the matrix inequality (6.2) turns into an equality.

Let's represent Q in terms of blocks P, R, q, used in Chap. 1 and having dimensionality $n \times n, n \times (N - q), (N - q) \times (N - q)$, respectively. We will represent G^{oT} in terms of the blocks $G(q, q)$ and $G(N - q, q)$, having dimensionality $q \times q$ and $(N - q) \times q$, respectively; we denote this as

$$P = G^o Q G^{o,T^{-1}}.$$

The matrix $G(q, q)$ is thought to be nonsingular.

Then

$$(PG(q,q) + RG(N-q))PG(q,q)^T = I_q, \tag{6.5}$$

$$(R^T G(q,q) - qG(N-q,q)PG(q,q) = 0_{q,q}, \tag{6.6}$$

$$(PG(q,q) + RG(N-q))PG(N-q,q)^T = 0_{q,N-q}, \tag{6.7}$$

$$(R^T G(q,q) = qG(N-q,q))PG(N-q,q))PG(N-q,q)^T = I_{N-q,q}. \tag{6.8}$$

But by building itself, $q > 0$. Hence, Eqs. (6.6) and (6.7) imply the equalities

$$R^T G(q,q) + qG(N-q,q) = 0,$$
$$PG(q,q) + RG(N-q) = 0.$$

The validity of Eqs. (6.5), (6.8) and, therefore, of Eq. (6.4) is impossible. Thus, one cannot find a matrix G^o satisfying the necessary conditions of extremum (6.3).

5.7 Computational Scheme of Organized Search in Bayes Interpretation

In this section, we outline an explicit computational scheme for the case when all the intermediate regions—obtained in the process of completing the partitioning sequence described in Sect. 5.2—are cubes. The scheme will also not change when some of the regions are parallelepipeds.

The computational scheme will correspond to the conceptual scheme (described in Sect. 5.2) if for cubes we assume (as these cubes serve region Ω_{theta} or its parts) that as operators $g(1,i), \ldots, g(k,i), \ldots$ over them are vectors $\hat{F}^{-1}(Y_N^c, d)$.

The elements of the a priori vectors $E(\theta)$, $E(W)$ and of matrices C^o, L, Q, which are necessary to build $\hat{F}^{-1}(Y_N^c, d)$, are defined by the computation of multidimensional integrals on a sequence of cubes described in Sect. 5.2. For this purpose, one should compute the values $H(k, x(k, \vartheta))$, which enter expressions of these integrals' integrands, in some points of the integration regions. With an increasing number of points, the time of computation increases as the nth power of a number, which is greater than 1. For this reason, we use a minimum number of points here.

We assume that—according to the Bayes interpretation—components of an identified vector are uniformly distributed on the segments $[-0.5, 0.5]$, and, hence, their initial variance is equal to 1/12. Let condition (1) from Sect. 5.2 be valid; at some step of the computational process, after the corresponding partitioning of the edges of intermediate cubes (or of parallelepipeds), a sought random parameter vector θ occurred in a cube (belonging to the given partition) whose edge is equal to $1/s$, where s increases as the number of steps in the computational process increases. Because of the uniform a priori distribution of this vector, the a priori matrix of covariances

C_0 is diagonal with elements (a priori variances of the vector components) equal to $1/(12s)^2$. Hence, at least for this cube, the estimation error variances of components of the vector θ by the vector $\hat{F}^{-1}(Y_N, d)$ will not be greater than $1/(12\,s)^2$.

So, performing step (1), completing a sufficient number of steps of the computational process, and having a sufficiently great integer r, we should see that defining the number of points of partitioning of cube edges will in principle enable the existence of at least one cube (for example, the cube to which the vector θ belongs) for which vector $g(K)$, equal to the corresponding vector $\hat{F}^{-1}(Y_N, d))$, will lead to the inequality

$$J(Y_N, g(K)) < \delta.$$

The above stated proves the validity of condition (2) from Sect. 5.2 only if the partitioning sequence satisfies condition (1). We notice that if components of the vector W are linear functions of components of the vector θ, then all vectors $\hat{F^{-1}}(Y_N, d)$ will coincide with θ, and the inequality given above will be valid for all the cubes already at step 1 of the computational process.

We build an for the dynamic system shown in Sect. 5.3, for the explicit computational process of organized search with the use of the Bayes interpretation. We will assume that action of the contraction operator over the vector of primary observations, whose components have numbers from 1 to 200, consists of building a vector of dimensionality 4, for which the ith component ($i = 1, \ldots, 4$) is formed by the summation of 50 primary observations, beginning from number $50(i - 1) + 1$ and through number $50i$. We notice that the summation imparts to the contraction operator a property to smooth the influence of possible random errors of primary observations such as white noise. Then we will assume that $r = 5$ and we'll conduct six steps of the computational process described above. We present the used values $a(1), \ldots, a(5)$ and vector $g(k)$, which is achievable at step k, the value of the sum of squares of residuals $J(Y_N, g(k))$, as well as del_i, $i = 1, \ldots, 4$—components of the relative estimation error vector. In all steps we assume $d = 1$. It means that for each cube, obtained upon ordinary partitioning, the conditional expectation vector of the estimated vector of unknown parameters is approximated by a linear vector-function of four components of the vector, found by using a contraction operator—that is, by a vectorial linear combination of these components, optimal in the root-mean-square sense.

We notice that the decrease in the estimation accuracy, caused by a small value d, is compensated by a great value r, which has defined small edge lengths of cubes. Upon computing four-dimensional integrals by the method of trapezes, the method defined, for each cube, integrands in cube points whose coordinates coincided with all (or with some) coordinates of cube vertices as well as with all (or with some of) coordinates of points—the middle points of cube edges.

$step1$	$a(1) = 1$			
$g(1)$	0.3077	−0.2347	−0.0169	−0.0308
$J(Y, g(1)) = 2.0046 \times 10^{-3}$				
del_i	−0.0548	0.0629	−0.0609	−0.9119
$step2$	$a(1) = 0.25$			
$g(2)$	0.3225	−0.2117	−0.0254	−0.2361
$J(Y, g(2)) = 1.9447 \times 10^{-4}$				
δ_i	−0.009447	−0.04176	0.4144	−0.3253
$step3$	$a(2) = 0.125$			
$g(3)$	0.3245	−0.2141	−0.0238	−0.3318
$J(Y, g(3)) = 2.2488 \times 10^{-5}$				
δ_i	−0.003407	−0.03076	0.3223	−0.05204
$step4$	$a(3) = 0.05$			
$g(4)$	0.3258	−0.2233	−0.01588	−0.3522
$J(Y, g(1)) = 2.8909 \times 10^{-6}$				
δ_i	0.000697	0.0107	−0.1174	0.00626
$step5$	$a(4) = 0.025$			
$g(5)$	0.3257	−0.2205	−0.01813	−0.3529
$J(Y, g(5)) = 7.2759 \times 10^{-7}$				
δ_i	0.000382	−0.001518	0.007409	0.008380
$step6$	$a(5) = 0.0025$			
$g(6)$	0.3256	−0.2211	−0.01789	−0.3511
$J(Y, g(6)) = 1.2588 \times 10^{-7}$				
δ_i	0.000156	0.000782	−0.00599	0.003261

The time of computation at each step of the computational process was about 12 min.

So, after computation, which took about 72 min, the unknown parameters of the mathematical model were estimated with errors not exceeding 0.5 %.

In comparison with a version using MATLAB's f_{mins} function, the time of computation was reduced by a factor of 2.5, and the estimation errors were reduced by a factor of 4.

Above, we used the simplest—a linear ($d = 1$)—approximation of the conditional expectation vector. A more complex approximation permits us to reduce the value ηr and the general time of identification. We'll use a quadratic approximation ($d = 2$), we'll assume $r = 2$, and we'll obtain the required accuracy of computing four-dimensional integrals if integrands are computed in cube vertices and in points of edges partitioning into four equal parts (a modified method of trapezes permits us to compute in this way only once for each point). Then the required identification accuracy is reached for 22 iterations (each iteration takes 85 s) and with a total time of about 30 min. As a result, we obtain

$$
\begin{array}{cc}
step22 & a(22) = 0.0006 \\
g(22) & 0.320019 \quad -0.219987 \quad -0.018052 \quad -0.350698 \\
J(Y, g(22)) = 4.9156 \times 10^{-8} & \\
del_i & 0.000062 \quad -0.000055 \quad 0.002922 \quad 0.001996.
\end{array}
$$

So, upon replacing a linear approximation for a quadratic one, we have reduced the identification time by more than a factor of 2, and the maximum estimation errors do not exceed 0.3. In the problem of identifying two parameters and a two-dimensional vector of initial conditions for an oscillatory link, we conducted a numerical analysis for five implementations of quadruples of these random variables, in which the unknown parameters $\theta_1, \theta_2, \theta_3, \theta_4$ were assigned five values (see below)—depending on the implementation number:

$$
\begin{array}{llll}
(1) \ 0.043 & 0.207 & -0.057 & -0.393 \\
(2) \ -0.350 & 0.303 & 0.433 & 0.348 \\
(3) \ -0.232 & -0.207 & -0.483 & 0.332 \\
(4) \ 0.031 & -0.437 & 0.217 & 0.192 \\
(5) \ -0.020 & 0.200 & 0.246 & -0.353.
\end{array}
$$

The relative estimation errors of these random parameters, depending on the implementation number, are as follows:

$$
\begin{array}{llll}
(1) \ 0.000020 & 0.000253 & 0.001366 & -0.000095 \\
(2) \ 0.000001 & 0.000005 & 0.000004 & -0.000015 \\
(3) \ 0.000007 & -0.000145 & 0.000028 & 0.000037 \\
(4) \ 0.000012 & -0.000071 & -0.000069 & -0.000118 \\
(5) \ 0.000731 & 0.000550 & -0.000462 & -0.000124.
\end{array}
$$

The given data imply that the polynomial approximation algorithm solves the problems of identification and of nonlinear filtration with high accuracy; moreover, in all implementations, the relative estimation errors are vanishing. After the zeroth approximation, this accuracy is reached for four iterations. The evolution of relative estimation errors in the iterative process is illustrated with an example:

$$
\begin{array}{lllll}
\text{Zeroth approximation} & -0.001369 & 0.179021 & -0.697325 & -0.058656 \\
\text{1st iteration} & -0.882636 & 0.003379 & 0.085963 & 0.013298 \\
\text{2nd iteration} & 0.094758 & -0.000561 & 0.015023 & 0.031883 \\
\text{3rd iteration} & 0.001326 & 0.076830 & 0.003879 & 0.008216 \\
\text{4th iteration} & 0.000012 & -0.000004 & -0.000069 & -0.000118.
\end{array}
$$

So, after four iterations, the relative estimation errors are of the order of 10^{-5}.

5.8 Smoothing, Filtration, and Forecasting (SFF) by Observations in Noise for a Nonlinear Dynamic System

In many applied problems of nonlinear dynamic system control, one needs estimates of the state vector—of the current one or at a fixed instant—when in discrete time one observes in noise some (generally speaking) nonlinear functions from current state vectors. If observations are made after a fixed instant, we have a smoothing problem; if the instants of current observations are coincident with the instants of estimation, we have a filtration problem; if observations are made before an estimation instant, we have a forecasting problem.

In the literature (see, for example, [4]) the optimal—in the root-mean-square-sense—solution of the above-mentioned problems is described for linear dynamic systems.

In Chap. 1 of this book, for a linear dynamic system disturbed by random vectors, and for linear observations, we gave—without a hypothesis about the laws of random variable distribution—a derivation of a well-known recurrent algorithm of a discrete Kalman filter (KF) based on a principle of observation decomposition. This algorithm delivers an optimal in the root-mean-square-sense solution to the problem of filtration: that is, to the problem of estimating the current state vector at the instant of receiving the current observation.

In Chap. 1, we also gave a solution to the problem of optimal linear interpolation, which is the problem of optimal smoothing. For this problem, at the instant of receiving the current observation, one should find an optimal in the root-mean-square-sense estimate of the state vector that existed earlier, at some given discrete-time instant.

There is a well-known heuristic solution (see, for example, [5]) of the nonlinear filtration problem that delivers a discrete algorithm of the extended Kalman filter (EFK). As a basis of the EKF structure is a sequence of linearizations of nonlinear functions of the mathematical model of a dynamic system in the neighborhood of the sequence of approximate estimation vectors.

The two-step scheme of the EFK algorithm is similar to that of the KF algorithm.

In step 1, after an estimation vector is built for an observation instant t, a forecasting—an approximate definition of statistical characteristics—occurs of the first and second moments for the state vector at instant $t + 1$. The forecast is carried out with the use of a matrix of private derivatives of the state vector (Jacobian) and, hence, implies differentiability of the right side of (8.1) and the possibility of its linearization with respect to incremental components of the state vector that is obtained when passing from t to $t + 1$.

In step 2, after factual observations at instant $t+1$, the algorithm performs a linear correction of the predicted state vector, optimal in the root-mean-square sense.

In recent years, a scheme of the extended Kalman filter that needs no linearization and computation of a Jacobian has been published and applied. A corresponding algorithm for an unscented Kalman filter (UKF) is presented in [6–9].

We must point out that the two-step scheme of the algorithm is rather plausible and in many applicable cases leads to quite satisfactory results. However, the scheme is heuristic, as in the published papers there is no theoretical substantiation for it.

So, incidentally, we know of no statistical measure of estimation accuracy being attained by the EKF and UKF algorithms.

Earlier in this chapter we examined a special case of a nonlinear smoothing problem (the problem of estimating a vector of stationary random parameters Ω_θ in the absence of random disturbances of the dynamic system and of random observation errors) by means of a contraction operator. With the use of formulas of the form (4.2) (Chap. 4), one can also easily conduct a similar consideration in the presence of additive random observation errors.

However, using a contraction operator is conjugate with loss of information on the estimation vector, delivered by the observation vector. Hence, we cannot assert that the estimates of vectors obtained in Sect. 5.7, are estimates over the observation vector, much like estimates, optimal in the root-mean-square sense.

The method of polynomial approximation, the decomposition of observations, and the ensuing recurrent algorithm presented in Chap. 1 permit one to create a recurrent algorithm of smoothing, filtration, and forecasting (RSFF algorithm) that builds a sequence of estimation vectors that is composed of approximations convergent to vectors of conditional expectations of estimation vectors. The RSFF algorithm is theoretically well founded by the content of Chaps. 1–4. The errors in approximations uniformly tend to zero with an increase in the integer d and in the number of nodes of the lattice, which covers the a priori parallelepiped Ω_{x_0} at computing corresponding integrals. Next, we present a scheme of the RSFF algorithm and its test on several nonlinear problems.

5.8.1 *Mathematical Model of Dynamic System and Observations*

Let the model of the dynamic system be presented by the differential equation

$$\mathrm{d}x/\mathrm{d}t = f(x, \eta, t), \tag{8.1}$$

where the vector x is a current system state vector, the vector of initial conditions $x(0)$ is fully or partly unknown, and $\eta(t)$ is a random vectorial process of disturbances. We suppose that there is a priori information on vector $x(0)$: $x(0) \in \Omega_{x(0)} \in R^q$.

The unknown parameters of the model serve as additional components of the state vector x of an extended dynamic system. The dependence from t in the right part of Eq. (8.1) can be a consequence of the fact that there is a control vector of the dynamic system that is a function of t. Normally, for the random vector $\eta(t)$, we assign a statistical structure that permits us to replace differential equation (8.1) for a discrete-time equation:

$$x_{k+1} = f_k(x_k) + \eta_k, \tag{8.1'}$$

where $x_k = x(tk)$, vector $f_k(x_k)$ is defined by numerical integration (8.1) from instant t_k to instant $t_k + 1$ at $\eta = 0$ and at initial condition x_k, and the random vectors are centered and independent,

$$E(\eta_k \eta_k^T) = \Psi_k.$$

As a result of successive observations, there is a scalar sequence of N random numbers $y_1, \ldots, y_k, \ldots, y_N$:

$$y_k = H_k(x_k) + \xi_k, \tag{8.2}$$

where

$$x_k = x(t_k), \quad x_k^T = \|x_{k,1}, \ldots, x_{k,i}, \ldots, x_{k,n}\|^T,$$

where ξ_k is a random observation error, $E(\xi_k)$=0, $E(\xi_i \xi_k)$=$\delta_{i,k}\sigma_k^2$, $t_1, \ldots, t_k, \ldots, t_N$, is a sequence of observation instants. The variables $y_1, \ldots, y_k$ are components of the vector Y_k.

5.8.2 *Conceptual Algorithm for Smoothing, Filtration, and Forecasting (SFF Algorithm)*

The conceptual SFF algorithm is based on a polynomial approximation of estimation vectors that are optimal in the root-mean-square sense. After observations of components of the vector Y_N, the SFF algorithm should build approximation vectors to the components of conditional expectation vectors of the estimation vectors x_0, x_N, x_N^*, and specifically, vectors $\hat{x}_0(d, Y_N), \hat{x}_N(d, Y_N), \hat{x}_N^*(d, Y_N)$ of dimensionality $n \times 1$, $N^* > N$. As is well known, the conditional expectation vectors are optimal—in the root-mean-square-sense—estimates of state vectors of the dynamic system x_0, x_N, x_N^* at instants $0, \ldots, t_N, \ldots, t_N^*$.

These estimates are functions of the vector Y_N and of the integer d, and also are functions of a priori region $x(0) \in \Omega_{x(0)} \in R^n$ and of the statistical characteristics of random variables ξ_i.

Furthermore, error covariance matrices of quasi-optimal estimates of these vectors should be defined.

The problem is solved by polynomial approximation for the vector $E(X_{(0,N,N^*)}|Y_N)$ of the conditional expectation of the vector

$$X_{(0,N,N^*)}^T = \|x_0^T \quad x_N^T \quad x_{N^*}^T\|^T$$

of dimensionality $3n \times 1$. The components of the estimation vector of the vector $X(0, N, N^*)$ are presented by polynomials with respect to components of the vector Y_N, whose power does not exceed a given integer d.

Let's select the integer d and build the vector $W(d, N)$ of dimensionality $m(d, N) \times 1$, whose components are random values of products of the form $y_1^{a_1}, \ldots, y_N^{a_N}$, $0 <= a_1 + \cdots + a_N <= d$, where $0 <= a_i$ are integers and $i = 1, \ldots, N$. Dimensionality $W(d, N)$ is equal to $m(d, N)$, that is, to the number of solutions of integer inequality $0 <= a_1 + \cdots + a_N <= d$. This number is defined by a recurrent formula from Sect. 2.3 in Chap. 2.

Let's build vector V of dimensionality $(q + m(d, N)) \times 1$:

$$V^T = \| X^T_{(0,N,N^*)} \quad W(d, N)^T \|.$$

As an a priori entry to the SFF algorithm, let's provide the a priori first and second statistical moments for components of the random vector V. These a priori data present an expectation vector

$$\hat{V} = E(V)$$

and a covariance matrix

$$Q_V = E((V - (\hat{V}))(V - \hat{V})^T).$$

The vector $\hat{V}$ and the matrix Q_V can be found by a Monte Carlo method if a statistical model has been set that permits us to generate on the computer a random process of disturbances of a dynamic system. If there are no random disturbances of a dynamic system, then the desired a priori first and second statistical moments are to be defined via computation of multidimensional integrals, using a modified method of trapezes. The further presentation and solution of model examples in this chapter correspond just to this case.

So, let us have an a priori vector $\hat{V} = E(V)$ and matrix $Q_V = E((V - \hat{V})(V - \hat{V})T)$. These can be divided into the vectors $E(X_{(0,N,N^*)})$, $E(W(d, N))$ and matrix blocks

$$C(X) = E((X_{(0,N,N^*)}) - E(X_{(0,N,N^*)})(X_{(0,N,N^*)}) - E(X_{(0,N,N^*)})^T),$$
$$L(X, W) = E((X_{(0,N,N^*)}) - E(X_{(0,N,N^*)})(W(d, N) - E(W(d, N)))^T),$$
$$Q(W) = E((W(d, N) - E(W(d, N)))(W(d, N) - E(W(d, N)))^T).$$

The estimation vector $\hat{X}_{(0,N,N^*)}(Y_N, d)$ and estimation error covariance matrix $C(\hat{X}, d)$ can be presented by Eqs. of Chaps. 1 and 2:

$$\hat{X}_{(0,N,N^*)}(Y_N, d) = E(X_{(0,N,N^*)} + L(X, W)Q(W)^{-1}(W(d, N) - E(W(d, N))), \tag{8.3}$$

$$C(\hat{X}, d) = C(X) - L(X, W)Q(W)^{-1}L(X, W)^T. \tag{8.4}$$

Equation (8.3) presents an estimation vector of the vector $X_{(0,N,N^*)}$, optimal in the root-mean-square sense, on a set of polynomials from components of the vector Y_N whose power does not exceed a given integer d. But

$$E(E(X_{(0,N,N^*)}|Y_N)) = E(X_{(0,N,N^*)}),$$

$$L(X|Y_N), Y_N), W) = L(X, W).$$

Hence,

$$\hat{E}(X_{(0,N,N^*)}|Y_N)(Y_N, d) = \hat{X}_{(0.N.N^*)}(Y_N, d).$$

So, formula (8.3) simultaneously presents the estimation vector of the conditional expectation vector $E(X_{(0,N,N^*)}|Y_N))$, having similar optimal qualities.

Let the matrix $C(\hat{E}(X|Y, d)$ be an estimation error covariance matrix $E(X_{(0,N,N^*)}|Y_N)$.

The basic theorem of Chap. 2 implies that

$$C_{\hat{E}}(X|Y, d) \to 0, d \to \infty. \tag{8.5}$$

So, the SFF conceptual algorithm, via formulas (8.3) and (8.4), delivers approximate solutions for nonlinear problems of smoothing, filtration, and forecasting. This solution is theoretically well founded, optimal (in the root-mean-square sense) on a set of approximating polynomials s of preset power d, and uniformly converges to the conditional expectation vector upon an increase in these polynomials' power.

5.8.3 *Qualitative Comparison of SFF Algorithm and* РФК *Algorithm*

Let's suppose that we only solve a nonlinear filtration problem:

$$X_{(0,N,N^*)} = x_N.$$

In this case, we conduct a qualitative comparison of the SFF algorithm and the recurrent EKF algorithm.

A scheme of a recurrent EKF algorithm looks like this: The a priori data for the vector X_0 is the vector of the a priori estimate (vector of the first statistical moments) $X(0|0)$ and the covariance matrix of the a priori estimation errors $\hat{X}(0|0)$.

Let's assume that after observations $y_1, \ldots, y_k$, at step k, $1 <= k <= N$, of the algorithm's computation, the estimation vector $x(k|k)$ and the estimation error covariance matrix $\hat{x}(k|k)$ were defined. Then the $(k+1)$st step of computation is divided into parts I and II.

I. Computation of forecast

$$\hat{x}(k+1|k) = f_k(\hat{x}(k|k)), \tag{8.6}$$

$$P(k+1|k) = F_k P(k,k) F_k^T + \Psi_k, \tag{8.7}$$

where F_k is a Jacobian vector-function $f_k(\hat{x}(k|k)$.

We notice that if the dynamic system is set by differential equation (8.1), then the definition of the Jacobian F_k requires numerical integration of the corresponding matrix system of differential equations on each segment $[tk, tk+1]$.

II. Computation of estimation vector and of estimation error covariance matrix

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)(y_{k+1} - h_{k+1}(\hat{x}(k+1|k)), \tag{8.8}$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)H(k+1)P(k+1|k), \tag{8.9}$$

where

$$K(k+1) = H(k+1)P(k+1|k)H(k)^T + \sigma_{k+1}^2,$$

and $H(k+1)$ is a Jacobian of function $h_{k+1}(\hat{x}(k+1|k))$.

1. *Let d = 1.*

The SFF delivers an optimal in the root-mean-square-sense solution of the filtration problem on a set of linear combinations of components of observation vector Y_N. The SFF uses the whole a priori information on a statistical connection between vectors x_N and Y_N that is contained in matrices of a priori second moments $L(X, W)$, $Q(W)$.

In the case under consideration [C(X) = P(0|0)], the EKF algorithm obviously does not use the a priori matrices $L(X, W)$, $Q(W)$, and relationship (8.8) presents only the obvious (linear) dependence of EKF estimation vectors from components of the observation vector. For this reason, at first sight, optimal in the root-mean-square-sense linear estimates of the SFF algorithm deliver an estimate accuracy that is at least not smaller than that of the EKF algorithm. However, components of vector $\hat{x}(k|k)$ depend on the observation vector Y_k and are initial conditions for the nonlinear differential equations whose numerical integration—on segment $[tk, tk+1]$—will define $\hat{x}(k+1|k)$ and F_k. Thus, the estimation vectors delivered by the EKF algorithm in fact nonlinearly depend on observations, and their accuracy can be greater than the accuracy of the linear estimates of the SFF algorithm at $d = 1$. The reality of such a situation confirms an example to be considered below.

2. *The EKF algorithm has no reserves to increase the accuracy of estimation reached by a modernization of the algorithm scheme being implemented.*

Increasing the number of terms of a power series expansion of nonlinear vector-functions cannot be regarded as a similar modernization, because usually one cannot evaluate the residual terms of these expansions.

Relationship (8.5) delivers exhausting data on the conceptual possibilities of the SFF algorithm. An increase in the number of terms of the polynomial approximation reduces estimation errors upon the average.

3. *The EKF algorithm is not applicable if the model of the dynamic system contains nondifferentiable functions.*

For the SFF algorithm, there are similar restrictions because its implementation requires only operations of numerical integration.

4. *Upon the practical implementation of the SFF algorithm, the definition of a priori vectors and matrices* $(X_{(0,N,N^*)})$, $E(W(d, N))$, $C(X)$, $L(X, W)$, *and* $Q(W)$ *can demand considerable computational time.* With an increase in a given integer d, this time quickly grows. However, a priori data that are found and saved in memory make it possible—for arbitrary but satisfying restrictions of model (8.2) observation vectors Y_N—to easily obtain [under formulas (8.3) and (8.4)] a polynomial approximation vector of the conditional expectation vector and an estimation error covariance matrix, corresponding to a selected d. In so doing, one simultaneously approximately solves nonlinear problems of smoothing, filtration, and forecasting (extended Kalman filter).

Practical implementation of the EKF algorithm can also demand considerable computational time to conduct multiple integrations of matrix differential equations to define a sequence of Jacobians F_k. For each new Y_N, the computational process should be repeated.

Note that the known type of EKF algorithm roughly solves only a nonlinear filtration problem.

5.8.4 Recurrent form of the SFF (RSFF) Algorithm

The process of the recurrent refinement of the vector $\hat{X}_{(0,N,N^*)}(d, Y_N)$ of the estimation vector $X(0, N, N^*)$ is composed of $m(d, N)$ steps. At each step, in compliance with the principle of the decomposition of observations (outlined in Chap. 1), by the new (being updated) a priori (for this step!) data, a refinement occurs to the current estimation vector of vector $X_{(0,N,N^*)}$ and of a vector, composed of components of vector $W(d, N)$, not yet used by the algorithm. The estimate of these components is their statistical forecast implemented after the algorithm has used a part of the components of the vector $W(d, N)$. Simultaneously, one computes an estimation error covariance maxtrix reached at this step. At the last, $m(d, N)$th step, one has no forecast, and the last refinement to the vector $X(0, N, N^*)$ and the definition of a total estimation error covariance matrix occur.

For step k of the computational process, we assume the following notations:

- The vector V_k is composed of $3n$ components of the vector $X(0, N, N^*)$ and of $m(d, V) - k$ components $w_k + 1, \ldots, w_m(d, N)$.
- The vector $\hat{V}_k(w_k))$ is a linear and optimal—in the root-mean-square-sense—estimate of the vector V_k after the algorithm's use of component w_k and of all the previous components $W(d, N)$.
- The scalar $\hat{w}_{k+1}(w_k))$ is the $(3n+1)$st component of the vector $\hat{V}_k(w_k)$ [an estimate of component w_{k+1} after the algorithm's use of component w_k and all the previous components $W(d, N)$].
- $\hat{V}_k(w_k)^1$ is a vector obtained from $\hat{V}_k(wk)$, with the exclusion of component w_{k+1} of this vector.
- The matrix $Q_k = E((V_k - \hat{V}_k(wk))(V_k - \hat{V}_k(wk))^T)$ is an estimation error covariance matrix of vector V_k after the algorithm uses a component V_k and all the previous components $W(d, N)$.
- The scalar q_k is the $(3n + 1)$st diagonal element of the matrix Q_k^1: Q_k [estimation error variance of estimate of component w_{k+1} after using the component w_k and all the previous components $W(d, N)$].
- The matrix Q_k^1 is a matrix obtained from matrix Q_k by excluding the $((3n+1)+1)$st row-vector and the $(3n + 1)$st column-vector.
- The vector b_k is the $(3q + 1)$st column-vector of the matrix Q_k from which one has excluded the $(3q + 1)$st component.

The recurrent algorithm is composed of $m(d, N)$ computational steps, in the process of which the vectors

$$V_1, V_2, \ldots, V_{m(d,N)} = X(0, N, N^*$$

are successively estimated by linear and optimal—in the root-mean-square-sense—functions from component $w_1, w_2, \ldots, w_{m(d,N)}$. At step k, the RSFF formulas are of the form

$$\hat{V}_{k+1}(w_{k+1}) = \hat{V}_k(w_k)^1 + q_k^{-1} b_k (w_{k+1} - \hat{w}_{k+1}(w_k)), \tag{8.10}$$

$$Q_{k+1} = Q_k^1 - q_k b_k b_k^T, \tag{8.11}$$

where

$$k = 0, \ldots, m(d, N) - 1, V_0 = V, \hat{V}_0(w_0)^T = V,$$
$$\hat{w}_1 = E(w_1), Q_0 = Q_V.$$

For the $(k + 1)$st step of the recurrent estimation process, the vector $\hat{V}_k(w_k)$ and matrix Q_k are new data on the first and second a priori (for step $k + 1$) statistical moments of components of the vector V_{k+1} before coming to the algorithm's input of value w_{k+1}.

We notice that the sequence of random variables of the form $w_{k+1} - z_{w_{k+l}}(w_k)$ creates a renewing sequence.

At $k = m(d, N)$, the RSFF algorithm defines a vector

$$\hat{X}_{(0,N,N^*)}(d, Y_N) = \hat{V}_m(d, N)(w_m(d, N)$$

of the last estimate of vector $\hat{X}_{(0,N,N^*)}$ after the algorithm uses the last component $w_m(d, N)$ and the estimation error covariance matrix $C^o = Q_m(d, N)$.

Let $W(k)$ be a vector of dimensionality $k \times 1$, composed of the first k components of the vector $W(d, N)$.

The vector $\hat{X}_{(0,N,N^*)}(W(k))$, composed of the first 3n components of the vector $\hat{V}_k(wk)$, is a linear and optimal in the root-mean-square-sense estimate of vector $\hat{X}_{(0,N,N^*)}$ after the algorithm uses vector $W(k)$ and at the same time vector

$$\hat{X}_{(0,N,N^*)}(0) = \bar{X}_{(0,N,N^*)}.$$

The top left C_k of matrix Q_k of dimensionality $3n \times 3n$ is an estimation error covariance matrix of vector $X_{(0,N,N^*)}$ after the algorithm uses vector $W(k)$.

Let $l(k)$ be a vector composed of the first $3n$ components of the vector b_k. Then the formula, which presents the evolution of the covariance matrix C_k as a function of the number k of components of vector $W(d, N)$ used, is of the form

$$C_k = Q(k) - q_1^{-1} l(1) l(1)^T - \ldots - q_k l(k-1) l(k-1)^T. \tag{8.12}$$

5.8.5 *About Computation of a Priori First and Second Statistical Moments*

The above implies that saving the computation of the first and second a priori statistical moments of the vector $\hat{X}_{(0,N,N^*)}$ is crucial for implementing the RSFF algorithm.

If the model of the dynamic system is presented in Eq. (8.1), then the Monte Carlo method is apparently a unique way of computing the above-mentioned a priori data. In the absence of random disturbances of the dynamic system [in Eq. (8.1), $\eta(t) = 0$] or upon representing the system model in the form of Eq. (8.1′), the a priori data are effectively computed by a modified method of trapezes.

The given equations imply that an additional solution of the forecasting problem does not change the algorithm's scheme and is reduced only upon an increase in the estimated vector's dimensionality by an amount n. That is why to simplify the statement, we have given the computational scheme only for smoothing and filtration problems—in the absence of random disturbances of the dynamic system. However, in model examples, illustrating the method's efficiency, one numerically solves smoothing, filtration, and forecasting problems.

The a priori statistical moments are defined by an a priori region Ω_{X_0} on which the random distribution of vectors X_0 is considered uniform and the distributions of independent random variables $\xi_i, i = 1, \ldots, N$, on segment $[-a, a]$, is also

considered uniform. These hypotheses are not conceptual. The method under study permits us to consider the given distributions of random vectors x_0, ξ_i both during a test run and during the computation of multidimensional integrals.

Assuming that value a is insignificant, and expanding the differences in integrands (4.2) of Chap. 4 into series, we will find appropriate representations.

5.8.6 *Evaluation of the Initial Conditions and Parameter of the Van der Pol Equation*

In [10] the problem of estimating the parameter λ and the derivative of $\dot{x}(t_0)$ for the nonlinear differential Van der Pol equation is

$$(\dot{\dot{x}}) + \lambda(x^2 - 1)\dot{x} + x = 0,$$

if the error solution is absent at three points in time. We consider the problem of estimating the parameter λ and the initial conditions $x(0), \dot{x}(0)$ if the equation is replaced by its discrete analog, which corresponds to the integration of a simple Euler's method for a discrete value of 1/120 s and a duration of observations of 12 s. Data reduction was performed by adding 120 consecutive primary observation variables $x(t)$, and the dimension of W is 12×1 for $d = 1$.

Suppose that there is a significant a priori dispersion of estimated values:

$$10(1 - 0.95) \le \lambda \le 10(1 + 0.95), (1 - 0.95) \le x(0) \le (1 + 0.95), (1 - 0.95) \le \dot{x}(0) \le (1 + 0.95).$$

The values of the estimated $\lambda, x(0), \dot{x}(0)$ are a random number and belong to a priori parallelepipeds in R^3. We assume that the estimation error is a measure of the ratio of $\sigma(pos)/\sigma(pr)$-relations of a posteriori standard deviation and a priori standard deviation. The following table shows that the relationship depends on the values of d and the corresponding values of $m(d, N_1)$, where N_1 is the number of compressed sequence elements:

d	$m(d, N_1)$	$\sigma(pos)/\sigma(pr)(\lambda)$	$\sigma(pos)/\sigma(pr)(x)$	$\sigma(pos)/\sigma(pr)\dot{x}$
1	12	0.875	$0.758 * 10^{-6}$	$0.937 * 10^{-4}$
2	90	0.550	$0.330 * 10^{-6}$	$0.411 * 10^{-4}$
3	454	0.454	$0.277 * 10^{-7}$	$0.344 * 10^{-4}$
4	1819	0.397	$0.123 * 10^{-7}$	$0.143 * 10^{-4}$

The table shows that the estimation errors of the initial conditions are small for all values of d. The parameter estimation error λ is slowly decreasing with increasing d.

The parameter estimation error λ is greatly reduced if the organized search reduces the estimated parameters. Hence, let the ribs of a priori the parallelepiped decrease

by 10 times. Then, for $d = 3$, $\sigma(pos)/\sigma(pr)(\lambda)$ was equal to 0.00576—a decrease of about a factor of 100.

5.8.7 *Smoothing and Filtration for a Model of a Two-Level Integrator with Nonlinear Feedback*

We assume that the equation of the form (8.1) for a model of a dynamic system is of the form

$$\mathrm{d}x_1/\mathrm{d}t = x_2(0)\mathrm{d}x_2/\mathrm{d}t = -x_1^3 + \sin(6t/3 + 0.5),$$

$$\Omega_{x(0)} : -b <= x_1(0), x_2(0), <= b.$$

Let's do a transition with a discrete-time step of τ to a dynamic system discrete model, obtained via numerical integration of the equations by a simple Euler's method with step τ.

Let's assume that the model of observations (8.2) looks like the following:

$$y_k = x_1(t_k) + x_2(t_k) + \xi_k,$$

where the intervals of time between successive observations are equal to 10τ, and six observations are conducted: $t_k = 10\tau k$, $\tau = 0.1$ s, $k = 1, \ldots, 6$.

The smoothing problem is solved for random variables $x_1(0)$, $x_2(0)$, and the filtration problem is solved for random variables $x_1(6)$, $x_2(6)$.

The accuracy of solving problems at instant t_N, reached from simulation modeling, is characterized by values of the standard deviation (SD) of estimation errors, where σ_1, σ_2 are smoothing errors, and σ_3, σ_4 are filtration errors.

The values σ_i were defined by the Monte Carlo method at 100,000 implementations of a recurrent algorithm when random variables $x_1(0)$, $x_2(0)$, ξ_k, $k = 1, \ldots, 6$ dispersed. Furthermore, the variables σ_i were defined upon computation of the estimation error covariance matrix C_N. The differences in experimental and calculated values σ_i did not exceed 1 This circumstance proves the correctness of formulas in the algorithm and of its implementation. Upon numerical integration of two-dimensional integrals, the value r changed within the limits 10–100, which had little influence on the accuracy of smoothing and filtration.

The variables $\sigma_i(0)$ are a priori (before observations) dispersion of estimated variables.

The computations were conducted at $d = 1, 2, 3$ [the components of the estimation vector of variables $x_1(0)$, $x_2(0)$, $x_1(6)$, $x_2(6)$ were linear, quadratic, and cubic polynomials from fixed values $y_1, \ldots, y_6$; at $d = 1, 2, 3$, the number of polynomial members $m(d, N)$ was equal, respectively, to 6, 27, 83], and at $a = 0.02$ (random errors of observations uniformly disperse in segment $[-0.02, 0.02]$).

Let $b = 1$. Following are the values of a priori $\sigma_i(0)$, characterizing the dispersion of estimated variables prior to the beginning of observations:

$$\begin{array}{cccc} \sigma_1(0) & \sigma_2(0) & \sigma_3(0) & \sigma_4(0) \\ 0.5774 & 0.5774 & 0.7435 & 0.9010. \end{array}$$

The values σ_i—versus the number of observations k at $d = 1$—are as follows:

k	σ_1	σ_2	σ_3	σ_4
1	0.5766	0.1749	0.7132	0.6565
2	0.2759	0.1730	0.4656	0.6250
3	0.2758	0.1669	0.4567	0.4838
4	0.2726	0.1254	0.3910	0.3124
5	0.2148	0.1185	0.1427	0.2806
6	0.2125	0.1169	0.0426	0.0436.

Comparing a priori $\sigma_i(0)$ to the values of the last row of the table, we see that six observations and their subsequent optimal linear processing in solving smoothing problems have reduced the possible dispersion of the estimated variable by a factor of about 3–5; in solving filtration problems, this dispersion has been decreased approximately 15–20 times.

At $d = 2$, the accuracy of smoothing practically did not change; at $d = 3$, the errors of smoothing and of filtration sharply decreased in comparison with the results for $d = 1$. The characteristics of the accuracy of smoothing and filtration are given for six observations:

d	σ_1	σ_2	σ_3	σ_4
2	0.2123	0.1159	0.0423	0.04321
3	0.0192	0.0168	0.0031	0.0038.

Thus, six observations and their optimal in the root-mean-square-sense processing at the third power of approximating polynomials have reduced the a priori dispersion of smoothing errors approximately 50 times; the a priori dispersion of filtration errors has been decreased approximately 200–300 times.

We notice that at $d = 1$, the computational time for 100,000 implementations on low productivity with a changing integer r is within the limits 10–100.

At $d = 3$, the computational time for 100,000 implementations sharply increases, and at $r = 300$ is 15 min. At these settings, the computation of vectors and matrices of the a priori data takes 2.5 min.

5.8.8 The Solution of a Problem of a Filtration by the EKF Algorithm

The EKF algorithm was implemented according to Eqs. (8.6)–(8.9) for a discrete model of the considered nonlinear dynamic system. Elements of Jacobians F_k were defined by the finite-difference method: by discrete numerical integration of the equations of the model on segments $[t_k, t_k + 1]$ at successive increments of initial conditions (of components of vectors $\hat{x}(k|k)$) by values of 0.00001, by definition of the corresponding differences, and by their division by these values.

The results of model by a Monte Carlo method at $b = 1$ and 100,000 implementations of the algorithm are as follows:

σ_1	σ_2	σ_3	σ_4
–	–	0.03626	0.03629.

From comparison with the data of the RSFF algorithm, it follows that the EKF algorithm delivers filtration errors that are smaller than those of the RSFF algorithm at $d = 1$. However, at $d = 3$, filtration errors are 10 times smaller than for the EKF. Setting $b = 1.5$, we will increase the a priori region of dispersion $\Omega_{x(0)}$ by 1.5 times. The errors of smoothing and filtration have increased. However, the ratio of the RMSE of these errors to the values of the a priori SD changed little.

The data, which are similar to the data presented just above, are as follows:

	$\sigma_1(0)$	$\sigma_2(0)$	$\sigma_3(0)/\sigma_4(0)$	
	0.8660	0.8660	1.4081	3.8242
d	σ_1	σ_2	σ_3	σ_4
1	0.3114	0.30183	0.5909	0.5910
3	0.0351	0.0323	0.0164	0.0174.

The further increase in the value b stops the functioning of the RSFF algorithm: At some initial conditions $x_1(0), x_2(0)$, the computational process diverges and its components reach values of the order of 10^{2000}–10^{5000}.

At $b = 1.1$, a similar termination of the functioning of the EKF algorithm occurs.

5.8.9 Identification of Velocity Characteristic of the Integrator and of the Nonlinearity of the Type "Backlash"

Let there be a dynamic system with unknown random parameters x_3 and x_4, where x_3 is a velocity characteristic of the integrator, and x_4 is a nonlinearity characteristic of the type "backlash." We regard them as additional components of a state vector of a dynamic system.

We use an approximate model of nonlinearity of the type "backlash" (y is an output of the integrator measured without random errors): If $x_2 > 0$, then $y = x_1 - x_4$; if $x_2 < 0$, then $y = x_1 + x_4$.

Then the equations of the models of the dynamic system and of observations will acquire the form

$$\mathrm{d}x_1/\mathrm{d}t = x_2, \quad \mathrm{d}x_2/\mathrm{d}t = x_3 100 \sin(20\pi/3i) + 0.5,$$

$$\mathrm{d}x_3/\mathrm{d}t = 0; \mathrm{d}x_4/\mathrm{d}t = 0,$$

$$\omega_{x0} : b_{1,1} <= x_4 <= b_{1,2}, b_{2,1} <= x_3 <= b_{2,2},$$

$$y_k = F(t_k) + \xi_k,$$

$$x_2(t_k) > 0 : F(t_k) = x_1(t_k) - x_4 \quad x_2(t_k) < 0 : F(t_k) = x_1(t_k) + x_4,$$

$$t_k = k10\tau, \quad k = 1, \ldots, N, \quad N = 6, \quad \tau = 0.1s,$$

$$x_1(0) = 0, \quad x_2(0) = 0.$$

Using the EKF algorithm is impossible because of the availability of the nondifferentiated function $F(t_k)$ in the model of observations.

Then, the smoothing problem (identification problem) is solved for random variables x_3, x_4, the filtration problem is solved for random variable $x_1(6)$, and the forecasting problem is solved for random variable $x_2(7)$.

Let

$$b_{1,1} = 0; b_{1,2} = 0.5; b_{2,1} = 0.5; b_{2,2} = 1;$$

for the characteristics of the estimate's accuracy, we will assume the notation used above.

Data of the mathematical model at $d = 1$ look like

1	$\sigma_1(0)$	$\sigma_2(0)$	$\sigma_3(0)$	$\sigma_4(0)$
1	0.1443	0.1443	0.1179	0.7510
k	σ_1	σ_2	σ_3	σ_4
1	0.0186	0.1441	0.1178	0.7501
6	0.0025	0.0035	0.0029	0.018.

As is obvious, the RSFF algorithm delivers a considerable accuracy of identification of the velocity characteristic of the integrator and of the nonlinearity of the type "backlash" despite the small number (6) of observations and the linear approximations of components of the estimation vector.

5.9 A Servo-System with a Relay Drive and Hysteresis Loop

One uses a drive model whose velocity x_2 assumes two different values, whose module is equal to constant x_3; velocity changes the sign when the module of the value of an error in tracking a signal—varying sinusoidally—will exceed a given constant x_4.

Then the equations of models of the dynamic system and of observations will acquire the form

$$\mathrm{d}x_1/\mathrm{d}t = x_2, \quad \mathrm{d}x_2/\mathrm{d}t = F(\epsilon), \quad \epsilon = \sin((2\pi/3)t + 0.5) - x_1,$$

$$\mathrm{d}x_3/\mathrm{d}t = 0, \quad \mathrm{d}x_4/\mathrm{d}t = 0,$$

$$\epsilon > -x_4 : F(\epsilon) = x_3; \quad \epsilon < x_4 : F(\epsilon) = -x_3,$$

$$\Omega_{x(0)} : b_{1,1} <= x_3 <= b_{1,2}, \quad b_{2,1} <= x_4 <= b_{2,2},$$

$$y_k = x_1(t_k) + \xi_k,$$

$$t_k = k10\tau, \quad k = 1, \ldots, N, \quad N = 6, \quad \tau = 0.1s,$$

$$x_1(0) = 0 = x_2(0) = 0.$$

The smoothing problem (identification problem) is solved for random variables x_3, x_4, the filtration problem is solved for random variable $x_1(6)$, and the forecasting problem is solved for random variable $x_2(7)$.

Let

$$b_{1,1} = 0.5; \quad b_{1,2} = 1; \quad b_{2,1} = 0; \quad b_{2,2} = 0.25;$$

for characteristics of an estimate's accuracy, we will assume the notations used above.

The data of the mathematical model look as follows:

	$\sigma_1(0)$	$\sigma_2(0)$	$\sigma_3(0)$	$\sigma_4(0)$
	0.1443	0.0721	0.9204	0.1534
k	σ_1	σ_2	σ_3	σ_4
6	0.0171	0.0724	0.0036	0.0006.

As is obvious, one reaches a considerable accuracy in solving problems of filtration and forecasting. The identification problem for constant x_4 is not solved at the analyzed composition of observations.

5.10 Evaluation of Principal Moments of Inertia of a Solid

Let an instrument frame with rectangular axes X, Y, Z and with the point O of their crossings be connected with a solid. The projections ω_x, ω_y, ω_z on these axes of a vector of absolute angular velocity of object ω are measured by sensors of angular velocity. Additionally, we assume to know M_x, M_y, M_z, which are projections (on these axes) of the moment of external forces, M.

In the general case, the solid has—with respect to arbitrary axes X, Y, Z—six moments of inertia, three of which are called centrifugal. It has been known that for point O, there exists a rectangular system of coordinates X_O, Y_O, Z_O, with respect to which centrifugal moments of inertia are equal to zero, and the remaining moments of inertia J_x, J_y, J_z are called principal moments of inertia for the point O.

Let α, β, γ—Euler's angles defining an angular position with respect to the set system of coordinates X, Y, Z, and A—be an orthogonal matrix of directing cosines for these angles.

Let ω_x^O, ω_y^O, ω_z^O, M_x^O, M_y^O, M_z^O be projections on axes X_O, Y_O, Z_O of vectors ω and M accordingly. We assume that ω^O, M^O are vectors ω, M in axes X_O, Y_O, Z_O. Then connection of non-observable variables ω_x^O, ω_y^O, ω_z^O, M_x^O, M_y^O, M_z^O and measurable variables ω_x, ω_y, ω_z, M_x, M_y, M_z will be as follows:

$$\omega^O = A\omega, \tag{10.1}$$

$$M^O = AM. \tag{10.2}$$

Functions ω_x^O, ω_y^O, ω_z^O satisfy Euler's differential equations

$$J^x \mathrm{d}\omega_x^O/\mathrm{d}t + (J_z - J_y)\omega_y^O \omega_z^O = X_x^O, \tag{10.3}$$

$$J^y \mathrm{d}\omega_y^O/\mathrm{d}t + (J_x - J_z)\omega_z^O \omega_x^O = X_y^O, \tag{10.4}$$

$$J^z \mathrm{d}\omega_z^O/\mathrm{d}t + (J_y - J_x)\omega_x^O \omega_y^O = X_z^O. \tag{10.5}$$

We obtain the equations of the nonlinear dynamic system by adding to Eqs. (10.3)–(10.5) the following equations:

$$\mathrm{d}J_x/\mathrm{d}t = 0; \quad \mathrm{d}J_y/\mathrm{d}t = 0; \quad \mathrm{d}J_z/\mathrm{d}t = 0,$$

$$\mathrm{d}\alpha/\mathrm{d}t = 0; \quad ; \mathrm{d}\beta/\mathrm{d}t = 0; \quad ; \mathrm{d}\gamma/\mathrm{d}t = 0.$$

Setting

$$y_x(t_k) = \omega_x(t_k), \quad y_y(t_k) = \omega_y(t_k), \quad y_z(t_k) = \omega_z(t_k),$$

$$y_x(i) = \sum_{i=1}^{l} y_x(t_{l(i-1+j)},$$

we find that at instant t_k, the components of the primary observation vector $y(t_k)$ are connected with non-observable values α, β, γ and with non-observable functions $\omega_x^O, \omega_y^O, \omega_z^O$ by the relationships

$$y(t_k) = A^T \omega^O(t_k). \tag{10.6}$$

Using the sequence of components $y_x(t_k)$, $y_y(t_k)$, $y_z(t_k)$ of primary observation vector and Eqs. (10.1) and (10.2), one needs to estimate the parameters J_x, J_y, J_z, α, β, γ.

Using a fourth-order Runge–Kutta method with constant step $\tau = 0.2$ s, we will pass to the model of the dynamic system in discrete time.

In the model of the estimation process, the primary observations $y_x(t_k)$, $y_y(t_k)$, $y_z(t_k)$ correspond to instants $t_k = 5\tau k, k = 1, \ldots, 100$.

We will find the secondary observations $y_x(i)$, $y_y(i)$, $y_z(i)$, which directly enter the estimation algorithm, as follows, summing the components of every i vectors of successive primary observations. In the model, $l = 25$, $N = 4$, and the file of scalar secondary observations is composed of 12 numbers.

For functioning of the RSFF algorithm, it is necessary to find a priori first and second statistical moments of an 18-dimensional vector, composed of estimated variables $\alpha, \beta, \gamma, J_x, J_y, J_z$ and of variables $y_x(i)$, $y_y(i)$, $y_z(i)$, $i = 1, \ldots, 4$. These a priori data were defined via computation of integrals by a modified method of trapezes in R^6 at $r = 2d$ and via numerical integration of Eqs. (10.3)–(10.5) under the following conditions:

$$\omega_x(0) = 1/20, \quad \omega_y(0) = 1/20, \quad \omega_z(0) = 1/20,$$

$$M_x = 100, \quad M_y = 100, \quad M_z = 100.$$

Upon computation of integrals, the variables $\alpha, \beta, \gamma, Jx, Jy, Jz$ varied within the limits that define an a priori parallelepiped $\omega_{x(0)}$:

$$-0.1 <= \alpha, \beta\gamma <= 0.1,$$

$$J_x^O(1 - 0.1) <= J_x <= J_x^O(1 + 0.1),$$

$$J_y^O(1 - 0.1) <= J_y <= J_y^O(1 + 0.1),$$

$$J_z^O(1 - 0.1) <= J_z <= J_z^O(1 + 0.1),$$

where

$$J_x^O = 4000, \quad J_y^O = 20000, \quad J_x^O = 15000.$$

The estimate's accuracy is presented by the ratios $\sigma_i/\sigma_i(0)$, $i = 1, \ldots, 6$ that have been defined by the Monte Carlo method at 100,000 implementations: d = 1

$\sigma_1/\sigma_1(o)$	$\sigma_2/\sigma_2(o)$	$\sigma_3/\sigma_3(o)$	$\sigma_4/\sigma_4(o)$	$\sigma_5/\sigma_5(o)$	$\sigma_6/\sigma_6(o)$
0.5733	0.0754	0.5791	0.1232	0.2005	0.3504.

To reduce the ratios $\sigma_i/\sigma_i(0)$, $i = 1, ..., 6$, we will divide $\Omega_{x(0)}$ into smaller parallelepipeds, and for each of them we will repeat computations of the RSFF algorithm. So, having divided each edge into 2, we obtain 64 smaller parallelepipeds.

For one of them, the ratio $\sigma_i/\sigma_i(0)$, $i = 1, ..., 6$, looks like d = 1

$\sigma_1/\sigma_1(o)$	$\sigma_2/\sigma_2(o)$	$\sigma_3/\sigma_3(o)$	$\sigma_4/\sigma_4(o)$	$\sigma_5/\sigma_5(o)$	$\sigma_6/\sigma_6(o)$
0.4716	0.0434	0.4680	0.0534	0.0561	0.0712.

If the RSFF algorithm uses $d = 2$, then for this parallelepiped the ratio $\sigma_i/\sigma_i(0)$, $i = 1, ..., 6$, appreciably decreases:

$\sigma_1/\sigma_1(o)$	$\sigma_2/\sigma_2(o)$	$\sigma_3/\sigma_3(o)$	$\sigma_4/\sigma_4(o)$	$\sigma_5/\sigma_5(o)$	$\sigma_6/\sigma_6(o)$
0.3616	0.3603	0.0228	0.0226	0.0233	0.0153.

The given data imply that the evaluation of the principal moments of inertia of a solid has been done with errors, which are on average 50 times less than the a priori errors. A further increase in the accuracy of evaluations is reached by increasing the time of primary observations and the integer d.

5.11 Nonlinear Filtration at Bounded Memory of Algorithm

In control problems, one often finds in the process of observations at time instants $t_1, ..., t_k, ...$ that the estimation algorithm is to send the estimation vectors of the current state vectors $x_1, ..., x_k, ...$ to the control system for feedback control.

The RSFF algorithm outlined earlier in principle solves a similar problem if one successively sets value t_N equal to $t_1, ..., t_k, ...$ and, under the formulas given earlier, defines estimation vectors $\hat{X}_{(0,1)}(d, Y_1), ..., \hat{X}_{(0,k)}(d, Y_k), ...$ The control system is input estimation vectors

$$\hat{X}_1(d, Y_1), ..., \hat{X}_i(d, Y_k), ...$$

that are composed of the last n components of the control vectors $\hat{X}_{(0,1)}(d, Y_1), ..., \hat{X}_{(0,k)}(d, Y_k), ...$ and serve as approximations, optimal in the root-mean-square-sense estimates of the current state vectors $x_1, ..., x_k, ...$ being implemented by polynomials from components of the vectors $Y_1, ..., Y_k, ...,$ whose power does not exceed d. However, a similar solution is unacceptable because of the growing

[with a rise in $t_{(k)}$] dimensionality of vectors and matrices of a priori data, and also because of an impermissible increase [due to inevitable errors in mathematical model (8.1), (8.1′) and computational errors] in the volume of the "last" state vectors and of observation vectors influencing the control selection at a given instant of time.

Hence, one needs a design of a quasi-optimal estimation algorithm with constant memory, when, at the present instant, the control selection is obviously influencing only a fixed number p of preceding state vectors and observation vectors. We present the following scheme for an algorithm of a quasi-optimal nonlinear filtration with finite memory whose special case is the EKF.

At instant t_k in computer memory, p scalar observations of the form (8.2) are fixed for instants $t_{k-p}, t_{k-p+1}, ..., t_k$. The sequence of these observations composes vector $Y_{k,p}$.

We assume that during the preceding algorithm steps, by results of observations of random variables $y_1, ..., y_{k-p-1}$, the a priori [for an instant t_{k-p}] vector $\hat{x}_{k-p}(Y_{k-p-1}, d)$ and the covariance matrix $\hat{x}_{k-p}(Y_{k-p-1}, d)$ were defined, which are a priori expectations and the second centered statistical moments of vector x_{k-p} components.

Then, it is necessary to create a statistical mechanism that would generate random implementations of vector x_{k-p} corresponding to the a priori statistical data of these vectors (they are listed above). These random implementations are random initial conditions for numerical integration of Eq. (8.1) [or (8.1′)] and for computation of the implementations of random state vectors $x_{k-p+1}, ..., x_k$ and random results of observations [by Eq. (8.2)] on an interval of time $[t_{k-p}, t_k]$.

Let's conduct the factorization of an a priori covariance matrix:

$$C_{x_{k-p}}(Y_{k,p-1}, d) = GG^T.$$

The factorization is implemented, for example, by a known Cholesky algorithm.

Next, the statistical mechanism consists of the implementation of random vectors x^*_{k-p}:

$$x^*_{t-p} = \hat{x}_{k-p}(Y_{k-p-1}, d) + G\rho,$$

where ρ is a random vector, whose covariance matrix is a unit diagonal matrix.

Practice with computations revealed that a simpler diagram scheme of a statistical mechanism not requiring factorization is also possible. In this case, building random implementations of vector x^*_{k-p} carries out a uniform dispersion of component x^*_{t-p} on the parallelepiped whose center vector is equal to vector $\hat{x}_{k-p}(Y_{k-p-1}, d)$ squares of edge lengths of the parallelepiped are equal to diagonal elements of the a priori covariance matrix $\hat{x}_{k-p}(Y_{k-p-1}, d)$, multiplied by 36.

We notice that the appreciable random errors of observations "wash away" the influence of the a priori dispersion of vectors x^*_{t-p} on the estimation accuracy.

The above-mentioned statistical mechanism using a Monte Carlo method or using a method of computation of multidimensional integrals will obtain the a priori data, allowing one to apply the above-offered RSFF algorithm. These a priori data are

the first and second statistical moments for the vector that has been composed from components of vectors x_{k-p+1}, x_k, from the predicted results of observations on segment $[t_{k-p}, t_k]$, and from their integer powers, corresponding to the given integer d.

Then, the RSFF algorithm, under the given above formulas, will find quasi-optimal estimates $\hat{x}_k(d, Yk, p)$ of vectors x_{k-p+1}, x_k and of the estimation error covariance matrix.

The vector $\hat{x}_k(d, Y_{k,p})$ serves as a quasi-optimal solution of the nonlinear filtration problem for instant t_k and for memory p; the vector $\hat{x}_{k-p+1}(d, Y_{k,p})$ and corresponding blocks of the matrix—having been found—of estimation error covariances give a priori data for implementation of the RSFF algorithm on a following interval of time $[t_{k-p+1}, t_{k+1}]$. The outlined scheme fully defines the computational process of the quasi-optimal nonlinear filtration with finite memory p.

Let $p = 0$. Then, at instant t_{k-1}, the preceding observations $y_1, ..., y_{k-1}$ and successive steps of the RSFF algorithm defined the a priori (for instant t_k) first and second statistical moments of vector x_{k-1}, which are the estimation vector $\hat{x}_{k-1}(d, Y_{k-1,p})$ and the estimation error covariance matrix x_{k-1}, respectively.

A statistical mechanism reproduces the corresponding dispersion of vector x^*_{k-1} and builds the first and second statistical moments for components of vector $\|x_k \quad y-k, y^d_k\|^T$. Then the RSFF algorithm defines the vector of quasi-optimal estimate of the current vector x_k and the error covariance matrix of this vector estimate, presenting its dispersion after fixing a scalar of observations y_k. Then the computational process is repeated under a similar scheme. Note that at $p = 0$ and $d = 1$, the RSFF algorithm is similar to the UKF algorithm.

At $p = 0$, the vector V, input earlier at derivation of the RSFF algorithm, is composed of components of vector x_k and of a sequence of variables $H_k(x_k) + \xi_k, ..., (H_k(x_k) + \xi_k)^d$.

Let the hypotheses on which the EKF algorithm is based be true: The right parts of Eqs. (8.1) and (8.2) are differentiable. After observations are fixed, the dispersion of random vector x_{k-1} with respect to estimation vector $\hat{x}_{k-1}(d, Y_{k-1,p})$ is small (the norm of its covariance matrix x_{k-1} is small).

Linearizing components of vector $f_{k-1}(x_{k-1})$ and variables

$$H_k(x_k) + \xi_k, ..., (H_k(x_k) + \xi_k)^d$$

with respect to components of vector $x_{k-1} - \hat{x}_{k-1}(d, Y_{k-1,p})$ and calculating matrices of private derivatives of the corresponding functions, we find expressions for the expectation vector $\hat{V} = E(V)$ and covariance matrices Q_V in the form of functions from estimation vector $\hat{x}_{k-1}(d, Y_{k-1,p})$ and estimation error covariance matrices x_{k-1}.

Then, under Eqs. (8.6) and (8.7), the RSFF delivers vector $\hat{x}_k(d, Y_{k,p})$ of the quasi-optimal estimate of vector x_k and estimation error covariance matrix x_k.

Thus, at differentiable right sides of equations of models of a dynamic system and of observations, and at $p = 1, d = 1$, the EKF algorithm is coincident with the RSFF algorithm.

References

1. Boguslavskiy JA (2005) The algorithm of the organized search for identification of parameters of models of nonlinear dynamic system. Appl Math Comput 163(1):357–372
2. Leung L (1991) System identification; user theory. Physmatlit, Moscow
3. Potemkin VG (1999) System for engineering and scientific calculations MATLAB 5.x. Dialogue-Moscow Engineering and Physics Institute, Moscow
4. Liptser RS, Shiryaev AN (1974) Statistics of stochastic processes. Nauka, Moscow (Science)
5. Ribeiro MI (2004) Kalman and extended Kalman filters: concept, derivation and properties. Institute for Systems and Robotics Instituto Superior, portugal
6. Julier SJ, Uhlmann JK, Durrant-Whyte HF (1995) A new approach for filtering nonlinear systems. In: Proceedings of the 1995 American control conference, pp 1628–1632
7. Julier SJ, Uhlmann JK (1997) A new extension of the Kalman filter to nonlinear systems. In: The proceedings of the 11th international symposium on aerospace/defense sensing, simulation and controls, multi sensor fusion, tracking and resource management II, SPIE, 1997
8. Julier SJ, Durrant-Whyte HF (1995) Navigation and parameter estimation of high speed road vehicles. In: Robotics and automation conference, pp 101–105
9. Wan EA, van der Merwe R (2000) The unscented Kalman filter for nonlinear estimation. In: Proceedings of symposium 2000 on adaptive systems for signal processing, communication and control (AS-SPCC), IEEE Press
10. Bellman RE, Kalaba RE (1965) Quasilinearization and nonlinear boundary-value-problems. The Rand Corporation, New York

Chapter 6
Estimating Status Vectors from Sight Angles

Recurrent finite memory and polynomial approximation are two techniques commonly used when one tackles a nonlinear estimation problem that includes space object and air and aircraft (AC) positioning and velocity vector component evaluation from angle-of-sight data provided by either an earth-based optoelectronic system or an AC-borne sighting system.

6.1 Space Object Status Vector Evaluation

Many observational astronomy results depend on numerical data processing, including serial data from space objects available in an optical range. The angles of sight are either measured by an optical sighting system or read from photo plate prints.

Angles of sight also provide the unique information on sunlit high-orbit earth satellite vehicles (ESVs), or sputniks, especially those with a faulty AC-borne radio system. In this case, earth-based sighting systems often cannot reach the distant ESVs since their radio-frequency energy performance is insufficient due to the large distance to the ESVs. Therefore, a data processing algorithm is highly desired to yield reasonably accurate estimates of the space object status vectors.

In a special case of high-orbit ESVs, information of this sort is available from dedicated optoelectronic systems: An example is the Window optoelectronic system [1] to monitor adjacent space in the optical range, which can acquire sight data of space objects as far as 40,000 km away.

We shall demonstrate in this chapter that it is the nonlinear smoothing, filtration, and prediction algorithm described in Chap. 5 that ensures highly accurate status vector estimates. The components of the difference vector between the initial (a priori) and true status vectors comprise about 10–20 % of the true status vector component values.

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_6,

6.1.1 Equations of Motion and Observation Data Model

The equations of motion of a low-mass body in a geocentric inertial coordinate system are given by

$$dX_i/dt = V_i, \tag{1.1}$$

$$dV_i/dt = -\mu X_i/R, \tag{1.2}$$

$$R = (X_1^2 + X_2^2 + X_3^2)^{1/2},$$

where μ is the gravitational constant, and X_i, $i = 1, \ldots, 3$ are Cartesian coordinates of the body.

The angles of sight are two directional angles to determine the direction of the sight ray relative to the Earth. The sight ray here is a vector directed from the observer to the space object.

We assume that, at a time t, the sighting center is a point located at the Earth's surface, geographically referenced by longitude and latitude angles $\lambda = \omega t$ and height (altitude) h, where ω is the Earth's angular velocity. In the rotating right-handed geocentric coordinate system x_i', the sighting center coordinates are

$$X_1' = r_E \cos(\varphi) - h \sin(\varphi), \quad X_2' = 0,$$
$$X_3' = r_E \sin(\varphi) + h \cos(\varphi),$$

where r_E is the Earth's radius.

Two sight vector orientation angles represented by y_1 in the sighting center's local horizon plane and by y_2 in the local vertical plane are given by

$$y_1 = \operatorname{arctg}(b/a), \quad y_2 = \operatorname{arctg}(c/(a^2 + b^2)^{1/2})), \tag{1.3}$$

where

$$a = \cos(\lambda)\cos(\varphi)X_1 + \sin(\lambda)\cos(\varphi)X_2 + \sin(\varphi)X_3 - r_E,$$

$$b = \sin(\lambda)X_1 + \cos(\lambda)X_2,$$

$$c = -\sin(\varphi)\cos(\lambda)X_1 + \sin(\varphi)\sin(\lambda)X_2 + \sin(\varphi)X_3 = h.$$

6.1.2 Scheme of Estimator

Next, we assume that the data processing algorithm is capable of tackling the filtration problem; namely, the algorithm has to evaluate, in the inertial geocentric coordinate

system, three Cartesian coordinates of the space object under sight along with three Cartesian components of its velocity vector, at some time instants.

The feasibility of the above algorithm in principle relies on the existence of a correlation between variable angles of sight and variable Cartesian coordinates of the space object. Such a correlation is due to the law of gravity, which affords the gravity acceleration vector. The acceleration vector, comprising a known function of the Cartesian coordinates of the space object, bends the path of motion.

In the absence of the acceleration vector, any estimation of the trajectory parameters via measurement would be impossible. Indeed, the law of variation is the same for all bodies moving uniformly and rectilinearly, provided that their velocity vectors are parallel and their range to the velocity vector's moduli ratios are constants. So, in the outlined situation, the trajectory parameters are unobservable; that is, they cannot be evaluated from the data.

The estimator has to evaluate the instant status vector components complying with a set of six nonlinear differential equations (1.1), (1.2): It is inherently a nonlinear filtration problem. An observation data (OD) sequence (1.3) is input to the estimator, the OD s being nonlinear functions of the instant values X_i, $i = 1, \ldots, 3$.

The estimator, in turn, uses the recurrent algorithm described in Chap. 1 along with the polynomial approximation technique (see Chap. 2); it also uses the linear contraction operator described in Chap. 5 when considering the model nondifferentiable-function nonlinear dynamic system parameter identification problem.

Let τ be the interval between sequential observations, that is, space object readings. When the first k observations are read, the algorithm generates the cumulative OD $y_{int,1}$, $y_{int,2}$; here $y_{int,1}$ is a sum of the first k recorded values k_1, while $y_{int,2}$ is a sum of the first k recorded values k_2.

The next step is build the cumulative pairs $y_{int3}, y_{int4}, \ldots, y_{int2s-1}, y_{int2s}, \ldots$, and so on, similarly.

The sequence of these random values is a sequence of the recurrent algorithm entry values at time $1, 2, \ldots, r, \ldots$, where $T = k$. Applying the summation operator reduces the random following error effect on the estimation accuracy.

Let s be a selected integer. Divide a sequence of the paired cumulative OD into a sequence of intervals, each of them containing s OD pairs. Next, apply the following definitions.

Iteration vector 1 is an evaluation vector generated upon receiving the first pair interval at the entry of the recurrent algorithm. Iteration vector 1 is defined at time point sT once observations are made.

Iteration vector 2 is an evaluation vector generated upon receiving the $2s$th pair interval at the entry of the recurrent algorithm. Iteration vector 2 is defined at time point $2sT$ once observations are made.

Iteration vectors 3, 4, and so forth, are defined similarly.

According to Chap. 1, the construction of iteration vector 1 first requires finding the a priori data vector and matrix, namely, the first and second statistical moments of a base random vector 1 whose first six components are the components of a true (still unknown) initial space object status vector and are products of $2s$ cumulative OD to an integer power, where the sum of powers is less than the predefined integer d.

If $d = 1$, then the base vector 1 has a dimensionality of $6 + 2s$; if $d = 2$, then the base vector 1 has a dimensionality of $6 + 2s + (2s + 1)s$, and so on.

It is the a priori dispersion of the vectors $X(0)$, $V(0)$ that determines the a priori dispersion of the remaining base vector 1 components via Eqs. (1.1)–(1.3). Next, assume that

$$X(0) = X(0)_N + \delta X(0), \quad V(0) = V(0)_N + \delta V(0),$$

where $X(0)_N$, $V(0)_N$ are the nominal initial radius vector and its velocity vector, respectively, selected from a priori considerations. We suppose that $V(0)_N$ is a velocity vector for an object moving along a circular orbit of radius $|X(0)_N|$. The random vectors $X(0)$ and $V(0)$ pertain to two a priori parallelepipeds Ω_X and ω_V in R^3, respectively. The components of the random vectors $\delta X(0)$ and $\delta V(0)$ are uniformly scattered according to the inequalities

$$-\alpha|X(0) < \delta X_1, \delta X_2, \delta X_3 < \alpha|x(0)|, \tag{1.4}$$

$$-\alpha|V(0)| < \delta V_1, \delta V_2, \delta V_3 < \alpha|V(0)|, \tag{1.5}$$

where $\alpha = 0.1 \div 0.2$.

Further modeling results in initial (reference) coordinate uncertainties as large as thousands of kilometers, while the initial (reference) velocity vector uncertainties amount to hundreds of m/s. The linearized-approach extended Kalman filter (EKF) is inapplicable as a data evaluation tool under the circumstances due to its fast divergence.

The a priori first and second statistical moments of the base vector 1 components are determined by integrals over domains Ω_X and Ω_V, respectively. The integrals are approximated using a modification of the trapezoidal method, where each parallelepiped edge is divided into r equal-length parts to define $(r + 1)^3$ points evenly covering the parallelepipeds. The integrands are evaluated at these points via integration of Eqs. (1.1), (1.3) under initial conditions corresponding to the above parallelepiped edge division points.

Once the a priori statistical measures of base vector 1 are determined, $2s$ random values arrive in sequence at the entry of the recurrent algorithm described in Chap. 1, the data comprising $2s$ cumulative angles of observed sight (AOS) taken in a time interval sT. Once $2s$ calculation steps are completed, the recurrent algorithm determines iteration vector 1, the space object status vector evaluation vector $\hat{X}_{(sT)}$, $\hat{V}_{(sT)}$ at a time sT.

The estimation accuracy is characterized by the estimation error covariance matrix calculated at every step of the recurrent algorithm. The root-mean-square (RMS) deviations of both Cartesian coordinate estimation errors $\sigma_{x,i}, i = 1, \ldots, 3$ and Cartesian velocity vector components $(\sigma_{v,i}, i = 1, \ldots, 3)$, however, are determined using a Monte Carlo sampling technique with the number of samplings as high as 100,000, to provide guaranteed estimation accuracy.

The procedure to determine iteration vector 2 is similar. So we assume

$$X(0) = \hat{X}(sT) + \delta X(0), \quad V(0) = \hat{V}(st) + \delta V(0),$$

given

$$-\alpha 3\sigma_{x,i} < \delta X_i < \alpha 3\sigma_{x,i}, \tag{1.6}$$

$$-\alpha 3\sigma_{v,i} < \delta V_i < \alpha 3\sigma_{v,i}, \tag{1.7}$$

where $i = 1, \ldots, 3$.

6.1.3 Model Predictions

The a priori initial status vector components correspond to an artificial Earth satellite (AES) on a circular orbit:

$$X_1(0) = 26 \times 10^6 \text{ m} \quad X_2(0) = X_3(0) = 0, \tag{1.8}$$

$$V_1(0) = 0, \quad V_2(0) = 3.6852 \times 10^3 \text{ m/s}, \quad V_3(0) = 0. \tag{1.9}$$

The sighting system is determined by a set of equalities:

$$t = 0 : \lambda(0) = 0. \quad \varphi = 0, \quad h = 0\,(y_1(0) = y_2(0) = 0).$$

The true (actual) initial status vector components are determined by a set of equalities (1.4), (1.5) given $\alpha = 0.1$ (option 1), or $\alpha = 0.2$ (option 2), $|X(0)| = 26 \times 10^6$ m, $|V(0)| = 3.685 \times 10^3$ m/s.

In a simulation environment, the recurrent algorithm ran under the conditions of $\tau = 10s, k = 20, s = 6$. These conditions mean that each 20 pairs of AOS y_1, y_2 observed during the sequence of the time intervals T = 200 s are replaced with the pairs $y_{int,1}$, $y_{int,2}$ of the cumulative OD. Iteration vectors 1, 2, 3, … form the resulting (output) parameters of the algorithm at times $6T$, $12T$, $18T$, ….

A polynomial approximation technique has been used when simulating the evaluation vector components as $d = 1$ and $d = 2$. The case $d = 1$ means that, at every step of the recurrent algorithm, the evaluation vector components comprise a linear combination of the last 12 components of the base vector consisting of 12 cumulative OD. The case $d = 2$ means that, at every step of the recurrent algorithm, the evaluation vector components comprise a linear combination of 90 items, which are the last 90 components of the base vector and consist of the products of 12 cumulative OD raised to the power of 0, 1, 2.

In the case $d = 1$, the integrals comprising the first and second statistical moments of the base vector components were calculated with $r = 2$.

In the case $d = 2$, the integrals comprising the first and second statistical moments of the base vector components were calculated with $r = 4$.

Listed next are the $\sigma_{x,i}$ values (in m) and $\sigma_{v,i}$ values (in m/s) resulting from several steps of the iteration process. The values were determined using a Monte-Carlo sampling technique with the number of samplings (NOS) set at 100,000. Such an NOS takes about 25 s of computing time when performed on a computer with a low productivity setting. In the case $d = 2$, the computing time increases up to 20 min.

Option 1, $\alpha = 0.1$.

The a priori RMS $\sigma^a_{x,i}$ and $\sigma^a_{v,i}$ representing the space object status vector component dispersion at a time 6T are as follows:

	$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
av,	31.87×10^6	1.87×10^6	1.83×10^6	2.71×10^2	2.69×10^2	2.60×10^2.

Iteration 1: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
8.34×10^4	1.57×10^4	1.04×10^4	4.625	10.15	2.98

Iteration 2: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
8.77×10^2	2.12×10^2	3.52×10^{-1}	3.09×10^{-2}	9.08×10^{-2}	1.12×10^{-4}

Iteration 3: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
7.11	3.63	1.13×10^{-5}	3.03×10^{-4}	1.10×10^{-3}	4.70×10^{-9}

Iteration 4: $d = 1$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
1.28	9.30×10^{-1}	1.03×10^{-12}	8.65×10^{-5}	2.11×10^{-4}	4.09×10^{-16}

The simulation results show that each iteration reduces the space object status vector component dispersion RMS values by a factor of ~100. In fact, the position as well as the velocity vector components seem precisely defined by the moment when the fourth iteration is completed.

Option 2, $\alpha = 0.2$.

The a priori RMS $\sigma^a_{x,i}$ and $\sigma^a_{v,i}$ representing the space object status vector component dispersion at a time 6T are as follows:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
3.76×10^6	3.73×10^6	3.67×10^6	5.45×10^2	5.39×10^2	5.23×10^2.

Iteration 1: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
5.00×10^5	1.60×10^5	7.89×10^4	3.44×10	6.55×10	1.62×10

Iteration 2: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
6.57×10^4	3.18×10^4	9.84×10	6.26×10^{-1}	8.34	8.94×10^{-3}

Iteration 3: $d = 2$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}$	$\sigma^a_{v,3}$
2.03×10^2	1.39×10^2	1.65×10^{-3}	8.71×10^{-3}	3.25×10^{-2}	1.53×10^{-7}

Iteration 4: $d = 1$:

$\sigma^a_{x,1}$	$\sigma^a_{x,2}$	$\sigma^a_{x,3}$	$\sigma^a_{v,1}$	$\sigma^a_{v,2}\sigma^a_{v,3}$
$5,\ 804,\ 228.78 \times 10^{-9}$	4.43×10^{-4}	9.40×10^{-4}	3.88×10^{-12}	

The data show that a twofold increase in the a priori dispersion resulted in slightly increased RMS values of the space object status vector component estimation error following the four iterations.

6.2 Estimation of the Air- and Space-Craft Status Vector, Local Vertical Orientation Angles, and AC-Borne Sighting System Adjustment

A polynomial approximation is the technique commonly used to solve a set of nonlinear algebraic equations providing the mathematical formulation of a newly developed autonomous (unaided) AC navigation method.

One possible situation in AC control practice, which is equally related to both manned and unmanned vehicles, requires the AC navigation (i.e., instantaneous

positioning and velocity vector determination in the frame fixed relative to the Earth) to be performed autonomously, as self-contained activities, without the use of radio signals from the Earth or from space. Such a situation may arise, for instance, when the AC has to force-land in an arctic environment or in a jungle, when performing urgent airlift freight delivery operations in unprepared terrain, or when the AC navigation system is to be adjusted in the event of a GPS receiver malfunction.

Note that any emergency/urgent landing or airlift delivery requires knowing the motion variables (trajectory parameters) of the AC relative to the terrain for which geographical positions are unknown. In such cases, the AC trajectory parameters received from the GPS are of no use to navigate the AC when the terrain elevation and/or clearance is unknown.

We shall treat a special case when the external data containing information on the AC motion relative to the Earth are instantaneous AOS values related to the fixed point on the top of the ground and are available from the AC-borne sighting system. This point appears to be "shining" in some wavelength range of the electromagnetic spectrum. The point may comprise a reflector disposed from the AC, to reflect (during the repeated target runs) the continuous laser radiation or millimeter-wave band radiation generated by the AC-borne sighting system. Alternatively, the "shining" spots may be some distinguished points, either natural or artificial, on the top of the Earth.

It must be emphasized that no information is required about the geographical position and height of the "shining point" (SP).

Let's introduce the AC-borne (Cartesian) frame (ACBF) fixed relative to the AC and the sight ray, a vector with its origin at the center of the optical sighting system and its endpoint at the SP. It is the center of the sighting system that serves as the ACBF origin. Two angles of sight measured by precise digital sensors determine the angular orientation of the sight ray in the ACBF. The instant AOS values provide the minimum adequate information to perform high-precision autonomous navigation of the AC relative to the Earth.

The presented modification of the AC navigation method does not require measuring the present distance on the SP (see [2] for a modification including the present distance measurement), thereby dramatically simplifying the design and scope of the AC-borne sighting system. The algorithm of the method uses polynomial approximation of the root vector, the vector representing a solution of the model nonlinear algebraic system to describe the autonomous navigation parameters.

The condition-specific model study of the AC flight has demonstrated that the information on the several fixed-point reference AOS values acquired for 15 s is adequate to determine the navigation parameters with minute uncertainties. Unexpectedly, a highly accurate estimation of the local vertical orientation angles has been achieved.

The algorithm may illustrate a computer-aided technology with a potential for getting the desired autonomous navigation "at a low price," that is, having used the ASC-borne sighting system with no distance-measuring equipment.

6.2.1 Primary Navigation Errors and Formulation of the Problem

The ASC-borne navigation equipment is assumed to include three angular velocity sensors to measure the ASC velocity vector projections on the ASCBF axes, and three accelerometers, to measure the similar projections of the ASC nongravitational acceleration vector. The projections are input at a high rate into the flight navigational computer.

Let's introduce the accompanying frame of reference (AFR), with the AFR origin aligning with the origin of the ACBF, the Z-axis being directed as the local gravity vector, and the X, Y pair of axes belonging to the local horizontal plane, which is in turn perpendicular to the local gravity vector. (Conventionally, the Z-axis is directed as the local weight vector; the distinction between definitions, however, is not critical in this case.) The X, Y pair of axes may be arbitrarily oriented within the local horizontal plane. For example, the X-axis may be directed meridionally toward the North Pole, and the Y-axis may be directed leftward to line up with the geographic parallel. In that case, the AFR will match the Geographical Reference System.

Three Euler angles (ψ, θ, γ) are calculated for the AC mission initiation time using a conventional initial alignment technique. The angles determine the initial ACBF orientation relative to the calculated AFR. The last frame does not match the "true" AFR due to instrumental errors, so the initial Euler angles define the ACBF orientation relative to the calculated AFR, with some minute uncertainties.

Suppose that, once the AC mission has started, the flight navigational computer integrates a set of kinematic and dynamic differential equations, where the right parts depend on the measured components of the AC angular velocity vector and the AC nongravitational acceleration vector. The integration procedure results in the instantaneous AC position data, AC velocity vector components, and ACBF orientation angles relative to the calculated AFR.

This frame of reference does not match the current "true" AFR; this is due to the initial alignment uncertainties, measurement errors related to the components of interest, and calculation errors. Then the calculated coordinates (position data), velocity vector components, and Euler angles deviate from the true values by errors that accumulate as the mission time elapses.

Assume next that the AC-borne sighting system can detect the SP at a distance of 10–15 km; the AC alters the course and flies toward the SP—this moment is referred to as the initial sighting time, $t = 0$. Then, during the time interval T, the sighting is performed followed by data acquisition and nonlinear estimation of navigation parameters. Once the elapsed time T is expired, the sighting is terminated and the algorithm yields the navigation parameters. These parameters are used for the initial conditions to support the subsequent operation of the independent inertial navigation system. T has an order of magnitude of about dozens of seconds.

At the time point $t = 0$, introduce the inertial (not rotary) orthogonal frame of reference (IFR) with the axes X, Y, Z. At that moment, the IFR is the same as the

calculated AFR. The XYZ-axis is thus directed as the calculated gravity vector, while the IFR X, Y-plane is the same as the local horizontal plane calculated at $t = 0$.

Due to the calculated AFR alignment errors, the ZX- and ZY-projections of the IFR will make small angles, δ_ϑ and δ_γ, respectively, with the Z-axis. The gravity vector projections on the X-, Y-, Z- axes are then equal to

$$-g\delta_\vartheta, \quad -g\delta_\gamma, \quad -g,$$

respectively, where g is the gravity force acceleration.

The gravity force field is plane-parallel for several dozens of seconds while the ASC flies toward the SP, so the δ_ϑ, δ_γ uncertainties may be considered constant. These uncertainties could be reasonably termed the local vertical alignment errors.

Now direct the IFR X-axis to the point within the calculated horizontal plane XY where the SP is projected. Then the calculated vertical IFR planes will be determined by the pairs of axes X and Z, Y and Z.

Next we define the sight ray orientation within the IFR using the angles f_v and f_h. As $t > 0$, f_v is an angle between the X-axis and the ZX-projection of the sight ray, where ZX is the calculated vertical plane while f_h is an angle between the X-axis and the XY-projection of the sight ray, and XY is the calculated horizontal plane. So, at $t = 0$, the sight ray belongs to the calculated vertical plane ZX and makes an angle $f_v(0)$ with the X-axis; at the same time, $f_h(0) = 0$.

While the AC moves toward the SP, the onboard computer determines the angles f_v, f_h as the functions of the instantaneous (actual) sight ray orientation angles relative to the ACBF as well as the functions of the calculated instantaneous Euler angles that characterize the ACBF orientation relative to the IFR. The last values are determined via integration of the ACBF angular velocity vector components with the initial conditions known with some uncertainties due to the Euler angle calculation errors at $t = 0$, as discussed above. Furthermore, there are sighting system optical assembly adjustment errors as well as the errors related to the setting/alignment of digital sensors intended to read the sight ray orientation angle relative to the ACBF. When combined, all types of errors will result in the calculated sight angles f_v, f_h deviating from the true values by the uncertainties δf_v and δf_h, which may be considered constant for the time the SP sighting is being performed.

With $t > 0$, let a_x, a_y, a_v be the calculated projections of the AC nongravitational acceleration vector on the X, Y, Z IFR axes, provided that the ACBF axial projections of the vector are measured by three accelerators. Let δa_x, δa_y, δa_v denote the uncertainties in a_x, a_y, a_v, respectively. These uncertainties are mainly due to the Euler angle calculation errors, where the Euler angles determine the ACBF orientation relative to the IFR. Each uncertainty is thus approximately a linear combination (with small coefficients) of the other k_{ij} axial projections of the nongravitational acceleration vector. For example,

$$\delta_{a_x} = k_{1,2}a_y + k_{1,3}a_v.$$

We assume here that the measured AC nongravitational acceleration vector consists of the first acceleration vector introduced to counterbalance the gravity force acceleration and of the second acceleration vector introduced to allow for bending the AC pathway. Furthermore, let $a_v = g + a_v^{'}$ and let $a_x, a_y, a_v^{'}$ be the components of the time-dependent nongravitational acceleration vector introduced to allow for bending the AC pathway.

With $t = 0$, let H be the measured SP altitude referred to the IFR (i.e., the Z-projection of the ST) and let V be the measured value of the ASC velocity vector. The measurements are performed by the AC-borne sensors with error $\delta(H), \delta(V)$, respectively. The radar altimeter and/or barometric altimeter and air speed meters will serve as the appropriate candidate sensors.

With the SP held at a horizontal plane XY and $t = 0$, the actual SP altitude equals the AC altitude referred to the AFR and is close to the value measured by the altimeters. In hilly country, however, an appreciable measurement error $\delta(H)$ may arise. Similarly, while the $\delta(V)$ error is generally small to negligible under the still-air conditions due to the minute airspeed meter uncertainties, the error may approach the wind speed itself under windy conditions. With $t = 0$, let $\delta(V)$ be an angle between the X-axis and the XZ-projection of the velocity vector (XZ is a vertical plane as above), and let h be an angle between the X-axis and the XY-projection of the velocity vector (XY is a horizontal plane as above).

From now on, elsewhere we shall take for the actual parameter value its measured or calculated value plus the measurement or calculation error.

For the purposes of the autonomous AC navigation (see some considerations above), it is adequate to know (i) the current AC position relative to the SP and referred to the IFR, (ii) the current velocity vector components, and (iii) the estimated vertical misalignment errors $\delta_\vartheta, \delta_\gamma$. These estimates are required within the SP local area context to recalculate the current (instantaneous) coordinates and velocity vector components previously referred to the IFR into the same navigation parameters now referred to the AFR.

The problem so formulated is to be solved by the nonlinear estimation algorithm with due account for the existence of the error types listed above.

6.2.2 *Navigation Parameters: The Nonlinear Estimation Problem*

Denote by $H(t)$, $A_x(t)$, $A_y(t)$, $V_h(t)$, $V_x(t)$, $V_z(t)$ the AC's navigation parameters referred to the IFR; the parameters characterize the current ASC position relative to the SP and the current AC velocity vector components, respectively. The parameters are time-dependent functions and can be represented by the equations

$$H(t) = H + \delta(H) + (V + \delta(V)) \sin(\theta_v)t - (g/2)t^2 + J_H(t), \qquad (2.1)$$

$$A_x(t) = ((H + \delta(H))/\mathrm{tg}(f_v + \delta_{f_v}))) + (V + \delta(V))\cos(\theta_\theta)t + (g/2)(\delta_v)t^2 + J_{A_x}(t), \tag{2.2}$$

$$A_y(t) = (V + \delta(V))\cos(\theta_h)t + (g/2)\sin(\delta_\gamma)t^2 + J_{A_y}(t), \tag{2.3}$$

where

$$J_H(t) = \int_0^t \left(\int_0^\tau (a_\vartheta + \delta_{a_\vartheta}) \right) dt,$$

$$J_{A_x}(t) = \int_0^t \left(\int_0^\tau (a_x + \delta_{a_x}) \right) dt,$$

$$J_{A_y}(t) = \int_0^t \left(\int_0^\tau (a_y + \delta_{a_y}) \right) dt;$$

differentiating (2.1)–(2.3), we obtain the expressions for $V_h(t)$, $V_x(t)$, and $V_z(t)$.

The AOS $f_\vartheta(t)$ and $f_h(t)$ are measured as functions of time and deviate from the "true" AOS values by the constant uncertainties δ_{f_ϑ} and f_h, respectively. The measurement results, $Y_\vartheta(t)$ and $Y_h(t)$, are related to the navigation parameters by the formulas

$$Y_\vartheta(t) = f_\vartheta(t) = \mathrm{arctg}(H(t)/A_x(t) + \delta_{f_\vartheta}, \tag{2.4}$$

$$Y_h(t) = f_h(t) = \mathrm{arctg}(H(t)/A_y(t) + \delta_{f_h}. \tag{2.5}$$

From (2.1)–(2.5) it follows that the measurement results depend nonlinearly on the eight unknown parameters, which include δ_{f_ϑ}, δ_{f_h}, $\delta(h)$, $\delta(\vartheta)$, $\delta(v)$, and $\delta(h)$. The error-controlling parameters δ_{a_x}, δ_{a_y}, and δ_{a_v} are omitted in the list of unknown parameters.

The nonlinear estimator problem is to evaluate the eight unknown parameters using the sequence of the measured AOS values stored in the flight navigational computer memory during flight time. The estimation results then have to be substituted into (2.1)–(2.3), whereupon the autonomous navigation parameters will be determined accordingly.

Theoretically, eight taken measurements will be enough. Then the unknown parameters will become the roots of a set of eight nonlinear algebraic equations. The equations are solved numerically using the polynomial approximation algorithm.

6.2.3 Calculation Model and Estimation Results

This section describes the model and the results obtained when considering the evaluation problem containing five unknown parameters $\delta_{f_v}, \delta(H), \delta(V), \delta_\vartheta, \theta_v$. The remaining three parameters—$\delta_{f_h}, \delta_\gamma, \theta_h$—are evaluated in a similar way.

Suppose that an a priori existence domain of five unknown parameters comprising the roots of a set of the five nonlinear algebraic equations is represented by an a priori dispersion parallelepiped in R^5 centered on the origin. Judgment-based data suggest that the following parameters of the a priori parallelepiped are acceptable for the purposes of the nonlinear estimation algorithm accuracy analysis:

$$|\delta_{f_\vartheta}| < 0.002, \quad |\delta(H)| < 100\,\text{m} \quad |\delta(V)| < 50\,\text{m/s},$$

$$|\delta_\vartheta| < 0.02, \quad |\theta_\vartheta| < 0.05.$$

These data slightly exceed the likely practical error threshold.

The evaluation scheme involves two iteration steps.

In iteration step 1, the data acquisition period T is divided into four equal parts (quartered). By $t_i, i = 1, \ldots, 4$, denote the respective division time points, and save the measurements $Y_\vartheta(t_i)$. The algorithm has to solve a set of four nonlinear algebraic equations:

$$Y_\vartheta(t_i) = \text{arctg}(H(t_i)/A_x(t_i)) + \delta_{f_\vartheta}, \quad i = 1, \ldots, 4,$$

on the assumption that $\delta(H), \delta(V), \delta_\vartheta$, and f_v are roots of this set of equations. The variable δ_{f_ϑ} affords a nuisance parameter and needs no estimation during iteration step 1. Even so, it was demonstrated by calculations that estimation error variances are small for values to be estimated during iteration step 1.

Replace the four unknown parameter values in formula (2.4) with the estimates obtained with estimation error values added to them. The result is that the respective four edges of the new a priori parallelepiped will be dramatically shrunk, to determine the reduced possible scattering of the unknown parameters upon completion of iteration step 1. The new edge lengths are equal to twice the square roots of the estimation error variances calculated in iteration step 1. The variances are determined according to the corresponding formulas of Chap. 4 or, alternatively, using a Monte Carlo sampling technique within the math simulation model of the autonomous navigation algorithm.

In iteration step 2, the data acquisition period T is divided into three equal parts (trisected). By $t_i, i = 1, \ldots, 3$, denote the respective division time points, and save the measurements $Y_\vartheta(t_i)$. The algorithm has to solve a set of three nonlinear algebraic equations:

$$Y_\vartheta(t_i) = \text{arctg}(H(t_i)/A_x(t_i) + \delta_{f_\vartheta}, \quad i = 1, \ldots, 3,$$

on the assumption that $\delta(H)$, $\delta(V)$, θv, and δ_ϑ are roots of this set of equations. The estimation errors (which arose in iteration step 1) related to the variables θ_ϑ and δ_ϑ afford nuisance parameters and need no estimation in iteration step 2.

After considering Eqs. (2.1)–(2.3) and the simulation data, one sees that the parameters to be evaluated are poorly observable whenever the AC nongravitational acceleration vector is lacking (vanishes), so there is no way of bending the pathway at the segment while the nonlinear estimation data are acquired. Hence, we suppose when simulating the algorithm's operation that, from the time $t = 0$ on the above nongravitational acceleration vector is perpendicular to the current AC velocity vector and rotates this vector with angular velocity during the time interval t. Denote the value of the vector by ag. Then the following equations hold:

$$J_H(t) = (g/2)t^2 + ag(\cos(\theta_v) - \cos(\theta_v + \omega t)/\omega^2) - (\sin(\theta_v)t/\omega)) + \int_0^t \left(\int_0^\tau \delta_{a_v} d\tau \right) dt,$$

$$J_{A_x}(t) = (g/2)t^2 + ag(-\sin(\theta_v) + \sin(\theta_v + \omega t)/\omega^2) - (\cos(\theta_v)t/\omega)) + \int_0^t \left(\int_0^\tau \delta_{a_x} d\tau \right) dt,$$

where $\omega = ag/V$.

In iteration step 1, the four-dimensional root vector has been approximated by a formal vector power series section, the series being composed of 14 vector items (addends), as specified by the polynomial approximation technique. Each item is proportional to the product of the four AOS measurements to an integer power f_ϑ, each AOS containing δ_{f_ϑ}, the random (still constant) adjustment errors, and Euler angle calculation errors. The sum of powers is 2 or less.

In iteration step 2, the three-dimensional root vector has been approximated by a formal vector power series section, the series being composed of 19 vector items (addends). Each item is proportional to the product of the three AOS measurements to an integer power f_ϑ, each AOS containing δ_ϑ, the random (still constant) adjustment errors, and Euler angle calculation errors. The sum of powers is 3 or less.

A modified trapezoidal method was used to evaluate both four- and three-dimensional integrals, representing, according to the polynomial approximation technique, the first and second statistical moments. The integrands were calculated at the division points selected to split the edges of the a priori parallelepiped into 10 equal-length parts.

The accuracy of estimation is characterized by the RMS values $\sigma(\delta(H))$, $\sigma(\delta(V))$, $\sigma(\theta_v)$, $\sigma(\delta_{f_v})$, and $\sigma(\delta_\vartheta)$ determined using a Monte Carlo sampling technique with the NOS to be 1,000 (in iteration steps 1 and 2). During each sampling, uniformly distributed random values were generated by the random number generator implemented into the Delphi 7 software package; the values fell within the limits dictated by the edge lengths of the parameter-specific a priori parallelepiped.

The simulation was performed subject to the conditions $T = 15\,\text{s}$, at the time point

$$t = 0,\ H = 1500\,\text{m},\ A_x = 15000\,\text{m},\ V = 150\,\text{m/s},$$

with the options of $ag = 1\,\mathrm{m/s^2}$ and $ag = 3\mathrm{m/s^2}$.

The resulting estimation error RMS values are as follows:

Option 1, $ag = 1\ \mathrm{m/(s^2)}$: Summarized errors after two iterations:

$\sigma(\delta(H))$, m	$\sigma(\delta(V))$, m/s	$\sigma(\theta_\vartheta), rad$	$\sigma(\delta_{f_v}), rad$	$\sigma(\delta_{f_v}), rad$
6.29	0.57	0.0029	0.001	0.0000003

Option 2, $ag = 3\mathrm{m/s^2}$: Summarized errors after two iterations:

$\sigma(\delta(H))$, m	$\sigma(\delta(V))$, m/s	$\sigma(\theta_\vartheta), rad$	$\sigma(\delta_{f_\vartheta}), rad$	$\sigma(\delta_{f_v}), rad$
2.47	0.6	0.0029	0.001	0.0000003

Note that simulating 1,000 steps of the nonlinear estimation procedure takes about 5 min of computing time when performed at the computer with low productivity.

It follows from the simulation results that the navigation parameters could be determined with a sufficiently high accuracy using $ag = 1 \div 3\ \ \mathrm{m/(s^2)}$.

Another point to notice is the highly accurate local vertical calculation error estimates: A δ_ϑ accuracy level is attainable within a 15-s measurement period. None of the currently available navigation-based real-time correction methods can maintain such a level of accuracy. Random errors such as the discrete stochastic ("white") noise inherently present in the sight-ray ST tracking error angles are easy to handle by modifying the nonlinear estimation algorithm according to the results of Chap. 4. Their smoothing can be performed, for example, by feeding the algorithm with RMS angles of sight measured with a frequency of $10 \div 100$ Hz at the selected points within the time interval T.

References

1. Lantratov K (2002) Optical-electronic complex of outer space control "OKNO." Cosmonautics Bull 3
2. Boguslavskiy JA, Egorova AV, Obrosov KV (1998) Aircraft instrument complex for autonomous information support of landing. Bull Russ Acad Sci Theor. Control Syst 2

Chapter 7
Estimating the Parameters of Stochastic Models

Division 1: Hidden Markov Models

The polynomial approximation method is used to evaluate the experimental data on the transition and emission probability matrices and the intensities of the matrix model of the Markov process with a finite number of states and with continuous time.

7.1 Introduction

The Baum–Welch algorithm is usually recommended to estimate the parameters of a hidden Markov model (HMM). However, it is not reliable, as it supplies the estimation vector, which corresponds to some nearness of a local maximum likelihood. This chapter offers a new algorithm-estimator to estimate the HMM parameters; using the polynomial approximation technique, the Bayes approach, and information compression, it builds approximations to a vector of the conditional expectation.

It considers sample observable symbols and for $s = 5$ hidden states generates multiple sequences of 15,000 observable characters. The algorithm-estimator calculates the estimates of the unknown 25 transition probabilities and 20 emission probabilities via third-order ($d = 3$) polynomial approximations. We have seen that from 45 relative errors of an estimation, almost all are less than 0.1. The input to the algorithm-estimator was 16 experimental frequencies received by compression of the primary information.

In this chapter, we consider the problem of estimating the parameters of the HMM when unobservable states and observable numbers belong to sets from a finite number of elements. The statistical design of an HMM has the following scheme:

Let $\{x_t\}$ be an s-state Markov chain, generated by an $s \times s$ stochastic matrix $A = \{a(i, j)\}, t = 1, \ldots, T : P(x_t = j|x_{t-1} = i) = a(i, j), i, j = 1, \ldots, s$. Let $\{y_t\}$ be a probabilistic function of $\{x_t\}$ that defines the emission probability via an $s \times r$ matrix $E = \{e(j, k)\}, k = 1, \ldots, r$:

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_7,

$$P(y_t = k | x_t = j) = e(j, k),$$

where each row sums to 1.

We suppose that at some whole $1 \le i, j \le s$, $1 \le k \le r$ values, $a(i, j), e(j, k)$ can be equal to zero. However, the matrix A should be ergodic. Next, we suppose that the $x_1, x_2, \ldots, x_T$ states are not observed, and the $y_1, y_2, \ldots, y_T$ symbols form an observable sequence S_T. The described statistical construction is HMM, with elements of the A and E matrices as parameters. The number of independent parameters is $s(s-1) + s(r-1)$.

The HMM is applied in the analysis and prediction of experimental data sequences. Examples of such sequences include a sequence of nucleotides in DNA analysis and a sequence of phonemes in a problem of speech recognition [1, 2].

The goal of this chapter's Sect. 7.1 is a solution of an inverse problem for the HMM: to create a Bayesian algorithm-estimator of elements of the matrices A and E. An input of the algorithm-estimator is the observable sequence S_T. These estimations are asymptotic optimal in the mean square.

Publication [3] presents *the Baum–Welch algorithm* for HMM parameter estimation. This algorithm involves an iterative sequence of *estimation and maximization (EM)* steps. At any estimation step, the forward and backward algorithms [1, 4] use the HMM parameters found during the previous step and define the likelihood for S_T. The estimates from these algorithms are used in the current maximization step in order to determine new values of the HMM parameters, which enlarge the value of the likelihood function for S_T.

The algorithm ensures the theoretical convergence of the calculation process to the local maximum likelihood function and determines elements of the A and E matrices corresponding to the maximum likelihood principle.

The disadvantages of the Baum–Welch algorithm can be described as follows:

1. "The Baum–Welch algorithm is guaranteed to find local maximum on the probability 'surface' but there is no guarantee that this local optimum is anywhere near the global optimum nor a biologically reasonable solution" [1, p. 154].
   This statement confirms an example (see Appendix) corresponding to a situation at $s = 2, r = 2$. A direct computational search determined a sequence of local maxima of the likelihood of the HMM in a neighborhood of true (known) matrices of the transition and emission probabilities. These local maxima did not coincide with elements of the true matrices and omitted from them expansion of the area of the computational search.
2. The Baum–Welch algorithm does not calculate a measure of the exactitude of an estimation (for example, an estimation error covariance matrix) on each pitch of iterations.
   All known algorithms of an applied statistic that use a maximum likelihood principle have the disadvantages just enumerated. References [1] and [4] do not contain examples of the application of the Baum–Welch algorithm in biological sequence analysis. Only the example for $s = 2, r = 6$ is demonstrated [1] (the example is based on a criminal situation in a casino).

In the task of biological sequence analysis, there are situations when an investigator possesses a priori data about the matrices A and E. Let an a priori HMM be given for some family of microorganisms. This HMM generates observable symbols of sequences; when aligned, the sequences of the nucleotides of the family of microorganisms are statistically similar.

We accept that [1, 4]

1. A new microorganism has been discovered whose structure differs from the structures of the mentioned family.
2. The aligned sequence of nucleotides of the new microorganism has been discovered.

There is interest in defining the parameters of a new a posteriori HMM that could generate the mentioned new sequence of nucleotides.

This example illustrates the expediency of creating a Bayesian algorithm-estimator of the HMM's parameters. This Bayesian algorithm-estimator should not contain the basic deficiencies of the Baum–Welch algorithm.

A review of some problems using the Bayesian algorithm for estimating parameters of the HMM has been presented in Chaps. 1 and 2 of this book. One possible solution of the problem with a Bayesian algorithm-estimator is stated in Chaps. 1 and 2 for an arbitrary dynamic system. In this case, the Bayesian algorithm-estimator is constructed via a multipolynomial series composed of vectorial linear combinations of products of powers of observations. These are empirical frequencies—statistics from the elements of the observable sequence. These vectorial series represent the optimum in the mean square approximations of the conditional expectation and are realized using the *multipolynomial approximation algorithm* (algorithm-estimator used in Chaps. 1 and 2).

The algorithm-estimator does not require the determination of the global maximum or the local maximum for the likelihood function. The algorithm-estimator calculates the current matrix's correlations of estimation errors during all calculation steps. This circumstance provides an analysis of a possibility of iterations. They are possible if $D(pri) > D(pos)$, where $D(\ldots)$ are the a priori and a posteriori variances of estimation errors.

From the many-dimensional analog of Weierstrass's theorem (a corollary of Stone's theorem [5]), it follows that with magnification of a power of a multipolynomial series, they uniformly converge to the conditional expectation.

This chapter describes the technology involved in using the algorithm-estimator to estimate the HMM parameters. The multipolynomial series is composed of vectorial linear combinations of products of powers of empirical frequencies, statistics from the elements of the observable sequence. The proof of the law of large numbers has been established. The law states that if the length of the sequence S_T is increased, then the random empirical frequencies converge to values proportional to the likelihood of empirical frequencies.

The algorithm builds the matrices $\widehat{A}$, $\widehat{E}$, which serve as estimates of the unknown A and E matrices and also are functions of the empirical frequencies. The matrices

$\widehat{A}, \widehat{E}$ are approximate solutions of a system of the algebraic equations whose right members are the likelihood of the empirical frequencies.

To realize the Bayesian concept, we suppose the matrices A and E have random elements. We present an algorithmic mechanism to generate these elements. The mechanism supposes a simple computational process; it determines a priori limits, in which the unknown elements of the transition and emission matrices dissipate relative to the given a priori (nominal) probabilities. The given nominal matrices and the given limits of a dispersion should be interpreted as a priori data supplying the initial approximate information about the estimated HMM.

We verify the algorithm by estimating the HMM parameters at $s = 5, r = 4$, where it is necessary to estimate 25 random transition and 20 random emission probabilities. In this case, each vector of random parameters has dimension 45×1 and determines an element of some random HMM. The algorithm-estimator acts on random observable sequences, which are generated by the elements of the aforementioned set of the HMM.

7.2 The Basic Structure of the Algorithm-Estimator

We shall briefly present the structure of the algorithm-estimator, which can be used to estimate the vector θ parameters of an arbitrary parameterized dynamic system, and not just the HMM. Following the Bayesian approach, we suppose that the θ vector is random, with some distribution that can be generated by a computer program.

We describe the basic structure of the algorithm-estimator in connection with the general problem of estimating the unknown θ, whose dimension is $q \times 1$, if there is an observable sequence S_T, which depends on θ:$S_T = S_T(\theta)$. The θ will be estimated upon fixing the sampling of S_T.

The algorithm-estimator's structure is described next.

Step 1 By constructing the Y_N vector *statistics* with $y_1, \ldots, y_N$ components, we compress the information contained in S_T.

Step 2 For given positive integers d and N, denote by $a_1, \ldots, a_N$ any system of nonnegative integers such that not all of them equal zero, and they satisfy the inequality $a_1 + \ldots + a_N \leq d$. Denote by $m(d, N)$ the number of all such systems. It can be shown by induction that $m(d, N)$ is determined by the recursion relation

$$m(d, N) = m(d-1, N) + (N + d - 1) \cdots N/d!, m(1, N) = N.$$

We introduce a vector $W_N(d)$ of dimension $m(d, N) \times 1$ with components $w_1, \ldots, w_{m(d,N)}$ such that each component is equal to $y_1^{a_1} \cdots y_N^{a_N}$ for some $a_1, \ldots, a_N$.

Next, we define a basic vector $V(d, N)$ of dimension $(q + m(d, N)) \times 1$, $V(d, N) = ||\theta \quad W_N(d)||^T$.

Step 3 Algorithmic maintenance will generate a set of the random θ parameters and the random observable sequences S_T. As a result, the computer memory will contain the set of realizations of the random basic vectors $V(d, N)$ sufficient for calculating—via the Monte Carlo method—the statistical characteristics of the basic vectors $V(d, N)$, namely, the expectation vector $\overline{V}(d, N) = E(V(d, N))$ and the covariance matrix $C_V(d, N) = E((V(d, N) - E(V(d, N)))(V(d, N) - E(V(d, N)))^T)$.

Step 4 For given d, N and the Y_N vector statistics, we introduce a vector $\hat{E}(\theta|Y_N)(W_N(d))$ that is regarded as a solution of the estimation problem. We shall construct the set of vectors of linear combinations of the components of the vector $W_N(d)$. Then the vector $\hat{E}(\theta|Y_N)(W_N(d))$ is an element of this set and is the mean-square-optimal (on the set of vectors of linear combinations!) estimate of the vector θ.

The $\hat{E}(\theta|Y_N)(W_N(d))$ vector can be presented as

$$\hat{E}(\theta|Y_N)(W_N(d)) = \sum_{a_1+\ldots+a_N \le d} \lambda(a_1, \ldots, a_N) y_1^{a_1} \cdots y_N^{a_N}, \tag{2.1}$$

where the $\lambda(a_1, \ldots, a_N)$ vectors are some weight vectorial factors.

The $\overline{V}(d, N)$ vector and the $C_V(d, N)$ matrix represent the initial data for the process of recurrent calculations of the weight vectorial factors $\lambda(a_1, \ldots, a_N)$ that consists of $m(d, N)$ steps. In the last step in Eq. (2.1), the $\lambda(a_1, \ldots, a_N)$ vectors are determined, together with the $C(d, N)$ matrix, which is the covariance matrix for the error of estimating the θ) vector via the $\hat{E}(\theta|Y_N)(W_N(d))$ estimation vector.

The formulas for the process of recurrent calculations are found using a principle of a decomposition of observations (see Chaps. 1 and 2). This formula operates correctly if the moving matrix of the estimation errors is singular.

Let $E(\theta|Y_N)|$ be a vector of conditional expectations. Using the multidimensional analog of Weierstrass's theorem (a corollary of Stone's theorem [5]), we prove that if the integer d is increased, the error estimation vector $|\hat{E}(\theta|Y_N)(W_N(d)) - E(\theta|Y_N)|$ uniformly tends to zero in some domain.

If the number $m(d, N)$ of the approximating polynomial series is not small, a large number of calculations has to be performed (after the initial choice of the integers q, N, d) to determine the vector coefficients $\lambda(a_1, \ldots, a_N)$. However, after these coefficients have been calculated and stored in computer memory, the estimation vectors are determined by simple calculations on the basis of Eq. (2.1) after any new observations of the Y_N vector statistics.

7.3 Statistics and Empirical Frequencies

We achieve information compression and suppose that the statistics—some functions from sample S_T—are input in the algorithm-estimator.

We let b be a positive integer. We split the sequence S_T into a sequence $S_T(b)$ of blocks, each of which contains b successive observable symbols.

We designate various (not coincident) blocks as $u_1, u_2, \ldots, u_L$, $L \le r^b$. The u_l is the block from b consecutive observable $y_1(l), \ldots, y_b(l)$.

The given sequence $S_T(b)$ is composed of the blocks $u_1, u_2, \ldots, u_L$, $L \le r^b$. If all the $e(j, k)$ emission probabilities are positive and the length of the sequence S_T is also large enough, then $L \to r^b$, $T \to \infty$ will hold. Additionally, $T(b) = [T/b]$, where $T(b)$ designates a length of $S_T(b)$. Furthermore, we believe that the transition and emission probabilities are all positive.

The block u_l is discovered $f(l, b, T)$ times in the sequence $S_T(b)$. We name the random integers $f(l, b, T)$ *empirical frequencies*. The vector composed of empirical frequencies $f(l, b, T)$ is the vector of the statistics ($N = L$) for the problem of estimating the HMM's parameters.

As the input for the algorithm-estimator, after forming the $S_T(b)$ sequence, we enter $f(l, b, T)$ random empirical frequencies, where the l values are integers $1, \ldots, L$.

7.4 The Law of Large Numbers

Denote by $L(l, b, A, E, T)$ the likelihood of the following event: The block of the mentioned $S_T(b)$ random subsequence is $u_l{:}y_1(l), \ldots, y_b(l)$.

Let $\varepsilon > 0$, and let $Q(i, j, \varepsilon)$ be the event that the following inequalities are satisfied:

$$|(f(l, b, T) - L(l, b, A, E, T))/T(b)| \ge \varepsilon, \tag{4.1}$$

where $l = i, i + 1, \ldots, j - 1, j$.

We will prove that for an ergodic HMM with a finite number of states,

$$P_r(Q(1, r^b, \varepsilon)) \to 0, T \to \infty. \tag{4.2}$$

This statement may be referred to as *the law of large numbers* for relative empirical frequencies. But

$$P_r(Q(1, r^b, \varepsilon)) \le \sum_{l=1}^{r^b} P_r(Q(l, l, \varepsilon).$$

To prove (4.2), it is enough to show that for any integers $l = 1, \ldots, r^b$, the statement

$$P_r(|(f(l, b, T) - L(l, b, A, E, T))/T(b)| > \varepsilon) \to 0, T \to \infty,$$

is correct and

$$L(l, b, A, E, T) \simeq f(l, b, T)/T(b), l = 1, \ldots, r^b. \tag{4.3}$$

However, the values $L(l, b, A, E, T)$ are some function by the unknown A and E matrices. Then Eq. (4.3) is a system of algebraic equations concerning elements of the A, E matrices and with right members discovered from the observable sequence $S_T(b)$. The Bayesian estimator actually solves system (4.3) approximately.

We give a simple example: the behavior of the relations $f(l, b, T)/T(b)$ for different values of l. For example, let $s = r = 2, b = 3$, and

$$\pi_1 = (1 - (a(1, 2) + a(2, 2)))/(a(1, 1) + a(2, 1) - a(1, 2) - a(2, 2)),$$

$$\pi_2 = 1 - \pi_1,$$

where π_1, π_2 are the stationary probabilities of nonobservable states.

The random blocks $u_1, \ldots, u_8$ look like u_l : 111, 112, 121, 122, 211, 212, 221, 222. Then Eq. (4.3) looks like

$$\pi_1(e(1, 1)a(1, 1)e(1, 1)a(1, 1)e(1, 1) + e(1, 1)a(1, 1)e(1, 1)a(1, 2)e(2, 1)$$

$$+e(1, 1)a(1, 2)e(2, 1)a(2, 1)e(1, 1) + e(1, 1)a(1, 2)e(2, 1)a(2, 2)e(2, 1))+$$

$$\pi_2(e(2, 1)a(2, 1)e(1, 1)a(1, 1)e(1, 1) + e(2, 1)a(2, 1)e(1, 1)a(1, 2)e(2, 1)+$$

$$e(2, 1)a(2, 2)e(2, 1)a(2, 1)e(1, 1) + e(2, 1)a(2, 2)e(2, 1)a(2, 2)e(2, 1))$$
$$\simeq f(1, b, T)/T(b);$$

$$\pi_1(e(1, 1)a(1, 1)e(1, 1)a(1, 1)e(1, 2) + e(1, 1)a(1, 1)e(1, 1)a(1, 2)e(2, 2)$$

$$+e(1, 1)a(1, 2)e(2, 1)a(2, 1)e(1, 2) + e(1, 1)a(1, 2)e(2, 1)a(2, 2)e(2, 2))+$$

$$\pi_2(e(2, 1)a(2, 1)e(1, 1)a(1, 1)e(1, 2) + e(2, 1)a(2, 1)e(1, 1)a(1, 2)e(2, 2)+$$

$$e(2, 1)a(2, 2)e(2, 1)a(2, 1)e(1, 2) + e(2, 1)a(2, 2)e(2, 1)a(2, 2)e(2, 2))$$
$$\simeq f(2, b, T)/T(b);$$

$$\pi_1(e(1, 1)a(1, 1)e(1, 2)a(1, 1)e(1, 1) + e(1, 1)a(1, 1)e(1, 2)a(1, 2)e(2, 1)$$

$$+e(1, 1)a(1, 2)e(2, 2)a(2, 1)e(1, 1) + e(1, 1)a(1, 2)e(2, 2)a(2, 2)e(2, 1))+$$

$$\pi_2(e(2, 1)a(2, 1)e(1, 2)a(1, 1)e(1, 1) + e(2, 1)a(2, 1)e(1, 2)a(1, 2)e(2, 1)+$$

$$e(2, 1)a(2, 2)e(2, 2)a(1, 1)e(1, 1) + e(2, 1)a(2, 2)e(2, 2)a(2, 2)e(2, 1))$$
$$\simeq f(3, b, T)/T(b);$$

$$\pi_1(e(1,1)a(1,1)e(1,2)a(1,1)e(1,2) + e(1,1)a(1,1)e(1,2)a(1,2)e(2,2)$$

$$+e(1,1)a(1,2)e(2,2)a(2,1)e(1,2) + e(1,1)a(1,2)e(2,2)a(2,2)e(2,2))+$$

$$\pi_2(e(2,1)a(2,1)e(1,2)a(1,1)e(1,2) + e(2,1)a(2,1)e(1,2)a(1,2)e(2,2)+$$

$$e(2,1)a(2,2)e(2,2)a(2,1)e(1,2) + e(2,1)a(2,2)e(2,2)a(2,2)e(2,2))$$
$$\simeq f(4,b,T)/T(b);$$

$$\pi_1(e(1,2)a(1,1)e(1,1)a(1,1)e(1,1) + e(1,2)a(1,1)e(1,1)a(1,2)e(2,1)$$

$$+e(1,2)a(1,2)e(2,1)a(2,1)e(1,1) + e(1,2)a(1,2)e(2,1)a(2,2)e(2,1))+$$

$$\pi_2(e(2,2)a(2,1)e(1,1)a(1,1)e(1,1) + e(2,2)a(2,1)e(1,1)a(1,2)e(2,1)+$$

$$e(2,2)a(2,2)e(2,1)a(2,1)e(1,1) + e(2,2)a(2,2)e(2,1)a(2,2)e(2,1))$$
$$\simeq f(5,b,T)/T(b);$$

$$\pi_1(e(1,2)a(1,1)e(1,1)a(1,1)e(1,2) + e(1,2)a(1,1)e(1,1)a(1,2)e(2,2)$$

$$+e(1,2)a(1,2)e(2,1)a(2,1)e(1,2) + e(1,2)a(1,2)e(2,1)a(2,2)e(2,2))+$$

$$\pi_2(e(2,2)a(2,1)e(1,1)a(1,1)e(1,2) + e(2,2)a(2,1)e(1,1)a(1,2)e(2,2)+$$

$$e(2,2)a(2,2)e(2,1)a(2,1)e(1,2) + e(2,2)a(2,2)e(2,1)a(2,2)e(2,2))$$
$$\simeq f(6,b,T)/T(b);$$

$$\pi_1(e(1,2)a(1,1)e(1,2)a(1,1)e(1,1) + e(1,2)a(1,1)e(1,2)a(1,2)e(2,1)$$

$$+e(1,2)a(1,2)e(2,2)a(2,1)e(1,1) + e(1,2)a(1,2)e(2,2)a(2,2)e(2,1))+$$

$$\pi_2(e(2,2)a(2,1)e(1,2)a(1,1)e(1,1) + e(2,2)a(2,1)e(1,2)a(1,2)e(2,1)+$$

$$e(2,2)a(2,2)e(2,2)a(2,1)e(1,1) + e(2,2)a(2,2)e(2,2)a(2,2)e(2,1))$$
$$\simeq f(7,b,T)/T(b).$$

The Bayesian estimator approximately solves these equations relative to the unknown values $a(i,j), e(k,l)$.

The monograph [6] states the proof of the law of large numbers for an ergodic Markov chain with a finite number of states. Some techniques of this proof are used later in the proof of statement (4.2).

Let's consider the ergodic HMM. Denote the vector of the stationary probabilities by π:

$$A^T\pi = \pi,$$

where $\pi_1, \ldots, \pi_s$ are components of the π vector.

Let the initial states of the HMM have a distribution of the stationary probabilities. Then any random member of the sequence of the states of the ergodic HMM has the same distribution.

The following ratios are evident:

$$L(l, b, A, E, T) = \sum_{i_1=1}^{s} \pi_{i_1} \Gamma(i_1), \tag{4.4}$$

where

$$\Gamma(i_1) = e(i_1, y_1(l)) \sum_{i_2=1}^{s} a(i_1, i_2) e(i_2, y_2(l)) \cdots$$

$$\sum_{i_b=1}^{s} a(i_{b-1}, i_b) e(i_b, y_b(l)).$$

Build an $I(l, i), i = 1, \ldots, T(b)$, sequence from $T(b)$ random values, where $I(l, i) = 1$ if u_l is an element of the $S_T(b)$ sequence with serial number equal to i. Otherwise, $I(l, i) = 0$.

The distribution of a random variable $I(l, i)$ does not depend on i since the random variables $I(l, i)$ and $I(l, j)$ are independent because of the stationary distribution for the initial states of the ergodic HMM. Then

$$P_r(I(l, i) = 1) = L(l, b, A, E, T). \tag{4.5}$$

But

$$f(l, b, T) = \sum_{i=1}^{T(b)} I(l, i)/T(b), \tag{4.6}$$

$$E(\sum_{i=1}^{T(b)} I(l, i)/T(b) = \sum_{i=1}^{T(b)} P(I(l, i) = 1))/T(b). \tag{4.7}$$

From (4.6), (4.7), it follows that

$$E(f(l, b, T)/T(b)) = L(l, b, A, E, T). \tag{4.8}$$

According to Chebyshev's inequality and (4.8),

$$P(|(f(l,b,T)/T(b)) - L(l,b,A,E,T)| > \varepsilon) < E((f(l,b,T)/T(b)) - L(l,b,A,E,T))^2)/\varepsilon^2.$$

Therefore, (4.2) will be proved if we show that

$$J(T) \to 0, T \to \infty, \tag{4.9}$$

where $J(T) = E((f(l,b,T)/T(b)) - L(l,b,A,E,T))^2$.

Then we get

$$E((f(l,b,T)/T(b))^2) = (1/T(b)^2) \sum_{i=1}^{T(b)} \sum_{j=1}^{T(b)} E(I(l,i)I(l,j)) =$$

$$(1/T(b))^2 \sum_{i=1}^{T(b)} \sum_{j=1}^{T(b)} P(I(l,i)=1)P(I(l,j)=1) = L(l,b,A,E,T)^2.$$

Therefore, the $J(T)$ value is identically equal to zero and the ratio (4.8) is fair. So the law of large numbers (4.2) is proved if we assume that the initial states of the HMM are stationary.

We get an approximate consideration of the general case if, in formula (4.4), instead of the components π_i, we write down components of another $\tilde{\pi}$ vector that does not coincide with the vector π.

We use the following basic property of the ergodic Markov chain.

If $P_r(i,j)^{(n)}$ is the transition probability of the j state from the i state in n steps, then for the ergodic Markov chain, one can find C and $0 < \rho < 1$, which are constants such that (see [7])

$$|P_r(i,j)^{(n)} - \pi_j| \le C\rho^n. \tag{4.10}$$

Therefore, by choosing the integer n in (4.10), we see that for any ε numbers in (4.2), it is possible to define the beginning of a new observable sequence so that the components of vectors $\tilde{\pi}$ and π differ small enough in (4.4). Then the magnitude of $J(T)$ becomes equal to a small value that is distinct from zero, but the validity of the law of large numbers (4.2) is saved.

A shift of the beginning of the observable sequence to the right reduces the empirical frequency by a constant but does not, in practice, influence its ratio to the $T(d)$ value whenever the length of the initial observable sequence is large enough.

The law of large numbers delivers the information on the moving accuracy of the estimation of elements of the A, E stochastic matrices. It is enough for this purpose

to calculate the moving values of a difference $(f(l, b, T)/T(b)) - L(l, b, \hat{A}, \hat{E}, T)$, where $\hat{A}$, $\hat{E}$ are matrices of moving estimations of the unknown A, E matrices.

The $L(l, b, \hat{A}, \hat{E}, T)$ value of the likelihood is convenient for calculations using the formula

$$L(l, b, \hat{A}, \hat{E}, T) = \pi^T D(y_1(l))\hat{A}D(y(l, 2))\hat{A}\cdots$$

$$D(y(l, b-1))\hat{A}D(y(l, d))I(s), \tag{4.11}$$

where $D(y_1(l)) = diag(\hat{e}(1, y_j(l)), \hat{e}(2, y_j(l)), \ldots, \hat{e}(s, y_j(l))$ $I(s)$ is the $s \times 1$ vector, whose elements are equal to 1.

Identity (4.11) is proved by induction.

7.5 A Bayesian Statistical Construction

The unknown random θ vector is a vector made of transition and emission probabilities—the elements $a(i, j)$ and $e(j, k)$ of the A, E stochastic matrices of dimension $s \times s + s \times r$. We shall designate the a priori (nominal) matrices A_{ap} and E_{ap}. In a mathematical simulation, these stochastic ergodic matrices are built with a random number generator. The random $a(i, j)$ and $e(j, k)$ elements of the matrices A, E were determined stochastically.

$$a(i, j) = \frac{a_{ap}(i, j)(1 + \rho\varepsilon_{i,j})}{\sum_{j=1}^{s} a_{ap}(i, j)(1 + \rho\varepsilon_{i,j})}, \tag{5.1}$$

where $1 \le i, j \le s$, and

$$e(j, k) = \frac{e_{ap}(j, k)(1 + \rho\varepsilon_{j,k})}{\sum_{k=1}^{r} e_{ap}(j, k)(1 + \rho\varepsilon_{j,k})}, \tag{5.2}$$

where $1 \le j \le s$, $1 \le k \le r$.

In (5.1) and (5.2), $\varepsilon_{i,j}$, $\varepsilon_{j,k}$ are random independent values that are generated by the sensor of random uniformly distributed numbers within the limits of $-1 \le \varepsilon_{i,j}, \varepsilon_{j,k} \le 1$. The constant ρ approximately realizes a priori representations about relative limits, in which elements of estimated stochastic matrices can be dissipated relative to their a priori values. So, for example, if $\rho = 0.5$, then elements of unknown stochastic matrices can approximately, within the limits of ∓ 50, be dissipated relative to the set of a priori values. At the prescribed value ρ, it is easy to determine using the Monte Carlo method or analytically if we have a boundary in which there can be a dissipation of elements of the stochastic matrices (5.1), (5.2).

For example, if the a priori probabilities are equiprobable for some row of a stochastic matrix, then relation (5.1) or (5.2) defines random variables, for which the $f(x)$ marginal density distribution is

$$f(x) = 1/(2^s \rho^2 (s-2)! x^2) \int_{1-\rho}^{1+\rho} y^2 \sum_{0}^{[(v+s-1)/2]} (-1)^k C_{s-1}^k (v+s-1-k)^{s-2} dy,$$

where $v = (y((1/x) - 1) + s - 1)/\rho, (1-\rho)/(s(1+\rho)) \le x \le (1+\rho)/(s(1-\rho))$. If this inequality is not fulfilled, then $f(x) = 0$. Then we can easily discover numerically a priori variances: performances of the dissipation of these random variables. Certainly, the value ρ can depend on the numbers i, j, k, although it is a constant in this example.

Formulas (5.1) and (5.2) determine the set of random HMMs. These HMMs generate the set of the observable sequences and correspond to the set of empirical frequencies.

Let's note that in [8] it is suggested to use Dirichlet's distribution for a special case of (5.1), (5.2) when $\rho = 1$; also, one a priori probability would be equaled. In this case, the a posteriori distribution of elements of stochastic matrices would be conjugate concerning the a priori distribution. But generating of set of stochastic matrices noticeably would become complicated using a Monte Carlo method: Instead of generating simple, uniformly distributed random numbers, generating the gamma distribution would be necessary.

7.6 Estimating Hidden Markov Model Parameters by the Algorithm-Estimator

We shall consider features and results of an application of the algorithm-estimator for the estimation of HMM parameters in some approximate task of biological sequence analysis [1, 4]. Furthermore, we consider the situation where $s = 5$ and $r = 4$. We assume that the 25 $a(i, j)$ transition probabilities and the 20 $e(j, k)$ emission probabilities are greater than zero. Therefore, it is necessary to estimate the 25 + 20 = 45 unknown parameters of the HMM. We arbitrarily select the nominal (a priori) stochastic matrices $a(i, j), e(i, j)$ using the standard program for generating random uniformly distributed values.

Using Chap. 5 from [1, 4], we find that a stochastic design supposes the following interpretation in the field of approximate representations and problems of biological sequence analysis.

Let the a priori HMM be known. This HMM generates a family of known sequences of nucleotides, which, as is known, consist of four kinds of symbols. The new sequence of nucleotides is observed with properties that differ from the properties of known sequences. We suppose that after smoothing, the observable new sequence is generated by some new HMM, which belongs to the set of random HMMs.

We believe that the algorithm for generating random elements of this set is presented by formulas (5.1) and (5.2) by $\rho = 0.5$: The values of the parameters of

the new (unknown) HMM differ from the parameters of the old HMM on random variables, which modulo do not surpass half the parameters of the old HMM.

Next, it is necessary to use the algorithm-estimator to estimate the 45 parameters corresponding to the new HMM; the algorithm-estimator's inputs are the observable new sequences of nucleotides. The estimation should be made for any stochastic matrices, determined by (5.1) and (5.2), that depend on the 45 parameters entered here.

Following a series of sequential steps, the next step is to show the technology of applying the common principles of the algorithm-estimator stated in Sect. 7.2 of this chapter to a solution of the concrete problem of the Bayesian estimation of the HMM parameters.

We suppose that the observable sequences, consisting of four numbers, are generated via 50,000 sequential pitches of the HMM by tuning of the algorithm-estimator and the subsequent modeling. We also suppose that $d = 2$ and $L = N = 16$.

Step 1 We construct the Y_N vector's *statistics* with $f(1, d, T), \ldots, f(L, d, T)$ components.

Step 2 For the given positive integers d and N, we introduce the vector $W_N(d)$ of dimension $m(d, N) \times 1$, consisting of powers of the empirical frequencies and the basic vector $V(d, N)$ of dimension $(58 + m(d, N)) \times 1$, $V(d, N) = ||\theta \quad W_N(d)||^T$.

Step 3 We adjust the algorithm-estimator. Using a Monte Carlo method, we calculate the statistical characteristics of the basic vectors $V(b, N)$, the expectation vector $\overline{V}(d, N) = E(V(d, N))$, and the covariance matrix $C_V(d, N) = E((V(d, N) - E(V(d, N)))(V(d, N) - E(V(d, N)))^T)$. The number of realizations in a Monte Carlo method equals 1,000.

Step 4 The algorithm-estimator determines the $\hat{\theta} = \hat{E}(\theta|Y_N)(W_N(d))$ estimation vector of dimension 45×1 and estimation error vector for the set of the random HMM, generated by formulas (5.1) and (5.2).

Every $\hat{\theta} = \hat{E}(\theta|Y_N)(W_N(d))$ vector is a mean-square-optimal (on the set of the linear combinations from components of the $W_N(d)$ vector!) estimate for the $E(\theta|W_N(d))$ vector of conditional expectations of the vector θ of dimension 45×1.

Let the random variable θ belong to a set of 45 estimated parameters, and let $\hat{\theta} = \hat{E}(\theta|Y_N)(W_N(d))$ be a set of their estimations. Then 45 values $\Delta = \frac{\hat{\theta}-\theta}{\theta}$ are random elements of the set of random relative estimation errors. Just ahead we present three random arrays composed from 45 relative errors $Delta$ of an estimation. The algorithm-estimator receives the arrays with $d = 1, \quad 2, \quad 3, \quad m(d, N) = 16, \quad 152, \quad 968$ and some set of 45 arbitrary random values $\varepsilon_{i,j}$ at (5.1), (5.2). All variables are statically independent and have a uniform distribution on segment $[-1, 1]$.

Relative error estimation of $a(i, j), d = 1, m(d, N) = 16$

$j\|i$	1	2	3	4	5
1	−0.53574	−0.95833	0.14018	0.25977	0.02330
2	0.19204	−0.31701	−0.63790	−0.23418	0.21034
3	−0.27376	0.31587	0.00589	−0.77283	−0.53393
4	0.12913	0.19244	−0.02867	0.02773	−0.22581
5	0.01988	−0.20032	0.35250	−0.09939	0.24392

Relative error estimation of $e(i, j)$

$j\|i$	1	2	3	4
1	0.00022	−0.17170	−0.12741	0.07749
2	0.00192	−0.00157	0.00881	−0.00650
3	−0.91501	−0.11010	0.08346	0.18855
4	−0.16284	−0.08714	0.13341	−0.22516
5	0.11035	0.41215	−0.31740	−0.09563

Relative error estimation of $a(i, j), d = 2, m(d, N) = 152$

$j\|i$	1	2	3	4	5
1	−0.46880	−0.84398	0.09516	0.25502	0.02875
2	0.15887	−0.24949	−0.52316	−0.25239	0.17226
3	−0.34296	0.24351	0.10000	−0.56704	−0.46242
4	0.10396	0.24173	0.01249	0.03510	−0.27459
5	−0.05516	−0.10083	0.39981	−0.074225	0.23523

Relative error estimation of $e(i, j)$

$j\|i$	1	2	3	4
1	0.06656	−0.15174	−0.16568	0.03388
2	0.08239	−0.12901	0.09147	−0.10179
3	−0.91128	0.11792	0.11097	0.10496
4	−0.13402	−0.07391	0.13880	−0.33272
5	0.00122	0.36223	−0.42267	0.03543

Relative error estimation of $a(i, j), d = 3, m(d, N) = 968$

$j\|i$	1	2	3	4	5
1	−0.21301	−0.34295	0.08603	0.05804	0.00854
2	0.03436	−0.04485	−0.17120	0.08529	0.03101
3	−0.13541	0.05226	0.06203	−0.30946	0.06254
4	0.18316	−0.06709	−0.05169	0.09499	−0.07779
5	−0.05086	0.08488	0.13496	0.00272	−0.03244

Relative error estimation of $e(i, j)$

$j\|i$	1	2	3	4
1	0.00187	0.09547	−0.28177	0.11510
2	0.00783	0.04463	−0.02480	−0.01512
3	−0.09949	−0.10106	0.02721	0.02622
4	0.06925	−0.04654	0.09503	−0.31728
5	−0.00730	−0.01923	−0.10126	0.02996

At $d = 3, m(d, N) = 968, N = 16$ [the algorithm-estimator's approximations use the linear combinations of of third-degree polynomials at $f(1, d, T), \ldots, f(L, d, T)$ components, value terms 968], almost all 45 relative errors of solving the inverse problem are smaller than 0.1.

Appendix

Let's construct an example HMM for which a point of the local maximum likelihood (stationary points of this function in the space of HMM parameters) essentially differs from the HMM's parameters, generating an observable sequence S_T.

Пусть $s = r = 2, a(1, 2) = sin^2(f_1), a(2, 1) = sin^2(f_2), e(1, 1) = sin^2(v_1), e(2, 1) = sin^2(v_2), \pi_1 = sin^2(f_0), \pi_2 = cos^2(f_0)$.

Let $s = r = 2, a(1, 2) = sin^2(f_1), a(2, 1) = sin^2(f_2), e(1, 1) = sin^2(v_1), e(2, 1) = sin^2(v_2), \pi_1 = sin^2(f_0), \pi_2 = cos^2(f_0)$.

We shall define an a priori HMM with independent parameters $f_0 = \pi/4$, $f_1 = 0.35\pi/2$, $f_2 = 0.75\pi/2$, $v_1 = 0.64\pi/2$, and $v_2 = 0.27\pi/2$. We shall designate O a point in R^4 with coordinates f_1, f_2, v_1, v_2. We shall accept that an a priori HMM builds an observable random sequence S_T, which is made from $T = 15{,}000$ 1 characters 1, 2. The log of likelihood $J(S_T)$ is a random variable. However, at $T = 15{,}000$, its dispersion is small upon matching with $J(S_T)$. Furthermore, we approximately believe the likelihood of the corresponding constant observable sequence S_T and HMM, which are close to the a priori HMM, is not random.

Around the center of point O, we shall construct a sequence of parallelepipeds enclosed around each other to which belong a vector of the HMM's parameters. At a sequential increase in the edge lengths of the parallelepipeds, the random search method allows [9] us to construct a finite-sequence HMM. The vector of parameters of every HMM supplies a local maximum likelihood on the set of the vectors corresponding to the parallelepiped. Local maximum likelihoods form an increasing sequence. Therefore, there is a sequence of HMMs that are not equal to the a priori HMM and have a greater local maximum. This fact contradicts the maximum likelihood principle, which states that for observable sequences S_T, the maximum likelihood should be reached on the a priori HMM. Therefore, it is not necessary to consider as reliable an algorithm as the Baum–Welch algorithm, which defines HMM parameters supplying local maximum likelihoods, when estimating HMM parameters.

Division 2: Estimating the Intensity Matrix for a Model of a Markov Process

7.7 Introduction

While working out a stochastic model of the dynamics of the evolution of a real process proceeding in a medium, there may be a situation when one has the right to suppose that the medium, and consequently the process depending on it, satisfy the a priori conditions as follows:

- At any instant, the process is only in one of a finite number of states, n;
- $Pr(t, t+\delta, i, j)$, the probability of transition at instant $t+\delta$ to a state j if at instant t the process was in a state i, is defined by a relationship

$$Pr(t, t+\delta, i, j) = \lambda_{i,j}\delta + o(\delta)$$

for $i \neq j$ and $1 \leq i, j \leq n$. Nonnegative values $\lambda_{i,j}$ are called intensities of the transition [10]; $n(n-1)$ of these values form an intensity matrix of dimension $n \times n$ whose diagonal elements are not defined. The values of the intensities are commonly considered the average frequencies of transitions, and the value $\lambda_{i,j}$ can serve as a measure of the influence of state i on state j. Hence, the estimator of elements of the intensity matrix for an evolving complicated real system, satisfying the listed a priori conditions, can be found useful for experimental determination of a numerical measure of connections between any states of the system. The given a priori conditions are known to define a stochastic construction of a homogeneous Markov process with a finite number of states and with continuous time. There are numerous examples of using models of such processes to describe the dynamics of the functioning of real systems (see, for example, [11]). The Kolmogorov equations [10] deliver exhaustive information about the dynamics of the process

$$dp_i/dt = \lambda_{1,i}p_1 + \cdots + \lambda_{i-1,i}p_{i-1} - \lambda_i p_i + \lambda_{i+1}pi + 1 + \cdots + \lambda_{n,i}p_n, \quad (2.1)$$

$$\lambda_i = \lambda_{i,1} + \cdots + \lambda_{i,i-1} + \lambda_{i,i+1} + \cdots + \lambda_{i,n}, \quad (2.2)$$

where, at the given initial conditions, p_i is the probability of continuing the process at instant t in a state with number i, $i = 1, \ldots, n$.

Hence, to research characteristics of a real process, it is necessary to estimate it previously on experimental data intensity matrix elements. In this section, we describe an application of the polynomial approximation method for this type of estimation if the Markov process under study is ergodic: There are final probabilities, independent of the initial conditions.

For this purpose, the conditions $\lambda_{i,j} > 0$ are sufficient to be true.

7.7.1 Maximum Likelihood Method of Observing Instants of Direct Transitions

Let's consider versions of constructing a set of observations whose results form a digital file of inputs to an algorithm of the statistical maximum likelihood method to calculate estimators of the intensity matrix's elements. Let's consider the simplest construction: when a sequence of observations results in a sequence of the fixed instants in which numbers of current states of the Markov process are changed.

Let's assume that the process at instant t is in state i. Then, according to the a priori conditions, this state generates a Poisson stream of requests of intensity λ_i (2.2) that compels the process to direct (without previously visiting intermediate states) the transition with intensities $\lambda_{i,1}, \ldots, \lambda_{i,i-1}, \lambda_{i,i+1}, \ldots, \lambda_{i,n}$ to one of the states with numbers $1, \ldots, i-1, i+1, \ldots, n$ at a random instant $t + T_i$. Then $q_j(T_{ij}$, a probability of the fact that the state number j will become this state, is defined by

$$q_i = (\lambda_{i,j}/\lambda_i)(1 - \exp(-\lambda_i T_{ij})) \tag{2.3}$$

where $t+T_{ij}$ is an instant of the direct transition from state number i to state number j.

Equation (2.3) follows from the solution at the initial conditions $q_1(t) = 0, \ldots, q_{i-1}(t) = 0, q_i(t) = 1, q_{i+1}(t) = 0, \ldots, q_n(t) = 0$ of the Kolmogorov equations if one assumes that the states $1, \ldots, i-1, i+1, \ldots, n$ are deadlocked [12].

Let the diagram of observations have been constructed so that for all pairs of integers i and $j (i \neq j$ and $0 \leq i, j \leq n)$, which are fixed by numbers of direct transition, and $k[k = 1, \ldots, K(i, j)]$, the random variables T_k, i, j, which are time intervals between instants of a direct transition from i to j, and also an integer $K(i, j)$ that is a number of such direct transitions in the general (nonrandom) time of the observation process. Let's try to estimate elements of the intensity matrix by the maximum likelihood method on the basis of results of observations of direct transitions. We will assume that

$$x_{1,2}, \ldots, x_{1,n},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_{i,1}, \ldots, x_{i,i-1}, x_{i,i+1}, \ldots, x_{i,n},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_{n,1}, \ldots, x_{n,n-1}$$

are unknown elements of the intensity matrix and

$$x_1 = x_{1,2} + \ldots + x_{1,n},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_i = x_{i,1} + \ldots + x_{i,i-1} + x_{i,i+1} + \ldots + x_{i,n},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_n = x_{n,1} + \ldots + x_{n,n-1}.$$

Then, if we take into consideration the Markov property of the random process, Eq. (2.3) implies that $L(x_{1,2}, \ldots, x_{n,n-1})$, a function of the likelihood of the results of observations constructed above, can be represented by

$$L(x_{1,2}, \ldots, x_{n,n-1}) = \prod_{i=1}^{n} (x_{i,1}/x_i)^{K(i,1)} \times$$

$$\cdots \times (x_{i,i-1}/x_i)^{K(i,i-1)} (x_{i,i+1}/x_i)^{K(i,i+1)} \cdots (x_{i,n}/x_i)^{K(i,n)} \times$$

$$(1 - \exp(-x_i T_{1,i,1})) \cdots (1 - \exp(-x_i T_{K(i,1),i,1})) \times \cdots$$

$$(1 - \exp(-x_i T_{1,i,n})) \cdots (1 - \exp(-x_i T_{K(i,n),i,n})). \tag{2.4}$$

According to the maximum likelihood method, as an estimator of elements of the intensity matrix, one ought to consider values $x_{1,2}, \ldots, x_{n,n-1}$, delivering a maximum of the function $L(x_{1,2}, \ldots, x_{n,n-1})$.

However, it is easy to verify that if $K(i, 1) = \ldots = K(i, i-1) = K(i, i) = K(i, i+1) = \ldots = K(i, n) = K$, then the method gives absurd values for the estimators of the variables $x_{i,1}, \ldots, x_{i,i-1}, x_{i,i+1}, \ldots, x_{i,n}$.

In fact, here we have

$$L(x_{1,2}, \ldots, x_{n,n-1}) =$$

$$\prod_{i=1}^{n} (x_{i,1} \cdots x_{i,i-1} x_{i,i+1} \cdots x_{i,n})^K / (x_i^{K(n-1)}) \times$$

$$(1 - \exp(-x_i T_{1,i,i})), \ldots, (1 - \exp(-x_i T_{K,i,1}))$$

$$(1 - \exp(-x_i T_{1,i,n})), \ldots, (1 - \exp(-x_i T_{K,i,n})). \tag{2.5}$$

But the harmonic mean is no more than the arithmetic mean and becomes equal to it at equality of all the arguments of these functions of many variables. Hence, at the fixed variable $x_i = c$, the function

$$(x_{i,1}/x_i)^K, \ldots, (x_{i,i-1}/x_i)^K, (x_{i,i+1}/x_i)^K, \ldots, (x_{i,n}/x_i)^K$$

reaches its maximum if

$$x_{i,1} = \cdots = x_{i,i-1} = x_{i,i+1} = \cdots = x_{i,n} = c/,$$

and then

$$L(x_{1,2}, \ldots, x_{n(n-1} = (1/(n-1))^{K(n-1)} \times$$

$$(1 - \exp(-cT_{1,i,1})), \ldots, (1 - \exp(-cT_{K,i,1})) \times$$

$$(1 - \exp(-cT_{1,i,n})), \ldots, (1 - \exp(-x_i T_{K,i,n})). \tag{2.6}$$

Maximum of the right of Eq. (2.4) is reached for $c \to \infty$, which leads to absurd values of estimators. Therefore, the maximum likelihood method is unsuitable for estimation at least in one special case of results of observations of direct transitions. Hence, later, we do not consider this kind of observation.

7.7.2 *Algorithm of Estimating Observation States in Instants of Indirect Transitions*

Let a process of observations be such that in the instants $0, \ldots, T, \ldots, k'$, the integer variables $i(0), \ldots, i(T), \ldots, i(kT), \ldots,$ that is, the numbers of states in which a random process was found to be in the instances, multiple to a selected variable T, are held fixed. In these instants, the process implements indirect transitions in the sense that during an interval of time $[0, T]$, there can be random instants when the process stays in other states.

The variable $\theta(i, j)$ is the probability that the process at instant $t + T$ is in state number j if at instant t it was at state number i. The variable is defined by numerical integration of the Kolmogorov equations (2.1) from instant $t = 0$ to instant $t = T$ at the initial conditions as follows:

$$p_1(0) = 0, \ldots, p_{i-1}(0) = 0, p_i(0) = 1, p_{i+1}(0) = 0, \ldots,$$

$$\ldots, p_n(0) = 0 : \theta(i, j) = p_j(T).$$

The variables $\theta(i, j)$, where $1 \le i, j \le n$, form a stochastic matrix of transition probabilities of a homogeneous Markov chain (Markov process in discrete time). These variables are a function $F(\ldots)$ from the matrix of coefficients of the Kolmogorov differential equations, defined by matrix Λ, an intensity matrix of the model of the Markov process:

$$\theta(i, j) = F(i, j, \Lambda).$$

The above-mentioned function is defined implicitly: by numerical integration of the Kolmogorov equations on a segment $[0, T]$ at the given initial conditions.

As an input of a standard procedure of calculations, one uses the given above vector of initial conditions and the intensity matrix Λ; the procedure returns a vector of state probabilities at instant T. This vector serves as a corresponding stochastic matrix of transition probabilities.

Let $f(i)$ be the number of observations when the Markov process was in state i. Let $\varphi(i, j)$ be the number of observations when at instant k' the process was in state j and at instants $(k-1)T$ it was in state i. The ratio $\varphi(i, j)/f(i)$ can be considered the average frequency of transitions from state i to state j at instants, multiple of T.

It is well known that this relation is an estimator $\hat{\theta}(i, j)$ of transition probability $\theta(i, j)$ of a homogeneous Markov chain (Markov process in discrete time) by the criterion of maximum likelihood function:

$$\hat{\theta}(i, j) = \varphi(i, j)/f(i). \tag{2.8}$$

If $\theta(i, j) > 0, 1 \leq i, j \leq n$, then estimator (2.8) is asymptotically unbiased and consistent.

The computational process to construct an estimator $\Lambda(d)$ for the intensity matrix Λ is composed of the following steps.

1. By observations at instants, multiple of T, one defines a matrix of average frequencies whose elements are the relations $\varphi(i, j)/f(i), 1 \leq i, j \leq n$.
2. By the method of polynomial approximation of an inverse function, one approximately defines a root vector of the system of algebraic equations:

$$F(i, j, \Lambda\hat{(d)} = \varphi(i, j)/f(i), 1 \leq i \leq n, 1 \leq j \leq n-1. \tag{2.9}$$

As components of the root vector of this system, there are $n(n-1)$ elements of the matrix of estimators $\Lambda(d)$ for elements of the intensity matrix Λ.

To construct an optimal (in the root-mean-square sense for a given integer d) estimator of elements of matrix Λ, it is necessary to calculate the integrals [over an a priori cube in $R^{n(n-1)}$], corresponding to the first and second a priori statistical moments. The coordinates of vertices of the cube are prescribed by positive numbers, namely, the left and right borders for intensities i, j, which are selected from a prioristic considerations.

The a priori cube is covered with a grid of nodes, which correspond to a set of intensity matrices of $\Lambda^{'}$. For every matrix $\Lambda^{'}$, $n(n-1)$-dimensional vector $F(i, j, \Lambda^{'})$ is defined as a result of numerical integration from $t = 0$ to $t = T$ of the Kolmogorov equations (2.1) for n vectors of initial conditions

$$p_1(0), \ldots, p_{i-1} = 0, p_i = 1, p_{i+1} = 0, \ldots, p_n 0 = 0, 1 \leq i \leq n.$$

The matrix of coefficients of the equations corresponds to the intensity matrix $\Lambda^{'}$.

Let $n = 2$; one estimates the intensities $\Lambda_{1,2}, \Lambda_{2,1}$. In this case, probabilities p_1 and p_2 will be defined from the relationships

$$\dot{p}_1 = -\Lambda_{1,2} p_1 + \Lambda_{2,1} p_2,$$

and $p_2 = 1 - p_1$. From here, the equations (II.3.2) for estimators $\hat{\Lambda}_{1,2}$, $\hat{\Lambda}_{2,1}$ will acquire the form

$$\exp(-(\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1})T)$$

$$+ \frac{\hat{\lambda}_{2,1}}{\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1}} \times (1 - \exp(-(\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1}))T) = \varphi(1, 2)/f(1), \tag{2.10}$$

$$\exp(-(\Lambda_{1,2} + \Lambda_{2,1})T)$$

$$+ \frac{\hat{\Lambda_{1,2}}}{\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1}}(1 - \exp(-(\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1}))T) = \varphi(2, 1)/f(2). \tag{2.11}$$

Adding Eqs. (2.10) and (2.11), we will find $\hat{\Lambda}_{1,2} + \hat{\Lambda}_{2,1}$ and then we will find a matrix of intensities without applying the polynomial approximation method.

However, already at $n = 3$, we will have six unknown intensities, a system of two differential Kolmogorov equations and six nonlinear algebraic equations with implicitly definable left parts for the estimators of elements of the intensity matrix. In this case, one should apply the polynomial approximation method.

Thus, it follows that for an ergodic Markov process, the estimators of intensities are unbiased and consistent once the system of algebraic equations (2.9) is solved precisely. The necessary accuracy is reached by increasing an integer d, which corresponds to a sufficiently large number of terms in a representation of the root vector of Eq. (2.9) in terms of a linear vectorial combination of integer powers of numbers $\varphi(i, j)/f(i)$. After the number of terms has been increased, this linear combination converges uniformly to an exact root vector.

Let's notice that the given statements are true only if variable T is assigned to be "not very big". In fact, for an ergodic process, at a big variable T, the vector of the solution of the Kolmogorov equations is coincident with the vector of final probabilities and, hence, does not depend on the initial conditions. Then, the left and right parts of system (2.9) do not depend on i, and to determine the unknown intensities, there remain only $n - 1$ algebraic equations.

Numerically, the question of the dependence of an estimator accuracy on the estimated variable is considered in an example presented in the next section.

7.7.3 A Numerical Example

Further, upon considering the numerical example, we used a fourth-order Runge–Kutta method to integrate the Kolmogorov equations.

We consider that all elements of the intensity matrix do not exceed 1 and are one realization of six random numbers, uniformly distributed on a segment [0, 1].

Then, upon generating an ergodic homogeneous Markov process with three states and continuous time, and also upon modeling the functioning of an algorithm that estimates intensities, we assume that factual variables of elements of the intensity matrix have had values as follows:

$$\lambda_{1,2} = 0,827353;\ \lambda_{1,3} = 0,099387;\ \lambda_{2,1} = 0,051879;$$
$$\lambda_{2,3} = 0,176995;\ \lambda_{3,1} = 0,009050;\ \lambda_{3,2} = 0,703354.$$

The accepted values of intensities show both strong influence (transitions $1 \to 2, 3 \to 2$), and weak influence (transitions $2 \to 1, 3 \to 1$) of states to each other. The number of time intervals of length T, at whose ends were fixed states obtained at indirect transitions, was equal to 1,000,000 in the statistical model. Here is an example of the implementations of values of stochastic functions $f(i)$ and $\varphi(i, j)$, found by statistical modeling at $T = 1$ s:

$$f(1) = 44,196\ \ \varphi(1,1) = 17,141\ \ \varphi(1,2) = 23,225\ \ \varphi(1,3) = 3,830;$$
$$f(2) = 761,571\ \varphi(2,1) = 24,323\ \varphi(2,2) = 646,764\ \varphi(2,3) = 90,484;$$
$$f(3) = 194,233\ \ \varphi(3,1) = 2,732\ \ \varphi(3,2) = 91,582\ \ \varphi(3,3) = 99,919.$$

Consistency of the digital statistical model of a Markovian process, controlled by the intensity matrix given above, and the method of numerical integration of the Kolmogorov equations on the interval $[0, T]$ confirms the approximate coincidence of values of transition probabilities $\theta(i, j)$, found by numerical integration, and their estimators $\hat{\theta}(i, j))$, defined by Eq. (2.8) on the basis of the results of statistical modeling.

The results of numerical integration and statistical modeling at $T = 1$s are presented next. As is obvious, for all i, j, the variables $\theta(i, j)$ and $\hat{\theta}(i, j))$ approximately coincide:

$$\theta(1,1) = 0.389\ \ \theta(1,2) = 0.522\ \theta(1,3) = 0.087.$$
$$\hat{\theta}(1,1)) = 0.387\ \hat{\theta}(1,2) = 0.525\ \hat{\theta}(1,3) = 0.086.$$
$$\theta(2,1) = 0.032\ \ \theta(2,2) = 0.848\ \theta(2,3) = 0.119.$$
$$\hat{\theta}(2,1)) = 0.031\ \hat{\theta}(2.2) = 0.849\ \hat{\theta}(2,3) = 0.118.$$
$$\theta(3,1) = 0.014\ \ \theta(3,2) = 0.471\ \theta(3,3) = 0.514.$$
$$\hat{\theta}(3,1) = 0.014\ \ \hat{\theta}(3,2) = 0.471\ \hat{\theta}(3,3) = 0.514.$$

The given data imply that for $T = 1$ s, the observations are informational because the variables $\theta(i, j)$ and $\hat{\theta}(i, j)$ appreciably depend on i and, hence, should define six unknown intensities from the solution of six nonlinear algebraic equations (2.9).

However, if one assumes $T = 4$ s, then the dependence of these variables on i essentially reduces, which is illustrated by the data as follows:

$$\begin{array}{lll} \hat{\theta}(1,1) = 0.052 & \hat{\theta}(1,2) = 0.758 & \hat{\theta}(1,3) = 0.188, \\ \hat{\theta}(2,1) = 0.044\ \hat{\theta}(2,2) = 0.763 * \hat{\theta}(2,3) = 0.191, & & \\ \hat{\theta}(3,1) = 0.042 & \hat{\theta}(3,2) = 0.752 & \hat{\theta}(3,3) = 0.205. \end{array}$$

Hence, at $T = 4$ s, there occur not six, but only three equations, carrying essential information about six unknown intensities.

The numerical data presented below illustrate the accuracy of estimation of intensity values with the help of an algorithm of the polynomial approximation method. In terms of $d(i, j)$, one denotes errors of estimating the intensity $\lambda_{i,j}$. The dimension of these variables is $1/c$.

Iteration 1 means that the Bayesian method of solving six nonlinear equations is applied in conditions when an a priori region of the roots' existence of system (2.9)—is a cube in R6 whose edges all have length 1 and are parallel to axes of coordinates, and all the cube center coordinates are equal to 1/2.

Iteration 2 means that the edge lengths of the a priori cube are reduced five times, and the center coordinates are equal to values of estimators of the intensities, found at iteration 1.

$T = 0.5$ s
Iteration 1:

$$\begin{array}{lll} d(1,2) = -0,003421 & d(l,3) = 0,000995 & d(2,l) = 0,000622 \\ d(2,3) = -0,001623 & d(3,1) = 0,002634 & d(3,2) = -0,000720 \end{array}$$

Iteration 2:

$$\begin{array}{lll} d(1,2) = -0,002057 & d(l,3) = -0,000779 & d(2,1) = -0,000286 \\ d(2,3) = 0,002098 & d(3,1) = 0,000487 & d(3,2) = -0,000858 \end{array}$$

$T = 1$ s
Iteration 1:

$$\begin{array}{lll} d(l,2) = -0,002386 & d(l.3) = -0,012476 & d(2,1) = -0,000499 \\ d(2,3) = -0,007302 & d(3.1) = -0,008781 & d(3,2) = -0,000570 \end{array}$$

Iteration 2:

$$\begin{array}{lll} d(l,2) = 0,006013 & d(l,3) = -0,001362 & d(2,1) = -0,000032 \\ d(2,3) = -0,001242 & d(3.1) = -0,000413 & d(3,2) = -0,000544 \end{array}$$

For $T = 2$–4 s, the accuracy of iteration 1 is insufficient for transition to iteration 2 (estimators of some intensities are negative) due to the above-noted small information of three of six equations. However, if the a priori data for iteration 2 are accepted

equal to those used for $T = 0.5$–1 s, then estimation is made in a situation of small information, though with appreciably smaller accuracy.

Iteration 2:
$T = 2$ s:

$$d(l,2) = 0,000374 \quad d(l,3) = 0,001687 \quad d(2,1) = 0,000160$$
$$d(2,3) = -0,001720\ d(3.1) = -0,000537\ d(3,2) = -0,001181$$

$T = 4$ s:

$$d(l,2) = -0,036779\ d(l,3) = 0,030037\ d(2,1) = -0,001369$$
$$d(2,3) = -0,005763\ d(3,1) = 0,003625\ d(3,2) = -0,015354.$$

Division 3: Nonlinear Filtration of Markov Random Process with Finite Number of States

We present and justify a multipolynomial approximation algorithm (MPA algorithm) for nonlinear filtering components of the state vector Markov random process with a finite number of states. The situations when the stochastic matrix conditional probability of Markov random process is known exactly, and when there are a priori errors of its elements, are considered. The statement of the nonlinear filtering problem is close to the general formulation of the problem from [11], Chap. 9. We assume that the current number of states of Markov random processes is measured with errors, the distribution of which is given. The algorithm is based on a multipolynomial approximation vector of conditional expectation and does not require review of decisions regarding the infinite-dimensional system of stochastic differential equations given in [11]. Examples considered are those in which an equal number of states of 25 Markov random processes' estimated vector consist of 150 successive states, measured with errors. The MPA algorithm of nonlinear filtering is reduced 10 times a posteriori dissipation compared with the a priori.

7.8 Introduction and Statement of the Problem

The monograph [13], Chap. 9, described the general problem of nonlinear filtering of a Markov random process, which has a finite or countable number of states. The monograph examines a pair of stochastic processes $(\theta, Y) = (\theta_t, y_t)$, $0 <= t <= T$, where the unobserved integer θ_t is a Markov random process with a finite or countable number of states, and the observed process y_t admits the stochastic differential

$$dy_t = A_t(\theta_t, Y)dt + B_t dW_t,$$

where W_t is a Wiener process, and $A_t(\ldots)$, B_t are some functionals. The Markov random process θ_t has continuity to the right and a set (known) probability density of transition from one state to another.

Many problems of applied statistics of random processes come via the scheme in [11], where the unobserved process takes discrete values and the noise has the character of a "white" noise. For the statement of the nonfiltering problem, define an algorithm that uses the observations $Y(0, t)$ to build $\hat{\theta}(0, T)|Y(0, T)$, which is estimation of the process θ_t. In [13] assume that $\hat{\theta}(0, T)$ is the posterior probability of θ_t and prove that it is the function satisfying Eq. (9.21) in [13] and following from it the infinite system of nonlinear stochastic differential equations (9.23) in [11]. Therefore, in general, these equations cannot constructively use the algorithm for nonlinear filtering. Our proposed consideration is close to [11] in their statement of the problem formulation, but, in contrast to [11], it gives an easily implemented nonlinear filtering algorithm.

We believe that t takes discrete values of $0, 1, \ldots, k, \ldots, T$. At these discrete moments, the sequence states that the Markov random process θ_t arises. Components of the process take the unobserved integers $\theta_0, \theta_1, \ldots, \theta_k, \ldots, \theta_T$, where θ_k is the number of states of a Markov random process. The maximum of these integers defines the order of the random process: the number M of its states.

It is assumed that successive observations are of the form

$$y_k = \theta_k + \eta_k, \tag{1.1}$$

where $k = 0, 1, \ldots, T$, η_k are random values with a given distribution.

An unobserved Markov random process is generated by the stochastic matrix of conditional probabilities, whose the elements $q(i, j) = P(i|j)$ are known with random errors $\varepsilon_{i,j}$ that have a given distribution.

In terms of features of the nonlinear filtering problem, for the estimation problem of a random vector $\theta(0, T)$, the estimated vector has a large dimension. The dimension of the estimated vector $\theta(0, T)$ set out in the following example is not small; it is equal to 150. Therefore, the proposed algorithms for nonlinear filtering provides data compression to reduce the dimension of the matrix used. Compression occurs by dividing the observed sequence $y_0, y_1, \ldots, y_T$ into segments and then summing.

Equation (1.1) and values $\tilde{q}(i, j) = q(i, j)(1 + \varepsilon_{i,j})$ model the situation in which the dynamic system has M types of consecutive random failures. Every one of them after an accident and instant repairs goes to one of the types of failures. The transition probabilities determine the a priori stochastic matrix of conditional probabilities.

Failure detection of a species produces a nonlinear filtering algorithm whose input is the value of y_k, which is measured from the current random error η_k, the random number of the current state. We consider the version where the matrix of conditional probabilities is known ($\varepsilon_{i,j} = 0$) and the option where the matrix of conditional probabilities is unknown $\tilde{q}(i, j) = q(i, j)(1 + \varepsilon_{i,j})$.

The MPA algorithm, which is justified ahead, is the basis of the proposed nonlinear filtering algorithm. The estimation algorithm builds $\hat{\theta}(0, T)|Y(0, T))$ in the approximate expression form for the conditional expectation vector $E(\theta(0, T)|Y(0, T))$.

Therefore, the evaluation is nearly optimal in the mean square. The algorithm approximates the conditional expectation $E(\theta(0, T)|Y(0, T)$ polynomials, which is a linear combination of powers of the components of the observed vector $Y(0, T)$. Vector estimates converge to the conditional expectation of uniformity on some of the increase in the degree of the approximating polynomials.

The algorithm determines a family of random virtual Markov random processes, which "immerses" this model (1.1). Its components are the corresponding perturbation matrices of conditional probabilities and perturbed observations of the form (1.1). Statistical analysis of the Monte Carlo family members is defined in the t corresponding vector of mean and covariance matrix estimation errors for basic vectors consisting of the perturbed quantities $\theta_0, \theta_1, \ldots, \theta_T, y_0, y_1, \ldots, y_T$. Through a system of linear algebraic equations, these data explicitly define the approximating polynomial coefficients of the vector and matrix evaluations of covariance estimation errors.

Diagonal elements of the covariance matrix are determined at the time of T a posteriori dissipation numbers, states of the unobserved Markov process θ_t. The effectiveness of the proposed nonlinear filtering algorithm is the value $\rho(0, T)$. It is the square root of the ratio of the a posteriori variance, which characterizes the dispersion of the posterior component of the estimated vector, to the a priori variance, which characterizes the dispersion of this vector in the absence of observations.

7.8.1 Basic Scheme of the Proposed Nonlinear Filtration Algorithm

Fundamentally, the MPA algorithm assumes that unknown vector sequential states of the Markov random process θ_t are random on the set of possible realizations. We assume that the a priori statistical generator for the computer-generated random values η_t, $\varepsilon_{i,j}$ in (1.1) is given. This generator makes the MPA algorithm estimating components of the vector $\theta(0, T)$ Bayesian. Further, for particular calculations, we assume that the random values η_k, $\varepsilon_{i,j}$ in (1.1) are distributed uniformly and can be called by the standard Random program in Turbo Pascal.

The MPA algorithm provides the approximation method we implement with the multidimensional power series of the vector $E(\theta(0, T)|Y(0, T)$ of the conditional mathematical expectation of the vector $\theta(0, T)$ if the vector of measurements $Y(0, T)$ is fixed and a priori statistical data on random values of errors η_t, $\varepsilon_{i,j}$ are given.

The vector $E(\theta(0, T)|Y(0, T))$ is known to be the optimal, in the mean-square-sense, estimate of the random vector $\theta(0, T)$.

We describe the steps of the MPA algorithm's operation.

Step 1 We assume that one has created a computer program that generates the Markov random process, corresponding to the option where the matrix of conditional probabilities is known without error or to the option where the matrix of conditional probabilities is known with bugs, which are given an a priori distribution. Using a statistical model of the distribution of

random errors of measurement $\eta_t, \varepsilon_{i,j}$, construct the set of possible realizations of random functions $Y(0,T)$, which consists of sequences of the form $y_0, y_1, \ldots, y_k, \ldots, y_T \in \Omega_{Y_T}$. One of these sequences is observable.

We believe that the nonlinear filtering algorithm implements multipolynomial representation in ideal degrees measured by the quantities $y_0, y_1, \ldots, y_T$ according to the model (1.1). The degree of the polynomials does not exceed a given integer d.

Suppose that d is a given positive integer and that the set of integers $a_1, \ldots, a_T$ consists of all nonnegative solutions of the integer inequality $a_1 + \ldots + a_T \le d$, whose number we denote by $m(d,T)$. The value $m(d,T)$ is given by the recurrent formula proved by induction. We obtain the vector $W_T(d)$ of dimension $m(d,T) \times 1$, whose components $w_1, \ldots, w_m(d,T)$ are all possible values $y_1^{a_1} \ldots y_T^{a_T}$ of the form that represents the powers of measurable values.

We will look for an approximate representation of the conditional expectation vector $E(\theta(0,T)|Y(0,T))$ on the set of polynomials with respect to components of the vector $W_T(d)$. Then we'll use an obvious statement: $E(\theta(0,T)|Y(0,T)) = E(\theta(0,T)|W_T(d))$.

Then, we construct the base vector $V(d,N)$ of dimension $(1+T+m(d,T)) \times 1$, $V(d,N) = \|\theta(0,T) \quad W_T(d)\|$. We apply the Monte Carlo method to find the prior first and second statistical moments of the vector $V(d,T)$, that is, the mathematical expectation $\bar{V}(d,N)$, and the covariance matrix $C_V(d,T) = E((V(d,T) - \bar{V}(d,T))(V(d,T) - \bar{V}(d,T))^T)$.

We consider the algorithm fundamental for solving the problem of finding the estimate of the conditional expectation vector $E(\theta(0,T)|W_T(d)$ that is optimal in the mean square sense. This vector is known to be the optimal in the mean-square-sense estimate of the vector $\theta(0,T)$ once the vector $Y(0,T)$ is fixed. Therefore, it is justified that it is the conditional expectation vector that tends to estimate.

We construct the algorithm that ensures polynomial approximation of the vector $E(\theta(0,T)|W_T(d))$. To do this, we find the approximate estimate of the vector $E(\theta(0,T)|W_T(d))$, which is linear with respect to components of the vector $W_T(d)$ and optimal in the mean square sense.

Step 2 For given d and T and a fixed vector $W_T(d)$, we assign the vector $\hat{\theta}(0,T)$ $(W_T(d))$ to be the solution to the estimation problem. This vector gives an approximate estimate of the vector $E(\theta(0,T)|W_T(d))$ that is optimal in the mean square sense on the set of vector linear combinations of components of the vector $W_T(d)$:

$$\hat{\theta})(0,T)(W_T(d)) = \sum_{a_1+\ldots+a_T \le d} \lambda(a_1, \ldots, a_T) y_1^{a_0} \cdots y_T^{a_T}. \tag{2.1}$$

The vector $\bar{V}(d,N)$ and the matrix $C_V(d,N)$ are the initial conditions for the process of recurrent calculations that realizes the principle of observation

decomposition of Chap. 1 and consists of $m(d, T)$ steps. Once the final step is performed, we obtain vector coefficients $\lambda(a_1, \ldots, a_T)$ for (2.1). Moreover, we determine the matrix $C(d, T)$, which is the estimation error covariance matrix for the vector $E(\theta(0, T)|W_T(d))$ of conditional mathematical expectation estimated by the vector $\hat{\theta}(0, T)(W_T(d)$.
To obtain the explicit expression for the estimation vector, we calculate elements of the vector $\overline{V}(d, T)$ and the covariance matrix $C_V(d, T)$ that are the first and second (centered) statistical moments for the vector $V(d, T)$, respectively.

This vector and this matrix can be divided into blocks of the following structure:

$$E(E(\theta(0, T)|W_T(d))) = E(\theta(0, T)),$$

$$E((E(\theta(0, T)|W_T(d)) - E(E(\theta(0, T)|W_T(d))((E(\theta(0, T)|W_T(d)$$
$$- E(E(\theta(0, T)|W_T(d))))^T) = E((\theta(0, T) - E(\theta(0, T))(\theta(0, T) - E(\theta))^T),$$

$$L_T(d) = E((E(\theta(0, T)|W_T(d) - E(E(\theta(0, T)|W_T(d)))$$

$$((E(\theta(0, T)|W_T(d) - E(E(\theta(0, T)|W_T(d)))))^T) =$$

$$E(\theta(0, T)W_T(d)^T) - E(E(\theta(0, T))E(W_T(d))^T,$$

$$Q_T(d) = E((W_T(d) - E(W_T(d)))(W_T(d) - E(W_T(d)))^T).$$

The right-hand sides of these blocks are the first and second (centered) statistical moments calculated by the Monte Carlo method. However, their left-hand sides also serve as the first and second (centered) statistical moments of components of the vector of conditional mathematical expectations. Hence, we can use mathematical models of form (1.1) to find these statistical moments experimentally for vectors of conditional expectations as well. This obvious proposition gives us the basis for the practical implementation of the computational procedure of estimating the vector of conditional expectations.

We introduce an estimate

$$\hat{\theta}(0, T)(W_T(d) = E(\theta) + \Lambda_T(d)(W_T(d) - E(W_T(d)), \tag{2.2}$$

where the matrix $\Lambda_T(d)$, $1 + T \times m(d, T)$ satisfies the equation

$$\Lambda_T(d)Q_T(d) = L_T(d).$$

We also introduce

$$\tilde{z}_{\theta(0,T)}(W_T(d)) = z + \tilde{\Lambda}_T(d)(W_T(d) - E(W_T(d))), \tag{2.3}$$

where z and $\tilde{\Lambda}_T(d)$ are the arbitrary vector and matrix of dimensions $1 + T \times 1$ and $1 + T \times m(d, T)$. Suppose $C(d, T)$ and $\tilde{C}(d, T)$ are the estimation error covariance matrices for the vector $E(\theta(0, T)|W_T(d))$ generated by the estimates $\hat{\theta}(0, T)(W_T(d))$ and $\tilde{z}_{\theta(0.T)}(W_T(d))$.

Lemma *The matrix $\tilde{C}(d, T) - C(d, T)$ is a nonnegative definite matrix: $C(d, T) \leq \tilde{C}(d, T)$.*

The lemma follows from the identity of Chap. 1. Corollary of the lemma. For the vector $E(\theta(0, T)|W_T(d))$, the vector $\hat{\theta}(0, T)(W_T(d))$ is the optimal in the mean-square-sense estimate among the set of estimates that are linear with respect to components of the vector $W_T(d)$. If $Q_T(d) > 0$, the estimation vector is unique and

$$\hat{\theta}(0, T)(W_T(d)) = E(\theta(0, T)) + L_T(d)Q_T(d)^{-1}(W_T(d) - E(W_T(d))). \quad (2.5)$$

The estimation error covariance matrix $C(d, T)$ of the vector $E(\theta(0, T)|W_T(d))$ is given by the formula

$$C(d, T) = C_{\theta(0,T)} - \Lambda_T(d)L_T(d). \quad (2.6)$$

If $Q_T(d) \geq 0$, the vectors that provide a linear and optimal in the mean-square-sense estimate are not unique; however, the variances of components of the difference between these vectors are zeros. Formula (2.1) gives explicit expressions for the vector coefficients of the form $\lambda(a_1, \ldots, a_T)$. To find these relations, we open the explicit expressions for components of the vector $W_T(d)$ and the right-hand side of (2.1) and equate them to the right-hand side of formula (2.1). We consider asymptotic estimation errors when we use (2.1). Suppose the vector Y_T is fixed. We assume that the vector $E(\theta(0, T)|W_T(d))$ is given by the function of $W_T(d)$ on some a priori region that is compact; the function is continuous on this region. Then the following theorem holds.

Theorem

$$Sup \quad _{Y_T \in \Omega_{Y_T}} |\hat{\theta}(0, T)(W_T(d)) - E(\theta(0, T)|W_T(d)| \Rightarrow 0, d \Rightarrow \infty. \quad (2.7)$$

Proof The multidimensional analog of Weierstrass's theorem, which is the corollary of Stone's theorem [14], states that for any number $\varepsilon > 0$, there exists a multidimensional polynomial $P(W_T(d_\varepsilon))$ such that

$$Sup \quad _{Y_T \in \Omega_{Y_T}} |P(W_N(d_\delta)) - E(\theta|W_T(d))| < \delta.$$

□

We can rewrite this relation as

$$Sup \quad _{Y_T \in \Omega_{Y_T}} |P(W_N(d)) - E(\theta(0, T)|W_T(d))| \Rightarrow 0, \quad d \Rightarrow \infty. \quad (2.8)$$

We assume that C is the covariance matrix of the random vector $P(W_T(d) - E(\theta(0, T)|W_T(d))$:

$$C = E((P(W_T(d)) - E(\theta(0, T)|W_T(d))(P(W_T(d)) - E(\theta(0, T)|W_T(d)))^T.$$

It follows from (2.8) that

$$C \Rightarrow 0_T, \quad d \Rightarrow \infty. \tag{2.9}$$

By construction, the vector estimate $\hat{\theta}(0, T)(W_T(d))$ provides the estimate of the vector $\theta(0, T)$ that is linear with respect to components $W_T(d)$ and optimal in the mean square sense. However, it follows from the lemma that for any other nonoptimal linear estimate, including estimates of the form $P(W_T(d)$, the relation $C \geq C(d, T)$ holds. Hence, taking into account (2.8), we obtain

$$C(d, T) \Rightarrow 0_{1+T}, d \Rightarrow \infty. \tag{2.10}$$

Proposition (2.9) is equivalent to (2.6). Thus, by (2.1), the algorithm determines the vector series that, with the increasing number $m(d, N)$ of its terms, approximates the vector of conditional mathematical expectation of the vector $\theta(0, T)$ of the estimated parameters with an arbitrary uniformly small mean square error.

7.8.2 *Effective Work of Nonlinear Filtration Algorithm at Estimating States of a Nominal Model Markov Random Process if Random Observation Errors are Large and Uniformly Distributed in* **[−100, 100]**

In mathematical simulation, it was assumed that the stochastic matrix conditional probability $q(i, j)$, which generates a Markov random process, i, j, has dimension 5×5 and, therefore, contains 25 members. The length m process contains $T = 15$ components $y_k, k = 0, 1, \ldots, 14$, integers, which is in line with the model (1.1), there are k, with errors η_k, independent random variables distributed uniformly on $[-100, 100]$.

For compression, every 10 consecutive numbers are added, and the resulting $T = 15$ value blocks the entrance to the construction of the approximating polynomials.

Consider a situation where there is no prior knowledge of the elements of the error of the conditional probabilities and there are options for multipolynomials in $T = 15$:

$d = 1$.

Approximating multipolynomials are linear combinations 1 degree to 15 whole sizes—15 successive states of the process—and the number of different multipolynomials of degree 1 is $m(d, T) = 15$.

$d = 2$.

Approximating multipolynomials are linear combinations 1 degree to 15 whole sizes—15 successive states of the process— and the number of different multipolynomials of degree 1 is $m(d, T) = 90$.

$d = 3$.

Multiapproximating polynomials are linear combinations no more than 3 degrees to 15, and the number of integers of all the different multipolynomials of degree at most 3 is $m(d, T) = 815$.

$d = 4$.

Multiapproximating polynomials are linear combinations no more than 4 degrees for 15 integers of the number of all different multipolynomials of degree at most 3 is $m(d, T) = 3876$.

Columns 3–6 of below table contain the values T guest $\hat{\theta}(0, T)$ of integers $\theta(0, 15)$ for different integers d, which are given in column 2 at the $T = 15$ segment $\theta(0, T)$ for the evaluated Markov random process, in the absence of random errors η.

		$d = 1$	$d = 2$	$d = 3$	$d = 4$
k	$\theta(0, T)$	$\hat{\theta}(0, T)$	$\hat{\theta}(0, T)$	$\hat{\theta}(0.T)$	$\hat{\theta}(0, T)$
1	4.000	7.955	4.111	4.310	3.555
2	2.000	6.827	3.281	3.142	2.928
3	1.000	4.643	3.441	2.712	0.819
4	3.000	5.317	2.620	3.424	4.225
5	4.000	6.093	4.033	2.010	3.281
6	4.000	5.278	2.797	3.165	4.491
7	4.000	5.366	2.670	2.532	3.623
8	2.000	8.225	3.398	3.239	2.085
9	4.000	5.236	0.385	1.879	4.162
10	2.000	6.624	3.712	1.553	1.781
11	5.000	6.590	3.651	3.036	4.754
12	4.000	6.855	3.225	3.119	4.643
13	1.000	5.130	2.023	3.419	1.661
14	3.000	7.501	2.174	3.115	2.700
15	4.000	6.186	2.241	−0.830	4.085

Division 4: Multipolynomial Approximations for Estimating the Parameters of a Time Series Generated by GARCH Models

In this section, the principles underlying an MPA algorithm are described, and the algorithm is used to estimate the parameters of control time series generated by STGARCH and MGARCH models (in the BEKK specification). These models are used in the consideration of nonlinear problems of financial mathematics.

7.9 Introduction

To analyze and forecast the volatility of a time series with members specified as the logarithms of successive ratios of current asset prices as functions of stock market closing times, financial analysts use mathematical models representing a priori information on the evolution of the series. Widely used are ARCH and GARCH models, which go back to [15] and [16] (evolution of the conditional variance of members in a scalar time series) and to Engle and Kroner (1995) (evolution of the conditional covariance matrix for a vector time series). Given a time series, the parameters of an a priori model for the evolution of its members, the initial conditions, and the forecast time series segments comprise the vector θ of unknown parameters. After obtaining experimental data, namely, time series members, and choosing a mathematical evolution model, one has to estimate the vector θ. This task is usually approached using iterative maximum likelihood estimation, which in theory works only asymptotically, namely, when the length of the time series increases indefinitely.

The implementation of this method encounters fundamental and technical difficulties (associated with the need for computing gradients and Hessians at every iteration step), which motivates attempts to develop new approaches to the estimation of model parameters and other unknowns. Some of the fundamental difficulties are as follows:

1. When a likelihood function is constructed, the distribution of the random elements in the sample has to be determined, taking into account the restrictions imposed on the components of θ, which is a difficult and frequently unfeasible task.
2. A maximizer of the likelihood function is determined numerically by applying a variant of Newton's iterative method, which is complicated by the need to guess an admissible vector of initial conditions (i.e., a first-approximation iteration vector that ensures the convergence of the iterations).
3. The numerical method does not necessarily yield a global maximizer of the likelihood function. Instead, one of the numerous local maximizers might be obtained.

It should be noted that in his fundamental book, Zaks [17] says that, contrary to many textbooks, maximum likelihood estimation is not a universally good approach, and it should not be applied indiscriminately.

Ahead we propose an MPA method (or algorithm) for estimating the components of θ. Free of the above shortcomings, the algorithm has been successfully used to estimate the components of multidimensional parameter vectors in complex dynamic systems. Given a fixed (and finite) time series, under certain conditions, the MPA algorithm produces an estimate vector that uniformly converges on a certain domain to the conditional expectation vector as the degrees of the approximating polynomials increase indefinitely. Therefore, the estimate vector generated by the MPA algorithm is roughly optimal in the mean square sense (it is well known that the conditional expectation vector gives an estimate that is optimal in the mean square sense).

It should be emphasized that in the particular computational problems considered ahead, we simultaneously estimate not only the unknown model parameters but also

the components of the initial condition vector and the vector of forecast time series segments, which is necessary for financial risk estimation.

The estimation accuracy is analyzed by applying mathematical modeling. For this purpose, we construct *control* time series such that the member structure of each of them corresponds exactly to the evolution model and to an arbitrarily given random vector of admissible parameters satisfying the restrictions of the models.

A *control* time series is input into the MPA algorithm, which estimates the vector θ and determines the estimate vector $\hat{\theta}$. Next, the exact vector $(\hat{\theta}_i - \theta_i)/\theta_i$ of relative errors in the estimate is determined; here, θ_i is the ith component of θ.

In the mathematical model, each *control* time series precisely corresponds to a certain precisely known vector θ consisting of a given vector of initial conditions, a given vector of model parameters satisfying the model restrictions, and a forecast vector computed using the evolution equations. Therefore, for each particular time series, the corresponding vector θ, the estimate vector, and, hence, the vector of relative errors in the estimate are precisely known to the estimation algorithm. Next, the dispersion characteristics of the estimation errors are determined using Monte Carlo computations.

For the STGARCH and MGARCH models considered later, the variances of the estimation errors were found to be nearly zero for suitably chosen parameters of the MPA algorithm.

The use of a *control* time series rather than an actual one (with the unknown correspondence of the latter to the evolution model used) and a diagnostic test is motivated by the need for analyzing the estimation accuracy in the MPA algorithm. Only *control* time series corresponding to arbitrarily given random vectors θ restricted by the model structure provide, in conjunction with the found vectors of relative errors, adequate data on the accuracy of the estimation algorithm.

Note that the analysis of the estimation accuracy by directly computing vectors of relative estimation errors differs from the traditional approach, in which the direct estimation error analysis is replaced by analyzing the conditions for the fulfillment of a prescribed significance level for the time series members with an actual (and unknown!) evolution model.

As justification of the MPA approach, we demonstrate a technically simple estimation procedure for STGARCH and MGARCH nonlinear evolution models with complex-structured parameter restrictions, in which case the traditional approach is hardly applicable because of the fundamental and technical difficulties mentioned above.

The implementation of the method proposed would considerably facilitate the analysis and forecasting of time series for complex nonlinear evolution models.

This Chapter, Division 4 is organized as follows. Section 7.10 describes the fundamentals of the MPA method. In Sects. 7.11 and 7.12, the MPA algorithm is used to numerically estimate the parameters of nonlinear GARCH models, such as STGARCH (nonlinear GARCH model) and MGARCH (multivariate GARCH model), in the BEKK specification. The conclusions are given at the Sect. 7.13.

7.10 Fundamentals of the Method

The MPA algorithm effectively estimates vectors of high dimensions (several hundreds) and does not require the determination of an extremum of the objective function (in this case, the likelihood function). The MPA algorithm is based on polynomial approximation of the conditional expectation vector. Therefore, the MPA algorithm is a polynomial approximation of the nonlinear least squares method.

A fairly general model of a time series is given by expressions (2.1) and (2.2):

$$h_t = F(t, h_{t-1}, \ldots, h_{t-q}, r_{t-1}^2, \ldots, r_{t-p}^2, \theta_0), \tag{2.1}$$

where $t = 1, \ldots, T$; h_t is the conditional variance of the time series member indexed by t (with respect to information fixed at time $t-1$); $r_{t-1}^2, \ldots, r_{t-p}^2$ are the squares of the values observed at the corresponding discrete times; q and p are given integers; $F(\ldots)$ is a given function; and θ_0 is the m-dimensional vector of unknown parameters in the domain defined by the model restrictions. The components of the total $n \times 1$ vector θ of unknown parameters include the unknown initial conditions $h_0, \ldots, h - q + 1$, the vector θ_0, and the forecast values $h_{T+1}, h_{T+2}, \ldots, h_{T+z}$ of h_t at the future stock market closing times $T + 1, \ldots, T + z$. Thus, $q + m + z = n$.

The observed vector R_T, that is, the time series $r_1, \ldots, r_T$, has the scalar components

$$r_t = h_t^{1/2} V_t, \tag{2.2}$$

where V_t is a sequence of independent identically distributed random variables with zero expectation and unit variance. Their distribution is arbitrary and is generated by a standard software program.

If the elements of the time series are vectors, then h_t in (2.1) is replaced by the conditional covariance matrix H_t and $h_t^{1/2}$ in (2.2) is replaced by $H_t^{1/2}$.

The goal is to estimate the vector θ, namely, construct an estimate $\hat{\theta}(R_T)$ of θ as a function of the components of R_T.

The MPA algorithm as applied to the estimation of θ can be schematically represented as the following sequence of steps.

Step 1 Define the domain Ω_θ of admissible parameters by applying two stages of Monte Carlo simulations.

In stage 1, all the components θ_i of θ are assumed to be independent random variables with the a priori structure $\theta_i = c_0(i) + c_1(i)\xi_i$, where c_0 and c_1 are a priori constants and ξ_i are independent random variables uniformly distributed on the interval [0, 1].

In stage 2, the random set of vectors θ obtained in stage 1 is "screened". The resulting set Ω_θ consists of the vectors that pass the screening procedure and satisfy a priori restrictions.

As a result, the set found in stage 1 is compressed and the resulting set accurately takes into account the restrictions and relations imposed on the

parameters by the model. Of course, after stage 2, the components of each random vector θ are no longer statistically independent.
Later in this section, in the parameter estimation for the STGARCH model (problem 1) and the MGARCH model (problem 2), the number of random vectors in the set after stage 1 ranges from several thousands (problem 1) to several hundreds of thousands (problem 2). After stage 2, in both problems, the resulting domain (Ω_θ) contains several hundreds of random vectors satisfying a priori constraints.

Step 2 Construct the domain $\Omega(R_T + z)$ of admissible time series in which T members are observed, while z members are not observed but forecasted (can be calculated). Each element of this domain is a $(T + z) \times 1$ vector whose components $r_1, \ldots, r_T, r_{T+1}, \ldots, r_{T+z}$ are obtained from (2.1) and (2.2) if θ is an element of Ω_θ. Each of the numbers $r_1, \ldots, r_T$ is consecutively computed using Eqs. (2.1) and (2.2) with a new random variable V_t generated by the standard program. The forecast numbers $r_{T+1}, \ldots, r_{T+z}$ are computed according to (2.1) with the use of random variables V_t for the construction of $r_{t-1}^2, \ldots, r_{t-p}^2$. In the course of constructing $\Omega(R_T + z)$, the integers T and z are given, while the vector θ runs over all the elements of Ω_θ.

Step 3 Assume that L is a given integer such that T/L is also an integer. Construct the domain $\Omega(R_T, T/L)$ of compressed admissible time series corresponding to the model. The goal is to reduce the dimension of the input matrix of the MPA algorithm in order to reduce the required storage. For this purpose, the time series with T elements is divided into nonoverlapping segments, each containing L consecutive elements. Each element of $\Omega(R_T, T/L)$ is the sum of the time series elements contained in an segment. As a result, the original time series $r_1, \ldots, r_T$ is replaced by a compressed one with the elements denoted by $g_1, \ldots, g_{T/L}$. The vector with these components is denoted by $G_{T/L}$.

Step 4 Construct the domain $\Omega_V(T/L, d)$ of basic vectors, each being an admissible input vector in the MPA algorithm.
Specifically, given a positive integer d, let $a_1, \ldots, a_{T/L}$ be nonnegative integers that solve the inequality $a(1) + \ldots + a(T/L) \le d$. The number of such solutions is denoted by $m(d, T/L)$ and is given by the following recurrence formula, which is proved by induction:

$$m(d, T/L) = \frac{m(d-1, T/L) + (T/L + d - 1) \cdots T/L}{d!}, m(1, T/L) = T/L.$$

As d increases, $m(d, T/L)$ increases quickly. For example, at $T/L = 6$,
$d \cdot \cdots \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdots 5 \cdots 6 \cdots 7 \cdots 8$
$M(d, T/L) \cdot 6 \cdot 27 \cdot 83 \cdot 209 \cdot 461 \cdot 923 \cdot 1715 \cdot 3002.$
Define a vector $W_{T/L}(d)$ of dimension $m(d, T/L) \times 1$ with components $w_1, \ldots, w_{m(d,T/L)}$. These components are all possible products of powers of $g_1, \ldots, g_{T/L}$ and have the form $g_1^{a(1)} \ldots g_{T/L}^{a(T/L)}$. The domain $\Omega_V(T/L, d)$

is the set of basic vectors $V(d, T/L)$ of dimension $(n+m(d, T/L)) \times 1$ that have the form $V(d, T/L) = \| \theta \cdot W_{T/L}(d) \|$.
By the definition of $W_{T/L}(d)$, it is clear that if $d_2 > d_1$, then we have the vector relation

$$W_{T/L}(d_1) \in W_{T/L}(d_2). \tag{2.3}$$

The random vectors $V(d, T/L)$ and their statistical characteristics, that is, the vector $\bar{V}(d, T/L)$ and the matrix $C_V(d, T/L)$, which are determined in step 5, describe the statistical relation between the observation vector $G_{T/L}$ and the components of θ, specifically, its parts—the unobserved vector of initial conditions and the vector of forecast values $r_{T+1}, \ldots, r_{T+z}$.
This statistical relation is used later to construct approximations to an optimal estimate vector and to the estimation error covariance matrix.

Step 5 Solve Eqs. (2.1) and (2.2) repeatedly with various random vectors $\theta \in \Omega_\theta$ generated using the statistical mechanism described above. The Monte Carlo approach is used to determine approximations to the a priori first and second statistical moments of $V(d, T/L)$, namely, the expectation vector, which is approximately equal to $E(V(d, T/L))$, and the matrix $C_V(d, T/L)$, which is approximately equal to the covariance matrix $E((V(d, T/L) - E(V(d, T/L)))(V(d, T/L) - E(V(d, T/L)))^T$. Step 5 implements the learning of the algorithm. As a result, the latter is adjusted to solve the particular problem given by Eqs. (2.1) and (2.2).

Step 6 Let the solution of the estimation problem with given d, T/L and a fixed vector $G_{T/L}$ be an $n \times 1$ vector $\hat{\theta}(W_{T/L}(d))$, which is an approximate estimate of the conditional expectation $E(\theta | G_{T/L})$ and has the form

$$\hat{\theta}(W_{T/L}(d)) = E(\theta) + \Lambda_{T/L}(d)(W_{T/L}(d) - E(W_{T/L}(d)), \tag{2.4}$$

where $\Lambda_{T/L}(d)$ is a matrix of size $n \times m(d, T/L)$ satisfying the equation

$$\Lambda_{T/L}(d) Q_{T/L}(d) = L_{T/L}(d),$$

$$L_{T/L}(d) = E(\theta - E(\theta))(W_{T/L}(d) - E(W_{T/L}(d))^T),$$

$$Q_{T/L}(d) = E((W_{T/L}(d) - E(W_{T/L}(d))(W_{T/L}(d) - E(W_{T/L}(d))^T).$$

The estimation error covariance matrix $C_{T/L}(d)$ is given by the formula

$$C_{T/L}(d) = C(0) - L_{T/L}(d) Q_{T/L}(d)^{-1} L_{T/L}(d))^T, \tag{2.5}$$

where

$$C(0) = E((\theta - E(\theta))(\theta - E(\theta))^T.$$

The vectors $E(\theta)$ and the matrices $L_{T/L}(d)$, $Q_{T/L}(d)$ formally involved in (2.4) and (2.5) are approximated by the corresponding blocks of the vector $\bar{V}(d, T/L)$ and the matrix $C_V(d, T/L)$. Equality (2.4) gives estimates for the components of θ, including the forecast values $h_{T+1}, h_{T+2}, \ldots, h_{T+z}$, since it takes into account the statistical relation between this vector and the observation vector $W_{T/L}(d)$. This relation is established for the second stochastic moments—the matrices $L_{T/L}(d)$ and $Q_{T/L}(d)$, whose approximations are obtained using the Monte Carlo method in step 5.

For the problem of estimating θ, formula (2.4) provides a solution that is optimal in the mean square sense on the set of linear combinations of the components of $W_{T/L}(d)$. This assertion follows from the matrix inequality

$$C_{T/L}(d) \le C, \tag{2.6}$$

where C is the estimation error covariance matrix obtained if the estimate vector is arbitrary [if the vector $E(\theta)$ and the weight matrix $\Lambda_{T/L}(d)$ in (2.4) are replaced by an arbitrary vector and an arbitrary matrix of suitable sizes].

Formulas (2.4) and (2.5) give an explicit approximate solution of the nonlinear estimation problem, namely, expressions for $\hat{\theta}(W_{T/L}(d))$ and $C_{T/L}(d)$.

However, these formulas are poorly implementable, since they involve matrix inversion; that is, the matrix equation satisfied by $\Lambda_{T/L}(d))$ has to be solved. Based on the observation decomposition principle, we construct a recurrence process such that not the entire vector $W_{T/L}(d)$ but rather only its single (current) component is used in each step in formula (2.4). The recurrence is implemented by sequentially running over the components of $W_{T/L}(d)$.

The vector $\bar{V}(T/L, d)$ of dimension $(n+m(d, T)) \times 1$ and the matrix $C_V(d, T/L)$ of dimension $(n + m(d, T)) \times (n + m(d, T))$ found for the component w_1 are used as the initial conditions for this process.

The process consists of $m(d, T/L)$ computing cycles, and the above dimensions are reduced at each of them. At the last cycle, the dimensions are n and $n \times n$, respectively. This cycle produces the estimate vector $\hat{\theta}(W_{T/L}(d))$ and the covariance matrix $C_{T/L}(d)$, satisfying (2.4) and (2.5).

The computation of the elements of $C_{T/L}(d)$ provides a method for preliminary observability analysis of the estimated parameters for the given model. The recurrence computations do not require matrix inversion and indicate situations where the current component of $W_{T/L}(d)$ is close to a linear combination of the preceding components, in which case $Q_{T/L}(d)$ is close to a singular matrix.

The algorithm is tuned by using Monte Carlo computations, which yield an approximate expectation vector $\bar{V}(T/L, d)$ and an approximate covariance matrix $C_V(d, T/L)$. Therefore, the algorithm takes into account a priori information on the stochastic structure of the components of the entire set of admissible vectors $R(T)$ and the corresponding vectors $W_{T/L}(d)$.

It should be emphasized that these vectors arise at all possible realizations of random vectors θ in Ω_θ. The above tuning is the price paid for the effective solution produced by the MPA algorithm for the nonlinear estimation problem. This is a key

difference of the MPA algorithm from, for example, the standard Kalman filter, which is intended for linear estimation problems. The same circumstance distinguishes the MPA algorithm from numerous attempts to extend the Kalman filter to nonlinear filtering problems.

Except for the computational errors, the estimates produced by the algorithm do not involve any additional errors (for example, those associated with the linearization of nonlinear functions). Therefore, it should be expected that the a priori dispersion characteristics of the estimated parameters are always higher than the calculated a posteriori dispersion characteristics, which substantiates the use of iteration.

The conditional expectation vector $E(\theta|G_{T/L})$ is optimal in the mean square sense. The estimation error covariance matrix corresponding to this estimate vector is denoted by $C(T/L, min)$. Consider a sequence of increasing integers d:1, 2, :, k, ...
It follows from (2.3) and (2.5) that as d increases, the expected estimation errors decrease in the sense of the sequence of matrix inequalities

$$C_{T/L}(1) \geq C_{T/L}(2) \geq ... \geq C_{T/L}(k) \geq ... \geq C(T/L, min), \tag{2.7}$$

which are bounded on the right. Moreover, all the estimate vectors corresponding to different integers d satisfy (2.4). Therefore, they are approximately optimal in the mean square sense. Taking into account this fact and inequalities (2.7), we can expect that as d increases, $E(\theta|G_{T/L})$ can be arbitrarily accurately approximated by the vector $\hat{\theta}(W_{T/L}(d))$ with the corresponding integer d. Indeed, assume that $E(\theta|G_{T/L})$ is a continuous function of the components of $G_{T/L}$ on a closed bounded domain $S \in R^{T/L}$. Then the Stone–Weierstrass theorem [18] implies that there exists a sequence of polynomials in the components of $G_{T/L}$ that converges uniformly to $E(\theta|G_{T/L})$ on S.

For any particular problem, no method is available that provides an a priori choice of d required for achieving an estimate of prescribed accuracy. No such methods are available for nearly all multistep computational processes, for example, for Newton's method. That is why we need a sequence of trial computations with several integers d. The numerical results presented ahead confirm that, with trial computations, the estimate vectors $\hat{\theta}(W_{T/L}(d))$ converge rapidly not only to the conditional expectation vector $E(\theta|GT/L)$ but also to the vector θ of unknown parameters. The formulas involved in the recurrence algorithm and their substantiation can be found.

7.11 Parameter Estimation for a Nonlinear STGARCH Model

Consider a nonlinear STGARCH model that has heteroskedasticity and smoothes the jumps in the time series elements (smooth transition) by using the logistic function. The model has the form

$$r_t = h_t^{1/2}\xi_t,$$

where h_t is the conditional variance, ξ_t is a sequence of independent identically distributed random variables with zero mean and unit variance, and

$$h_t = \theta_1 + \theta_2 r_{t-1}^2 + (\theta_3 + \theta_4 r_{t-1}^2)(1 + exp(-\theta_5 r_{t-1} - c))^{-1} + \theta_6 h_{t-1}.$$

The components of θ must satisfy inequalities ensuring that the conditional variances are positive and the elements of the time series have unconditional variances $\theta_1 > 0, \theta_1 + \theta_3 > 0, \theta_2 > 0, \theta_2 + \theta_4 > 0, \theta_6 > 0, \theta_5 > 0$, $\theta_2 + \theta_6 < 1$, and $\theta_2 + \theta_4 + \theta_6 < 1$.

According to the above publications, while constructing the domain Ω_θ (step 1), we can set $c_0(1) = 0$, $c_1(1) = 1$, $c_0(2) = 0$, $c_1(2) = 1$, $c_0(3) = -1$, $c_1(3) = 2$, $c_0(4) = -1$, $c_1(4) = 2$, $c_0(5) = 0$, $c_1(5) = 3$, $c_0(5 + i) = 0$, and $c_1(5 + i) = 1$ for $1 \le i \le 7$.

The goal is to estimate $n = 12$ parameters, of which six $(\theta_1, \ldots, \theta_6)$ are model parameters, one (θ_7) is the parameter of initial conditions $h(0)$, and five parameters $(\theta_8, \ldots, \theta_{12})$ are forecast values of the conditional variances $h(T+1), \ldots, h(T+5)$. Thus, $q = p = 1$, $m = 6$, and $z = 5$. Let $T = 330$ and $L = 55$.

For 10 arbitrary 12-dimensional vectors θ chosen from Ω_θ, the MPA algorithm was used to determine the relative estimation errors. The order of these errors turned out to be identical for all the vectors. The following table presents the relative errors $(\theta_i - \hat{\theta}_i)/\theta_i$ in the estimate of θ_i and the components of the estimate vector $\hat{\theta}$ for one of these vectors. Here, d is the degree of the approximating polynomials and $m(d, T/L)$ is the number of terms in the approximating polynomials.

$d = 4, m(d, T/L) = 209$

i	$\frac{(\theta_i - \hat{\theta}_i)}{\theta_i}$	$\hat{\theta}_i$
1	-2.810×10^{-6}	9.473×10^{-1}
2	3.044×10^{-6}	7.756×10^{-1}
3	-3.966×10^{-6}	9.452×10^{-1}
4	5.093×10^{-6}	-6.594×10^{-1}
5	6.170×10^{-6}	1.4983×10^{-1}
6	1.825×10^{-6}	6.745×10^{-4}
7	1.280×10^{-6}	3.151×10^{-1}
8	-8.053×10^{-6}	1.357×10^{-0}
9	-6.717×10^{-6}	2.062×10^{-0}
10	-7.470×10^{-6}	2.383×10^{-0}
11	-6.076×10^{-6}	2.024×10^{-0}
12	-6.530×10^{-6}	1.751×10^{-0}

For $d = 1, 2, 3$, the modeling results have shown that the relative estimate errors are large (specifically, on the order of 1). For $d = 4$, however, the quantitative changes lead to an abrupt qualitative change and the relative estimate errors become close to zero. For given d, t, L, the components of the estimate vector were computed within several seconds.

7.12 Parameter Estimation for a Multivariate MGARCH Model

A multivariate MGARCH model is used if each element of a time series is a vector with components equal to the logarithms of the price ratios for several assets at the current market closing time t. Such a situation arises, for example, in optimal portfolio construction.

In financial mathematics, MGARCH models in the BEKK specification are used. Here we estimate the parameters of the BEKK model presented in [16, 19–21]. At time t, the number of different assets is s and they differ from each other by the index $i = 1, \ldots, s$. Let $s = 5$, and let the length of the vector time series be $T = 250$.

Consider a stochastic vector process r_t, $t = 1, \ldots, 250$, of dimension $s \times 1$ such that $E(r_t) = 0$. Let F_{t-1} denote the information set generated by the observed vector series including the time $t - 1$. According to the above-mentioned publications, for a fixed F_{t-1}, the vector r_t is assumed to satisfy

$$r_t = H_t^{1/2} \xi_t, \tag{4.1}$$

where $H_t = (H_t^{1/2})^T H_t^{1/2}$ ($H_t = [h_{ijt}]$) is the $s \times s$ conditional covariance matrix of the components of r_t and ξ_t is a sequence of independent identically distributed random vectors such that $E(\xi_t \xi_t^T) = I$. These assumptions define a standard multivariate GARCH model with no linearly dependent structures for r_t. In finance, r_t is interpreted as the vector of the differences between the logarithms of the asset prices of s asset types. The symmetric BEKK model [19] is defined by the relation

$$H_{t+1} = C^T C + A^T r_t r_t^T A + B^T H_t B, \tag{4.2}$$

where A, B, C are $s \times s$ matrices whose elements are unknown parameters.

Assume that the matrices $H_t^{1/2}$ and $(H_t^{1/2})^T$ are defined via Cholesky factorization and $H_t^{1/2}$ is an upper triangular matrix. The matrix $H_t^{1/2}(0)$ is an initial diagonal matrix whose elements are unknown along with the elements of A, B, C.

The MPA algorithm is used to estimate the matrix of initial conditions and the matrices of the model, altogether $n^2 + n^2 + n^2 + n = 80$ parameters. The following a priori restrictions are imposed on A, B, C, $H_t^{1/2}(0)$ when their elements are estimated:

1. The elements of all the matrices belong to the interval $[-1/2, 1/2]$ and are uniformly distributed.
2. The symmetric BEKK model has time-invariant conditional covariances if and only if the eigenvalues of the matrix $A \bigotimes A + B \bigotimes B$ are less in absolute value than 1; here, $\bigotimes$ denotes the Kronecker product of matrices [19].

We construct an a priori domain $\Omega(\theta)$ of random variables that are the components of the 80-dimensional parameter vectors of the estimated matrices subject to restrictions 1 and 2. Using Eqs. (4.1) and (4.2), the MPA algorithm constructs time series for

each of the 80-dimensional vectors and produces $\Omega_\theta(R_T)$—the space of admissible time series with parameter vectors satisfying a priori restrictions 1 and 2. For any given 80-dimensional vector from $\Omega(\theta)$, the MPA algorithm uses the corresponding (250×5)-dimensional vector of observations to determine an 80-dimensional estimate vector that approximates the conditional expectation vector. The latter is approximated more accurately if the degree d of the approximating polynomials is higher and, accordingly, their length $m(d, T/L)$ is longer.

For $L = 200$ and for an arbitrary 80-dimensional vector from $\Omega(\theta)$, the following tables give the relative errors of the estimates (left column) and the estimates themselves (right column) for the elements of A, B, C, $H_t^{1/2}(0)$ for various d.

For all the components of the 80-dimensional vectors, the relative errors of the estimates are similar to each other. For this reason, the results are given only for the first five and last five components of these vectors.

$d = 1, m(d, T/L) = 6$

i	$\frac{(\theta_i - \hat{\theta}_i)}{\theta_i}$	$\hat{\theta}_i$
1	-1.316×10^{1}	-9.663×10^{-2}
2	4.214×10^{-2}	-1.244×10^{-1}
3	1.774×10^{-1}	-1.258×10^{-1}
4	-1.351×10^{0}	-1.209×10^{-1}
5	5.998×10^{-1}	-9.619×10^{-2}
76	2.922×10^{-1}	6.831×10^{-1}
77	2.755×10^{-2}	7.5369×10^{-1}
78	-2.097×10^{-3}	7.617×10^{-1}
79	-8.845×10^{-1}	8.087×10^{-1}
80	-2.441×10^{-1}	1.005×10^{0}

$d = 2, m(d, T/L) = 27$

i	$\frac{(\theta_i - \hat{\theta}_i)}{\theta_i}$	$\hat{\theta}_i$
1	-1.466×10^{1}	-1.06×10^{-1}
2	1.216×10^{-1}	-1.140×10^{-1}
3	1.506×10^{-1}	-1.299×10^{-1}
4	-1.066×10^{0}	-1.063×10^{-1}
5	7.263×10^{-1}	-6.577×10^{-2}
76	3.864×10^{-1}	5.92208×10^{-1}
77	1.715×10^{-1}	5.922×10^{-1}
78	-5.517×10^{-2}	8.021×10^{-1}
79	-7.169×10^{-1}	7.368×10^{-1}
80	-1.166×10^{-1}	9.019×10^{-1}

$d = 3, m(d, T/L) = 83$

i	$\frac{(\theta_i - \hat{\theta}_i)}{\theta_i}$	$\hat{\theta}_i$
1	-1.003×10^{1}	-7.523×10^{-2}
2	4.352×10^{-1}	-7.335×10^{-2}
3	1.046×10^{-1}	-1.369×10^{-1}
4	-1.125×10^{0}	-1.093×10^{-1}
5	5.029×10^{-1}	-1.194×10^{-1}
76	1.397×10^{-1}	8.302×10^{-1}
77	3.460×10^{-1}	5.068×10^{-1}
78	-5.469×10^{-2}	8.017×10^{-1}
79	-6.423×10^{-1}	7.048×10^{-1}
80	-1.672×10^{-1}	9.428×10^{-1}

$d = 4, m(d, T/L) = 209$

i	$\frac{(\theta_i - \hat{\theta}_i)}{\theta_i}$	$\hat{\theta}_i$
1	-4.192×10^{-6}	-6.820×10^{-3}
2	-1.079×10^{-7}	-1.298×10^{-1}
3	1.349×10^{-6}	-1.529×10^{-1}
4	-5.584×10^{-6}	-5.143×10^{-1}
5	5.446×10^{-7}	-2.403×10^{-1}
76	5.602×10^{-6}	9.651×10^{-1}
77	2.066×10^{-6}	7.750×10^{-1}
78	1.207×10^{-6}	7.601×10^{-1}
79	-1.160×10^{-6}	4.291×10^{-1}
80	3.969×10^{-6}	8.077×10^{-1}

It can be seen that the relative errors for $d = 4$ are close to zero. For given d, T, L, the time required for computing the estimates does not exceed several seconds.

7.13 Conclusions

The MPA algorithm is an effective high-speed tool for accurate estimation of the parameters of control time series generated by the STGARCH and MGARCH models. The main advantages of the method are as follows:

1. The MPA method constructs approximations to the conditional expectation vectors for the unknown model parameters. Therefore, the approximations to the estimate vector are optimal in the mean square sense. The minimization of the mean square errors in the estimate is a transparent criterion, because it is directly related to the estimate errors rather than to an abstract likelihood function, for

which the estimate vector for its maximum exhibits good properties only for a large sample size.
2. In contrast to the usual approach used in statistical analysis, the asymptotic accuracy of the estimate is not a function of the sample size but rather a function of the sum d of the degrees of the approximating polynomials at a constant sample size. The Stone–Weierstrass theorem (which is the multidimensional analog of Weierstrass's approximation theorem) implies that the absolute value of the estimation error decreases uniformly with increasing d. The modeling results obtained for GARCH models of two types (the tables of relative estimate errors) have demonstrated that the estimates have nearly zero errors starting with $d = 4$. A similar situation occurs in models of other types (e.g., A-FIGARCH).
3. The MPA method as applied to parameter estimation for control time series has the following key advantages over the maximum likelihood method: There is no need to guess a good first approximation or global (not local!) maximizers; the parameters (coefficients) of the model equations are estimated simultaneously with the components of the vector of initial conditions and the forecast conditional variances; and the method does not involve cumbersome calculations of gradients and Hessians.

References

1. Rubin R, et al (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge
2. Cappe O, Moulines E, Ryden T (2005) Interference in hidden Markov models. Springer, New York
3. Baum E, Petrie T (1966) Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37(6):1554–1563
4. Borodovsky M, Ekisheva S (2006) Problems and solutions in biological sequence analysis. Cambridge University Press, Cambridge
5. Dzyadyk VK (1977) Introduction to the theory of polynomial uniform functional approximation (Science). Nauka, Moscow
6. Schiryaev AN (1996) The probability, 2nd edn. Springer, New York
7. Boguslavskiy JA (2000) Analysis on estimation accuracy of Markov model transition probability. Bulletin of the Russian Academy of Sciences. Theory and Control Systems, No. 1
8. Boguslavskiy JA (2003) Direct computational method for the task of optimal linear operating speed. Bulletin of the Russian Academy of Sciences. Theory and Control Systems, No. 3
9. Boguslavskiy JA (2005) Autonomous aircraft navigation through sight angle data. Bulletin of Computer and Information Technologies, No. 3, Machine Engineering
10. Rozanov YA (1989) Theory of probability, stochastic processes and mathematical statistics (Science). Nauka, Mascow
11. Liptser RS, Shiryaev AN (1974) Statistics of stochastic processes (Science). Nauka, Moscow
12. Ventsel ES, Ovcharov LA (1991) Theory of stochastic processes and its engineering applications (Science). Nauka, Moscow
13. Albert A (1977) Regression, pseudo inversion and recurrent valuation. Physmatlit, Moscow
14. Boguslavskiy JA (1996) Bayesian estimations of nonlinear regression and related questions. Bulletin of the Russian Academy of Sciences. Theory and Control Systems, No. 4
15. Engle RF (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50:987–1007

16. Silvernnoinen A, Terasvirta T (2008) Multivariate GARCH models. SSE/EFI Working paper series in economics and finance, 2008
17. Zaks S (1975) Theory of statistical inferences (World). Wiley, New York
18. Timan AF (1960) Approximation theory of real-approximate functions of real variable (Science). Macmillan, New York
19. Engle R, Kroner K (1995) Multivariate simultaneous generalized ARCH. Econometric Theor 11:122–150
20. Hammoudeh S, Yuan Y, McAleer M (2010) Exchange rate and industrial commodity volatility transmissions, asymmetries and hedging strategies. CIRJE-F-741, 2010
21. Bauwens L, Sebastien Laurent S, Rombouts J (2003) Multivariate GARCH models. J Appl Econometrics 21(1):79–109

Chapter 8
Designing Motion Control to a Target Point of Phase Space

8.1 Introduction

Numerous applications consider the ultimate goal of the object to be motion control, a situation where, at some time point, the current phase coordinates of the object become aligned with the desired phase coordinates. Thus, an important task to handle when developing computer-controlled information complexes of modern mobile objects is to develop algorithms and computer programs capable of "carrying" the mobile object to a target point with specified phase coordinates.

Currently, there are numerous pilot and mockup projects of low-thrust engines designed to ensure space maneuvering with low propellant consumption. Nevertheless, there are no publications where the minimum-time earth satellite vehicle (ESV) near-circular orbit correction problem would be considered using the low-thrust engines.

(Note that the minimum time of the motion all powered by the low-thrust engines corresponds to the motion to a target point optimized for minimum propellant consumption). Here we demonstrate the use of the simple search and polynomial approximation techniques as numerical design tools for time-optimal control implementation. When applied to a special case of two low-thrust engines in on–off mode, the optimized control lets us perform, in minimum time, the near-circular ESV orbit correction—or, alongside with orbit correction, adjust, in minimum time, the in-orbit ESV position. The last task is a commonplace occurrence when two ESVs are in the proximity stage prior to docking.

The methods are of specific interest as applied to the air- and spacecraft (ASC) governed by load factor and roll orientation parameters. In this case, the methods numerically solve a problem of designing the proper control to "carry" the ASC to a target point of phase space.

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_8,

8.2 Setting Boundary Value Problems and Problem-Solving Procedures

Let a controlled object be represented by the equation

$$dx/dt = f(x, u), \tag{2.1}$$

where $x \in R^n$ is the status vector of the dynamic system, u is the control vector constrained by condition $u \in U$, $f(x, u)$ is the dimension-$(n \times 1)$ vector-function, and $x(0) = x_0$ is the prescribed initial condition vector.

When modeling the control of the motion to a given phase-space point, the vector-function $u(t)$ solves a boundary value problem, that is, operates to fulfill the condition

$$x(T) = x_T, \tag{2.2}$$

where x_T is the prescribed vector pertaining to the attainability domain relevant to the vector x_0.

The following treatment explains boundary value problems in detail.

This boundary value problem arises when designing the control configuration to provide minimum-time control functionality, which means the "as soon as possible" (ASAP) principle applied to minimize the T value in (2.2).

The solution of the boundary value problem reduces to the determination of a vector of unknown parameters

$$\theta^T = \|\theta_1, \ldots, \theta_n\|^T;$$

this vector determines the structure of the control vector. It is assumed next that there exists an a priori domain $\Omega_\theta \in R^n$ (typically, a parallelepiped in R^n), which may (or must) contain the vector of parameters: $\theta \in \Omega_\theta$.

It was found that, within the context of the motion control problems, the vector θ is precisely the root vector of some set of nonlinear algebraic equations. Thus, a numerical procedure to solve a boundary value problem is considered established once the efficient procedure is established to solve numerically a relevant set of nonlinear algebraic equations of the form

$$F(\theta) = Y, \tag{2.3}$$

where Y is the dimension-$(n \times 1)$ vector defined as the context of the boundary value problem dictates, and $F(\theta)$ is the prescribed $n \times 1$ vector-function with the condition-specific structure as the boundary value problem dictates.

A characteristic property of the boundary value problem that arises when designing the control configuration to provide motion to a given phase-space point is that the vector-function $F(\theta)$ in (2.3) is assigned implicitly, as a result of the numerical solution of a set of differential equations.

In the boundary value problem

$$u = u(t, x, P),$$

P is the conjugate variable vector (see below for detail), and

$$|P(0)| = 1, \quad P(0) = \varphi(\theta(n-1)), \quad \theta(n-1)^T = \|\theta_1, \ldots, \theta_{n-1}\|,$$
$$Y = x(\theta_n),$$

$\varphi(\ldots)$ is the prescribed vector-function; for every θ, the vector Y in (2.3) is found by numerical integration of the equations

$$dx/dt = f(t, x, u(t, x, P)),$$

$$dP/dt = \Psi(t, x, P, u(t, x, P)),$$

over a domain $[0, n]$, where $\Psi(\ldots)$ is the known vector-function.

For some types of boundary value problems, the control vector components comprise the piecewise continuous functions of time and belong to the relay control class. So, enormous difficulties, both conceptual and computational, arise when attempting to find the Jacobian—a $\partial F/\partial\theta$—partial derivative matrix $F(\theta)$ vector components to vector components) in a specific case of the implicitly assigned vector-function $F(Y)$.

This fact prevents us from using the conventional modifications of Newton's method as a tool to solve the set of equations (2.3) numerically. The Newtonian procedure is simply a way to develop the iteration process

$$\theta^i = \theta^{i-1} - G(\partial F(\theta^{i-1})/\partial\theta)^{-1}(F(\theta^{i-1}) - Y),$$

where θ^i is an approximate solution vector for the set (2.3) at the ith iteration step, and G is a positive definite matrix.

Along with the need to calculate the inverse(s) of the Jacobian matrices, the Newtonian procedure typically requires a selection (from a priori considerations) of a zeroth-order approximation vector (first guess) θ^0 closely adjacent to the root vector of the set (2.3).

Next, a combination of a simple search technique and a polynomial approximation technique as applied to the inverse vector-function $F^{-1}(Y)$ is used to find an approximate solution of the set (2.3). The simple search procedure is to cover the a priori domain with a family of points (vectors ϑ), then calculate the residual vectors $F(\vartheta) - Y$, and finally determine the vector ϑ^o:

$$\vartheta^o = \arg\min_{\vartheta\in\Omega_\theta} |F(\vartheta) - Y|.$$

The vector ϑ^o is used for the initial approximation to the root vector of the set (2.3) and comprises a center of some domain $\Omega_\theta^* \in \Omega_\theta$. The refined approximation to the root vector is then found over the points of the * using a polynomial approximation technique.

The procedure affords the asymptotic representation for $\hat{F}^{-1}(Y)$, which is an integral counterpart of the multidimensional Taylor series.

The vector-weighting factors contributing to the vector series representation are found as the first and second statistical moments of the $F(\theta)$ vector-function components to an integer power, calculated on the hypothesis that the components of an unknown vector θ comprise random values evenly distributed over Ω^*. Determining these factors therefore requires calculation of the n-order integrals over the domain Ω^*; sometimes, it is a computation-intensive task.

A need for such computational work is commonly justified by the reasons that a two-point boundary value problem is to be solved per single computation cycle for all vectors Y belonging to the same a priori domain Ω_Y [the domain is defined by Eq. (2.3) with the Ω^* domain being prescribed].

Theoretically, the uniform convergence is guaranteed by the underlying theorem of the polynomial approximation method as set forth above. The practical experience as applied to the numerical solution of boundary value problem has shown, however, that computational difficulties exist, especially regularly arising at $d > 8 \div 10$ due to the ill-conditioned character of matrix algorithm to solve a set of linear algebraic equations (the process implemented here is to find vector coefficients of a power series).

If the Ω_θ^* domain is small, however, a small d value of about $3 \div 5$ is enough to reasonably minimize the errors during polynomial approximation of the root vector in (2.3)—only if the vector lies within Ω_θ^*. Then the above computational difficulties will be eliminated.

This is the reason why the following computational scheme was applied to solve the selected boundary value problems. The edges of a parallelepiped Ω_θ^* are divided into r equal parts; this results in the formation of r^n smaller parallelepipeds. In each of them, the polynomial approximation algorithm finds the estimated root vector with $d = 3 \div 5$. The estimated vector for which the sum of squared error (SSE) is minimized when solving the set (2.3) is refined via several iterations and becomes the output of the algorithm designed to find a solution of the set (2.3).

The described procedure to divide the domain Ω_θ^* into smaller subdomains is equally necessary in a special case where several root vectors of the set (2.3) are about to exist within Ω_θ^*.

The remaining sections of this chapter describe the principal scheme and computational process adapted to handle applied problems (i.e., minimum-time–oriented) using the simple search and polynomial approximation procedures.

8.3 Necessary and Sufficient Conditions for Time-Optimal Control

The Pontryagin maximum principle [1], when applied to real-time processes, expresses the necessary conditions for time-optimal control. Furthermore, for linear dynamic systems, it expresses the sufficient conditions for time-optimal control and affords a two-point boundary value problem for a set of differential equations satisfied by the status vector of the object as well as the conjugate variable vector. This boundary value problem is a case study of the situation where a vector-function of a set of nonlinear algebraic equations of form (2.3) is assigned implicitly.

Nondifferentiable and/or discontinuous functions represented by an optimum control vector in real time, with an allowance for control constraints, prevent us from applying the variety of gradient methods to find a numerical solution of a boundary value problem. These functions, however, have no effect on the computational algorithm of a modification of the polynomial approximation method applied to the inverse vector-function where no gradient calculation is required.

Note that it is the occurrence of such functions that prevents us from applying the two-point boundary value problem solution algorithms described in [2] with no restrictions, to implement them in optimum control design.

Let the simulation model of the control object be represented by the equation

$$dx/dt = Ax + b_1u_1 + \cdots + b_ku_k, \tag{3.1}$$

where n is a dimension-$(n \times n)$ matrix, $b_1, \ldots, b_k$ are dimension-$(n \times 1)$ vectors, and $u_1, \ldots, u_k$ are unknown scalar controls (functions of time constrained by

$$|u_i(t)| < 1, \quad i = 1, \ldots, k.$$

Then the conjugate variable vector $P \in R^n$ will be represented by the equation

$$dP/dt = -A^T P. \tag{3.2}$$

Necessary and sufficient conditions for the time-optimal linear control introduced in the form of the constraints $u_1^o, \ldots, u_k^o$, to transfer, in a minimum of time θ_n, a vector x from a fixed point $x(0)$ to a fixed point $x(\theta_n)$, state that the optimum scalar controls $u_1^o, \ldots, u_k^o$ must comply with the maximum principle (see [1] for details):

$$u_1^o, \ldots, u_k^o = \arg \max_{|u_1|<1,\ldots,|u_k|<1} P^T(b_1u_1 + \cdots + b_ku_k), \tag{3.3}$$

from which

$$u_1^o = u_1(P), \ldots, u_k^o = u_k(P).$$

Furthermore, the following condition must hold:

$$H(\theta_n) > 0, \tag{3.4}$$

where H is a Hamiltonian:

$$H = P^T(Ax + b_1u_1 + \cdots + b_ku_k).$$

Let $p_1^o, \ldots, p_n^o$ be the unknown components of the vector P(0). From the homogeneity of Eq. (3.2), it follows that the condition may be imposed on $P(0)$:

$$|P(0)| = 1. \tag{3.5}$$

Let us pass on to the new unknowns $\theta_1, \ldots, \theta_{n-1}$, connected to $p_1^o, \ldots, p_n^o$ via trigonometric formulas that ensure the fulfillment of the condition (3.5):

$$p_1^o = \cos(\theta_1),$$

$$p_2^o = \sin(\theta_1)\cos(\theta_2),$$

.....

$$p_{n-1}^o = \sin(\theta_1)\sin(\theta_2)\cdots\cos(\theta_{n-1}),$$

$$p_n^o = \sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{n-1}),$$

where $n > 1$. The a priori domain for the unknowns $\theta_1, \ldots, \theta_{n-1}$ is specified by inequations

$$0 < \theta_1, \ldots, \theta_{n-1} < 2\pi. \tag{3.6}$$

The corresponding vectors $P(0)$ evenly cover the surface of a unit ball in R^n, while the unknowns $\theta_1, \ldots, \theta_{n-1}$ are subjected to fixed-increment variations.

If we substitute $u_1(P), \ldots, u_k(P)$ for $u_1, \ldots, u_k$ in (3.1), (3.2), we arrive at a two-point boundary value problem where, for Eq. (3.2) and for a set of second-order differential equations

$$dx/dt = Ax + b_1u_1(P) + \cdots + b_ku_k(P), \tag{3.7}$$

Equation (2.3), n initial conditions x_0, and n terminal conditions x_T are defined. $n-1$ unknown parameters $1, \ldots, n-1$ combined with the time-optimal unknown time of motion $n = T$ create the vector of unknowns θ, with its components $\theta_1, \ldots, \theta_{(n-1)}, \theta_n$.

Suppose an a priori domain of existence Ω_θ is defined for the vector θ and a numerical integration method is selected for Eqs. (3.2), (3.7). Then, for any $\theta \in \Omega_\theta$,

the corresponding vector $x(\theta_n)$ can be found. We take it as vector Y; the functional relationship of Y and is found via numerical integration of Eqs. (3.2), (3.7).

Hence, the solution of the two-point boundary value problem to afford a time-optimal dynamic system control (3.1) was reduced to the solution of the equation of the form (2.3); that is, the inverse vector-function $F(Y)^{-1}$ should be determined given that the vector-function $F(Y)$ is defined implicitly, via numerical integration of Eqs. (3.2), (3.7). One can see from these equations that $F(Y)^{-1}$ is a continuous function. Therefore, the principal requirement is fulfilled, which claims the uniform convergence of polynomial approximation for the inverse vector-function $F^{-1}(Y)$.

Numerous examples of tackling the time-optimal control problems are given in [1] for cases where the dynamic system equations have an order of 2 or less. Approximate methods were developed [3] for more sophisticated problems where a nonlinear programming technique is implemented, so there is no need for solving the boundary value problem for Eqs. (3.2), (3.7).

Note that these methods are quite difficult to implement: The procedure involves consideration of various special cases of convergence and typically requires conversational programming system to be used. That is why it is often claimed that "the most accurate and precise numerical solutions of the 'calculus of variations' problems are associated with the solution of relevant systems; the successful treatments, however, were extremely scarce, despite the repeated attempts" [3, p. 115]. The two-point boundary value problem is a special case of the system considered in [3].

Later in this chapter, we will detail the calculation process by stage to produce the approximate solution of the boundary value problem stated above. The individual stages involve the repeated numerical solutions of the Cauchy problem for Eqs. (3.2), (3.7).

The conditions of the maximum principle given above are the necessary conditions and the sufficient conditions at one time. So, for any dimension-$(n-1\times 1)$ θ vector (the initial condition vector in Eq. (3.2)) as well as for any positive quantity θ_n, Eqs. (3.2), (3.7) determine the time-optimal controls when transferring, in a minimum of time θ_n, a prescribed initial condition vector x_0 to some terminal condition vector $x(\theta_n)$.

The last-mentioned vector is determined after joint integration of Eqs. (3.2), (3.7) given the initial conditions x_0, θ_{n-1} and the integration time θ_n.

It is well known from the optimal control theory that the Hamiltonian value amounts to a constant nonnegative quantity $H = const > 0$ over an entire optimum motion pathway, including the control discontinuity points (control sign reversal points).

The currently used numerical integration technique is feasible when solving the Cauchy problem provided that the above condition is met over an object motion pathway to a sufficient degree of accuracy for all allowable dimension-$(n-1)$ vectors θ as well as for the quantity θ_n.

8.4 The Stages of the Calculation Process

Stage 1. Upper-bound estimation of θ_n for prescribed vectors x_0, Y.

What is the way to get the upper-bound estimate for the quantity θ_n, the unknown minimum time to move a dynamic system from a point x_0 to the prescribed point Y. We take as an estimate ϑ_n, a time value attainable when using some reasonable—but, in general, far from being optimal—control satisfying the imposed constraints.

Let n of the time points $\vartheta_1, \ldots, \vartheta_{n-1}, \vartheta_n$ satisfy the inequalities

$$\vartheta_1, \ldots, \vartheta_{n-1}, \vartheta_n,$$

and let some control u_1 reverse its sign at these points, to amount to ± 1 or ∓ 1, the remaining $n-1$ control vanishing.

It is the explicit algebraic equations or numerical integration of the model dynamic system (3.1) with imposed boundary conditions that determine Π algebraic equations in unknowns $\vartheta_1, \ldots, \vartheta_{n-1}, \vartheta_n$.

$$x_o, \quad x(\vartheta_n) = x_T.$$

For these equations, a vector-function $F(\vartheta)$ is prescribed explicitly or implicitly: a vector Y, the prescribed vector Y, will be found via integration of (3.1). In the case studies of problems described ahead, a fundamental matrix $\Phi(\vartheta)$ of system (3.1) is known, and a set of algebraic equations is as follows [given $u1(0) = 1$]:

$$\Phi(\vartheta_n)x_0 + b_1 \sum_{i=1}^{n} (-1)^{i-1} \int_{\vartheta_{i-1}}^{\vartheta_i} \Phi(\vartheta_n - \vartheta) d\vartheta = x_T.$$

Similar sets will be obtained when we replace $u_1(0) = 1$ with $u_1(0) = -1$, u_1 with u_2, and so forth.

Solving such sets of equations approximately using a simple search technique (amplified with a polynomial approximation technique applied to inverse functions if necessary), we shall find $2n$ numbers n comprising the values of possible time points when a particular case of reasonable control will be terminated. Name the least of these numbers ϑ_n an upper-bound estimate for the quantity θ_n. The inequalities (3.6) given above, together with the inequality $\theta_n < \vartheta_n$, define an a priori domain $\Omega_\theta \in R^n$.

Stage 2. Approximate determination of the root vector of system (2.3) using a simple search technique.

Divide the edges of a parallelepiped Ω_θ into r equal parts; this results in the parallelepiped being covered with a grid of rn points. Take their coordinates as the components of the parameter vector θ corresponding to r^n numerical integrations of Eqs. (3.2), (3.7) given that $x(0)$ is a prescribed vector. The integrations will result in r^n difference vectors [the prescribed vector $x(T)$ minus the vectors x at times when

relevant integration(s) are terminated]. Then the components of the approximate root vector θ^* of set (2.3) are equal to the coordinates $\theta_1^*, \ldots, \theta_{n-1}^*, \theta_n^*$ of the point where the difference vector has the minimum length.

Stage 3. Refining the root vector of set (2.3) using a polynomial approximation technique. The vector θ^* is a center of a parallelepiped Ω^*. The subsequent calculation process was described in Sect. 8.2.

8.5 Near-Circular Orbit Correction in Minimum Practicable Time Using Micro-Thrust Operation of Two Engines

Let's employ the conceptual calculation scheme described in Sect. 8.1 in handling a problem of the time-optimum earth satellite vehicle (ESV) orbit correction. Suppose here that the satellite follows a near-circular orbit.

The difference between the actual ESV orbit and its circular (nominal) orbit is assumed to be small, so the acceleration induced by the Earth's gravitational field can be first power linearized using difference ESV coordinates (actual position less nominal position); note that the linearization is not mandatory when using a polynomial approximation technique with the algorithm common for linear and nonlinear systems. The actual ESV motion parameters comply approximately with known linear differential equations in a rotating Cartesian "local vertical local horizontal (LVLH) frame, or vehicle-centric system. These equations represent the "disturbed-in-small," near-circular-orbit ESV motion.

We shall find the equations by writing the rigid-body dynamic equations of motion (EOMs) referred to a Cartesian frame rotating with an angular velocity of $\omega = V/R$, where V is the nominal ESV velocity, and R is the length of the nominal ESV radius-vector.

The origin of the LVLH frame is precisely the nominal ESV position at the near-circular orbit, the x_1-axis being directed as the nominal ESV radius-vector, and the x_2-axis being directed as the ESV velocity vector. Two engines are used to adjust the orbit of the actual ESV. Due to thrust forces, the accelerations a_1, a_2 are created directly either as x_1, x_2 or in the opposite directions; the limit acceleration value is a_0. Then the instantaneous position coordinates s_1 and s_2 as well as the orbital velocity components v_1 and v_2 in the LVLH frame satisfy the differential equations of motion for the actual ESV:

$$ds_1/dt = v_1, \quad dv_1/dt = 3\omega^2 s_1 + 2\omega v_2 + a_1,$$
$$dv_2/dt = -2\omega v_1 + a_2, \quad ds_2/dt = v_2.$$

We shall conveniently call an independent variable (the quantity $f = \omega t$) the nominal ESV radius-vector rotation angle. Then the equations of motion will be as follows:

$$dx_1/df = x_2, \tag{5.1}$$

$$dx_2/df = 3x_1 + 2x_3 + bu_1, \tag{5.2}$$

$$dx_3/df = 2x_2 + bu_2, \tag{5.3}$$

where

$$x_1 = s_1, \quad x_2 = v_1/\omega, \quad x_3 = v_2/\omega, \quad b = a_0/\omega^2, \quad |u_1|, |u_2| < 1.$$

The equations for the conjugate variable vector components are as follows:

$$dp_1/df = -3p_2, \tag{5.4}$$

$$dp_2/df = -p_1 + 2p_3, \tag{5.5}$$

$$dp_3/df = -2p_2, \tag{5.6}$$

with the initial conditions

$$p_1^o = \cos(\theta_1), \quad p_2^o = \sin(\theta_1)\cos(\theta_2), \quad p_3^o = \sin(\theta_1)\sin(\theta_2).$$

The Hamiltonian of the problem is defined by the expression

$$H = p_1x_2 + p_2(3x_1 + 2x_3 + bu_1) + p_3(-2x_2 + bu_2).$$

It follows from the maximum principle that the optimum controls will be defined by

$$u_1^o = sign(p_2), \quad u_2^o = -sign(p_3). \tag{5.7}$$

Now we set $\omega = 10^{-3}, s^{-1}$. The following values are predetermined:

$$x_1(0) = 1000, \quad x_2 = 1000, \quad x_3 = -1000, \quad b = 1000.$$

The selected b-value corresponds to micro-thrust when the maximum acceleration produced by the force of thrust from each engine is 1 mm/s^2.

We have to select the controls u_1, u_2 in such a way as to transfer a three-dimensional status vector to an origin in minimum time, namely, to satisfy the conditions

$$x_1(\omega T) = x_2(\omega T) = x_3(\omega T) = 0.$$

Then the optimum controls will guarantee the minimum propellant consumption when adjusting the near-circular ESV orbit.

With this in mind, we shall now solve a two-point boundary value problem; namely, we need to find the unknown initial conditions θ_1, θ_2 and the unknown quantity $\theta_3 = f$ at the moment when the orbit adjustment is terminated.

Stage 1. According to the aforesaid, find an upper-bound estimate for the control time T that is required to select the a priori domain Ω_θ. For the purposes of this intermediate task, define the reasonable control as follows. Put $u_1 = 0$, and let us search the relay (two-position) control u_2 characterized by two unknown sign switching points.

At these times, let the f be the gain the values of ϑ_1 and ϑ_2, respectively.

The quantity $\vartheta_3 = \omega T$ is an f value at the moment when the orbit adjustment is terminated—which is the time by when the status vector is to be put to the origin. Furthermore, the following inequalities must hold:

$$\vartheta_1 < \vartheta_2 < \vartheta_3.$$

With $u_1 = 0$, the solution of a set of differential equations (5.1)–(5.3) is as follows:

$$x(f) = \Phi(f)x(0) + b\int_0^f \Phi(f-\tau)u(\tau)d\tau, \tag{5.8}$$

where

$$u(\tau)^T = \|0 \quad 0 \quad u_2(\tau)\|.$$

$\Phi(f)$ is the fundamental matrix for Eqs. (5.1)–(5.3) and is defined by the equation

$$\Phi(f) = \begin{pmatrix} 4 - 3\cos(f) & \sin(f) & 2(1-\cos(f)) \\ 3\sin(f) & \cos(f) & 2\sin(f) \\ -6(1-\cos(f)) & -2\sin(f) & 4\cos(f) - 3. \end{pmatrix} \tag{5.9}$$

Put

$$u_2(0) = 1 \quad |u_2(f)| = 1.$$

From (5.8), (5.9), we shall now determine a set of three algebraic equations with which the unknowns ϑ_1, ϑ_2, and ϑ_3 must comply:

$$x_1(0)(4 - 3\cos(\vartheta_3)) + x_2(0)\sin(\vartheta_3) + x_3(0)2(1 - \cos(\vartheta_3))$$

$$+4b(\vartheta_1 - \vartheta_2 - 0.5\vartheta_3 + \sin(\vartheta_3 - \vartheta_1)$$

$$-\sin(\vartheta_3 - \vartheta_2)) = 0,$$

$$x_1(0)3\sin(\vartheta_3) + x_2(0)\cos(\vartheta_3) + x_3(0) + 2\sin(\vartheta_3)$$

$$+4b(-0.5 - x_2(0)\cos(\vartheta_3) + \cos(\vartheta_3 - \vartheta_1) + \cos(\vartheta_3 - \vartheta_2) + 1) = 0,$$

$$-x_1(0)6(1 - \cos(\vartheta_3)) - x_2(0)2\sin(\vartheta_3) + x_3(0)(4\cos(\vartheta_3) - 3)$$

$$+b(6(-\vartheta_1 + \vartheta_2 - 0.5\vartheta_3) + 8(0.5\sin(\vartheta_3)$$

$$-\sin(\vartheta_3) - \vartheta_1) + \sin(\vartheta_3 - \vartheta_2)) = 0.$$

The edges of the cube $0 < \vartheta_1, \vartheta_2, \vartheta_3 < 5\pi$ are divided into 500 parts, and the error vectors are determined at the points $\vartheta_1 < \vartheta_2 < \vartheta_3$ via solving a set of algebraic equations where the right parts stand for the vector components. The shortest vector affords the value 3, which is taken for the upper-bound estimate for the correction time. We have found $\vartheta_3 = 11.10867$. At that, the errors of solving a set of algebraic equations were, respectively,

$$\delta_1 = -7.10, \quad \delta_2 = -6.78, \quad \delta_3 = -8.90.$$

Stage 2. The edges of the square $0 < \vartheta_1, \vartheta_2 < 2\pi$ are divided into 200 parts, and the square is covered by 200^2 points. Then we apply the simple Euler procedure with an increment of 0.01 radian, and integrate Eqs. (5.1)–(5.6) with fixed $\theta_3 < \vartheta_3$ and imposed controls (5.7), to find 200^2 approximate minimum-time control pathways. Upon varying the value $\theta_3 < \vartheta_3$, the algorithm selects a pathway from the ensemble of pathways, to minimize the $|x(f)|$ value at the endpoint of integrations.

Corresponding ϑ_1, ϑ_2, and ϑ_3 values are taken for the first guess when solving the time-optimal control problem:

$$\vartheta_1^* = 0.665, \quad \vartheta_2^* = 0.085, \quad \vartheta_3^* = 2.198.$$

The orbit correction errors are therefore

$$\delta_1 = 514.516, \quad \delta_2 = 3.625, \quad \delta_3 = 247.763.$$

Obviously, the simple search technique is a fairly "rough-and-ready" method: It produces significant errors when adjusting the near-circular ESV orbit, despite the huge number of points to cover the parallelepiped.

Stage 3. Now we shall refine the solution of the boundary value problem using a polynomial approximation technique.

Prior to iteration 1, let's define a parallelepiped Ω_θ^* via the conditions

$$\theta_1^* - 0.005, \theta_1 < \theta_1^* + 0.005,$$

$$\theta_2^* - 0.005, \theta_2 < \theta_2^* + 0.005,$$

$$\theta_3^* - 0.5, \theta_3 < \theta_3^* + 0.5.$$

Then perform iteration 1 over the domain Ω_θ^*. We put $d = 3$ and apply a polynomial approximation technique combined with the numerical integration of the Eqs. (5.1)–(5.6) using the simple Euler procedure with an increment of 0.0001 radian. The first approximation vector components will be obtained:

$$\theta_1^{(1)} = 0.667386, \quad \theta_2^{(1)} = 0.079556, \quad \theta_3^{(1)} = 2.67411.$$

The orbit correction errors have been reduced and now are as follows:

$$\delta_1 = 33.46 \quad \delta_2 = 119.56 \quad \delta_3 = 16.76.$$

Let's define a parallelepiped Ω_θ^1 via the conditions

$$\theta_1^{(1)} - 0.005 < \theta_1 < \theta_1^{(1)} + 0.005,$$

$$\theta_2^{(1)} - 0.005 < \theta_2 < \theta_2^{(1)} + 0.005,$$

$$\theta_3^{(1)} - 0.5 < \theta_3 < \theta_3^{(1)} + 0.5.$$

Then, with $d = 4$, we will perform iteration 2 using a polynomial approximation technique; the second approximation vector components will be obtained accordingly:

$$\theta_1^{(2)} = 0.667574 \quad \theta_1^{(2)} = 0.667574 \quad \theta_1^{(2)} = 0.667574.$$

The orbit correction errors have been dramatically reduced and now are as follows:

$$\delta_1 = 0.21, \quad \delta_2 = 0.16, \quad \delta_3 = -0.42.$$

Iteration 3 results in some correction error reduction as well:

$$\delta_1 = -0.16, \quad \delta_2 = -0.23, \quad \delta_3 = 0.13.$$

Thus, the boundary value problem, under conditions of interest, has been solved almost exactly following two to three iterations, and, therefore, a time-optimum near-circular ESV orbit correction problem has been solved almost exactly as well.

One can see from the comparison of the $\theta_3^{(2)}$ value and the 3 value found in stage 1 that the minimum orbit correction time has dropped by factor of 4 due to the use of two low-thrust engines as against the minimum orbit correction time attainable with a single low-thrust engine.

When implementing the time-optimal relay correction, the u_1, u_2 values will be as follows:

$$0 < f < 1.560, \quad u_1 = -1;$$

$$1.560 < f < 2.665, \quad u_1 = +1;$$

$$0 < f < 1.1.140, \quad u_2 = -1;$$

$$1.140 < f < 1.980, \quad u_2 = +1;$$

$$1.980 < f < 2.665, \quad u_2 = -1.$$

The Hamiltonian H is nearly constant at the final minimum-time pathway. Here are those first digits of the f value that do not change within the sign constancy intervals of the controls u_1, u_2:

$$0 < f < 1.140, \quad H = 432.459;$$

$$1.1.140 < f < 1.560, \quad H = 432.5;$$

$$1.560 < f < 1.980, \quad H = 432.6;$$

$$1.980 < f < 2.665, \quad H = 432.676.$$

It must be emphasized that the described calculation process consists of the repeated, typical numerical solutions of the Cauchy problem for a variety of initial conditions. Apart from the last stage, where a few iterations are included, the numerical solutions use a simple Euler procedure with reasonably large increments.

8.6 Correcting the Near-Circular Orbit and Position of the Earth Satellite Vehicle in Minimum Practicable Time Using Micro-Thrust Operation of Two Engines

A more challenging problem is solved in the similar way; namely, the time-optimal controls u_1, u_2 have to adjust the near-circular orbit and eliminate the ESV position error along its pathway. A task of this sort may arise in the proximity path section of two ESVs prior to docking. If the actual ESV x_2-coordinate is x_4, then Eqs. (5.1)–(5.3) must be augmented by the equation

$$dx_4/df = x_3,$$

and the equations for the conjugate variable vector components will be as follows:

$$\begin{aligned} dp_1/df &= -3p_2, \\ dp_2/df &= -p_1 + 2p_3, \\ dp_3/df &= -2p_2 - p_4, \\ dp_4/df &= 0, \end{aligned}$$

with the initial conditions

$$\begin{aligned} p_1^o &= \cos(\theta_1), \\ p_2^o &= \sin(\theta_1)\cos(\theta_2), \\ p_3^o &= \sin(\theta_1)\sin(\theta_2\cos(\theta_3), \\ p_4^o &= \sin(\theta_1)\sin(\theta_2\sin(\theta_3). \end{aligned}$$

Next, we solve the boundary value problem for the initial conditions

$$x_1(0) = 1000, \quad x_2(0) = 1000, \quad x_3(0) = -1000, \quad x_4(0) = 1000.$$

The optimum controls $u_1^o(f), u_2^o(f)$ will be defined by Eq. (5.7).

Let θ_4 be the unknown f at the endpoint of the minimum-time adjustment. Then, upon the solution of the boundary value problem, the parameters $\theta_1, \theta_2, \theta_3$, and θ_4 must allow the goal of the adjustment to be met, that is, satisfying the terminal conditions

$$x_1(\theta_4) = x_2(\theta_4) = x_3(\theta_4) = x_4(\theta_4) = 0.$$

Stage 1. In order to find the ϑ_4 value, an upper-bound estimate for θ_4, define the reasonable control as follows:

$$u_1(f) = 0, \quad |u_2| = 1, \quad u_2(0) = 1.$$

The function $u_2^o(f)$ reverses its sign once the current f value turns into precisely the unknown values

$$\vartheta_1 < \vartheta_2 < \vartheta_3 < \vartheta_4;$$

the control terminates once $f = \vartheta_4$. The fundamental dimension-(4×4) matrix $\Phi(f)$ for the equations can be found from matrix (5.9) with the bottom row

$$\|6(\sin f - f) \quad -2(1-\cos(f)) \quad -3f + 4\sin(f) \quad 1\|$$

and the rightmost column $\|0 \quad 0 \quad 0 \quad 1\|^T$ added to it.

The unknown parameters satisfy a set of four algebraic equations:

$$x_1(0)(4 - 3\cos(\vartheta_4)) + x_2(0)\sin(\vartheta_4) - x_3(0)2(1-\cos(\vartheta_4))$$

$$+4b(\vartheta_1 - \vartheta_2 + \vartheta_3 - 0.5\vartheta_4 - 0.5\sin(\vartheta_4)$$

$$+\sin(\vartheta_4 - \vartheta_1) - \sin(\vartheta_4 - \vartheta_2) + \sin(\vartheta_4 - \vartheta_3)) = 0,$$

$$x_1(0)3\sin(\vartheta_4)) + x_2(0)\cos(\vartheta_4) - x_3(0)2\sin(\vartheta_4)) +$$

$$4b(-0.5 - 0,5\cos(\vartheta_4)$$

$$+\cos(\vartheta_4-\vartheta_1)-\cos(\vartheta_4-\vartheta_2)+\cos(\vartheta_4-\vartheta_3))=0,$$

$$-x_1(0)6(1-\cos(\vartheta_4))-x_2(0)\sin(\vartheta_4)+x_3(0)2(4\cos(\vartheta_4)-3)$$

$$+b(6(-\vartheta_1+\vartheta_2-\vartheta_3+0.5\vartheta_4)+8(0.5\sin(\vartheta_4)$$

$$-\sin(\vartheta_4-\vartheta_1)+\sin(\vartheta_4-\vartheta_2)-\sin(\vartheta_4-\vartheta_3)))=0,$$

$$x_1(0)6(\sin(\vartheta_4)-\vartheta_4)-x_2(0)2(1-\cos(\vartheta_4))+x_3(0)2(3\vartheta+4\sin(\vartheta_4))+$$

$$x_4(0)+b(3(\vartheta_1^2-\vartheta_2^2+\vartheta_3^2-0.5\vartheta_4^2)+6(-\vartheta_1+\vartheta_2-\vartheta_3)\vartheta_4$$

$$+8(-0.5\cos(\vartheta_4)+\cos(\vartheta_4-\vartheta_1)+\cos(\vartheta_4-\vartheta_2)+\cos(\vartheta_4-\vartheta_3)+0.5))=0.$$

Apply simple search to find the approximate root values for a set of algebraic equations:

$$\vartheta_1=0.99\quad\vartheta_2=6.12\quad\vartheta_3=11.72\quad\vartheta_4=14.18.$$

So henceforth we adopt $\theta_4<14$.

Stage 2. Applying the simple search technique over a variety of the minimum-time pathways, we have found an approximate solution for the boundary value problem:

$$\theta_1^*=0.419;\quad\theta_2^*=0.350;\quad\theta_3^*=0.468;\quad\theta_4^*=4.250,$$

which affords significant errors. There are fairly large orbit correction errors at this stage:

$$\delta_1=-40.10,\quad\delta_2=-143.03,\quad\delta_3=254.20,\quad\delta_4=485.43.$$

Now we take the approximate solution so found for the first guess when seeking the exact solution of the problem.

Stage 3. Now we shall refine the solution of the boundary value problem using a polynomial approximation technique. Let's define a parallelepiped Ω_θ^* via the conditions

$$\theta_1^*-0.001<\theta_1<\theta_1^*+0.001,$$

$$\theta_2^*-0.001<\theta_2<\theta_2^*+0.001,$$

$$\theta_3^*-0.001<\theta_3<\theta_3^*+0.001,$$

$$\theta_4^*-0.001<\theta_4<\theta_4^*+0.001.$$

Then we perform iteration 1 over the domain Ω_θ^*. We put $d=3$ and apply a polynomial approximation technique; the first approximation vector components will be obtained:

$$\theta_1^{(1)} = 0.419; \quad \theta_2^{(1)} = 0.345; \quad \theta_3^{(1)} = 0.470; \quad \theta_4^{(1)} = 4.246.$$

The correction errors are still fairly large and now are as follows:

$$\delta_1 = -101.75; \quad \delta_2 = -81.51; \quad \delta_3 = 91.60; \quad \delta_4 = 62.38.$$

Let's define a parallelepiped Ω_1^1 via the conditions

$$\theta_1^1 - 0.0001 < \theta_1 < \theta_1^1 + 0.0001,$$

$$\theta_2^1 - 0.0001 < \theta_2 < \theta_2^1 + 0.0001,$$

$$\theta_3^1 - 0.0001 < \theta_3 < \theta_3^1 + 0.0001,$$

$$\theta_4^1 - 0.0001 < \theta_4 < \theta_4^1 + 0.0001.$$

Then, with $d = 3$, perform iteration 2 over the domain Ω_θ^1 using a polynomial approximation technique; the second approximation vector components will be obtained:

$$\theta_1^{(2)} = 0.419; \quad \theta_2^{(2)} = 0.345; \quad \theta_3^{(2)} = 0.470; \quad \theta_4^{(2)} = 4.265.$$

The correction errors have been reduced dramatically and now are as follows:

$$\delta_1 = -0,40; \quad \delta_2 = -0.81; \quad \delta_3 = 0.41; \quad \delta_4 = -0.42.$$

These small errors can be reduced even more due to iteration 3.

It is seen that the near-circular ESV orbit correction and the ESV position error elimination problem have been solved almost exactly following as few as two iterations, which is due to the polynomial approximation algorithm applied.

One can see from the $\theta_4^{(2)}$ value and the 4 value found in stage 1 that the minimum correction time has dropped by factor of 3 due to the use of two low-thrust engines as compared to the minimum correction time attainable with a single low-thrust engine.

When we implement the time-optimal correction, the relay law control governing the control variations u_1, u_2 will be as follows:

$$0.000 < f < 2.220, \quad u_1 = -l,$$

$$2.220 < f < 2.870, \quad u_1 = +1,$$

$$2.870 < f < 4.260, \quad u_1 = -1,$$

$$0.000 < f < 1.190, \quad u_2 = -1,$$

$$1.190 < f < 2.510, \quad u_2 = +l,$$

$$2.510 < f < 3.950, \quad u_2 = -l,$$

$$3.950 < f < 4.260, \quad u_2 = +1.$$

The Hamiltonian H is nearly constant at the minimum-time pathway.

References

1. Pontryagin LS et al (1969) Mathematical theory of optimal processes. Physmatgiz, Moscow
2. Polak E (1974) Numerical methods of optimization. Universal Approach, World
3. Fedorenko RP (1978) Approximated methods on solution of optimal control problems. Physmatgiz, Moscow

Chapter 9
Inverse Problem of Dynamics: The Algorithm for Identifying the Parameters of an Aircraft

9.1 Introduction

The development of efficient parameter identification methods for the model of a dynamic system based on real-time measurements of some components of its state vector should be taken as one of the most important problems of applied statistics and computational mathematics. Calculating the motion of the system given the initial conditions and its mathematical model is conventionally called the *direct problem of dynamics*. The inverse problem of dynamics would be the problem of identifying the system model parameters based on measurements of certain components of the state vector provided that the general structural scheme of the model is known from physical considerations. Such an inverse problem corresponds to an identification problem for the dynamic system representing an aircraft. In this case, the general structural scheme of the model (motion equations) follows from the fundamental laws of aerodynamics.

In many cases, modern computational methods and wind tunnel experiments can provide sufficient data on nominal parameters of the mathematical model, which are the nominal aerodynamic characteristics of the aircraft. Nevertheless, there are problems [1] that require correcting the nominal parameters based on measurements taken in real flights. These imply

(1) Verifying and interpreting theoretical predictions and results of wind tunnel experiments (flight data can also be used to improve ground prediction methods).
(2) Obtaining more exact and complete mathematical models of the aircraft dynamics to be applied in designing stability enhancement methods and flight control systems.
(3) Designing flight simulators that require a more accurate dynamic aircraft profile in all flight modes (many motions of aircrafts and flight conditions can be neither reconstructed in the wind tunnel nor calculated analytically to a sufficient accuracy or efficiency).

J. A. Boguslavskiy, *Dynamic Systems Models*, DOI: 10.1007/978-3-319-04036-3_9,

(4) Extending the range of flight modes for new aircrafts, which can include a quantitative determination of stability and impact of control when the configuration is changed or when special flight conditions are realized.
(5) Testing whether the aircraft specification is compliant.

Furthermore, dimensionless numbers at the nodes of one- or two-dimensional tables found in wind tunnel experiments serve as nominal values in the aerodynamic parameter identification problem of the aircraft. This causes the vector that corrects these parameters, which are determined by the algorithm processing the digital data flows received from the aircraft sensors, to have a significant dimension of the order of about several tens or hundreds.

An implementation of multiple NASA-recommended algorithms for identification problems, the Systems Identification Programs for Aircraft (SIDPAS) software package written in the MATLAB® M-files language is available on the Internet as an appendix to [1]. Various existing identification methods published in monographs on statistics and computational mathematics are widely reviewed in [1].

For the most general identification method, one should take the known nonlinear least squares method [2] that forms the sum of squared errors, or the differences between the real measurements and their calculated analogs obtained by numerical integration of motion equations of the system for some realization of the vector of unknown parameters.

Successful identification yields the vector of parameters that delivers the global minimum to the above-mentioned sum of squared errors. Still, this criterion is statistically valid only for linear identification problems, in which measurements are linear with respect to the unknown vector of parameters.

Implementing the nonlinear least squares method to correct nominal parameters of the aircraft based on its test flight data involves computational challenges. These arise when the dimension of the correction vector is big and the sum of squared errors as the function of the correction vector has multiple relative minima or when variations of the Newton's method are applied with the sequence of local linearization performed to find stationary points of this function. In [1], the regression method supported by the *lesq.m, smoo.m, derive.m, and xstep.m* files in SIDPAS is recommended for practical applications.

Suppose the motion equations of the system and the sequence of measurements have the form

$$dx/dt = f(x, \vartheta + \theta, u) \tag{1.1}$$

$$y_k = H_k(x(t_k)) + \xi_k, \tag{1.2}$$

where $x(t_k)$ is the $(n \times 1)$-dimensional vector of the system states at the current instant t and at the given instants $t_k, k = 1, \ldots, N$, ϑ is the $(r \times 1)$-vector of nominal (known) parameters of the system, θ is the vector of unknown parameters that serves as the correction vector for the nominal vector ϑ after the results of measurements are stochastically processed, u is the control vector of the system, $f(\ldots)$ is the given vector-function, y_k is the sequence of vector-results of measurements, $H_k(\ldots)$ is the

given vector-function, and ξ_k, $k = 1, \ldots, N$, is the sequence of random vector-errors of measurements with the given random generator for the mathematical simulation.

We can state the identification problem for the vector θ as follows. Find the estimate as the function of the vector Y_N formed of the results of all measurements $y_1, \ldots, y_N$.

The regression method given in [1] solves this problem under the following limitations:

(1) all components of the state vector can be measured: $y_k = x(t_k) + \xi_k$;
(2) at the measurement instants t_k, the algorithm constructs the estimate of the vector of derivatives dx/dt;
(3) the vector-function $f(x, \vartheta + \eta, u)$ linearly depends on the vector η.

Relations (1.1) and (1.2) show that when conditions (1)–(3) are met and N is sufficiently big, the estimation vector satisfies the overdetermined system of linear algebraic equations, with methods to solve it being well known. The given conditions seem to be rather rigid and may be hard to implement. For instance, it is arguable whether one can construct the vector of derivatives dx/dt sufficiently accurately given the real turbulent atmospheric conditions, which imply that the outputs of the angle of attack and sideslip sensors inevitably include random and unpredictable frequency components.

All this justifies the development of new identification algorithms that can be applied to dynamic systems of a rather general class and do not possess drawbacks of NASA algorithms. The proposed multipolynomial approximation algorithm (MPA algorithm) serves as such a new identification algorithm.

9.2 Statement of the Problem and Basic Scheme of the Identification Algorithm

The general scheme for identifying aerodynamic characteristics of the aircraft by the test flight data is as follows [1]. Motion equations of the aircraft (1.1) and system (1.2) of measurements of motion characteristics of the aircraft are given. The vector ϑ is the vector of nominal aerodynamic parameters determined in the wind tunnel experiment. Calculated by the results of real (test) flight, the vector η is used to correct the vector ϑ.

When the aircraft flies, its computer fixes the digital array of initial conditions and time functions, namely, current control surface angles and measurements of some motion parameters of the aircraft [some components of the vector $x(t)$ of the state of the aircraft] received from its sensors. Note that selecting the criterion for optimal or, at least, rational mode to control the test flight is a separate problem and lies beyond our further consideration. The current motion characteristics measured as the time function, such as angles of attack and sideslip, and components of the vector of angular velocity and g-load obtained by the inertial system of the aircraft are registered for

real (not known for sure) aerodynamic parameters of the aircraft (parameters $\vartheta + \eta$) and can be called measured characteristics of the perturbed motion.

Once the flight under the mentioned (given) initial conditions and time functions (control surface angles) is completed, nominal motion equations [equations of form (1.1) for $\theta = 0$] are integrated numerically for the nominal aerodynamic parameters of the aircraft. For the calculated characteristics of the nominal motion of the aircraft, one should take the obtained data—components of the state vector of the aircraft—as the function of discrete time. Differences between measurable characteristics of the perturbed motion and calculated characteristics of the nominal motion serve as carriers of data on the unknown vector η, which shows the difference between real and nominal aerodynamic parameters.

The input of the MPA identification algorithm receives the vector of initial conditions and control surface angles as functions of time and arrays of characteristics of nominal and perturbed motions.

The output of the algorithm is $\hat{\theta}(Y_N)$, which is the correction vector for nominal aerodynamic parameters.

The identification algorithm is efficient if the motion equations, integrated numerically with the corrected aerodynamic parameters, yield such motion characteristics $\vartheta + \hat{\theta}(Y_N)$ (*corrected* characteristics, in what follows) that are close to real (measurable) characteristics.

In this work, we consider the technology of applying the Bayesian MPA algorithm [3, 4] to solve identification problems on the example of the aircraft, for which nominal aerodynamic parameters of the pitching motion are the nominal parameters of a "pseudo" F-16 aircraft.

We replace real flights by mathematical simulation, with characteristics of the perturbed motion obtained by integrating the motion equations of the aircraft numerically. In these equations, nominal aerodynamic parameters at the nodes of the corresponding tables are changed to random values that do not exceed in modulus the given 25–50 % of nominal values at these nodes.

Fundamentally, the MPA algorithm assumes that the vector of unknown parameters η is random on the set of possible flights. We assume that the apriori statistical generator for computer-generated random vectors η and ξ_k is given. This generator makes the algorithm estimating components of the vector η (the identification algorithm) Bayesian. Further, for particular calculations, we assume that random components of the mentioned vectors are distributed uniformly and can be called by the standard Random program in Turbo Pascal.

The MPA algorithm provides the approximation method we implement with the multidimensional power series of the vector $E(\theta|Y_N)$ of the conditional mathematical expectation of the vector η if the vector of measurements Y_N is fixed and apriori statistical data on random vectors θ and ξ_k are given.

The vector $E(\theta|Y_N)$ is known to be the optimal, in the root-mean-square sense, estimate of the random vector θ.

We describe the steps of operation of the MPA algorithm when it identifies the vector θ [5, 6].

We assume that readers have created a computer program that generates a Markov process, corresponding to option 1 (matrix of conditional probabilities is known without error) or option 2 (matrix of conditional probabilities with known bugs), which is given an apriori statistical distribution. Using a statistical model of the distribution of random errors of measurement θ_t, construct the set of possible realizations of random functions Y_t, which consists of sequences of the form $y_0, y_1, \ldots, Y_k, \ldots$. One of these sequences is observable.

Step 1. Suppose d is a given positive integer number and the set of integer numbers $a_1, \ldots, a_N$ consists of all nonnegative solutions of the integer inequality $a_1 + \cdots + a_N \leq d$, whose number we denote by $m(d, N)$. The value $m(d, N)$ is given by the recurrent formula proved by induction.
We obtain the vector $W_N(d)$ of dimension $m(d, N) \times 1$, whose components $w_1, \ldots, w_m(d, N)$ are all possible values $y_1^{a_1} \cdots y_N^{a_n}$ of the form that represent the powers of measurable values.
Then we construct the base vector $V(d, N)$ of dimension $(r+m(d, N)) \times 1$, $V(d, N) = \|\theta W_N(d)\|$.

Step 2. We use a known statistical generator of random vectors θ and ξ_k to solve repeatedly the Cauchy problem for Eq. (1.1) for given initial conditions $x(0)$, a control law $u(t)$, and various realizations of random vectors η and xi_k.
We apply the Monte Carlo method to find the prior first and second statistical moments of the vector $V(d, N)$, that is, the mathematical expectation $\bar{V}(d, N)$, and the covariance matrix $C_V(d, N) = E((V(d, N) - \bar{V}(d, N))(V(d, N) - \bar{V}(d, N))^T)$.
Implementation of step 2 is a learning process for the algorithm, adjusting it to solve the particular problem described by Eqs. (1.1) and (1.2).

Step 3. For given d and N and a fixed vector Y_N, we assign the vector $\hat{\theta}(W_N(d))$ to be the solution to the estimation problem. This vector gives an approximate estimate of the vector $E(\eta|Y_N)$ that is optimal in the root-mean-square sense on the set of vector linear combinations of components of the vector $W_{N_1}(d)$:

$$\hat{\theta}(W_N(d)) = \sum_{a_1+\cdots+a_N \leq d} \lambda(a_1, \ldots, a_N) y_1^{a_1} \cdots y_N^{a_N}. \tag{2.1}$$

The vector $\bar{V}(d, N)$ and the matrix $C_V(d, N)$ are the initial conditions for the process of recurrent calculations that realizes the principle of observation decomposition [5] and consists of $m(d, N)$ steps. Once the final step is performed, we obtain vector coefficients $\lambda(a_1, \ldots, a_N)$ for (2.1). Moreover, we determine the matrix $C(d, N)$, which is the estimation error covariance matrix for the vector $E(\theta_N|Y_N)$ of conditional mathematical expectation estimated by the vector $\hat{\theta}(W_N(d))$.

Calculating the elements of the matrix $C(d, N)$, we have the method of preliminary (prior to the actual flight) analysis of observability of identified parameters for the given control law, structure of measurements, and their expected random errors.

Recurrent calculations do not require matrix inversion and indicate the situations when the next component of the vector $W_N(d)$ is close to a linear combination of its previous components. To implement the recursion, we process the components of the vector $W_N(d)$ one after another. However, the adjustment of the algorithm performed by applying the Monte Carlo method to find the vector $\bar{V}(d, N)$ and the matrix $C_V(d, N)$ takes into account a priori ideas on the stochastic structure of components of the whole set of possible vectors $W_N(d)$ that can appear in any realizations of the random vectors θ and ξ_k allowed by the apriori conditions.

This adjustment is the price we have to pay if we want the MPA algorithm to solve nonlinear identification problems efficiently. This is what makes the MPA algorithm differ fundamentally from, for instance, the standard Kalman filter designed to solve linear identification problems only or from multiple variations of algorithms resulting from attempts to extend the Kalman filter to nonlinear filtration problems.

In [5], a multidimensional analog of the Weierstrass's theorem (the corollary of Stone's theorem [7]) is used to prove that when the integer d increases, then the error estimates of the vector $E(\eta|Y_N)$ and the vector $|\hat{\theta}(W_N(d)) - E(\theta|Y_N)|$ tend to zero uniformly on some region. Formulas of the recurrent algorithm are given and justified in [3, 4].

This scheme for the MPA algorithm operation shows that it can be applied to identify parameters of almost any dynamic system provided that the structures of the motion equations and measurements of forms (1.1) and (1.2) and prior statistical generators of random unknown parameters and errors of measurements are given. The MPA algorithm is devoid of the above-listed limitations and drawbacks, which gives it substantial advantages over NASA identification algorithms. Apart from errors of computations, the algorithm does not add any other errors (such as errors due to linearization of nonlinear functions) into the identified parameters. Therefore, one should expect the apriori spread of identifiable parameters always to be greater than the posterior spread. This is why we can use iterations.

Let's compare the sequential steps of the standard discrete Kalman filter and the MPA algorithm.

1. The Kalman filter identifies the vector η, which can be represented by part of the components of the state vector of the linear dynamic system for the observations that linearly depend on state vectors. The apriori data are the first and second moments of components of random initial state vectors, uncorrelated random vectors of perturbations, and observation errors. We need these data for sequential (recurrent) construction of the estimation vector that is root-mean-square optimal. Usually assigned, apriori data can also be determined by the Monte Carlo method if the complex mechanism of their appearance is given.
2. To find an asymptotic solution to the nonlinear identification problem, the MPA algorithm, unlike the Kalman filter, requires a priori statistical data on both the initial and all hypothesized future state vectors of the dynamic system and observations. These a priori data are represented by the first and second statistical moments for the random vector $V(d, N)$: the vector $\bar{V}(d, N))$ and the

matrix $C_V(d, N)$. These moments are calculated using the Monte Carlo method. However, there are cases when they can be obtained by numerical multidimensional region integration.

1.1. Once conditions from step 1 are met, the Kalman filter constructs the recurrent process, at every step of which the current estimation vector optimal in the root-mean-square sense and the estimation error covariance matrix are calculated.

2.1. Based on step 2, the MPA algorithm implements the recurrent computational process, which does not require matrix inversion. At each step of the process, we construct

   a. the current estimation vector $\hat{\theta}(W_N(d))$ linear with respect to components of the vector $W_N(d)$ and optimal in the root-mean-square sense on the set of linear combinations of components of this vector; moreover, the uniform convergence $\hat{\theta}(W_N(d)) \to E(\theta|Y_N), d \to \infty$, is attained on some region;
   b. the current estimation error covariance matrix (we emphasize that known numerical methods of constructing approximations of the vector of nonlinear estimates cannot calculate current estimation error covariance matrices).

Implementation of items 2 and 2.1 makes the MPA algorithm more efficient than any known linear identification algorithm since it

i. does not involve linearization,
ii. does not apply variants of the Newton method to solve systems of nonlinear algebraic equations,
iii. forms the estimation vector that tends uniformly to the vector of conditional mathematical expectation for the growing integer d,
iv. obtains the estimation error covariance matrix.

It is worth emphasizing that in this work we just develop the fundamental basis of the computational technique for solving the complex problem of aircraft parameter identification.

9.3 Identification of Aerodynamic Coefficients of the Pitching Motion for a Pseudo F-16 Aircraft

We illustrate the efficiency of the offered MPA algorithm on an example of the identification of 48 dimensionless aerodynamic coefficients for an aircraft close to an F-16, which we shall conditionally name “pseudo F-16”. The term “close” is justified because the coefficients are taken from SIDPAS [1] but are perturbed by the addition of some random numbers.

Table 9.1 Nominal values of the functions $C_{Z_0}(\alpha)$, $C_{m_0}(\alpha)$, $C_{Z_q}(\alpha)$, $C_{m_q}(\alpha)$

Number α_i	$C_{Z_0}(\alpha_i)$	$C_{m_0}(\alpha_i)$	$C_{Z_q}(\alpha_i)$	$C_{m_q}(\alpha_i)$
1	0.7700	−0.1740	−8.8000	−7.2100
2	0.2410	−0.1450	−25.8000	−5.4000
3	−0.1000	−0.1210	−28.9000	−5.2300
4	−0.4160	−0.1270	−31.4000	−5.2600
5	−0.7310	−0.1290	−31.2000	−6.1100
6	−1.0530	−0.1020	−30.7000	−6.6400
7	−1.3660	−0.0970	−27.7000	−5.6900
8	−1.6460	−0.1130	−28.2000	−6.0000
9	−1.9170	−0.0870	−29.0000	−6.2000
10	−2.1200	−0.0840	−29.8000	−6.4000
11	−2.2480	−0.0690	−38.3000	−6.6000
12	−2.2290	−0.0060	−35.3000	−6.0000

Table 9.2 Nominal values of increments $\Delta(C_{Z_0}(\alpha_i))$, $\Delta(C_{m_0}(\alpha_i))$, $\Delta(C_{Z_q}(\alpha_i))$, $\Delta(C_{m_q}(\alpha_i))$

Number α_i	$\Delta(C_{Z_0}(\alpha_i))$	$\Delta(C_{m_0}(\alpha_i))$	$\Delta(C_{Z_q}(\alpha_i))$	$\Delta(C_{m_q}(\alpha_i))$
1	0.7700	−0.1740	−8.8000	−7.2100
2	−0.5290	0.0290	−17.0000	1.8100
3	−0.3410	0.0240	−3.1000	0.1700
4	−0.3160	−0.0060	−2.5000	−0.0300
5	−0.3150	−0.0020	0.2000	−0.8500
6	−0.3220	0.0270	0.5000	−0.5300
7	−0.3130	0.0050	3.0000	0.9500
8	−0.2800	−0.0160	−0.5000	−0.3100
9	−0.2710	0.0260	−0.8000	−0.2000
10	−0.2030	0.0030	−0.8000	−0.2000
11	−0.1280	0.0150	−8.5000	−0.2000
12	0.0190	0.0630	3.0000	0.6000

Tables 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 9.10, 9.11 here show that identification errors are small; modules of their relative values do not surpass several hundredths. The considered problem corresponds to minimization of the object function of 48 variables, which comprise the sum of squared differences of the actual and computational angles of attack, g-load, and pitch angles, observable with a frequency of 10 Hz during 25 s of flight of the aircraft maneuvering in a vertical plane.

9.3.1 Pitching Motion Equations

We use the XYZ rectangular coordinate system adopted by NASA. Then, for the unperturbed atmosphere and conditions $V = \text{const}$, the pitching motion equations have the form [1]

Table 9.3 The characteristics $\alpha(t)$, $N_Z(t)$, $\theta^*(t)$ of the nominal motions for the chosen control law $\delta_s(t)$

Number obs. k	$\delta_s(k)$	$\alpha(k)$	$N_Z(k)$	$\theta^*(k)$
1	−0.0200	3.6820	0.1021	0.0132
11	−0.2200	8.2964	−0.1685	−0.3945
21	−0.4200	11.0977	−0.5334	−1.2461
31	−0.6200	13.1919	−0.4477	−2.2247
41	−0.8200	17.0629	−0.7728	−0.2382
51	−1.0200	19.7287	−0.6512	0.6855
61	−1.2200	19.9789	−1.1308	1.1146
71	−1.4200	20.0598	−1.1344	1.4359
81	−1.6200	20.6696	−1.1186	2.8558
91	−1.8200	24.6576	−0.9641	7.2201
101	−2.0200	32.4354	−1.6031	17.5706
111	−2.1800	35.9159	−1.7715	25.8128
121	−1.9800	34.2309	−1.4610	28.3457
131	−1.7800	31.7080	−1.5248	29.5992
141	−1.5800	30.0324	−1.1643	30.9954
151	−1.3800	29.9805	−1.1621	34.2532
161	−1.1800	29.9772	−1.1614	37.1835
171	−0.9800	29.8635	−1.1625	39.6161
181	−0.7800	27.4286	−1.2336	39.7559
191	−0.5800	18.7935	−0.5743	32.3636
201	−0.3800	12.4703	−0.4193	23.5913
211	−0.1800	9.8365	−0.0427	17.6449
221	0.0200	9.9999	−0.0398	12.7981
231	0.2200	9.6174	−0.0423	6.7279
241	0.4200	5.0095	−0.2203	−3.3684
249	0.5800	−0.9453	0.5373	−15.2658

Table 9.4 Relative errors of the identifications of $C_{Z_0}(\alpha_i)$ by $\rho = 0.25$

Number α_i	Nom.koef. $C_{Z_0}(\alpha_i)$	Perturb.koef. $C_{Z_0}(\alpha_i)$	$\delta(C_{Z_0}(\alpha_i))$
1	0.6512	0.6326	0.02854
2	0.0205	0.0260	−0.26410
3	−0.3778	−0.3646	0.03491
4	−0.7395	−0.7213	0.02456
5	−1.0610	−1.0657	−0.00443
6	−1.4038	−1.4016	0.00159
7	−1.7679	−1.7424	0.01444
8	−2.0582	−2.0453	0.00627
9	−2.2774	−2.3388	−0.02693
10	−2.4568	−2.5459	−0.03625
11	−2.5639	−2.6698	−0.04130
12	−2.5404	−2.6505	−0.04334

Table 9.5 Relative errors of the identifications of $C_{m_0}(\alpha_i)$ by $\rho = 0.25$

Number α_i	Nom.koef. $C_{m_0}(\alpha_i)$	Perturb.koef. $C_{m_0}(\alpha_i)$	$\delta(C_{m_0}(\alpha_i))$
1	−0.2130	−0.2054	0.03582
2	−0.1816	−0.1783	0.01851
3	−0.1567	−0.1550	0.01061
4	−0.1618	−0.1611	0.00439
5	−0.1634	−0.1631	0.00209
6	−0.1427	−0.1388	0.02754
7	−0.1372	−0.1338	0.02439
8	−0.1495	−0.1502	−0.00467
9	−0.1175	−0.1220	−0.03771
10	−0.1139	−0.1190	−0.04484
11	−0.0957	−0.1043	−0.08937
12	−0.0399	−0.0394	0.01236

Table 9.6 Relative errors of the identifications of $C_{Z_q}(\alpha_i)$ by $\rho = 0.25$

Number α_i	Nom.koef. $C_{Z_q}(\alpha_i)$	Perturb.koef. $C_{Z_q}(\alpha_i)$	$\delta(C_{Z_q}(\alpha_i))$
1	−9.9636	−8.8984	0.10691
2	−25.2235	−26.1655	−0.03735
3	−28.4644	−29.2857	−0.02885
4	−31.4821	−31.8270	−0.01096
5	−31.3125	−31.6274	−0.01006
6	−30.8417	−31.1249	−0.00918
7	−27.5461	−28.0921	−0.01982
8	−28.1388	−28.6036	−0.01652
9	−28.9682	−29.4069	−0.01515
10	−29.7908	−30.2114	−0.01412
11	−38.6789	−38.7933	−0.00296
12	−35.7355	−35.8053	−0.00195

$$
\begin{aligned}
d\alpha/dt &= \omega_Y + (g/V)(N_Z + cos(\theta^* - \alpha)),\\
d\omega_Y/dt &= M_Y/J_Y,\\
d\theta^*/dt &= \omega_Y,\\
N_Z &= C_Z(\alpha, \delta_s)qS/G,\\
M_Y &= C_m(\alpha, \delta_s)qSb,
\end{aligned}
$$

where $V = 300$ ft/s, $H = 20{,}000$ ft, α is the angle of attack, N_Z is the g-load, which is the vector of aerodynamic forces projected onto the Z-axis and divided by the weight of the aircraft, M_Y is the vector of the moment of aerodynamic forces projected onto the Y-axis, ω is the vector of the angular velocity of the aircraft projected onto the Y-axis, θ is the angle between the X-axis and the horizontal plane, q is the value of

Table 9.7 Relative errors of the identifications of $C_{m_q}(\alpha_i)$ by $\rho = 0.25$

Number α_i	Nom.koef. $C_{m_q}(\alpha_i)$	Perturb.koef. $C_{m_q}(\alpha_i)$	$\delta(C_{m_q}(\alpha_i))$
1	−5.5807	−6.1771	−0.10686
2	−4.1294	−4.3066	−0.04291
3	−3.9913	−4.1368	−0.03645
4	−4.0250	−4.1662	−0.03510
5	−4.9363	−5.0012	−0.01315
6	−5.5024	−5.5314	−0.00527
7	−4.5272	−4.5870	−0.01320
8	−4.8711	−4.8936	−0.00462
9	−5.0970	−5.0915	0.00108
10	−5.3245	−5.2912	0.00626
11	−5.5637	−5.4908	0.01310
12	−4.8726	−4.8937	−0.00434

Table 9.8 Relative errors of the identifications of $C_{Z_0}(\alpha_i)$ by $\rho = 0.50$

Number α_i	Nom.koef. $C_{Z_0}(\alpha_i)$	Perturb.koef.$C_{Z_0}(\alpha_i)$	$\delta(C_{Z_0}(\alpha_i))$
1	0.5324	0.4255	0.20083
2	−0.1999	−0.2092	−0.04637
3	−0.6556	−0.5969	0.08959
4	−1.0629	−0.9303	0.12481
5	−1.3911	−1.2772	0.08188
6	−1.7546	−1.6697	0.04839
7	−2.1699	−2.0331	0.06304
8	−2.4704	−2.3417	0.05209
9	−2.6379	−2.6342	0.00138
10	−2.7936	−2.8450	−0.01839
11	−2.8799	−2.9723	−0.03208
12	−2.8518	−2.9530	−0.03548

the dynamic pressure, G is the weight, J_Y is the moment of inertia with respect to the Y-axis, S is the area of the surface generating aerodynamic forces, b is the mean aerodynamic of the wing, $C_Z(\alpha, \delta)$ and $C_m(\alpha, \delta)$ are dimensionless coefficients of the aerodynamic force and moment, respectively, and δ_s is the angle of the stabilizer deflectors measured in degrees.

The functions $C_Z(\alpha, \delta_s)$ and $C_m(\alpha, \delta_s)$ are given by the relations [1]

$$C_Z(\alpha, \delta_s) = C_{Z_0}(\alpha) - 0.19(\delta_s/25) + C_{Z_q}(\alpha)(b/(2V))\omega_Y,$$
$$C_m(\alpha, \delta_s) = C_{m_0}(\alpha)\delta_s + C_{m_q}(\alpha)(b/(2V))\omega_Y + 0.1C_Z.$$

Table 9.9 Relative errors of the identifications of $C_{m_0}(\alpha_i)$ by $\rho = 0.50$

Number α_i	Nom.koef. $C_{m_0}(\alpha_i)$	Perturb.koef. $C_{m_0}(\alpha_i)$	$\delta(C_{m_0}(\alpha_i))$
1	−0.2520	−0.2441	0.03123
2	−0.2183	−0.2166	0.00781
3	−0.1924	−0.1934	−0.00523
4	−0.1966	−0.1994	−0.01457
5	−0.1979	−0.2014	−0.01792
6	−0.1834	−0.1747	0.04747
7	−0.1773	−0.1697	0.04301
8	−0.1860	−0.1858	0.00145
9	−0.1481	−0.1599	−0.08004
10	−0.1438	−0.1569	−0.09149
11	−0.1225	−0.1420	−0.15942
12	−0.0738	−0.0811	−0.09934

Table 9.10 Relative errors of the identifications of $C_{Z_q}(\alpha_i)$ by $\rho = 0.50$

Number α_i	Nom.koef. $C_{Z_q}(\alpha_i)$	Perturb.koef. $C_{Z_q}(\alpha_i)$	$\delta(C_{Z_q}(\alpha_i))$
1	−11.1272	−8.6840	0.21957
2	−24.6470	−25.6672	−0.04139
3	−28.0288	−28.8049	−0.02769
4	−31.5642	−31.3356	0.00724
5	−31.4249	−31.1306	0.00937
6	−30.9833	−30.6296	0.01142
7	−27.3921	−27.6113	−0.00800
8	−28.0776	−28.1104	−0.00117
9	−28.9364	−28.9144	0.00076
10	−29.7817	−29.7303	0.00172
11	−39.0577	−38.2130	0.02163
12	−36.1709	−35.1346	0.02865

9.3.2 *Parametric Model of Aerodynamic Forces and Moments*

The nominal values of four functions of the angle of attack $C_{Z_0}(\alpha)$, $C_{m_0}(\alpha)$, $C_{Z_q}(\alpha)$, $C_{m_q}(\alpha)$ are given with the argument step of $(55 - 1)/12$ degrees at 12 nodes (Table 9.1) in the range $-10° \le \alpha \le 45°$.

To determine the values of functions between the nodes, we use linear interpolation. Having analyzed Table 9.1, we can see that functions $C_{Z_0}(\alpha_i)$, $C_{m_0}(\alpha_i)$, $C_{Z_q}(\alpha_i)$, $C_{m_q}(\alpha_i)$ are essentially nonlinear. Table 9.2 confirms this visual impression. It presents increments for functions of each step of Table 9.1. As is apparent, the increments noticeably vary.

We study the identification problem for the perturbed analogs of the functions $C_{Z_0}(\alpha)$, $C_{m_0}(\alpha)$, $C_{Z_q}(\alpha)$, $C_{m_q}(\alpha)$. The number of nominal coefficients that

Table 9.11 Relative errors of the identifications of $C_{m_q}(\alpha_i)$ by $\rho = 0.50$

Number α_i	Nom.koef. $C_{m_q}(\alpha_i)$	Perturb.koef. $C_{m_q}(\alpha_i)$	$\delta(C_{m_q}(\alpha_i))$
1	−3.9514	−6.9359	−0.75528
2	−2.8588	−5.1596	−0.80480
3	−2.7526	−4.9893	−0.81258
4	−2.7899	−5.0189	−0.79894
5	−3.7625	−5.8672	−0.55939
6	−4.3649	−6.3936	−0.46477
7	−3.3644	−5.4530	−0.62079
8	−3.7422	−5.7627	−0.53993
9	−3.9940	−5.9631	−0.49299
10	−4.2490	−6.1601	−0.44976
11	−4.5274	−6.3594	−0.40464
12	−3.7451	−5.7631	−0.53882

determine these functions is $12 + 12 + 12 + 12 = 48$. Let us single out the problem that is the most complex for the MPA algorithm, when the actual coefficients differ from the nominal coefficients by the unknown bounded by the prior limit value η_i at each point of the table. Then, for accumulated results of measurements of parameters of the perturbed motion, the MPA algorithm is to estimate 48 components of the vector of random estimates, the vector of differences between the actual and nominal coefficients.

Suppose ϑ_i and B_i are the ith components of the nominal and actual (perturbed) vectors of aerodynamic coefficients ϑ, $i = 1, \ldots, 48$; namely, the number of actual coefficients to be identified is 48 in this case. We assume that the parametric model

$$B_i = \vartheta_i + \theta_i$$

holds. The vector θ serves as the vector of perturbations of nominal data errors of aerodynamic parameters, and identification yields the estimates of its components. We give the structure of these components by the formula $\theta_i = \vartheta_i \rho_i \varepsilon_i, 0 < \rho_i < 1, -1 < \varepsilon_i < 1$. The positive number ρ_i gives the maximum value that, by identification conditions, can be attained by the ratio of the absolute values of the random value of perturbations θ_i and nominal coefficients ϑ_i.

9.3.3 Transient Processes of Characteristics of Nominal Motions

We wish to identify—estimate—during one test flight the 48 unknown aerodynamic coefficients for the set of angles of attack $\alpha_i, i = 1, \ldots, 12$. For a testing maneuver, the characteristics $\alpha(t), N_Z(t), \theta^*(t)$ of transient processes are as carrier of

information of the identified coefficients. Therefore, during flight, the aircraft should "visit" vicinities of angles of attack $-10^\circ \leq \alpha \leq 45^\circ$.

9.3.4 *Estimating Identification Accuracy of 48 Errors of Aerodynamic Parameters of the Aircraft*

The primary task of the MPA algorithm consists of identification (via estimation) of 48 increments of 4 functions. If entry conditions and increments are determined, values of the unknown coefficients follow from obvious recurrent formulas.

To estimate the accuracy, we assume that the current values of α, N_Y, θ^* are measured every 0.1 s over a 25 s time period. We assume that random errors of measurement represent the discrete white noise bounded by the true measurable value multiplied by the given value ϵ. A number of the primary observations equals $3 * 250 = 750$.

We compress primary observations for smoothing the high-frequency errors and reducing a dimension of the matrix covariance. The file of the primary observations is divided into 12 groups and as an input of the algorithm of the identification the dimension-(12×1) vector serves. Components of this vector are the sums of elements of each of 12 groups.

To characterize the accuracy of identification of the random parameter θ_i, the degree of perturbation of the aerodynamic coefficients ϑ, we determine the relative errors of estimation $(\theta_i - \hat{\theta}_i)/\theta_i$ for every component of the identifiable functions. The relative errors designate $\delta(C_{Z_0}(\alpha_i))$, $\delta(C_{m_0}(\alpha_i))$, $\delta(C_{Z_q}(\alpha_i))$, $\delta(C_{m_q}(\alpha_i))$, $i = 1, \ldots, 12$.

As is apparent, relative errors of identification are small and do not surpass several hundredths at $\rho = 0.25$.

Practice calculations for the implementation methodology detected a significant effect on the accuracy of the estimation of random errors of the aircraft equipment, whose statistical characteristics are unknown. Therefore, a regularization procedure that significantly reduced the impact of the mentioned errors was needed. This procedure added the data from the aircraft equipment random process similar to white noise, whose intensity is selected experimentally. Note that this procedure was referred to earlier in the theory of artificial neural networks [8].

9.4 Conclusions

The presented data show that the multipolynomial approximation algorithm can provide a computational basis for developing an efficient parameter identification technique for the nonlinear dynamic system, including identification of the aerodynamic parameters of an aircraft. We emphasize that tables characterizing a sufficiently high

accuracy of aerodynamic parameter identification are obtained when there are no iterations and $d = 1$, which corresponds to the case when the estimation vector $(\vartheta \hat{+} \theta)(W_N(d))$ is represented by the vector linear combination of measured data that is optimal on the family of linear operators over the vector of measurements. This is due to good (in terms of the identification problem) properties of the parametric system of equations of the pitching motion of the "pseudo F-16" aircraft. It can become much more complicated when it comes to the identification problem of the parametric system of equations of complete (spatial) motion of the aircraft. In such a case, we may need to use polynomials of the power $d > 1$ and increase requirements on the computer performance and RAM. This was the case for identification attempts made for some parameters of F-16 complete motion equations. We emphasize that the inputs of the MPA algorithm we considered were not real (were not the results of the operation of real sensors of the aircraft during its test flight); they were determined by mathematical simulation, that is, via the numerical integrations of motion equations for perturbed parameters of aerodynamic forces and moments.

References

1. Klein V, Morelli AG (2006) Aircraft system identification: theory and methods. American Institute of Aeronautics and Astronautics, Reston
2. Leung L (1991) System identification; user theory. Physmatlit, Moscow
3. Boguslavskiy JA (1996) Bayesian estimations of nonlinear regression and related questions. In: Theory and control systems, vol 4. Bulletin of the Russian Academy of Sciences, Moscow.
4. Boguslavskiy JA (2006) A polynomial approximation for nonlinear problems of estimation and control. Physmatlit, Moscow [in Russian]
5. Boguslavskiy JA, Egorova AV, Obrosov KV (1998) Aircraft instrument complex for autonomous information support of landing. In: Theory and control systems, vol 2. Bulletin of the Russian Academy of Sciences, Mascow
6. Boguslavskiy JA (2000) Analysis on estimation accuracy of Markov model transition probability. In: Theory and control systems, vol 1. Bulletin of the Russian Academy of Sciences, Mascow
7. Boguslavskiy JA (2003) Direct computational method for the task of optimal linear operating speed. In: Theory and control systems, vol 3. Bulletin of the Russian Academy of Sciences, Mascow
8. Bishiop CM (1995) Training with noise is equivalent to Tikhanov regularization. Neural Comput 7:108–116