Slovak Technical University Bratislava
Faculty of Mechanical Engineering

Tomáš Hrúz, Ing.

Inversion of Ill-posed Problems in Control of Distributed Parameter Systems

Ph.D. Thesis

B r a t i s l a v a

1997

To Katka, Martina, Miriam and my parents.

I am very grateful to Professor Rainer Schrader and Professor Achim Bachem for kindly providing a possibility to make this work in a stimulating environment of Institute of Parallel Computing at University of Cologne.

In particular I would like to thank Professor David W. Clarke for introducing me to the field of predictive control. Without him and his support the Copernicus project would certainly be not possible.

My thanks are also going to my colleagues Associate Professor Boris Rohál-Ilkiv and Dr. Országhová with whom some of the preliminary work has been done.

The stay at the University of Cologne has been funded by European Union COPERNICUS Project PREDCON 1174

Contents

List of Principal Symbols

Abbreviations

PDE	partial differential equation
a	in equations: heat conductivity coefficient
b	in equations: contains heat transfer coefficient h i.e. $b = h/(c\rho)$
MBPC	Model Based Predictive Control
GPC	Generalized Predictive Control
Res	Residuum

Operators and Standard Symbols

$\frac{\partial y}{\partial x}$	partial derivative of y with respect to x
∂D	boundary of domain D
Δ	Laplace operator, $\sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$
$\delta(x)$	delta function
(x, y)	scalar product in functional space
A^*	adjoint operator to A
$\|.\|_i$	norm in operator space
$Y(x,t)$	distributed state of the system
$u(t)$	boundary control signal
u^w	boundary signal computed as a reference signal
w	reference signal
$M^\alpha(u)$	smoothing functional
ψ, φ	eigenfunctions of Sturm-Liuville problem

λ_k	eigenvalues of Sturm-Liuville problem
Δz	$\Delta z = z(t) - z(t-1)$ difference operator
$\mathbf{E}^{(l)}$	transition matrix of Crank-Nicolson finite difference method
$\mathcal{P}_{g,d}^{k,l}$	local k-ahead predictor
$\rho_{(h,\delta)}(\alpha)$	generalized residuum

Chapter 1

Introduction

In the recent time it is more and more obvious that the energy circulation on earth caused by humans must be radically restructured to avoid catastrophic impact on ecology caused by current industrial development.

There is an idea of sustainable growth [19] which teaches how industry and culture ought to be restructured to obtain certain reasonable industrial and technological growth yet maintaining the irreversible destruction of the utmost complex ecological system on earth as low as possible.

From the industrial and technological point of view, this work is targeting certain concrete situations in heavy industry (mainly the steel industry) to optimize the energy consumption in such technology processes.

On the other hand, research has also its internal logic which can lead, to certain extent, outside from the original technology idea. However, doing an engineering research means just to have the technology background always active in mind. In our case, we fulfill this condition putting emphasis on the following aspects.

- *Efficient computability.* We put considerable stress on computational and algorithmic aspects in our research, aiming to minimize the inevitable investments to the computer systems capable to realize our control approach.

- *Energy consumption optimization.* The criteria (objective functions) which we are using during the optimization have energetic interpretation. Even

if the optimization algorithms serve also other purposes, they minimize the energy consumption of the system as well.

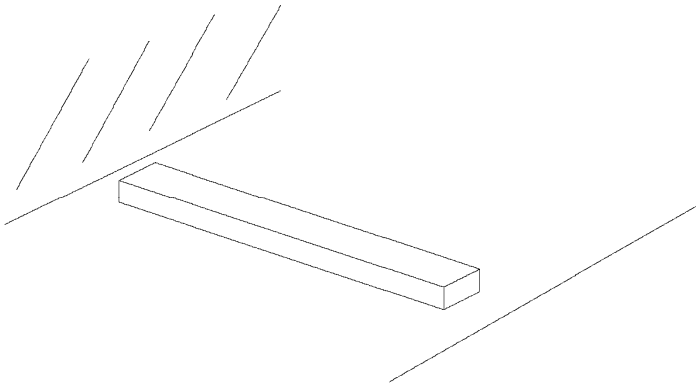
- *Dimension reduction.* Even if the target case is 3-dimensional, we suppose various approximative steps to decompose and to approximate the real process by 1-dimensional systems. The reason is that the proposed control algorithms must be efficiently computable in industrial conditions. Bearing in mind the advances of computer technology, it is well possible to solve 1-dimensional partial differential equation in recent industrial control system. But higher dimensionality is currently out of scope of standard control systems.

The basic technological process behind our research is schematically drawn in Figure 1.1. It is a model of reheating furnace, where ingots with simple geometrical shape are reheated for further processing steps (usually rolling). The idea of this process is abstracted from various concrete industrial plants as are described for example in [43, 23]. Another kind of reheating furnace where the slabs (ingots) revolve on the rotating part of the furnace is in figure 1.2.

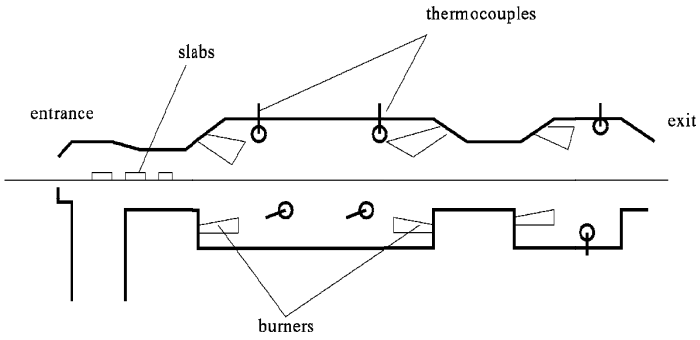
When we try to think accurately about the processes that occur inside such reheating furnace, soon we run into tremendous problems - theoretical as well as practical. For example, the burning gas inside the furnace is in a turbulent state, so accurate modeling and control of such subsystem is not possible with current computer technology.

Fortunately, there are certain simplifications verified in practical situations that can make modeling and control feasible. These simplifications depend strongly on the concrete technological process. For the case of furnace, we can collect them in the following points.

- Usually, the furnace is split to certain number of zones which are heated by few burners. Even if the temperature inside the gas near the burners can have large fluctuations, the inner side of the walls and other parts of the zone have rather constant temperature. The surface of ingots is heated by radiation from the walls and other parts of the zone. This is the reason



a)



b)

Figure 1.1: In the figure a) there is a sketch of the basic situation in which we are interested. A slab (ingot) of a rectangular parallelepiped shape moves in a hot chamber of a reheating furnace. In the figure b) is a longitudinal cross section of a concrete type of reheating furnace operating in continuous mode. The slabs are continuously moving through the furnace. The controlled parameters are the gas input flows to the burners and the speed of the movement. The temperature inside the furnace is measured with a system of thermocouples.

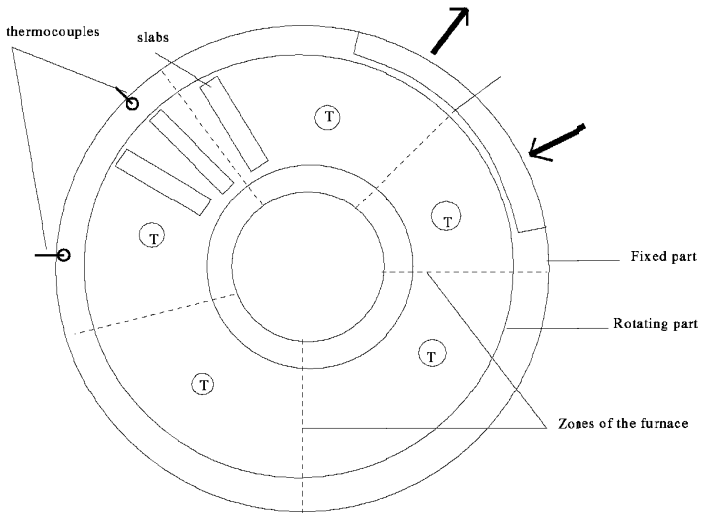


Figure 1.2: A ground plot of the rotating reheating furnace. There are two sets of thermocouples. The first set denoted with "T" is on the ceiling and the second set is on the wall at the height of rotating slabs.

we can suppose that there is approximately spatially constant boundary condition on the whole surface of ingot. Moreover, the design of reheating furnaces usually supports this assumption to make the control feasible.

- Based on the previous point, one zone of reheating furnace acts as an integrated unit, providing spatially constant, possibly time varying source of temperature on the boundary of the ingot.

- The ingots have simple geometric shape determined by further technological steps like rolling etc. We can suppose a shape of rectangular parallelepiped.

- The control system has to set finite number (often less than 10) inputs representing amount of gas flowing to the burners. In general, the objective of control is a constant temperature or a prescribed time development of the temperature inside the ingot achieved under various conditions like minimal energy consumption and/or prescribed time etc.

- There is a small number (between 10 and 20) of thermocouples inside the furnace which allows us to measure the temperature of atmosphere in the furnace. There can be also a pyrometric measurement of the temperature on the surface of ingot. However, we can not suppose to measure the temperature inside the ingot. Therefore, the heat conduction from the surface of ingot to its interior must be modeled in a computer system to achieve good performance.

Even if we apply all these reasonable simplifications, still the control of large technological process as operation of reheating furnace is a very complex task. There are many subsystems and subtasks on which the research can focus to make them more efficient or reliable.

Our research is focusing to the optimal boundary conditions which would give a prescribed temperature profile in space and time. Then, it is a role of other subsystems to track accurately the computed boundary conditions.

The decomposition, which separates this sub-problem, permits us also to experimentally verify the proposed algorithms on laboratory specimens. The labora-

tory system used for the experiments which are targeting this research is described in chapter 3 (see also [42]).

The approach we present here is based on the idea to use a boundary control for modeling of the above mentioned technological plants. But the conditions (state) in the interior of a distributed parameter system are related to the boundary conditions through inverse model. The solution of this model is often so called "ill-posed" because it incorporates an inversion which lacks good properties of the original model. Among the most critical problems are the following: there is no solution to the inverse problem, the solution is not stable (inverse operator is not continuous). Generally, the methods developed for this class of problems consist in using certain " a priori" information to make the problem solvable or to "regularize" the problem.

The presented work contributes in two aspects. First of all, there is a methodological contribution. The new methodology consists in the following point of view to the predictive control of distributed parameter systems.

In general, the synthesis of model based predictive control (MBPC) algorithm is derived from a minimization of certain least squares criteria which express the distance between the controlled variables and the reference signal. We can write this in an operator form as:

$$|| Au - w ||^2_{||X||,D_1} + \lambda || u ||^2_{||Y||,D_2} \tag{1.1}$$

where Au describes a future behaviour of the system as a function of future control, $|| X ||, || Y ||$ denote the corresponding norms in the functional spaces X, Y chosen to express the distance, and D_1, D_2 denote the discretizations.

The above general formulation contains most of the so called *tuning knobs* of model based predictive control schemes because various weighting factors in fact define a special norms. Tuning knobs related to the horizons can be expressed by the particular functional spaces choice. The sampling frequency is the discretization factor and so on. In model based predictive control the parameter λ is interpreted as a weighting factor which introduces the smoothness of the control u to the criterion of control quality (1.1). However, deeper interpretation

of this parameter is not provided in the MBPC framework. On the other hand, in practise it is often inevitable to use this factor to avoid an ill-conditioned character of the equations even in a finite dimensional systems described by ordinary differential equations. As was noted for example in [14] it is difficult to choose this factor in a useful way.

The tuning knobs are then used to obtain a reasonable performance and numerical stability of the control algorithm. When we consider a concrete model based predictive control scheme, we find more tuning knobs of various complexity. This creates rather complex decision space.

We would like to stress that the possibility to successfully find the appropriate value of the tuning parameter λ is strongly dependent on the "spectral behavior" of operator A in the criterion (1.1). We can identify two big classes: one with the spectrum polynomially falling to zero and another, where the spectrum falls exponentially or faster. The latter being the case of infinite dimensional evolution operators (systems described by partial differential equations). In this case it is almost impossible to cope with the severely ill-conditioned numerical behaviour without deeper theory.

Fortunately, such theory is at disposal if we interpret differently the criterion (1.1). We can see the minimization of (1.1) as a tool for solving an ill-posed operator equation

$$Au = w \tag{1.2}$$

in an appropriate functional space. In the case of infinite dimensional evolution operators which are so called "infinitively smoothing" (as is for example the heat conduction equation) there is a well developed theory of ill-posed problems solving which teaches how to invert the equation (1.2) even in the case when the spectrum falls to zero exponentially. This theory includes also the case when the equation (1.2) has no solution and we are looking for the element Au nearest in some sense to w.

So the change of the focus is in the fact that we can consider the control algorithm synthesis as a solving of ill-posed problem (1.2). Then the *regularization*

(ill-posed problem solving) methods give us an algorithm for solution of (1.2) based on the minimization of the following functional:

$$M^{\alpha}(u) = \| Au - w \|^2_{||X||,D_1} + \alpha \| u \|^2_{||Y||,D_2} \tag{1.3}$$

This looks to be the same as (1.1) but now the parameter α plays a crucial role and we have a deep theory how to choose and interpret this so called regularization parameter. Extending this point of view further, it can be seen that discretization step (sampling frequency) is also in fact a regularization parameter as well as various weights. So adopting this point of view, we obtain the possibility to make a synthesis of control algorithm for infinite dimensional infinitively smoothing evolution operators. Moreover, we have a deeper interpretation of tuning knobs as being in fact the regularization parameters also for the classical predictive control case. Particularly the possibility to interpret the appropriate value of λ seems to be interesting.

The above methodology has been a driving force behind the concrete development of the presented work which is centered around the following main contributions:

- Close-loop predictive control algorithm for thermal system described in Section 3.4.
- A robust method for efficient computation of Green's functions in Section 3.2.
- Analysis of prediction/simulation accuracy near the heated boundary of thermal system in Section 3.2.1.2.
- Off-line identification of heat-transfer and thermal diffusivity coefficients in Section 3.1.
- Open-loop step-wise control in Section 3.3.

In the next chapter we collect some background material which establishes the basic frameworks within which we express our work.

In the first part we shortly place our undertakings in the framework of distributed parameter systems control. Another part gives an introduction to ill-posed inverse problems solving. There is also a short introduction to the model based predictive control.

The central part of the work is in Chapter 3 where we proceed from identification and simulation to an open loop step-wise control and finally to a regularization based predictive control.

At the end of the work there are two appendices. The first one derives a solution of the one dimensional heat conduction equation from the first principles as far as much of the text is based on knowledge of this otherwise classical derivation.

To provide an example of software complexity related to the basic algorithms developed here we give a listing of some procedures in the second appendix. The software system which is in the background of this work recently counts more than 13 000 lines of a code in C language. However, the code is written to support experiments and research work and is not meant to be highly efficient "production" implementation. In the near future we plan to rewrite the basic parts of this system with respect to efficiency and portability in Java language.

Except some auxiliary parts of the work, we use the notation $Y(x,t)$ for the temperature distribution to stress that the infinite dimensional distribution along the spatial domain is the output - controlled parameter. In the same way $u(t)$ denotes the boundary value signal for corresponding partial differential equation which represents the control.

A more complicated situation is in Section 3.3 where the boundary condition is pre-computed and serves as a reference signal to one dimensional spline based predictive controller. We use notation $u^w(t)$ to stress this fact.

Chapter 2

Preliminaries

2.1 Boundary control as an ill-posed problem

The process of reheating of an ingot in a reheating furnace, as we have mentioned it in the introduction can be described mathematically with the following linear partial differential equation (PDE):

$$\begin{aligned} Y_t &= a^2 \Delta Y \\ Y(x,y,z,0) &= Y_0(x,y,z), (x,y,z) \in D \\ Y(x,y,z,t) &= u(t), (x,y,z) \in \partial D \end{aligned} \tag{2.1}$$

where the initial and boundary conditions are Y_0, resp. $u(t)$.

Because of an industrial infeasibility of higher dimensional PDEs we devote most of our efforts to the modeling, inversion and control of the following 1-dimensional PDE (2.2):

$$\begin{aligned} \tfrac{\partial}{\partial t} Y(x,t) - a^2 \tfrac{\partial^2}{\partial x^2} Y(x,t) + bY(x,t) &= 0 \\ Y(x,t_0) &= Y_0(x) \\ 0 \leq x \leq L, \quad t \geq t_0, \quad a &\neq 0 \end{aligned} \tag{2.2}$$

with some combination of the following boundary conditions:

$$\begin{aligned} Y(0,t) &= u_1(t), \quad \frac{\partial}{\partial x}Y(0,t) = u_1'(t) \\ Y(L,t) &= u_2(t), \quad \frac{\partial}{\partial x}Y(L,t) = u_2'(t) \end{aligned} \tag{2.3}$$

It serves us also as a test bed for more general ideas as are described for example in [40]. The system (2.2) has also an advantage, that it can be easily realized in laboratory, yet providing interesting experimental insights into the character of the problem.

The particular case, when the boundary conditions are $Y(0,t) = u(t)$ and $\frac{\partial}{\partial x}Y(L,t) = 0$ (see equation (2.13)), models a laboratory experimental apparatus in Figure 3.1. One end of the bar is isolated and the other one is heated. In equation (2.2) there is also a term $bY(x,t)$ representing heat transfer to the surrounding atmosphere which makes the identification of both parameters a, b nontrivial.

In both models (2.1) and (2.2) the state of the system can not be described in a finite dimensional space because it depends also on space in a continuous manner. A control of such systems has been studied in theory of distributed parameter systems control.

Basics of this theory are going back to late sixties. The emphasis at the beginning was to transfer the main concepts from theory of finite dimensional systems control. The fundamental contributions have been worked out by A.G. Butkovskij [11], J.L. Lions [30] and others. However, because of an infinite dimensional character of the system description, the theory is much more complicated and relies mainly on functional analytic methods. Indeed, up to date the distributed parameter systems theory is far from a unification achieved in classical control theory.

More recent is a state space theory for distributed parameter systems (see for example the book of R.F.Curtain and A.J.Pritchard [15]) based on an operator semi-group approach.

In fact, in our work we design a sort of optimal control where a signal which has to be tracked is obtained via inversion of the system model. In this sense our approach shares some ideas with that in [55].

The receding horizon state-space strategy for distributed parameter systems

which we also employ can be related to that found in [5], however, we derive our approach more from the recent development in model based predictive control as is described in [13]. A basic ideas of this approach are summarized in the following section.

As was stressed by Tichonov school [20] and others [44], to derive an optimal boundary signal from prescribed end-state of the system is an ill-posed problem. Also the author in [17] has encountered a numerical ill-conditioning in his method which has made the realization of his approach difficult.

The severely ill-conditioned character of the problem has forced us to consider a usage of special methods for regularization of ill-posed problems. They are introduced at the end of this chapter in section 2.3.

2.2 Model Based Predictive Control

In this section we briefly mention basic ideas and building blocks of model based predictive control. Among the first model based predictive control approaches have been the Identification Command [39] and Dynamic Matrix Control [18]. Even if, there are more approaches which belong to this rather general class of predictive control we focus on Generalized Predictive Control (GPC) [13] as one of the latest developments in this field.

The model based predictive control is based on the following set of ideas.

- The *process model* is used to derive the future process behaviour as a function of past inputs/outputs and as a function of hypothetical future *controls*. There is (usually finite) time domain window over which the predictions are made.
- A cost function is defined which measures the tracking error between the future system outputs as a function of future controls and reference signal. This cost is measured over certain time domain which is a sub-domain of the prediction window.
- The cost function is optimized with respect to hypothetical future control. The optimization provides a vector of optimal future controls leading to minimal tracking error.
- The control loop is closed using a so called *receding horizon* strategy where only the first element of the above vector of optimal controls is transmitted to the plant and the whole processing window is moved 1 step ahead.

GPC in particular, uses a Controlled Auto-Regressive and Integrated Moving-Average (CARIMA) process model in the following form:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta}\epsilon(t). \tag{2.4}$$

Where Δ is the difference operator $(1 - q^{-1})$ i.e. $\Delta z = z(t) - z(t-1)$, A, B represent the plant dynamics and A, C the disturbance.

The above model is used in GPC to provide a *long-range predictive* strategy where the future outputs of the model are predicted up to a *prediction horizon.* As inputs to the prediction operator the past and present outputs $(y(t-i), i \geq 0)$ and the past controls $(u(t-i), i < 0)$ are used. A distinctive feature of GPC is that the predictions are expressed as a function of future control increments $\Delta u(t+i), i \geq 0$.

According to [13] the predicted output can be decomposed to two terms. First is so called free response of the system

$$y_1(t+i) = y(t) + F_i(q^{-1})\Delta y(t) + G_1(q^{-1})\Delta u(t-1) \tag{2.5}$$

where the future controls $u(i), i \geq t$ equal $u(t-1)$ i.e. the future control increments are zero. y_1 is only a function of current output $y(t)$ and past outputs and controls.

The second element of the output decomposition

$$y_2(t+i) = G_2(q^{-1})\Delta u(t+i-1) \tag{2.6}$$

is dependent on future control increments $u(t+i), i \geq 0$.

The future system response for the whole prediction interval $i = 1, .., N$ can be written in a matrix form as

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{p} + \epsilon \tag{2.7}$$

where $\mathbf{y}$ is the vector of future outputs $y(t+i), i > 0$, $\mathbf{u}$ is the vector of future control increments, $\mathbf{p}$ is the vector of predictions according to the free response $y_1(t+i)$ and ϵ is the vector of error due to future noise terms.

If we denote as $w(t+i), i > 0$ the reference signal then the control algorithm can be synthesized from the following quadratic cost function with constraints

$$J(N1, N2, NU, \lambda) = \sum_{i=N1}^{N2} e^2(t+i) + \lambda \sum_{i=1}^{NU} \Delta u^2(t+i-1) \tag{2.8}$$

$$\Delta u(t+i) = 0, i \geq NU \tag{2.9}$$

where $\mathbf{e} = (w(t+1) - y(t+1), ..., w(t+N) - y(t+N))$ is a vector of tracking errors, and $N1, N2, NU$ and λ are the tuning parameters of GPC, namely:

- N1 - the minimum costing horizon
- N2 - the maximum costing horizon
- NU - the control horizon
- λ - the weighting factor

For a deterministic plant with $\epsilon(t+i) = 0$ the prediction equation can be substituted into the cost function. We obtain the following normal equations for the minimizer of future controls:

$$\mathbf{u} = (\mathbf{G^T G} + \lambda \mathbf{I})^{-1} \mathbf{G^T} (\mathbf{w} - \mathbf{p}) \tag{2.10}$$

Given the normal equations above we can specialize the general model based predictive approach to obtain the overall algorithm of GPC as follows:

1. If (adaptive) { Update the model parameters using recursive identification method using for example Recursive Least Squares}.
2. Compute the free response y_1.
3. Compute the reference vector $\mathbf{w}$.
4. Compute the the step response elements of the matrix $\mathbf{G}$ from the current model.
5. Solve the normal equations with a-priori given values of tuning parameters.
6. Apply the first element of computed vector of control increments to the plant i.e. apply the control $u(t) = u(t-1) + \mathbf{u(1)}$
7. Shift the data vectors one element ahead and goto step 1.

2.3 Regularization methods

It is often the situation when quantitative data are sought, that the unknown quantity can be derived only from a secondary information produced by certain equipment. Usually, the unknown quantity $\mathbf{x}$ is related to the secondary information at hand by a certain equation

$$\mathbf{Ax} = \mathbf{b} \tag{2.11}$$

When this equation describes certain measurement process it is frequently a Fredholm integral equation of the first kind. As will be described later, a numerical solution of such equation can be a real obstacle in a way to obtaining reasonable values $\mathbf{x}$ of the quantity sought. This sort of problem when a measurement is described by an integral equation like

$$b(y) = \int_a^b K(y,\tau)x(\tau)d\tau \tag{2.12}$$

arises in spectroscopics, physics, meteorology, astronomy, optics, geophysics and many other fields.

When the reader thinks more deeply how integral operator works, it can be perhaps seen intuitively why a solution of an equation (2.12) is a problem. The integration destroys the original information contained in function $\mathbf{x}$.

We can interpret the kernel $K(y,\tau)$ in the equation (2.12) as a response function of a device, which must be determined by calibration. An ideal measurement device would have response $\delta(y-\tau)$, which would immediately deliver the value $x(y) = b(y)$ sought. In the practise, we must expect much less "specific" response, which at best has a form of Gaussian distribution, which somehow contaminates and debases the input data. Unfortunately, to have such response function is still too idealistic, we have often to deal with kernels like those in Figure 2.1 resp. in Figure 2.2. These kernels describe a response of thermal system to boundary heating and they are characterized by large plateaus with a shape far from anything like Gaussian distribution.

A similar situation occurs when we are trying to describe certain physical system with a parameterized mathematical model. Let us say that a thermal

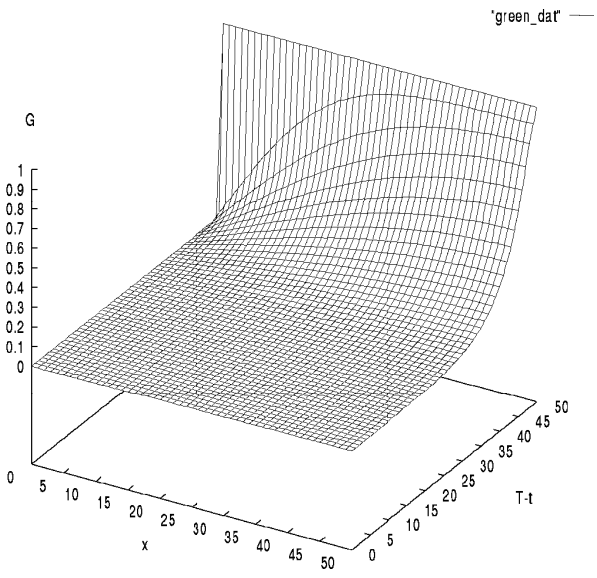


Figure 2.1: A kernel describing the response of the thermal system (2.13)

system with the configuration in Figure 3.1 is described by the following partial differential equation

$$\frac{\partial}{\partial t}Y(x,t) - a^2\frac{\partial^2}{\partial x^2}Y(x,t) + bY(x,t) = 0$$

$$Y(x,t_0) = Y_0(x), Y(0,t) = u(t), \frac{\partial}{\partial x}Y(L,t) = 0 \tag{2.13}$$

$$0 \leq x \leq L, \quad t \geq t_0, \quad a \neq 0$$

then we can ask for the values a, b for a particular system. In general, a response of the system which takes as input the system behavior and as an output the parameter values has an unstable character. This means that minor perturbation in data can cause substantial change in parameter values. This task is known as system resp. parameter identification.

Another class of problems can be posed in the same system (2.13), namely, the task of reconstruction of the temperature behavior on the one boundary if the temperature time history is known on the other boundary. Even harder question is the boundary temperature time history reconstruction from the system end-state. This is the problem described by the kernels in Figure 2.1, 2.2. Both problems are governed by integral equations of the first order of the sort (2.12). We meet such tasks, for example, when we are trying to control in an optimal way a thermal system to a prescribed interior temperature profile with a boundary heating.

Recently, there is a big interest in engineering design to formulate a new sort of inverse problems. To mention one illustrative example [56], let us suppose that we have an ingot with flat end which is rolled. The shape of the end of the ingot is generally defective after the rolling process. The aim is to pre-form an end-shape which would result in flat end *after* the rolling. The situation just described is schematically shown in Figure 2.3.

There are many other areas where engineers are trying to prepare certain special shape in order to obtain a prescribed shape after the processing. For example, a solidification front speed and shape can be planned in this way in solid state technology [57].

Another application field for regularization and ill-posed problem solving is detection of internal cracks resp. inhomogeneities from boundary data. Interesting problems are solved also in astrophysics, where surface structures of stars are reconstructed from spectroscopic and photometric rotational modulation [24].

There are many approaches developed so far in various application domains mentioned above, but generally, it can be said [24] that the regularization resp. ill-posed problem solving is an art of introducing a general prior information (like positiveness of the solution, smoothness etc.) into the inversion of the governing equation of the process. It is almost impossible to give a general framework for all methods. What can perhaps be said is that there are two fundamental frameworks. One which uses functional analytic language and arguments and another where a statistical interpretation and language is used. As an example of the former we mention a Tikhonov school, the latter is described in an axiomatic way in [46] or in [47].

In the next section we define more precisely the notion of ill-posed problem. As an example we use the equation (2.12). Then we describe in more detail the Tikhonov regularization approach. The list of references which we provide contains also sources of alternative methods.

2.3.1 Ill-posedness of the problem

It was Hadamard in 1923 who first gave a formal definition of well-posed problem, calling the rest of the problems as ill-posed.

Definition 1 *A problem is called well-posed if the following conditions are valid:*

1. *There exists a globally-defined solution for all reasonable data;*
2. *The solution is unique;*
3. *The solution depends continuously on given data;*

The problem is called ill-posed if it is not well-posed.

As an example Hadamard gave the following problem for the heat conduction equation (2.13):

$$\frac{\partial}{\partial t}Y(x,t) - \frac{\partial^2}{\partial x^2}Y(x,t) = 0$$

$$Y(x,0) = Y_0(x), Y(0,t) = 0, Y(1,t) = 0 \tag{2.14}$$

$$0 \leq x \leq 1, \quad t > 0,$$

We can formulate two problems based on this equation. One is so called **forward** problem, where the input (the data) is the temperature distribution $Y(x,0) = Y_0(x)$ at the moment $t = 0$ and the problem is to find the temperature distribution $Y(x,1) = Y_1(x)$ at the later time moment $t = 1$. This problem is well-posed, which means that function Y_1 continuously depends on Y_0. But we can state also the **inverse** problem, where the function Y_1 is given whereas we are looking for the temperature distribution Y_0 back in time. The inverse problem is highly instable.

To see the instability of the inverse problem (2.14) we can use separation of variables to obtain an invertible description of the forward problem as:

$$Y_1(x) = \sum_{k=1}^{\infty} e^{-k^2\pi^2}(Y_0, \psi_k)\psi_k$$

$$\psi_k(x) = 1/\sqrt{2}\sin k\pi x \tag{2.15}$$

$$(x,y) = \int_0^1 x(\xi)y(\xi)d\xi$$

in operator form we can write:

$$Y_1(x) = \int_0^1 G(x,\xi,t)Y_0(\xi)d\xi = (G, Y_0)$$

$$G(x,\xi,t) = \sum_k^{\infty} e^{-k^2\pi^2}\psi_k(\xi)\psi_k(x) \tag{2.16}$$

$$Y_1 = AY_0$$

where A is a compact integral operator from Hilbert space Z to Hilbert space U ($Z = U = L_2(0,1)$). The spectrum of operator A consists of values $e^{-n^2\pi^2}$. Now the instability of the inverse problem is clear, because the inverse problem can be expressed in operator form as:

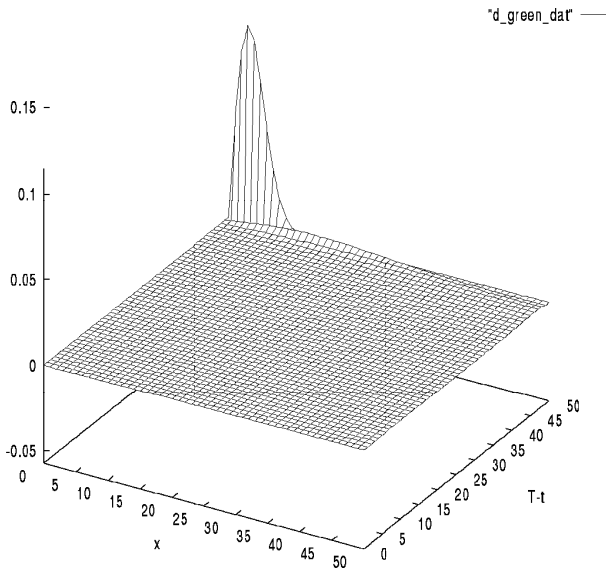


Figure 2.2: A different kernel for the same thermal system (2.13)

$$Y_0(x) = \sum_{k=1}^{\infty} e^{n^2\pi^2}(Y_1, \psi_n)\psi_n = A^{-1}Y_1 \tag{2.17}$$

where the operator A^{-1} has the spectrum $\lambda_n = e^{n^2\pi^2}$. To express such spectrum geometrically, we can say that there are directions in which arbitrarily large response to input data T_1 is possible.

At this point it is worthwhile to make two notes.

- As was said above, the main problem in the given example was that 0 is an accumulation point of the spectrum. This means that the inverse operator has arbitrarily large eigenvalues, therefore the response to input data in an inverse problem is arbitrarily large. On the other hand the example

above represents in certain sense "the worst case" because the eigenvalues approach zero with extreme speed. In many applications, the spectrum tends to zero much more slowly making the ill-posedness of the problem weaker.

- The ill-posed character is persistent also in discretized versions of the problem. Here, what we obtain is a bad conditioned matrix with a ratio σ_1/σ_n (the largest and the smallest singular value) which usually spans the whole resolution of the floating point representation. The solutions are wildly oscillatory when standard methods for solution are used.

Now, let us turn back to the definition 1. The discussion above has shown how the instability in ill-posed inverse problem occurs. But we did not discus the problems of existence and uniqueness expressed in the points 1,2 of definition 1. In the example, we have supposed that the operator is invertible, so there is a unique inverse element.

However, for linear ill-posed problems we can reformulate the basic problem

$$\mathbf{Ax} = \mathbf{b} \tag{2.18}$$

as minimizing the following least squares functional:

$$\mathbf{F(x)} = ||\mathbf{Ax} - \mathbf{b}||^2 \tag{2.19}$$

Then all minimal elements must satisfy the normal equation

$$\mathbf{A}^*\mathbf{Ax} = \mathbf{A}^*\mathbf{b} \tag{2.20}$$

It can be shown that under reasonable assumptions there is unique solution of the above normal equation, therefore we can generally suppose that there is unique solution to (2.11) and concentrate the effort to solving the instability problem.

2.3.2 Tikhonov regularization

As we have illustrated above the instability of an ill-posed problem is caused by accumulation of spectrum near the zero. Tikhonov (in 1943 - see the historical notes in [48]) has introduced a foundation for the method which use a certain prior information to decide how to "shift" the spectrum away from zero. The basic idea is that instead of minimizing the least squares functional (2.19) we minimize the following "smoothing" functional:

$$\mathbf{M}^{\alpha}(\mathbf{x}) = ||\mathbf{A}_h\mathbf{x} - \mathbf{b}_\delta||_1^2 + \alpha||\mathbf{B}\mathbf{x}||_2^2 \tag{2.21}$$

where $\mathbf{A}, \mathbf{A}_h, \mathbf{B}$ are operators from Hilbert space Z to Hilbert space U; D is closed, convex set of constraints, $||\mathbf{A}_h - \mathbf{A}|| \leq h$ and $\mathbf{b}_\delta$ is perturbed right hand side of the equation (2.11). α is so called *smoothing resp. regularization parameter*. Then the normal equations for (2.21) are:

$$(\mathbf{A}^*\mathbf{A} + \alpha\mathbf{B}^*\mathbf{B})\mathbf{x} = \mathbf{A}^*\mathbf{b}_\delta \tag{2.22}$$

The stabilization of the ill-posed problem is in fact achieved by minimizing certain functional $||\mathbf{B}\mathbf{x}||_2$ together with the solution itself. Choosing appropriately the operator B respectively the norm $||.||_2$ allows to express various physically reasonable features of the solution like smoothness etc. In practise, B is often chosen as E identity operator and the norm $||.||_2$ as the W_2^1 norm. There is also a possibility to express non-negativity and monotonicity within this framework by setting appropriately the constraints D. But here instead of solving normal equation (2.22) special methods are needed [50].

However, when we turn back to original problem, the most difficult part is to identify the appropriate value of smoothing factor α. If α is chosen too small, then the problem is still too ill-posed (the solutions are oscillatory), on the other hand if α is too large, the solution is excessively damped and the original equation has a weak influence to it.

There are different approaches how to choose the factor α [33], but the standard method consists in the following. We define so-called *generalized residuum* as :

$$\rho(\alpha) = ||\mathbf{A}_h\mathbf{x}_\alpha - \mathbf{b}_\delta||^2 - (\delta + h||\mathbf{x}_\alpha||)^2 - \mu^2(\mathbf{b}_\delta, \mathbf{A}_h) \tag{2.23}$$

where $\mathbf{x}_\alpha$ is a solution of the smoothing functional (2.21) and

$$\mu^2(\mathbf{b}_\delta, \mathbf{A}_h) = \inf_{y \in D} ||\mathbf{A}_h\mathbf{y} - \mathbf{b}_\delta|| \tag{2.24}$$

is the *degree of inconsistency*. The regularization parameter α of the smoothing functional is chosen by *generalized principle of residuum*, which is the following.

If the condition:

$$||\mathbf{b}_\delta||^2 > \delta^2 + \mu^2(\mathbf{b}_\delta, \mathbf{A}_h) \tag{2.25}$$

is not fulfilled, the approximate solution of the equation (2.11) is $\mathbf{x} = 0$. If the condition (2.25) is fulfilled, then the generalized deviation (2.23) has a positive root α^* and solution of the equation (2.11) is chosen as *minimum* x_{α^*} of the smoothing functional (2.21).

The fact that the generalized deviation $\rho(\alpha) = 0$ has a unique positive root when the condition (2.25) holds needs relatively fine analysis [50]. However, finally we can find a regularized solution of the equation (2.11) by the following iterative procedure :

1. Choose an arbitrary (sufficiently large) value of the parameter α
2. Minimize the functional $\mathbf{M}^\alpha(\mathbf{x})$ with respect to constraints respectively solve the normal equations (2.22)
3. Evaluate the generalized deviation $\rho(\alpha)$
4. Search for the root α^* of the equation $\rho(\alpha) = 0$ with accuracy ε, it means check the condition:

$$\mid \rho(\alpha) \mid \leq \varepsilon \tag{2.26}$$

where $\varepsilon = C * \delta$ and $C < 1$ is a constant which depends on the desired accuracy of the root α^*

5. If the condition (2.26) is not fulfilled the procedure is repeated from the point 2. Otherwise the *minimum* x_{α^*} is the regularized solution of the equation (2.11).

In this short introduction, we did not mention another relatively large area of investigations where ill-posed problems are interpreted within statistical framework like [21] resp. [47] which contain also exhaustive lists of references.

It must be also noted that the instability in the examples shown (inversion of the heat transfer equation resp. deconvolution) is substantial in the sense that having arbitrarily accurate data (with an arbitrary small error - perturbation) does not avoid the difficulties. Perhaps, this is the difference between another class of problems where statistical approach seems to be fruitful. The latter being a class of problems where the data are given with a large uncertainty.

There are also different methods working within the same framework as Tikhonov regularization. Among others we mention sequential method of Beck [8] and Mollification Method [36]. The conference proceedings [56] represent also a rich collection, where all above mentioned approaches and their combinations are presented on some engineering application.

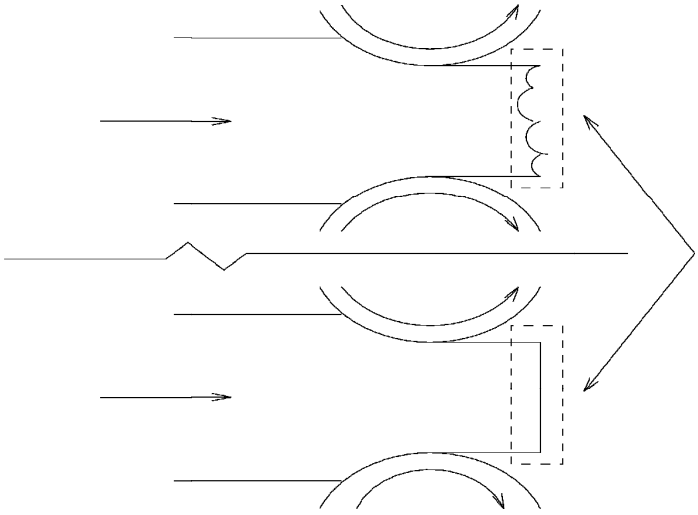


Figure 2.3: An inverse problem associated with the rolling process. A profile is sought, which will give flat-end *after* the processing.

Chapter 3

Predictive Boundary Control of Distributed Parameter System

In the following chapter we describe a predictive approach to the control of distributed parameter systems. To maintain a certain simplicity of description we speak during the whole chapter about a concrete 1-dimensional distributed parameter system.

However, except the identification procedure, the approach presented here can be straightforwardly generalized to more spatial dimensions. With the word "straightforward" we mean that the basic conceptual framework would be the same. On the other hand this step is by no means trivial because it involves replacing all the basic blocks of our 1-dimensional procedure with the 2/3-dimensional blocks with the same functionality. The complexity of the description and computation grows radically during this replacement. But this is a common problem for processes described by partial differential equations.

Our approach is built step by step starting with a simulation-prediction analysis in Section 3.2 proceeding through open-loop step-wise control to the final close-loop predictive algorithm.

The identification step in Section 3.1 is auxiliary to our development but allows us to clearly link our approach to the experimental verification on a laboratory specimen. Therefore the identification procedure is particularly specialized (off-line) to an identification of the parameters of the laboratory specimen.

The basic idea of our approach to predictive boundary control of distributed parameter system starts with a rather straightforward replacement of the basic predictive control blocks (i.e. predictor, tracking error optimization and receding horizon strategy) with similar blocks for a distributed parameter system. But this replacement is not trivial in its aspects.

The main problem lies in the extreme ill-conditioning characteristic for distributed parameter system which do not allow to carry out the algorithm without using special methods to solve it. Here an almost perfect analogy, between predictive control and known schemes developed in inverse engineering field for ill-posed problems solving, which we have found, is of considerable help. This basic analogy is described in more detail in Section 3.4.

There is one well developed aspect of predictive control of finite dimensional systems which we are lacking in the following sections and that is the modeling and a treatment of disturbance - noise in the controlled system. However, there are at least two reasons why we postpone this question to future investigations. Firstly, the complexities involved even in a deterministic case are so big that including of a noise model would draw us out of a time frame for this project. Another important reason is that it seems to be practical to wait until at least the deterministic scheme proves its usefulness and settles down to some canonic design.

As was said at the beginning, for the sake of simplicity we would speak in this chapter about the following concrete distributed parameter system:

$$\begin{aligned} \frac{\partial}{\partial t}Y(x,t) - a^2\frac{\partial^2}{\partial x^2}Y(x,t) + bY(x,t) = 0 \\ Y(x,t_0) = Y_0(x), Y(0,t) = u(t), \frac{\partial Y(L,t)}{\partial x} = 0 \\ 0 \leq x \leq L, \quad t \geq t_0, \quad a \neq 0 \\ a^2 = \frac{\lambda}{c.\rho}, \quad b = \frac{h}{c.\rho} \end{aligned} \tag{3.1}$$

where: L is the length of the bar in meters; λ is thermal conductivity coefficient; a^2 is thermal diffusivity coefficient in m^2/s; c is the specific heat in $\frac{J}{kg\ K}$; ρ is specific mass of the bar in kg/m^3 and h is the heat-transfer coefficient in

$\frac{W}{m^2\ K}$. In the remaining text $Y(x,t)$ denotes the deviation of temperature from the surrounding fluid in K.

As the notation suggests it is a heat conduction/heat transfer 1-Dimensional equation which describes heat conduction in a metal bar from the heated end at the spatial coordinate 0 to the isolated end at the spatial coordinate L with a thermal diffusivity coefficient a^2. The equation describes also a heat transfer to the surrounding air expressed by term $bY(x,t)$ where b contains the heat transfer coefficient.

The control task is to drive the temperature along the spatial coordinate to the prescribed reference sequence of temperature profiles with a boundary heating represented by the function $Y(0,t) = u(t)$.

The physical model described by the above equation consists in a metal bar with a boundary heater on one side and an isolation on the other side. The laboratory specimen is described in Figure 3.1.

There is also another reason why we have chosen this particular representation of distributed parameter system and it is that there is certain hope to find a dimensionality reduction procedure leading to a description of a rectangular slab heating which uses exactly this sort of 1-Dimensional systems as subsystems.

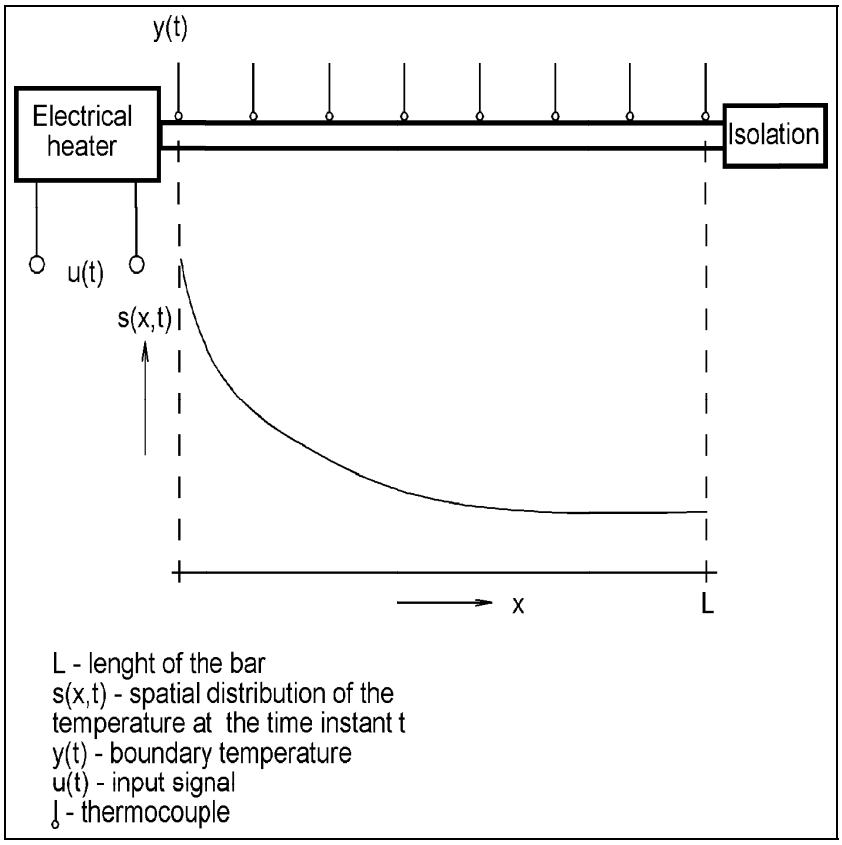


Figure 3.1: The experimental specimen consisting in the heated metal bar. On the one end of the bar there is a heater. Another end is isolated. There are eight thermocouples along the bar. The main physical processes are heat conduction from the heated end and a heat transfer to the surrounding air. The power of the heater allows a temperature range up to a 300 degrees Celsius. In the figure an exception from the notation convention is made because the controlled variable here is not meant to be the distributed state but the boundary temperature.

3.1 Identification

Certainly, making a relation between equation (3.1) and the laboratory specimen in Figure 3.1 raises a question about the accuracy of the system description given by equation (3.1). As will be seen later, when the values of thermal diffusivity coefficient a^2 and heat transfer coefficient h are appropriately chosen the description is rather accurate with exception of transient states with fast dynamics. However, when the process slows down, the error drops back to a value of about 1%.

The accurate off-line identification of coefficients a, b is based on the observation that switching off the heater change the boundary condition to the isolated end with Neumann boundary condition. Subsequently, this boundary condition change pushes the system faster to the state with constant temperature along the bar. Once we can suppose the temperature to be constant along the bar at a sufficiently high value, we have a starting point from which the system behaviour is described by a simple ordinary differential equation. From this equation the accurate identification of the value of coefficient b (containing the heat transfer coefficient) follows. Knowing the value of b we can accurately identify the value of a by driving the system to a stable state or to a quasi-stationary process.

The above drawn method can be seen as a physical realisation of Furier's variable separation method because it is based on processes which drive the system to the states where $\frac{\partial}{\partial t}Y(x,t)$ or $\frac{\partial^2}{\partial x^2}Y(x,t)$ vanish.

In the subsequent parts, the identification of coefficient b is described in more detail. Then based on the latter the coefficient a is identified. The comparision of simulated system which uses identified coefficients and measured data is postponed to section 3.2 where we put also some notes about possible model improvements.

3.1.1 Heat Transfer Coefficient h and Coefficient b

The configuration of an experimental specimen on which the method has been verified is in Figure 3.1. It consists of coper metal bar which is heated on one boundary and isolated on other boundary. There are 8 thermocouples installed on the bar which can be used for temperature profile verification.

The offline identification is based on the fact that after reaching stable state of the system on some higher temperature we can switch the boundary condition to the isolated-end Neumann type condition simply by switching off the heater. The system will reach relatively soon a state when the $\frac{\partial Y(x,t)}{\partial x}(x,t) \leq \epsilon$ for all $0 \leq x \leq L$ i.e. the temperature along the bar is nearly constant. In our case the system has been heated to the stable state with boundary temperature 260 degrees Celsius. Subsequently after switching off the heater the system reached the state where ϵ was under the level of noise at about 60 degrees. From this moment the behaviour of the system is described by a simple first order equation $\frac{d}{dt}Y + bY = 0$.

The solution of the latter equation is $Y(t) = Y(0)e^{-bt}$ where $Y(0)$ is a temperature at time $t = 0$ and $Y(t)$ is a temperature at time t. From this we can express parameter b as $b = -\frac{1}{t}\ln\frac{Y(t)}{Y(0)}$. The great simplification of the governing equation provides a possibility for high precision identification of the heat transfer coefficient.

The course of the identification experiment is shown in Figure 3.2 and in Figure 3.3. In both figures the x-axis represents a spatial coordinate where point 0 (heater) is in lower left corner and L (isolation) is in lower right corner. The y-axis represents a temperature in the range 0-270 degree. Each figure containg a system evolution has a small subgraph in the upper right corner displaying the boundary temperature as a function of time (heating).

The Figure 3.2 shows the whole experiment starting with full power heating, then achieving a stable state at 260 degrees Celsius and then switching off the heater and temperature fall to zero with heater switched off. The heated-end of the bar is also isolated therefore if the heater is switched off the end of the bar behaves as an isolated end.

The Figure 3.3 displays only the part of the experiment which is relevant to the identification. It starts from the stable state which is visible as a thick temperature profile. Then the heater is switched off and the temperature falls to zero. There is also an interesting effect of momentary loss of convexity of the solutions stressed by a circle.

The results of the experiment in terms of values of the coefficient b are in

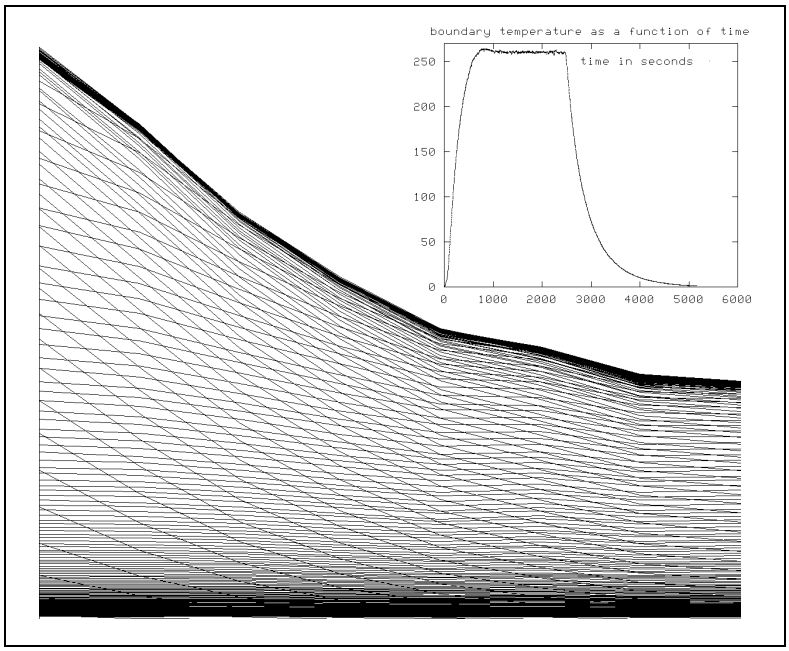


Figure 3.2: In the figure the x-axis represents a spatial coordinate where point 0 (heater) is in lower left corner and L (isolation) is in lower right corner. The y-axis represents a temperature in the range 0-270 degree. The subgraph in the upper right corner displays the boundary temperature in time (heating). The figure shows the whole experiment starting with full power heating, then achieving a stable state at 260 degrees Celsius and then switching off the heater and temperature fall to zero with heater switched off.

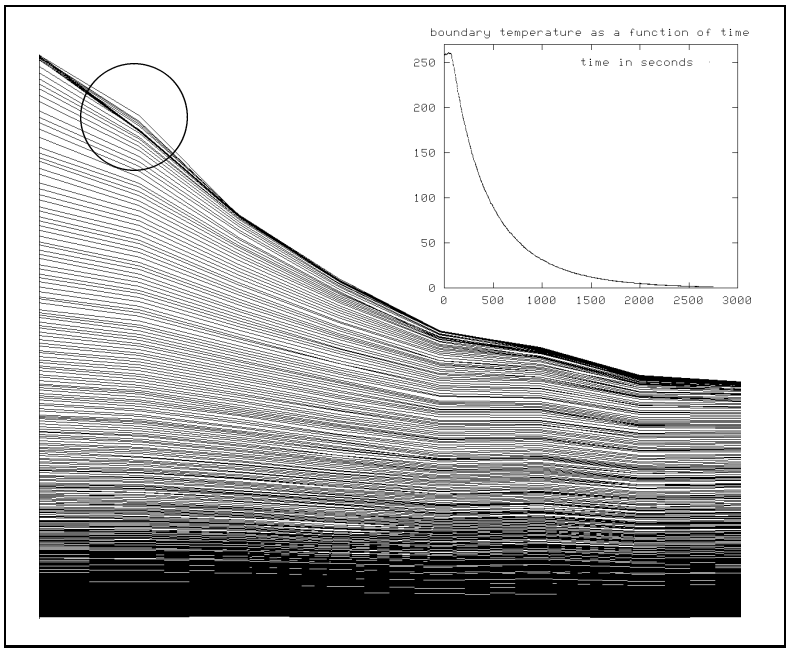


Figure 3.3: A part of the whole system evolution from Figure 3.2. The evolution in this figure starts with the stable state, covers the heater switching and temperature fall to zero. An interesting momentary loss of convexity is labeled with the circle.

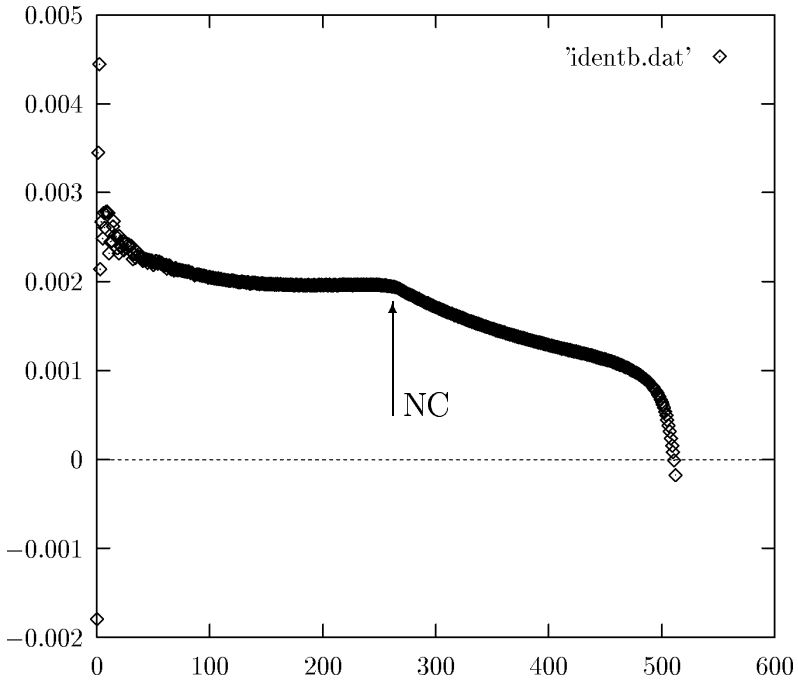


Figure 3.4: The identification of parameter b. The unit on the x-axis represents 10 seconds. The x-axis shows the computation of b in time going backwards

Figure 3.4. We use more and more points for computing b going back in time. The moment when the state of the system is going to be significantly different from a constant along the bar is labeled with the marker NC. The resulting value is $b = 0.002$.

Once we know accurately b (resp. the heat transfer coefficient h) it is easy to compute the thermal diffusivity coefficient by driving the system to a steady state and solving the boundary problem for the first order ordinary differential equation.

3.1.2 Thermal Diffusivity Coefficient a^2

When the thermal system described by equation 3.1 resp. modelled by laboratory specimen in Figure 3.1 is in steady state or in quasi-stationary process we can suppose that $\frac{\partial Y(x,t)}{\partial t} = 0$.

Making the above assumption simplifies the description of the thermal process to the following boundary problem for the second order ordinary differential

equation:

$$-a^2\frac{d^2Y(x)}{dx^2} + bY(x) = 0. \qquad (3.2)$$
$$Y(0) = Y_0$$
$$\frac{dY(L)}{dx} = 0.$$

The general solution of equation 3.2 is given as

$$Y = C_1e^{-rx} + C_2e^{rx} \qquad (3.3)$$

with the characteristic equation $r^2 = \frac{b}{a^2}$. The boundary conditions give $Y_0 = C_1+C_2$ and $\frac{C_1}{C_2} = \frac{e^{rL}}{e^{-rL}}$. Then the values of the constants C_1 and C_2 are $C_1 = \frac{Y_0e^{2rL}}{1+e^{2rL}}$ and $C_2 = \frac{Y_0}{1+e^{2rL}}$. Subsequently, we substitute C_1 and C_2 to the solution 3.3 and write a value $Y(L)$ of the solution in the space point L (the isolated end of the bar). We obtain the following expression

$$Y(L) = 2\frac{Y_0e^{rL}}{1+e^{2rL}} \qquad (3.4)$$

When the value $Y(L)$ is supposed to be known from the experiments the constant r can be computed from the expression 3.4 using a substitution $s = e^{rL}$. After the substitution we obtain a quadratic equation for s, $ks^2 - s + k = 0$ where $k = \frac{Y(L)}{2Y_0}$. Taking into the account the characteristic equation of (3.2) the value of a follows.

The identification experiment is displayed in Figure 3.5. The system is brought (heated) to a stable state at a boundary temperature 260 degrees. Then a quasi-static decrease of the boundary temperature to 60 degrees occurs. The temperature decrease is linear in time and in each moment we can take a "snapshot" of the system state and use it to determine the coefficient a according to the above described procedure.

The resulting values of the coefficient a during the whole process are in Figure 3.6. The relevant part of the data for identification, which consist in the data

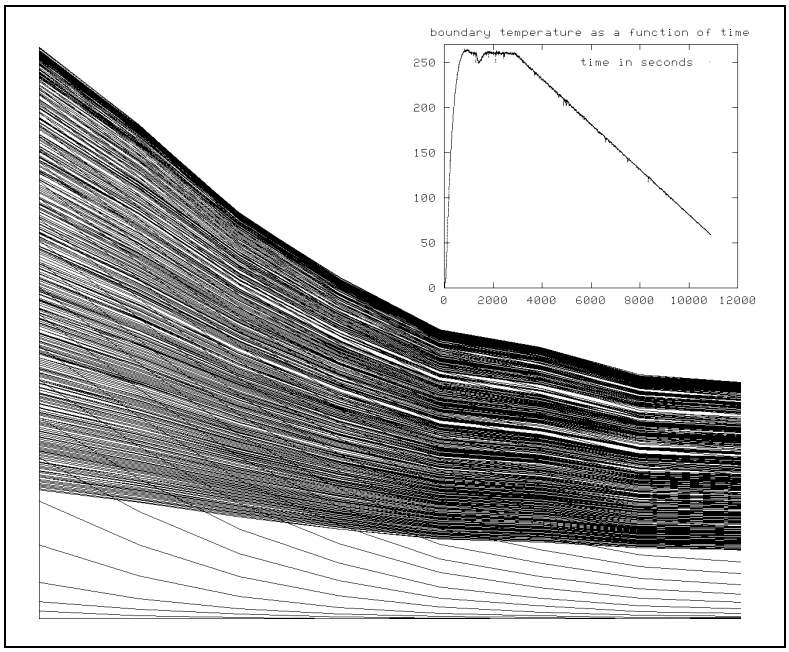


Figure 3.5: The system is heated to a stable state at a boundary temperature 260 degrees. Then a quasi-static decrease of the boundary temperature to 60 degrees occurs.

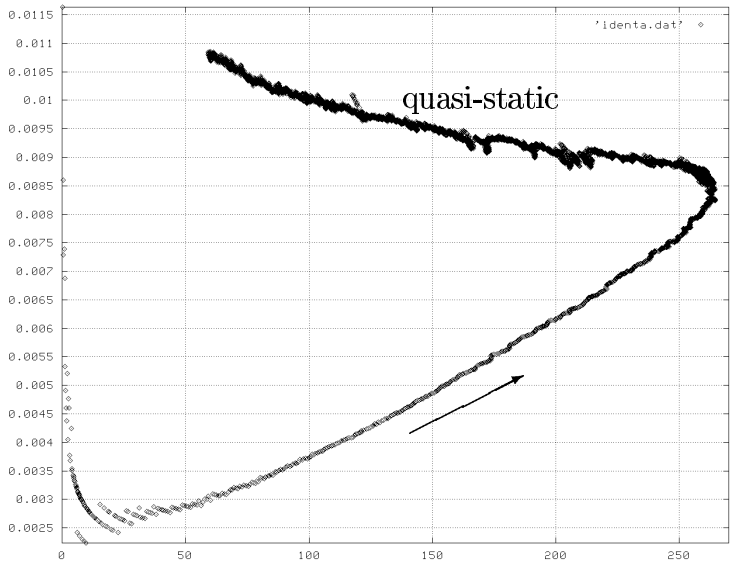


Figure 3.6: On the x-axis is a temperature and on the y-axis is a value of the coefficient a. The relevant part of the data for identification of the coefficient a, which consist in the data obtained from the quasi-static process are marked with "quasi-static". The arrow shows the direction of time during the system evolution.

obtained from the quasi-static process are marked with "quasi-static". The arrow shows the direction of time during the system evolution.

It is also interesting to notice that the first part of the curve is almost linear even if during the first phase the system is going through transient states which are far from the stationary state for the respective boundary temperature.

It is well known fact that the coefficients a, b are temperature dependent. It can be observed also on the results of the identification proposed above, however including the temperature dependency of the coefficients makes the model non-linear. This fact can be a considerable obstacle for certain approaches as is for example for the modelling with Green's functions. Then, a possibility how to proceed is to choose appropriately a working point for the system.

As the results of simulation show this is well possible if the temperature range in which the system is operating is not very large. The experiments made during this work on an experimental device (see Figure 3.1) in the temperature range 0-300 degrees Celsius show that temperature range of few hundred degrees can

be addressed with suitably chosen constant coefficients.

On the other hand, there are methods which are less sensitive to the above mentioned non-linearity and we will use them in the following sections.

3.2 Simulation - Prediction

As we have stated in the introduction of this chapter to implement successfully a predictive control scheme we need a prediction tool for a distributed parameter system. In fact the words prediction and simulation stress a two different points of view to the same thing. It is the solution of the governing equation of the system.

However, the solution i.e. the system simulation have to have some "good" features to be useful as a prediction tool for predictive control scheme. Mainly we need a numerical stability and efficient computability.

In this section we analyse two approaches to simulation on which a predictor can be built. An approach with integral representation and Green functions and another one with finite difference schemes.

3.2.1 Integral Representation with Green's Function

It is well known that the solution of the equation 3.1 can be expressed in the following integral form:

$$Y(x,t) = \int_{t_0}^{t} \int_{0}^{L} G(x,\xi,t-\tau)\ w(\xi,\tau)\ d\tau d\xi \tag{3.5}$$

where $w(x,t)$ is a *standardizing* function (see [12]):

$$w(x,t) = Y_0(x)\delta(t) + a^2\delta'(x)u(t)$$

which includes an exciting function, boundary and initial conditions and $\delta(.)$ is Dirac function. The heating of the bar is controlled through the boundary temperature $u(t) = Y(0,t)$ and the task is to find such function $u(t)$ - boundary heating of the bar - which ensures us attainment of the required spatial distribution of the bar temperature $Y(x,t)$ at specified time instant t_v. In this situation the relation (3.5) simplifies to the following form (see also Appendix A for a derivation from first principles):

$$Y(x,t_v) = a^2 \int_{t_0}^{t_v} \frac{\partial}{\partial \xi} G(x,\xi,t_v-\tau) \mid_{\xi=0} \; u(\tau)\, d\tau + \\ + \int_0^L G(x,\xi,t_v-t_0)\; Y_0(\xi)\, d\xi \tag{3.6}$$

where $Y_0(x) = Y(x,t_0)$ is a given initial condition. The Green function $G(x,\xi,t)$ for the above problem is:

$$G(x,\xi,t) = \frac{2}{L} \sum_{k=0}^{\infty} \exp\left(-bt - \lambda_k t\right) \varphi_k(x) \varphi_k(\xi) \tag{3.7}$$

where $\varphi_k(x) = \sin \frac{\sqrt{\lambda_k}}{a} x = \sin \frac{(2k+1)\pi}{2L} x$ are eigenfunctions of Sturm-Liuville problem for (3.1) with eigenvalues $\lambda_k = \frac{(2k+1)^2 \pi^2 a^2}{4L^2}$. It is worthwhile at this moment to see (Figure 3.7) how the spectrum of the above Sturm-Liuville problem looks for the concrete values of the thermal diffusivity coefficient a identified in the previous section.

To be able to predict the system behaviour according the equation (3.6) it is necessary to have an efficient computational procedure for the kernels

$$\frac{\partial}{\partial \xi} G(x,\xi,t) \mid_{\xi=0} = G_{\partial\xi}(x,t)$$

$$G(x,\xi,T) = G_T(x,\xi)$$

A simple derivation from (3.7) or a derivation from the first principles according to Appendix A shows that

$$G_{\partial\xi}(x,t) = \frac{\pi}{L^2} \sum_{k=0}^{\infty} (2k+1) \exp\left(-bt - \lambda_k t\right) \varphi_k(x) \\ G_T(x,\xi) = \frac{2}{L} \sum_{k=0}^{\infty} \exp\left(-bT - \lambda_k T\right) \varphi_k(x) \varphi_k(\xi) \tag{3.8}$$

An optimized numerical procedure for computation of the above kernels (Fourier series) is a subject of the next subsection. Then we conclude this part with subsection 3.2.1.2 providing a discussion of numerical difficulties near the heated boundary.

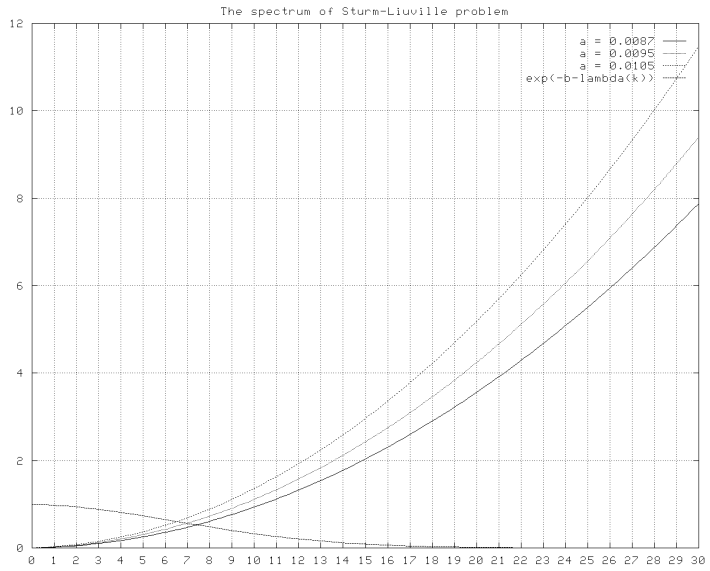


Figure 3.7: A sensitivity of the spectrum of Sturm-Liuville problem to the change of thermal difussivity coefficient. The coefficient range is used according to the identified values for the experimental device. On the x-axis are the values of k and the y-axis represents the eigenvalues size.

3.2.1.1 Optimization of Green's function computation

As was noted in [49] (pp.218), the summation of the Fourier series is itself an ill-posed problem. Even if the ill-posedness is weaker than that of the heat conductivity inversion it is still good to know the reasons for the numerical problems which sometimes occur. During the computation of the Green's function it can be observed that slight changes in the algorithm, or bounds, are leading to very different results.

We propose an optimized method for the computation of the Green's function which contains accurate error bounds. The advantage of such a method can be seen when accurate results are needed near zero in time and near the heated boundary in space. In this case the points near time zero can easily require many thousands of the terms in the series but as the time increases to seconds this number immediately drops to less than 10 (see the numerical study at the end of this section).

We first turn our attention to the kernel $G_{\partial\xi}(x,t)$ and later it will be seen that the kernel $G_T(x,\xi)$ can be expressed in a similar form.

The following well known convergence criterion together with a bound is used in the sequel:

Theorem 2 (Dirichlet criterion) *Let* $s_n = \sum_{k=0}^{n} a_k$ *is an infinite series of complex numbers and all the partial sums are bounded by* $|s_n| \leq K$. *Let* α_k *is a monotone sequence of real numbers with* $\lim_{k\to\infty} \alpha_k = 0$. *Then the series* $\sum_{k=0}^{n} \alpha_k a_k$ *is convergent and* $|\sum_{k=0}^{\infty} \alpha_k a_k| \leq K|\alpha_0|$.

The algorithm is based on the following lemma:

Lemma 3 *Let* $0 < x \leq L$ *and* $0 < t$. *If* $l \geq k_0 = \lceil (\frac{L}{\pi a}\sqrt{\frac{2}{t}} - 1)/2 \rceil$ *then the residual part* $R(l) = \sum_{k=l}^{\infty} (2k+1) \exp(-bt - \lambda_k t)\varphi_k(x)$ *of the series for* $G_{\partial\xi}(x,t)$ *is bounded by* $|R(l)| \leq (2l+1) \exp(-bt - \lambda_l t) \frac{2}{|sin\overline{x}|}$ *where* $\overline{x} = \frac{\pi x}{2L}$.

Proof. Let us denote as $M_k = (2k+1) \exp(-bt - \lambda_k t)$. $M_k > 0$ is positive for all $k \geq 0$. M_k as a function of $k \geq 0$ has only one extremal point as follows from the properties of exponential function. This extremal point can be computed taking the derivative of M_k as a function of k.

$$\frac{d}{dk}((2k+1)\exp(-bt-\lambda_k t))=0$$

$$k_0=\lceil(\frac{L}{\pi a}\sqrt{\frac{2}{t}}-1)/2\rceil$$

Thus for all $k \geq k_0$ the sequence M_k is monotonically decreasing to zero.

Now, recall that

$$\sum_{k=0}^{n}\sin kx=\frac{\cos\frac{x}{2}-\cos(n+\frac{1}{2})x}{2\sin\frac{x}{2}} \tag{3.9}$$

$$\sum_{k=0}^{n}\cos kx=\frac{\sin(n+\frac{1}{2})x+\sin\frac{x}{2}}{2\sin\frac{x}{2}} \tag{3.10}$$

The above sums are bounded by $|\sum_{k=1}^{n}\sin kx| \leq 1/|\sin\frac{x}{2}|$, $|\sum_{k=1}^{n}\cos kx| \leq 1/|\sin\frac{x}{2}|$. Let us bound the sum $\sum_{k=1}^{n}\sin(2k+1)x$. Use the substitution $y=2x$

$$\sum_{k=0}^{n}\sin(2k+1)x=\sum_{k=0}^{n}\sin(ky+\frac{y}{2})=\cos\frac{y}{2}\sum_{k=0}^{n}\sin ky+\sin\frac{y}{2}\sum_{k=0}^{n}\cos ky$$

$$|\sum_{k=0}^{n}\sin(2k+1)x| \leq \frac{2}{|\sin\frac{y}{2}|}=\frac{2}{|\sin x|}$$

Note also that by the same reasoning $|\sum_{k=l}^{n}\sin(2k+1)x| \leq \frac{2}{|\sin x|}$ because $\sum_{k=l}^{n}\sin(2k+1)x=\sum_{k=0}^{n-l}\sin(2k(2l+1))x$.

Finally, using a substitution $\overline{x}=\frac{\pi x}{2L}$ we are prepared to obtain the lemma statement because of Dirichlet criterion.

$$|R(l)|=|\sum_{k=l}^{\infty}M_k\sin(2k+1)\overline{x}| \leq M_l\frac{2}{|\sin\overline{x}|}=$$

$$=(2l+1)\exp(-bt-\lambda_l t)\frac{2}{|sin\overline{x}|}$$

QED

If an absolute error is needed, the number of series members can be computed in advance. In the case of relative error, the algorithm must check the ratio:

$$\frac{\sum_{k=0}^{l-1}(2k+1)\exp(-bt-\lambda_k t)\varphi_k(x)}{(2l+1)\exp(-bt-\lambda_l t)\frac{2}{|sinx|}}$$

Finally, concerning the extremal cases, apparently $G_{\partial\xi}(0,t)=0$ for $0<t$ but for $t=0$ the value of $G_{\partial\xi}(x,t)$ is not defined because the series is divergent.

The kernel $G_T(x,\xi)$ represents a more complicated case. First of all, $G_T(x,\xi)$ is not defined when $T=0$. For $T>0$, $G_T(x,\xi)$=0 if $x=0$ or $\xi=0$. Except for the special case when $x=\xi$, similar reasoning can be used as in Lemma 3 when we take additionally into the account the following:

$$\varphi_k(x)\varphi_k(\xi)=\frac{1}{2}[\cos\frac{(2k+1)\pi}{2L}(x-\xi)-\cos\frac{(2k+1)\pi}{2L}(x+\xi)]. \tag{3.11}$$

Therefore

$$|\sum_{k=0}^{n}\varphi_k(x)\varphi_k(\xi)|\leq\frac{1}{|\sin(\overline{x}-\overline{\xi})|}+\frac{1}{|\sin(\overline{x}+\overline{\xi})|}.$$

We can prove

Lemma 4 *Let $0<x\leq L$, $0<\xi\leq L$, $x\neq\xi$ and $0<T$. If $l\geq 0$ then the residual part $R(l)=\sum_{k=l}^{\infty}\exp(-bt-\lambda_k t)\varphi_k(x)\varphi_k(\xi)$ of the series for $G_T(x,\xi)$ is bounded by $|R(l)|\leq\exp(-bt-\lambda_l t)\left(\frac{1}{|\sin(\overline{x}-\overline{\xi})|}+\frac{1}{|\sin(\overline{x}+\overline{\xi})|}\right)$ where $\overline{x}=\frac{\pi x}{2L}$.*

A same note concerning the relative error of a computation applies as before.

The case when $x=\xi$ needs a special consideration because in this case the term $\sum_{k=0}^{n}\varphi_k(x)\varphi_k(\xi)=\sum_{k=0}^{n}\varphi_k^2(x)$ is not bounded. However, the series is convergent because of the exponential part. To know the error bounds we can write using (3.11) that

$$G_T(x,x)=\frac{1}{L}\sum_{k=0}^{\infty}\exp(-bT-\lambda_k T)\cos(2k+1).0-$$
$$-\frac{1}{L}\sum_{k=0}^{\infty}\exp(-bT-\lambda_k T)\cos(2k+1)(\overline{x}+\overline{x})=$$
$$=\frac{1}{L}\sum_{k=0}^{\infty}\exp(-bT-\lambda_k T)-\frac{1}{L}\sum_{k=0}^{\infty}\exp(-bT-\lambda_k T)\cos(2k+1)(2\overline{x})$$

For the second term above, by the same method as in Lemma 3 we obtain the error bound $|R(l)| \leq \exp(-bT - \lambda_l T)\frac{1}{|\sin 2\overline{x}|}$. The first term is more interesting because it contains a new case of series with positive elements for which we can propose the following method

Lemma 5 *If* $l \geq \left(\frac{1}{8\overline{a}T} - \frac{1}{2}\right)$ *then the residual part* $R(l) = \sum_{k=l}^{\infty} \exp(-bT - \lambda_k T)$ *of the series* $\sum_{k=0}^{\infty} \exp(-bT - \lambda_k T)$ *is bounded by* $|R(l)| \leq \exp(-bT - \lambda_l T + l)2^{1-l}$ *where* $\overline{a} = \frac{\pi a}{2L}$, a^2 *being the thermal diffusivity coefficient.*

Proof. Let $\sum_{k=0}^{\infty} x_k$ is the original series with positive elements $x_k > 0$ for which we want to derive an error bound. Denote $R(l) = \sum_{k=l}^{\infty} x_k$ the residual part of the series.

The main idea of the method is that certain fraction of the "convergent" character of the series is extracted via multiplying and dividing $\sum_{k=0}^{\infty} \frac{x_k}{\alpha_k}\alpha_k$ by a suitably chosen convergent series with positive elements. α_k is chosen so that we know the sum $A = \sum_{k=0}^{\infty} \alpha_k$ and at the same time the sequence $\frac{x_k}{\alpha_k}$ goes monotonically to zero. Then by virtue of Dirichlet criterion we obtain $|R(l)| = |\sum_{k=l}^{\infty} \frac{x_k}{\alpha_k}\alpha_k| \leq \frac{x_l}{\alpha_l}A$. It is even better if we know explicitly also the residual part $A(l) = \sum_{k=l}^{\infty} \alpha_k$ of the auxiliary series. Then we can improve the bound as $|R(l)| \leq \frac{x_l}{\alpha_l}A(l)$.

For the special case of the series in the lemma statement, we choose $\alpha_k = q^k$, then the residual part equals $A(l) = \sum_{k=l}^{\infty} q^k = \frac{q^l}{1-q}$. The last question is to find l_0 and q so that the sequence $\frac{\exp(-bT-\lambda_k T)}{q^k}$ is monotonically convergent to zero for $k \geq l_0$.

$$\frac{\exp(-bT - \lambda_k T)}{q^k} = \exp(-bT)\exp(-(2k+1)^2\overline{a}^2 T - k\ln q)$$

The requirements can be reformulated in terms of derivative of the exponent in the above expression as

$$4(2k+1)\overline{a}^2 T + \ln q \geq 0$$

Substituting $q = 1/2$ we obtain the statement of the lemma by virtue of the Dirichlet criterion

$$k \geq \left(\frac{1}{8\overline{a}T} - \frac{1}{2}\right).$$

QED

In the next section the kernel $G_1(x,t) = \frac{4}{\pi}\sum_{k=0}^{\infty}\frac{1}{2k+1} \times$
$\times \exp(-bt - \lambda_k t)\varphi_k(x)$ is also mentioned from the point of view of numerical properties. Therefore an algorithm based on the following lemma 6 can be provided:

Lemma 6 *Let* $0 < x \leq L$ *and* $0 \leq t$. *If* $l \geq 0$ *then the residual part* $R(l) = \sum_{k=l}^{\infty}\frac{1}{(2k+1)}\exp(-bt - \lambda_k t)\varphi_k(x)$ *of the series for* $G_1(x,t)$ *is bounded by* $|R(l)| \leq \frac{1}{(2l+1)}\exp(-bt - \lambda_l t)\frac{2}{|sin\overline{x}|}$ *where* $\overline{x} = \frac{\pi x}{2L}$.

The proof is analogical as for lemma 3. The series is convergent also for $t = 0$.

To illustrate how the above principles can be used to build an efficient computational procedure we give in Figure 3.8 an example of an algorithmic primitive for the computation of the kernel $G_{\partial\xi}(x,t)$.

Providing an analysis of error behaviour in the computation of the above mentioned kernels, there is a question how the kernels actually behave, i.e. how many series term are required to obtain a given relative accuracy depending on time and space coordinates in which the kernels are computed.

In Figures 3.9, 3.10 we can see a number of the series terms as a function of the time in the most critical regions for the kernels computation. All computations have been done with relative precision 0.0001 i.e. the fourth decadic digit was required to be valid.

It can be concluded, based on the above numerical study that the described analysis actually provide an efficient and robust means for computing of Green's functions for our problem.

Finally, having a reliable procedure for the respective kernels, we can visualize the kernels. In Figure 3.12 we have $G_T(x,\xi)$ and in Figure 3.11 the kernels $G_{\partial\xi}(x,t)$ and $G_1(x,t)$.

```
double d_green(double x, double t, double a, double b, double l, double eps)
{
10   double series_term;
20   double lambda;
30   double x_over,a_over,maj_term,sum;
40   int k_mono;

50   num_of_ele=0;
60   sum=0.;

70   x_over=M_PI*x/(2.*l);
80   a_over=M_PI*a/(2.*l);

90   lambda=(2.*num_of_ele+1.)*(2.*num_of_ele+1.)*a_over*a_over;
100  maj_term=(2.*num_of_ele+1.)*exp(-(lambda+b)*t);
110  series_term=maj_term*sin((2.*num_of_ele+1.)*x_over);
120  sum=sum+series_term;
130  num_of_ele++;

140  k_mono=(int)((l/(2.*M_PI*a))*sqrt(2./t)-0.5)+1;

150  while(1) {
160       lambda=(2.*num_of_ele+1.)*(2.*num_of_ele+1.)*a_over*a_over;
170       maj_term=(2.*num_of_ele+1.)*exp(-(lambda+b)*t);
180       series_term=maj_term*sin((2.*num_of_ele+1.)*x_over);

190       if(num_of_ele>=k_mono) {
200            if(fabs(series_term) <= PRESNOST_PC){ break;}
210            if(fabs(sum*sin(x_over)/(maj_term*2.)) > 1./eps) {break;}
220            }

230       sum=sum+series_term;
240       num_of_ele++;
250       }
260  sum=M_PI*a*a*sum/(l*l);
270  return(sum);
}
```

Figure 3.8: Algorithmic primitive for the kernel $G_{\partial\xi}(x,t)$. The variable num_of_ele counts series terms, the variable $lambda$ contains the eigenvalues of the Sturm-Liuville problem and the variable maj_term contains the majorization part of the series terms. The constant $PRESNOST\_PC$ represents computer floating point accuracy.

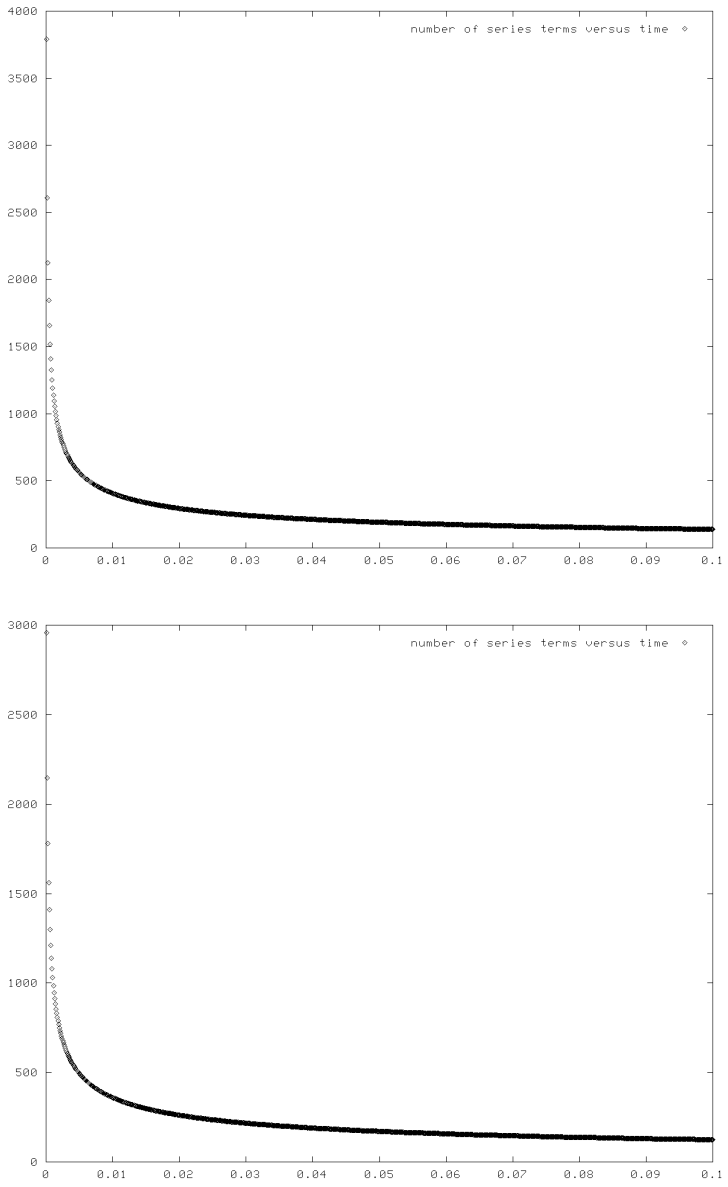


Figure 3.9: The numerical study of the number of the series terms as a function of time for the most critical spatial domain. The upper figure is for the kernel $G_{\partial\xi}(x,t)$ and the lower figure for the kernel $G_1(x,t)$. The relative precision is 0.0001 and the kernels are computed along the line $x = 0.000297$ in space for 1000 equidistant time points. On the x-axis is the time in seconds and y-axis represents the number of terms.

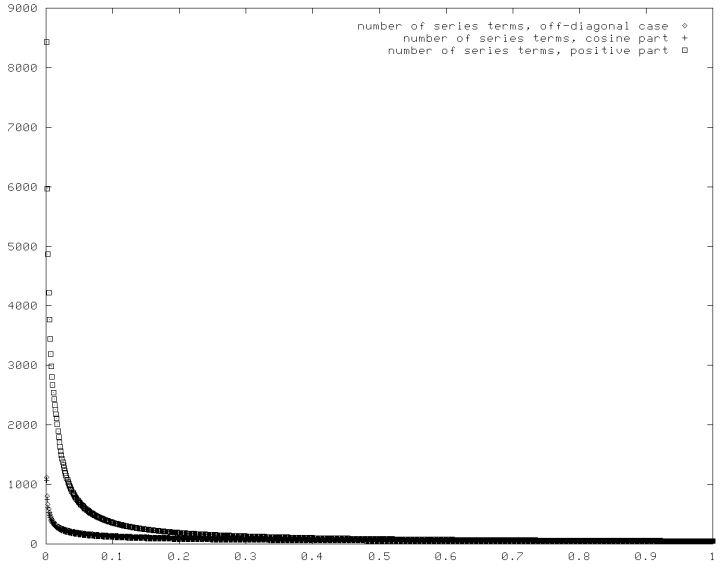


Figure 3.10: The numerical study of the number of the series terms as a function of time for the most critical spatial domain for the kernel $G_T(x,\xi)$. According to the analysis in the text, three different sub-kernels are computed. One for the off-diagonal case when $x \neq \xi$, another for the diagonal case when $x = \xi$. The latter one is split to two sub-kernels, the first being a cosine Fourier series and the second is a series with positive terms. The relative precision is 0.0001 and the kernel is computed along the line $x = 0.000297, \xi = 0.000594$ for off-diagonal case and along the line $x = 0.000297, \xi = 0.000297$ for the diagonal case. On the x-axis is the time in seconds and y-axis represents the number of terms.

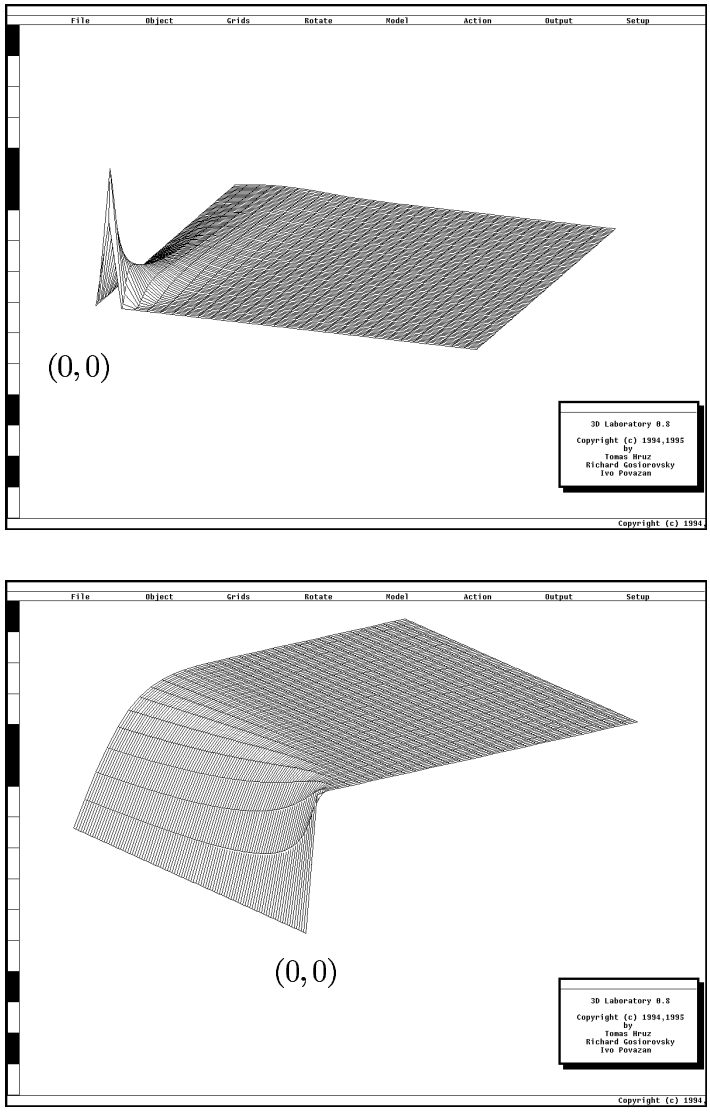


Figure 3.11: In the upper figure is the kernel $G_{\partial\xi}(x,t)$ and in the lower figure the kernel $G_1(x,t)$. In both figures the more dense grid is the time grid. The point $(0,0)$ is labeled by a marker.

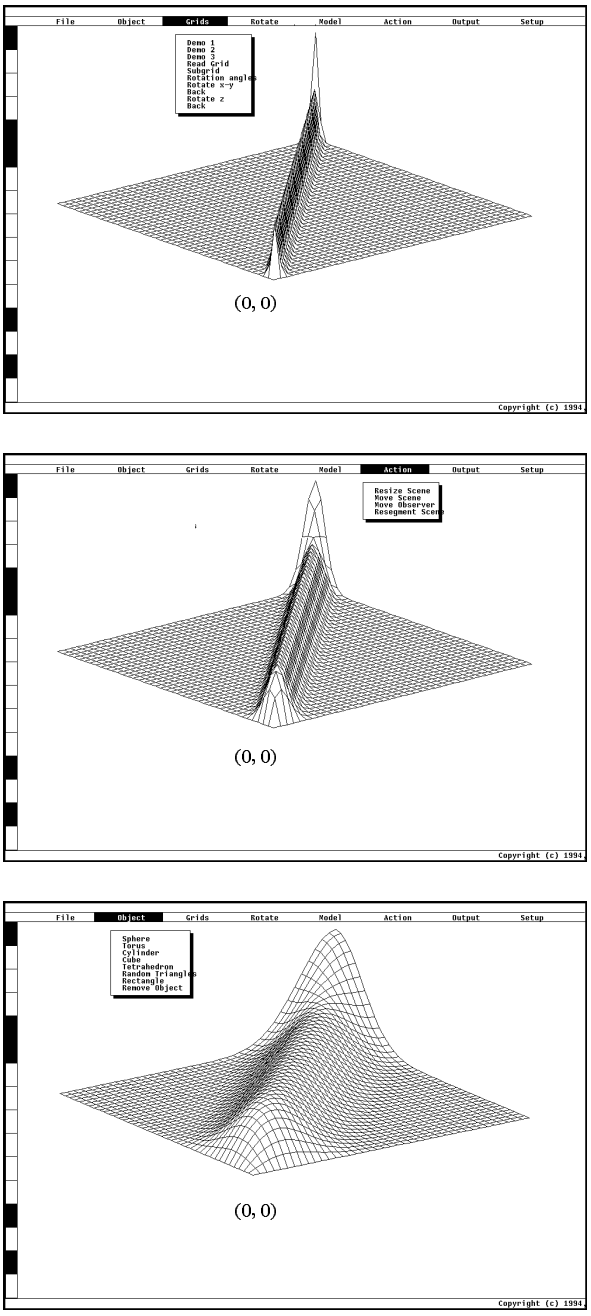


Figure 3.12: Visualization of the kernel $G_T(x, \xi)$. The time is used as a parameter. In the upper figure $T = 0.1$ second, in the middle figure $T = 1$ and in the lower figure $T = 10$. The point $(0, 0)$ is labeled with a marker.

3.2.1.2 Accuracy of the simulation near the heated boundary

The major numerical problems encountered near the boundary during the simulation and inversion are related to the kernel $G_{\partial\xi}(x,t)$. Therefore, let us suppose for the simplicity that in (3.6) the initial condition $Y_0(x)$ is zero. Then the state of the system in a certain time instant t can be written according to equation (3.6) as:

$$Y(x,t) = a^2 \int_{t_0}^{t} G_{\partial\xi}(x, t-\tau)u(\tau)\, d\tau \tag{3.12}$$

A numerical study of the model (3.12) with the parameter values for the actual experimental device (see Figure 3.1) which is illustrated in Figure 3.13 shows that when we use equidistant grids in space and time they are not well adapted. The time grid must be much more dense than that in space in order to obtain sufficient accuracy near the boundary $x = 0$. In Figure 3.13, the results of 4 computations with the same grid in space containing 20 points and 4 different grids in time, one with 20 points, another with 200 points, next with 2000 points and the last with 4000 points, are presented. The results show that the 200 point grid is still not sufficiently dense to get reasonably accurate results.

One possibility how to deal with the above problem (which is actually used in the open loop step-wise control described in section 3.3) is to use non-equidistant B-spline approximation with knot points which are more dense near time zero.

Another possibility is the following. The integral equation (3.12) describes the solution only in the interior ($0 < x < L$) of the spatial domain. Using the Fourier separation of variables technique and deriving from first principles, as is described in detail in appendix A (esp. equation (A.15)), we can obtain the following description of the solution for the whole spatial domain $0 \leq x \leq L$,

$$Y(x,t) = u(t) + \int_{t_0}^{t} G_1(x, t-\tau)[-\frac{du}{dt}(\tau) - bu(\tau)]\, d\tau \tag{3.13}$$

where

$$G_1(x,t) = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{1}{2n+1} \exp\left(-bt - \lambda_n t\right)\varphi_n(x)$$

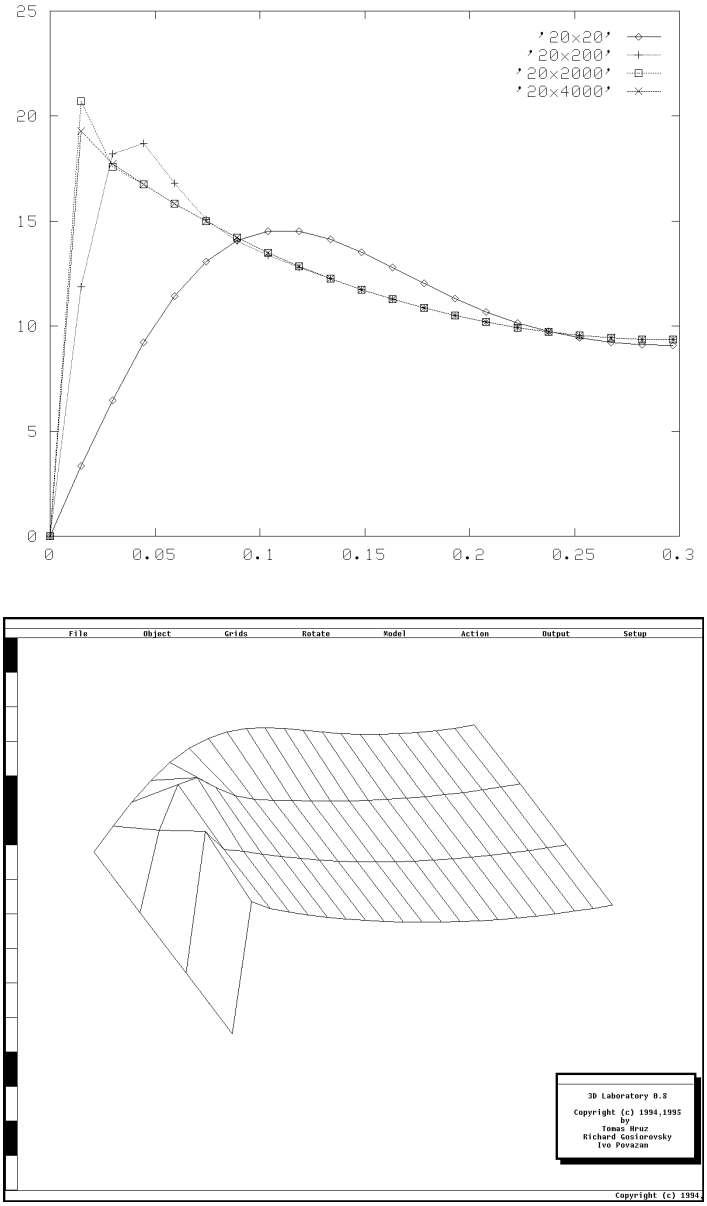


Figure 3.13: A numerical study of the relation between the space and time grids for the equation (3.12). In the upper figure the x-axis represents spatial coordinate in meters and the y-axis the temperature in degrees Celsius. The boundary condition was constant, namely $u(t) = 20$.. The lower figure shows the same four curves in 3-D view. In this figure the most accurate result (the densest grid in time) is the nearest to the observer.

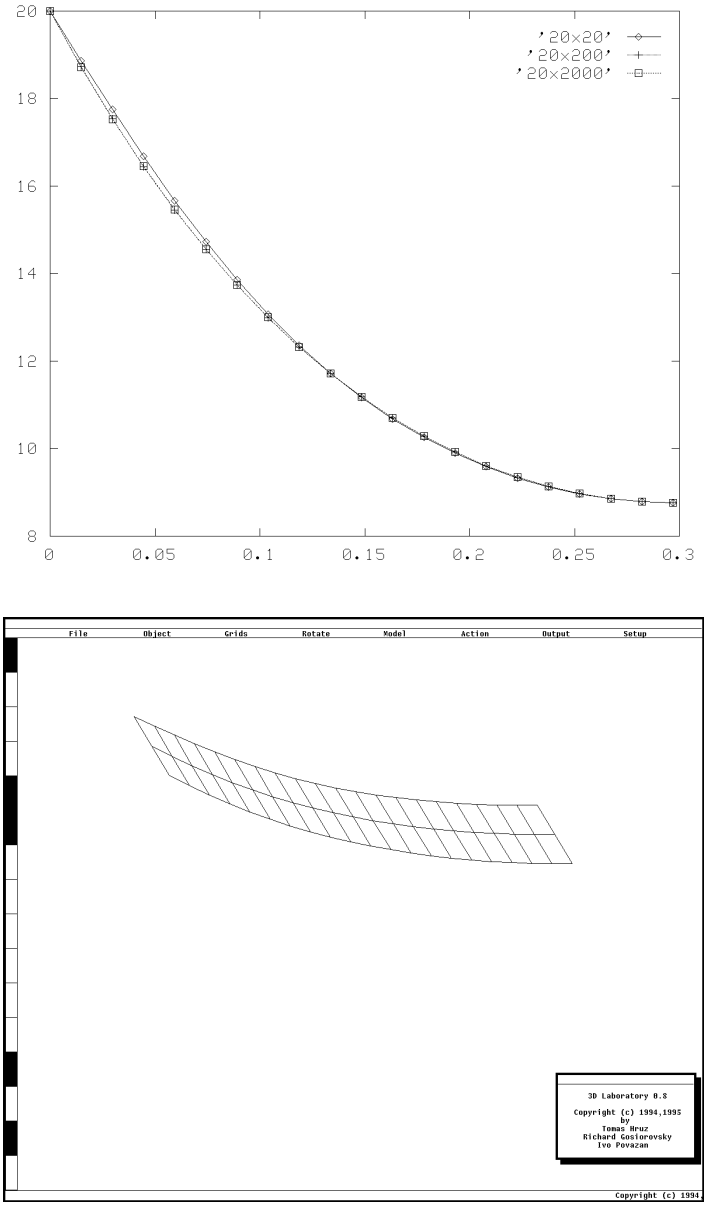


Figure 3.14: A numerical study of the relation between the space and time grids for the equation (3.13). In the upper figure the x-axis represents spatial coordinate in meters and the y-axis the temperature in degrees Celsius. The boundary condition was constant, namely $u(t) = 20.$. The lower figure shows the same three curves in 3-D view. In this figure the most accurate result (the densest grid in time) is the nearest to the observer.

$$= \frac{4}{\pi} \sum_{n=0}^{\infty} \exp\left(-bt - \lambda_n t\right) \frac{\varphi_n(x)}{2n+1} \tag{3.14}$$

The advantage of the kernel $G_1(x,t)$ is its better numerical behaviour. The series for this kernel converges faster and the grids are better adapted. A numerical study in Figure 3.14 clearly shows the difference in the behaviour near the boundary between the kernels $G_{\partial\xi}(x,t)$ and $G_1(x,t)$. For the kernel $G_1(x,t)$ we have computed only 3 solutions with the grids (space grid x time grid) 20x20, 20x200 and 20x2000 because even the 20x20 grid gives sufficiently accurate results.

However, the cost of this improvement is that the intuitive relation between the solution of the integral equation (3.12) and the function $u(t)$ is lost. As can be seen from (3.13), during the inverse problem solution there are two steps, firstly, the integral equation has to be solved and then the first order ordinary differential equation must be solved.

To see the relation between the system description (3.12) and (3.13), the equation (3.13) can be modified by integration by parts to give

$$\begin{aligned} Y(x,t) &= u(t) - G_1(x,0)u(t) + \\ &+ a^2 \int_{t_0}^{t} G_{\partial\xi}(x, t-\tau) u(\tau)\, d\tau \end{aligned} \tag{3.15}$$

This gives the same description inside the spatial domain as in (3.12) because for $0 < x \leq L$, $G_1(x,0) = 1$. But for the point $x = 0$ the value $G_1(x,0)$ is zero as well as the value of the integral kernel in (3.15) leading to the value $u(t)$ at the boundary.

3.2.2 Simulation with Finite Difference Schemes

In this section we describe an approach to the predictor based on a class of well-known numerically stable finite difference schemes. The finite difference method is very well developed therefore in this introductory part we basically follow the standard approach [54, 32]. The basic idea of finite difference method is that we choose a finite set of points (usually as a rectangular grid) in the domain in which we are solving the equation. Then the derivatives of the solution occurring in the partial differential equation are substituted by linear combinations of function values (finite differences) in the points of a given grid. This substitution leads to a finite system of linear equations with the values of a solution in the grid points as unknowns.

Because we would like to extend the prediction also to the more general cases where the thermal diffusivity is a function of temperature we start with a more general setting of the basic parabolic partial differential equation.

$$c\frac{\partial Y}{\partial t} + LY \equiv c(x,t)\frac{\partial Y}{\partial t} - \frac{\partial}{\partial x}\left(p(x,t)\frac{\partial Y}{\partial x}\right) + q(x,t)Y = f \qquad (3.16)$$

in the domain $R = (0,L) \times (0,T)$, with an initial condition
$Y(x,0) = g(x), \qquad 0 < x < 1$

and a boundary conditions (Dirichlet problem)
$Y(0,t) = \gamma^{(0)}(t),$
$Y(L,t) = \gamma^{(L)}(t), \qquad 0 < t < T$

or (Newton problem)
$(I^{(0)}Y)(t) \equiv -p(0,t)\frac{\partial Y}{\partial x}(0,t) + \beta^{(0)}(t)Y(0,t) = \gamma^{(0)}(t),$
$(I^{(L)}Y)(t) \equiv p(L,t)\frac{\partial Y}{\partial x}(L,t) + \beta^{(L)}(t)Y(L,t) = \gamma^{(L)}(t),$
$0 < t < T.$

$c, p, q, f, \beta^{(0)}, \beta^{(L)}, \gamma^{(0)}, \gamma^{(L)}$ and g are known functions, where we suppose for simplicity that they are continuous and the function p has continuous derivative

with respect to the variable x in R. Moreover the functions c and p are positive in R and the functions $\beta^{(0)}$ and $\beta^{(L)}$ are nonnegative in $< 0, T >$.

Let us construct a rectangular grid as an intersection of the lines $x = x_k$ and $t = t_l$, where $x_k = kh$, $k = 0, ..., n$, $h = L/n$, $t_l = l\tau$, $l = 0, ..., r$, $\tau = T/r$ and n, r are integers.

Using so called Marchuk identities ([54] page. 252) we can approximate the parabolic equation (3.16) with the following class of difference schemes given by the parameter α:

$$\begin{aligned}
[\alpha c(x_k, t_l) + (1-\alpha)c(x_k, t_{l-1})]\frac{Y_k^{(l)} - Y_k^{(l-1)}}{\tau} - \\
-\frac{\alpha}{h^2}\{p(x_k - h/2, t_l)Y_{k-1}^{(l)} - [p(x_k - h/2, t_l)]Y_k^{(l)} + \\
+p(x_k + h/2, t_l)Y_{k+1}^{(l)}\} - \frac{1-\alpha}{h^2}\{p(x_k - h/2, t_{l-1})Y_{k-1}^{(l-1)} - \\
-[p(x_k - h/2, t_{l-1}) + p(x_k + h/2, t_{l-1})]Y_k^{(l-1)} + \\
+p(x_k + h/2, t_{l-1})Y_{k+1}^{(l-1)}\} + \alpha q(x_k, t_l)Y_k^{(l)} + \\
+(1-\alpha)q(x_k, t_{l-1})Y_k^{(l-1)} = \alpha f(x_k, t_l) + (1-\alpha)f(x_k, t_{l-1}) \\
k = 1, ..., n-1, l = 1, ..., r.
\end{aligned} \tag{3.17}$$

The initial condition can be written as

$$Y_k^{(0)} = g(x_k), k = 0, ..., n. \tag{3.18}$$

The boundary conditions for the Dirichlet problem are

$$Y_0^{(l)} = \gamma^{(0)}(t_l), Y_n^{(l)} = \gamma^{(L)}(t_l), l = 1, ..., r, \tag{3.19}$$

resp. the boundary conditions for the Newton problem can be rewritten as

$$\frac{1}{2}h[\alpha c(x_0, t_l) + (1-\alpha)c(x_0, t_{l-1})]\frac{Y_0^{(l)} - Y_0^{(l-1)}}{\tau} +$$

$$
\begin{aligned}
&+\alpha[-p(x_0+h/2,t_l)\frac{Y_1^{(l)}-Y_0^{(l)}}{h}+\beta^{(0)}(t_l)Y_0^{(l)}]+\\
+(1-\alpha)[-p(x_0+h/2,t_{l-1})&\frac{Y_1^{(l-1)}-Y_0^{(l-1)}}{h}+\beta^{0}(t_{l-1})Y_0^{(l-1)}]+\\
&+\frac{1}{2}h\alpha q(x_0,t_l)Y_0^{(l)}+\frac{1}{2}h(1-\alpha)q(x_0,t_{l-1})Y_0^{(l-1)}=\\
&=\frac{1}{2}h\alpha f(x_0,t_l)+\frac{1}{2}h(1-\alpha)f(x_0,t_{l-1})+\\
&+\alpha\gamma^{(0)}(t_l)+(1-\alpha)\gamma^{(0)}(t_{l-1}),\\
\frac{1}{2}h[\alpha c(x_n,t_l)+(1-\alpha)c(x_n,t_{l-1})]&\frac{Y_n^{(l)}-Y_n^{(l-1)}}{\tau}+\\
&+\alpha[p(x_n-h/2,t_l)\frac{Y_n^{(l)}-Y_{n-1}^{(l)}}{h}+\beta^{(L)}(t_l)Y_n^{(l)}]+\\
+(1-\alpha)[p(x_n-h/2,t_{l-1})&\frac{Y_n^{(l-1)}-Y_{n-1}^{(l-1)}}{h}+\beta^{(L)}(t_{l-1})Y_n^{(l-1)}]+\\
&+\frac{1}{2}h\alpha q(x_n,t_l)Y_n^{(l)}+\frac{1}{2}h(1-\alpha)q(x_n,t_{l-1})Y_n^{(l-1)}=\\
&=\frac{1}{2}h\alpha f(x_n,t_l)+\frac{1}{2}h(1-\alpha)f(x_n,t_{l-1})+\\
&+\alpha\gamma^{(L)}(t_l)+(1-\alpha)\gamma^{(L)}(t_{l-1}),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad(3.20)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad l=1,...,r.
\end{aligned}
$$

To describe the algorithm to which the above formulas with respective initial and boundary conditions are leading we can use a matrix notation. Let $\mathbf{Y}^{(l)}$ is a vector of the solution with the coordinates $Y_k^{(l)}$. If the equations (3.17) are multiplied by the integration step τ and the equations (3.20) are multiplied by the value $\frac{\tau}{h}$ we obtain

$$
\begin{aligned}
&\mathbf{A}_U^{(l)}\mathbf{Y}^{(l)}=\mathbf{A}_L^{(l)}\mathbf{Y}^{(l-1)}+\frac{\tau}{h^2}[\alpha\mathbf{f}^{(l)}+(1-\alpha)\mathbf{f}^{(l-1)}] \qquad (3.21)\\
&l=1,...,r.
\end{aligned}
$$

where the matrices $\mathbf{A}_U^{(l)}$ and $\mathbf{A}_L^{(l)}$ are defined as follows

$$
\mathbf{A}_U^{(l)}=\alpha\mathbf{C}^{(l)}+(1-\alpha)\mathbf{C}^{(l-1)}+\alpha\frac{\tau}{h^2}\mathbf{P}^{(l)}, \qquad (3.22)
$$

$$\mathbf{A}_L^{(l)} = \alpha\mathbf{C}^{(l)} + (1-\alpha)\mathbf{C}^{(l-1)} - (1-\alpha)\frac{\tau}{h^2}\mathbf{P}^{(l-1)},$$
$$l = 1, ..., r.$$

The matrix $\mathbf{C}^{(l)} = [c_{km}^{(l)}]$ is diagonal, $\mathbf{P}^{(l)} = [p_{km}^{(l)}]$ is a tridiagonal matrix and $\mathbf{f}^{(l)} = [f_k^{(l)}]$ is a vector. The dimension of the above matrices as well as the dimension of the equation (3.21) depends on the boundary conditions. For the case of Dirichlet conditions on the both ends of the domain L we have the dimension $n-1$, for the case of Newton conditions the dimension is $n+1$. Problems with mixed boundary conditions have dimension n as will be our particular case later.

For the case of pure Dirichlet conditions resp. pure Newton conditions we can write

$$\begin{aligned}
k = 1, ..., n-1, \quad & c_{kk}^{(l)} = c(x_k, t_l), \\
& \text{resp. for the Newton case} \\
& c_{00}^{(l)} = \frac{1}{2}c(x_0, t_l), \\
k = 1, ..., n-1, \quad & c_{kk}^{(l)} = c(x_k, t_l), \\
& c_{nn}^{(l)} = \frac{1}{2}c(x_n, t_l),
\end{aligned}$$

$$\begin{aligned}
k = 1, ..., n-1, \quad & p_{kk}^{(l)} = p(x_k - h/2, t_l) + p(x_k + h/2, t_l) + h^2 q(x_k, t_l), \\
k = 1, ..., n-2, \quad & p_{k,k+1}^{(l)} = p_{k+1,k}^{(l)} = -p(x_k + h/2, t_l), \\
& \text{resp. for the Newton case} \\
& p_{00}^{(l)} = p(x_0 + h/2, t_l) + h\beta^{(0)}(t_l) + \frac{1}{2}h^2 q(x_0, t_l), \\
k = 1, ..., n-1, \quad & p_{kk}^{(l)} = p(x_k - h/2, t_l) + p(x_k + h/2, t_l) + h^2 q(x_k, t_l), \\
& p_{nn}^{(l)} = p(x_n - h/2, t_l) + h\beta^{(L)}(t_l) + \frac{1}{2}h^2 q(x_n, t_l), \\
k = 0, ..., n-1, \quad & p_{k,k+1}^{(l)} = p_{k+1,k}^{(l)} = -p(x_k + h/2, t_l),
\end{aligned}$$

$$\begin{aligned}
& f_1^{(l)} = h^2 f(x_1, t_l) + p(x_0 + h/2, t_l)\gamma^{(0)}(t_l), \\
k = 2, ..., n-2, \quad & f_k(l) = h^2 f(x_k, t_l),
\end{aligned}$$

$$f_{n-1}^{(l)} = h^2 f(x_{n-1}, t_l) + p(x_n - h/2, t_l)\gamma^{(L)}(t_l),$$

resp. for the Newton case

$$f_0^{(l)} = \frac{1}{2}h^2 f(x_0, t_l) + h\gamma^{(0)}(t_l),$$

$$k = 1, ..., n-1, \qquad f_k^{(l)} = h^2 f(x_k, t_l),$$

$$f_n^{(l)} = \frac{1}{2}h^2 f(x_n, t_l) + h\gamma^{(L)}(t_l).$$

The matrix $\mathbf{A}_U^{(l)}$ is tridiagonal for any grid (i.e. for any value of the ratio τ/h^2). For $\alpha = 0$ it is a diagonal matrix therefore it is easy to compute the inverse matrix, so in this case the method would be explicit. In all other cases we have implicit methods and for each time discretization point t_l a solution of $n-1$ resp. $n+1$ tridiagonal linear equations must be found. An appropriate method for this task is a standard Gauss elimination method for sparse matrices.

A particular case occurs when $\alpha = \frac{1}{2}$. The method is called *Crank-Nicolson method* and has particularly good features. In this case the discretization error has an order $O(\tau^2 + h^2)$ and therefore it is natural to take $\tau = O(h)$. Let $\tau = \omega_1 h$ where ω_1 is a positive number. The equation (3.21) can be multiplied by $2h^2/\tau$ to make the elements of matrices $\tilde{\mathbf{A}}_U^{(l)}, \tilde{\mathbf{A}}_L^{(l)}$ approximately as big as is the function p. In the original version the elements of these matrices are of order p/h. We obtain

$$\tilde{\mathbf{A}}_U^{(l)}\mathbf{Y}^{(l)} = \tilde{\mathbf{A}}_L^{(l)}\mathbf{Y}^{(l-1)} + \mathbf{f}^{(l)} + \mathbf{f}^{(l-1)}, \tag{3.23}$$

where

$$\tilde{\mathbf{A}}_U^{(l)} = \frac{h}{\omega_1}(\mathbf{C}^{(l)} + \mathbf{C}^{(l-1)}) + \mathbf{P}^{(l)}, \tag{3.24}$$

$$\tilde{\mathbf{A}}_L^{(l)} = \frac{h}{\omega_1}(\mathbf{C}^{(l)} + \mathbf{C}^{(l-1)}) - \mathbf{P}^{(l-1)}$$

The advantage of the above scaling with the factor $2h^2/\tau$ depends on the size of the coefficient p. If p is very small it is better not to use this scaling because h is usually small so p/h will be of better size with respect to the computer precision.

The above general scheme can be specialized for the particular problem (3.1) which is considered in this work. In this case $c = 1$. To incorporate also nonlinearity caused by dependence of a, b on temperature we suppose that the functions

$p(x,t) = a(Y(x,t))$, $q(x,t) = b(Y(x,t))$ are not constant. The nonlinearity can be treated by "freezing" the value of the parameters during one integration period and using the same scheme as for the linear case. Note that because of mixed boundary conditions the dimension of the finite difference scheme is n.

We obtain the following algorithm in Figure 3.15.

$$p(x,t) = a(Y(x,t)),\ q(x,t) = b(Y(x,t)),\ \omega = \frac{\tau}{2h^2} \tag{3.25}$$

$$\begin{aligned} \mathbf{A}_U^{(l)}\mathbf{Y}^{(l)} &= \mathbf{A}_L^{(l)}\mathbf{Y}^{(l-1)} + \omega(\mathbf{f}^{(l)} + \mathbf{f}^{(l-1)}) \\ \mathbf{A}_U^{(l)} &= \mathbf{C} + \omega\mathbf{P}^{(l)}, \\ \mathbf{A}_L^{(l)} &= \mathbf{C} - \omega\mathbf{P}^{(l-1)}, \end{aligned} \tag{3.26}$$

$$\begin{aligned} k = 1, ..., n-1, \qquad & c_{kk} = 1 \\ & c_{nn} = \frac{1}{2} \end{aligned} \tag{3.27}$$

$$\begin{aligned} k = 1, ..., n-1, \qquad & p_{kk}^l = p(x_k - \frac{h}{2}, t_l) + p(x_k + \frac{h}{2}, t_l) + h^2 q(x_k, t_l) \\ & p_{k,k+1}^l = p_{k+1,k}^l = -p(x_k + \frac{h}{2}, t_l) \\ & p_{nn} = p(x_k - \frac{h}{2}, t_l) + \frac{1}{2}h^2 q(x_k, t_l) \end{aligned} \tag{3.28}$$

$$\begin{aligned} & f_1^l = p(x_0 + \frac{h}{2}, t_l)u(t_l) \\ k = 2, ..., n, \qquad & f_k^l = 0 \end{aligned} \tag{3.29}$$

Figure 3.15: A finite difference scheme for the problem (3.1)

.

3.2.3 Comparison of Simulations with Experiments

Having fully developed both simulation methods (integral representation with Green's functions, see section 3.2.1 resp. finite difference schemes, see section 3.2.2) we can compare them with experimental data measured on the real laboratory system in Figure 3.1. In this section we mainly compare the methods against the experimental data and the comparison between the both methods we postpone to the concluding section when more information will be at the disposal.

This is also a moment to discuss identification of unknown values of the model parameters, namely the thermal diffusivity coefficient a^2 and the coefficient b. When we plug the results of identification procedure to the model and compare the results with a real behaviour of the experimental system the quality of the parameters is clearly seen. We obtain also some knowledge about the directions of further model improvements.

During the experiments, if there was a need to follow certain function on the boundary or to stabilize the boundary temperature on certain value a PID regulator has been used with pre-tuned parameters. However, the quality of this control is not relevant for the simulation aspect of this work, because the boundary data has been measured and they are exactly used during the system simulations as a boundary condition, so that the simulator can follow real evolution of the system. The PID regulator merely provided a certain guide-lines of the system development so that it was not chaotic. Also it provided a possibility to measure steady states of the system relatively easily.

The system states during the experiments has been stored, the boundary condition being an integral part of the data. From this moment on, we can forget about any controller which was responsible for obtaining the data. What we have now are dynamic evolutions of the system and our aim is to tune the simulation methods so that we obtain good fit between the experimental data and simulated system behaviour **inside the spatial domain**.

First off all, we describe the experimental data and some parameters and notations which will be constant during this section. From large set of experimental data we have chosen the following three experiments on which we illustrate the

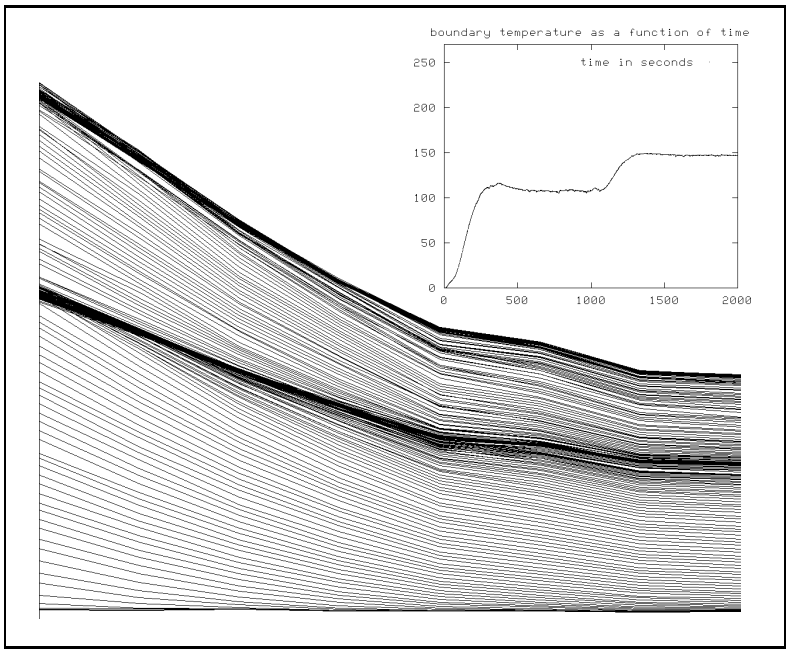


Figure 3.16: The monotonic dynamic process. For the axis description see Figure 3.2.

phenomena of further interest.

1. **Quasi-static process.** The experiment starts from a stable state on a high temperature. Then the system is brought through quasi-static process to the stable state in the lower part of the temperature range. With the term quasi-static we mean that the decrease of the boundary temperature has been so slow that we can suppose that the system is going through a continuous set of steady-states. The experimental data are in Figure 3.5 together with a development of the boundary temperature.

2. **Monotonic dynamic process.** The system evolution starts in a steady state with a low temperature, then the boundary is approximately linearly brought to the higher temperature where the system is stabilized. The former process occurs twice during the experiment. The experimental data are in Figure 3.16.

3. **General dynamic process.** The system is brought through randomly cho-

sen transient and steady states. The experimental data are in Figure 3.17.

The above three sets of data are further referenced by their respective numbers. During this section some parameters are supposed to be constant and there are also another assumptions valid for the whole section, namely:

- The discretization of spatial domain for the simulation has dimension 20. This density (denoted by N) provides a reasonable accuracy compared with the experiments where we have 8 spatial data in equidistant intervals. The length of the spatial domain (the length of the heated bar in the experimental device) is 0.297m.

- The maximal temperature range of experiments is 0-300 degrees Celsius given by the construction of the device.

- As a measure of the fit of a solution $Y(x, t_i) = Y_i(x)$ in a time instant t_i we use the discrete L_2 norm defined as

$$|| D_i(x) ||_{L_2,D} = \sqrt{\frac{L}{N} \sum_{k=0}^{N} D_i^2(x_k)} \tag{3.30}$$

  where $D_i^2(x_k)$ is interpreted here as a deviation between the experimental and simulation data.

- The number of time steps is denoted by S and the time domain by T. The overall fit of a solution on the whole integration domain ($< 0, L > \times < 0, T >$) is described by a *residuum* which is defined as follows

$$Res = \sqrt{\frac{T}{S} \sum_{k=0}^{S} (|| D_k(x) ||_{L_2,D})^2}. \tag{3.31}$$

- The experimental data are stored with a time step 1 second. This is the precision with which we always provide the boundary condition for the simulations independently of the step length of simulation as well as step length used for residuum computations. The time domain of the typical experiments ranges from few thousand seconds up to 8000 seconds for the quasi-static process.

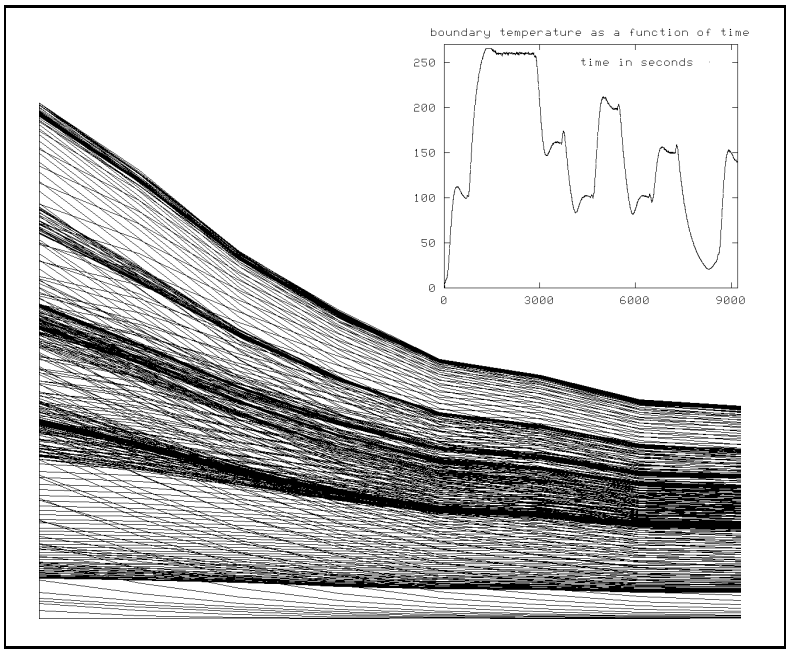


Figure 3.17: The general dynamic process. For the axis description see Figure 3.2.

- The noise in experimental data has a L_2 discrete norm (a standard deviation along the spatial domain) at low temperatures about value 0.1 and at high temperature about 0.5 . Therefore we cannot expect the residuum to be less than about $0.3\sqrt{T}$ over the whole integration interval T.
- In all figures where the states (or differences of states) are shown the x axis corresponds always to the spatial domain with length $L = 0.297m$ and the y axis represents a temperature in degrees Celsius. The temperature range maximal and minimal values are written on the lower and upper part of the axis. The scales are always linear.
- If it is not explicitly stated differently, all simulations are made with the values of coefficients identified in the section 3.1, namely $a = 0.0095, b = 0.0020$.

First of all, we would like to know the overall performance (precision) of the simulation methods. Therefore both methods are run with the most complex

data 3. The results are in Figure 3.18, where two figures are placed, each for one method. On each of them there are two curves drawn in the same scale, one represents the boundary temperature and the other one the L_2 norm of the difference between the simulation and real data (simulation error).

The results are good with the peak error under 2% and most of the time under 1%. However, according to the level of noise we can think about the value of residuum $0.3\sqrt{T} = 26.83$ and the residue for both methods are in order larger. In the next Figure 3.19 the same simulations are shown but different scales are used for temperature and simulation error to be able to distinguish in which situations the L_2 norm of the simulation error grows.

A basic observation which can be made in Figure 3.19 is that the error increases rapidly during the transient states and then drops back during the steady states of the system. This fact is even more apparent in Figure 3.20 for the data 2 which has been chosen to isolate this effect. In this figure the simulation error is also shown with a different scale for the error as for the boundary temperature.

The next question about which we can obtain more information from the comparison between real data and simulations is the computational complexity of both methods. We use the most complex data set 3 for this purpose. First of all a relation between step length and residuum is computed for the finite difference method. Then we choose the same step length of the Green's function method and find a relation between the integration sub-step inside this basic step and a residuum.

The results of the above procedure are in Figure 3.21, where on the first graph is a relation between the step length and the residuum for the finite difference method. The method stays at the minimal value of the residuum up to the step length 40sec. Therefore we choose the same step for the Green's functions as an integration step. Then the number of discretization steps inside this basic interval is measured. The relation between the number of discretization steps and the residuum for the basic step length 40sec for the Green's function method is in the lower graph in Figure 3.21.

It must be stressed in favor of the Green's function scheme that the above comparisons are made with equidistant grid for the integration in the formula (3.5).

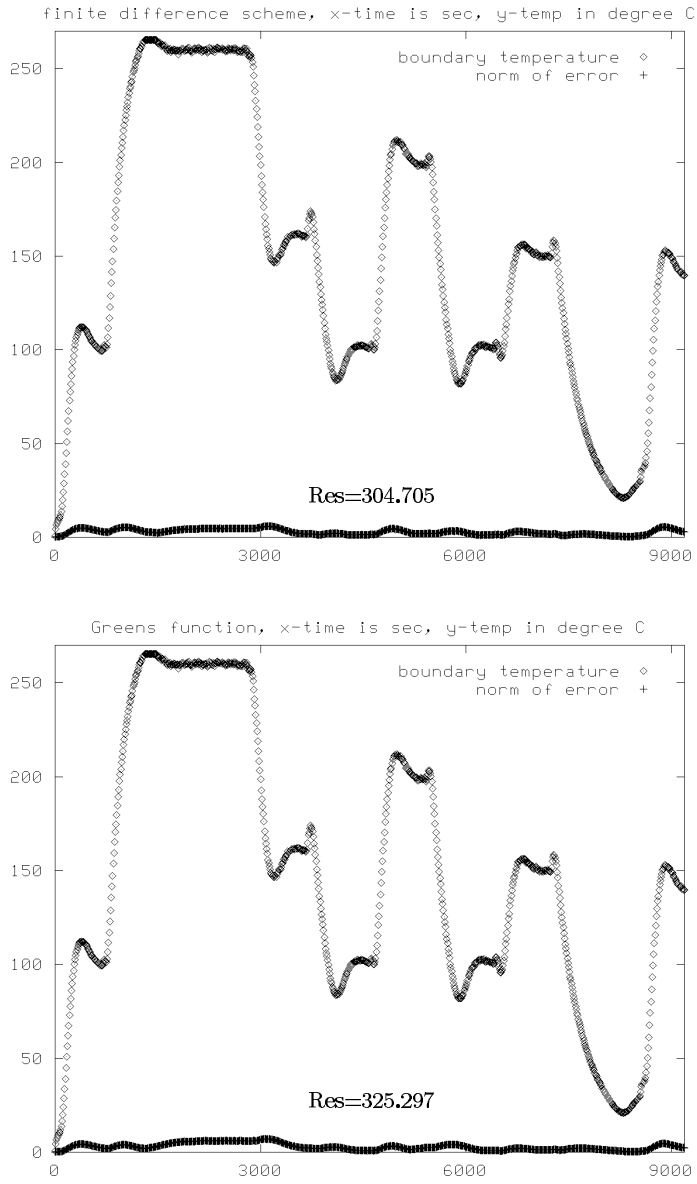


Figure 3.18: Simulations versus experimental data accuracy.

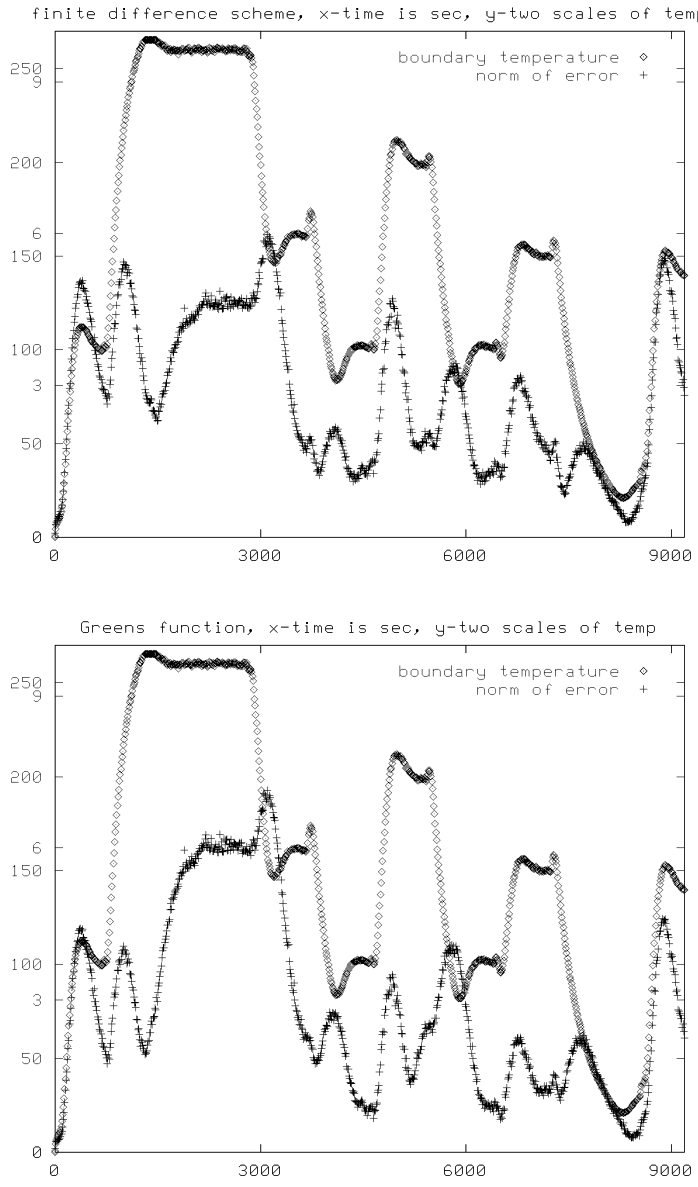


Figure 3.19: Simulations versus experimental data accuracy. More detailed study.

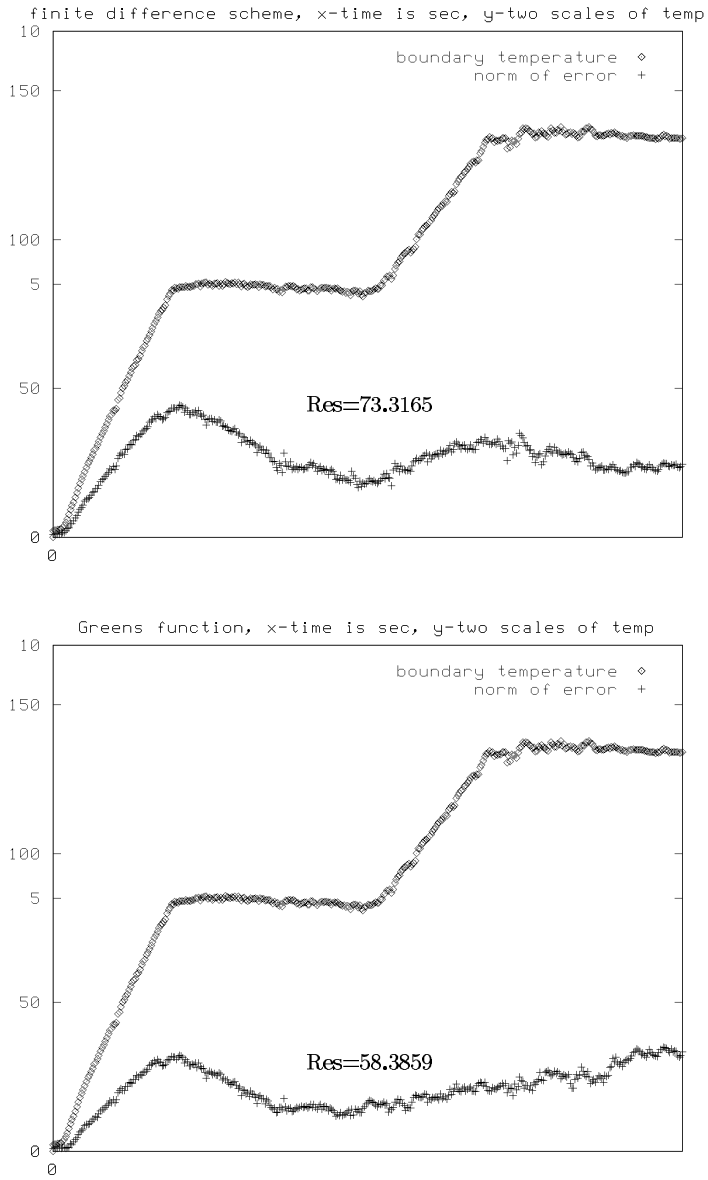


Figure 3.20: Simulations versus experimental data accuracy. A more detailed study of an approximately monotonic process. On the x-axis is a time scale from zero to 2000 seconds. For explanation see the text.

As the analysis in section 3.2.1.2 has shown the efficiency can be improved by choosing non-equidistant grid with higher density near the zero in space. Indeed, such option has been used in the open-loop scheme described in section 3.3, where by choosing more dense description near the zero in space reasonable accuracy has been achieved with number of discretization steps of about 10.

However, even reducing the density of grid for Green's function scheme, it still seems to be considerably slower than finite differences because the matrices are dense compared with tridiagonal matrices in finite differences. So, the grid density 3 points for Green's function gives approximately the same complexity as 1 step of finite differences. Therefore, if 10 points grid is required for Green's function integration we obtain about 3 times slower simulation.

The last set of questions is related to the identification of the thermal diffusivity coefficient a^2 resp. the heat transfer coefficient h. The experiment on which we consider the identification questions is the one in Figure 3.5. We have already used the data set in section 3.1 to identify the value of parameter a. The choice of experiment already suggests that the focus of this part will be on the thermal diffusivity coefficient. Indeed, there are more reasons why concentrate on a^2. The coefficient a^2 is a thermo-physical material constant therefore it can be usable in different circumstances where the same material is involved. This is not the case of coefficient b containing the heat transfer coefficient h (also called the film coefficient). The latter one is in a very complex way dependent on conduction (also convection in the case of fluid) state of surrounding material which makes the coefficient in fact totally dependent on particular experimental or technological devise. As the above simulation results have shown the value which we have identified in section 3.1 gives reasonable accuracy. Therefore in the following we satisfy ourselves with the constant value 0.0020 identified earlier.

As was already noted at the end of section 3.1 it is a well known fact that the coefficient a^2 is thermally dependent. This dependency makes the model nonlinear therefore in the rest of this section only finite differences will be considered for simulation.

First of all, the shape of the simulation error can be visualized (see Figure 3.22) providing a possibility to identify some pattern in the data.

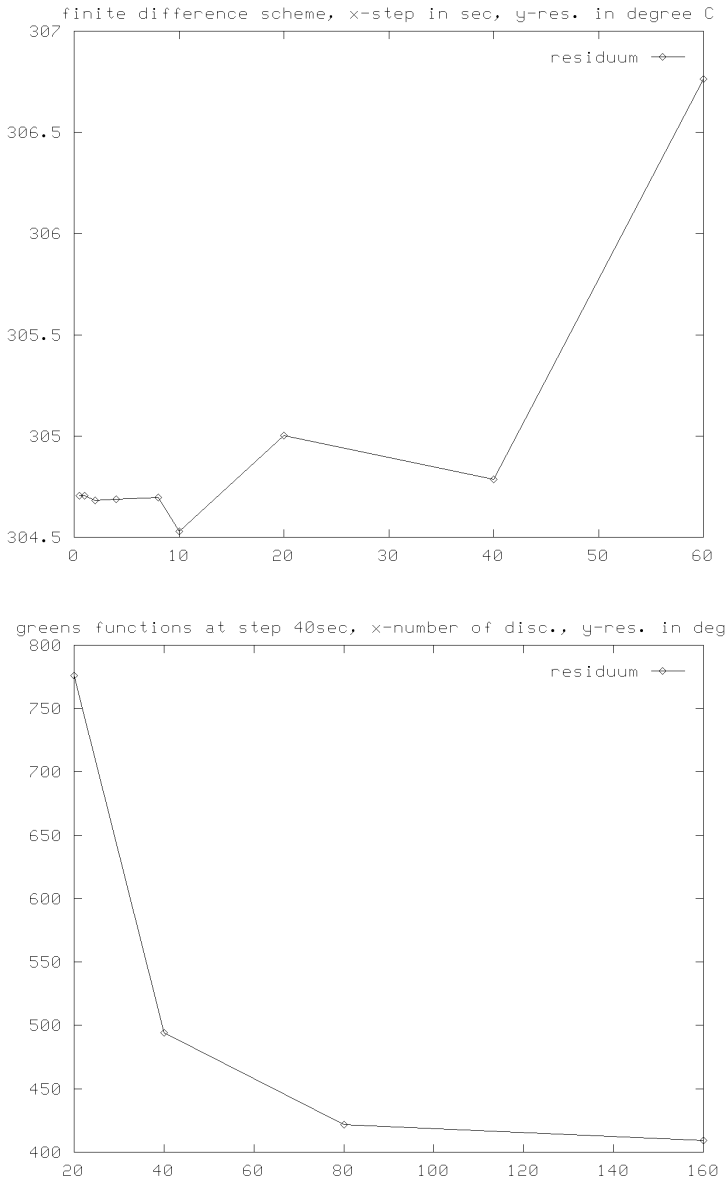
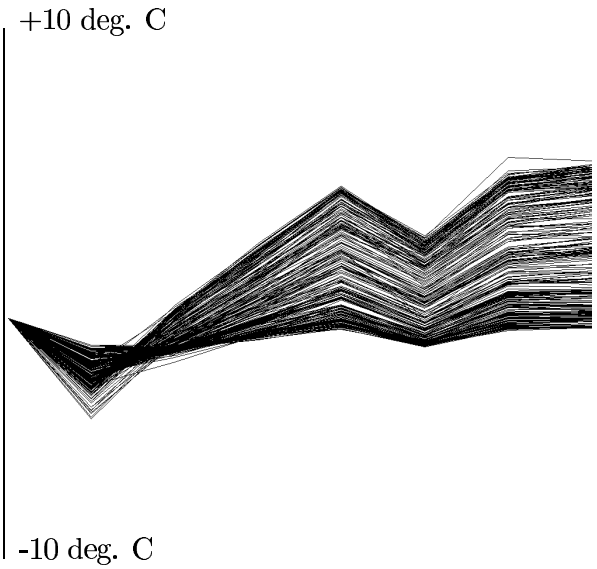


Figure 3.21: Computation speed at comparable values of residuum.



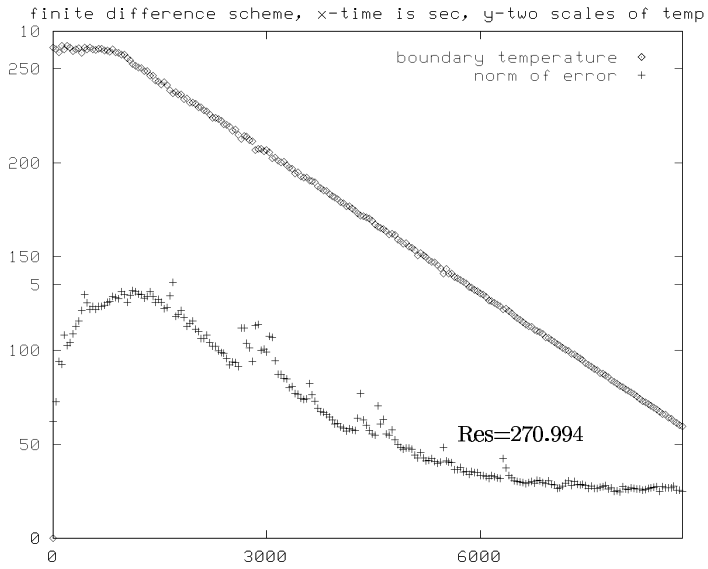


Figure 3.22: Simulations versus experimental data accuracy. A more detailed study of a quasi-static process.

Indeed, there is an observable pattern in the data which can not be caused by a noise. This is supported also by the value of residuum which is 270 but concerning the level of noise and the integration interval we arrive at a value approximately 26. Therefore, the problem is basically the shape of the curve not the noise at this stage.

A look back to Figure 3.6 can lead to considering as a first possibility that the nonlinearity might be of first order, so that an approximation of the heat diffusivity with a linear function of temperature should improve the value of residuum. But, it does not sufficiently, leading to a conclusion that the nonlinearity is of higher order in a significant sense.

In Figure 3.23 four possible curves of the relation between a and temperature are suggested and a residuum is computed for all of them. The curve 1 is a constant lying in the middle of the parameter range obtained before in section 3.1, curve 2 is a linear approximation and the curves 3,4 have been obtained by an exhaustive search in a 1 dimensional parametrized space of functions with the same starting and ending point.

The last set of experiments which we did used Levenberg-Marquart algorithm with Tichonov regularization in a Minpack implementation. These experiments have revealed several facts.

- There is a systematic error on the thermocouple 1 resp. 4 with a correction -8 resp +6 degree's Celsius at the maximal temperature.
- The measurements are not convex, therefore a convex hull filter can be used.
- There is a strong coupling between the parameters a, b.
- The structural aspect must be emphasized i.e. the system model must be updated from a structural point of view to obtain better results.

Using any of the above filters on data was accompanied by a significant decrease of the residuum value. Therefore we conclude that the accuracy of measurement must be improved in order to obtain more precise values of a, b resp. the experiment should be restructured to minimize the influence of the film coefficient.

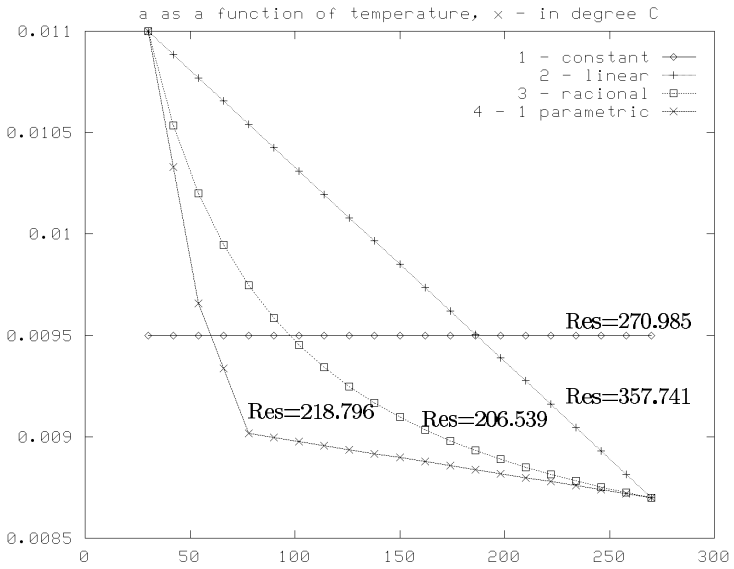


Figure 3.23: Simulations versus experimental data accuracy. A more detailed study of a quasi-static process.

We stop here with a note that an identification of temperature dependence of thermal diffusivity as an severely ill-posed problem requires special methods to be successfully done particularly in the sequential setting. They are a subject of current intensive research in inverse engineering field as can be seen in the recent book [56].

3.3 Open Loop Step-Wise Control

In the following section we describe briefly a method developed by the author and others [40, 41] for open-loop step-wise optimal boundary control of thermal system. The method uses an integral representation, as described in detail in sections 3.2.1, 3.2.1.2 and in appendix A, which is augmented with a splines discretization. We focus here mainly on a description of ill-posed problem inversion and overall algorithm description. Other details can be found elsewhere [40, 41].

The main reason for the numerical difficulties in the boundary control of thermal process inside a heated solid material stems from the ill-possednes of the model inversion which always is, in one way or other, embedded within the control algorithm [55].

We propose a control algorithm which clearly identifies the ill-posed part and splits the system into a pair of subsystems - a subsystem with measurable outputs (surface - boundary temperature of the heated object) which is easy to control by a feedback and a subsequent subsystem which is driven by the preceding subsystem and whose distributed state is inaccessible to direct measurement (inside temperature of heated object). Control of this inside temperature is achieved not by a feedback but by a maintaining of pre-calculated temperature time profile at the boundary of the object. The calculation of this boundary temperature time development is exactly the ill-posed part of the controller as was noted for example in [20].

The method uses a predictive control system working with the measurable output of the thermal system and tracking a specified, optimally pre-calculated reference signal for boundary control of the system in order to obtain a required spatial temperature profile in the heated object at a selected time instants t_v. For a given spatial temperature profile, the reference signal for the system boundary control is obtained by inverting a distributed parameter model, which describes the dynamics of the unmeasurable temperature distribution in the second subsystem of the given thermal system. In the method the inverse problem is converted to some *regularization* problem and is solved by a *stepwise* technique. This technique seems to be suitable for on-line control of thermal systems under a condition

of stochastic disturbances acting on the controlled systems. The dynamics of the first subsystem is modelled by continuous-time convolutional integrals with finite-support kernels. The input and output signals of the subsystem are considered to be polynomial splines. The B-splines are taken as base functions of these splines. The control synthesis is based on minimization of an integral continuous-time quadratic loss function, after the spline approximation is transformed to a simple matrix quadratic form. To minimize this form a quadratic programming is employed. The allowed control input signal is then defined by a set of suitably selected linear equality and inequality constraints which act on the vector of the polynomial coefficients of this signal.

In the following parts of this section we briefly mention the step-wise method, inversion task and the overall algorithm. At the end, the experimental results justifying the method are described. We conclude with some notes leading to the next section which concerns a new close loop predictive scheme for the thermal system.

3.3.1 Step-wise method of optimal boundary control

The laboratory device on which the method is verified consists in a boundary heated metal bar which is cooled by heat tranfer to the surrounding air (for more detailed description of the device see Figure 3.1 in section 3).

Then the behaviour of the unmeasured temperature field of the metal bar $Y(x,t)$ at the time instant t and the position x of the bar is described by the parabolic partial differential equation (3.1).

The heating of the bar is controlled through the boundary temperature $u^w(t) = Y(0,t)$ and the task is to find such function $u^w(t)$ - boundary heating of the bar - which ensures the attainment of the required spatial distribution of the bar temperature $Y(x,t)$ at a specified time instant t_v. In this situation we can use the integral representation (see section 3.2.1) which have the following form:

$$Y(x,t_v) = a^2 \int_{t_0}^{t_v} \frac{\partial}{\partial \xi} G(x,\xi,t_v-\tau) \mid_{\xi=0} \; u^w(\tau)\, d\tau +$$

$$+\int_0^L G(x,\xi,t_v-t_0)\ Y_0(\xi)\ d\xi \tag{3.32}$$

where $Y_0(x) = Y(x,t_0)$ is given initial condition and G is the Green's function. The second part of equation (3.32) is known for known initial conditions. Let us denote it as $Y_c(x,t_v)$ and define the modified state $Y_m(x,t_v)$ as:

$$Y_m(x,t_v) = Y(x,t_v) - Y_c(x,t_v)$$

then

$$Y_m(x,t_v) = \tag{3.33}$$
$$= a^2 \int_{t_0}^{t_v} \frac{\partial}{\partial\xi} G(x,\xi,t_v-\tau)\ |_{\xi=0}\ u^w(\tau)d\tau$$

This equation can be written in an operator form:

$$Y_m = Au^w \qquad Y_m \in S \quad ,\ u^w \in D \subseteq Z \tag{3.34}$$

where A is the linear integral operator of relation (3.33), Z and S are Hilbert spaces, D is a closed convex set, build by a priori limitations of the control task. The relation (3.34) represents an Fredholm integral equation of the first type and the solution of this equation fulfils the definition of an *ill-posed problem* in the Hadamard's sense. Therefore it is necessary to use some *regularization* method, which will give satisfactory results. We employ the method of Tikhonov [50], where the task of solving the equation (3.34) is replaced by the task of the minimization of the following *smoothing functional* $M_\alpha[y]$:

$$M_\alpha[y] = \| A_h y - Y_{m\delta} \|^2 + \alpha \| u^w \|^2$$

where $\alpha > 0$ is the regularization parameter.

A_h is an operator which approximates the operator A with defined error h, that means

$$\| A_h - A \| \ \leq\ h, \tag{3.35}$$

$Y_{m\delta}$ is the left hand side of (3.34), which is specified by the error δ, i.e.

$$|| Y_m - Y_{m\delta} || \leq \delta, \tag{3.36}$$

The so-called *generalized residuum* is defined as :

$$\begin{aligned} \rho(\alpha) =& || A_h \, u_\alpha^w - Y_{m\delta} ||^2 - \\ -(\delta^2 + h \, || \, u_\alpha^w \, ||^2) &- (\mu \, (Y_{m\delta}, A_h))^2 \end{aligned} \tag{3.37}$$

where

$$\mu \, (Y_{m\delta}, A_h) = \inf_{y \in Y} \; || \, A_h y - Y_{m\delta} \, ||$$

is the *degree of inconsistency.*

The regularization parameter α of the smoothing functional is chosen by *generalized principle of residuum*, which is the following. If the condition:

$$|| \, Y_{m\delta} \, ||^2 > \delta^2 + (\mu(Y_{m\delta}, A_h))^2 \tag{3.38}$$

is not fulfilled,the approximate solution of the equation (3.34) is $y = 0$. If the condition (3.38) is fulfilled, then the generalized residuum (3.37) has a positive root α^* and the solution of equation (3.34) is a minimum $u_{\alpha^*}^w$ of the smoothing functional (3.35).

In [50] various properties of the generalized residuum has been proved. An important fact for our algorithm is that in the presence of constraints the generalized residuum is not a differentiable function of α and therefore the numerical method used can only employ the values of function $\rho(\alpha)$ in order to localize the root. However, the generalized residuum as a function of α is continuous, monotonic and therefore has only one zero.

As regards to the accuracy of the proposed procedure, the Tikhonov theory do not specify the precision to which a computation of the solution of equation $\rho(\alpha) = 0$ has to be carried out. However, in an experimental situation with noisy data few decadic places were enough.

3.3.2 Overall algorithm

In the preceding section the inverse task for the distributed model (3.1) was transformed to the problem of solving the operator equation (3.34) for a given left hand side. The solution $u^w_{\alpha^*} = u_r(t)$ of this equation forms the reference signal for the optimal boundary control of the heated bar. The iterative regularization method used for solving equation (3.34) is valid only for in advance given and constant integral bounds t_0, t_v. This fact is necessary to take into account in the designing of the generator of the reference signal $u_r(t)$ for on-line boundary control of the thermal system. One way is to base the generator structure on *stepwise triggering* the inversion task in equidistantly located discrete time instants t_v. The distance between the time instants determines the bounds of integrals in the numerical solution of the equation (3.34) and in next explanation we will call this distance the *inversion horizon* H_i. From a practical point of view the length of the horizon H_i depends on several factors, namely,

- Technological needs for the heating process and the goal of the heating.
- Dynamical properties of the thermal system.
- Time behaviour of disturbances acting on the measurable system output.

In the process of the stepwise triggering of the inversion task with the time period H_i it is necessary to know at the particular starting time instant a *true* profile of the unmeasurable temperature distribution $Y(x,t)$ in the heated bar. The true profile $Y(x,t)$, which is really reached at the end of a preceding period, creates the initial condition for the inversion in a subsequent period. Because the temperature profile $Y(x,t)$ is not measured, its true time development can be only *simulated* using a responce of the model (3.1) due to the actually measured system output signal $u^w(t)$. For numerical calculation of the true response $Y(x,t)$, it is advantageous to utilize again the operator form (3.34) of the model. The length of the time interval during which the integration in (3.34) with the real signal $u^w(t)$ is performed we will call a *simulation horizon* H_s. For numerical reasons it is suitable to choose $H_s = H_i/ni$, where ni is a given positive integer.

The starting point for the numerical solution of the simulation and the inversion tasks consists in a suitable discretization of the basic relation (3.34) and its transformation to a matrix form. The resulting matrix form oriented to the simulation we will call a *simulation* model and the matrix form aimed at the inversion we will call an *inversion* model. Based on the above models, the generator of the reference signal $u_r(t)$ is constructed. The required profiles $Y_z(x)$ enter the generator with time period H_i and the real measured signal $u^w(t)$ enters the generator with period H_s.

3.3.3 Experimental verification

The configuration of an experimental device on which the method has been verified is shown in Figure 3.1. It consists of a copper metal bar which is heated at one boundary and insulated at the other one. There are 8 thermocouples installed on the bar which can be used for the verification of the proposed method.

Equation (3.1) models the experimental device with a reasonable accuracy if the thermal diffusivity and heat transfer coeficients are properly identified.

Let us recall shortly from the section 3.1. We have used an off-line identification method, based on the fact that after reaching steady state of the system on some higher temperature we can switch the boundary condition at the heater side to the insulated-end Neumann type condition by simply switching off the heater. The system will relatively soon reach a state when the $\frac{\partial}{\partial x}Y(x,t) \leq \epsilon_1$ for all $0 \leq x \leq L$ i.e. the temperature along the bar is nearly constant. ϵ_1 is an apriori chosen small constant. In our case the system has been heated to the steady state with boundary temperature 260 degrees Celsius. Subsequently, after switching off the heater the system reached the state where ϵ_1 was under the level of the noise at about 60 degrees. From this moment the behaviour of the system is described by the simple first order differential equation $\frac{d}{dt}Y + bY = 0$. This equation is solved analytically thus providing a possibility for simple identification of the heat transfer coefficient. Once we know the heat transfer coefficient it is easy to compute the thermal diffusivity coefficient by driving the system to steady state and solving the boundary problem for the first order ordinary

differential equation. The resulting values are $b = 0.002$, $a = 0.0906$.

The experimental results are presented on the two sets of experiments. Both sets have the same structure but use the different values of the parameters a, b to illustrate the sensitivity of the proposed method to those parameter values.

Each set of the experiments is organized as follows: the experiment starts in a certain state of a system, then we submit subsequently in a step-wise manner 5 different states which the system must reach in an apriori given time horizon. At the end of each time interval we compare the prescribed goal state with the state which the system has really reached and the internal simulation state which is maintained as a starting state for further steps of the step-wise method, as was described in detail in [40]. The actual measured state of the system is shown in 15 points but only every second one is obtained by direct measurement. The points between are interpolated by the use of splines.

The first set of experiments is given in Figures 3.24, 3.25 and the second in Figures 3.26, 3.27. In both sets, Figure **A** shows the boundary temperature. Figure **B** contains the heater controls and the remaining Figures **t0-t5** show the states at the subsequent time steps.

3.3.4 Notes

The experimental results support the developed step-wise method. However, during the verification process we have also identified certain drawbacks which are solved by the new method described in the next section.

Some difficulties can be overcomed by carefully choosing the spline representation base functions. Others are more inherent to the Green's function approach and could be overcomed only by choosing a different approach. For example, when the equation is of a more complicated nature the explicit form of the Green's function could be not at hand. Also, during the forward problem solving the simulation with the integral kernels is rather computationaly complex. Another problem is the reduced flexibility of the method due to the fixed time integration horizon, which asks for recalculation of the matrices each time the basic time step t_v has to be changed.

One promising direction is the use of the well known numericaly stable difference schemes, or finite elements, for the forward modelling. However, in this case the inverse problem formulation as well as the numerical behaviour are more complex as is described in the next section.

There is also another interesting aspect of the experiments mentioned above which is related to the identification of the system parameters, namely, that the proposed method seems to be more robust with respect to the parameters a, b as is the corresponding forward simulation alone. We suggest that this increased robustness is due to the ill-posed problem solution embedded within the method. However, this aspect would need more experiments to carry out as well as further theoretical analysis to decide whether it could provide a promising direction of further research.

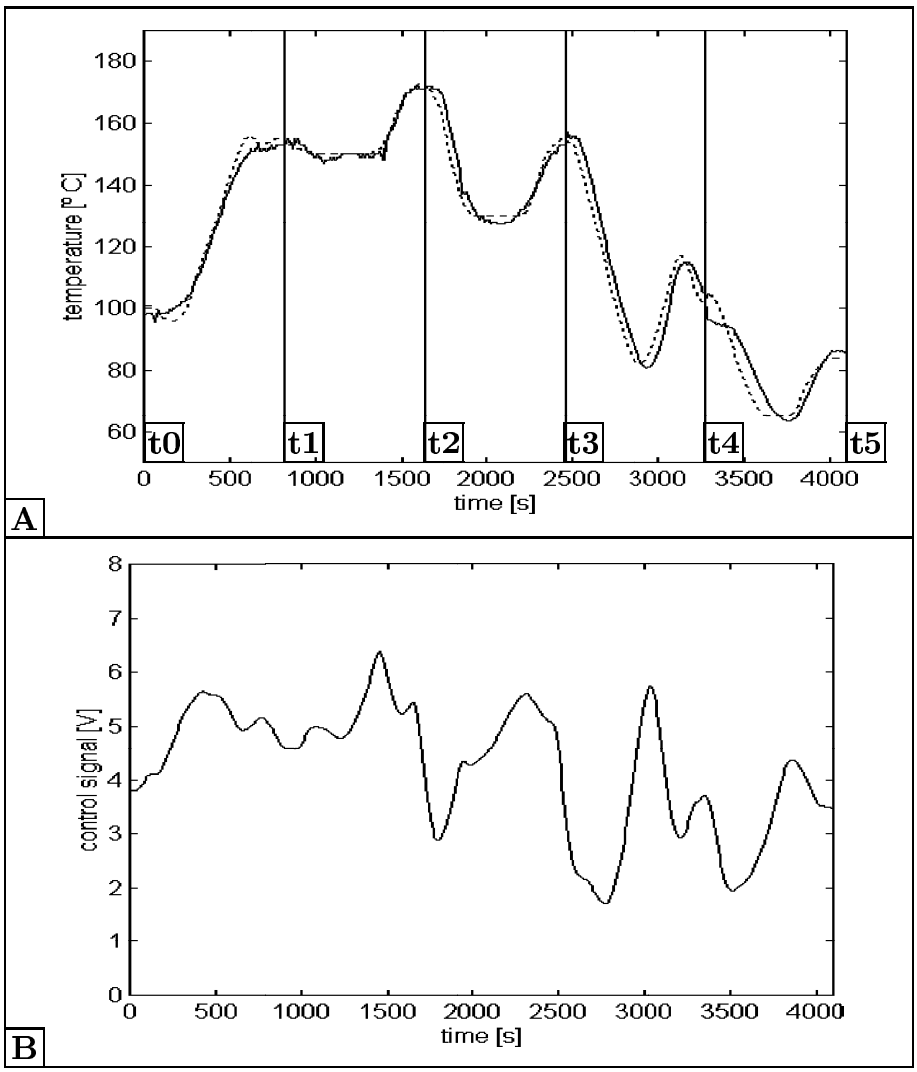


Figure 3.24: The set of experiments representing a continuous run of the system through goal states in step-wise way for the parameter values $a = 0.00906, b = 0.002$. In Figure **A** the full line shows the predicted temperature computed by solving the ill-posed inverse problem, the dashed line is the experimental temperature. In Figure **B** there is a control signal - the input to the heater. In the Figures **t0-t5** which are in Figure 3.25 there is a comparison of the prescribed goal state (full line) with the actually measured state (the stars) and the internal simulation state (dashed line) which is maintained as a starting state for further steps.

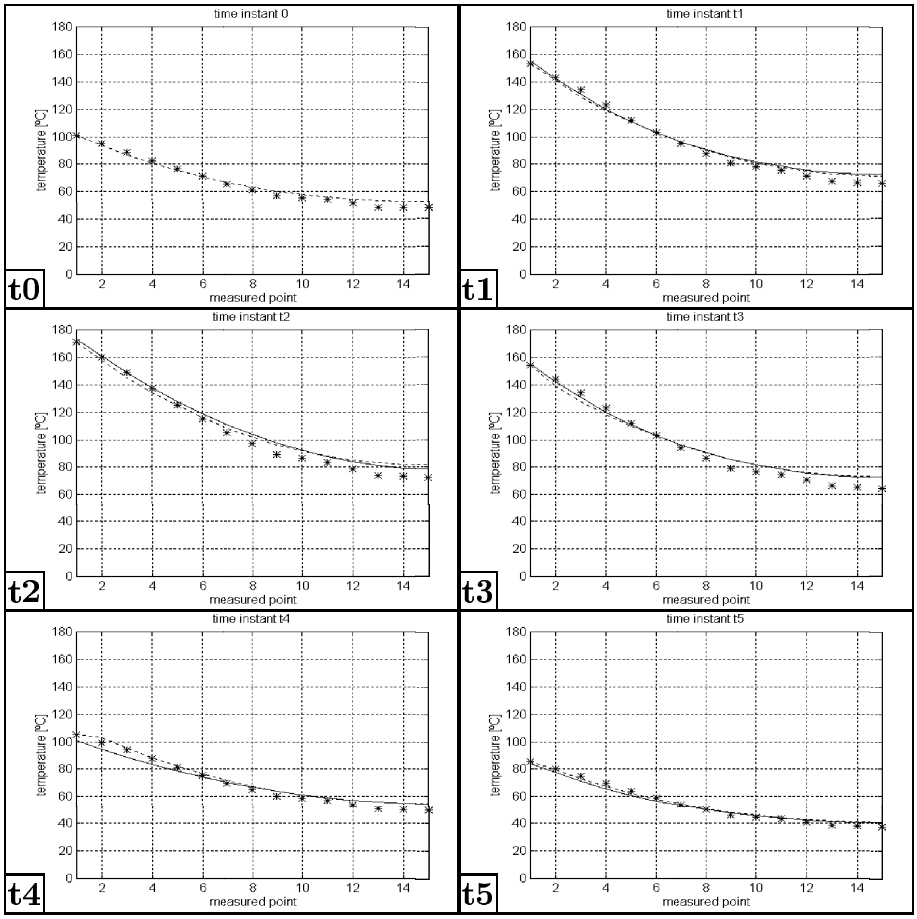


Figure 3.25: In the Figures **t0-t5** there is a comparison of the prescribed goal state (full line) with the actually measured state (the stars) and the internal simulation state (dashed line) which is maintained as a starting state for further steps. For more details see Figure 3.24 first.

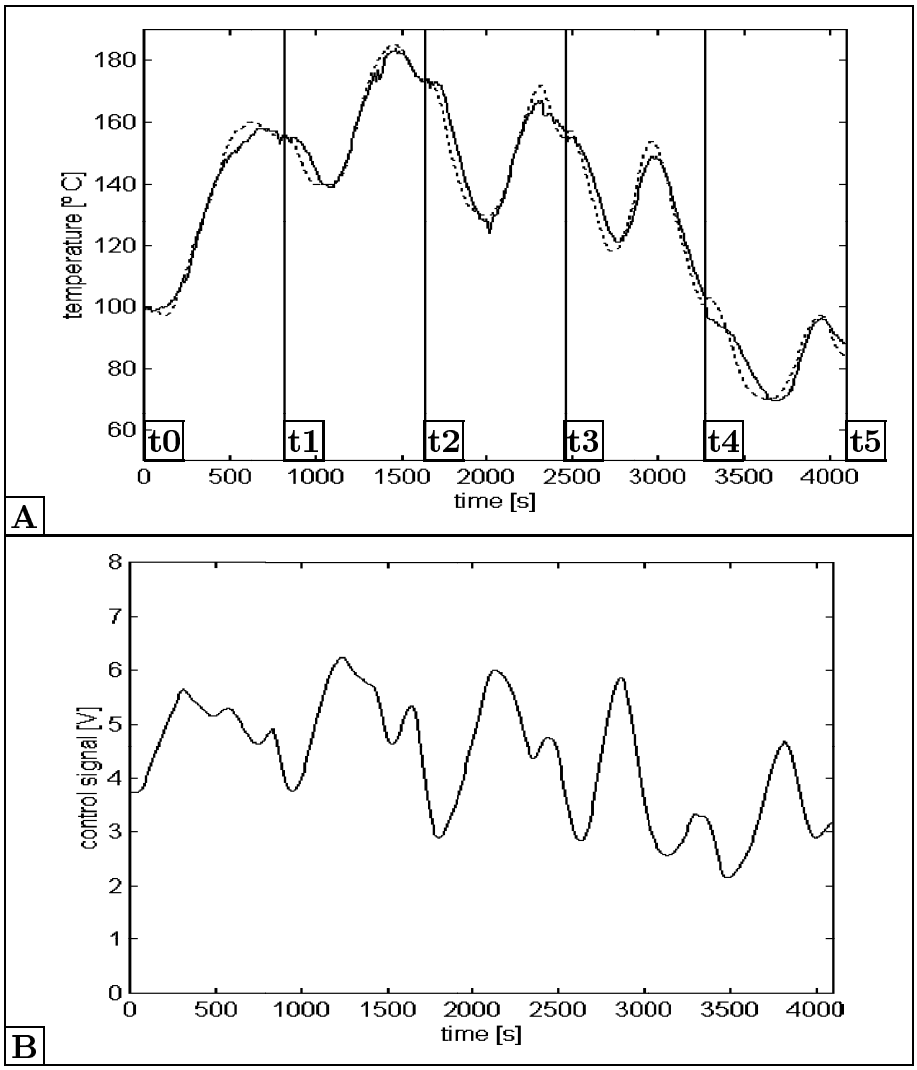


Figure 3.26: The set of experiments representing a continuous run of the system through goal states in step-wise way for the parameter values $a = 0.0135, b = 0.00485$. In Figure **A** the full line shows the predicted temperature computed by solving the ill-posed inverse problem, the dashed line is the experimental temperature. In Figure **B** there is a control signal - the input to the heater. In the Figures **t0-t5** which are in Figure 3.27 there is a comparison of prescribed goal state (full line) with the actually measured state (the stars) and the internal simulation state (dashed line) which is maintained as a starting state for further steps.

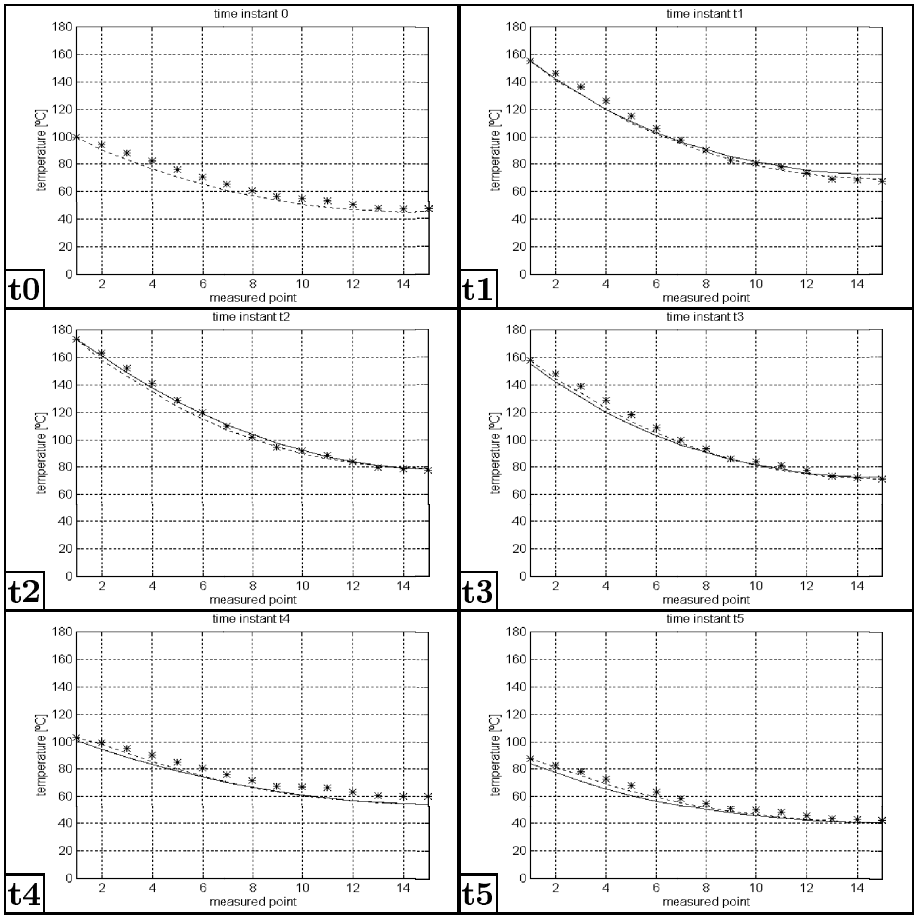


Figure 3.27: In the Figures **t0-t5** there is a comparison of the prescribed goal state (full line) with the actually measured state (the stars) and the internal simulation state (dashed line) which is maintained as a starting state for further steps. For more details see Figure 3.24 first.

3.4 Regularization Based Predictive Control

In this section we describe a close-loop predictive control scheme for boundary control of thermal system which we denote Regularization Based Predictive Control (RBPC). The method incorporates the ill-posed problems solving theory into the control law thus providing a concrete realization of conceptual ideas mentioned in the introductory chapter (chapter 1) as well as in the introduction to chapter 3.

The finite difference method is used as a basis for derivation of the predictor. Because this method naturally includes nonlinearities we consider an augmented version of the original model in the form:

$$\begin{aligned} \frac{\partial}{\partial t}Y(x,t) - \frac{\partial}{\partial x}\left(a^2(Y)\frac{\partial}{\partial x}Y(x,t)\right) + bY(x,t) = 0 \\ Y(x,t_0) = Y_0(x), Y(0,t) = u(t), \frac{\partial Y(L,t)}{\partial x} = 0 \\ 0 \leq x \leq L, \quad t \geq t_0, \quad a \neq 0 \\ a^2 = \frac{\lambda}{c.\rho}, \quad b = \frac{h}{c.\rho} \end{aligned} \tag{3.39}$$

where all quantities have the same meaning as in the equation (3.1) except the thermal diffusivity coefficient $a^2(Y)$ which is now understood to be a function of temperature.

We consider two basic sorts of applications for the RBPC algorithm. The first one is a situation where the feedback information is available from inside of the spatial domain. In this setting the boundary temperature is considered to be the control and the distributed state of the system $Y(x,t)$ is the output. A development of such algorithm is a main subject of this section.

However, the above situation is usually not present in the industry. In many industrial plants (like reheating furnaces which are our main concern here) only the boundary temperature can be measured. In the latter case we must rely on a simulation to provide the information about an internal state of the system along the spatial domain. The structure of a control algorithm in this more complicated setting is supposed to be similar as was in the previous section 3.3 and consists of three basic parts:

1. RBPC algorithm which in every time step provides a reference signal for cascaded low dimensional predictive algorithm of a GPC class.

2. SISO (or low dimensional) GPC algorithm which controls the finite dimensional heater subsystem and as a reference value for output uses boundary temperature produced by RBPC.

3. Simulation model which based on the real measured temperature on the surface of the system provides a feedback information about the internal temperature distribution to RBPC algorithm.

The above mentioned structure has already proved (also experimentally) its usefulness in the control algorithm described in the previous section. The difference is now in the following four factors (i) the control from the boundary of the system to its internal state is now made in a recursive close-loop predictive way compared with the open-loop method used in the previous section (ii) the simulation and control is based on a completely different formalism i.e. on finite difference schemes (iii) there is a new control law which incorporates the theory of ill-posed problems inversion and (iv) the nonlinearity caused by the temperature dependence of the thermal diffusivity coefficient is incorporated within the algorithm.

As far as other blocks mentioned in the above "meta-algorithm" are well developed (see previous sections, as well as [13], [40, 41]) in the rest of this section we focus on development and simulation tests of the RBPC algorithm.

3.4.1 Predictor

Having defined the model of the process as is (3.39) we can recall the finite difference scheme from section 3.2.2 with a new notation which will be used consistently during this section (see Table 3.28). The definition of matrices $\mathbf{P}^{(l)}$ and $\mathbf{C}$ remains the same therefore it is not repeated in figure 3.28. However, we repeat the definition of vector $\mathbf{f}^{(l)}$ to stress that in the case of boundary control only the first element of this vector is not zero. This fact allows a derivation of a computationally efficient predictor.

$$\begin{aligned} \mathbf{A}_U^{(l)}\mathbf{y}^{(l)} &= \mathbf{A}_L^{(l)}\mathbf{y}^{(l-1)} + \omega(\mathbf{f}^{(l)} + \mathbf{f}^{(l-1)}) \\ \mathbf{A}_U^{(l)} &= \mathbf{C} + \omega\mathbf{P}^{(l)}, \\ \mathbf{A}_L^{(l)} &= \mathbf{C} - \omega\mathbf{P}^{(l-1)}, \end{aligned} \tag{3.40}$$

$$\mathbf{E}^{(l)} = (\mathbf{A}_U^{(l)})^{-1}\mathbf{A}_L^{(l)},$$
$$\tfrac{1}{2}\mathbf{A}^{(l)} = (\mathbf{A}_U^{(l)})^{-1}$$

$$\begin{aligned} & f_1^l = p(x_0 + \frac{h}{2}, t_l)u(t_l) \\ k = 2, ..., n, \quad & f_k^l = 0 \end{aligned} \tag{3.41}$$

$$\mathbf{v}^l = \omega\mathbf{f}^{(l)}$$

Figure 3.28: A new notation for the finite difference scheme.

We start this section with deriving what we call a *local predictor* until it is summarized in Theorem 7 below. It is a predictive scheme which predicts the state k-steps ahead for the whole spatial domain. Then, at the end of the section, the predictor is defined as a set of local predictors with an option not to consider all spatial points of certain local predictors.

In the following, l denotes a discrete time starting from 0. Let k is a fixed number $k > l$. The matrices considered have dimension $n \times n$ or $n \times k$ (depending on a context) where n is the discretization density in space of the finite difference scheme and k determines the number of steps ahead to which the predictor computes the future state.

The basic equation can be written in the new notation as

$$\mathbf{y}^l = \mathbf{E}^{(l)}\mathbf{y}^{l-1} + \tfrac{1}{2}\mathbf{A}^{(l)}(\mathbf{v}^l + \mathbf{v}^{l-1}). \tag{3.42}$$

Then the state $\mathbf{y}^{l+k}$ can be expressed by a recursive application of equation (3.42) as

$$\begin{aligned}
\mathbf{y}^{l+k} = \mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\mathbf{E}^{(l+1)}\mathbf{y}^l + \\
+\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\ \tfrac{1}{2}\mathbf{A}^{(l)}(\mathbf{v}^l + \mathbf{v}^{l+1}) + \\
+\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+3)}\ \tfrac{1}{2}\mathbf{A}^{(l+1)}(\mathbf{v}^{l+1} + \mathbf{v}^{l+2}) + \\
+ \dots + \\
+\mathbf{E}^{(l+k)}\ \tfrac{1}{2}\mathbf{A}^{(l+k-1)}(\mathbf{v}^{l+k-2} + \mathbf{v}^{l+k-1}) + \\
+\ \tfrac{1}{2}\mathbf{A}^{(l+k)}(\mathbf{v}^{l+k-1} + \mathbf{v}^{l+k})
\end{aligned}$$

Now, we observe that the matrix $\mathbf{T}^{(k,l)} = \mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+1)}$ can be computed by the following matrix recursion:

$$\begin{aligned}
\mathbf{T}^{(k,l+1)} &= \mathbf{E}^{(l+k+1)}\mathbf{T}^{(k,l)}(\mathbf{E}^{(l+1)})^{-1} \\
\mathbf{T}^{(k,0)} &= \mathbf{E}^{(k)} \dots \mathbf{E}^{(2)}\mathbf{E}^{(1)}.
\end{aligned}$$

Moreover, because the vectors $\mathbf{v}^i$ have only the first coordinate nonzero, the terms $\mathbf{E}^{(i+k)} \dots \mathbf{E}^{(i+2)}\ \frac{1}{2}\mathbf{A}^{(i)}(\mathbf{v}^i + \mathbf{v}^{i+1})$ can be interpreted as the first columns of the respective matrices $(\mathbf{E}^{(i+k)} \dots \mathbf{E}^{(i+2)} \times$
$\frac{1}{2}\mathbf{A}^{(i)})_{\bullet 1}$ times a scalar $\mathbf{v}_1^i + \mathbf{v}_1^{i+1}$ ($X_{\bullet i}$ denotes the i-th column of matrix X).

Employing the idea above and intending to obtain an incremental form of the predictor it can be written:

$$\begin{aligned}
\mathbf{y}^{l+k} = \mathbf{T}^{(k,l)}\mathbf{y}^l + \\
+(\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\ \tfrac{1}{2}\mathbf{A}^{(l)})_{\bullet 1}(2\mathbf{v}_1^l + \Delta\mathbf{v}_1^l) + \\
+(\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+3)}\ \tfrac{1}{2}\mathbf{A}^{(l+1)})_{\bullet 1}(2\mathbf{v}_1^l + 2\Delta\mathbf{v}_1^l + \Delta\mathbf{v}_1^{l+1}) + \\
+ \dots + \\
+(\ \tfrac{1}{2}\mathbf{A}^{(l+k)})_{\bullet 1}(2\mathbf{v}_1^l + 2\Delta\mathbf{v}_1^l + \dots + 2\Delta\mathbf{v}_1^{l+k-2} + \Delta\mathbf{v}_1^{l+k-1})
\end{aligned}$$

where $\Delta \mathbf{v}^l = \mathbf{v}^{l+1} - \mathbf{v}^l$.

The above columns can be collected in the matrix $\mathbf{F}^{(k,l)}$ defined as

$$\mathbf{F}^{(k,l)} = (\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)} \ \tfrac{1}{2}\mathbf{A}^{(l)})_{\bullet 1} \mid \dots \mid (\ \tfrac{1}{2}\mathbf{A}^{(l+k)})_{\bullet 1}).$$

Then, two new vectors can be defined as

$$\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} = \underbrace{(\mathbf{v}_1^l, \dots, \mathbf{v}_1^l)^T}_{\text{k times}}$$
$$\Delta\tilde{\mathbf{u}}^{k,l} = (\Delta\mathbf{v}_1^l, \dots, \Delta\mathbf{v}_1^{l+k-1})^T$$

Using the new notation the vector on the right hand side of the matrix $\mathbf{F}^{(k,l)}$ can be expressed as

$$\begin{pmatrix} 2\mathbf{v}_1^l + \Delta\mathbf{v}_1^l \\ 2\mathbf{v}_1^l + 2\Delta\mathbf{v}_1^l + \Delta\mathbf{v}_1^{l+1} \\ \dots \\ 2\mathbf{v}_1^l + 2\Delta\mathbf{v}_1^l + \dots + 2\Delta\mathbf{v}_1^{l+k-2} + \Delta\mathbf{v}_1^{l+k-1} \end{pmatrix} = 2\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l}$$

where the matrix $\mathbf{B}$ is a lower-triangular matrix with one's on the diagonal and two's under the diagonal

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & \dots & 0 \\ & \dots & \dots & \\ 2 & \dots & 1 & 0 \\ 2 & 2 & 2 & 1 \end{pmatrix} \tag{3.43}$$

The predicted state can be written using the above derivation as

$$\mathbf{y}^{l+k} = \mathbf{T}^{(k,l)}\mathbf{y}^l + \mathbf{F}^{(k,l)}(2\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l})$$
$$\mathbf{y}^{l+k} = \mathbf{T}^{(k,l)}\mathbf{y}^l + 2\mathbf{F}^{(k,l)}\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \mathbf{F}^{(k,l)}\mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l}$$

The formula contains three parts, the first represents the evolution of the initial condition with a zero boundary condition, the second one is the impact of the current applied control and the last term is a contribution from future controls. Therefore, the future control part has been isolated from the rest of the evolution operator.

There is also a nice recursive formula for the matrix $\mathbf{F}^{(k,l)}$. Following the above derivation it can be seen that proceeding from the state $\mathbf{y}^l$ to the state $\mathbf{y}^{l+1}$ we must (roughly said) multiply by matrix $\mathbf{E}^{(l+k+1)}$ and add a term containing the new boundary condition. Because only the first columns of the respective matrices are present in matrix $\mathbf{F}^{(k,l)}$ a straightforward examination shows that the following recursion is valid:

$$\mathbf{F}^{(k,l+1)} = \mathbf{S}_{-1}(\mathbf{E}^{(l+k+1)}\mathbf{F}^{(k,l)},\ \tfrac{1}{2}\mathbf{A}_{\bullet 1}^{(l+k+1)})$$

where $\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z})$ is a shift operator which shifts the columns of matrix $\mathbf{X}$ to the left, in a way that $\mathbf{X}_{\bullet 1}$ is forgotten, the second column is moved to the first position, the third to the second and so on. The last column is set to be equal to vector $\mathbf{z}$.

$$\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z}) = (\mathbf{X}_{\bullet 2}, \mathbf{X}_{\bullet 3}, \ldots, \mathbf{X}_{\bullet k}, \mathbf{z})$$

We can summarize the information obtained so far about the predictor in the following theorem.

Theorem 7 (RBPC Local Predictor) *Let a finite difference scheme with n discretization steps is given and let l denotes the discrete time starting at $l = 0$. Then a k-step ahead incremental predictor $\mathcal{P}^{k,l}$ assigned to this scheme equals*

$$\mathbf{y}^{l+k} = \mathcal{P}^{k,l}(\mathbf{y}^l, \Delta\mathbf{u}^{k,l}) = \mathbf{T}^{(k,l)}\mathbf{y}^l + 2\omega p_0 \mathbf{F}^{(k,l)}\mathbf{u_1}^{k,l} + \omega p_0 \mathbf{F}^{(k,l)}\mathbf{B}\Delta\mathbf{u}^{k,l}.$$

$\mathbf{T}^{(k,l)}$ and $\mathbf{F}^{(k,l)}$ are $n \times n$ resp. $n \times k$ matrices defined by the following matrix recurences

$$\begin{aligned} \mathbf{T}^{(k,l+1)} &= \mathbf{E}^{(l+k+1)}\mathbf{T}^{(k,l)}(\mathbf{E}^{(l+1)})^{-1} \\ \mathbf{T}^{(k,0)} &= \mathbf{E}^{(k)} \ldots \mathbf{E}^{(2)}\mathbf{E}^{(1)} \end{aligned} \tag{3.44}$$

$$\mathbf{F}^{(k,l+1)} = \mathbf{S}_{-1}(\mathbf{E}^{(l+k+1)}\mathbf{F}^{(k,l)},\ {}^{\frac{1}{2}}\mathbf{A}_{\bullet 1}^{(l+k+1)}) \tag{3.45}$$
$$\mathbf{F}^{(k,0)} = (\mathbf{E}^{(k)} \ldots \mathbf{E}^{(2)}\ {}^{\frac{1}{2}}\mathbf{A}^{(1)})_{\bullet 1} \mid \ldots \mid (\ {}^{\frac{1}{2}}\mathbf{A}^{(k)})_{\bullet 1})$$

where $\mathbf{S}_{-1}$ *is a shift operator defined as*

$$\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z}) = (\mathbf{X}_{\bullet 2}, \mathbf{X}_{\bullet 3}, \ldots, \mathbf{X}_{\bullet k}, \mathbf{z}).$$

Vectors ${\mathbf{u_1}}^{k,l}$, $\Delta\mathbf{u}^{k,l}$ *and matrix* $\mathbf{B}$ *are defined as*

$${\mathbf{u_1}}^{k,l} = (u^l, \ldots, u^l)^T,\ u^l = u(t_l)$$
$$\Delta\mathbf{u}^{k,l} = (\Delta u^1, \ldots, \Delta u^k)^T,\ \Delta u^i = u(t_{l+i}) - u(t_{l+i-1})$$
$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & \ldots & 0 \\ & \ldots & \ldots & \\ 2 & \ldots & 1 & 0 \\ 2 & 2 & 2 & 1 \end{pmatrix}.$$

$\omega = \frac{\tau}{2h^2}, p_0 = p(x_0 + h/2, t_l)$ *and* τ, h *are the discretization steps in time resp. in space.*

Some of the complexities in the above derivation of the predictor are related to the nonlinearity in the model (3.39). In the case when the thermal diffusivity coefficient and coefficient b are constant the matrices $\mathbf{T}^{(k,l)}$ and $\mathbf{F}^{(k,l)}$ would be constant which would lead also to a faster algorithm.

The local predictor from the above theorem can be used for constructing the generalized version of a predictor where more predictions to the various depth in time and space are made. This is necessary to gain a control over the spatial domain behavior of the system. Before we proceed defining the final version of the predictor there are two notes to the above theorem.

The pre-multiplication with ωp_0 is important because it allows to use more local predictors with different space grids located at a different future times. The unknown variable of control increments is made independent of the particular grid density of a local predictor.

Another aspect of the theorem is that it serves well in organizing the structure of the predictor but it can not be directly used as an algorithm. The problem is in the matrix recursion (3.44). The characteristic high stability features of Crank-Nicolson scheme stems from the fact that the transition matrix $\mathbf{E}^{(k)}$ has all eigenvalues less than 1. That means that the matrix $(\mathbf{E}^{(l+1)})^{-1}$ which is on the right hand side of the recursion (3.44) has all eigenvalues larger than 1 which makes the matrix recursion totally unstable. However, we are not in fact interested in the recursion itself but in the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ which can be computed by the simplified scheme

$$\mathbf{A}_U^{(l)}\mathbf{y}^{(l)} = \mathbf{A}_L^{(l)}\mathbf{y}^{(l-1)}$$

because the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ corresponds exactly to the solution of the system with zero boundary conditions. Moreover this is in fact also computationally much more efficient as far as both matrices in the recursion are dense to the contrary to standard Crank-Nicolson step which involves only tridiagonal matrices. Therefore the computational complexity of computing the matrix recurrence would be $O(n^3)$ whereas the complexity of computing the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ by k simplified steps as above is only $O(kn)$.

On the other hand the second recurrence (3.45) is perfectly stable as far as it involves only the stable matrix $\mathbf{E}^{(k)}$.

In general, the predictor is constructed taking a set of local predictors

$$\mathcal{P}^{k,l}(\mathbf{y}^l, \Delta\mathbf{u}^{k,l})$$

all starting from the same state but having different time k of the prediction as well as different discretization grid. Moreover from each predictor only a certain subset of points can be chosen.

In more detail, suppose that we are at a time moment l and the horizon of the prediction k_h is given. Then the predictor is constructed as a linear map from the space of control increments $\Delta\mathbf{u}^{k,l} \in \mathbf{R}^{k_h}$ to the space of discretized solutions given in a system of spatial points which reads as

$$\begin{pmatrix} \mathcal{P}^{k_1,l}_{g_1,d_1} \\ \cdots \\ \mathcal{P}^{k_i,l}_{g_i,d_i} \\ \cdots \\ \mathcal{P}^{k_m,l}_{g_m,d_m} \end{pmatrix} \begin{pmatrix} \Delta u^1 \\ \cdots \\ \Delta u^{k_m} \end{pmatrix} = \begin{pmatrix} w_1^{k_1} \\ \cdots \\ w_{j_1}^{k_1} \\ \cdots \\ w_{d_1}^{k_1} \\ \cdots \\ \cdots \\ w_1^{k_i} \\ \cdots \\ w_{j_i}^{k_i} \\ \cdots \\ w_{d_i}^{k_i} \\ \cdots \\ \cdots \\ w_1^{k_m} \\ \cdots \\ w_{j_m}^{k_m} \\ \cdots \\ w_{d_m}^{k_m} \end{pmatrix}$$

where the upper indexes have the same meaning as in Theorem 7. The bottom left index means the number of equidistant discretization intervals and the bottom right denotes the depth of the respective local predictor. On the right hand size we have reference signal at the corresponding time moments and space points.

It is a collection of local predictors according to the Theorem 7 but only certain number of spatial points (depth) is considered. The above construction is illustrated in Figure 3.29 where the points considered for each predictor are emphasized by a larger bullet.

In fact the time positions of the respective local predictors can be made also non-equidistant and we can choose only internal points but for the simplicity (and computational complexity) reasons in the following we restrict the consideration to the equidistant time grid and to the depth parametrization of each predictor.

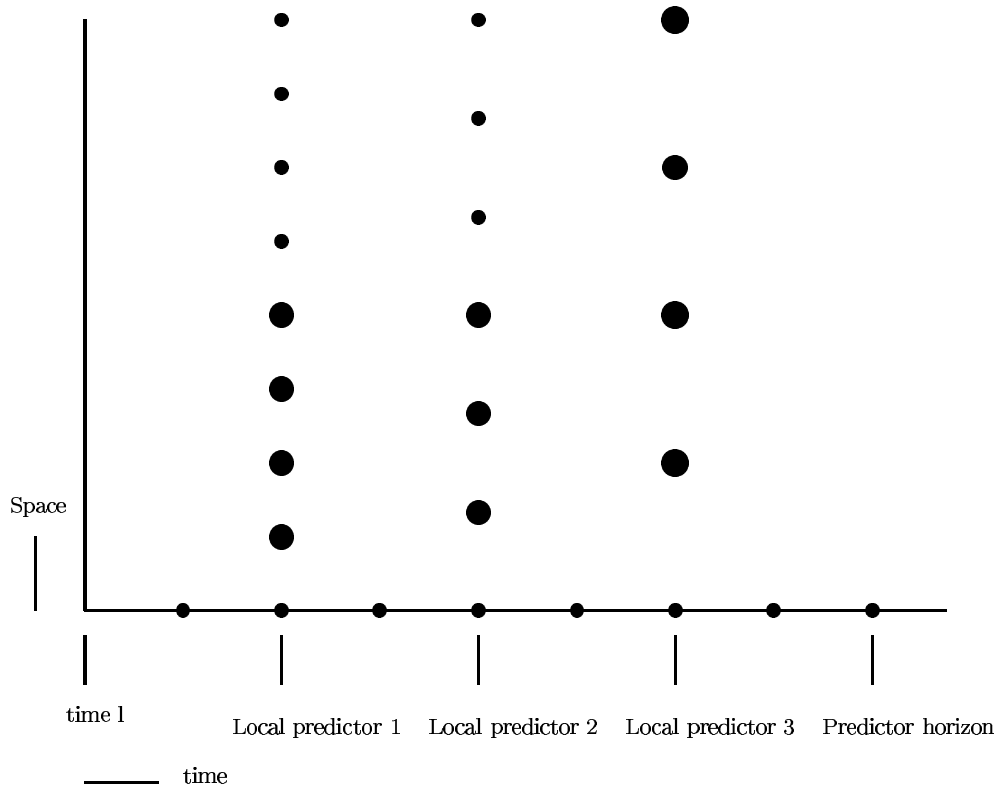


Figure 3.29: A predictor structure example. The predictor is constructed from 3 local predictors. Note also that the time horizon can go beyond the last local predictor. See the text for explanation.

(There might be some reasons why to consider non-equidistant times for the respective local predictors, but this will need further research.)

For the schemes as in Figure 3.29 we use a notation (h.(t.g.d)....(t.g.d)) where h is an overall prediction horizon, each triple represents a local predictor, the first element t in each triple is a prediction horizon of the local predictor, g is a number of discretization intervals for the local predictor and d is the local predictor depth. For example the predictor in Figure 3.29 is denoted by (8.(2.8.4).(4.6.3).(6.4.4)).

The number of possibilities how to design a predictor is immense, however, we have observed some pattern for the heat equation. As far as the system we consider is exponentially smoothing the signal is strongly damped also in the spatial direction. We can draw the contour plots for unit step response of the system (3.39). They are in Figure 3.30. The first contour (labeled with an arrow) is of interest because it shows the boundary between the area where the signal is non-zero and the rest. We adapt the basic predictor structure to this pattern and one such predictor with a structure (15.(1.8.1).(3.8.2).(7.8.4).(15.8.8)) is schematically drawn in the figure. We have experimented with many different schemes as will be to some extend described in the next sections but the scheme in Figure 3.30 seems to have the best performance so far. The predictor of this scheme can be written as

$$\begin{pmatrix} \mathcal{P}_{8,1}^{1,l} \\ \mathcal{P}_{8,2}^{3,l} \\ \mathcal{P}_{8,4}^{7,l} \\ \mathcal{P}_{8,8}^{15,l} \end{pmatrix} \begin{pmatrix} \Delta u^1 \\ \cdots \\ \Delta u^{15} \end{pmatrix} = \begin{pmatrix} w_1^1 \\ w_1^3 \\ w_2^3 \\ w_1^7 \\ w_2^7 \\ w_3^7 \\ w_4^7 \\ w_1^{15} \\ \cdots \\ w_8^{15} \end{pmatrix}$$

Another aspect in designing the predictor is the fact that the space points must be traded against the controls if we want to avoid rank deficiency of re-

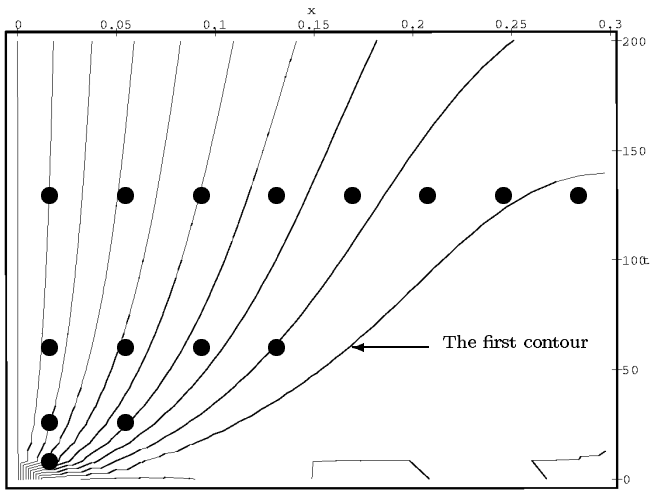


Figure 3.30: The contours for a unit response of the heat equation. x-axis represents the spatial coordinate, y-axis the time. The point $(0,0)$ is in lower left corner of the picture. The contours drawn in the figure represent the curves with a constant value of the unit response. The first contour which separates a domain with a non-zero value is labeled by an arrow. The domain with non-zero values of the unit response lies to the left of this contour. The predictor spatial points are represented with the bullets. The predictor in this example consists of four local predictors all having the same dicretization step. Only the active depth of each local predictor is drawn. The depth of the predictors as well as their position in the time has exponentially decreasing density.

sulting matrices (non-controlability). Therefore not all control increments have a corresponding spatial point at the given time.

3.4.2 Regularized solutions

In order to formulate a control law for RBPC we must recall some notions and results from the theory of ill-posed problems solving in a more precise way. We have mentioned them already in an informal way in section 2.3.

We follow [50, 51] in defining the basic notions.

For the purpose of this short section we denote by Z, U Hilbert spaces, D is a closed convex set of constraints with $0 \in D$ and operator A maps from Z to U.

The aim of regularization is to find a solution of the operator equation

$$Az = u$$

. However, as far as in practice we usually do not have accurate form of the operator at disposal, it is supposed that there is an approximation of A denoted by $A_h, \| A_h - A \| \leq h$ as well as an approximation of the right hand side denoted by $u_\delta, \| u_\delta - u \| \leq \delta$. The perturbed equation

$$A_h z = u_\delta$$

can even have no solutions, however in the setting we use here there is an assumption that the unperturbed equation $Az = u$ has a unique solution.

This fits very well to our case because the system we are investigating is controllable that means there is always a boundary condition $u(t)$ which can drive the system to the desired end-state in a finite time. However, the finite dimensional approximation can well have no solution in some circumstances.

The possible non-uniqueness of the original equation can be handled easily by defining the *normal solution* of the problem as a solution of the extremal problem $\| \bar{z} \|^2 = inf \| z \|^2$ where $z \in \{z \mid z \in D, Az = u\}$. Therefore we assume without lost of generality that there is a unique solution to the original equation.

As was already mentioned in section 2.3 instead of directly solving the equation $Az = u$ the following *smoothing functional* is defined

$$M^\alpha(z) = \| A_h z - u_\delta \|^2 + \alpha \| z \|^2$$

. The functional is minimized for an appropriate value of the *regularization parameter* α and the way how this parameter is found is the key to the regularization method.

There is a basic fact which must be established in order to obtain a worthwhile approach. We must know that if the perturbation (i.e. error) (h, δ) goes to zero in limit that there is a possibility to design the regularization parameter $\alpha(h, \delta)$ as a function of the perturbation parameters so that the minimizer of the smoothing functional goes to the solution of the original equation in limit.

A well established design [50] for the regularization parameter consists in the following:

Definition 8 (Generalized residuum) *The real function* $\rho_{(h,\delta)}(\alpha) = \| A_h z^\alpha - u_\delta \|^2 - (\delta + h \| z^\alpha \|)^2 - \mu^2(u_\delta, A_h)$ *is called* generalized residuum *where* $\mu^2(u_\delta, A_h) = \inf_{u \in D} \| A_h z - u_\delta \|$ *is a* degree of inconsistency *and* z^α *is a minimizer of the smoothing functional* $M^\alpha(z)$.

The *regularized solution* is given by the algorithmic *Principle of generalized residuum.*

Definition 9 (Principle of generalized residuum) *If* $\| u_\delta \|^2 \leq \delta^2 + \mu^2(u_\delta, A_h)$ *the regularized solution* z^α *is defined to be identically zero. Otherwise the regularized solution is given as* $z^{\alpha^\star}$ *where* $\alpha^\star$ *is the root of generalized residuum* $\rho(\alpha^\star) = 0$.

The correctness of the above definition (i.e. the usefulness of the algorithm) is given by the theorem:

Theorem 10 (Tikhonov [50]) *Let* A, A_h, u, u_δ *are given as above and* $(h, \delta) \to 0$. *Then the generalized solution* $z^{\alpha^\star}$ *given by the Principle of generalized residuum is convergent* $\lim_{(h,\delta)\to 0} z^{\alpha^\star} = z$ *to the unique solution of the operator equation* $Az = u$. *In the case* A *is not injective the convergence to the normal solution is established.*

An important design aspect of particular regularization method which follows the general theory mentioned above is the selection of norms (resp. scalar products) in the spaces Z, U. Because an inversion of the parabolic equation (3.39) is severely ill-posed problem, we choose a strongly dumping norm in the space of controls, namely the Sobolev W_2^1 norm given as $\int_0^T z^2(\tau) + \frac{dz}{d\tau}^2(\tau)\, d\tau$. The distance on the right hand side (deviation from the reference signal) is expressed with L_2 norm.

The last thing we need (at least in the basic setting) is that the generalized residuum has reasonable features so that an efficient algorithm for searching of its root can be established. Indeed:

Theorem 11 (Tikhonov [50]) *The generalized residuum has the following features:*

1. *$\rho(\alpha)$ is monotonic and nondecreasing for $\alpha > 0$.*

2. *$\lim_{\alpha\to+\infty} = \| u_\delta \|^2 - \delta^2 - \mu^2(u_\delta, A_h)$.*

3. *$\lim_{\alpha\to 0+0} = -\delta^2$.*

4. *If $\| u_\delta \|^2 > \delta^2 + \mu^2(u_\delta, A_h)$ there is a root $\alpha^\star$ of the generalized residuum which defines the regularized solution $z^{\alpha^\star}$ in an unique way.*

3.4.3 Control Law and Control Algorithm

Having prepared the necessary background we can state the control law for Regularization Based Predictive Control as:

Control Law 12 (Regularization Based Predictive Control) *Let the system is described by parabolic heat equation as in (3.39). A is an evolution operator given by the equation (3.39) and $u(t)$ is a boundary control signal. Let the perturbed operator A_h is given by the predictor $\mathbf{P}^l = (\mathcal{P}^{k_1,l}_{g_1,d_1}, \ldots, \mathcal{P}^{k_i,l}_{g_i,d_i}, \ldots, \mathcal{P}^{k_m,l}_{g_m,d_m})$ consisting in a set of local predictors as defined in section 3.4.1. Then control is given as the first $n \leq k_h$ elements of the* regularized solution *of an operator equation $Au = w$ over a finite discretization horizon k_h where w is a reference signal known over the prediction horizon.*

The practical interpretation of the perturbation parameters (h, δ) deserves some discussion at this point. The parameter h is clearly interpreted as a discretization error with which the discrete version of the predictor approximates the evolution operator given by (3.39). In different words, it is the discretization error of the finite difference Crank-Nicolson method. This equals to $O(\tau^2 + s^2)$ where τ, s are discretization steps in time resp. in space. In practice the constant in the $O(\tau^2 + s^2)$ can be easily established by few experiments.

The parameter δ defines in fact a neighborhood around the right hand side of the governing equation. Concerning this, it is well reasonable to set its value to the standard deviation of data noise.

On the other hand, it is usually sufficient to use a very conservative estimates of h, δ which can be easily obtained and then decrease their values. The reason

is that the Tikhonov regularization method trades smoothness against the right hand side fit. Therefore we can by few experiments decrease the values of the both parameters until the solution is getting to lose the desired smoothness.

Now the control algorithm defined by the above control law is refined in successive steps until we obtain computationally feasible solution. Here is the first version:

Control Algorithm 13 (Regularization Based Predictive Control) *Applying the Control law 12 we have the following control algorithm.*

1. *(*PREDICTION*) Construct the predictor from the local predictors as described in section 3.4.1.*

2. *(*INVERSION*) Find the regularized solution of the equation*

$$A(\Delta u) = w$$

.

3. *(*CONTROL*) Apply the first $n \leq k_h$ controls to the plant.*

4. *(*RECEDING HORIZON*)* $l = l + n$

5. *(*CLOSE LOOP*) Repeat from the Point 1.*

Because the Point 1. of the above algorithm has been discussed in section 3.4.1 we focus now on the Point 2. i.e. how to efficiently find the regularized solution.

To simplify the writing of formulas we denote Δu as q and we drop the superscript on the predictor $\mathbf{P}$. To find a regularized solution means first of all to be able to minimize the smoothing functional $M^{\alpha}(q)$ as was described in the previous section. The Euler (normal) equation in this case reads

$$A^{\star}Aq + \alpha q = A^{\star}w.$$

Then the equation must be discretized.

The same representation can be obtained by taking the discrete version of smoothing functional and finding the normal equation. Taking into the account

the definition of the L_2 and W_2^1 (the discrete form of W_2^1 can be obtained by an approximation of the integral $\int_0^T z^2(\tau) + \frac{dz}{d\tau}^2(\tau)\, d\tau$ by $\sum_{i=1}^{k_h} q_i^2\tau + \sum_{i=2}^{k_h} \frac{(q_i - q_{i-1})^2}{\tau}$) we obtain the discrete version of the normal equation which reads as

$$\mathbf{P}^T\mathbf{P}q + \alpha\mathbf{G}q = \mathbf{P}^T w.$$

The tridiagonal matrix $\mathbf{G}$ has a form

$$\mathbf{G} = \begin{pmatrix} 1+\frac{1}{\tau^2} & -\frac{1}{\tau^2} & 0 & 0 \\ -\frac{1}{\tau^2} & 1+\frac{2}{\tau^2} & \dots & 0 \\ & \dots & \dots & \\ & \dots & 1+\frac{2}{\tau^2} & -\frac{1}{\tau^2} \\ & & -\frac{1}{\tau^2} & 1+\frac{1}{\tau^2} \end{pmatrix} \tag{3.46}$$

Now we follow [50] in Choleski decomposition of $\mathbf{G}$ to $\mathbf{G} = \mathbf{H}^T\mathbf{H}$. This gives

$$\mathbf{P}^T\mathbf{P}q + \alpha\mathbf{H}^T\mathbf{H}q = \mathbf{P}^T w.$$

Multiplying by $\mathbf{H^{-1}}$ we obtain

$$(\mathbf{PH^{-1}})^T\mathbf{PH^{-1}}(\mathbf{H}q) + \alpha(\mathbf{H}q) = (\mathbf{PH^{-1}})^T w.$$

Introducing a change of variables $y = \mathbf{H}q$ we have the following basic form of the equation

$$\mathbf{J}^T\mathbf{J}y + \alpha y = \mathbf{J}^T w$$

where $\mathbf{J} = \mathbf{PH^{-1}}$. Now in [50] a Householder transform is used to produce a tridiagonal matrices. We proceed differently; the matrix $\mathbf{J}$ is decomposed using singular value decomposition (SVD) to the form $\mathbf{J} = \mathbf{LDR}$ where $\mathbf{L}$ resp. $\mathbf{R}$ are orthogonal transforms ($\mathbf{R^{-1}} = \mathbf{R}^T$) and $\mathbf{D}$ is a diagonal matrix of singular values (in fact the eigenvalues of $\sqrt{\mathbf{J}^T\mathbf{J}}$).

It might be objected that SVD decomposition introduces an iteration process into the control which can be a cause of algorithm failure in the case if the process is not convergent. This might be true in some situations, then we have to rely on Householder transform and tridiagonal matrices. However, the computational

simplicity as well as an elegance achieved with SVD is very appealing, moreover we did not found any instance during the simulations where the SVD algorithm would fail (even if the ill-conditioning of the matrices was of order 10^9-10^{10}), therefore we suggest that those situations can be also excluded theoretically so that we can fully rely on SVD.

To continue the derivation we can write

$$\mathbf{R}^T\mathbf{D}^T\mathbf{D}\mathbf{R}y + \mathbf{R}^T(\alpha\mathbf{R}y) = \mathbf{R}^T\mathbf{D}^T\mathbf{L}^T w.$$

Multiplying with $\mathbf{R}$ from left and introducing a new variables $\mathbf{R}y = x$ we obtain

$$(\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I})x = \mathbf{D}^T\mathbf{L}^T w.$$

Now the solution can be explicitly written as

$$x = \frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I}}\mathbf{L}^T w. \tag{3.47}$$

Let us denote by d_r the dimension of reference signal and by d_c the dimension of control increments. A short examination of the formula above shows some aspects of the solution which are now apparent. As far as $\mathbf{L}^T$ is always a regular matrix (in fact a rotator) we see that the reference signal is rotated and multiplied by the matrix $\frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D}+\alpha\mathbf{I}}$ which has on the diagonal the elements $\frac{D_{ii}}{D_{ii}^2+\alpha}$ for $i \leq min(d_r, d_c)$. The rest of the matrix $\frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D}+\alpha\mathbf{I}}$ is zero thus giving an account of the effects given by the conditions $d_r < d_c$ resp. $d_r > d_c$. The former case means introducing redundancy in control whereas the latter case means introducing a rank deficiency (uncontrollability). This leads to the conclusion that the default setting should have the same dimension of controls as of the reference signal.

It might be also interesting to consider redundant control as far as the transformation back from the above canonic representation is made firstly by $\mathbf{R}^T$ which is a rotator and then by smoothing matrix $\mathbf{H^{-1}}$. This could contain interesting geometrical moment that rotation from the subspace to the full dimension made by the matrix $\mathbf{R}^T$ can make the vector shorter. Unfortunately, we did not observe the rotation of this sort. The investigation of reasons we postpone to future research.

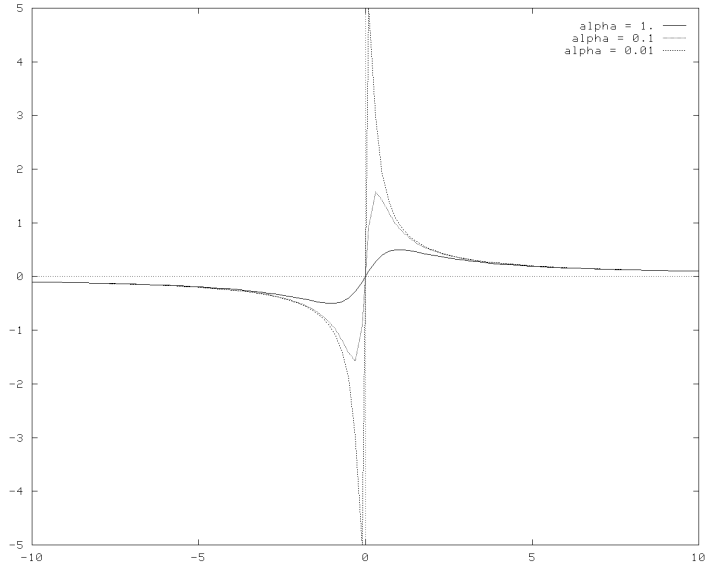


Figure 3.31: The graph of the function $\frac{x}{x^2+\alpha}$ for three values of α.

The regularization itself is now achieved by the function

$$f(x,\alpha) = \frac{x}{x^2+\alpha} = \frac{1}{x+\frac{\alpha}{x}}$$

which acts on the diagonal of the matrix $\mathbf{D}$. It is also worthwhile to see the graph of this function (in Figure 3.31) which is in fact a bounded approximation of the function $\frac{1}{x}$.

The next step is the computation of the generalized residuum because α is supposed to be its root. It would be advantageous if we would not need to go back to the original coordinates while computing the function $\rho_{(h,\delta)}(\alpha) = \| \mathbf{P}q^\alpha - w_\delta \|^2 - (\delta + h \| q^\alpha \|)^2 - \mu^2(w_\delta, \mathbf{P})$. Indeed, it is possible to use the system in the canonic form. To see this, we write

$$\| \mathbf{J}y^\alpha - w_\delta \| = \| \mathbf{PH^{-1}H}u - w \| = \| \mathbf{P}u - w \| .$$

On the other hand

$$\| \mathbf{J}y^\alpha - w_\delta \| = \| \mathbf{LDR}y^\alpha - w \| = \| \mathbf{LD}x - \mathbf{LL}^T w \| = \| \mathbf{L}(\mathbf{D}x - \mathbf{L}^T w) \| = \| \mathbf{D}x - \mathbf{L}^T w \|$$

because $\mathbf{L}$ is an isometric operator (rotator).

But the solution is expressed explicitly in the canonic coordinates, therefore the final expression for residuum reads

$$|| \mathbf{P}q^{\alpha} - w_{\delta} ||=|| \frac{\alpha}{\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I}}\mathbf{L}^T w || .$$

The degree of inconsistency $\mu^2(w_{\delta}, \mathbf{P})$ is not computed at all. A short inspection of the formula (3.47) for the solution of the normal equation as well as the formula for residuum shows that the degree of inconsistency being nonzero can be caused by two reasons. Either the rank deficiency occurs because there are not enough controls from the beginning or the dimension of the control is larger or equal than the dimension of reference signal but in this case the system itself is not controllable. In all cases the rank deficiency means zeros on the diagonal of matrix $\mathbf{D}$ therefore these elements of vector $\mathbf{L}^T w$ are not present in the residuum value. On the other hand they constitute exactly the value of the degree of inconsistency $\mu^2(w_{\delta}, \mathbf{P})$. Therefore instead computing residuum minus the degree of inconsistency, a quantity which we call *consistent residuum* is computed. The computation of consistent residuum is simple because it means only to add the values of residuum where the diagonal of $\mathbf{D}$ is nonzero.

Finally, the quantity $(\delta + h \,||\, q^{\alpha} \,||)$ is computed according to the W_2^1 approximation mentioned above. Here note that

$$|| \, q \, ||=|| \, \mathbf{H^{-1}}y \, ||=|| \, \mathbf{H^{-1}R}^T x \, ||\leq || \, \mathbf{H^{-1}} \, || \, || \, \mathbf{R}^T x \, ||=|| \, \mathbf{H^{-1}} \, || \, || \, x \, ||$$

because matrix $\mathbf{R}^T$ is an isometric operator. The matrix $\mathbf{H^{-1}}$ is upper triangular therefore its operator norm can be easily computed by computing the W_2^1 norm of its last column.

Summarizing the above discussion we can see that for the solving of the normal equations as well as for the evaluation of generalized residuum we need in fact only the diagonal matrix $\mathbf{D}$ and the vector $r = \mathbf{L}^T w$. If $\mathbf{D}$ is provided as a vector which is natural, then the algorithm which computes the root of generalized residuum, as well as the final solution, can work using only these two vectors (and scalars like error parameters (h, δ), discretization steps for discrete norm computing etc) which leads to very efficient implementation.

Before we formulate the final version of the RBPC control algorithm there are two notes. Firstly, an examination of the above procedure for the computation of generalized residuum shows that it is in fact a function with highly specific structure. This fact suggests a possibility to have a very specific (i.e. efficient) method for finding the root of this function, even a possibility of more-less explicit solution is not completely excluded.

Secondly, according to the analysis in [50] the generalized residuum is under certain circumstances ($h = 0$, no constraints) even continuously differentiable and convex therefore a Newton method can be used for its solving. However, if $h \neq 0$ which is important condition in our case only the basic features mentioned in the preceding section are valid. Therefore we must use equation solver which do not use the derivative or convexity.

On the other hand the sequential character of the algorithm provides good approximations for the future values of parameter α and the possibility to compute the solution using only the two vectors as stated above makes the algorithm very efficient therefore some loss of speed due to the less efficient equation solver is not critical. In fact the more general equation solver provides a robustness against a crash of the algorithm due to the divergence of a specialized method.

The final version of RBPC algorithm reads as

Control Algorithm 14 (Regularization Based Predictive Control) *Applying the Control law 12 we have the following control algorithm.*

1. *(*PREDICTION*) Construct the predictor from the local predictors as described in section 3.4.1.*
   - *(a) Read (or compute) the reference signal.*
   - *(b) Compute the free response of the system according to Theorem 7 for all local predictors.*
   - *(c) Form the predictor matrix by recursion (3.45) from Theorem 7 for each local predictor.*
   - *(d) Compute the current control term.*

(e) *Form the predictor matrix and the right hand side using the depth information for local predictors.*

2. *(*INVERSION*) Find the regularized solution of the equation*

$$A(\Delta u) = w$$

.

(a) *Compute SVD of the predictor matrix. Compute the matrices* $\mathbf{H^{-1}}, \mathbf{R}^T$ *and* $\mathbf{L}^T$.

(b) *Compute the vector* $r = \mathbf{L}^T w$.

(c) *Find a root of generalized residuum using the vectors* d, r, *where the vector* d *contains the diagonal elements of* $\mathbf{D}$.

(d) *Transform the solution back from canonic coordinates to control increments.*

3. *(*CONTROL*) Apply the first* $n \leq k_h$ *controls to the plant.*

4. *(*RECEDING HORIZON*)* $l = l + n$

5. *(*CLOSE LOOP*) Repeat from the Point 1.*

3.4.4 Simulation Experiments and Extensions

To illustrate the functionality and basic features of the method developed in the previous sections we provide results of simulations which show a boundary control of the system described by the equation (3.39).

The control algorithm 14 is illustrated on a process with load disturbance randomly changing the set point. The control goal is to reject the load disturbance shown in Figure 3.32 where the set point is changed four times. The states of the system corresponding to four set points have the boundary temperature 100, 50, 200 and 70 degrees Celsius as it is shown in the figure.

First of all, we investigate some basic predictor configurations according to the predictor synthesis in section 3.4.1. To illustrate the quality of control the

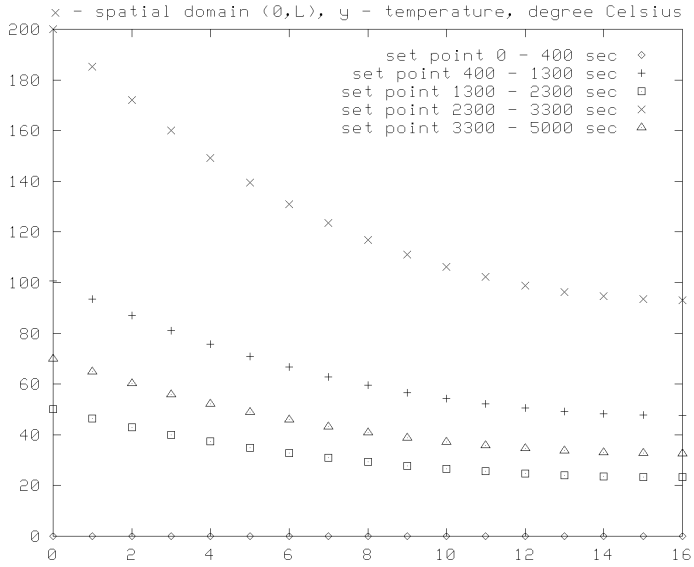


Figure 3.32: The load disturbance for simulation tests of RBPC algorithm.

discrete L_2 norm of set point against the discrete L_2 norm of system state is shown. These plots are linked with plots of control signal corresponding to each simulation.

Four basic configurations are computed, namely:

- (32.(8.8.8).(16.8.8).(24.8.8).(32.8.8)) - this configuration considers the same depth for each local predictor with equidistant positioning of local predictors in time. The simulation results are in Figure 3.33.

- (32.(4.8.8).(8.8.8).(16.8.8).(32.8.8)) - the configuration considers the same depth for each local predictor with exponential positioning of local predictors in time. The simulation results are in Figure 3.34.

- (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)) - the predictor structure which concerns the exponential smoothing as is shown in Figure 3.30. For simulation results see Figure 3.35.

- (32.(32.32.32)) - the predictor structure concerning one local predictor at the end of the prediction horizon. The simulation results are in Figure 3.36.

All four simulations are made for the same values of perturbation parameters $h = 0.01, \delta = 0.1$.

One important conclusion from the above simulations is that the algorithm designer can choose between different responses considering different predictor structures. The predictor structure developed in the previous sections seems to be enough flexible to capture different needs with respect to the spatial distribution of tracking error.

On the other hand the design of predictor from the point of view of optimal placement of local predictors would need a further theoretical investigation. Experimentally, the configuration with the best overall performance is suggested in Figure 3.35. This configuration has better overall performance compared with the configurations in Figure 3.33 and in Figure 3.34.

The difference in behavior between the scheme (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)) and (32.(32.32.32)) deserves a note. The scheme in Figure 3.34 has a slightly better behavior of control (smoother controls) but as can be observed on the simulation results there is also a considerable drawback of this scheme. The control algorithm with this scheme has a feature that at the moment when a set point change is seen in the future the algorithm immediately leaves current set point. In the case of our concrete experimental data it means that the system leaves the current set point about 300 seconds earlier as necessary. Therefore, despite a slightly better controls of the scheme (32.(32.32.32)), we consider the configuration (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)) to have the best overall performance. However, if the above mentioned feature is not important for the controlled process a designer can choose also the simpler scheme (32.(32.32.32)).

We use the configuration (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)) to illustrate the behavior of regularization parameter α as well as to show the impact of the smoothing parameters h, δ.

In Figure 3.37 the time development of the regularization parameter α is shown for the same simulation as is in Figure 3.35. α shows a characteristic

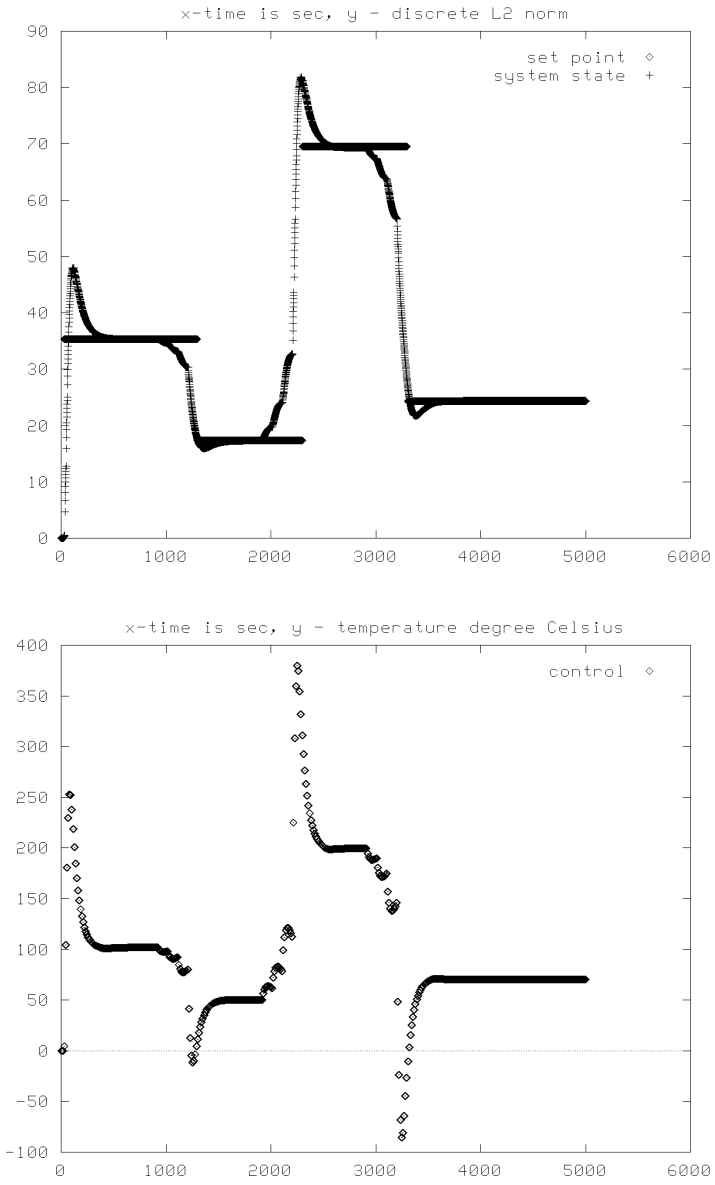


Figure 3.33: The control algorithm simulation with predictor structure (32.(8.8.8).(16.8.8).(24.8.8).(32.8.8)). The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

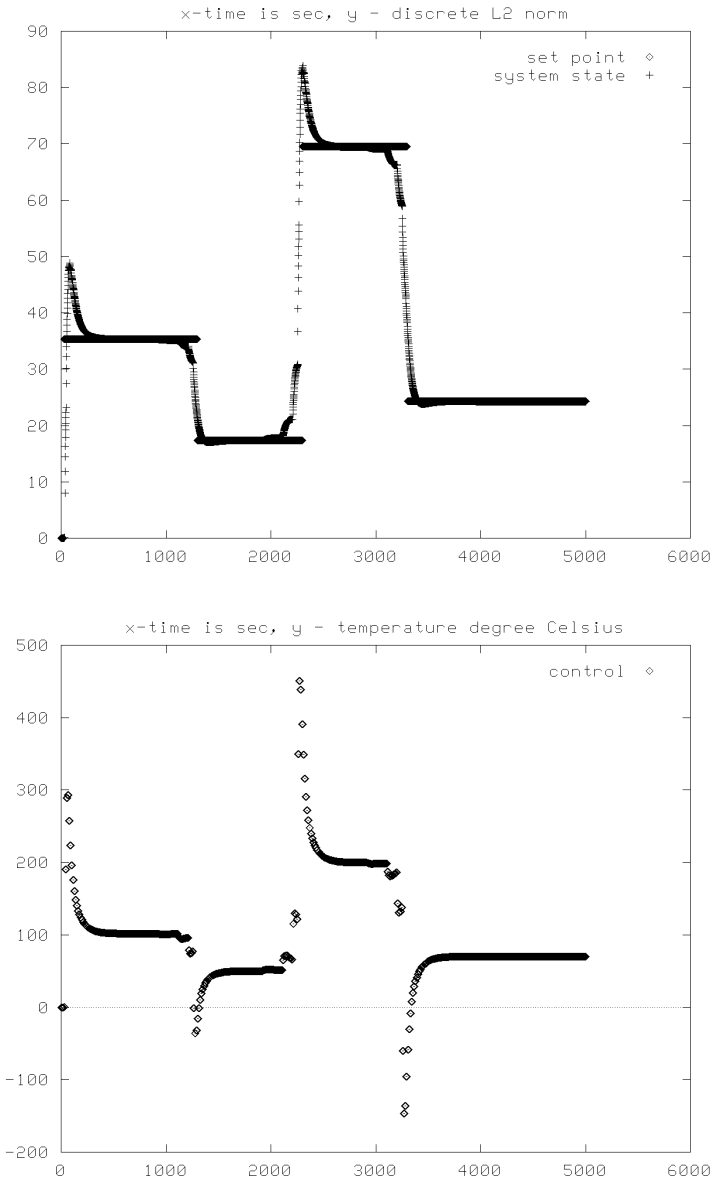


Figure 3.34: The control algorithm simulation with predictor structure (32.(4.8.8).(8.8.8).(16.8.8).(32.8.8)). The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

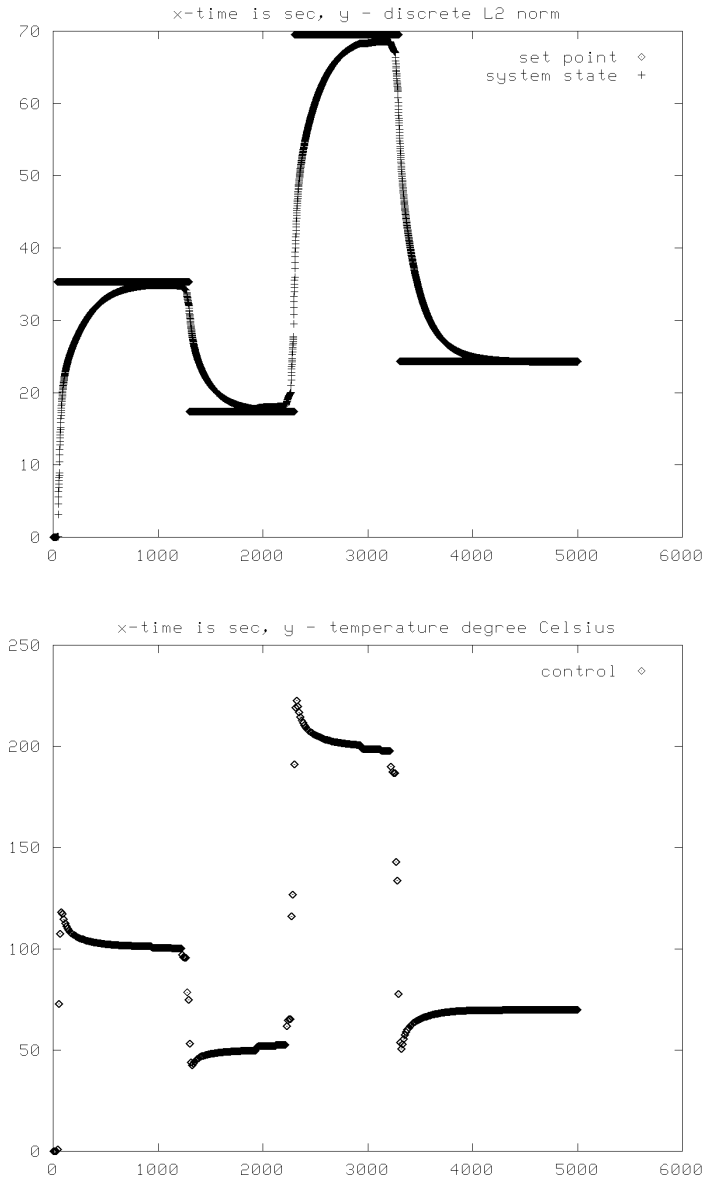


Figure 3.35: The control algorithm simulation with predictor structure (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)). The values of perturbation parameters are $h = 0.01, \delta = 0.1$. The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

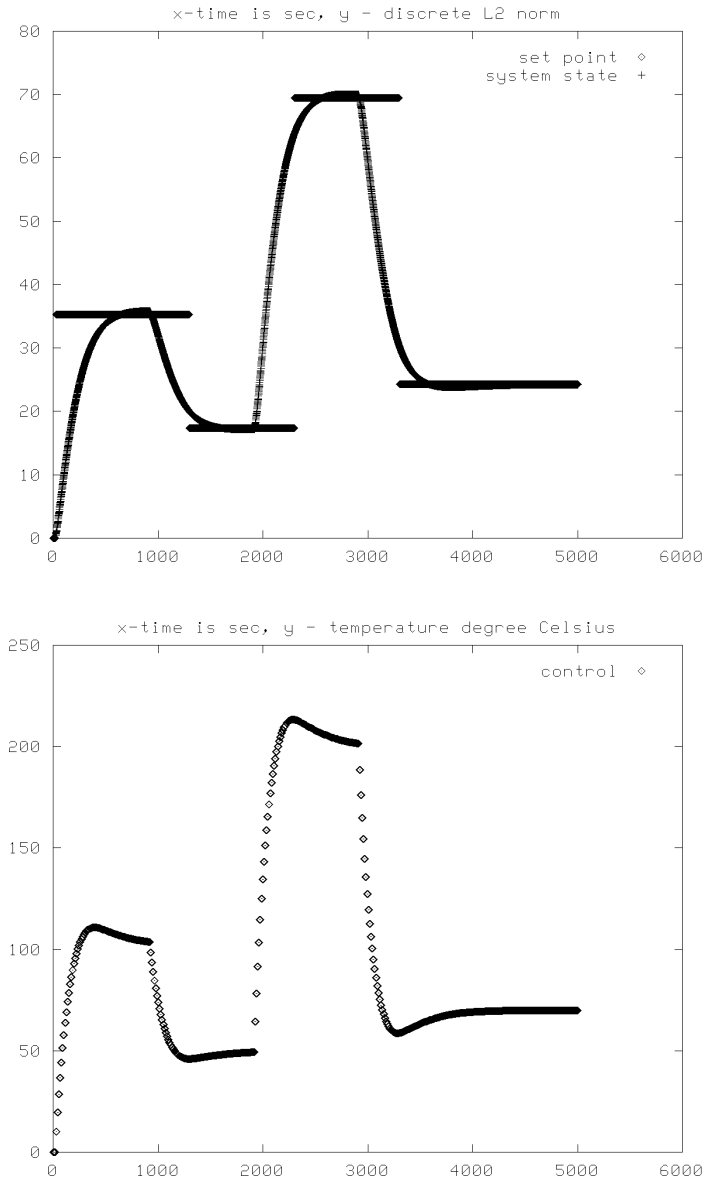


Figure 3.36: The control algorithm simulation with predictor structure (32.(32.32.32)). The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

pattern where during the transient states when the system is enough excited the value of α is very small which means that only a very small level of smoothing (regularization) is needed. When the system reaches the stable state the value of α grows significantly because the information coming from the right hand side of the equation approaches zero and the inversion needs much higher stabilizing factor. Finally, if there is no further disturbance the condition $|| u_\delta ||^2 \leq \delta^2 + \mu^2(u_\delta, A_h)$ from the Principle of generalized residuum is fulfilled leading to the zero value of control increments until a new load disturbance or noise will not provide enough large (in L_2 norm) right hand side for the inversion task.

In Figure 3.37, bottom picture, the number of generalized residuum evaluations needed for computation of the parameter α is shown as a function of time during the same simulation. As was mentioned in the previous section as far as the searching for root of generalized residuum can be completely done in canonic coordinates we prefer the simplest (but very robust) root searching method - namely, dividing the interval to halves. If, for example, regula falsi or some variation of Quasi-Newton method is used for the root searching of generalized residuum less evaluations are typically needed but during the transient states they can be divergent or too many iterations can occur.

In the previous section an interpretation was provided for the perturbation parameters h, δ. Decreasing the values of these parameters means in fact that smaller fraction of residuum value is at the disposal for smoothing. This fact is illustrated in Figure 3.38 where the same simulation as in Figure 3.35 is made with values $h = 0.001, \delta = 0.01$. On the other hand a better solution fit at the end of the prediction horizon is provided as will be seen below.

The last aspect which we want to discuss is related to the fact that incorporating the regularization into the heard of predictive algorithm we have a strong stabilizing factor. Therefore, we can try to stabilize the system in the *neighborhood* of unstabilizable state.

Let us consider a constant value reference signal along the spatial domain as an example for the system with a nonzero film coefficient $b \neq 0$. Denote this state **C**. An inspection of the modal system of ordinary differential equations (A.12) shows that there is no constant value of a boundary condition which can stabilize

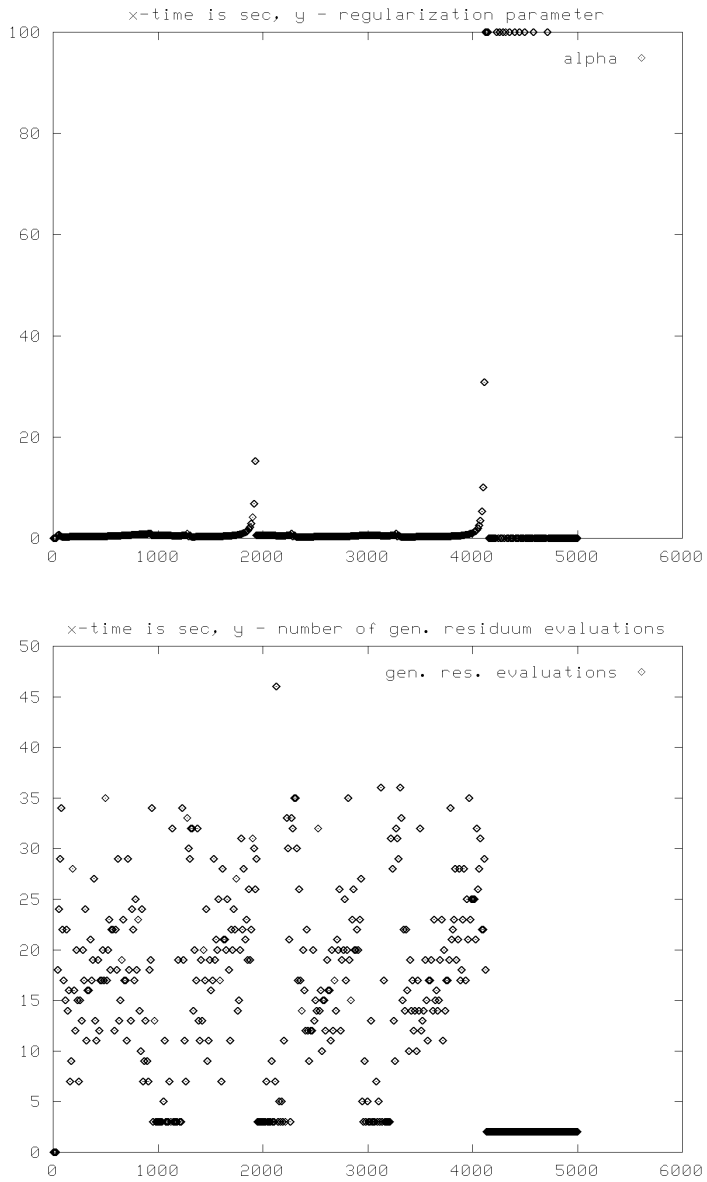


Figure 3.37: The time behavior of regularization parameter α during the simulation. In the lower picture, the number of generalized residuum evaluations is shown for the same simulation. For more details, see the text.

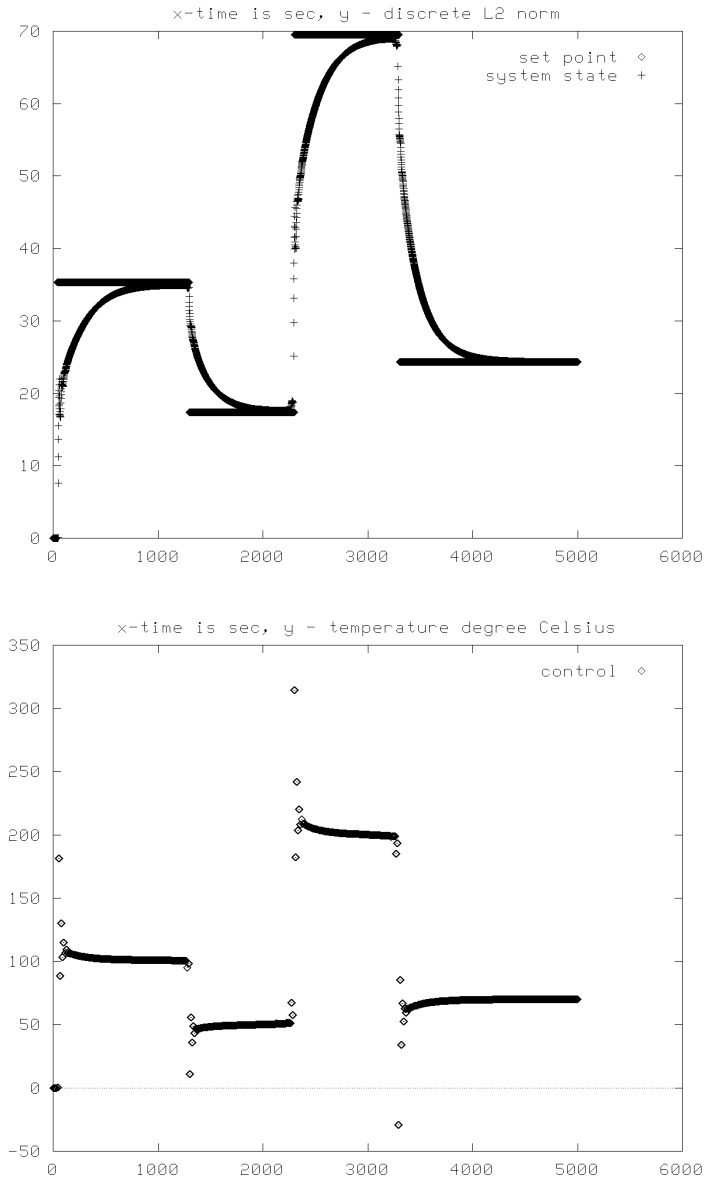


Figure 3.38: The control algorithm simulation with predictor structure (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)). The values of perturbation parameters are decreased to $h = 0.001, \delta = 0.01$.

the system at this state. However, we can experiment with RBPC algorithm to find a stable periodic orbit containing the desired state in its interior. Interior is meant in a special sense that it is a domain defined as a union of intervals obtained taking a maximal and a minimal temperature at certain spatial point during the whole orbit period of the system.

We suggest that such orbit must also contain a stable state of the system which has the same minimal temperature at the point $x = L$ as is the constant temperature of the desired state **C**. In fact it intuitively follows from the maximality principle for parabolic equation (see for example [52]).

If we take for example the state **C** with constant temperature 100 degrees, then for the values $a = 0.0095, b = 0.0020$ we obtain that the above orbit must also contain a stable state of the system with temperature 100 degrees at the point $x = L$. Denote this state **S**. It has a temperature 230 degrees at the boundary $x = 0$. Therefore we have that any stable orbit containing the state **C** must also contain the state **S**.

The situation is illustrated in Figure 3.39 where the orbit computed by the RBPC algorithm is shown in L_2 norm together with the corresponding controls. The algorithm provides almost the best possible orbit as far as the minimal temperature and maximal temperature on the boundary are few percent below and above the best possible values according the considerations in the preceding paragraph.

It must be stressed that to achieve the behavior in Figure 3.39 we must apply all controls computed for the prediction horizon k_h. That means in fact that the advantage of the receding horizon is lost and the algorithm is "blind" to disturbances during the whole prediction horizon time until the new sequence is computed at the beginning of a new horizon (this gives also a period of the orbit).

It seems that a sensitivity to disturbances can be gained back by a different approach based on the fact that a system on a stable orbit must see its *past states* on the horizon. Therefore, we can speculate about an algorithm which (i) uses a predictor to compute the sequence of controls which is then (ii) successively applied but (iii) at each step the sequence itself is *verified* by including past states as reference signals in the future and recomputing the sequence. If there

is no disturbance and the system is on the stable orbit we must obtain the same sequence. Otherwise the current sequence is invalidated and a new sequence computed. However, to put this note on a solid ground a further research would be necessary.

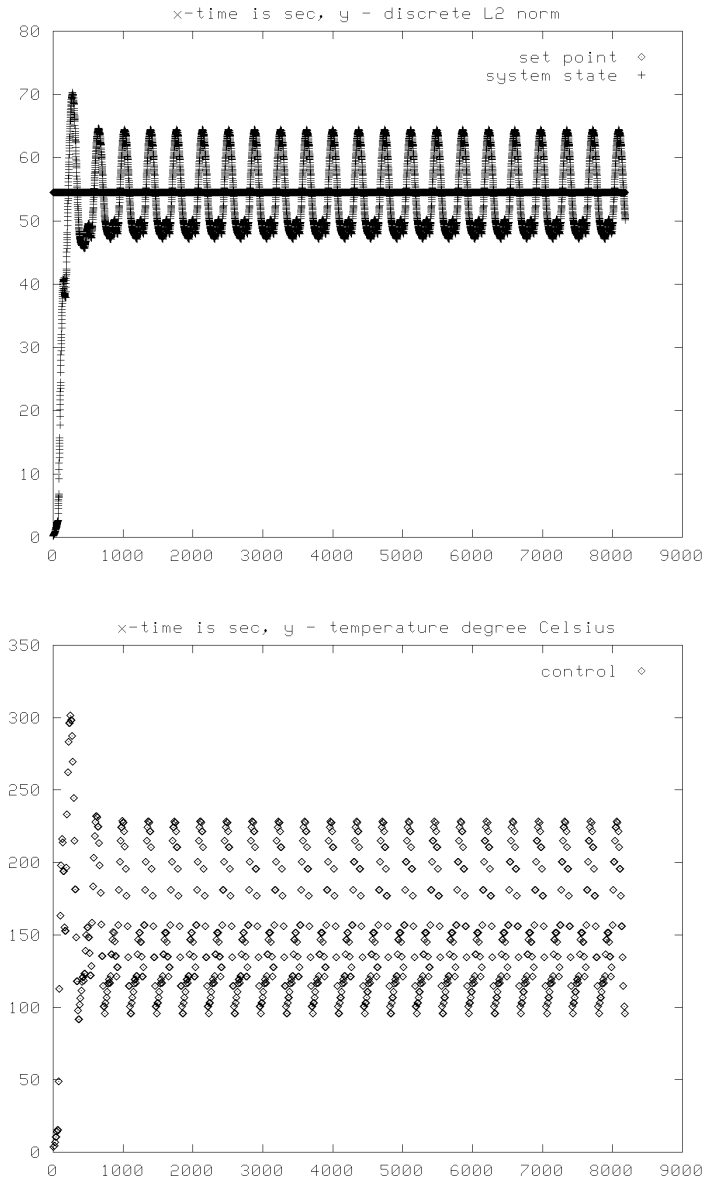


Figure 3.39: The stable orbit computed by RBPC algorithm.

Chapter 4

Conclusions

Before we start the closing discussion, the contributions of the present work can be summarized as follows.

- A new framework for model based predictive control is suggested which incorporates regularization directly within the receding horizon algorithm. It also gives a consistent interpretation of the weighting factor λ used in the quadratic criterion for GPC.
- A new model based predictive control algorithm (Regularization Based Predictive Control) for boundary control of distributed parameter systems is proposed. The efficient implementation of the algorithm is developed and the functionality of the algorithm is shown on computer simulations.
- An efficient method for computation of Green's functions is developed together with an analysis of the simulation accuracy near the heated boundary of the distributed parameter thermal system. This is our contribution to the simulation and control of distributed parameter systems which uses Green's functions for the system description.

Besides these main contributions we have also discussed

- The regularization and overall aspects of the step-wise spline-based control algorithm.

- The off-line identification procedure for the thermal diffusivity and film coefficients.

- The comparison of integral representation with Green's functions and the system description based on finite differences.

In the concluding discussion we would like to provide some notes focusing on four aspects, namely, (i) what can be concluded with respect to the comparison of integral representation and finite difference schemes, (ii) what the Regularization Based Predictive Control (RBPC) can tell to the already established model based predictive algorithms as for example GPC, (iii) what are possible directions of further research and (iv) how the preceding chapters are related to the technological processes in a reheating furnace.

Integral representation model versus finite differences There are three aspect which we want to consider in the comparison.

- A structural aspect, namely, to what extend the method provides a structural information and deeper insight into the problem.

- A computational aspect i.e. which of the methods is computationally more expensive.

- How the method can handle nonlinearities caused by temperature dependent thermal diffusivity and heat transfer coefficients.

Before we start this, it must be said that integral representation is heavily building on the explicit knowledge of Green's function or at least of the eigenfunctions of the related Sturm-Liuville problem. If these are not at the disposal the method becomes to be numerically infeasible.

As with the first aspect, it seems that the structural information provided by Green's functions i.e. by solution of related Sturm-Liuville is much deeper then using the finite differences.

For example given the discretization step in spatial domain we can estimate accurately the highest frequency necessary to get reasonable approximation at

the first point of spatial domain. Knowing the eigenfunction system (see the appendix A) we know that the solution of the evolution operator has a zero value at the boundary which means that the nearest maximum to zero is obtained at the point where the highest frequency has a maximum. Moreover using the fact that the system is exponentially smoothing (which also comes from this analysis) we can compute the highest frequency which can be represented given a computer precision. From this an appropriate discretization step can be estimated. In fact, this is a different regularization method which consists in the estimation of appropriate discretization step. This examination also suggests what would be the appropriate scaling which would allow to introduce higher frequencies: the space variable have to be scaled.

On the other hand, the integral representation is computationally slower which is caused by the fact that the matrices involved are dense (for more details see sections 3.2.1 and 3.2.3).

Also handling the nonlinearity seems to be more complicated using the integral representation. It can be handled in the same way as with finite differences, namely considering the values of the coefficients to be constant during some control periods, however it boils down to recomputation or updating of Green's functions which is computationally expensive for the same reasons as mentioned above i.e. because of dense matrices involved.

RBPC and MBPC - GPC There is one aspect which might be of interest also with relation to the "classical" well developed model based control approaches like GPC. The thing is that RBPC framework gives clear interpretation of the weighting factor λ introduced for example in GPC to handle the ill-conditioning of the matrices considered there. It can be objected that instead one design parameter λ we have now two, namely, h and δ. It is true, but both of them have clear interpretation and can be easily set. Then the value of smoothing parameter comes via well established regularization theory. Moreover, its values can be determined recursively so that the previous values can serve as a good approximation of the current value.

Further research As was already said in this work, we are still lacking two aspects: the incorporation of noise model and the recursive identification of thermal coefficients. We have done some steps to identify them off-line, however the online recursive identification is one of the subjects of further interest. This question is in fact in center of current research in the regularization theory where first sequential algorithms start to emerge [56].

We have also seen that the measurements must be improved if we want to have a more precise identification of thermal coefficients.

With respect to noise, this will definitively need further research to obtain a full equipped predictive control for partial differential equations.

There are extensions of Tikhonov regularization which can handle a case when there is no solution to the original equation and we want to keep this feature in a stable way under perturbations. This can be a case of constraints in some circumstances, however we leave this aspect for further investigation in future.

The basic strategy developed here is not constrained only to finite differences. If the system would be higher dimensional and the boundary is more complicated we can resort to finite elements with the same RBPC framework because the following Crank-Nicolson like semi-discrete formulation is possible [27] (in a weak formulation):

$$\left(\frac{u_h^n - u_h^{n-1}}{k_n}, v\right) + a\left(\frac{u_h^n + u_h^{n-1}}{2}, v\right) = \left(\frac{f(t_n) + f(t_{n-1})}{2}, v\right)$$

where u_h is the solution and $a(.,.)$ is the scalar product given by $a(u,v) = \int_\Omega (\nabla u.\nabla v + uv)\ dx$. The difference then boils down to a different and more complicated predictor.

Looking at the equation A.12 we can observe that the ordinary equations given there are in fact independent. That means a possibility to think about a sort of modal GPC control using the GPC control strategy in spectral domain. If this would be possible the full force of GPC method together with a noise model and recursive parameter estimation would be at the disposal for control of distributed parameter systems.

Reheating of rectangular slabs Finally, we return to the beginning of our work with few words about a possibility to provide an industrially feasible solution for control of reheating process.

The dimensionality reduction provided in RBPC allows to compute the main iteration of the control algorithm on vectors instead of matrices which effectively reduces the dimension by one. Therefore, the computational complexity does not restrict applications of the method developed in the previous sections in industrial conditions.

Bibliography

[1] Alifanov, O. M., 1988, *Obratnyje zadachi teploobmena [Inverse Heat Transfer Problems, in Russian]* (Mashinostroenie, Moscow)

[2] Alifanov, O. M., Artyukhin, E. A. and Rumyancev, C. V., 1988, *Ekstremalnyje metody reshenija nekorektnykh zadac, [Extrem methods of Ill-Posed Problem Solving, in Russian]* (Nauka, Moscow)

[3] Aris, R., 1975, The Methematical Theory of Diffusion and Reaction in Permeable Catalysts, *Clarendom Press, Oxford.*

[4] Bakushinskij, A. B. and Goncarckij, A. V., 1989, *Iterativnyje metody reshenija nekorektnykh zadach, [Iterative Methods for Ill-Posed Problem Solving, in Russian]* (Nauka, Moscow)

[5] Banks, S. P., 1984, State-space and frequency-domain methods in the control of distributed parameter systems, *Peter Peregrinus Ltd.*

[6] Banks, R. B., 1994, Growth and Diffusion Phenomena, *Springer-Verlag, Berlin.*

[7] Barataud, F., Moyne, Ch., Stemmelen, D., Identification of the Hydraulic Diffusivity of Two Porous Media: a Forestry Soil and Glass Beads, *Proc. of The Second International Conference on Inverse Problems in Engineering*, June 1996, LeCroisic, France.

[8] Beck, J. V., Blackwell, B. and St. Clair, C. R., Jr., 1985, *Inverse Heat Conduction: Ill-Posed Problems*, (Wiley-Interscience, NY)

[9] Beck, J. V., 1993, Comparison of the Iterative Regularization and Function Specification Algorithms for The Inverse Heat Conduction Problem, *Inverse Problems in Engineering, Theory and Practice, Proceedings of the First Conference in a Series on Inverse Problems in Engineering, Palm Coast, Florida* (Engineering Foundation by the American Society of Mechanical Engineers), pp. 23-30

[10] Bui, H. D., Tanaka M. et al, *Inverse Problems in Engineering Mechanics, Proceedings of the Second International Symposium on Inverse Problems - ISIP'94, Paris, France*, 1994, (A.A. Balkema, Rotterdam, Brookfield)

[11] Butkovskij, A. G., 1975, Metody upravlenija sistemami s raspredelonnymi parametrami. *Nauka, Moskva 1975*

[12] Butkovskiy, A.G., Structural theory of distributed parameter systems, *Elis Horwood Limited*, UK, (1983).

[13] Clarke, D. W., Mohtadi, C., Tuffs, P. S., 1987, Generalised Predictive Control,*Automatica*, Vol. 23, pp. 137-160.

[14] Clarke, D. W., 1988, Application of Generalised Predictive Control to Industrial Processes,*IEEE Control Systems Magazine*, Vol. 8, pp. 49-55.

[15] Curtain, R. F., Pritchard, A. J., 1978, Infinite dimensional linear systems theory, *Springer-Verlag, Berlin.*

[16] Curtain, R. F., 1982, Finite-Dimensional Compensator Design for Parabolic Distributed Systems with Point Sensors and Boundary Input, IEEE Transactions on Automatic Control, Vol. AC-27, No.1, pp. 98-104.

[17] Curtain, R. F., 1984, Finite Dimensional Compensators for Parabolic Distributed Systems with Unbounded Control and Observations, SIAM J. Control and Optimization, Vol. 22, No.2, pp. 255-276.

[18] Cutler, C. R., Ramaker, B. L., 1980, Dynamic Matrix Control: a Computer Control Algorithm, in *American Control Conference*, San Francisco 1980, USA.

[19] Gell-Mann, M., 1990, Visions of Sustainable Word, *Santa Fe Institute, Research Report No. 90-021.*

[20] Glasko, V. B., Zacharov, A. V., Iljin, M. E., Noveshenko, J. A., Tichonov, A. N., 1984, O nekotorych zadachah optimizaciji kvaziravnomernogo nagreva krupnogabaritnych detalej, *Zhurnal vychislitelnoj matematiki i matemati cheskoj fiziki,* Vol. 24, pp. 686-693.

[21] Heath, M. T., 1974, *The Numerical Solution of Ill-Conditioned Systems of Linear Equations*, (Oak Ridge National Laboratory, Union Carbide Corporation)

[22] Hillis, W. D., 1985, The Connection Machine. *The MIT Press series in artificial intelligence.*

[23] Iwahashi, Y., Takanashi, K., Fujii, S., Mitsuoka, H., 1981, Computer Control System for Continuous Reheating Furnaces, *Preprints of IFAC 8th World Congress*, Vol. 18, pp. 147-152.

[24] Jankov, S., 1992, *Imagerie Tomographique d'Étoiles de Type Solaire, Thèse de Doctorat d'Astrophysique et Techniques Spatiales de l'Université Paris VII*, (Service Impression IPN - ORSAY)

[25] Jarny, Y., Delaunay, D., Bailleul J. L., The Curing Process of Composite Materials Inverse Analysis of the Heating Conditions, *Proc. of The Second International Conference on Inverse Problems in Engineering*, June 1996, LeCroisic, France.

[26] Johnson, R. W. and Shore, J. E., 1983, *Comments on and Correction to "Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy"* , (IEEE Transactions on Information Theory), **vol. 29**, 6, pp. 942-943

[27] Johnson, C., 1987, *Numerical solutions of partial differential equations by the finite element method*, (Cambridge University Press)

[28] Krishnamurthy, E. V., 1989, Parallel Processing, Principles and Practise, *Addison-Wesley, Sydney.*

[29] Lair, P., Dumoulin, J., Millan, P., Fluxes and temperature fields characterization using infrared thermography in die forging, *Proc. of The Second International Conference on Inverse Problems in Engineering*, June 1996, LeCroisic, France.

[30] Lions, J.-L., 1971, Optimal Control of Systems Governed by Partial Differential Equations, *Springer Verlag, Berlin, Heidelberg, New York 1971*

[31] Matsevity, J. M. and Lushpenko, C. F., 1990, *Identifikacija teplofizicheskikh svojstv tvjordykh tel, [Identification of Thermophysical Properties of Solid Bodies, in Russian]* (Naukova dumka, Kiew)

[32] Mitchel, A. R., 1969, Computational Methods in Partial Differential Equations, (J. Wiley & Sons).

[33] Morozov, V. A., 1984, *Methods of Solving Incorrectly Posed Problems* (Springer-Verlag, New York)

[34] Morozov, V. A., 1987, *Metody regulyarizacii neustojchivykh zadach, [Methods for Regularization of Unstable Problems, in Russian]* (Izdavatelstvo Moskovskovo Universiteta, Moscow)

[35] Morozov, V. A., 1987, *Reguljarnyje metody resenija nekorektno postavljennych zadac, [Regularization Methods for Ill-Posed Problem Solving, in Russian]* (Nauka, Moscow)

[36] Guo, L. and Murio, D. A.,1991, A mollified space-marching finite difference algorithm for the two-dimesional inverse heat conduction problem with slab symmetry, *Inverse Problems*, **vol. 7**, pp. 247-259

[37] Okubo, A., 1980, Diffusion and Ecological Problems: Mathematical Models, *Springer-Verlag, Berlin.*

[38] Quinn, M. J., 1987, Designing Efficient Algorithms for Parallel Computers, *McGraw-Hill, New York.*

[39] Richalet, J., Rault, A., Testud, J. L. and Papon, J., 1978, Model Predictive Heuristic Control: Application to Industrial Processes, *Automatica*, Vol. 14, pp. 413-428.

[40] Roháľ-Ilkiv, B., Országhová, Z., and Hrúz, T., A Stepwise Technique for Inverse Problem in Optimal Boundary Control of Thermal Systems, *Proc. First International Conference on Inverse Problems in Engineering*, June 1993, in "Inverse Problems in Engineering: Theory and Practice" Zabaras, Woodbury, Raynaud, editors, ASME Book No. I00357, pp. 147-154.

[41] Hrúz, T., Roháľ-Ilkiv, B. and Országhová, Z., An Optimal Boundary Control of Thermal System with Experimental Verification, *Proc. of The Second International Conference on Inverse Problems in Engineering*, June 1996, LeCroisic, France.

[42] Roháľ-Ilkiv, B., Hrúz, T., Hammel, P., 1994, An Approach to the Problem of Boundary Control of Thermal Systems, in *Proceedings of International Workshop on Advanced Education in Aut. and Control Technology*, Prague 1994.

[43] Zelinka,P., Richter, R., Roháľ-Ilkiv, B., Sroka, I., Országhová, Z., 1993, Experimentálna implementácia adaptívneho riadenia tepelného režimu karuselovej pece, *Automatizace*, Vol. 36, No. 5, pp. 130-134.

[44] Saliga, V., Darienko, V., Obruchev, V., 1986, Design of Heting Control Algorithms by Solving an Inverse Dynamics Problem, *IFAC Stochastic Control, Vilnius, USSR, 1986*

[45] Scarpa, F., Milano, G. and Pescetti, D., 1993, Thermophysical Properties Estimation from Transient Data: Kalman Versus Gauss Approach, *Inverse Problems in Engineering, Theory and Practice, Proceedings of the First Conference in a Series on Inverse Problems in Engineering, Palm Coast, Florida*

(Engineering Foundation by the American Society of Mechanical Engineers), pp. 109-116

[46] Shore, J. E. and Johnson, R. W., 1980, Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy, *IEEE Transactions on Information Theory*, **vol. 26**, 1, pp. 26-37

[47] Tarantola, A., 1987, *Inverse Problem Theory, Methods for Data Fitting and Model Parameter Estimation*, (Elsevier)

[48] Tikhonov, A. N. and Arsenin, V. J., 1977, *Solution of Ill-Posed Problems* (V. H. Winston, Washington, D.C.)

[49] Tikhonov, A. N. and Arsenin, V. J., 1979, *Metody reshenija nekorektnykh zadac, [Methods for Ill-Posed Problem Solving, in Russian]* (Nauka, Moscow)

[50] Tikhonov, A. N., Goncsarskij, A. V., Stepanov, V. V. and Jagola, A. G., 1990, *Chislennyje metody reshenija nekorrektnykh zadach, [Numerical Methods for Ill-Posed Problem Solving, in Russian]* (Nauka, Moscow)

[51] Tikhonov, A. N., Goncsarskij, A. V., Stepanov, V. V. and Jagola, A. G., 1995, *Numerical Methods for the Solution of Ill-Posed Problems*, (Kluwer Academic Publisher)

[52] Tikhonov, A. N. and Samarskij, A. A., 1955, *Rovnice matematicke fysiky, Equations of Mathematical Physics, in Czech, translation from Russian]*, (CSAV Praha).

[53] Vajteršic, M., 1988, Moderné algoritmy na riešenie niektorých eliptických parciálnych diferenciálnych rovníc, *Veda, SAV, Bratislava*.

[54] Vitasek, E., 1987, *Numericke Metody*, (SNTL Praha).

[55] Widrow, B., 1987, Adaptive Inverse Control, in *2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, Sweden, Pergamon Press, Oxford*

[56] Zabaras, N. and Badrinarayanan, S. 1993, Inverse Problems and Techniques in Metal Forming Processes, *Inverse Problems in Engineering, Theory and Practice, Proceedings of the First Conference in a Series on Inverse Problems in Engineering, Palm Coast, Florida* (Engineering Foundation by the American Society of Mechanical Engineers), pp. 65-76

[57] Zabaras, N., Ruan, Y. and Richmond, O., 1992, Design of Two-Dimensional Stefan Processes with Desired Freezing Front Motions, *Numerical Heat Transfer, Part B*, (Hemisphere Publishing Corporation), **vol. 21**, pp. 307-325

[58] *Ill-posed Problems, The International Conference, Moscow*, 1991, (Keldysh Inst. of Appl. Math)

[59] *Obratnyje zadachi dlja matematicheskikh modelej fizicheskikh processov, Sbornik nauchnykh trudov, [Inverse Problems for Mathematical Models of Physical Processes, in Russian]*, 1991, (Moscow Inistitute of Engineering and Physics)

Appendix A

1-D Heat Conduction

In the following appendix we derive a solution of the following heat conduction and heat transfer equation:

$$\begin{gathered} \frac{\partial}{\partial t}Y(x,t) - a^2\frac{\partial^2}{\partial x^2}Y(x,t) + bY(x,t) = 0 \\ Y(x,0) = Y_0(x), Y(0,t) = y(t), \frac{\partial Y(L,t)}{\partial x} = 0 \\ 0 \leq x \leq L, \quad t \geq t_0, \quad a \neq 0 \end{gathered} \tag{A.1}$$

where: L is the length of the bar; a, b contain the heat conduction and the heat transfer coefficients (for more details about the model see section 3).

The solution follows the well-known principles of separation of variables. However, it is good to have at hand a tailor-made derivation for the particular boundary conditions as well as for the particular form of the basic equation.

We also suppose that the boundary and initial conditions are associated, i.e.

$$Y_0(0) = y(0), \frac{\partial Y_0(L)}{\partial x} = 0 \tag{A.2}$$

It is well known (see for example [52]) that the above conditions can be relaxed in the case of integral representation which we will derive later.

First of all, we make a substitution to the equation (A.1) to simplify the boundary conditions. $Y(x,t)$ is substituted with

$$Y(x,t) = v(x,t) + u(x,t) \tag{A.3}$$

where $u(x,t) = y(t)$. Then the function $v(x,t)$ conforms to the simplified boundary conditions:

$$\begin{aligned} v(0,t) &= Y(x,t) - u(x,t) = y(t) - y(t) = 0 \\ \frac{\partial v}{\partial x}(L,t) &= \frac{\partial Y}{\partial x}(L,t) - \frac{\partial u}{\partial x}(L,t) = 0 - 0 = 0 \end{aligned} \tag{A.4}$$

The initial condition is

$$v(x,0) = v_0(x) = Y_0(x) - y(0). \tag{A.5}$$

Substituting $Y(x,t)$ into the basic equation (A.1) we obtain

$$\begin{aligned} \frac{\partial v}{\partial t} + \frac{\partial u}{\partial t} - a^2\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 u}{\partial x^2}\right) + b(v+u) &= 0 \\ \frac{\partial v}{\partial t} - a^2\frac{\partial^2 v}{\partial x^2} + bv &= \underbrace{-y'(t) - by(t)}_{f(x,t)}. \end{aligned} \tag{A.6}$$

This is an equation with a homogeneous boundary conditions but with a nonzero right hand side. To solve it, applying the superposition principle, we can proceed by solving first the homogeneous equation with a nonzero initial condition

$$\begin{aligned} \frac{\partial v}{\partial t} - a^2\frac{\partial^2 v}{\partial x^2} + bv &= 0 \\ v(x,0) = v_0(x), \quad v(0,t) = 0, \quad \frac{\partial v}{\partial x}(L,t) &= 0 \end{aligned} \tag{A.7}$$

and then solving the non-homogeneous equation with the zero initial condition

$$\begin{aligned} \frac{\partial v}{\partial t} - a^2\frac{\partial^2 v}{\partial x^2} + bv &= -y'(t) - by(t) \\ v(x,0) = 0, \quad v(0,t) = 0, \quad \frac{\partial v}{\partial x}(L,t) &= 0. \end{aligned} \tag{A.8}$$

The final solution of (A.6) is a superposition of the solutions of (A.7) and (A.8). This is substituted to (A.3) to obtain $Y(x,t)$.

Now, following the principle of separation of variables for equation (A.7), we are looking for a nontrivial solution of (A.7) which can be written in the form $v(x,t) = X(x)T(t)$.

$$\begin{aligned} T'X - a^2X''T + bXT &= 0 \\ \frac{T'}{T} - a^2\frac{X''}{X} + b &= 0 \\ \frac{T'}{T} + b = \frac{a^2X''}{X} &= -\lambda \end{aligned}$$

Now to find the nontrivial solution of the Sturm-Liouville problem $\frac{a^2X''}{X} = -\lambda$ we can briefly derive: $X'' + \frac{\lambda}{a^2}X = 0$, (char. eq.) $r^2 + \frac{\lambda}{a^2} = 0$, (Case 1.) $\lambda = 0$, $X'' = 0$, $X(x) = c_1x + c_2$, $X(0) = c_2 = 0$, $X'(L) = c_1 = 0$, (Case 2.) $\lambda < 0$, $r^2 = -\frac{\lambda}{a^2}$, $r = \pm\frac{\sqrt{-\lambda}}{a}$, $X(x) = c_1e^{\frac{\sqrt{-\lambda}}{a}x} + c_2e^{-\frac{\sqrt{-\lambda}}{a}x}$, $X(0) = c_1 + c_2 = 0 \Rightarrow c_1 = -c_2$, $X'(L) = c_1\frac{\sqrt{-\lambda}}{a}e^{\frac{\sqrt{-\lambda}}{a}L} - c_2\frac{\sqrt{-\lambda}}{a}e^{-\frac{\sqrt{-\lambda}}{a}L} = 0$, $c_1\frac{\sqrt{-\lambda}}{a}(e^{\frac{\sqrt{-\lambda}}{a}L} + \frac{1}{e^{\frac{\sqrt{-\lambda}}{a}L}}) = 0 \Rightarrow$, $c_1 = 0 \Rightarrow c_2 = 0$, (Case 3.) $\lambda > 0$, $r^2 + \frac{\lambda}{a^2} = 0$, $r^2 = -\frac{\lambda}{a^2}$, $r = \pm\frac{i\sqrt{\lambda}}{a}$, $X(x) = c_1\sin\frac{\sqrt{\lambda}}{a}x + c_2\cos\frac{\sqrt{\lambda}}{a}x$, $X(0) = c_2\cos\frac{\sqrt{\lambda}}{a}0 = 0 \Rightarrow c_2 = 0$, $X'(L) = c_1\frac{\sqrt{\lambda}}{a}\cos\frac{\sqrt{\lambda}}{a}L = 0$, $\frac{\sqrt{\lambda}}{a}L = (2k+1)\frac{\pi}{2}$, $k = 0, 1, \ldots$, $\sqrt{\lambda} = \frac{(2k+1)\pi a}{2L}$, $\lambda_k = \frac{(2k+1)^2\pi^2a^2}{4L^2}$, $\varphi_k(x) = \sin\frac{\sqrt{\lambda_k}}{a}x = \sin(2k+1)\frac{\pi x}{2L}$.

Similarly, for the eigenvalues of the above Sturm-Liouville problem, we can solve the time dependent part: $T' + (b + \lambda_k)T = 0$, $r + (b + \lambda_k) = 0$, $r = -b - \lambda_k$, $T(t) = c_ke^{(-b-\lambda_k)t}$.

Let us create formaly a series for the general solution of (A.7):

$$v(x,t) = \sum_{k=0}^{\infty} c_ke^{(-b-\lambda_k)t}\sin\frac{\sqrt{\lambda_k}}{a}x$$

for the initial condition must hold

$$v(x,0) = v_0(x) = \sum_{k=0}^{\infty} c_k\sin\frac{\sqrt{\lambda_k}}{a}x$$

where $c_k = \frac{2}{L}\int_0^L v_0(\xi)\varphi_k(\xi)\,d\xi$ are the Fourier coefficients for the development of function $v_0(x)$.

Finally, substituting c_k to the above sum for $v(x,t)$

$$v(x,t) = \frac{2}{L}\sum_{k=0}^{\infty}\left\{\int_0^L v_0(\xi)\varphi_k(\xi)\,d\xi\right\} e^{(-b-\lambda_k)t}\varphi_k(x) \tag{A.9}$$

and exchanging the order of integration and sumation (for the correctness of the steps see [52]) we obtain

$$v(x,t) = \int_0^L \underbrace{\left\{\frac{2}{L}\sum_{k=0}^{\infty} e^{(-b-\lambda_k)t}\varphi_k(x)\varphi_k(\xi)\right\}}_{G(x,\xi,t)} v_0(\xi)\,d\xi =$$

$$= \int_0^L G(x,\xi,t)\,v_0(\xi)\,d\xi \tag{A.10}$$

where $G(x,\xi,t)$ is a Green's function of the appropriate problem.

The second step is to solve equation (A.8). Now, the initial as well as the boundary conditions are zero. The solution is supposed to be in the form of a Fourier series

$$v(x,t) = \sum_{k=0}^{\infty} v_k(t)\varphi_k(x) \tag{A.11}$$

where time t is treated as a parameter. To find the coefficients $v_k(t)$ also the right hand side of the equation is expressed in the form of a Fourier series

$$f(x,t) = f(t) = -y'(t) - by(t) = \sum_{k=0}^{\infty} f_k(t)\varphi_k(x)$$

with the Fourier coefficients as follows

$$f_k(t) = \frac{2}{L}\int_0^L f(t)\varphi_k(\xi)\,d\xi = \frac{2}{L}f(t)\int_0^L \varphi_k(\xi)\,d\xi =$$

$$= \frac{2}{L}\{y'(t) + by(t)\}\left[\cos\frac{\sqrt{\lambda_k}}{a}x\right]_0^L \frac{a}{\sqrt{\lambda_k}} = -\frac{4}{(2k+1)\pi}\{y'(t) + by(t)\}$$

The Fourier series of the solution together with the series for the right hand side are substituted into equation (A.8).

$$\sum_{k=0}^{\infty}\varphi_k(x)\left\{v_k'(t) + v_k(t)\lambda_k + bv_k(t) - f_k(t)\right\} = 0$$

The result is a system of ordinary differential equations for the coefficients $v_k(t)$

$$\begin{aligned} v_k'(t) + v_k(t)(\lambda_k + b) &= f_k(t) \\ v_k'(t) &= -(\lambda_k + b)v_k(t) + f_k(t) \end{aligned} \tag{A.12}$$

Because the initial condition is zero, we obtain

$$v(x,0) = \sum_{k=0}^{\infty} v_k(0)\varphi_k(x) = 0 \Rightarrow v_k(0) = 0$$

The characteristic equations for the homogeneous version of the above system are $r + \lambda_k + b = 0$. When we take to the account the right hand side and the zero initial conditions then the solution of the system can be written as

$$v_k(t) = \int_0^t e^{-(\lambda_k+b)(t-\tau)} f_k(\tau)\, d\tau \tag{A.13}$$

The expression for the Fourier coefficients $f_k(t)$ is substituted to $v_k(t)$ and the coefficients $v_k(t)$ are substituted to the solution (A.11). We obtain in general

$$\begin{aligned} v(x,t) &= \sum_{k=0}^{\infty} \left\{ \int_0^t e^{-(\lambda_k+b)(t-\tau)} \left\{ \frac{2}{L} \int_0^L f(\tau,\xi)\varphi_k(\xi)\, d\xi \right\} d\tau \right\} \varphi_k(x) \\ v(x,t) &= \int_0^t \int_0^L \left\{ \frac{2}{L} \sum_{k=0}^{\infty} e^{-(\lambda_k+b)(t-\tau)} \varphi_k(x)\varphi_k(\xi) \right\} f(\tau,\xi)\, d\xi\, d\tau \\ v(x,t) &= \int_0^t \int_0^L G(x,\xi,t-\tau) f(\tau,\xi)\, d\xi\, d\tau \end{aligned} \tag{A.14}$$

where $G(x,\xi,t-\tau)$ is the Green's function in the same form as in the final expression for the initial condition part of the solution (A.7).

What was derived above in general, can be specialized for the particular case of the problem A.1. The special form of Fourier coefficients of f, $f_k(t) = -\frac{4}{(2k+1)\pi}\{y'(t) + by(t)\}$ is substituted.

$$\begin{aligned} v(x,t) &= \sum_{k=0}^{\infty} \left\{ \int_0^t e^{-(\lambda_k+b)(t-\tau)} \left\{ -\frac{4}{(2k+1)\pi} (y'(\tau) + by(\tau)) \right\} d\tau \right\} \varphi_k(x) \\ v(x,t) &= \int_0^t \underbrace{\left\{ \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{1}{(2k+1)} e^{-(\lambda_k+b)(t-\tau)} \varphi_k(x) \right\}}_{G_1(x,t-\tau)} (-y'(\tau) - by(\tau))\, d\tau \end{aligned}$$

$$v(x,t) = \int_0^t G_1(x, t-\tau)(-y'(\tau) - by(\tau))\, d\tau \tag{A.15}$$

One possibility is to stop here and use the above form for system simulation. Another possibility is to proceed with intergration by parts to get rid of the derivative of y.

$$v(x,t) = \int_0^t G_1(x, t-\tau)(-y'(\tau) - by(\tau))\, d\tau$$

$$v(x,t) = -\int_0^t G_1(x, t-\tau)y'(\tau)\, d\tau - b\int_0^t G_1(x, t-\tau)y(\tau))\, d\tau$$

$$v(x,t) = -\left\{G_1(x,0)y(t) - G_1(x,t)y(0) - \int_0^t \frac{\partial}{\partial\tau}G_1(x, t-\tau)y(\tau))\, d\tau\right\} -$$
$$-b\int_0^t G_1(x, t-\tau)y(\tau))\, d\tau$$

$$v(x,t) = -G_1(x,0)y(t) + G_1(x,t)y(0) +$$
$$+\int_0^t \underbrace{\left\{\frac{\partial}{\partial\tau}G_1(x, t-\tau) - bG_1(x, t-\tau)\right\}}_{G_2(x,t-\tau)} y(\tau))\, d\tau$$

$$v(x,t) = -G_1(x,0)y(t) + G_1(x,t)y(0) + \int_0^t G_2(x, t-\tau)y(\tau))\, d\tau$$

A straightforward derivation shows that

$$G_2(x,t) = \frac{\pi a^2}{L^2}\sum_{k=0}^{\infty}(2k+1)e^{-(\lambda_k+b)t}\varphi_k(x) = a^2 G_{\partial\xi}(x,t) = \frac{\partial}{\partial\xi}G(x,\xi,t)\mid_{\xi=0}.$$

Therefore the final expression for $v(x,t)$ can be written as

$$v(x,t) = -G_1(x,0)y(t) + G_1(x,t)y(0) + a^2\int_0^t G_{\partial\xi}(x, t-\tau)y(\tau))\, d\tau. \tag{A.16}$$

The solution $Y(x,t)$ of the original problem (A.1) is obtained by the superposition of (A.10), (A.16) and $u(x,t)$

$$Y(x,t) = y(t) - G_1(x,0)y(t) + G_1(x,t)y(0) + a^2\int_0^t G_{\partial\xi}(x, t-\tau)y(\tau))\, d\tau +$$
$$+\int_0^L G(x,\xi,t)\underbrace{(Y_0(\xi) - y(0))}_{v_0(\xi)}\, d\xi. \tag{A.17}$$

Taking into the account that $\int_0^L G(x,\xi,t)\,d\xi = G_1(x,t)$ we obtain

$$Y(x,t) = y(t) - G_1(x,0)y(t) + a^2 \int_0^t G_{\partial\xi}(x,t-\tau)y(\tau))\,d\tau + \\ + \int_0^L G(x,\xi,t)\,Y_0(\xi)\,d\xi. \tag{A.18}$$

From the basic theory of Fourier series it follows that $G_1(x,0) = 0$ for $x = 0$ and $G_1(x,0) = 1$ for $0 < x \leq \frac{\pi}{2}$. This fact is also responsible for the correctness of the above expression for the boundary $x = 0$ of the spatial domain $< 0, L >$.

We have obtained two possibilities how to simulate the system response to boundary condition based on the kernel $G_1(x,t)$ or on the kernel $G_{\partial\xi}(x,t)$. The differences between them are discussed in the section 3.2.1.2.

Appendix B

Example of RBPC C code

In this appendix we include an example of a driver which reads a predictor definition from a file. Then the RBPC main control procedures together with a regularization are listed.

The files have the following names:

- driver-control-gen.c - driver.
- control.c, control.h - RBPC main upper level procedures. procedures.
- 8.1-2-4-8.1-3-7-15 -example of a predictor definition.