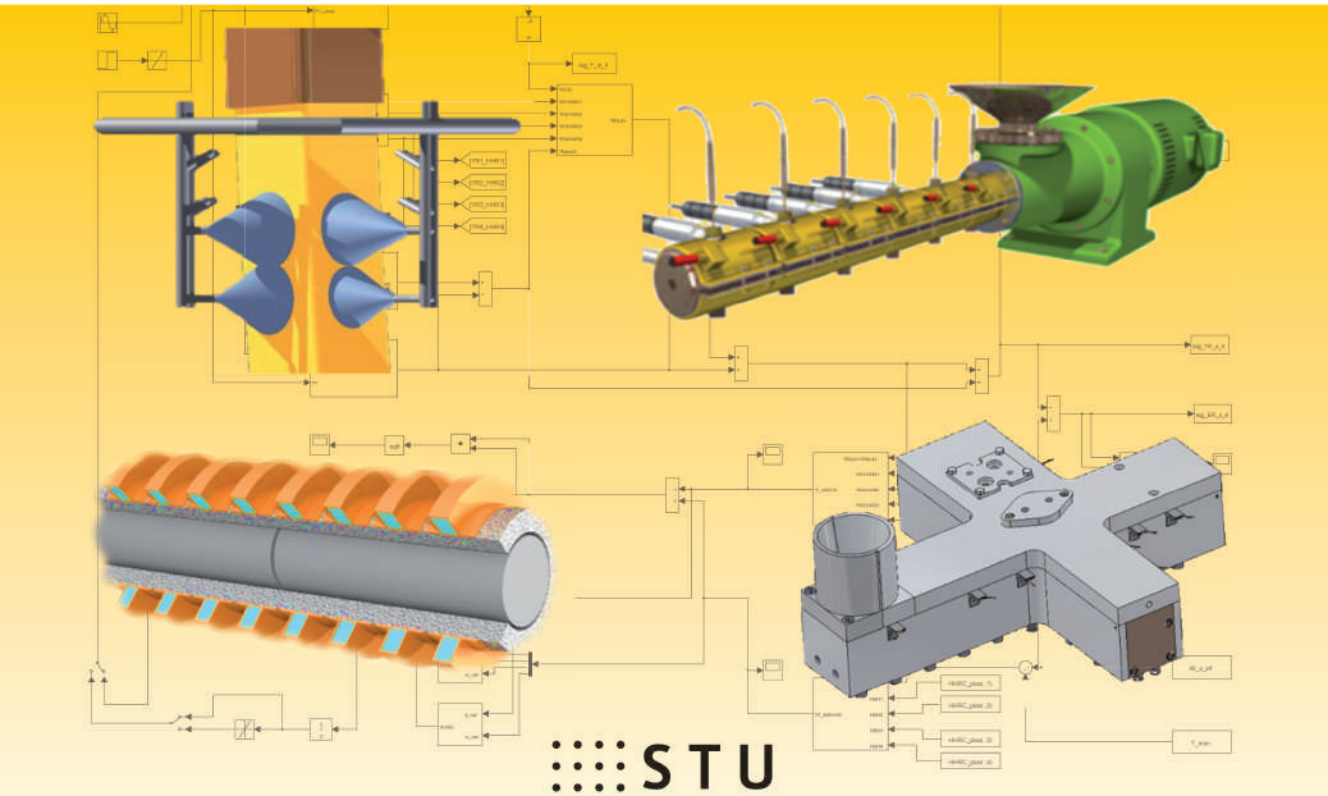
Gabriel Hulkó - Cyril Belavý - Pavol Noga - Karol Ondrejkovič

CONTROL OF TECHNOLOGICAL AND PRODUCTION PROCESSES AS DISTRIBUTED PARAMETER SYSTEMS

SUPPORTED BY ADVANCED NUMERICAL MODELING



STU

CONTROL OF TECHNOLOGICAL AND PRODUCTION PROCESSES AS DISTRIBUTED PARAMETER SYSTEMS

SUPPORTED BY ADVANCED NUMERICAL MODELING

Gabriel Hulkó - Cyril Belavý - Pavol Noga - Karol Ondrejkovič

CONTROL OF TECHNOLOGICAL AND PRODUCTION PROCESSES AS DISTRIBUTED PARAMETER SYSTEMS

SUPPORTED BY ADVANCED NUMERICAL MODELING

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
2014

Advanced numerical models, built for the computer analysis of the dynamics of various technological and production processes offer time and space dependent dynamic characteristics of complex physical phenomena. These numerical models are also suitable for the design of lumped input and distributed parameter output systems. Control tasks are solved for technological and production processes as distributed parameter systems in the linearized neighborhood of steady-state operating modes and between these. A development tool is proposed for the design and optimization of control systems based on cosimulation–meaning a collaboration of Matlab & Simulink and different software products built for dynamical analysis of controlled technological and production processes. The book demonstrates this potential throughout various benchmark control problems in the field of casting, induction heating, extrusion of plastics, continuous casting of steel and groundwater remediation. The book is recommended for students, graduates of technical universities and researchers involved with the research and development of the control of various technological and production processes based on numerical simulations and engineers with an interest in the control of distributed parameter systems.

All rights reserved. No part of this publication may be reproduced in any form or by any means without either the prior written permission of authors or publisher.

© prof. Ing. Gabriel Hulkó, DrSc. – prof. Ing. Cyril Belavý, CSc.
Ing. Pavol Noga, PhD. – Ing. Karol Ondrejkovič, PhD.

Reviewers: Dr. h. c. prof. Ing. Anton Víteček, CSc.
prof. Ing. Mikuláš Alexík, PhD.
prof. Ing. Juraj Spalek, PhD.

Approved by scientific board at Faculty of Mechanical Engineering in Bratislava.

ISBN 978-80-227-4290-0

Authors:

Gabriel Hulkó
Professor at Slovak University of Technology in Bratislava
Institute of Automation, Measurement and Applied Informatics
Center for Control of Distributed Parameter Systems
Faculty of Mechanical Engineering
gabriel.hulko@stuba.sk

Cyril Belavý
Professor at Slovak University of Technology in Bratislava
Institute of Automation, Measurement and Applied Informatics
Center for Control of Distributed Parameter Systems
Faculty of Mechanical Engineering
cyril.belavy@stuba.sk

Pavol Noga
Research engineer at Slovak University of Technology in Bratislava
Institute of Automation, Measurement and Applied Informatics
Center for Control of Distributed Parameter Systems
Faculty of Mechanical Engineering
pavol.noga@stuba.sk

Karol Ondrejkovič
Research engineer at Slovak University of Technology in Bratislava
Institute of Automation, Measurement and Applied Informatics
Center for Control of Distributed Parameter Systems
Faculty of Mechanical Engineering
karol.ondrejkovic@stuba.sk

Reviewers:

Dr. h. c. Prof. Ing. Anton Víteček, CSc.
VŠB Technical University of Ostrava, Czech Republic

Prof. Ing. Mikuláš Alexík, PhD.
University of Žilina, Slovak Republic

Prof. Ing. Juraj Spalek, PhD.
University of Žilina, Slovak Republic

ACKNOWLEDGEMENTS

The authors would like to gratefully thank for the support of the Slovak Scientific Grant Agency VEGA within the framework of the project VEGA-1/0036/08 „*Advanced methods for modelling, control and design of distributed parameter systems*“, VEGA 1/0138/11 „*Control of dynamical systems given by numerical structures as distributed parameter systems*“, moreover, the Slovak Research and Development Agency APVV in the framework of projects APVV-0160-07 „*Advanced methods of modelling, control and design of mechatronic systems as lumped input and distributed parameter output systems*“ and APVV-0131-10 „*High-tech solutions for technological processes and mechatronic components as controlled distributed parameter systems*“. Furthermore, the authors would like to thank the Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic for the Structural Funds of European Union ASFEU for the support provided within the framework of the projects ITMS-26240220072 „*Center of competence for intelligent technologies and informatization of systems and services*“ and ITMS-26240220084 „*University scientific park STU in Bratislava*“.

Given the interdisciplinary nature of the book, the authors gratefully acknowledge the consultations in the area of casting and continuous casting of steel provided by Dr. Ladislav Stanček of the Institute of Technologies and Materials of the Faculty of Mechanical Engineering, STU in Bratislava. In the field of extrusion, we would like to acknowledge the assistance of prof. Marián Peciar and Dr. Roman Fekete from the Institute of Chemical and Hydraulic Machines and Equipment of the Faculty of Mechanical Engineering, STU in Bratislava. We obtained valuable insight in the area of numerical modelling of electromagnetic fields from prof. Ivo Doležal from the Department of Electroenergetics, Faculty of Electrical Engineering, Czech Technical University in Prague and Dr. Pavel Karban from the Department of Theory of Electrical Engineering, Faculty of Electrical Engineering at the University of West Bohemia, Plzeň. Moreover, we would like to thank the assistance of prof. Mihály Réger from the Institute of Material Sciences and Technology, Donát Bánki Faculty of Mechanical and Safety Engineering, Óbuda University, Budapest; for providing thermo-physical properties of materials heated by induction heating.

The entire team of authors would like to thank P. Buček, PhD., J. Camber, PhD. J. Kapusta, PhD. and S. Lipár, PhD. for cooperating during the preparation of the manuscript. Furthermore, we would like to acknowledge the hard work of the doctoral students at the Institute of Automation, Measurement and Applied Informatics of the Faculty of Mechanical Engineering STU in Bratislava, to name a few, M. Bartko, L. Bartalský, B. Barbolyas and M. Mendel for their assistance with the graphical work in the manuscript. We would like to thank for the help provided in proofreading the book by G. Takács, PhD. and J. Takács Sipos. We owe our gratitude for the valuable advice and comments of reviewers; prof. A. Víteček, prof. M. Alexík and prof. J. Spalek.

FOREWORD

The dramatic developments in information technology open up new possibilities for modelling technological and production processes as time-space continuum systems. Currently, engineering practice routinely utilizes numerical simulations for the analysis of the dynamics of these processes, based on the underlying nonlinear partial differential equations…

Let us start form the fact that various technological and production processes that are in reality non-linear distributed parameter systems, are normally used near certain steady-state operating modes or between such operating modes. This down-to-earth approach defines the control tasks in the linearized vicinity of the operating modes and control problems along linearized segments of the nonlinear dynamics of transitions. Here, these control tasks are solved by linearized distributed parameter models of the plants, controlled as linear lumped input and distributed parameter output systems. Linearized models are developed based on the results of the numerical models of the controlled processes. This book proposes a control method that is specifically designed to control linearized dynamic models of the controlled technology or production process. This concept offers the decomposition of the dynamics of control systems as well as control synthesis into temporal and spatial components. The spatial problems of control synthesis are solved based on the extremal tasks of approximation of functions and temporal problems as control synthesis of lumped parameter control loops – and these are no more complex than the solution of the lumped parameter control tasks usually encountered in engineering. In general, versatile schemes are obtained for the design of distributed parameter control systems, showing the potential of the approach on demonstration examples taken from the field of casting, induction heating, extrusion of plastics, continuous casting of steel and groundwater remediation.

In order to model and optimize the control loops laid out in this book, a co-simulation scheme is offered through the cooperation of Matlab & Simulink with different software products for the numerical simulation of various physical phenomena. The task of control design, which is at the center point of this book, is supported by the DPS Control for Matlab & Simulink software suite. The software support provided here, along with the demonstration models of control problems aim to provide an impulse for the engineering community, to solve industrial control problems by the means of distributed parameter systems. Everything is ready and surprisingly simple; one just needs to try it…

1. INTRODUCTION

The works of A. Hrennikoff „*Solution of Problems in Elasticity by the Framework Method*“ (1941) and R. Courant „*Variational Methods for Solution of Equilibrium and Vibration*“ (1943) initiated the development of numerical methods for the solution partial differential equations at the beginning of the 40’s of the last century. Today, a wide range of sophisticated numerical methods are offered for solving partial differential equations, integral equations and integro-partial differential equations defined on complex three dimensional shapes.

The last decades saw an immense growth in novel software tools for various engineering disciplines. These are based on numerical methods, and provide the solution and post-processing tools necessary for problems from practically all the scientific and engineering fields. Software products of multi-national corporations offer a deeper insight into the time-space behavior of continuum type matter. Just to name a few, these are ANSYS, ANSYS Polyflow, COMSOL Multiphysics, ProCAST, ESI Weld Planner, COMPUPLAST, etc. Numerical dynamical analyses made possible and contributed to numerous technical innovations and high-tech solutions, as well as a large number of patent claims.

The popularity of numerical simulations in engineering practice can be attributed to the fact that simulation studies facilitate in-depth analyses and contribute to production cost reductions up to 10-30% while shortening the time in which a product gets from the first draft to the market by 20-50%.

This, at the same time, means that dynamical models of various technological and production processes given in the form of numerical solutions of the underlying partial differential equations are routinely being dealt with. The frequent use of these numerical tools may also open new horizons for not just the simulation, but also the closed-loop control of technological and production processes as distributed parameter systems.

Real systems described by partial differential equations are considered to be distributed parameter systems from the perspective of systems and control theory. Research in this field has a long and rich history. At the very first conference of the International Federation of Automatic Control (IFAC) in Moscow (1960) L. S. Pontryagin, R. Bellman and R. E. Kalman presented their relevant results within the “Optimal Systems of Control“ session, while A. G. Butkovskiy and A. Y. Lerner, (1960), formulated basic control problems for systems described by partial differential equations, or in other words, distributed parameter systems. The first

important books on the topic were published within the same decade: P. K. C. Wang „*Control of distributed parameter systems"* (1964), A. G. Butkovskiy „*Optimal control of distributed parameter systems"* (1965) and later J. L. Lions „*Optimal control of systems governed by partial differential equations"* (1971). These works are based on analytical solutions of partial differential equations and deal with optimization problems of semi product preheating in metallurgy, steel production, heat treatment, structural damping of buildings, hydromechanical systems, aerospace and space-flight problems as well as nuclear technology. A number of exciting results have been achieved in the mathematical theory of distributed parameter systems control as well, where the problem is often referred to as the *control of partial differential equations*: Curtain, R., F., Zwart, H., J. (1995), Lasiecka, I., Triggiani, R. (2000), Komornik, V., Loreti, P. (2005), Krstič, M., Shmyslayev, A. (2008), Zwart, H., J. (2013), etc. However, it has to be said that the applicability of the results of this type of academic and mathematical theory in engineering practice is severely limited; especially by the possibilities of evaluating the analytical solutions of the underlying dynamics of the controlled plants. This is the main reason why practically all time/space dynamical analysis in engineering practice is solved by numerical methods. It is also the reason why we have started the search for new control solutions based on validated numerical models of technological and production processes.

At the beginning of the eighties of the last century, a new lumped input and distributed output approach for distributed parameter systems control was proposed and elaborated at the Institute of Automation, Measurement and Applied Informatics of the Faculty of Mechanical Engineering of the Slovak University of Technology in Bratislava (IAMAI-FME-STU), which was subsequently updated according to needs of engineering practice.

Real systems described by partial differential equations relating various quantity fields on the left and right hand sides of the equation are, in fact, distributed input and distributed output systems. Once we attach actuators and distributed input generators to the input of this system, we obtain a controlled lumped input and distributed parameter output system, which relates the lumped inputs of actuators and distributed quantity fields at the output. Currently, it is possible to construct models of these controlled systems based on validated numerical models of various phenomena.

The practical engineering approach to control distributed parameter systems nowadays comprises of a number of different approaches, methods and software products, by which it is possible to:

- assign lumped input and distributed parameter output systems to controlled distributed parameter objects,
- perform the analysis and decomposition of dynamics of lumped input and distributed parameter output systems to time and space domain components,

- formulate and solve control problems by the means of their decomposition within the control synthesis problem to time and space domain problems, where the time domain part follows standard SISO or MIMO controller design and the space domain part leads to approximation problems,
- reconstruct controlled distributed quantities measured in finite locations on the definition domain of the controlled system by numerical models,
- make use of the broad spectrum of software support offered by software products for numerical analysis of dynamics of technological and production processes,
- work without the discretization of mathematical models, while the precision of models utilized during the solution is on the precision level of the validated numerical models,
- make use of the software support offered by the Distributed Parameter Systems Blockset for Matlab & Simulink for the solution of the control synthesis issue,
- utilize the possibilities of co-simulation, meaning the simultaneous cooperation of software products for the dynamical analysis of technological and production processes with the Distributed Parameter Systems Blockset for Matlab & Simulink for the investigation and optimization of the distributed parameter control loop characteristics and properties,
- make use of the software support offered by the Interactive Control service via the Internet.

The exploitation of technological and production processes as non-linear distributed parameter systems is usually carried out within the bounds of a select steady-state operating mode or between such modes. This practical reality defines the control tasks in the linearized vicinity of the operating modes and control problems along linearized segments of the nonlinear dynamics of transitions. These control tasks are solved by linearized distributed parameter models of technological and production processes as linear lumped input and distributed parameter output systems. Linearized models of technological and production processes are built based on validated numerical models of controlled processes. This book proposes the solution of control by using methods designed to handle the linearized dynamical models of the plant.

The basic description of the underlying principles and methods for the control of technological and production processes based on linearized lumped input and distributed parameter output models is provided in Chapters *2* to *7* of this book. Chapter *6*, demonstrates the possibilities of the software support offered by the DPS Blockset for Matlab & Simulink, including features of co-simulation. The next chapters show some of the applications of the proposed approach in the field of casting, induction heating, plastics extrusion, continuous casting and groundwater

remediation. The benchmark control problems are based on technical solutions common in engineering practice, where control may be understood as tracking a set-point, whilst the structural layout of the facilities virtually eliminates the possibilities of direct actions of outer disturbances on the distributed controlled variable. In these configurations, manipulated variables or outputs are measured only in a few physical locations of the plant (controlled object) and then this output is available only in the form of partial responses in the temporal relation, affecting the overall concept of the control synthesis. The situation is different, in case the manipulated variables are quantity fields measurable by camera methods, for example by a thermal imaging camera, for certain types of welding, temperature fields in forest fires, etc. In this case, we may actually consider a system with distributed parameter outputs on the entire domain of definition.

The design of control for distributed parameter systems is based on numerical models of the controlled technological and production processes, which are built by software products, including but not limited to ProCAST, Quick Field, COMSOL Multiphysics, COMPUPLAST and ANSYS Polyflow. In general, the controlled systems (plants) are nonlinear systems from the physical viewpoint; therefore, their numerical models are also based on numerical solutions of the underlying nonlinear partial differential equations. The dynamical analysis, formulation and solution of these control problems is carried out in the linearized neighborhoods of the steady-state operating modes of interest; while the nonlinear transition from one operating point to another is solved separately based on the segmentation of nonlinear dynamics. In case there are significant uncertainties in the model, robust control methods are utilized. A co-simulation framework is utilized for the design and optimization of distributed parameter control systems. This scheme means the high-level cooperation of software products (such as ProCAST, COMSOL Multiphysics, ANSYS and others) and Matlab & Simulink using the DPS Blockset for Matlab & Simulink. These demonstrations are based on the work of the home institute of the authors (IAMAI-FME-STU) and industry. One has to keep in mind, that the possible technical fields benefiting from the control problem formulated for distributed parameter systems is enormous, as it has been suggested by Dochain, D. (1992, 1997), Christofides, P. D. (2001, 2008), Han-Xiong Li (2009, 2010), Kugi, A., Sawodny, O (2010), Meurer, T., Kugi, A., (2009, 2011) and others.

The issue of early and late discretization of mathematical models often comes up in the field dealing with the design of distributed parameter control. However, the methods proposed in this book do not require the discretization of mathematical models at all. As the utilized models and dynamic characteristics are based on numerical modeling, their precision expressing the changes in the manipulated variables is given by the precision of the numerical models themselves.

This book sums up the results of the long-term research carried out at the IAMAI-FME-STU in the fields of modelling and control of distributed parameter systems. In reality, this ongoing research started in the sixties of the last century with our participation in the assignments of the government for fundamental research in the programs of the Czechoslovak Academy of Sciences, coordinated by Institute of Information Theory and Automation Czechoslovak Academy of Sciences - UTIA ČSAV in Prague. At the end of the eighties, the results concerning the design of the engineering approach for the control of distributed parameter systems based on the lumped input and distributed parameter output approach were first published. These results were later summed up in the book by Hulkó et al: *Modelling and Control of Distributed Parameter Systems with Demonstrations in Matlab* (1998). The software support package titled *Distributed Parameter Systems Blockset for Matlab & Simulink* (DPS Blockset in short) (2003-present) was designed within the framework of the proposed engineering approach at the IAMAI-FME-STU.

Recent works in the field were carried out with the support of the grants offered by the Slovak grant agencies VEGA, APVV and ASFEU. These were the grants VEGA-1/0036/08 *„Advanced methods for modelling, control and design of distributed parameter* systems*“,* VEGA 1/0138/11 *„Control of dynamical systems given as numerical structures as distributed parameter systems“*, APVV-0160-07 *„Advanced methods for modelling, control and design of mechatronic systems as distributed parameter systems“,* APVV-0131-10 *„High-tech solutions for technological processes and mechatronic components as controlled distributed parameter systems*“, *ITMS-26240220072 „Center of competence for intelligent technologies and informatization of systems and services“* and *ITMS-26240220084 „University scientific park STU in Bratislava“.* Moreover, the authors are grateful for the cooperation and support of our industrial partners, namely *Železiarne Podbrezová (Podbrezová Steelworks) Research and Development Center, Welding Research Institute – Industrial Institute of SR, TEN Slovakia - The High-Tech Company,* etc.

The results of our research efforts were presented in papers at international conferences and peer-reviewed journals, Hulkó (2009, 2012, 2013), as well as a part of book chapters, Hulkó (2011), Belavý (2012). As a part of our overall research efforts, several doctoral theses were successfully submitted and defended at the IAMAI-FME-STU; within the study programs "*Automation and control of machines and processes*" and "*Mechatronics*". Some of these were authored by J. Szuda, PhD., P. Hurák, PhD., P. Buček, PhD., P. Noga, PhD., K. Ondrejkovič, PhD., J. Kapusta, PhD., J. Camber, PhD., M. Michalečko, PhD., S. Ľelo, PhD., S. Lipár, PhD., M. Vlček, PhD. and F. Vitáloš, PhD.

2. DPS – DDS – LDS

Technological and production processes having a continuum-based nature as real distributed parameter systems (DPS) are usually described by linear partial differential equations (PDE) in the linearized neighborhoods of a steady-state operating mode. From an input/output perspective, dynamical systems described by PDE between quantity fields on the right and left hand sides of the differential equations are, in fact, distributed input and distributed parameter output systems (DDS), (Fig. 2.1).



Fig. 2.1 Distributed input and distributed parameter output system.

DDS - distributed input and distributed parameter output system.

$U(\boldsymbol{x},t)$ - $U(x,y,z,t)$ - distributed input.

$Y(\boldsymbol{x},t)$ - $Y(x,y,z,t)$ - distributed output.

When actuators and generators of distributed inputs are connected for the purposes of control to DDS, we obtain controlled systems as lumped input and distributed parameter output systems (LDS), (Fig. 2.2, Fig. 2.3).

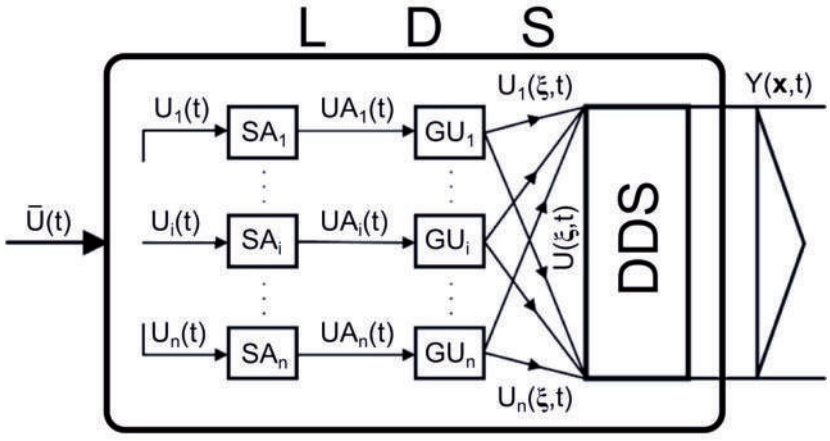


Fig. 2.2 Structure of lumped input and distributed parameter output system.

DDS - distributed input and distributed parameter output system.

LDS - lumped input and distributed parameter output system.

$\{GU_i\}_i$ - generators of distributed inputs.

$\{SA_i\}_i$ - actuators of lumped inputs.

$\bar{U}(t)$ - vector of continuous lumped inputs.

$U(\boldsymbol{\xi},t)$ - overall distributed input of DDS.

$\{U_i(t)\}_i$ - components of continuous lumped input vector.

$\{U_i(\boldsymbol{\xi},t)\}_i$ - partial continuous distributed inputs of DDS.

$\{UA_i(t)\}_i$ - continuous lumped inputs of $\{GU_i\}_i$.

$Y(\boldsymbol{x},t)$ - $Y(x,y,z,t)$ - distributed output.

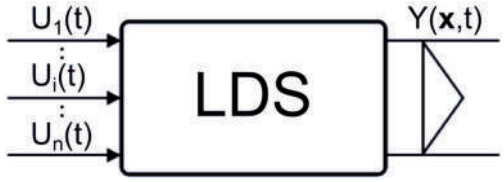


Fig. 2.3 Lumped input and distributed parameter output system.

LDS - lumped input and distributed parameter output system.

$\{U_i(t)\}_i$ - components of the continuous lumped input vector.

$Y(\boldsymbol{x},t)$ - $Y(x,y,z,t)$ - distributed output.

Assuming zero initial and boundary conditions, the input/output relationship for the system is given in the continuous form as

$$Y(\boldsymbol{x},t)=\sum_{i=1}^{n}Y_i(\boldsymbol{x},t)=\sum_{i=1}^{n}\mathcal{G}_i(\boldsymbol{x},t)\otimes U_i(t)=\sum_{i=1}^{n}\int_0^t \mathcal{G}_i\left(\boldsymbol{x},t-\tau\right)U_i\left(\tau\right)d\tau \tag{2.1}$$

while in the discrete form we have

$$Y(\boldsymbol{x},k)=\sum_{i=1}^{n}Y_i(\boldsymbol{x},k)=\sum_{i=1}^{n}\mathcal{G}H_i(\boldsymbol{x},k)\oplus U_i(k)=\sum_{i=1}^{n}\sum_{q=0}^{k}\mathcal{G}H_i(\boldsymbol{x},k-q)U_i(q) \tag{2.2}$$

where $\otimes$ indicates the convolution product and $\oplus$ the convolution sum, $\{U_i(t)\}_i$ and $\{U_i(k)\}_i$ are continuous and discrete lumped input variables, $\{G_i(\mathbf{x},t)\}_i$ are the LDS distributed impulse responses to the lumped inputs, $\{GH_i(\boldsymbol{x},k)\}_i$ their discrete versions assuming zero order holds H: HLDS – with unit sampling period, (Fig. 2.4). Where

$$\{Y_i(\boldsymbol{x},t)\}_i = \{G_i(\boldsymbol{x},t) \otimes U_i(t)\}_i = \left\{\int_0^t G_i(\boldsymbol{x},t-\tau)U_i(\tau)\,d\tau\right\}_i \tag{2.3}$$

and

$$\{Y_i(\boldsymbol{x},k)\}_i = \{GH_i(\boldsymbol{x},k) \oplus U_i(k)\}_i = \left\{\sum_{q=0}^{k} GH_i(\boldsymbol{x},k-q)U_i(q)\right\}_i \tag{2.4}$$

are individual, partial distributed output variables belonging to individual lumped inputs.



Fig. 2.4 Discrete lumped input and distributed parameter output system with zero order hold units.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$\{U_i(k)\}_i$ - components of the continuous lumped input vector.

$Y(\boldsymbol{x},t)$ - $Y(x,y,z,t)$ - distributed output.

By applying the Laplace transform we obtain

$$Y(\boldsymbol{x},s) = \sum_{i=1}^{n} Y_i(\boldsymbol{x},s) = \sum_{i=1}^{n} S_i(\boldsymbol{x},s)U_i(s) \tag{2.5}$$

$$\{Y_i(\boldsymbol{x},s) = S_i(\boldsymbol{x},s)U_i(s)\}_i \tag{2.6}$$

and after using the Z-transform we will have

$$Y(\boldsymbol{x},z) = \sum_{i=1}^{n} Y_i(\boldsymbol{x},z) = \sum_{i=1}^{n} S_i(\boldsymbol{x},z)U_i(z) \tag{2.7}$$

$$\{Y_i(\boldsymbol{x},z) = S_i(\boldsymbol{x},z)U_i(z)\}_i \tag{2.8}$$

3. ANALYSIS AND DECOMPOSITION OF DYNAMICS

In order to simplify the discussion, let us consider the time invariant linear HLDS distributed system on a one-dimensional interval [0, L] along the x-axis, Fig. 3.1, in the following text. Results obtained this way will be generally valid on complex shaped 3D definition domains as well.

3.1 Decomposition of Dynamics Based on Transient Characteristics

By applying a discrete unit step change in the i-th input we obtain the i-th discrete distributed step response $\mathcal{H}H_i(x,k)$. Furthermore, let us take the point x_i, from the interval [0, L], where distributed step response reaches its maximum amplitude in the steady-state. At this point, we obtain a partial response $\mathcal{H}H_i(x_i,\mathrm{k})$ of the distributed step response along the time axis, that is, variable k. We also obtain partial response characteristics $\mathcal{H}H_i(x,k)$ in the spatial direction, in time steps $\{k\}_k$, that is, variable x, (Fig. 3.1).

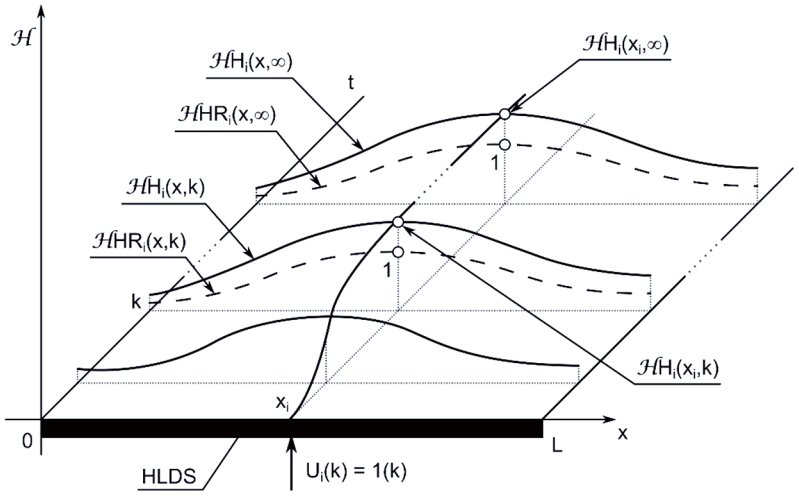


Fig. 3.1 Time and space components of the i-th individual distributed unit step response of HLDS.

HLDS	- discrete lumped input and distributed parameter output system with zero order hold units.
$U_i(k)=1(k)$	- *i*-th lumped input as a discrete unit step function.
$\mathcal{H}H_i(x,k)$	- *i*-th discrete partial distributed step response in *k* variable *x*.
$\mathcal{H}H_i(x_i,k)$	- *i*-th discrete partial distributed step response in x_i, variable *k*.
$\mathcal{H}HR_i(x,k)$	- *i*-th reduced discrete partial distributed step response in *k*, variable *x*.
$\mathcal{H}H_i(x,\infty)$	- *i*-th discrete partial distributed step response in steady-state.
$\mathcal{H}HR_i(x,\infty)$	- *i*-th reduced discrete partial distributed step response in steady-state.

We get a set of discrete individual distributed step responses $\{\mathcal{H}H_i(x,k)\}_{i,k}$ for $\{i\}_i$, $i=1,n$ individual inputs. Moreover, we obtain partial distributed step response characteristics $\{\mathcal{H}H_i(x_i,k)\}_{i,k}$ depending on time (variable *k*) in the locations $\{x_i\}_i$ from the interval [0, *L*]. Let us consider the discrete partial distributed step response in the space domain $\{\mathcal{H}H_i(x,k)\}_{i,k}$ at time steps $\{k\}_k$, that is, alongside variable *x*. On this set, $\{\mathcal{H}H_i(x,k)\}_{i,k}$, we define reduced responses

$$\left\{\mathcal{H}HR_i(x,k)=\frac{\mathcal{H}H_i(x,k)}{\mathcal{H}H_i(x_i,k)}\right\}_{i,k} \tag{3.1}$$

for $\{\mathcal{H}H_i(x_i,k)\neq 0\}_{i,k}$.

In fact, this means that at a step change in the *i*-th input of HLDS on the output at x_i we will get a time domain function $\mathcal{H}H_i(x_i,k)$, and the values of the distributed output $\mathcal{H}H_i(x,k)$ in the space domain are given by the reduced profiles $\mathcal{H}HR_i(x,k)$ multiplied by $\mathcal{H}H_i(x_i,k)$ at each time step $\{k\}_k$: $\mathcal{H}H_i(x,k)=\mathcal{H}HR_i(x,k)\mathcal{H}H_i(x_i,k)$. In this sense, these dynamic characteristics decompose into time and space components:

Time components

$$\{\mathcal{H}H_i(x_i,k)\}_i \qquad \text{– for given } i \text{ and chosen } x_i \text{ – variable } k \tag{3.2}$$

Space components

$$\left\{\mathcal{H}HR_i\left(x,k\right)\right\}_i \qquad \text{– for given } i \text{ and chosen } k \text{ – variable } x \tag{3.3}$$

Often, it is useful to assign transfer functions to the partial time domain characteristics. Then we have:

Time components

$$\left\{SH_i\left(x_i,z\right)\right\}_i \qquad \text{– for given } i \text{ and chosen } x_i \text{ – variable } z. \tag{3.4}$$

Space components

$$\left\{\mathcal{H}HR_i\left(x,k\right)\right\}_i \qquad \text{– for given } i \text{ and chosen } k \text{ – variable } x. \tag{3.5}$$

Let us now consider the reduced profiles of the partial distributed step responses in steady-state, i.e. when $k \to \infty$ (Fig. 3.2)

$$\left\{\mathcal{H}HR_i\left(x,\infty\right)=\frac{\mathcal{H}H_i\left(x,\infty\right)}{\mathcal{H}H_i\left(x_i,\infty\right)}\right\}_i \tag{3.6}$$

for $\left\{\mathcal{H}H_i\left(x_i,\infty\right)\neq 0\right\}_i$. For the distributed output in steady-state we get

$$\begin{aligned} &Y\left(x,\infty\right)=\sum_{i=1}^{n}\mathcal{H}H_i\left(x,\infty\right)U_i\left(\infty\right)= \\ &=\mathcal{H}HR_1\left(x,\infty\right)\mathcal{H}H_1\left(x_1,\infty\right)U_1\left(\infty\right)+... \\ &+\mathcal{H}HR_i\left(x,\infty\right)\mathcal{H}H_i\left(x_i,\infty\right)U_i\left(\infty\right)+... \\ &+\mathcal{H}HR_n\left(x,\infty\right)\mathcal{H}H_n\left(x_n,\infty\right)U_n\left(\infty\right) \end{aligned} \tag{3.7}$$

Thus,

$$\begin{aligned} &Y\left(x,\infty\right)=Y_1\left(x_1,\infty\right)\mathcal{H}HR_1\left(x,\infty\right)+\ldots+Y_i\left(x_i,\infty\right)\mathcal{H}HR_i\left(x,\infty\right)+\ldots \\ &+Y_n\left(x_n,\infty\right)\mathcal{H}HR_n\left(x,\infty\right)=\sum_{i=1}^{n}Y_i\left(x,\infty\right) \end{aligned} \tag{3.8}$$

According to Fig. 3.3 for

$$\begin{aligned} &\left[Y_1\left(x_1,\infty\right),\ldots,Y_i\left(x_i,\infty\right),\ldots,Y_n\left(x_n,\infty\right)\right]= \\ &=\left[\mathcal{H}H_1\left(x_1,\infty\right)U_1\left(\infty\right),\ldots,\mathcal{H}H_i\left(x_i,\infty\right)U_i\left(\infty\right),\ldots,\mathcal{H}H_n\left(x_n,\infty\right)U_n\left(\infty\right)\right] \end{aligned} \tag{3.9}$$

we will obtain

$$\left[Y_1(x,\infty),\ldots,Y_i(x,\infty),\ldots,Y_n(x,\infty)\right]=$$
$$=\begin{bmatrix} Y_1(x_1,\infty)\,\mathcal{H}HR_1(x,\infty),\ldots,Y_i(x_i,\infty)\,\mathcal{H}HR_i(x,\infty),\ldots, \\ Y_n(x_n,\infty)\,\mathcal{H}HR_n(x,\infty) \end{bmatrix} \tag{3.10}$$

In this limiting case, the steady-state characteristics of the dynamics will decompose into:

Time components

$$\mathcal{H}H_1(x_1,\infty),\ldots,\mathcal{H}H_i(x_i,\infty),\ldots,\mathcal{H}H_n(x_n,\infty) \tag{3.11}$$

Space components

$$\mathcal{H}HR_1(x,\infty),\ldots,\mathcal{H}HR_i(x,\infty),\ldots,\mathcal{H}HR_n(x,\infty) \tag{3.12}$$

From the viewpoint of solving control problems in steady-state, based on the analysis presented above we may note that it is possible to uniquely assign vector $\{Y_i(x_i,\infty)\}_i$ to each vector $\{U_i(\infty)\}_i$, along with the partial distributed outputs $\{Y_i(x,\infty)\}_i$ and set-point $Y(x,\infty)$. Similarly, we can assign distributed outputs $\{Y_i(x,\infty)\}_i$ and $Y(x,\infty)$ to each vector $\{Y_i(x_i,\infty)\}_i$.

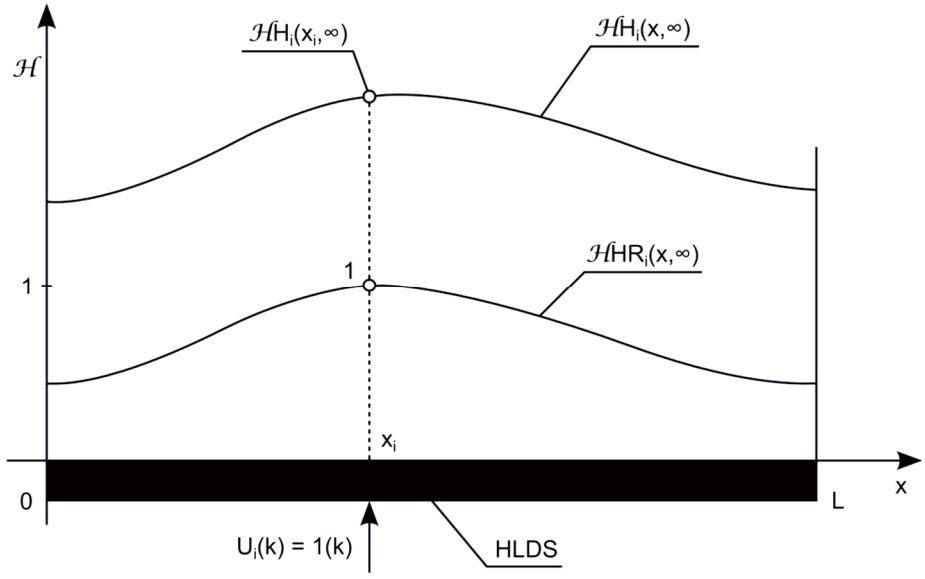


Fig. 3.2 Steady-state of the i-th distributed unit step response and the reduced profile.

HLDS - lumped input and distributed parameter output system with zero order hold units.

$U_i(k) = 1(k)$ - *i*-th lumped input as discrete unit step function.

$\mathcal{H}H_i(x,\infty)$ - *i*-th discrete partial distributed step response in steady-state.

$\mathcal{H}HR_i(x,\infty)$ - *i*-th reduced discrete partial distributed step response in steady-state.

$\mathcal{H}H_i(x_i,\infty)$ - *i*-th discrete partial distributed step response in steady-state for x_i.

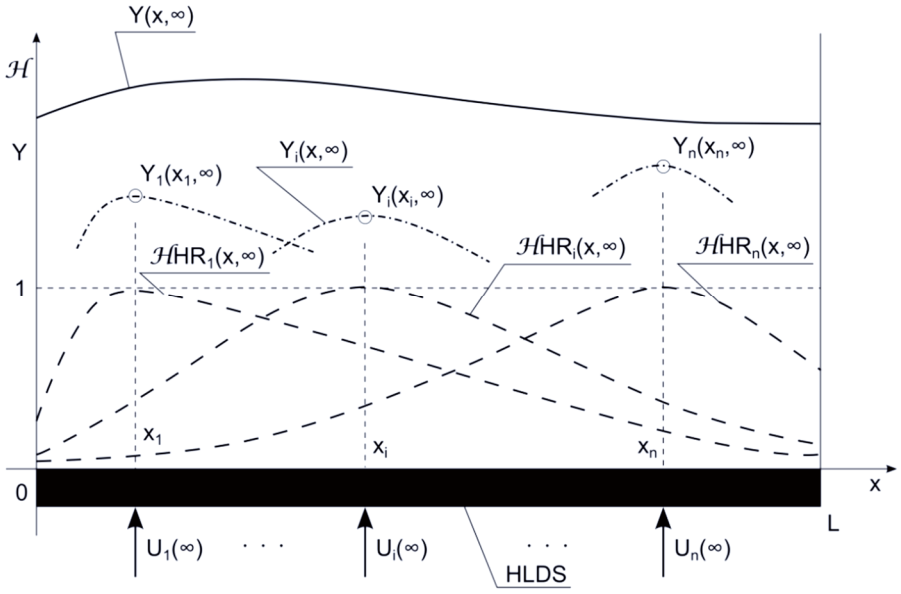


Fig. 3.3 Components of the distributed output in steady-state.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - reduced discrete partial distributed step responses in steady-state.

$\{U_i(\infty)\}_i$ - components of lumped input vector in steady-state.

$Y(x,\infty)$ - overall distributed output in steady-state.

$Y_i(x,\infty)$ - *i*-th partial distributed output in steady-state.

$\{Y_i(x_i,\infty)\}_i$ - components of partial distributed outputs vector in steady-state in locations with coordinates $\{x_i\}_i$.

For the sum of partial distributed outputs in the steady-state it is clear that

$$Y(\boldsymbol{x},\infty)=\sum_{1}^{n}Y_i(\boldsymbol{x}_i,\infty)\mathcal{H}\dot{H}R_i\left(\boldsymbol{x},\infty\right)=\sum_{i=1}^{n}\left[U_i\left(\infty\right)\mathcal{H}H_i\left(\boldsymbol{x}_i,\infty\right)\right]\mathcal{H}\dot{H}R_i\left(\boldsymbol{x},\infty\right)=$$
$$=\sum_{i=1}^{n}U_i\left(\infty\right)\left[\mathcal{H}H_i\left(\boldsymbol{x}_i,\infty\right)\mathcal{H}\dot{H}R_i\left(\boldsymbol{x},\infty\right)\right]=\sum_{1}^{n}U_i(\infty)\mathcal{H}H_i\left(\boldsymbol{x},\infty\right) \tag{3.13}$$

Furthermore, for $\left\{x_j\right\}_j$, $j=1,n$ we will get

$$\left\{Y\left(x_j,\infty\right)\right\}_i=\sum_{1}^{n}U_i\left(\infty\right)\mathcal{H}H_i\left(x_i,\infty\right)\mathcal{H}\dot{H}R_i\left(x_j,\infty\right) \tag{3.14}$$

This means, that for the system with the steady-state responses of the distributed transient responses $\left\{\mathcal{H}H_i\left(x,\infty\right)\right\}_i$ a uniquely determined relationship holds between the pair $\left\{Y\left(x,\infty\right),\bar{U}_i\left(\infty\right)\right\}$ and $\left\{\bar{Y}\left(x_j,\infty\right),\bar{U}_i\left(\infty\right)\right\}$ in the steady-state.

3.2 Decomposition of Dynamics Based on Impulse Characteristics

This procedure applies to the set of discrete individual distributed impulse characteristics of HLDS $\left\{\mathcal{G}H_i\left(x,k\right)\right\}_{i,k}$ as well (Fig. 3.4). In the locations $\{x_i\}_i$ we obtain the partial response $\left\{\mathcal{G}H_i\left(x_i,k\right)\right\}_{i,k}$, by which we may define the reduced (impulse response) profiles

$$\left\{\mathcal{G}\dot{H}R_i(x,k)=\frac{\mathcal{G}H_i(x,k)}{\mathcal{G}H_i(x_i,k)}\right\}_{i,k} \tag{3.15}$$

for $\left\{\mathcal{G}H_i\left(x_i,k\right)\neq 0\right\}_{i,k}$.

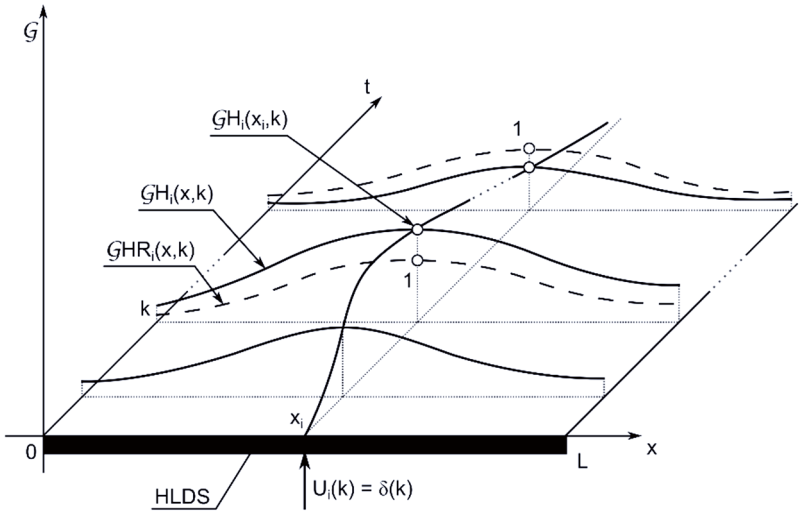


Fig. 3.4 Time and space components of the i-th discrete individual unit impulse response.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$U_i(k) = \delta(k)$ - *i*-th lumped input as discrete unit impulse function.

$\mathcal{G}H_i(x,k)$ - *i*-th discrete partial distributed impulse response in *k*, variable *x*.

$\mathcal{G}H_i(x_i,k)$ - *i*-th discrete partial distributed impulse response in x_i, variable *k*.

$\mathcal{G}HR_i(x,k)$ - *i*-th reduced discrete partial distributed impulse response in *k*, variable *x*.

In this way, spatial and temporal partial distributed impulse characteristics are introduced on the set of discrete partial distributed impulse characteristics, as temporal and spatial components of the characteristics given by:

Time components

$$\left\{\mathcal{G}H_i(x_i,k)\right\}_i \quad \text{– for given } i \text{ and chosen } x_i \text{ – variable } k. \tag{3.16}$$

Space components

$$\left\{\mathcal{G}HR_i(x,k)\right\}_i \quad \text{– for given } i \text{ and chosen } k \text{ – variable } x. \tag{3.17}$$

The partial system outputs in locations $\{x_i\}_i$, given as responses to input sequences, are then given by the time components of HLDS dynamics as

$$\{Y_i(x_i,k)\}_i = \{\mathcal{G}H_i(x_i,k) \oplus U_i(k)\}_i = \left\{\sum_{q=0}^{k} \mathcal{G}H_i(x_i,k-q)U_i(q)\right\}_i \tag{3.18}$$

and the individual distributed outputs using $\{\mathcal{G}HR_i(x,k)\}_{i,k}$ become

$$\begin{aligned}\{Y_i(x,k)\}_i &= \{\mathcal{G}H_i(x_i,k)\mathcal{G}HR_i(x,k) \oplus U_i(k)\}_i = \\ &= \left\{\sum_{q=0}^{k} \mathcal{G}HR_i(x,k)\mathcal{G}H_i(x_i,k-q)U_i(q)\right\}_i\end{aligned} \tag{3.19}$$

see the illustration in Fig. 3.5.

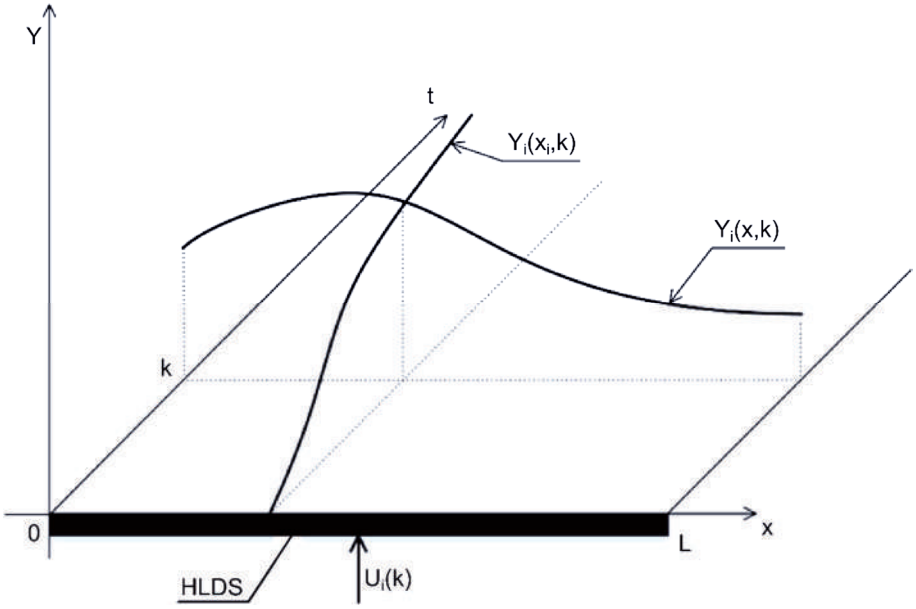


Fig. 3.5 Partial response of the i-th individual distributed parameter output in time and space relation.

HLDS	- discrete lumped input and distributed parameter output system with zero order hold units.
$U_i(k)$	- *i*-th discrete lumped input.
$Y_i(x_i,k)$	- *i*-th partial distributed output in time relation for point x_i, variable *k*.
$Y_i(x,k)$	- *i*-th partial distributed output in space relation for *k*, variable *x*.

Let us now denote the individual impulse characteristics by $\{\mathcal{G}H_j(x,k)\}_j$ and also consider locations $\{x_i\}_i$ for the partial responses $\{\mathcal{G}H_j(x_i,k)\}_{j,i}$. This way, we obtain partial responses of the overall distributed output at locations $\{x_i\}_i$ as follows:

$$\{Y(x_i,k)\}_i = \left\{\sum_j^n \mathcal{G}H_j(x_i,k) \oplus U_j(k)\right\}_i = \left\{\sum_j^n \sum_{q=0}^k \mathcal{G}H_j(x_i,k-q)U_j(q)\right\}_i \tag{3.20}$$

3.3 Decomposition of Dynamics Based on Partial Outputs

The same methodology can be applied to the set of individual distributed outputs $\{Y_i(x,k)\}_{i,k}$. Let us introduce the reduced profiles

$$\{YR_i(x,k)\}_i = \left\{\frac{Y_i(x,k)}{Y_i(x_i,k)}\right\}_{i,k} \tag{3.21}$$

for $\{Y_i(x_i,k) \neq 0\}_{i,k}$. The resulting overall output at time step k becomes

$$\begin{aligned} Y(x,k) &= \sum_i Y_i(x,k) = \sum_i \mathcal{G}H_i(x,k) \oplus U_i(k) = \\ &= \sum_i Y_i(x_i,k) YR_i(x,k) \end{aligned} \tag{3.22}$$

Finally, we get time and space components of the individual distributed outputs in the form:

Time components

$$\{Y_i(x_i,k)\}_i \qquad \text{– for given } i \text{ and chosen } x_i \text{ – variable } k \tag{3.23}$$

Space components

$$\{YR_i(x,k)\}_i \qquad \text{– for given } i \text{ and chosen } k \text{ – variable } x \tag{3.24}$$

Meanwhile for $k \to \infty$ the responses $\{YR_i(x,k)\}_i$ will converge to $\{\mathcal{H}HR_i(x,\infty)\}_i$ as

$$\left.\begin{aligned} &YR_i(x,\infty) = Y_i(x,\infty) / Y_i(x_i,\infty) = \\ &= U_i(\infty)\,\mathcal{H}H_i(x,\infty) / U_i(\infty)\,\mathcal{H}H_i(x_i,\infty) = \mathcal{H}HR_i(x_i,\infty) \end{aligned}\right\}_i \tag{3.25}$$

Moreover, according to (3.22) it is evident that in points $\{x_i\}_i$ we will have

$$\begin{bmatrix} Y(x_1,k) \\ \vdots \\ Y(x_i,k) \\ \vdots \\ Y(x_n,k) \end{bmatrix} = \begin{bmatrix} YR_1(x_1,k) & ,..., & YR_i(x_1,k) & ,..., & YR_n(x_1,k) \\ & & & & \\ YR_1(x_i,k) & ,..., & YR_i(x_i,k) & ,..., & YR_n(x_i,k) \\ & & & & \\ YR_1(x_n,k) & ,..., & YR_i(x_n,k) & ,..., & YR_n(x_n,k) \end{bmatrix} \begin{bmatrix} Y_1(x_1,k) \\ \vdots \\ Y_i(x_i,k) \\ \vdots \\ Y_n(x_n,k) \end{bmatrix} \tag{3.26}$$

(Fig. 3.6).

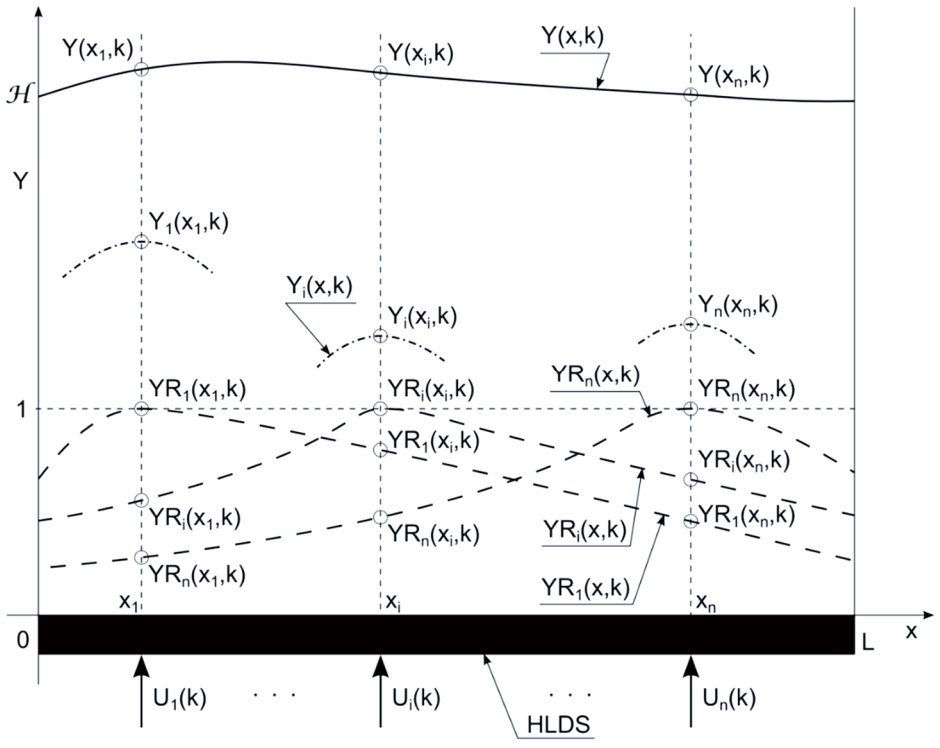


Fig. 3.6 Relations between vectors $\{Y(\boldsymbol{x}_i,k)\}_i$ *and* $\{Y_i(\boldsymbol{x}_i,k)\}_i$.

4. SYSTEM IDENTIFICATION

Technological and production processes are usually maintained to follow a set-point that is close to steady-state. Nonlinear transients between steady-state set-points are often represented using the segmentation of the nonlinear dynamic behavior with a set of linearized models. In order to solve control problems at these linearized regions, linear lumped input and distributed parameter output models are used. The linearized models of technological and production processes, actuators or generators of distributed inputs are created in software environments using a procedure that is analogous to building linear models based on the analytical solution of PDE. Thus, the material of this chapter begins with the introduction of analytical system identification methods for lumped input and distributed parameter output systems. The approach will be based on standard formulations of the underlying PDE. Next, the basics of two frequently used numerical techniques to solve PDE are introduced, namely the finite difference method and the finite element method. The issues that are typical for the identification of systems with uncertain dynamics and nonlinear transition between set-points is discussed in the chapters titled *Robust control of the casting mold temperature field* and in *Modeling and control of the continuous casting of steel*.

Sometimes we may encounter situations in engineering practice, where the manipulated variable and output can be measured directly using camera methods e.g. for thermal imaging. In these cases the basic characteristics—distributed transient characteristics—may be obtained directly, using thermal imaging. Meanwhile, the models of the plants can be drafted similarly to the case of numerical methods, where a suitable number of optimized measuring locations are selected, then spline methods are employed to interpolate the values in-between the locations. We also introduce the reader to the possibilities of designing a model for a certain required approximation precision in the closing pages of this chapter.

4.1 Analytical System Identification

The analytical system identification procedure starts with a mathematical and physical analysis of the problem. As a result, one obtains a mathematical description of the controlled technological and production process in the general form of a nonlinear PDE. We assign linear PDE to the controlled DDS in a linearized neighborhood of the given steady-state operating mode, along with distributed parameter input generators

$\{GU_i\}_i$ and ordinary differential equations to the lumped parameter actuators $\{SA_i\}_i$, (Fig. 4.2).

Next, we transform these equations subject to initial and boundary conditions to a standard form, (Butkovskiy (1979, 1982)), and obtain lumped and distributed impulse responses. These lumped and distributed impulse responses, called Green's functions, are the basis on which the lumped and distributed impulse responses of the identified DPS are estimated.

Let the Green's functions of these members be given in the general form

$$\{SA_i\}_i \quad \rightarrow\rightarrow \quad \{gA_i(t,\tau)\}_i \tag{4.1}$$

$$\{GU_i\}_i \quad \rightarrow\rightarrow \quad \{G_i(\xi,t,\tau)\}_i \tag{4.2}$$

$$\text{DDS} \quad \rightarrow\rightarrow \quad G(x,\xi,t,\tau) \tag{4.3}$$

Then, the distributed impulse responses become

$$\left\{ \mathcal{G}_i(x,t,\tau) = \int_0^t \int_0^t \int_0^L G(x,\xi,t,\tau_1) G_i(\xi,\tau_1,\tau_2)\, gA_i(\tau_2,\tau)\, d\xi d\tau_1 d\tau_2 \right\}_i \tag{4.4}$$

The dynamics of distributed input generators can be modelled as follows

$$\{GU_i\}_i \quad \rightarrow\rightarrow \quad \{gGU_i(t,\tau) T_i(x)\}_i \tag{4.5}$$

where the functions $\{T_i(x)\}_i$, space domain shaping functions, on intervals $\{\langle x_{i1}, x_{i2}\rangle\}_i \in \langle 0, L\rangle$ represent space domain dynamics of distributed input generators $\{GU_i\}_i$. Various shaping functions may be used for this purpose, which are illustrated in Fig. 4.1. The time domain dynamics of these are given as impulse responses $\{gGU_i(t,\tau)\}_i$. Combining them, we obtain

$$\left\{ \mathcal{G}_i(x,t,\tau) = \int_0^t \int_0^t \int_0^L G(x,\xi,t,\tau_1) T_i(\xi)\, gGU_i(\tau_1,\tau_2) gA_i(\tau_2,\tau) d\xi d\tau_1 d\tau_2 \right\}_i \tag{4.6}$$

The books by Butkovskiy (1979, 1982) list practically all standard forms of PDE that are important for engineering problems and more than 500 boundary problems. Furthermore, a collection of transfer functions for each standard form can be found

there as well, which enables creating both lumped input and distributed parameter output models in the form of transfer functions, (Fig. 4.2). Let us assign the following transfer functions to each subsystem and element of the LDS

$$\{SA_i\}_i \quad \rightarrow\rightarrow \quad \{SA_i(s)\}_i \tag{4.7}$$

$$\{GU_i\}_i \quad \rightarrow\rightarrow \quad \{SG_i(\xi,s)\}_i \tag{4.8}$$

$$\text{DDS} \quad \rightarrow\rightarrow \quad S(x,\xi,s) \tag{4.9}$$

Then the transfer functions between $\{U_i(s)\}_i$ and $\{Y_i(x,s)\}_i$ become

$$\left\{S_i(x,s)=\int_0^L S(x,\xi,s)SG_i(\xi,s)SA_i(s)\,d\xi\right\}_i \tag{4.10}$$

If

$$\{GU_i\}_i \quad \rightarrow\rightarrow \quad \{SG_i(s)T_i(x)\}_i \tag{4.11}$$

then

$$\left\{S_i(x,s)=\int_0^L S(x,\xi,s)T_i(\xi)SG_i(s)SA_i(s)\,d\xi\right\}_i \tag{4.12}$$

see Fig. 4.2 with the relation of the transfer functions.

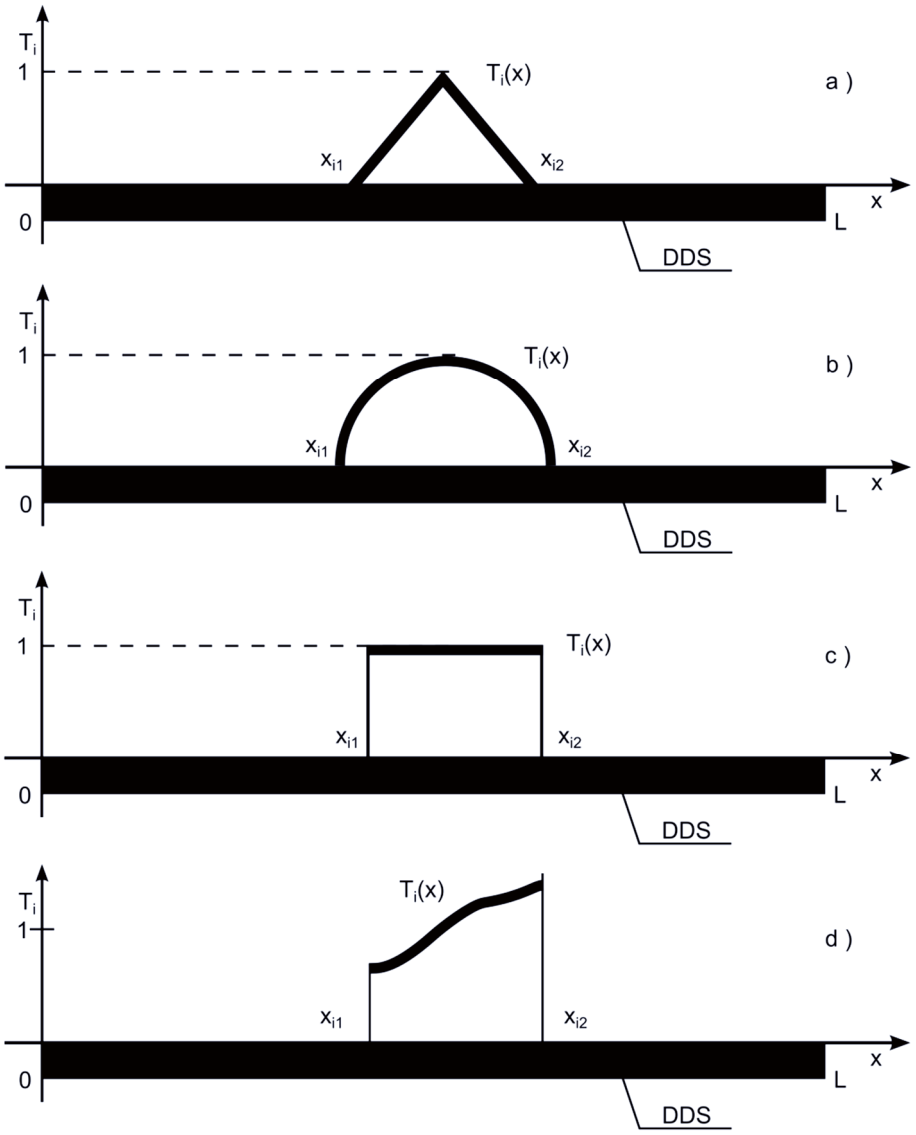


Fig. 4.1 Various types of shaping functions used to represent the space domain part of the dynamics of the distributed input generators.

DDS - distributed input and distributed parameter output system.

$\langle x_{i1}, x_{i2} \rangle$ - effective interval of the shaping function.

a) - triangular shaping function in spatial relation $T_i(x)$.

b) - sinusoidal shaping function in spatial relation $T_i(x)$.

c) - uniform shaping function in spatial relation (square function) $T_i(x)$.

d) - general shaping function in the spatial relation $T_i(x)$.

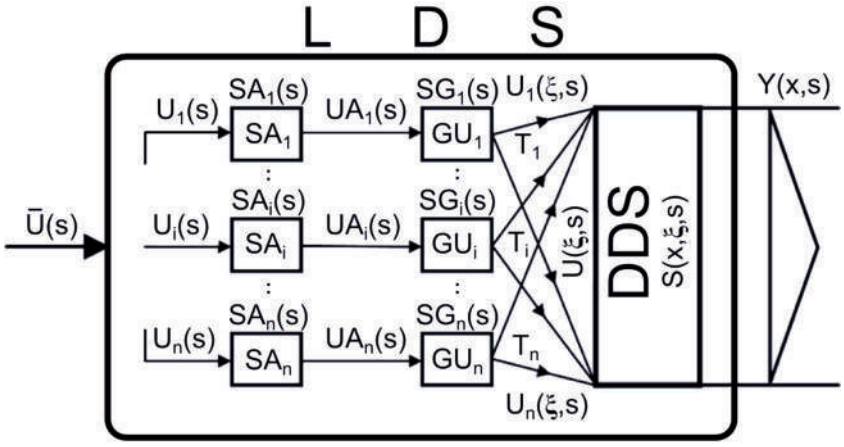


Fig. 4.2 Transfer functions of the lumped input and distributed parameter output system.

DDS - distributed input and distributed parameter output system.

LDS - lumped input and distributed parameter output system.

$S(x,\xi,s)$ - transfer function of the distributed input and distributed parameter output system – DDS.

$\{SA_i(s)\}_i$ - transfer functions of actuators of the lumped inputs $\{SA_i\}_i$.

$\{SG_i(s)\}_i$ - transfer functions of the generators of distributed inputs $\{GU_i\}_i$.

$\bar{U}(s)$ - transfer function of the continuous lumped parameter input vector.

$\{U_i(s)\}_i$ - transfer functions of the continuous lumped parameter input vector components.

$U(\xi,s)$ - transfer function of the overall distributed input of block DDS.

$\{U_i(\xi,s)\}_i$ - transfer functions of individual distributed inputs of block DDS.

$\{UA_i(s)\}_i$ - transfer functions of lumped inputs for generators $\{GU_i\}_i$.

$Y(x,s)$ - transfer functions of overall distributed output of LDS.

$\{T_i = T_i(x)\}_i$ - shaping units of distributed inputs in the spatial relation.

This brief analysis shows that, for the analytical identification of linearized models of lumped input and distributed parameter output systems, one can use all results applicable from analytic modelling of lumped as well as distributed parameter systems.

4.1.1 Some Examples of Analytical Models in Standardized Forms

To provide some examples, we list some ordinary and partial differential equations types as lumped and distributed parameter systems in a standard form in the following discussion, along with their transfer functions as shown by Butkovskiy (1979, 1982):

Model 1

$$a\frac{dY(t)}{dt}+bY(t)=U(t) \tag{4.13}$$

$$Y(0)=Y_0,\ \ t\geq 0$$

Standardizing function

$$R(t)=U(t)+aY_0\delta(t) \tag{4.14}$$

Green's function

$$G(t)=\frac{1}{a}\exp\left(-\frac{b}{a}t\right) \tag{4.15}$$

Transfer function

$$S(s)=\frac{1}{as+b} \tag{4.16}$$

The output is then obtained by the convolution of the Green's function with the standardizing function:

$$Y(t)=\int_0^t G(t-\tau)R(\tau)d\tau=\frac{1}{a}\exp\left(-\frac{b}{a}t\right)\left[\int_0^t \exp\left(\frac{b}{a}\tau\right)U(\tau)d\tau+aY_0\right] \tag{4.17}$$

Model 2

$$\frac{d^2Y(t)}{dt^2}+a\frac{dY(t)}{dt}+bY(t)=U(t) \tag{4.18}$$

$$Y(0)=Y_0,\quad \frac{dY(0)}{dt}=Y_1,\quad t\ge 0,\quad 4b>a^2$$

$$R(t)=U(t)+(Y_1+aY_0)\delta(t)+Y_0\delta'(t) \tag{4.19}$$

$$G(t)=\frac{1}{\omega}\exp(\alpha t)\sin(\omega t) \tag{4.20}$$

where

$$\omega=\frac{1}{2}\sqrt{4b-a^2},\quad \alpha=-\frac{a}{2}$$

$$S(s)=\frac{1}{s^2+as+b} \tag{4.21}$$

Model 3

$$a_0\frac{d^mY(t)}{dt^m}+a_1\frac{d^{m-1}Y(t)}{dt^{m-1}}+\ldots+a_{m-1}\frac{dY(t)}{dt}+a_mY(t)=U(t) \tag{4.22}$$

$$Y(0)=Y_0,\quad \frac{dY(0)}{dt}=Y_1,\cdots,\quad \frac{dY^{m-1}(0)}{dt^{m-1}}=Y_{m-1},\quad t\ge 0$$

$$R(t)=U(t)+\sum_{k=0}^{m-1}\left[\sum_{i=0}^{m-k-1}a_iY_{m-k-i-1}\right]\delta^{(k)}(t) \tag{4.23}$$

$$G(t)=\sum_{k=0}^{m}\frac{V_k\left[q_1(0),\ldots,q_m(0)\right]}{V\left[q_1(0),\ldots,q_m(0)\right]}q_k(t) \tag{4.24}$$

where

$$V\left[q_1(t),\ldots,q_m(t)\right]=\begin{vmatrix} q_1(t), & q_2(t), & \ldots, & q_m(t)\\ q_1'(t), & q_2'(t), & \ldots, & q_m'(t)\\ \vdots & \vdots & & \vdots\\ q_1^{(m-1)}(t), & q_2^{(m-1)}(t), & \ldots, & q_m^{(m-1)}(t)\end{vmatrix}$$

and $q_1(t), q_2(t), \ldots, q_m(t)$ are linearly independent solutions of the homogeneous differential equation (4.22). We obtain V_k by substituting the vector $[0,0,\ldots,0,1]$ into the *k*-th column of determinant *V*.

$$S(s) = \frac{1}{a_0 s^m + a_1 s^{m-1} + \ldots + a_{m-1} s + a_m} \tag{4.25}$$

Model 4

$$\frac{\partial Y(x,t)}{\partial t} - a^2 \frac{\partial^2 Y(x,t)}{\partial x^2} = U(x,t) \tag{4.26}$$

$$Y(x,0) = Y_0(x), \quad Y(0,t) = g_1(t), \quad Y(L,t) = g_2(t)$$

$$0 \le x \le L, \quad t \ge 0, \quad a \ne 0$$

$$R(x,t) = U(x,t) + Y_0(x)\delta(t) - a^2 \delta'(x) g_1(t) + a^2 \delta'(L-x) g_2(t) \tag{4.27}$$

$$G(x,\xi,t) = \frac{2}{L} \sum_{n=1}^{\infty} \sin\frac{n\pi x}{L} \sin\frac{n\pi\xi}{L} \exp\left[-\left(\frac{n\pi a}{L}\right)^2 t \right] \tag{4.28}$$

$$S(x,\xi,s) = \frac{2}{L} \sum_{n=1}^{\infty} \frac{\sin\frac{n\pi x}{L} \sin\frac{n\pi\xi}{L}}{s + \left(\frac{n\pi a}{L}\right)^2} \tag{4.29}$$

Model 5

$$\frac{\partial^2 Y(x,t)}{\partial t^2} - a^2 \frac{\partial^2 Y(x,t)}{\partial x^2} = U(x,t) \tag{4.30}$$

$$Y(x,0) = Y_0(x), \quad \frac{\partial Y}{\partial t}(x,0) = Y_1(x)$$

$$Y(0,t) = g_1(t), \quad Y(L,t) = g_2(t)$$

$$0 \le x \le L, \quad t \ge 0, \quad a \ne 0$$

$$R(x,t)=U(x,t)+Y_0(x)\delta'(t)+Y_1(x)\delta(t)+a^2\delta'(x)g_1(t)- \\ -a^2\delta'(L-x)g_2(t) \tag{4.31}$$

$$G(x,\xi,t)=\frac{2}{\pi a}\sum_{k=1}^{\infty}\sin\frac{k\pi x}{L}\sin\frac{k\pi\xi}{L}\sin\frac{ak\pi t}{L} \tag{4.32}$$

$$S(x,\xi,\mathrm{s})=\frac{2}{L}\sum_{k=1}^{\infty}\sin\frac{k\pi x}{L}\sin\frac{k\pi\xi}{L}\frac{1}{s^2+\left(\frac{ak\pi}{L}\right)^2} \tag{4.33}$$

Model 6

$$\frac{\partial^2 Y(x,y,z,t)}{\partial t^2}-a^2\left[\frac{\partial^2 Y(x,y,z,t)}{\partial x^2}+\frac{\partial^2 Y(x,y,z,t)}{\partial y^2}+\frac{\partial^2 Y(x,y,z,t)}{\partial z^2}\right]= \\ =U(x,y,z,t) \tag{4.34}$$

$$Y(x,y,z,0)=Y_0(x,y,z),\quad \frac{\partial Y}{\partial t}(x,y,z,0)=Y_1(x,y,z)$$

$$Y(0,y,z,t)=g_1(y,z,t),\quad Y(L_1,y,z,t)=g_2(y,z,t)$$

$$Y(x,0,z,t)=g_3(x,z,t),\quad Y(x,L_2,z,t)=g_4(x,z,t)$$

$$Y(x,y,0,t)=g_5(x,y,t),\quad Y(x,y,L_3,t)=g_6(\mathrm{x},\mathrm{y},t)$$

$$0\le x\le L_1,\quad 0\le y\le L_2,\quad 0\le z\le L_3,\quad t\ge 0,\quad a\ge 0$$

$$R(x,y,z,t)=U(x,y,z,t)+Y_0(x,y,z)\delta'(t)+Y_1(x,y,z)\delta(t)- \\ -a^2\delta'(x)g_1(y,z,t)-a^2\delta'(L_1-x)g_2(y,z,t)- \\ -a^2\delta'(y)g_3(x,z,t)-a^2\delta'(L_2-y)g_4(x,z,t)- \\ -a^2\delta'(z)g_5(x,y,t)-a^2\delta'(L_3-z)g_6(x,y,t) \tag{4.35}$$

$$G(x,y,z,\xi,\eta,\varsigma,t)=\frac{8}{L_1L_2L_3}\sum_{k,m,n=1}^{\infty}\sin\frac{k\pi x}{L_1}\sin\frac{m\pi y}{L_2}\sin\frac{n\pi z}{L_3}\sin\frac{k\pi\xi}{L_1}\cdot \\ \cdot\sin\frac{m\pi\eta}{L_2}\sin\frac{n\pi\varsigma}{L_3}\sin a\pi t\sqrt{\frac{k^2}{L_1^2}+\frac{m^2}{L_2^2}+\frac{n^2}{L_3^2}} \tag{4.36}$$

$$G(x,y,z,\xi,\eta,\varsigma,t) =$$

$$= \frac{8}{L_1 L_2 L_3} \sum_{k,m,n=1}^{\infty} \frac{\sin\frac{k\pi x}{L_1}\sin\frac{m\pi y}{L_2}\sin\frac{n\pi z}{L_3}\sin\frac{k\pi\xi}{L_1}\sin\frac{m\pi\eta}{L_2}\sin\frac{n\pi\varsigma}{L_3}}{s^2 + a^2\pi^2\left(\frac{k^2}{L_1^2}+\frac{m^2}{L_2^2}+\frac{n^2}{L_3^2}\right)} \tag{4.37}$$

Model 7

$$\frac{\partial^2 Y(x,t)}{\partial t^2} + a^2\frac{\partial^4 Y(x,t)}{\partial x^4} = U(x,t) \tag{4.38}$$

$$Y(x,0) = Y_0(x), \quad \frac{\partial Y}{\partial t}(x,0) = Y_1(x)$$

$$Y(0,t) = g_1(t), \quad \frac{\partial^2 Y}{\partial x^2}(0,t) = g_2(t)$$

$$Y(L,t) = g_3(t), \quad \frac{\partial^2 Y}{\partial x^2}(L,t) = g_4(t)$$

$$0 \le x \le L, \quad t \ge 0, \quad a \ne 0$$

$$\begin{aligned} R(x,t) = {} & U(x,t) + Y_0(x)\delta'(t) + Y_1(x)\delta(t) + a^2\delta'''(x)g_1(t) + \\ & + a^2\delta'(x)g_2(t) + a^2\delta'''(L-x)g_3(t) + a^2\delta'(L-x)g_4(t) \end{aligned} \tag{4.39}$$

$$G(\mathrm{x},\xi,t) = \frac{2L}{a\pi^2}\sum_{n=0}^{\infty}\frac{1}{n^2}\sin\frac{n\pi x}{L}\sin\frac{n\pi\xi}{L}\sin\frac{an^2\pi^2}{L^2}t \tag{4.40}$$

$$S(x,\xi,s) = \frac{2}{L}\sum_{n=0}^{\infty}\sin\frac{n\pi x}{L}\sin\frac{n\pi\xi}{L}\frac{1}{s^2+\frac{a^2n^4\pi^4}{L^4}} \tag{4.41}$$

4.1.2 Models for the Control of Partial Differential Equations

The mathematical theory of DPS often deals with PDE control by means of the analytical solutions of these formulations. As a demonstration, let us have a closer look

at the PDE of *Model 4* and define its LDS structure, where the standardizing function, i.e. the DDS input takes the form

$$R(x,t) = U(x,t) + Y_0(x)\delta(t) - a^2\delta'(x)g_1(t) + a^2\delta'(L-x)g_2(t) \tag{4.42}$$

and in the S-domain

$$R(x,s) = U(x,s) + Y_0(x) - a^2\delta'(x)g_1(s) + a^2\delta'(L-x)g_2(s) \tag{4.43}$$

where

$U(x,t),\ \ U(x,s)$ – distributed input

$Y(x,0) = Y_0(x)$ – initial condition

$Y(0,t) = g_1(t),\ \ Y(L,t) = g_2(t)$, and respectively $g_1(s),\ \ g_2(s)$ are boundary conditions.

Because the LDS distributed transfer function is $S(x,\xi,s)$, its distributed output subject to the initial condition $Y_0(x)$ is

$$Y(x,s) = \int_0^L S(x,\xi,s)Y_0(\xi)d\xi \tag{4.44}$$

and in time domain

$$Y(x,t) = \mathcal{G}(x,t) \otimes Y_0(x) \tag{4.45}$$

with $U(x,s) = 0,\ \ g_1(s) = 0$ and $g_2(s) = 0$

Furthermore, when subjected to the boundary conditions, the distributed output will be transformed to

$$\begin{aligned} Y(x,s) &= -a^2\int_0^L S(x,\xi,s)\delta'(\xi)g_1(s)d\xi + a^2\int_0^L S(x,\xi,s)\delta'(L-\xi)g_2(s)d\xi = \\ &= a^2\left.\frac{\partial S(x,\xi,s)}{\partial\xi}\right|_{\xi=0} g_1(s) + a^2\left.\frac{\partial S(x,\xi,s)}{\partial\xi}\right|_{\xi=L} g_2(s) \end{aligned} \tag{4.46}$$

with $U(x,s)=0$ and $Y_0(x)=0$, while keeping in mind that operations with δ - functions obey the following rules

$$\int_{0-c}^{0+c} f(\xi)\delta^{(k)}(\xi)d\xi = (-1)^k f^{(k)}(0) \text{ for } c>0 \tag{4.47}$$

$$\int_{L-c}^{L+c} f(\xi)\delta^{(k)}(\mathrm{L}-\xi)\,\mathrm{d}\xi = f^{(k)}(\mathrm{L}) \text{ for } c>0 \tag{4.48}$$

The boundary conditions are assigned to respective distributed impulse characteristics based on Eq. (4.46), that is

$$\mathcal{G}g_1(x,t),\ \ \mathcal{G}g_2(x,t) \tag{4.49}$$

and then

$$Yg_1(x,t) = \mathcal{G}g_1(x,t) \otimes g_1(t) \tag{4.50}$$

$$Yg_2(x,t) = \mathcal{G}g_2(x,t) \otimes g_2(t) \tag{4.51}$$

Finally, assuming the use of discrete sampling time, we obtain the controlled system in the HgDDS form, (Fig. 4.3).

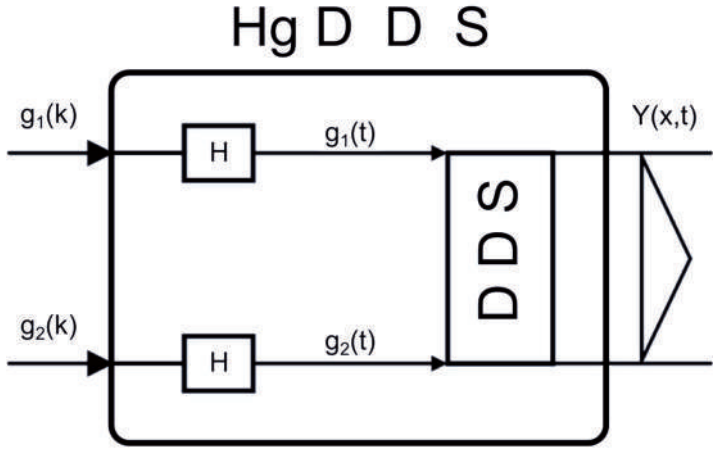


Fig. 4.3 The HgDDS block, transfer of the boundary conditions to the system output.

DDS/PDE - distributed input and distributed parameter output system/ partial differential equation.
H - zero order hold units.
$g_1(k),\ g_2(k)$ - discrete inputs of block HgDDS.

$g_1(t),\ g_2(t)$	- continuous inputs of block DDS (boundary conditions).
$Y(x,t)$	- overall continuous output.

Let us now consider a distributed input expressed by $U(x,t)=\sum_{i=1}^{n}U_i(t)T_i(x)$, where $\{T_i(x)\}_i$ are shaping units in the space domain. Then, according to Eqs. (4.1-6) and (4.45-51) and given nonzero initial and boundary conditions it is possible to express the following relation for the overall distributed output $Y(x,t)$

$$Y(x,t)=\mathcal{G}g_1(x,t)\otimes g_1(t)+\sum_{i=1}^{n}\mathcal{G}_i(x,t)\otimes U_i(t)+ \\ +\mathcal{G}g_2(x,t)\otimes g_2(t)+\mathcal{G}(x,t)\otimes Y_0(x) \tag{4.52}$$

The distributed input on the interval $[0, L]$ assumes the form $U(x,t)=\sum_{i=1}^{n}U_i(t)T_i(x)$ after inserting unity transfer functions $\{SA_i(s)=SG_i(s)=1\}_i$ in the LDS block (Fig. 4.2). In essence, the LDS block remains to be solely a DDS with shaping functions $\{T_i(x)\}_i$, which when chosen properly, can generate various classes of input functions $U(x,t)$. We obtain a special LDS block by using this procedure, in which the transformation of $U(x,t)$ takes place from a chosen class of functions to the DDS output $Y(x,t)$, considering zero initial conditions. As the input will have a discrete time nature, we obtain the H_{Ti}DDS system (Fig. 4.4) that in fact models the dynamic relation between quantities on the right and left sides of the PDE of interest. This way, the solution of PDE control is possible in the framework of LDS control.

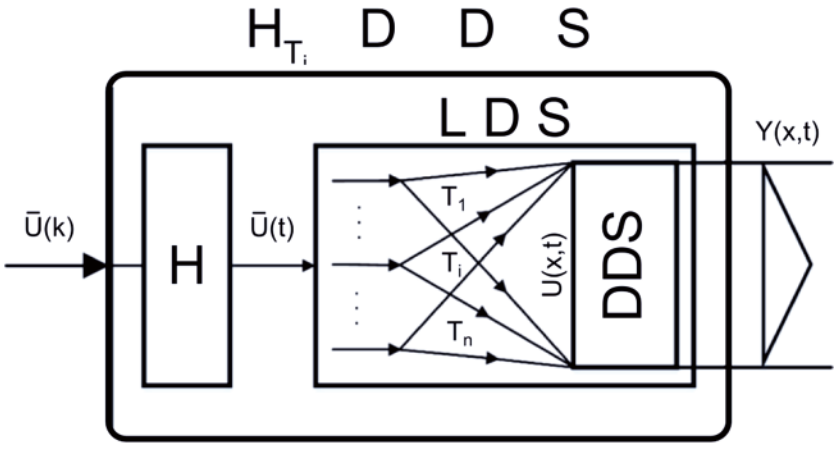


Fig. 4.4 Block H_{Ti}DDS.

DDS/PDE - distributed input and distributed parameter output system / partial differential equation.

LDS - lumped input and distributed parameter output system.

$\{T_i\}_i$ - shaping units of distributed inputs in the space domain.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\bar{U}(t)$ - vector of continuous lumped inputs.

$U(x,t)$ - overall distributed input of the DDS block.

$Y(x,t)$ - overall continuous distributed output.

4.2 Numerical Methods for Identification

The mathematical and physical analysis of technological and production processes usually leads to nonlinear PDE defined over a complex 3D geometry. These mathematical models are processed by software environments that utilize the numerical solutions of PDE. In this section, we discuss the principles of two of the most common numerical methods that are widely used in engineering computation, the finite difference method and the finite element method.

4.2.1 Finite Difference Method

The finite difference method (FDM) is a very straightforward method for the solution of PDE. Its development dates back to the 40's and 50's of the 20th century. This method is relatively easy to implement and fast in nature. The name *finite difference method* reflects the fact that it is based on approximating derivatives by *finite-size* differences (instead of infinitely small ones). Let us start the introduction to the finite difference method with the definition of the derivative of a function by stating

$$f'(x) = \frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{4.53}$$

This is approximated by a fraction of differences $\Delta f(x) = f(x+h) - f(x)$ and $\Delta x = h$

$$f'(x) \approx \frac{\Delta f(x)}{\Delta x} \frac{f(x+h) - f(x)}{h} = \frac{\Delta f(x)}{h} \tag{4.54}$$

For higher order derivatives we have

$$f^{(n)} \approx \frac{\Delta^n f(x)}{h^n} \quad n \in N \tag{4.55}$$

e.g. for $n = 2$ we get

$$f''(x) \approx \frac{\Delta^2 f(x)}{h^2} = \frac{1}{h^2}\left(f(x+2h) - 2f(x+h) + f(x)\right) \tag{4.56}$$

Usually, in simulations we use *central differences*, which are defined as follows

$$\begin{aligned} f^{(2k)}(x) &\approx \frac{1}{h^{2k}} \Delta^{2k} f(x - kh) \\ f^{(2k+1)}(x) &\approx \frac{1}{h^{2k+1}}\left(\Delta^{2k+1} f(x - (k+1)h) - \Delta^{2k+1} f(x - kh)\right) \end{aligned} \tag{4.57}$$

We will denote central differences by the symbol $\partial^n f(x) \approx f^{(n)}(x)$ in this chapter. The main principle of the finite differences method will be demonstrated using an example.

Explicit FDM

Let us consider the one dimensional heat transfer problem

$$\begin{aligned} &\frac{\partial Y}{\partial t} = D \frac{\partial^2 Y}{\partial x^2} \quad x \in [0, L], t \geq 0 \\ &Y(0,t) = Y_0 \quad Y(L,t) = Y_L \\ &Y(x,0) = v(x) \quad \Omega = [0, L] \times [0, \infty) \subset \boldsymbol{R}^2 \end{aligned} \tag{4.58}$$

where $Y = Y(x,t)$ is the system temperature at x at time t and D is the thermal diffusivity.

The definition domain of the problem Ω (system geometry) is discretized by introducing a uniformly spaced mesh of nodes. We divide the interval $[0, L]$ in the x

direction to I subintervals with a step length $h = L/I$, and for the time domain with step size k. Individual nodes are indexed with $i = 0,1,2,\ldots,I$ in the x direction and $n = 0,1,2\ldots$ in the t direction.

The definition domain is reduced to a set of nodes $\Omega_{i,n} = \{0,1,2...I\} \times \{0,1,2...\} \subset \Omega$ and the function $Y(x,t)$ will be evaluated only in these points (i,n). Next, let us denote the temperature in the node (i,n) as

$$Y(x,t) = Y(ih,nk) = Y_i^n \tag{4.59}$$

By using the Taylor expansion of functions $Y_{i+1}^n, Y_{i-1}^n, Y_i^{n+1}$ in the neighborhood of the point, (i,n) we obtain the FDM equations

$$Y_{i+1}^n = Y_i^n + h\left(\frac{\partial Y}{\partial x}\right)_i^n + \frac{1}{2}h^2\left(\frac{\partial^2 Y}{\partial x^2}\right)_i^n + \frac{1}{6}h^3\left(\frac{\partial^3 Y}{\partial x^3}\right)_i^n + O\left(h^4\right) \tag{4.60}$$

$$Y_{i-1}^n = Y_i^n - h\left(\frac{\partial Y}{\partial x}\right)_i^n + \frac{1}{2}h^2\left(\frac{\partial^2 Y}{\partial x^2}\right)_i^n - \frac{1}{6}h^3\left(\frac{\partial^3 Y}{\partial x^3}\right)_i^n + O\left(h^4\right) \tag{4.61}$$

$$Y_i^{n+1} = Y_i^n + k\left(\frac{\partial Y}{\partial t}\right)_i^n + O\left(k^2\right) \tag{4.62}$$

where $O\left(h^4\right)$ and $O\left(k^2\right)$ represent the discarded part of respective Taylor expansions and thus, the approximation error.

Adding Eq. (4.60) and Eq. (4.61) we get $Y_{i+1}^n + Y_{i-1}^n = 2Y_i^n + h^2\left(\frac{\partial^2 Y}{\partial x^2}\right)_i^n + O\left(h^4\right)$, which after rearrangement takes the form

$$\left(\frac{\partial^2 Y}{\partial x^2}\right)_i^n = \frac{Y_{i+1}^n - 2Y_i^n + Y_{i-1}^n}{h^2} + O\left(h^2\right) \tag{4.63}$$

Furthermore, Eq. (4.62) becomes

$$\left(\frac{\partial Y}{\partial t}\right)_i^n = \frac{Y_i^{n+1} - Y_i^n}{k} + O\left(k^2\right) \tag{4.64}$$

Substituting Eq. (4.63) and Eq. (4.64) to the original heat transfer equation (4.58) yields

$$\frac{Y_i^{n+1} - Y_i^n}{k} + O\left(k^2\right) = D\frac{Y_{i+1}^n - 2Y_i^n + Y_{i-1}^n}{h^2} + O\left(h^2\right) \tag{4.65}$$

from which we express Y_i^{n+1} as a function of Y_i^n and get the difference equation as

$$Y_i^{n+1} = Y_i^n + D\frac{k}{h^2}\left(Y_{i+1}^n - 2Y_i^n + Y_{i-1}^n\right) \tag{4.66}$$

Here, we neglected $O\left(k^2\right) + O\left(h^2\right)$. Equations (4.63) and (4.64) are in fact approximations of derivatives by central differences.

Given the initial conditions $Y_i^n\Big|_{n=0} = Y_i^0$, values of Y_i^{n+1} in the next time step are given by (4.66) for $i = 0,1,2,\ldots,I$. We obtain the solution $Y\left(x,t\right)$ as a time sequence of function values in the computational nodes $Y\left(x,t\right) \approx \left\{Y_i^n, i = 0,1,2\ldots I\right\}_{n\in[0,t]}$ by iterating this equation in time.

This FDM formulation is called explicit, as each value Y_i^{n+1} is expressed depending on past values of $Y_i^n\left(i = 0,1,2,\ldots,I\right)$ (from the previous time step) and not on actual values $Y\left(x,t\right)$ in neighboring nodes, i.e. Y_{i+1}^{n+1} and Y_{i-1}^{n+1}.

Computational precision depends on how one choses *h* and *k*. Neglecting the remainder of Taylor expansions $O\left(k^2\right) + O\left(h^2\right)$ affects the overall approximation error (the whole result), however, this can be minimized by carefully choosing *h* and *k*. Although the smaller these values are, the more we approach the exact solution $\Delta x \to 0 \Rightarrow \partial f \to f'$, the computational mesh gets denser and denser, which of course significantly increases computation time.

Still, one has to keep in mind that the choice of *h* and *k* is not completely arbitrary, since the stability of the solution depends on them. This is only guaranteed when the so-called *Neumann stability condition* is fulfilled.

Neumann Stability Condition

Let us rewrite (4.66) in the form of

$$Y_i^{n+1} = \lambda Y_{i-1}^n + (1-2\lambda) Y_i^n + \lambda Y_{i+1}^n \equiv \left(E_{kh} Y^n\right) \tag{4.67}$$

where $\lambda = D\dfrac{k}{h^2}$. The identity (4.67) yields the operator E_{kh} and the solution can be expressed as

$$Y_i^n = \left(E_{kh}^n Y^0\right)_i \tag{4.68}$$

If the coefficients of E_{kh} are non-negative and their sum equals one, then

$$\sup_i \left|Y_i^{n+1}\right| \le \sup_i \left|Y_i^n\right| \tag{4.69}$$

Furthermore, we introduce

$$\|v\| = \sup_i |v_i| \tag{4.70}$$

which implies

$$\left\|Y^{n+1}\right\| = \left\|E_{kh} Y^n\right\| \le \left\|Y^n\right\| \tag{4.71}$$

and finally

$$\left\|Y^n\right\| = \left\|E_{kh}^n v\right\| \le \|v\| \tag{4.72}$$

Thus, the operator E_{kh} is stable, and consequently, the solution is stable as well. In general, E_{kh} has the form

$$Y_i^{n+1} = E_{kh} Y_i^n = \sum_p a_p Y_{i-p}^n \tag{4.73}$$

where $a_p = a_p(\lambda)$ and the sum is finite.

We need to investigate the characteristic polynomial of E_{kh} regarding stability

$$E(\xi) = \sum_p a_p e^{-jp\xi} \tag{4.74}$$

Theorem: (Neumann condition) The necessary condition for the stability of operator E_{kh} is, that its characteristic polynomial fulfils the following inequality

$$\left|E(\xi)\right| \leq 1 \quad \forall \xi \in \boldsymbol{R} \tag{4.75}$$

Now, let us apply this theorem on (4.67), so we get

$$E(\xi) = \lambda e^{-j\xi} + (1-2\lambda) + \lambda e^{j\xi} = 1 - 2\lambda + 2\lambda\cos\xi = 1 - 4\lambda \sin^2 \frac{\xi}{2} \tag{4.76}$$

The function $\sin^2 x$ is periodic with a maximum equal to 1. To fulfil the condition (4.75), the following must hold

$$\max_{\xi} \left|1 - 4\lambda \sin^2 \frac{\xi}{2}\right| = \left|1 - 4\lambda\right| \leq 1 \tag{4.77}$$

which directly implies $\lambda \leq 1/2$.

Restrictions on discretization step length depend on the problem formulation, type and form of governing PDE, etc. By carefully choosing the discretization method we can even achieve unconditional stability (convergence of the solution), as in case of some implicit methods.

Implicit FDM

Let us now rewrite the problem (4.66) in an implicit form

$$Y_i^{n+1} = Y_i^n + \lambda\left(Y_{i+1}^{n+1} - 2Y_i^{n+1} + Y_{i-1}^{n+1}\right) \tag{4.78}$$

After rearrangement we get

$$Y_i^{n+1} + 2\lambda Y_i^{n+1} - \lambda Y_{i+1}^{n+1} - \lambda Y_{i-1}^{n+1} = Y_i^n \tag{4.79}$$

which can be put in matrix form

$$(\boldsymbol{I} - \lambda \boldsymbol{K})\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^n \tag{4.80}$$

where $\boldsymbol{Y}^n = \left(Y_i^n\right)_{i=0,1,2\ldots I}$ and the matrix

$$\boldsymbol{K}_{I\times I} = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & \vdots \\ 0 & 0 & 1 & \ddots & 1 \\ 0 & 0 & \ldots & 1 & -2 \end{bmatrix}$$

is tridiagonal. We get the solution in the form

$$\boldsymbol{Y}^{n+1} = \left(\boldsymbol{I} - \lambda \boldsymbol{K}\right)^{-1} \boldsymbol{Y}^n \tag{4.81}$$

The E_{kh} operator as in (4.73) in this case becomes

$$E_{kh}\boldsymbol{Y}^n = \left(\boldsymbol{I} - \lambda \boldsymbol{K}\right)^{-1} \boldsymbol{Y}^n \tag{4.82}$$

and the characteristic polynomial of the operator E_{kh} for the problem (4.79) then is

$$E\left(\xi\right) = \frac{1}{1 - \lambda e^{-j\xi} + 2\lambda - \lambda e^{j\xi}} = \frac{1}{1 + 2\lambda\left(1 - \cos\xi\right)} \tag{4.83}$$

which fulfils the condition (4.75)

$$\max_{\xi} \left| \frac{1}{1 + 2\lambda\left(1 - \cos\xi\right)} \right| = 1 \quad \forall \lambda \in \boldsymbol{R} \tag{4.84}$$

classifying the method as unconditionally stable.

Implicit methods, however, are much more computation-intensive as the entire space domain has to be processed at once during each time step. For more on the subject, see Rektorys (1999) and Smith (2004). Now, we briefly show how the problem is solved in the case of a 3D body.

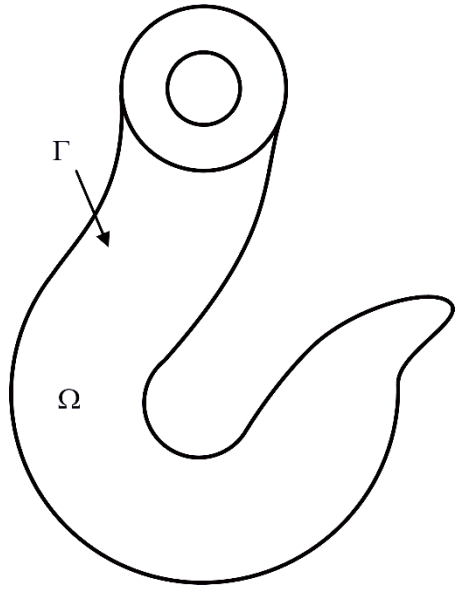


Fig. 4.5 Definition domain Ω (geometry of interest).

Let us again solve the heat transfer problem, this time in all Cartesian directions

$$\begin{aligned} &\frac{\partial Y}{dt} = D\nabla^2 Y, \quad (x,y,z) \in \Omega \subset \boldsymbol{R}^3, t \geq 0 \\ &Y\left(x,y,z,0\right) = v, \quad Q = \Omega \times \left[0,\infty\right) \subset \boldsymbol{R}^4 \end{aligned} \tag{4.85}$$

A computational mesh is created by dividing Ω uniformly in each direction – x, y and z with step length h and the time step s. The resulting space-domain model of the problem created by FDM is shown in Fig. 4.5.

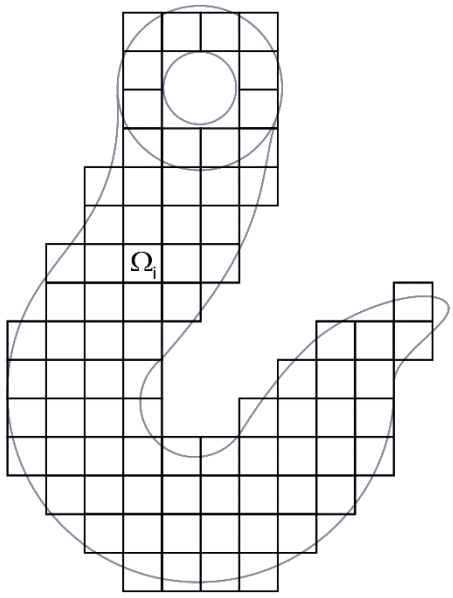


Fig. 4.6 Discretized model of Ω.

Next, we derive the differential equation similarly as in the 1D case. After algebraic manipulation we finally get (in explicit form here)

$$\begin{aligned} Y_{i,j,k}^{n+1} &= \ldots \\ &= Y_{i,j,k}^{n} + D\frac{s}{h^2}\left(Y_{i+1,j,k}^{n} + Y_{i,j+1,k}^{n} + Y_{i,j,k+1}^{n} - 6Y_{i,j,k}^{n} + Y_{i-1,j,k}^{n} + Y_{i,j-1,k}^{n} + Y_{i,j,k-1}^{n}\right) \end{aligned} \tag{4.86}$$

Its implicit form is created using the same approach as in Eq. (4.80), however, assembling the matrix $\boldsymbol{K}$ is not such a trivial task anymore.

Simplicity, fast calculation and low memory requirements (in case of explicit forms) all add to the popularity of FDM. However, its drawback is that we get solutions only in nodes and information in between them is not available, thus has to be estimated by additional means, if needed. Choosing discretization steps carefully might eliminate this problem, nevertheless, depending on the required precision (which in most cases imposes much more stringent requirements than the stability condition) the mesh might get too dense and the computation too slow. In this case other methods must be sought for, where the mesh density is lower while ensuring the same precision, e.g. the finite element method.

4.2.2 Finite Element Method

The development of the Finite Element Method also dates back to the 50's of 20th century, however, it was used rather seldom until 1968, when *Zlámal* (1968) published his paper about the error analysis of the method. Nowadays, it is a very well established approach and is one of the most popular in engineering and technical computing.

The definition domain is divided into a finite number of subsets – *elements*

$$\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i, \quad \Omega_i \cap \Omega_j \big|_{i \neq j} = \varnothing, \quad \bar{\Omega}_i \cap \bar{\Omega}_j \big|_{i \neq j} = \Gamma_{ij} \tag{4.87}$$

where Γ_{ij} is the set of common nodes, edges, surfaces of neighboring elements, i.e. the border between them.

Inside the element, in between the nodes all continuous quantities will be approximated by so-called *basis functions*. The principle is explained below Fig. 4.7.

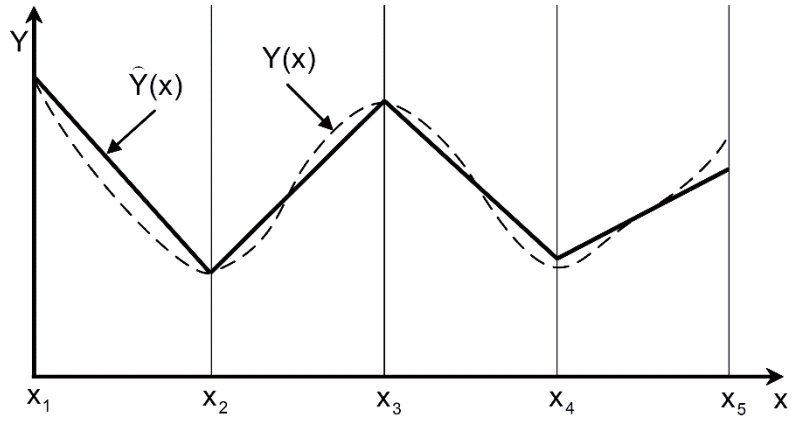


Fig. 4.7 Approximation of the function $Y(x) \approx \hat{Y}(x)$.

For the sake of simplicity, let us consider the 1D case as shown in the figure above. We subdivide the interval $[0, L]$ in the x-direction to N sub-intervals and get $N+1$ nodes x_i. Let us denote the elements $E_e = \left[x_1^e, x_2^e \right] = \left[x_e, x_{e+1} \right] \equiv \bar{\Omega}_e$, $e = 1, 2, \ldots, N$. Now, in each sub-interval we approximate the function $Y(x) \approx \hat{Y}(x)$, which represents e.g. the temperature field along a bar $[0, L]$. This operation has to preserve the continuity of the original function in the approximating one, thus the condition $\hat{Y}\left(x_2^e\right) = \hat{Y}\left(x_1^{e+1}\right)$ has to be fulfilled.

Basis Functions

Let us define $\hat{Y}(x)$ on the element E_e

$$Y^e(\xi) = (1-\xi)Y_1^e + \xi Y_2^e \tag{4.88}$$

where $\xi \in [0,1]$ is a reduced coordinate along $\hat{Y}(x)$ of element E_e, $Y_1^e = \hat{Y}(x_1^e)$, $Y_2^e = \hat{Y}(x_2^e)$ are function values in the element nodes. Let us denote

$$\begin{aligned} \varphi_1(\xi) &= (1-\xi) \\ \varphi_2(\xi) &= \xi \end{aligned} \tag{4.89}$$

and after substitution we get

$$Y^e(\xi) = \varphi_1(\xi)Y_1^e + \varphi_2(\xi)Y_2^e \tag{4.90}$$

The functions $\varphi_1(\xi)$ and $\varphi_2(\xi)$ are called *basis functions* and they are associated with the nodal parameters Y_1^e and Y_2^e.

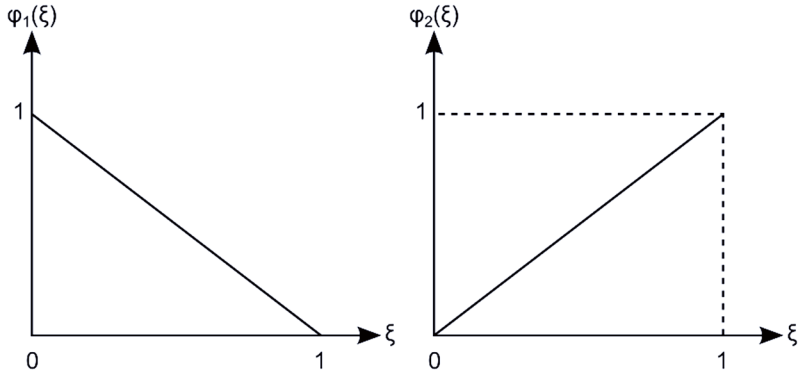


Fig. 4.8 Linear basis functions $\varphi_1(\xi)$ and $\varphi_2(\xi)$.

We used local node numbering above and since computations are made in a global context, it is useful to introduce the relation between global (Y_i) and local (Y_n^e) node numbers

$$Y_n^e = Y_{\Delta(n,e)} \tag{4.91}$$

where $\Delta(n,e)$ gives the global node number of Y_n^e.

For our demonstration example, we obtain the matrix

$$\Delta_{n\times e}=\begin{bmatrix}1 & 2 & 3 & \dots & N\\ 2 & 3 & 4 & \dots & N+1\end{bmatrix} \tag{4.92}$$

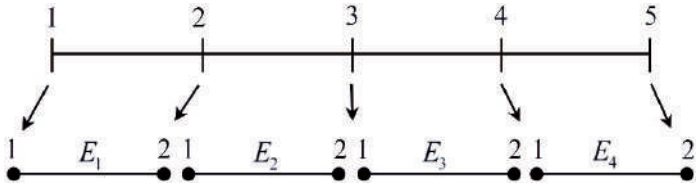


Fig. 4.9 Relation between global and local node numbering.

A basis function may be understood as weighting for nodal parameters. They provide information about the fraction each nodal parameter contributes to the function value $\hat{Y}(x)$ at any location within the element.

The function (4.88) in nodes „1“ and „2“ takes the value of the respective nodal parameter Y_1^e and Y_2^e, in between it is a linear combination of these, while going from „1“ to „2“ the weight of Y_1^e decreases and the weight of Y_2^e increases. In the same way we may define interpolation for *x* according to

$$x^e(\xi)=\varphi_1(\xi)x_1^e+\varphi_2(\xi)x_2^e \tag{4.93}$$

as well. In general, the following relation holds

$$x^e(\xi)=\sum_n \varphi_n(\xi)x_n^e \tag{4.94}$$

$$Y^e(\xi)=\sum_n \varphi_n(\xi)Y_n^e \tag{4.95}$$

where $n=\{1,2,\dots,N_u\}$ and N_u is the number of nodes of the element.

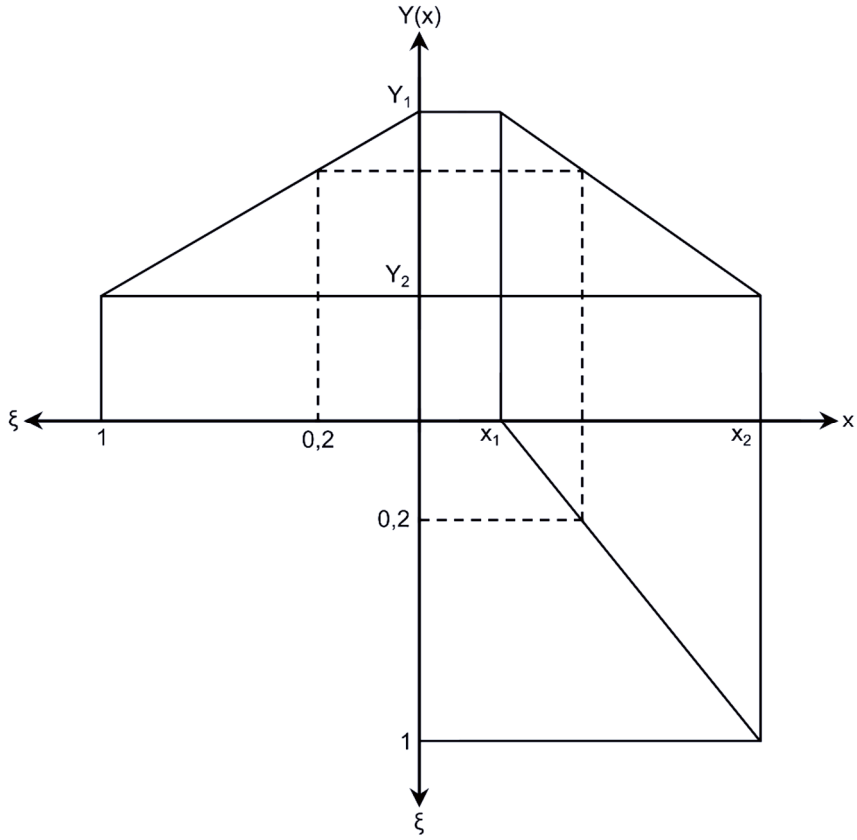


Fig. 4.10 Relation between Y and x via the normalized coordinate ξ of element E_e in case of linear basis functions.

By applying this methodology for 3D systems we eventually get *trilinear* basis functions. In fact, we just expand Eq. (4.94) and Eq. (4.95)

$$\begin{aligned} x^e(\xi_1,\xi_2,\xi_3) &= \sum_n \varphi_n(\xi_1,\xi_2,\xi_3)\, x_n^e \\ y^e(\xi_1,\xi_2,\xi_3) &= \sum_n \varphi_n(\xi_1,\xi_2,\xi_3)\, y_n^e \\ z^e(\xi_1,\xi_2,\xi_3) &= \sum_n \varphi_n(\xi_1,\xi_2,\xi_3)\, z_n^e \\ Y^e(\xi_1,\xi_2,\xi_3) &= \sum_n \varphi_n(\xi_1,\xi_2,\xi_3)\, Y_n^e \end{aligned} \tag{4.96}$$

For example, for an eight-node element we have the following basis functions

$$\begin{aligned}
&\varphi_1(\xi_1,\xi_2,\xi_3) = (1-\xi_1)(1-\xi_2)(1-\xi_3) \quad && \varphi_2(\xi_1,\xi_2,\xi_3) = \xi_1(1-\xi_2)(1-\xi_3)\\
&\varphi_3(\xi_1,\xi_2,\xi_3) = (1-\xi_1)\xi_2(1-\xi_3) && \varphi_4(\xi_1,\xi_2,\xi_3) = \xi_1\xi_2(1-\xi_3)\\
&\varphi_5(\xi_1,\xi_2,\xi_3) = (1-\xi_1)(1-\xi_2)\xi_3 && \varphi_6(\xi_1,\xi_2,\xi_3) = \xi_1(1-\xi_2)\xi_3\\
&\varphi_7(\xi_1,\xi_2,\xi_3) = (1-\xi_1)\xi_2\xi_3 && \varphi_8(\xi_1,\xi_2,\xi_3) = \xi_1\xi_2\xi_3
\end{aligned} \tag{4.97}$$

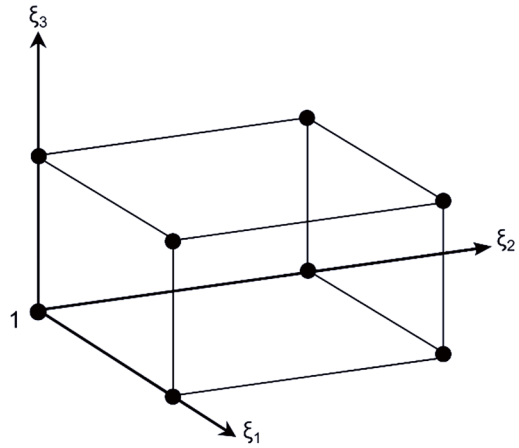


Fig. 4.11 Eight-node trilinear element.

The choice of basis function and its type depends not only on the required precision but also on the order of the modelled PDE, where one has to ensure the continuity of derivatives of basis functions at the nodes as well. Many types are available and can be used; the most popular are quadratic, cubic (*Hermite*) functions, splines etc.

Triangular and Tetrahedral Elements

The most popular and most frequently used element types are triangles for 2D and tetrahedrons for 3D geometry. Their main advantage is that they do not impose any limits to modelled shapes, i.e. are the most universal ones.

Definition: *n-simplex* in $\boldsymbol{R}^n$ is a convex hull of $(n+1)$ points $a_j = \left(a_{ij}\right)_{i=1}^{n} \in \boldsymbol{R}^n$, which are called the vertices, for which the matrix

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n+1} \\ a_{21} & a_{22} & \dots & a_{2n+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn+1} \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{4.98}$$

is regular, i.e. the points a_j are not located on the same hyperplane (for 2D on a line, for 3D on a plane).

According to this definition, the triangle is a 2-simplex and the tetrahedron a 3-simplex. For triangular and tetrahedral elements we define the so-called *barycentric* coordinates (with respect to the center of mass).

Definition: Barycentric coordinates $\xi_j = \xi_j(x), 1 \le j \le n+1$ of any point $\boldsymbol{x} \in \boldsymbol{R}^n$, with respect to the points a_j are defined as the only solution of the linear system

$$\boldsymbol{A}\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \tag{4.99}$$

from which

$$\boldsymbol{\xi} = \boldsymbol{A}^{-1} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \tag{4.100}$$

Let us solve the system (4.99) for a triangle. Making use of Cramer's rule we get

$$\xi_1(x,y) = \frac{\begin{vmatrix} x & x_2 & x_3 \\ y & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}} = \frac{S_{\Delta x23}}{S_{\Delta 123}} \tag{4.101}$$

Similarly, for the remaining coordinates

$$\xi_2(x,y) = \frac{S_{\Delta 1x3}}{S_{\Delta 123}}$$
$$\xi_3(x,y) = \frac{S_{\Delta 12x}}{S_{\Delta 123}} \tag{4.102}$$

where $S_{\Delta\ldots}$ is the area of respective triangles. Figure 4.12 clarifies the matter.

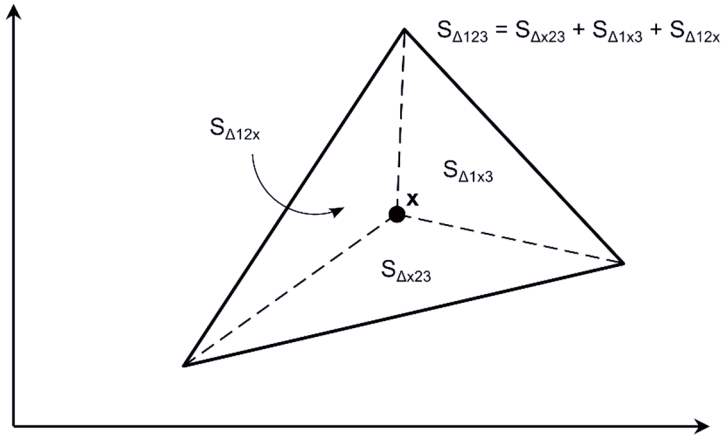


Fig. 4.12 Definition of barycentric coordinates of a triangular element.

The equations (4.99) also imply the following property of coordinates

$$\sum_{j=1}^{n+1} \xi_j = 1 \tag{4.103}$$

and, for the center of mass we have

$$\xi_1 = \xi_2 = \ldots = \xi_n = \xi_{n+1} = \frac{1}{n+1}$$

Accordingly, linear basis functions for the triangular element are the following

$$\varphi_1(\xi_1,\xi_2) = \xi_1$$
$$\varphi_2(\xi_1,\xi_2) = \xi_2 \tag{4.104}$$
$$\varphi_3(\xi_1,\xi_2) = \xi_3 = 1 - \xi_1 - \xi_2$$

The same procedure applied to a tetrahedron yields

$$\begin{aligned}
\varphi_1(\xi_1,\xi_2,\xi_3) &= \xi_1 \\
\varphi_2(\xi_1,\xi_2,\xi_3) &= \xi_2 \\
\varphi_3(\xi_1,\xi_2,\xi_3) &= \xi_3 \\
\varphi_4(\xi_1,\xi_2,\xi_3) &= 1-\xi_1-\xi_2-\xi_3
\end{aligned} \tag{4.105}$$

Transformation of local coordinates to global coordinates is defined by means of the *Jacobian* matrix $\boldsymbol{J}$

$$\boldsymbol{x} = \boldsymbol{J}\boldsymbol{\xi} \tag{4.106}$$

where

$$\boldsymbol{J} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} = \left(\frac{\partial x_i}{\partial \xi_j} \right)_{i,j} \tag{4.107}$$

For the tetrahedron it becomes

$$\boldsymbol{J} = \begin{bmatrix} x_1 - x_4 & x_2 - x_4 & x_3 - x_4 \\ y_1 - y_4 & y_2 - y_4 & y_3 - y_4 \\ z_1 - z_4 & z_2 - z_4 & z_3 - z_4 \end{bmatrix} \tag{4.108}$$

Finally, the backward transformation

$$\boldsymbol{\xi} = \boldsymbol{J}^{-1}\boldsymbol{x} = \frac{\boldsymbol{J}^{\mathrm{T}}}{6V_{\mathrm{t}}}\boldsymbol{x} \tag{4.109}$$

where $V_{\mathrm{t}} = |\boldsymbol{J}|/6$ is the tetrahedron volume.

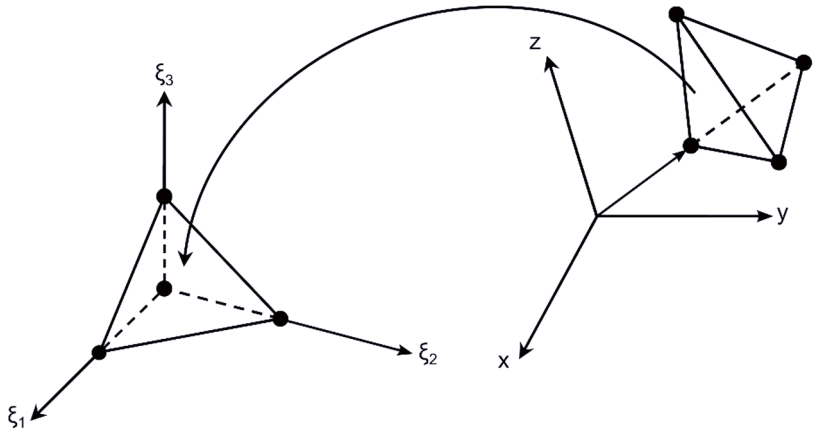


Fig. 4.13 General and reference tetrahedron.

Remark: Sometimes it happens that the FEM mesh generator creates elements with very small volumes, which might become zero due to round off errors of the particular computer system. Thus, at the given system precision, their Jacobians might become singular. This problem is to be kept in mind.

General Problem Formulation

Let V be a Hilbert space, $a(\cdot,\cdot)$ an elliptic bilinear form and $L \in V^*$ a linear functional. Let us try to solve the variational problem

$$a(Y,v) = L(v), \quad \forall v \in V \tag{4.110}$$

Furthermore, let V_h be a finite dimensional subspace of V. We look for an approximate solution

$$a(Y_h,v_h) = L(v_h), \quad \forall v_h \in V_h \tag{4.111}$$

According to the *Lax-Milgram* theorem (see next section), there exists a unique solution, which we denote as Y_h.

The solution of the problem (4.110) is equivalent to the solution of the matrix equation

$$\boldsymbol{A}\boldsymbol{Y} = \boldsymbol{u} \tag{4.112}$$

where $\boldsymbol{A}=\left(a\left(\varphi_i,\varphi_j\right)\right)_{i,j}$, $i=1,2\ldots N(h)$, $j=1,2\ldots N(h)$, $\{\varphi_i\}_{i=1,2\ldots N(h)}$ is the basis of subspace V_h, $\boldsymbol{Y}=\begin{bmatrix}\alpha_1 & \alpha_2 & \ldots & \alpha_{N(h)}\end{bmatrix}$, $N(h)$ is number of element nodes, i.e. dimension of V_h and $\boldsymbol{u}=\begin{bmatrix}L(\varphi_1) & L(\varphi_2) & \ldots & L\left(\varphi_{N(h)}\right)\end{bmatrix}$.

As mentioned before, (4.95), Y_h can be expressed as

$$Y_h=\sum_{i=1}^{N(h)}\alpha_i\varphi_i \tag{4.113}$$

In case of mechanical problems $\boldsymbol{A}$ is the stiffness matrix and $\boldsymbol{u}$ the load vector.

Error Estimation

Now assume V_h is a finite dimensional subspace of $\boldsymbol{V}$, $a_h(\cdot,\cdot)$ a bilinear form and $L_h(\cdot)$ is a linear functional, both defined only on V_h. The goal is to find the differences between Y_h and Y. These may be obtained by solving the variational problems

$$a_h\left(Y_h,v_h\right)=L_h\left(v_h\right) \tag{4.114}$$

and

$$a\left(Y,v\right)=L\left(v\right) \tag{4.115}$$

where $a_h(\cdot,\cdot)$, $a(\cdot,\cdot)$, $L_h(\cdot)$, $L(\cdot)$ fulfill conditions given by the Lax-Milgram theorem, (Ciarlet (2002)).

Definition: Let $\|\cdot\|=\sqrt{\langle\cdot,\cdot\rangle_V}$. The form $a_h(\cdot,\cdot)$ associated to the set of subspaces V_h is called uniformly V_h-elliptic, if there exists $\alpha>0$ independent on subspaces V_h such that $a_h\left(Y_h,v_h\right)\geq\alpha\|v_h\|^2$.

Theorem: (Lax-Milgram theorem) let $\boldsymbol{V}$ be a Hilbert space, $a(\cdot,\cdot):V\times V\rightarrow\boldsymbol{R}$ a continuous V-elliptic form and $L(\cdot):V\rightarrow\boldsymbol{R}$ a continuous linear functional. Then the variational problem $a\left(Y,v\right)=L\left(v\right)\quad\forall v\in V$ has a unique solution.

To be able to analyze error estimation, the convergence condition has to be fulfilled first

$$\lim_{n\to\infty}\left\|Y-Y_n\right\|=0 \tag{4.116}$$

The error estimation is based on *Strang's theorem*, see Siddiqi **(**1993**)**.

Theorem: (first Strang's theorem) Consider a set of problems defined by Eq. (4.114), for which the bilinear forms $a_h\left(\cdot,\cdot\right)$ are uniformly V_h-elliptic.

There exists then a number C independent on the space V_h, such that

$$\left\|Y-Y_h\right\|\le C\left(\inf_{v_h\in V_h}\left\{\left\|Y-v_h\right\|+\sup_{w_h\in V_h}\frac{\left|a\left(v_h,w_h\right)-a_h\left(v_h,w_h\right)\right|}{\left\|w_h\right\|}\right\}+\right.$$
$$\left.+\sup_{w_h\in V_h}\frac{\left|L\left(w_h\right)-L_h\left(w_h\right)\right|}{\left\|w_h\right\|}\right) \tag{4.117}$$

In case $a_h\left(\cdot,\cdot\right)=a\left(\cdot,\cdot\right)$ and $L_h\left(\cdot,\cdot\right)=L\left(\cdot,\cdot\right)$, which actually is problem (4.110), this expression becomes much simpler and we get *Céa's theorem*, (Siddiqi (1993))

$$\left\|Y-Y_h\right\|_V\le C\inf_{v_h\in V_h}\left\|Y-v_h\right\|_V . \tag{4.118}$$

Problem Solution, Galerkin Formulation

Let us solve the following general linear elliptic PDE

$$K\left(Y\right)=b \tag{4.119}$$

For an approximate solution in the form of Eq. (4.113) we express the residual of this solution (deviation from the exact value) as

$$K\left(Y_h\right)-b=R\ne 0 \tag{4.120}$$

We seek such a solution that the residual is uniformly distributed across the whole subspace V_h

$$\sum_{j=1}^{N(h)}\int_{V_h}R\psi_j dV=0 \tag{4.121}$$

where ψ_j are weights of the residual. If the weights are chosen to be the basis functions of V_h, i.e. $\psi_j = \varphi_j$, $j = 1,2\ldots N(h)$, we arrive at the *Galerkin* formulation (also called weighted residual method)

$$\left\langle R, \varphi_j \right\rangle_{L_2(\Omega)} = \int_\Omega R\varphi_j \, d\Omega = 0 \quad \forall j \tag{4.122}$$

We ensure the residual to be orthogonal to the base of V_h, hence minimizing its contribution Y_h. Let us once more solve the heat transfer problem

$$-\nabla \cdot (k \nabla Y) = 0 \tag{4.123}$$

where k is thermal diffusivity. We solve the problem on the domain Ω with the boundary Γ.

For any basis function v the right hand side of Eq. (4.122) becomes

$$\int_\Omega Rv \, d\Omega = \int_\Omega -\nabla \cdot (k \nabla Y) v \, d\Omega = 0 \tag{4.124}$$

Applying the *Green-Gauss* theorem as in Madelunk (1975), we obtain

$$\int_\Omega \nabla \cdot (-k \nabla Y) v \, d\Omega = \int_\Omega k \nabla Y \cdot \nabla v \, d\Omega - \int_\Gamma k \frac{\partial Y}{\partial \boldsymbol{n}} v \, d\Gamma \tag{4.125}$$

where $\boldsymbol{n}$ is the unit normal vector of the surface $d\Gamma$

$$\begin{aligned} &\int_\Omega \nabla \cdot (-k \nabla Y) v \, d\Omega = \left\langle \nabla \cdot (-k \nabla Y), v \right\rangle_{L_2(\Omega)} \\ &\left\langle \nabla \cdot (-k \nabla Y), v \right\rangle_{L_2(\Omega)} = \left\langle -k \nabla Y, \nabla v \right\rangle_{L_2(\Omega)} - \int_\Gamma k \frac{\partial Y}{\partial \boldsymbol{n}} v \, d\Gamma \end{aligned} \tag{4.126}$$

Let us define the bilinear form

$$a(Y, v) = \left\langle \nabla \cdot (-k \nabla Y), v \right\rangle_{L_2(\Omega)} \tag{4.127}$$

Then the functional (4.110) becomes $L(v) = 0$. This problem might be reformulated as

$$J(Y_h) = \inf_{v_h \in V_h} \left(\frac{1}{2} a(Y_h, v_h) - L(v_h) \right) \tag{4.128}$$

The unique solution of Eq. (4.128) is equivalent to the solution of Eq. (4.110) as well as Eq. (4.120). The problem is to find the solution giving the smallest residuum (i.e. being the closest to the exact one) at a given mesh density. The latter mentioned formulation is the most popular one in engineering practice. ***J*** is often called the energy functional, then the problem is understood as finding the energetic minimum. This formulation is also equivalent to the so-called *virtual work principle*.

Now, let us divide the geometry of interest to elements. Nowadays, there is a vast number of mesh generation algorithms available and their description is out of the scope of this book. Mostly, they are already implemented in various FEM software packages, like ANSYS, COMSOL, ProCAST, etc. For more on FEM mesh generation see Zienkiewicz (2005), Frey (2008) or Ho-Le (1988).

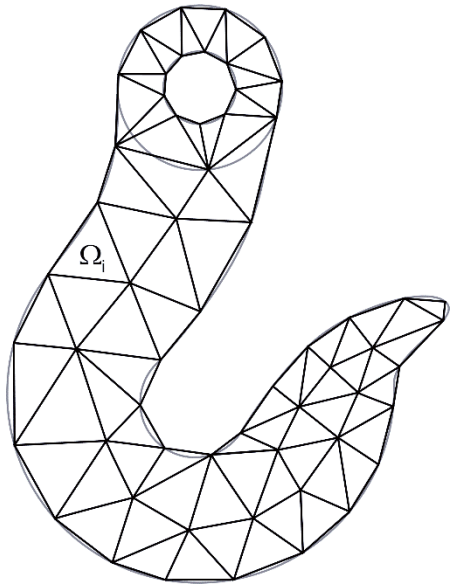


Fig. 4.14 Discretized definition domain Ω.

The integral (4.124) takes the form

$$\int_{\Omega} -\nabla\cdot\left(k\nabla Y\right)v\,d\Omega = \sum_{e=1}^{E}\int_{\Omega_e} -\nabla\cdot\left(k\nabla Y\right)v\,d\Omega_e \tag{4.129}$$

where E is number of elements. These integrals will be evaluated for each element individually. Using the Galerkin formulation and substituting Eq. (4.95) into Eq. (4.125) we get for the element

$$\sum_{j=1}^{N(E)} Y_j^e \int_{\Omega_e} k\left(\frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x}+\frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y}+\frac{\partial\varphi_i}{\partial z}\frac{\partial\varphi_j}{\partial z}\right)d\Omega_e = \int_{\Gamma_e} k\frac{\partial Y^e}{\partial \boldsymbol{n}}\varphi_i\,d\Gamma_e \tag{4.130}$$

where $i = 1,2 \ldots N(E)$, $N(E)$ is the number of element nodes. The integrals are evaluated in local coordinates and the transformation of derivatives from global context to local context is done using the Jacobian mentioned before.

Rewriting Eq. (4.130) we get

$$\sum_{j=1}^{N} A_{ij}^{e} Y_{j}^{e} = u_{i}^{e} \tag{4.131}$$

which actually is the matrix equation

$$\boldsymbol{A}^{e} \boldsymbol{Y}^{e} = \boldsymbol{u}^{e} \tag{4.132}$$

for the element *e*. One of the advantages of the Galerkin method is that it creates a symmetric stiffness matrix $\boldsymbol{A}^{e}$. As expected, the aforementioned integrals cannot be treated analytically and have to be solved numerically, e.g. by *Gauss quadrature* as outlined by Moin (2010).

What remains now is the assembly of the global matrices $\boldsymbol{A}$, $\boldsymbol{Y}$ and $\boldsymbol{u}$. The vector $\boldsymbol{Y}$ contains values of function *Y* at the nodes in the global context, so the numbering is global. *N* is the number of nodes in the region $\bar{\Omega}$

$$\boldsymbol{Y}^{\mathrm{T}} = \begin{bmatrix} Y_1 & Y_2 & \ldots & Y_N \end{bmatrix} \tag{4.133}$$

The load vector $\boldsymbol{u}$ contains inputs to the nodes as well as boundary conditions in the global context

$$\boldsymbol{u}^{\mathrm{T}} = \begin{bmatrix} u_1 & u_2 & \ldots & u_N \end{bmatrix} \tag{4.134}$$

Now let us assemble the stiffness matrix $\boldsymbol{A}$. First, we prepare augmented stiffness matrices of size $(N \times N)$ for each of the elements

$$A_{\mathrm{Glob}\,p,q}^{e} = A_{ij}^{e}, \quad i, j = 1,2 \ldots N(E) \tag{4.135}$$

where $p = \Delta(i,e), q = \Delta(j,e)$, are global node numbers of *i*-th and *j*-th node of element *e*. The remaining positions in the matrix are zero.

An example of augmented stiffness matrix for a two-node element it looks like

$$A^e_{\text{Glob}} = \begin{array}{c} \begin{matrix} \quad & p \downarrow & \quad & q \downarrow & \quad \end{matrix} \\ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & A^e_{11} & \cdot & A^e_{12} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & A^e_{21} & \cdot & A^e_{22} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \end{array} \begin{matrix} \\ \\ \leftarrow p \\ \\ \leftarrow q \end{matrix} \tag{4.136}$$

The global stiffness matrix is a sum of individual augmented stiffness matrices of elements

$$\boldsymbol{A} = \sum_{e=1}^{E} \boldsymbol{A}^e_{\text{Glob}} \tag{4.137}$$

Finally we solve the linear system

$$\boldsymbol{A}\,\boldsymbol{Y} = \boldsymbol{u} \tag{4.138}$$

Contrary to FDM where function values are evaluated only at computational nodes, FEM estimates these in any location throughout the definition domain without additional procedures needed.

Both methods have their advantages and disadvantages. Which is more suitable to a particular problem depends on many factors – complexity of the problem (order of PDE, geometry), required modelling precision, available computation time, etc. One can improve precision by densening the computational mesh, however, it is not always worth the effort as the approximate solution approaches the exact one asymptotically in an exponential manner, thus when a higher precision is required, another method or problem reformulation is sought for.

FEM discretization is used in the majority of engineering and technical computing for the space domain and FDM for the time domain. This approach is also utilized in well-established simulation packages like ANSYS, COMSOL Multiphysics, ProCAST and others.

If high speed and simplicity are the main concern, rather than high precision, FDM is successfully utilized for modelling of the space domain as well (e.g. Shoumeia (2002)), like in QuickCAST.

For the sake of completeness, we mention another, albeit less often used approach: the *boundary element method* (Hunter (2002), Wrobel (2002), Alonso (2006)); which

is based on the same principle as FEM with the difference that the FEM mesh is only done on the boundary of the object of interest. This significantly reduces the computational effort and memory requirements, although the solution is obtained for the surface only. The relation between the state at the boundary and in the bulk is sometimes possible to estimate, e.g. by applying the Green-Gauss theorem. This method is most suitable for problems in which the ratio of the system surface to its volume is large and the inside is not of much interest.

4.3 Experimental System Identification

If the distributed manipulated variable is measurable using a scanner or other contact-less inspection and measurement method, it is possible to employ experimental identification. The measurement points are chosen on the definition domain, in a one dimensional case [0, *L*] according to Fig. 4.15.

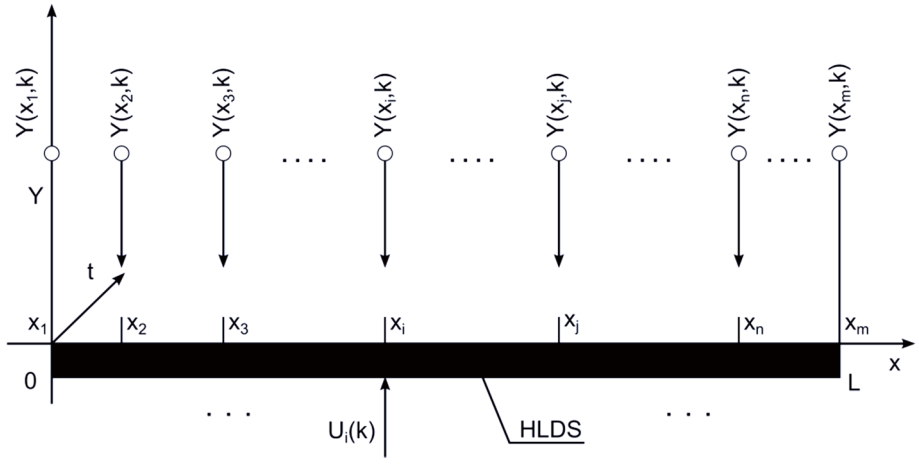


Fig. 4.15 Layout of measurement points at the output of HLDS.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$U_i(k)$ - *i*-th discrete lumped input.

$\{Y(x_j,k)\}_{j=1,m}$ - distributed output in selected points $\{x_j\}_{j=1,m}$, variable *k*.

We obtain partial distributed step responses $\{\mathcal{H}H_i(x_j,k)\}_{i,j,k}$ in locations $\{x_j\}_j$ by applying individually acting discrete step changes in lumped inputs $\{U_i(k)\}_i$ in the linearized neighborhood of the selected steady-state operating mode. These are further

processed to obtain time and space domain response characteristics. These are given by $\{\mathcal{H}H_i(x_i,k)\}_{i,k}$, $\{SH_i(x_i,z)\}_i$, etc. in the time domain. The space domain characteristics are estimated by spline approximation.

The number and location of measurement points determines the final model precision, or conversely, the number and location of measurements are to be designed in order to achieve a desired precision of the identified model. The method used to analyze model precision, the design of the number and location of measurement points will be demonstrated for a steady-state of the system and a given output response $Y(x,\infty)$ according to Fig. 4.16. The differential properties of the function $Y(x,\infty)$ are evaluated by means of continuity modulus ν or by the norms of higher order derivatives in chosen function spaces, (Hagen (1992), Kornejčuk (1976,1987), Singer (1974), Zavjalov (1980)).

$$\nu\left[Y(x,\infty),h\right]=\max_{\substack{x',x''\in\langle 0,L\rangle\\ |x''-x'|\le h}}\left|Y(x'',\infty)-Y\left(x',\infty\right)\right| \tag{4.139}$$

for $h \le L$

$$\nu\left[Y'(x,\infty),h\right],\quad \nu[Y''(x,\infty),h],... \tag{4.140}$$

$$\left\|Y(x,\infty)\right\|_\infty=\operatorname{ess}\sup_{x\in(0,L\rangle}\left|Y(x,\infty)\right|,\quad \left\|Y'(x,\infty)\right\|_\infty,\quad \left\|Y''(x,\infty)\right\|_\infty,\ldots \text{ etc.} \tag{4.141}$$

Table 4.1 lists the approximation error limits for basic function classes, when interpolated by cubic spline approximation with a uniform spacing of measurement points, i.e. when the difference between the measured distributed quantity $Y(x,\infty)$ and its approximation $Ya(x,\infty)$ is lower than these numbers, (Zavjalov (1980)).

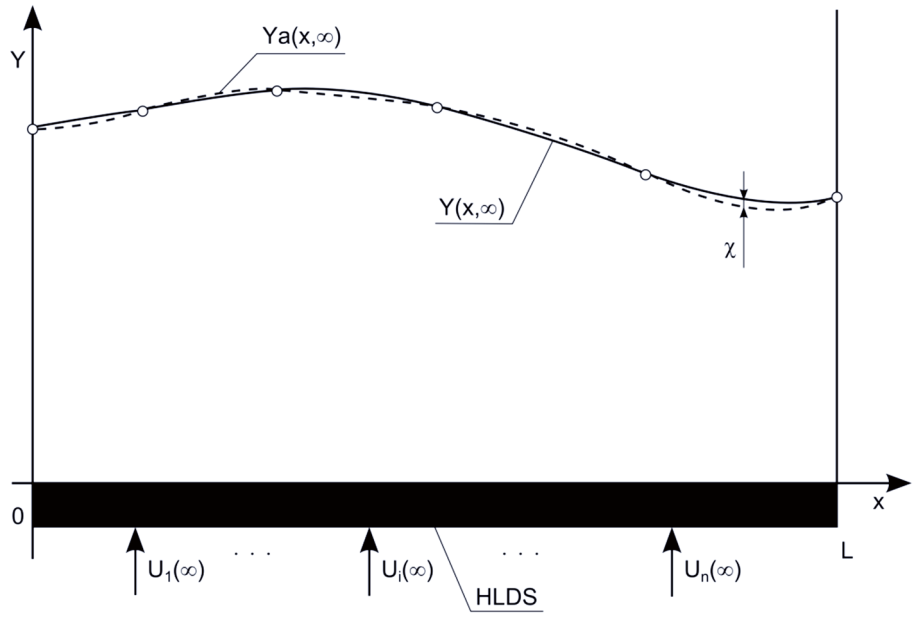


Fig. 4.16 Approximation of the distributed output with prescribed precision.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$U_i(\infty)$ - *i*-th lumped input in steady-state.

$Y(x,\infty)$ - overall distributed output in steady-state.

$Ya(x,\infty)$ - approximation of overall distributed output in steady-state.

χ - accuracy of approximation.

Approximated function class $Y(x,\infty)$	**Error limits difference between** $Y(x,\infty)$ **and** $Ya(x,\infty)$
$Y(x,\infty) \in C^1[0,L]$	$\frac{9}{8} h w[Y'(x,\infty), h]$
$Y(x,\infty) \in W_\infty^2[0,L]$	$\frac{13}{48} h^2 \|Y''(x,\infty)\|_\infty$
$Y(x,\infty) \in C^2[0,L]$	$\frac{19}{96} h^2 w[Y''(x,\infty), h]$
$Y(x,\infty) \in W_\infty^3[0,L]$	$\frac{41}{864} h^3 \|Y'''(x,\infty)\|_\infty$

$Y(x,\infty) \in C^2 C_\Delta^3 [0,L]$	$\frac{41}{1728} h^3 \nu [Y'''(x,\infty), h]$
$Y(x,\infty) \in C^2 W_{\Delta,\infty}^4 [0,L]$	$\frac{5}{384} h^4 \left\| Y^{(4)}(x,\infty) \right\|_\infty$

Table 4.1 Error limits for $Y(x,\infty)$ *and* $Ya(x,\infty)$.

Δ - approximation mesh nodes on the interval $[0,L]$ $\Delta: 0 = x_1 < ... < x_i < x_{i+1} < ... < x_m = L$, with the interval length $[x_i, x_{i+1}]: x_{i+1} - x_i = h$.

ν - modulus of continuity.

$C^k [0,L]$ - class of functions with continuous derivatives up to order k.

$W_p^n [0,L]$ - class of functions $f(x)$ on $[0,L]$, having continuous derivatives up to order *n*-1 and derivatives of order *n* are from the space $L_P [0,L]$, $1 \le p \le \infty$

$C^k C_\Delta^n [0,L]$ - where $n > k$ is the class of functions $f(x)$ such that $f(x) \in C^k [0,L]$ and $f(x) \in C^n [x_i, x_{i+1}]$, $i = 1,2,...,m-1$.

$C^k W_{\Delta,\mathrm{p}}^n [0,L]$ - where $n > k$, $1 \le p \le \infty$ is a class of functions $f(x)$ such that $f(x) \in C^k [0,L]$ and $f(x) \in W_p^n [x_i, x_{i+1}]$, $i = 1,2,...,m-1$.

On the other hand, given a required precision χ, we are also able to determine the limits on distances between uniformly distributed measurement points on the interval $[0, L]$ for measurement of $\{Y(x_j,\infty)\}_j$ based on Table 4.1. For example, from the second row of Table 4.2 at a given approximation precision χ we have a relation

$$h \le \sqrt{\frac{48\chi}{13\left\| Y''(x,\infty) \right\|_\infty}} \tag{4.142}$$

while the maximum value of *h* satisfying this relation is considered, of which the closest integer fraction is *L*. By this procedure, we may move from Table 4.1 to Table 4.2.

If the distribution of measurement points satisfies Eq. (4.142) and measured values of the steady-state distributed quantity are $\{Ya(x_j,\infty)\}_j$, then the cubic spline

approximation $Ya(x,\infty)$ approximates the distributed output $Y(x,\infty)$ in the function space $W_\infty^2[0,L]$ with the precision χ.

Approximated function class $Y(x,\infty)$	**Limit on** h
$Y(x,\infty)\in C^1[0,L]$	$h\le\dfrac{8\chi}{9\nu[Y'(x,\infty),h]}$
$Y(x,\infty)\in W_\infty^2[0,L]$	$h\le\sqrt{\dfrac{48\chi}{13\left\|Y''(x,\infty)\right\|_\infty}}$
$Y(x,\infty)\in C^2[0,L]$	$h\le\sqrt{\dfrac{96\chi}{19\nu[Y''(x,\infty),h]}}$
$Y(x,\infty)\in W_\infty^3[0,L]$	$h\le\sqrt[3]{\dfrac{864\chi}{41\left\|Y'''(x,\infty)\right\|_\infty}}$
$Y(x,\infty)\in C^2C_\Delta^3[0,L]$	$h\le\sqrt[3]{\dfrac{1728\chi}{41\nu[Y'''(x,\infty),h]}}$
$Y(x,\infty)\in C^2W_{\Delta,\infty}^4[0,L]$	$h\le\sqrt{\dfrac{384\chi}{5\left\|Y^{(4)}(x,\infty)\right\|_\infty}}$

Table 4.2 Limits on step size for given χ.

It is well known that in the case of cubic spline approximation at a given precision, significant reduction can be achieved in the number of measurement points by non-uniform distribution of $\{x_j\}_j$. For a given profile $Y(x,\infty)$ and precision χ, we can get an approximation with minimum number of measurements $\left\{Y(x_j,\infty)\right\}_j$, by minimizing the norm $\|Y-Ya\|$ with non-uniform spacing of nodes. This gives a model with minimal complexity, (Zavjalov (1980)).

5. CONTROL SYNTHESIS

This chapter outlines the basic distributed parameter control problems of technological and production processes and offers a solution for the linearized neighborhood of the steady-state operating modes that are represented by linear discrete lumped input and distributed parameter output systems. The aim will be to track the distributed references, while rejecting disturbances around steady-state. Control problems and uncertainties of technological and production processes, along with nonlinear transitions between the steady-state operating modes are also outlined in the Chapter *9. Robust control of the casting mold temperature field* and in the Chapter *12. Modelling and control of the continuous casting of steel.*

The analysis of system dynamics leads to the time and space domain components of the controlled HLDS dynamics. Using a very similar technique, the control synthesis of HLDS is also decomposed into time domain control synthesis and space domain control synthesis. Time domain synthesis utilizes the well-known SISO control methods, while space domain synthesis leads to an approximation problem. Distributed outputs, their disturbances, distributed system characteristics, etc. at time steps "*k",* respectively in the steady-state*,* will form a strictly convex normed function space ***X***. Section *5.1 Open loop control* explains the basic concepts of control synthesis and its decomposition as outlined previously in Hulkó (1998, 2009, 2011). These concepts are then utilized throughout the rest of this book.

5.1 Open Loop Control

Let us consider an open distributed parameter control loop in a steady-state operating mode of the controlled technological and production process. The control process takes place in the linearized neighborhood of the steady-state, where the activity of the control loop compensates for the desired changes with respect to the steady-state. In order to simplify our discussion, steady-state will be considered to be zero and the control process will be discussed as a control problem of a linear HLDS, (Fig. 5.1).

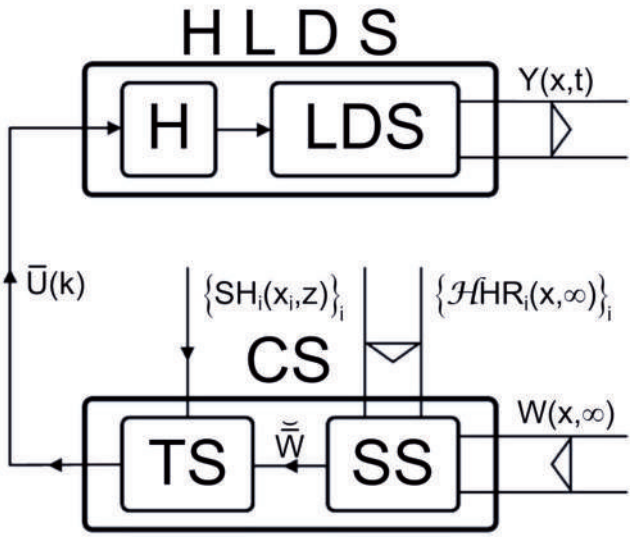


Fig. 5.1 Open DPS control loop.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

H - zero order hold units.

LDS - lumped input and distributed parameter output system.

CS - control system.

TS - time part of control synthesis.

SS - space part of control synthesis.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - space components of the dynamics of the controlled system.

$\{SH_i(x_i,z)\}_i$ - time components of controlled system dynamics.

$\bar{U}(k)$ - discrete lumped input vector.

$\bar{\bar{W}}$ - discrete lumped reference vector.

$W(x,\infty)$ - overall reference in steady-state.

$Y(x,t)$ - overall continuous manipulated variable.

Let subject this loop to a step change of the distributed reference $W(x,k)=W(x,\infty)$, according to the discussion presented above (Fig. 5.2).

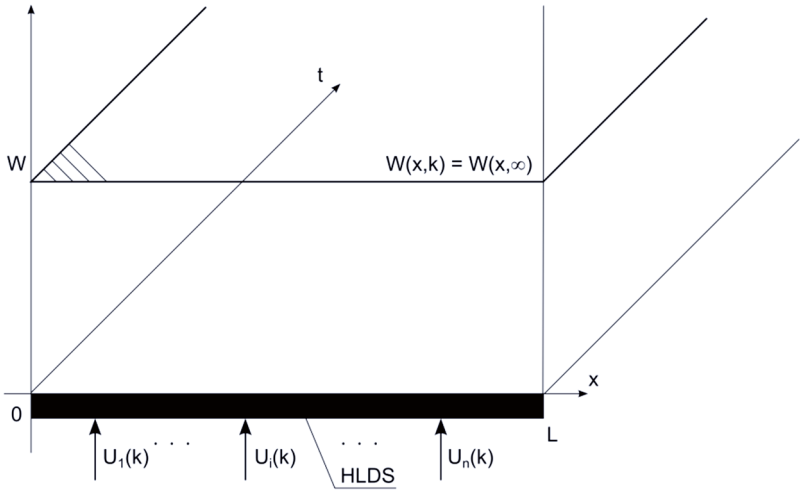


Fig. 5.2 Step change of the distributed reference.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$\{U_i(k)\}_i$ - components of discrete lumped input vector.

$W(x,k)$ - overall discrete reference.

Let the distributed parameter control synthesis problem be defined as finding a sequence of inputs $\bar{U}(k) = \{U_i(k)\}_i$ such that, the control error $E(x,k)$ and its quadratic norm reaches minimum $\|\breve{E}(x,\infty)\|$ in steady-state for $k \to \infty$,

$$\min\|E(x,\infty)\| = \min\|W(x,\infty) - Y(x,\infty)\| = \|\breve{E}(x,\infty)\| \tag{5.1}$$

First, we solve the approximation problem

$$\min\left\|W(x,\infty) - \sum_{i=1}^{n} W_i(x_i,\infty)\,\mathcal{H}HR_i(x,\infty)\right\| \tag{5.2}$$

in the space synthesis block (SS), where $\{\mathcal{H}HR_i(x,\infty)\}_i$ are reduced steady-state profiles of partial distributed step responses of the controlled HLDS and $\{W_i(x_i,\infty)\}_i$ are approximation coefficients. Meanwhile note, that the finite set of approximation functions $\{\mathcal{H}HR_i(x,\infty)\}_i$ is the basis of the finite-dimensional linear subspace of steady-state responses of distributed outputs *Fn*, on the linear and strictly convex

normed space of functions $\boldsymbol{X}$ — that is, in this case on the interval [0, L], where the approximation problem is solved (Fig. 5.3).

It is known from approximation theory that, once given a strictly convex normed linear function space $\boldsymbol{X}$ (e.g. a Hilbert space or Lebesque space $L_p[0, L]$ with $p>1$) and its finite dimensional linear subspace Fn with elements of type $\sum_{i=1}^{n} W_i(x_i,\infty)\mathcal{H}HR_i(x,\infty)$, for each reference $W(x,\infty)\in \boldsymbol{X}$ there exists a unique best approximation $\breve{W}O(x,\infty)=\sum_{i=1}^{n}\breve{W}_i(x_i,\infty)\mathcal{H}HR_i(x,\infty)$ which minimizes the norm (5.2) with the vector of optimal approximation coefficients $\breve{\breve{W}}=\{\breve{W}_i(x_i,\infty)\}_i$, Shadrin (2005). Thus

$$\begin{aligned}&\min\left\|W(x,\infty)-\sum_{i=1}^{n}W_i(x_i,\infty)\,\mathcal{H}HR_i(x,\infty)\right\|=\\&=\left\|W(x,\infty)-\sum_{i=1}^{n}\breve{W}_i(x_i,\infty)\,\mathcal{H}HR_i(x,\infty)\right\|=\left\|W(x,\infty)-\breve{W}O(x,\infty)\right\|\end{aligned}\tag{5.3}$$

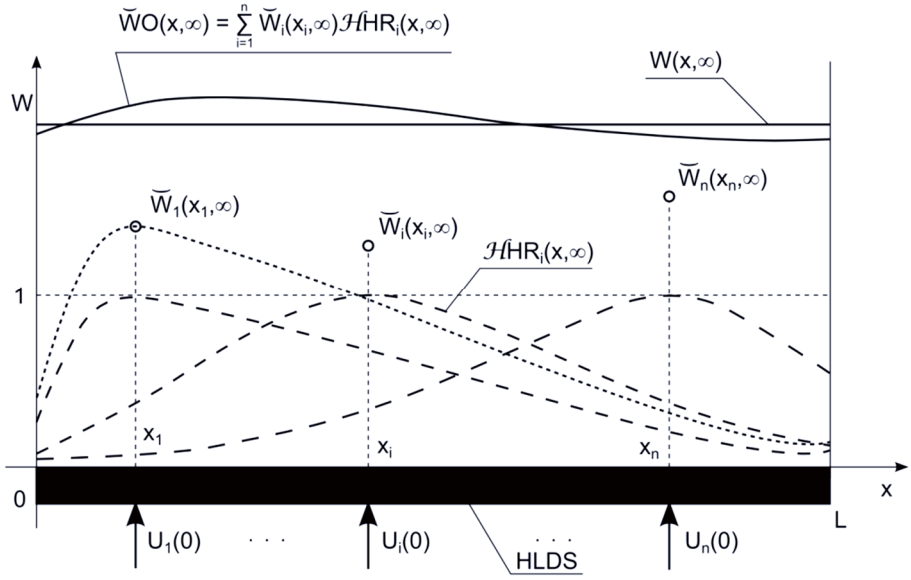


Fig. 5.3 Approximation problem quantities in the block of space synthesis – SS.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - reduced discrete partial step responses in steady-state.

$\{U_i(0)\}_i$ - component of discrete lumped inputs in the initial steady-state.

$W(x,\infty)$ - overall distributed reference in steady-state.

$\{\breve{W}_i(x_i,\infty)\}_i$ - components of discrete lumped references in steady-state.

$\breve{W}O(x,\infty)$ - best approximation of overall discrete set-point in steady-state.

Parameters $\breve{\bar{W}} = \{\breve{W}_i(x_i,\infty)\}_i$ now enter the time synthesis block TS. We have n SISO control loops with plant and controller pairs $\{SH_i(x_i,z), R_i(z)\}_i$ inside this block (see Fig. 5.4). Controlled systems in these are respective HLDS time domain components of dynamics: $\{SH_i(x_i,z)\}_i$.

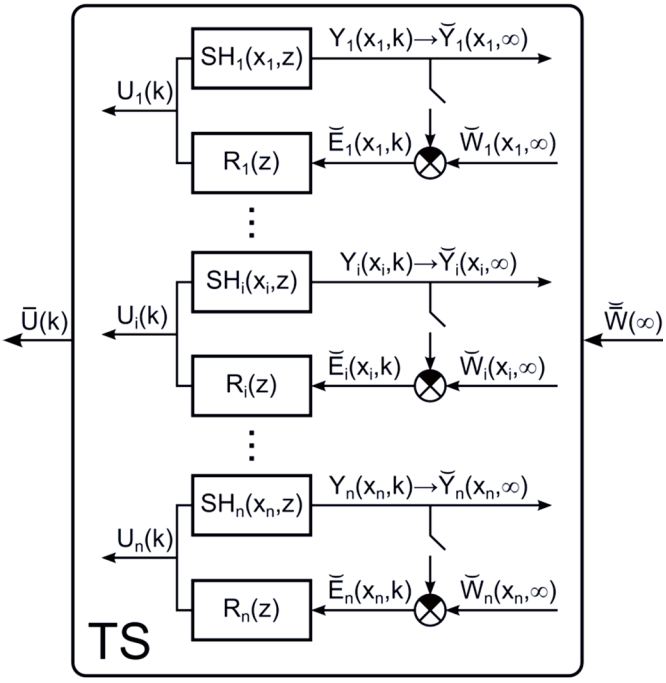


Fig. 5.4 SISO control loops in the block of time synthesis –TS.

TS - time component of control synthesis.

$\{\breve{E}_i(x_i,k)\}_i$ - components of partial control error vectors.

$\{R_i(z)\}_i$ - Z-transfer function of regulators.

$\{SH_i(x_i,z)\}_i$ - time components of the dynamics of the controlled system.

$\bar{U}(k)$	- vector of discrete lumped inputs.
$\{U_i(k)\}_i$	- components of discrete lumped input vector.
$\bar{\bar{W}}(\infty)$	- vector of discrete lumped reference.
$\{\breve{W}_i(x_i,\infty)\}_i$	- components of discrete lumped reference vector in steady-state.
$\{Y_i(x_i,k)\}_i$	- components of partial distributed manipulated variable vector in locations $\{x_i\}_I$.
$\{\breve{Y}_i(x_i,\infty)\}_i$	- components of partial distributed manipulated variable vector in locations $\{x_i\}_I$ in steady-state.

The controllers $\{R_i(z)\}_i$ are designed, by given requirements on character of controlled variables, in order to reach zero steady-state error in each SISO control loop

$$\left\{\lim_{k\to\infty}\breve{E}_i(x_i,k)=\lim_{k\to\infty}\left[\breve{W}_i(x_i,\infty)-Y_i(x_i,k)\right]=0\right\}_{i,k} \tag{5.4}$$

When individual components of the vector $\bar{\bar{W}}=\{\breve{W}_i(x_i,\infty)\}_i$ act on respective SISO control loops $\{SH_i(x_i,z),\ R_i(z)\}_i$, individual control processes take place. From this, we obtain sequences of inputs $\{U_i(k)\}_{i,k}$ and at the respective outputs of each loop responses of lumped manipulated variables will converge for $k\to\infty$ to the respective lumped reference quantities

$$\left\{Y_i(x_i,k)\to Y_i(x_i,\infty)=\breve{W}_i(x_i,\infty)\right\}_{i,k} \tag{5.5}$$

Let us denote these steady-state values of lumped manipulated variables as

$$\left\{\breve{Y}_i(x_i,\infty)=\breve{W}_i(x_i,\infty)\right\}_i \tag{5.6}$$

(see Fig. 5.4 and Fig. 5.5).

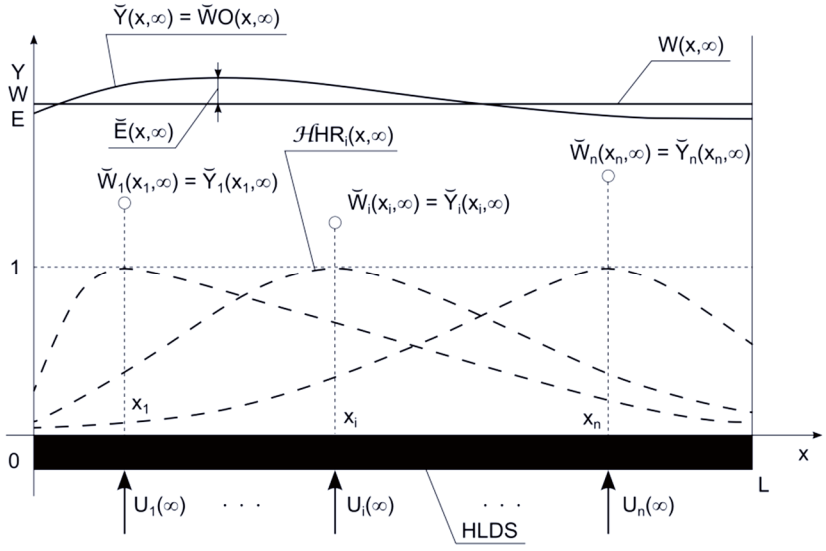


Fig. 5.5 Quantities of control loop in the new steady-state.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

$\breve{E}(x,\infty)$ - overall distributed control error in steady-state with minimal norm.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - reduced discrete partial step responses in steady-state.

$\{U_i(\infty)\}_i$ - vector of discrete lumped inputs in steady-state.

$W(x,\infty)$ - overall distributed reference in steady-state.

$\{\breve{W}_i(x_i,\infty)\}_i$ - components of discrete lumped reference vector in steady-state.

$\breve{W}O(x,\infty)$ - best approximation of overall distributed reference in steady-state.

$\breve{Y}(x,\infty)$ - overall distributed manipulated variable in steady-state.

$\{\breve{Y}_i(x_i,\infty)\}_i$ - components of partial distributed manipulated variable vector in locations $\{x_i\}_i$ in steady-state.

Then, in the steady-state we will have

$\{\breve{Y}_i(x_i,\infty)\mathcal{H}HR_i(x,\infty) = \breve{W}_i(x_i,\infty)\mathcal{H}HR_i(x,\infty)\}_i$ which means that at time $k \to \infty$ the overall distributed output denoted as $\breve{Y}(x,\infty)$ is the best approximation of the distributed reference $W(x,\infty)$, or

$$\breve{Y}(x,\infty)=\sum_{i=1}^{n}\breve{Y}_i(x_i,\infty)\breve{\mathcal{H}}HR_i(x,\infty)=\sum_{i=1}^{n}\breve{W}_i(x_i,\infty)\breve{\mathcal{H}}HR_i(x,\infty)=\breve{W}O(x,\infty) \quad (5.7)$$

Following from the discussion above, the minimum-norm control error is also uniquely given by

$$\begin{aligned}&\left\|\breve{E}(x,\infty)\right\|=\left\|W(x,\infty)-\breve{Y}(x,\infty)\right\|=\left\|W(x,\infty)-\sum_{i=1}^{n}\breve{Y}_i(x_i,\infty)\breve{\mathcal{H}}HR_i(x,\infty)\right\|=\\&\left\|W(x,\infty)-\sum_{i=1}^{n}\breve{W}_i(x_i,\infty)\breve{\mathcal{H}}HR_i(x,\infty)\right\|=\left\|W(x,\infty)-\breve{W}O(x,\infty)\right\|\end{aligned} \quad (5.8)$$

The latter solves the control Problem (5.1) by eliminating the SISO control errors $\left\{\breve{E}_i(x_i,\infty)=0\right\}_i$, (Fig. 5.4).

In essence, we may state that the distributed parameter control synthesis decomposes into:

Time domain control synthesis – by SISO control loops

Space domain control synthesis – by solving an approximation problem

In reality, control synthesis is subject to imprecisions mainly due to modelling errors and disturbances. Denoting these errors in the steady-state as $EY(x,\infty)$, we have

$$\begin{aligned}&\left\|W(x,\infty)-\left[\breve{Y}(x,\infty)\pm EY(x,\infty)\right]\right\|\le\\&\le\left\|W(x,\infty)-\breve{Y}(x,\infty)\right\|+\left\|EY(x,\infty)\right\|=\\&=\left\|W(x,\infty)-\breve{W}O(x,\infty)\right\|+\left\|EY(x,\infty)\right\|=\\&=\left\|\breve{E}(x,\infty)\right\|+\left\|EY(x,\infty)\right\|\end{aligned} \quad (5.9)$$

which implies that for a given desired control quality in the spatial direction δ, one needs to optimize the number and distribution of actuators and measurement points in order to fulfil the condition

$$\left\|\breve{E}(x,\infty)\right\|+\left\|EY(x,\infty)\right\|\le\delta \quad (5.10)$$

5.2 Closed Loop Control with the RHLDS Block

Let us now consider the closed distributed parameter control loop as shown in Fig. 5.6. This loop is subject to the same conditions as outlined in the previous section. The block RHLDS generates the reduced responses of partial distributed outputs $\{YR_i(x,k)\}_{i,k}$, while blocks SS1 and SS2 solves approximation tasks. The TS block (Fig. 5.7) is in fact a set of SISO controllers $\{R_i(z)\}_i$, each tuned to compensate for the respective time domain dynamics of HLDS $\{SH_i(x_i,z)\}_i$, in order to satisfy Eq. (5.4).

The reduced responses of partial distributed outputs $\{YR_i(x,k)\}_{i,k}$ that are generated in the block RHLDS will be now considered in step *k* as the basis functions for the finite-dimensional linear subspace of the functions *Yn* on ***X***. According to Eq. (3.22) the relation $Y(x,k) \in Yn$ will hold after the effect of the identical sequence of inputs $\bar{U}(k)$ on the block RHLDS (and also HLDS). Then, by solving the approximation problem in the block SS2 we will obtain

$$\min \left\| Y(x,k) - \sum_{i=1}^{n} Y_i(x_i,k) YR_i(x,k) \right\| = \left\| Y(x,k) - \sum_{i=1}^{n} \breve{Y}_i(x_i,k) YR_i(x,k) \right\| = 0 \quad (5.11)$$

in the quadratic norm, working out a vector of time domain components of the individual distributed outputs $\breve{\bar{Y}}_i(x_i,k) = \{\breve{Y}_i(x_i,k)\}_i$ in locations $\{x_i\}_i$, (Components (3.20-26)). Meanwhile, the SS1 block calculates lumped references $\{\breve{W}_i(x_i,\infty)\}_i$, (Problem 5.2). Then, the output of the algebraic block will yield $\breve{\bar{E}}_i(x_i,k) = \breve{\bar{W}}_i(x_i,\infty) - \breve{\bar{Y}}_i(x_i,k)$.

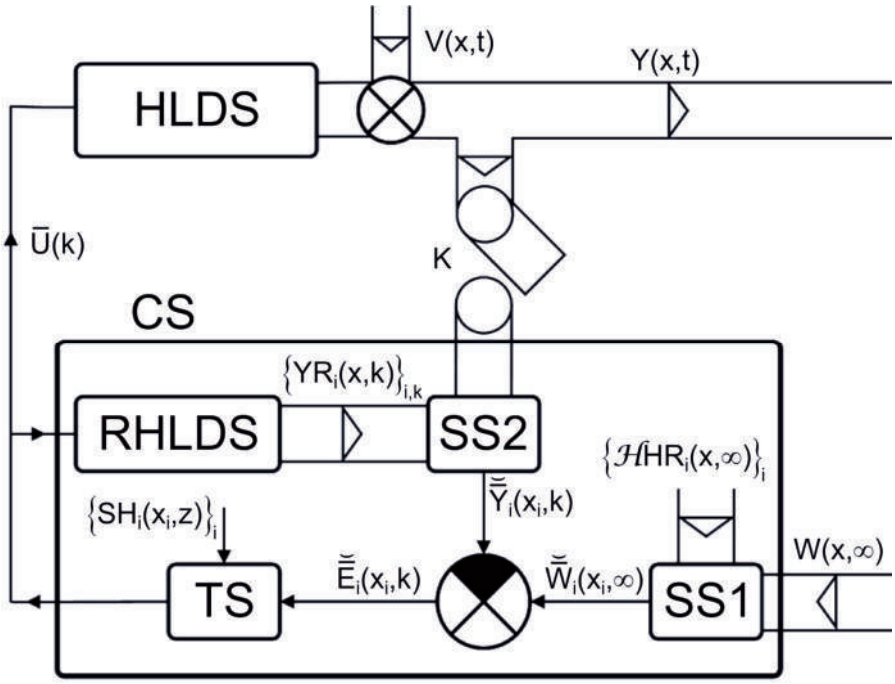


Fig. 5.6 Closed DPS control loop with RHLDS block.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

RHLDS - block for generating reduced characteristics.

CS - control system.

TS - time part of control synthesis.

SS1, SS2 - space parts of control synthesis.

K - sampling.

$\bar{\bar{E}}_i(x_i,k)$ - vector of partial control errors in locations $\{x_i\}_i$.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - space components of controlled system dynamics.

$\{SH_i(x_i,z)\}_i$ - time components of controlled systems dynamics.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$V(x,t)$ - continuous distributed disturbance.

$W(x,\infty)$ - overall distributed reference in steady-state.

$\bar{\bar{W}}_i(x_i,\infty)$ - vector of discrete lumped reference in steady-state.

$Y(x,t)$ - overall continuous distributed manipulated variable.

$\breve{\bar{Y}}_i(x_i,k)$ - vector of partial manipulated variables in locations $\{x_i\}_i$.

$\{YR_i(x,k)\}_{i,k}$ - reduced discrete partial distributed manipulated variables.

The control error $\breve{\bar{E}}_i(x_i,k)$ enters the TS block and its components the respective SISO controllers $\{R_i(z)\}_i$, (Fig. 5.7). Inputs $\bar{U}(k)=\{U_i(k)\}_i$ then act on the HLDS block. This way the SISO control loops $\{SH_i(x_i,z),R_i(z)\}_i$ are closed via the TS and HLDS blocks as Fig. 5.4 indicates.

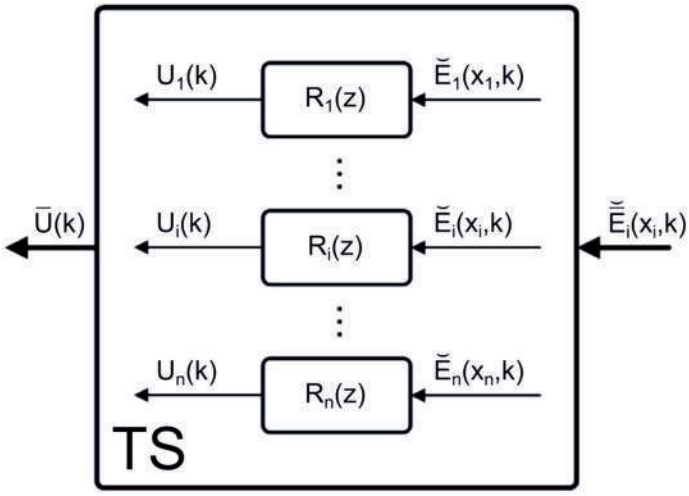


Fig. 5.7 Time synthesis block.

TS - time component of control synthesis.

$\breve{\bar{E}}_i(x_i,k)$ - vector of partial control error in locations $\{x_i\}_i$.

$\{\breve{E}_i(x_i,k)\}_i$ - components of the vector of partial control errors in locations $\{x_i\}_i$.

$\{R_i(z)\}_i$ - Z-transfer functions of regulators.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\{U_i(k)\}_i$ - components of vector of discrete lumped inputs.

For $k\to\infty$ we have $\{YR_i(x,k)\to\mathcal{H}HR_i(x,\infty)\}_{i,k}$, $\breve{\bar{Y}}_i(x_i,k)\to\breve{\bar{Y}}_i(x_i,\infty)=\breve{\bar{W}}_i(x_i,\infty)$ and $\{\breve{E}_i(x_i,k)\to\breve{E}_i(x_i,\infty)=0\}_{i,k}$, which solves the control Problem (5.1) according to

Eq. (5.8). Similar relations hold as in the previous Conditions (5.9), (5.10) in case the control synthesis is subject to errors quantified by $EY(x,\infty)$.

Let us now consider the approximation of $W(x,k)$ in block SS1 by the set $\{YR_i(x,k)\}_i$, Fig. 5.6. The control process then yields a sequence of references $\{\breve{W}_i(x_i,k)\}_i$ entering SISO control loops $\{SH_i(x_i,z), R_i(z)\}_i$, which are closed via the TS and HLDS blocks, Fig. 5.8.

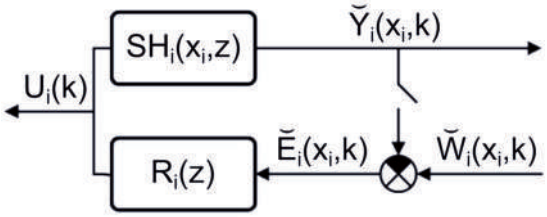


Fig. 5.8 i-th SISO control loop.

$\breve{E}_i(x_i,k)$	- *i*-th component of partial control error vector in location x_i.
$R_i(z)$	- *i*-th Z-transfer function of regulator.
$SH_i(x_i,z)$	- *i*-th time component of controlled system dynamics.
$U_i(k)$	- *i*-th discrete lumped input.
$\breve{W}_i(x_i,k)$	- *i*-th component of discrete lumped reference vector.
$\breve{Y}_i(x_i,k)$	- *i*-th partial distributed manipulated variable in location x_i, variable *k*.

Let the distributed reference converge to $W(x,\infty)$ for $k\to\infty$. Then $\{YR_i(x,k)\to\mathcal{H}HR_i(x,\infty)\}_{i,k}$, $\{Y_i(x_i,k)=\breve{Y}_i(x_i,k)\}_{i,k}\to\{\breve{Y}_i(x_i,\infty)=\breve{W}_i(x_i,\infty)\}_i$ together with $\{\breve{E}_i(x_i,k)\to\breve{E}_i(x_i,\infty)=0\}_{i,k}$, which solves the control Problem (5.1) according to Eq. (5.8).

Similarly to the previous conclusion, we can state that control process in the control loop with RLHDS block decomposes to:

Time domain control synthesis – by SISO control loops,

Space domain control synthesis – by solving approximation problems.

As a solution of the approximation problem in block SS1 on the set $\{YR_i(x,k)\}_i$, i.e.

$$\min\left\|W(x,\infty)-\sum_{i=1}^{n}W_i(x_i,k)YR_i(x,k)\right\|=\left\|W(x,\infty)-\sum_{i=1}^{n}\breve{W}_i(x_i,k)YR_i(x,k)\right\| \quad (5.12)$$

in time step *k* we get

$$W(x,\infty)=\sum_{i=1}^{n}\breve{W}_i(x_i,k)YR_i(x,k)+\breve{E}(x,k) \quad (5.13)$$

where $\breve{E}(x,k)$ is the approximation error, uniquely given on the set of approximation functions $\{YR_i(x,k)\}_i$. We get from the approximation in the SS2 block

$$Y(x,k)=\sum_{i=1}^{n}\breve{Y}_i(x_i,k)YR_i(x,k) \quad (5.14)$$

in the same manner. Let us now demonstrate the principle of disturbance rejection in the system. The disturbance $V(x,t)=V(x,\infty)$ acts in time t on the control loop, subject to the same conditions as above, where for the sake of simplicity, the set-point is in the form $W(x,\infty)=\sum_{1}^{n}\breve{W}_i(x_i,\infty)\mathcal{H}HR_i(x,\infty)$. In block SS2 for $k\geq t$ the approximation tasks are solved sequentially as

$$\begin{aligned}&\min\left\|YV(x,k)-\sum_{i=1}^{n}YV_i(x_i,k)YR_i(x,k)\right\|=\\&=\left\|YV(x,k)-\sum_{i=1}^{n}\breve{Y}V_i(x_i,k)YR_i(x,k)\right\|=\left\|\breve{E}V(x,k)\right\|\end{aligned} \quad (5.15)$$

where $YV(x,k)=Y(x,k)+V(x,k)$. In the control process for $k\to\infty$ $\{YR_i(x,k)\to\mathcal{H}HR_i(x,\infty)\}_{i,k}$ and $\{\breve{Y}V_i(x_i,\infty)=\breve{W}_i(x_i,\infty)\}_i$, meaning that $\breve{E}V(x,\infty)$ is the deviation from the set-point given the disturbance $V(x,t)_{t\to\infty}=V(x,\infty)$ with a minimal quadratic norm in the steady-state.

In order to conclude this section, let us note that instead of locations $\{x_i\}_{i=1,n}$, it is also possible to consider a suitable configuration of locations $\{x_i'\}_{i=1,n}$ when designing

a control system. Then $\left\{\mathcal{HH}R_i'(x,\infty)=\mathcal{HH}_i(x,\infty)/\mathcal{HH}_i(x_i',\infty)\right\}_i$ for $\left\{\mathcal{HH}_i(x_i',\infty)\neq 0\right\}_i$ and $\left\{YR_i'(x,k)\right\}_i=\left\{Y_i(x,k)/Y_i(x_i',k)\right\}_{i,k}$ for $\left\{Y_i(x_i',k)\neq 0\right\}_{i,k}$, meanwhile the transfer functions will be $\left\{SH_i(x_i',z)\right\}_i$. Finally, $\breve{\breve{E}}_i'(x_i,k)=\breve{\breve{W}}_i'(x_i,\infty)-\breve{\breve{Y}}_i'(x_i,k)$ will enter the block of time synthesis with controllers $\left\{R_i'(z)\right\}_i$, which are tuned in the feedback loops $\left\{SH_i(x_i',z),R_i'(z)\right\}_i$.

5.3 Closed Loop Control

Let us now consider the feedback control loop as in Fig. 5.9, under the same conditions as in the latter case. Now we solve the approximation problem

$$\begin{aligned}&\min\left\|E(x,k)-\sum_{i=1}^{n}E_i(x_i,k)YR_i(x,k)\right\|=\\&=\left\|E(x,k)-\sum_{i=1}^{n}\breve{E}_i(x_i,k)YR_i(x,k)\right\|\end{aligned}\tag{5.16}$$

The result of this is, that in step k we will obtain vector $\breve{\breve{E}}_i(x_i,k)$. Using Eq. (5.14) it is possible to evaluate

$$\begin{aligned}&\left\|E(x,k)-\sum_{i=1}^{n}\breve{E}_i(x_i,k)YR_i(x,k)\right\|=\\&=\left\|W(x,\infty)-Y(x,k)-\sum_{i=1}^{n}\breve{E}_i(x_i,k)YR_i(x,k)\right\|=\\&=\left\|W(x,\infty)-\sum_{i=1}^{n}\left[\breve{Y}_i(x_i,k)+\breve{E}_i(x_i,k)\right]YR_i(x,k)\right\|\end{aligned}\tag{5.17}$$

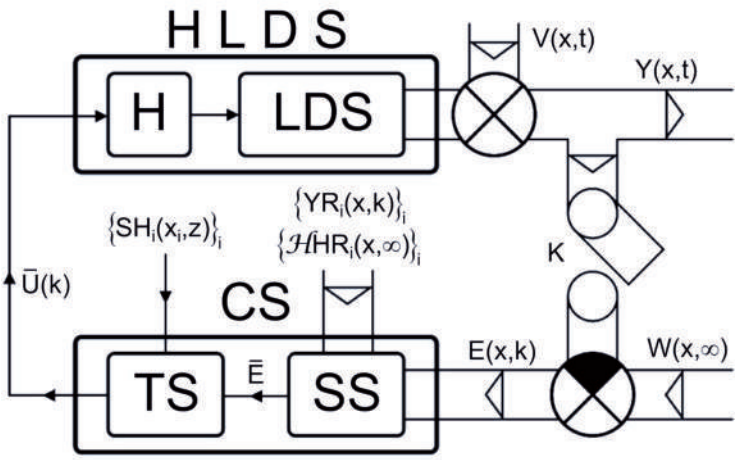


Fig. 5.9 Closed distributed parameter control loop.

HLDS - discrete lumped input and distributed parameter output system with zero order hold units.

CS - control system.

TS - time part of control synthesis.

SS - space part of control synthesis.

K - sampling.

$\bar{\bar{E}}(k)$ - control error vector.

$E(x,k)$ - overall distributed control error.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - space components of controlled system dynamics.

$\{SH_i(x_i,z)\}_i$ - time components of controlled system dynamics.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$V(x,t)$ - continuous distributed disturbance.

$W(x,\infty)$ - overall distributed reference in steady-state.

$Y(x,t)$ - overall continuous distributed manipulated variable.

$\{YR_i(x,k)\}_i$ - reduced discrete partial distributed manipulated variables.

and hence

$$W(x,\infty)=\sum_{i=1}^{n}\left[\breve{Y}_i(x_i,k)+\breve{E}_i(x_i,k)\right]YR_i(x,k)+\breve{E}(x,k) \tag{5.18}$$

Comparing this result to Eq. (5.13) we get

$$\begin{aligned}&\sum_{i=1}^{n}\breve{W}_i\left(x_i,k\right)YR_i\left(x,k\right)+\breve{E}\left(x,k\right)=\\&=\sum_{i=1}^{n}\left[\breve{Y}_i\left(x_i,k\right)+\breve{E}_i\left(x_i,k\right)\right]YR_i\left(x,k\right)+\breve{E}\left(x,k\right)\end{aligned}\tag{5.19}$$

which gives

$$\sum_{i=1}^{n}\breve{W}_i\left(x_i,k\right)YR_i\left(x,k\right)=\sum_{i=1}^{n}\left[\breve{Y}_i\left(x_i,k\right)+\breve{E}_i\left(x_i,k\right)\right]YR_i\left(x,k\right)\tag{5.20}$$

and finally

$$\left\{\breve{W}_i\left(x_i,k\right)=\breve{Y}_i\left(x_i,k\right)+\breve{E}_i\left(x_i,k\right)\right\}_{i,k}\tag{5.21}$$

Taking just vector quantities we have

$$\begin{aligned}&\breve{\breve{W}}\left(k\right)=\breve{\breve{Y}}\left(k\right)+\breve{\breve{E}}\left(k\right)\Rightarrow\\&\quad\Rightarrow\breve{\breve{E}}\left(k\right)=\breve{\breve{W}}\left(k\right)-\breve{\breve{Y}}\left(k\right)=\left\{\breve{E}_i\left(x_i,k\right)=\breve{W}_i\left(x_i,k\right)-\breve{Y}_i\left(x_i,k\right)\right\}_{i,k}\end{aligned}\tag{5.22}$$

The vector $\breve{\breve{E}}_i\left(x_i,k\right)$ enters the TS block for controllers $\left\{R_i\left(z\right)\right\}_i$, and the control system works in line with the idea of the SISO control loops $\left\{SH_i\left(x_i,z\right),\ R_i\left(z\right)\right\}_i$, closed via the HLDS block (Fig. 5.8). In steady-state, $k\to\infty$ and $W\left(x,\infty\right)$ we get $\left\{YR_i\left(x,k\right)\to\mathcal{H}HR_i\left(x,\infty\right)\right\}_i$ and $\breve{\breve{Y}}_i\left(x_i,k\right)\to\breve{\breve{Y}}_i\left(x_i,\infty\right)=\breve{\breve{W}}_i\left(x_i,\infty\right)$, and thus $\left\{\breve{E}_i\left(x_i,k\right)\to 0\right\}_i$.

Assembling these, the following holds for the steady-state

$$\begin{aligned}
&\left\| E(x,\infty) - \sum_{i=1}^{n} \breve{E}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) \right\| = \\
&= \left\| W(x,\infty) - \breve{Y}(x,\infty) - \sum_{i=1}^{n} \breve{E}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) \right\| = \\
&= \left\| \begin{array}{l} \sum_{i=1}^{n} \breve{W}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) + \breve{E}(x,\infty) - \\ -\sum_{i=1}^{n} \breve{Y}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) - \sum_{i=1}^{n} \breve{E}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) \end{array} \right\| = \\
&= \left\| \begin{array}{l} \sum_{i=1}^{n} \left[\breve{W}_i(x_i,\infty) - \breve{Y}_i(x_i,\infty) \right] \mathcal{H}HR_i(x,\infty) + \breve{E}(x,\infty) + \\ -\sum_{i=1}^{n} \breve{E}_i(x_i,\infty)\, \mathcal{H}HR_i(x,\infty) \end{array} \right\| = \left\| \breve{E}(x,\infty) \right\|
\end{aligned} \tag{5.23}$$

With this, the control task outlined by Eq. (5.1) is solved according to Eq. (5.8) with an optimal course of the distributed manipulated variable in the steady-state $\breve{Y}(x,\infty)$.

In case the control synthesis is subject to errors, whose effect is quantified by $EY(x,\infty)$, then similar relations hold as in the previous case, that is Eqs. (5.9), (5.10).

The control synthesis in this case also decomposes to:

Time domain control synthesis – by SISO control loops,

Space domain control synthesis – by solving approximation problems.

5.4 Additional Control Loop Configurations

Sections *5.1* to *5.3* presented control feedback loops with distributed parameters. Based on the results introduced previously, we will suggest other types of control feedback loops that may be useful to solve various benchmark examples of controlling technological and production processes as distributed parameter systems.

5.4.1 Control Loop with Internal Model

Let us consider the control loop with an internal model, MHLDS, as shown in Fig. 5.10. In case the MHLDS ideally models the controlled system dynamics, i.e. $YH(x,k) \equiv YM(x,k)$, then the system operates as in the open distributed parameter control loop case from Section *5.1*.

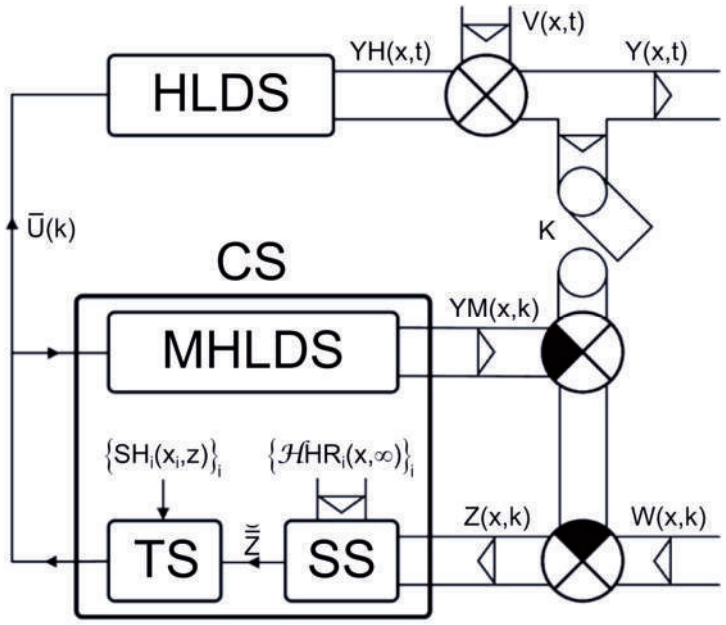


Fig. 5.10 Distributed parameter control loop with internal model.

HLDS - discrete lumped input and distributed parameter output system with zero order hold.

MHLDS - model of the controlled system.

CS - control system.

TS - time part of control synthesis.

SS - space part of control synthesis.

K - sampling.

$\{\mathcal{H}HR_i(x,\infty)\}_i$ - space components of controlled system dynamics.

$\{SH_i(x_i,z)\}_i$ - time components of controlled systems dynamics.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$V(x,t)$ - continuous distributed disturbance.

$W(x,k)$	- overall distributed reference.
$Y(x,t)$	- overall continuous manipulated variable.
$YH(x,t)$	- overall continuous distributed manipulated variable of block HLDS.
$YM(x,k)$	- overall continuous distributed output of block MHLDS.
$\breve{\breve{Z}}$	- vector of modified lumped reference.
$Z(x,k)$	- modified distributed reference.

5.4.2 Control of Partial Differential Equations

In the following discussion, we will show control loop arrangements for the solution of basic PDE control problems (Hulkó (1998)) by using the HgDDS and H_{Ti}DDS blocks introduced in Section *4.1 Analytical system identification.*

Implementing a control loop with internal model, we get feedback loops for DDS boundary control (see Fig. 5.11), and for DDS control by means of distributed input from a class of functions, (see Fig. 5.12)

$$U(x,t) = \sum_{i=1}^{n} U_i(t) T_i(x) \tag{5.24}$$

specified by the shaping functions $\{T_i(x)\}_i$.

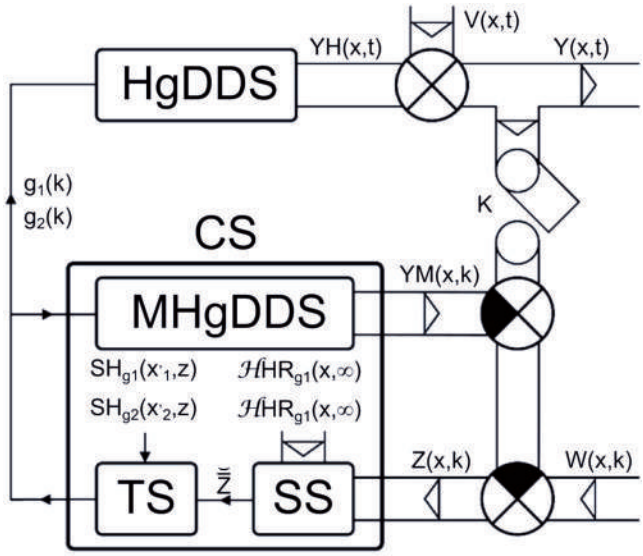


Fig. 5.11 DDS/PDE Boundary control.

HgDDS - discrete controlled system of boundary control.

MHgDDS - model of the discrete controlled system of boundary control.

CS - control system.

TS - time part of control synthesis.

SS - space part of control synthesis.

K - sampling.

$g_1(k), g_2(k)$ - discrete inputs of the block HgDDS.

$\mathcal{H}\hat{H}Rg_1(x,\infty)$
$\mathcal{H}\hat{H}Rg_2(x,\infty)$ - space components of controlled system dynamics.

$SHg_1(x_1',z)$
$SHg_2(x_2',z)$ - time components of controlled systems dynamics.

$V(x,t)$ - continuous distributed disturbance.

$W(x,k)$ - overall distributed reference.

$Y(x,t)$ - overall continuous distributed manipulated variable.

$YH(x,t)$ - overall continuous distributed output of the block HgDDS.

$YM(x,k)$ - overall continuous distributed output of the block MHgDDS.

$Z(x,k)$ - modified distributed reference.

$\breve{\breve{Z}}$ - vector of modified lumped reference.

In accordance with the conclusion of Section *5.1 Open loop control*, the reference of SISO control loops in the TS block

$$\breve{\breve{Z}} = \left[\breve{Z}_1 (x_1', k), \breve{Z}_2 (x_2', k) \right] \tag{5.25}$$

is obtained by solving the approximation problem

$$\begin{aligned} &\min \left\| Z(x,k) - \sum_{i=1}^{2} Z_i (x_i', k) \mathcal{H}HRg_i (x, \infty) \right\| = \\ &= \left\| Z(x,k) - \sum_{i=1}^{2} \breve{Z}_i (x_i', k) \mathcal{H}HRg_i (x, \infty) \right\| \end{aligned} \tag{5.26}$$

where x_1', x_2' are (internal) locations on the interval $[0, L]$, located in the vicinity of boundaries.

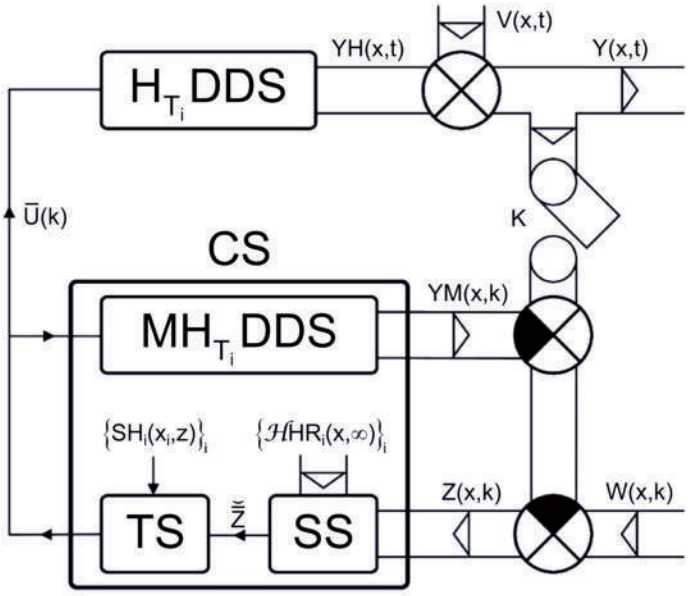


Fig. 5.12 DDS/PDE Control.

$H_{Ti}DDS$ - discrete controlled system with shaping units in the spatial relation.

$MH_{Ti}DDS$ - model of the discrete controlled system with shaping units in the spatial relation.

CS - control system.

TS - time part of control synthesis.

SS - space part of control synthesis.

K - sampling.

$\left\{\mathcal{H}HR_i\left(x,\infty\right)\right\}_i$ - space components of the dynamics.

$\left\{SH_i\left(x_i,z\right)\right\}_i$ - time components of the dynamics.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$V\left(x,t\right)$ - continuous distributed disturbance.

$W\left(x,k\right)$ - overall distributed reference.

$Y\left(x,t\right)$ - overall continuous distributed manipulated variable.

$YH\left(x,t\right)$ - overall continuous distributed output of block $H_{Ti}DDS$.

$YM\left(x,k\right)$ - overall continuous distributed output of block $MH_{Ti}DDS$.

$\bar{\bar{Z}}$ - vector of modified lumped reference.

$Z\left(x,k\right)$ - modified distributed reference.

Control loop arrangements from Fig. 5.11 and Fig. 5.12 use the dynamic characteristics defined in the same way as in previous sections:

$SHg_1\left(x_1',z\right)$, $SHg_2\left(x_2',z\right)$ – lumped transfer functions between boundary conditions and partial distributed outputs, i.e. between lumped quantities in locations x_1' and x_2'. These are points in the vicinity of 0 and *L*, the locations for the effect of the control variables $g_1(k)$, $g_2(k)$ thus the boundary conditions $g_1(t)$, $g_2(t)$.

$\mathcal{H}HRg_1\left(x,\infty\right)$, $\mathcal{H}HRg_2\left(x,\infty\right)$ – reduced steady-state profiles of individual distributed step responses to step changes in boundary conditions.

$\left\{SH_i\left(x_i,z\right)\right\}_i$ – lumped transfer functions between *i*-th input and the controlled PDE in location x_i, belonging to the blocks $H_{Ti}DDS$.

$\left\{\mathcal{H}HR_i\left(x,\infty\right)\right\}_i$ – reduced steady-state profiles of individual distributed step responses to step changes in lumped inputs, belonging to the block $H_{Ti}DDS$.

Then the vector of lumped modified references

$$\breve{\breve{Z}} = \left\{ \breve{Z}_i \left(x_i, k \right) \right\}_i \tag{5.27}$$

is obtained as a result of the approximation problem

$$\begin{aligned} &\min \left\| Z\left(x,k\right) - \sum_{i=1}^{n} Z_i \left(x_i, k \right) \mathcal{H}HR_i \left(x, \infty \right) \right\| = \\ &= \left\| Z\left(x,k\right) - \sum_{i=1}^{n} \breve{Z}_i \left(x_i, k \right) \mathcal{H}HR_i \left(x, \infty \right) \right\| \end{aligned} \tag{5.28}$$

with corresponding $\left\{ \mathcal{H}HR_i \left(x, \infty \right) \right\}_i$ to model $H_{Ti}DDS$.

5.5 Robust and Adaptive Control

The design of robust and adaptive control systems is based on lumped input and distributed parameter output numerical models introduced in second chapter *DPS – DDS – LDS*. After the robustification of these models, the synthesis of control in the spatial relation is solved as an approximation problem. Classical lumped parameter robust control methods can be used in the temporal relation. For the needs of adaptive-predictive control systems, online identification methods are used, which represent the linear approximation of the system dynamics at each sampling interval. After the decomposition of the dynamics of the predictors, control synthesis is solved just as above, that is, in the time and space dependence (Hulkó (1998)).

As it has been previously mentioned, in engineering practice it is often sufficient to solve control problems in the neighborhood steady-state operating modes by linearized models. On the other hand, real-life control processes are subject to various uncertainties; moreover, it is often necessary to consider the nonlinear and time-variant nature of the systems dynamics in transients. Individual cases should be dealt with separately, as their solution is specific and cannot be generalized. Therefore, this book will introduce the distributed aspects of robust and adaptive control in the following chapters as applied to specific case studies. The Chapter *9. Robust control of the casting mold temperature field* introduces the problem of controlling the temperature fields in a casting molds, while the Chapter *12. Modeling and control of the continuous casting of steel* will apply adaptive control methods to regulate the transition of operating modes in continuous casting by influencing the temperature fields in the secondary cooling zone.

6. SOFTWARE SUPPORT

The development of computing technology opens new horizons to establish software support designed to solve the problems of control for technological or production processes as systems with distributed parameters. This chapter deals with some basic case studies for this software support, which may serve as a future guide for software development for specific industry conditions and requirements.

The DPS Blockset for Matlab & Simulink is a software package developed over several years. It has been built within the framework of the Connections program of The MathWorks Company, and it is a partnership product of The MathWorks Inc., offering the package on the website of the company. The main ambitions of this software are didactical and directed at demonstrating the flexible control possibilities of systems with distributed parameters, by using the interface of Matlab & Simulink.

It deals with several issues in the fields of various industrial and technological processes, mechatronics and environmental protection. The PID control, Algebraic Control, State-Space Control and Robust Control Simulink blocks are offered for control synthesis in the time domain. The DPS Blockset described in this chapter is aimed at applications in the field of technological processes, showing a specific example of the usage for modelling the continuous casting process. The procedure is performed by an algorithmic interconnection of ProCAST with Matlab & Simulink, the so-called Block MtoP module. These blocks are designed to integrate the control loops into the information and management system for the continuous casting of steel.

As it has been previously established, technological and production processes are essentially systems with nonlinear dynamics. The numerical models obtained via software environments also describe nonlinear models of these processes. The DPS Blockset for Matlab & Simulink operates with linear models of controlled systems, which are obtained in the linearized region around a steady-state operating mode. The toolbox offers co-simulation systems to investigate control tasks in different steady-state operating modes (respectively between different steady-state operating modes), for example by the segmentation of nonlinear dynamics in transitions. The cooperation of Matlab & Simulink and a compatible software environment for numerical modelling is ensured within the framework of co-simulation contained in the toolbox. The co-simulation between ProCAST and Matlab & Simulink supporting the solution of control problems in the field of continuous casting is described here. These possibilities are detailed in Chapter *12. Modelling and control of the continuous casting of steel.*

The end of this chapter also outlines the fundamental idea of the Interactive Control system, offering the possibility to perform interactive solutions of control tasks for technological and production processes via the Internet.

6.1 DPS Blockset

The Distributed Parameter Systems Blockset for Simulink (DPS Blockset) provides software support to ensure control synthesis of distributed parameter systems in Matlab & Simulink. The core software is written in C++ and mathematical operations use algorithms from the LAPACK libraries. Input-output quantities are considered as discrete values. The discrete quantities of the controlled system in the space domain are considered as values in the nodes of the computational mesh, respectively the measurement locations on the definition domain. These discrete quantities will provide a continuous approximation of the investigated distributed quantities by the chosen methodology of interpolation. The discrete quantities will represent distributed quantities for the chosen methodology of interpolation, therefore these will not be denoted separately in further discussion. Furthermore, it shall be clear from the context that the quantities are temporally and spatially distributed.

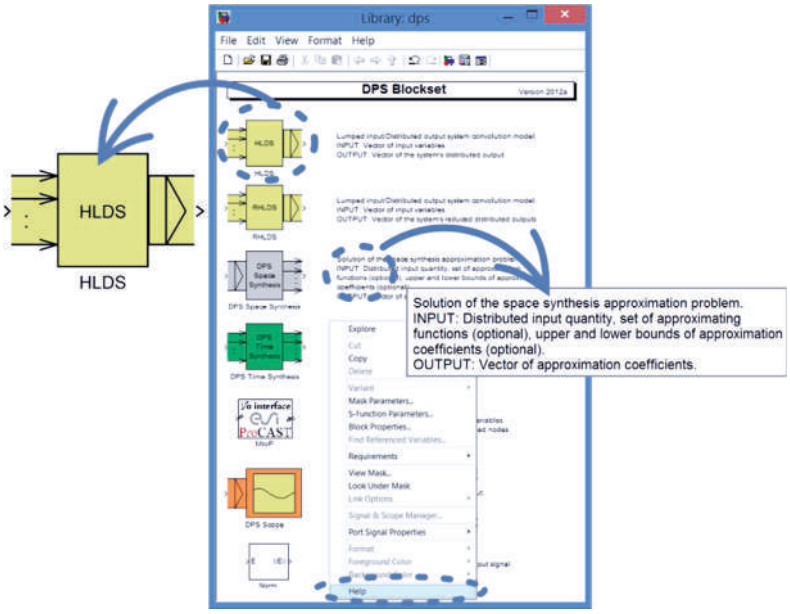


Fig. 6.1 DPS Blockset block library.

The graphic design of blocks demonstrates the input and output structure of the system. At the right hand side of the blockset, a short note describes the function of the

block and the structure of its inputs and outputs. The DPS Blockset library (Fig. 6.1) also includes a block named MtoP, constituting an interface for exchanging data between Matlab & Simulink and ProCAST. This feature allows one to use the full potential of the DPS Blockset and Matlab & Simulink to design control systems for casting and continuous casting, based on numerical models of these processes in the ProCAST software environment.

6.1.1 Block: HLDS

The block HLDS provides the numerical model of the controlled system in the form of a discrete system with lumped inputs and distributed outputs with a zero order hold unit, (Fig. 6.2).

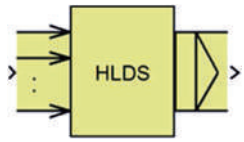


Fig. 6.2 Icon of the HLDS block.

The input is given as a vector of lumped inputs $\{U_i(k)\}_i$, specified in the form of an *i*-dimensional vector at *k* samples. The output of the block is the output $Y(\boldsymbol{x},k)$, which is distributed in time and space. The HLDS block computes the distributed output in response to any lumped input, based on the discrete convolution equation

$$Y(\boldsymbol{x},k)=\sum_{i=1}^{nU}\sum_{q=0}^{k}\mathcal{G}H_i(\boldsymbol{x},k-q)U_i(q) \tag{6.1}$$

where $Y(\boldsymbol{x},k)$ is a block output; $U_i(q)$ is a *i*-th block lumped input; *nU* is the number of lumped inputs and $\mathcal{G}H_i(\boldsymbol{x},k)$ is an *i*-th distributed impulse characteristic, given in the form of a discrete variable.

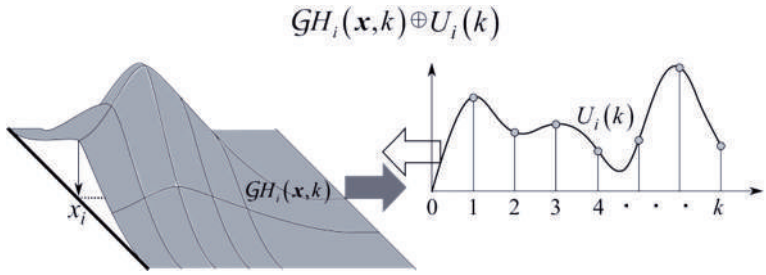


Fig. 6.3 Discrete convolution of distributed impulse characteristics with a lumped input signal.

The result of the convolution in Fig. 6.3 is a partial distributed output $Y_i(\boldsymbol{x},k)$. Summing $Y_i(\boldsymbol{x},k)$ gives the overall distributed output

$$Y(\boldsymbol{x},k)=\sum_{i=1}^{nU}Y_i(\boldsymbol{x},k) \tag{6.2}$$

The distributed impulse response is calculated within the HLDS block according to the equation

$$\mathcal{G}H_i(\boldsymbol{x},k)=\mathcal{H}H_i(\boldsymbol{x},k)-\mathcal{H}H_i(\boldsymbol{x},k-1) \tag{6.3}$$

for $i=1,2\ldots nU$ and $k=0,1,2\ldots N$, where $\mathcal{H}H_i(\boldsymbol{x},k)$ is the *i*-th distributed transient characteristic of the system with lumped input and distributed output.

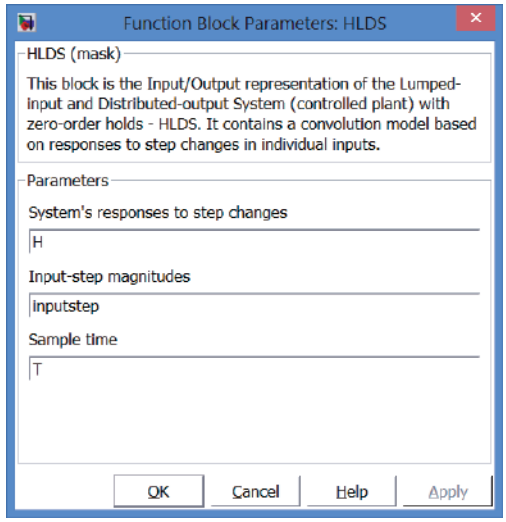


Fig. 6.4 Mask of the HLDS block.

The parameter *„System's responses to step changes“* (Fig. 6.4) are individual distributed step responses by individual lumped inputs, defined as the structure <1 × *nU* cell>. Each cell of this structure contains one step response, specified in the form <*Nodes* × *N* double>, where the variable *Nodes* is the number of points at which the distributed quantity is given and *N* is the length of the transient response in steps. Dimensions of *Nodes* and *N* must be the same in every cell structure.

If all distributed responses are also distributed transient responses, i.e. they are responses to a unit step, then the parameter *„Input-step magnitudes“* is a <1 × *nU* double> structure equal to the vector: ones(1, nU). Otherwise, this parameter must be specified for each lumped input separately. The parameter *„Sample time“* is a common sampling period of distributed transient characteristics.

6.1.2 Block: RHLDS

The RHLDS block calculates the reduced partial responses of the controlled system (HLDS) at the sampling times, (Fig. 6.5).

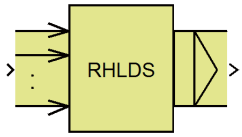


Fig. 6.5 Icon of RHLDS block.

The input is a vector of lumped inputs $\{U_i(k)\}_i$, specified in the form of an *i*-dimensional vector at each sample *k*. The output is a distributed output, given in the form of reduced spatial discrete outputs $\{YR_i(\boldsymbol{x},k)\}_i$, computed by the equation

$$\{YR_i(\boldsymbol{x},k) = Y_i(\boldsymbol{x},k) / Y_i(\boldsymbol{x}_i,k)\}_i \tag{6.4}$$

where $\{Y_i(\boldsymbol{x}_i,k) \neq 0\}_i$. The field of definition $\{\boldsymbol{x}_i\}_i$ is chosen as a reference point, where the steady-state progress of the individually spaced distributed characteristics reach their maximum amplitude. Partial distributed outputs $\{Y_i(\boldsymbol{x},k)\}_i$ are evaluated by the discrete convolution in Eq. (6.1). The distributed output is then given by

$$Y(\boldsymbol{x},k) = \sum_{i=1}^{nU} Y_i(\boldsymbol{x},k) = \sum_{i=1}^{nU} Y_i(\boldsymbol{x}_i,k)\, YR_i(\boldsymbol{x},k) \tag{6.5}$$

It is obvious that the RHLDS block is a modification of HLDS, which reflects the impact of nU lumped inputs on nU reduced spatial discrete outputs $YR_i(\boldsymbol{x},k)$ in each sample. For this reason, the input parameters of the RHLDS block are identical to the input of the HLDS block.

6.1.3 Block: DPS Space Synthesis

This is a block governing the spatial aspects of the control synthesis, (Fig. 6.6).

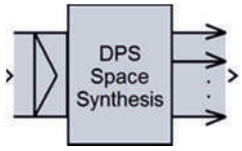


Fig. 6.6 Icon of the Space synthesis block.

The input of the block $P(\boldsymbol{x},k)$ is in the form of a temporally and spatially distributed quantity. At step k it is denoted as $\boldsymbol{P}(k)$. The output of the block is a coefficient vector of the spatial approximation of the distributed quantity, given by the equation

$$\begin{aligned}\breve{\boldsymbol{\alpha}}(k) &= \arg\min_{\boldsymbol{\alpha}(k)}\left\| P(\boldsymbol{x},k) - \sum_{i=1}^{nU} \mathcal{H}\breve{H}R_i(\boldsymbol{x},\infty)\,\alpha_i(k)\right\| = \\ &= \arg\min_{\boldsymbol{\alpha}(k)}\left\|\boldsymbol{P}(k) - \boldsymbol{HR}\cdot\boldsymbol{\alpha}(k)\right\|\end{aligned} \tag{6.6}$$

where the value of the coefficient can be limited by boundaries $\{lb_i \le \alpha_i \le ub_i\}_{i=1,nU}$.

The columns of matrix $\boldsymbol{HR}$ are formed by the reduced discrete transient characteristics in steady-state $\{\mathcal{H}\breve{H}R_i(\boldsymbol{x},\infty)\}_{i=1,nU}$, or by reduced partial responses $\{YR_i(\boldsymbol{x},k)\}_{i=1,nU}$ in step k. Similarly as above $\{\mathcal{H}\breve{H}R_i(\boldsymbol{x},\infty) = \mathcal{H}\breve{H}_i(\boldsymbol{x},\infty)/\mathcal{H}\breve{H}_i(\boldsymbol{x}_i,\infty)\}_{i=1,nU}$, $\{\mathcal{H}\breve{H}_i(\boldsymbol{x}_i,\infty) \ne 0\}_{i=1,nU}$, (where $\{\mathcal{H}\breve{H}_i(\boldsymbol{x},\infty)\}_{i=1,nU}$) are distributed transient characteristics in steady-state. Furthermore, $\{\boldsymbol{x}_i\}_{i=1,nU}$ are the locations at which these characteristics reach their maximum amplitude, $\breve{\boldsymbol{\alpha}}(k)$ is an optimal coefficient vector of the approximation task in time step k and nU is the number of lumped inputs.

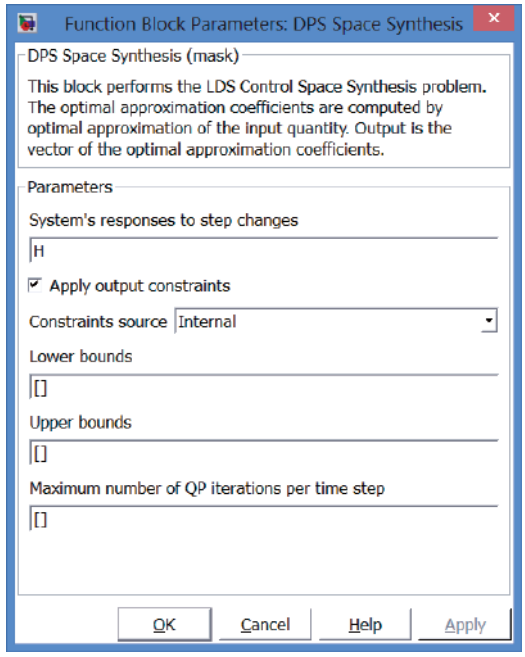


Fig. 6.7 Mask of Space Synthesis block.

The parameter H in the mask (Fig. 6.7) is a file of $\left\{\mathcal{H}H_i\left(\boldsymbol{x},k\right)\right\}_{i=1,nU}$. In case the use of constraints is required, this can be done through the „*Apply output constraints*" option. Constraints are lower and upper values „*lb"* and „*ub"*. Upon activating the option „*external*", it is possible to provide these constraints by the external input ports of the block. A quadratic programming procedure, specifying the number of iterations „*maxiter*" is used to solve this task, (Ferreau (2007-2014)).

6.1.4 Block: DPS Time Synthesis

The block for the temporal aspect of the control synthesis is essentially designed to work with a set of SISO controllers, (Fig. 6.8).

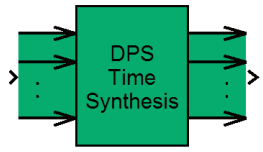


Fig. 6.8 Icon of the Time synthesis block.

This block automatically creates a set of regulators working in parallel according to the scheme specified by the user in the form of a Simulink model template. In case

there are several controllers, the block simplifies the tedious and repetitive task of creating controllers and addressing variables, (Fig. 6.9)

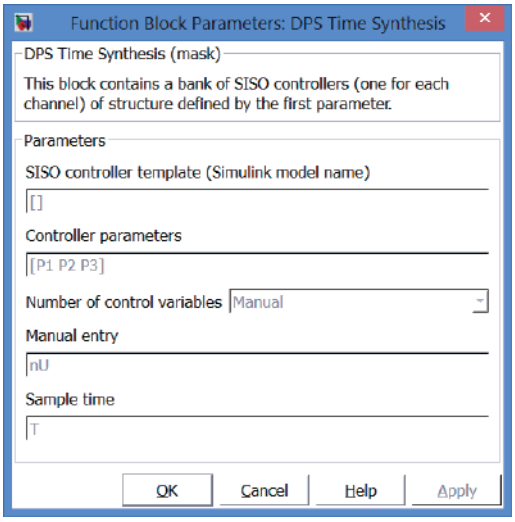


Fig. 6.9 Mask of Time Synthesis block.

Example 6.1

Let us create a set of five PID controllers, where each controller is defined by the equation

$$C(s) = P + I\left(\frac{1}{s}\right) + D\left(\frac{Ns}{s+N}\right) \tag{6.7}$$

where P, I, D are independent controller constants, while the derivative component contains a filter with one pole whose location is defined by the constant N. The corresponding control scheme as the subsystem in Simulink model is shown in the Fig. 6.10.

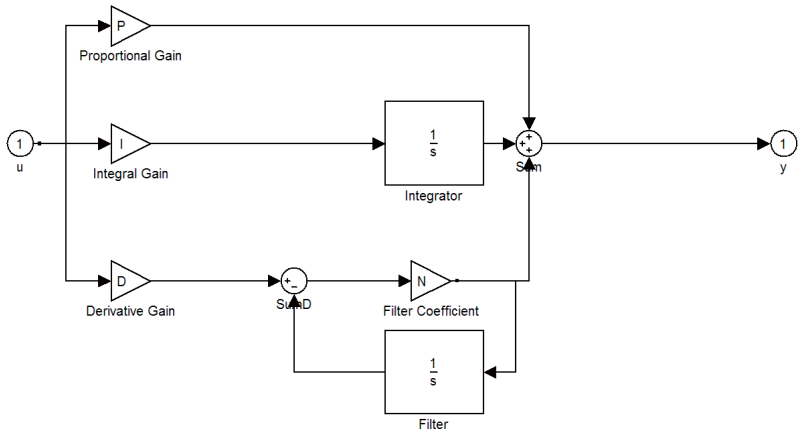


Fig. 6.10 Scheme of a parallel PID controller as the subsystem in Simulink model.

Let us assign a mask to the subsystem in Fig. 6.10 and save this Simulink model under any name in the current folder (e.g. Custom_pid.mdl). The values of the constants are transferred to the controller, so the names of constants in the column „*Variable*“ are identical to the names of constants in the original scheme itself. Initialize the parameters to any number in the dialog box of the mask subsystem and then save the model.

#	Prompt	Variable	Type	Evalu...	Tunab...	Tab name
1	Const. P:	P	edit	☑	☑	
2	Const. I:	I	edit	☑	☑	
3	Const D:	D	edit	☑	☑	
4	Const N:	N	edit	☑	☑	

Fig. 6.11 Parameters of the subsystem mask.

Each controller requires a set of P, I, D and N constants included in structures of type <1 × 5 cell>. These need to be present in the Matlab Workspace during simulation.

Now, let us create a Simulink model (in the current folder) in which we wish to use the controller Custom_pid.mdl. Insert a DPS Time Synthesis block from DPS Blockset library to the new Simulink model and set the parameters in Fig. 6.9 as follows:

SISO controller template (Simulink model name): custom_pid

Controller parameters: [Kp Ki Kd N].

Manual entry: 5

It is possible to enter the numbers of controllers directly by value (Manual entry) or it is set by the number of transient characteristics (Get from system characteristics). After pressing the OK button, the DPS Time Synthesis Block creates a set of 5

subsystems connected in parallel, while each subsystem contains a scheme defined in Simulink model Custom_pid.mdl with completed masks.

■

6.1.5 Block: MtoP

The interface designed to exchange data between the ProCAST and Matlab & Simulink environment is included in the block titled MtoP, (Fig. 6.12).

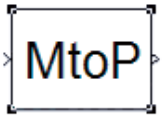


Fig. 6.12 Icon of the MtoP block.

A more detailed description of the co-simulation settings and data exchange in terms of functionality is shown in the Section *6.2 Co-simulation*. The mask of the block is shown in the Fig. 6.13.

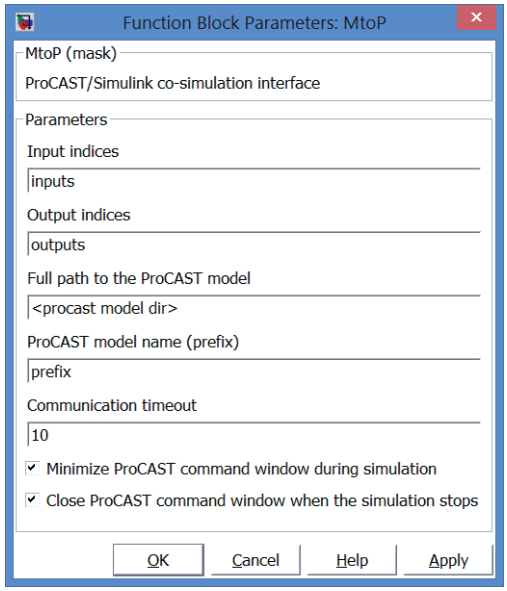


Fig. 6.13 Mask of MtoP block.

Input indices	Vector with indices of boundary conditions (numBC) in ProCAST models
Output indices	Vector with numbers of nodes, which provides data on the temperature for co-simulation.

Full path to the ProCAST model	Absolute path to the folder with the ProCAST model, entered without apostrophes.
ProCAST model name (prefix)	ProCAST model name (prefix), entered without apostrophes.
Communication timeout	The length of time in seconds for communication timeout.

In case the communication between the software packages is not established, the software will prompt the user to hold or break the connection. Pressing „*Yes*“ will attempt to keep the connection active, while „*No*“ will break the connection.

We can set the behavior of the ProCAST command window by the selectable options *Minimize ProCAST command window during simulation* and *Close ProCAST command window when the simulation stops.*

Remark: Indices are filled in the ProCAST simulation as integer data types (int). Obtaining indices of the boundary conditions (*numBC*) used for calling specific *User functions*, is possible via the statement list displayed on the screen during the ProCAST simulation. Indices of node points (*node#*) can be determined after the results of the ProCAST simulation are displayed, resp. by decoding the binary file *prefixg.unf*.

6.1.6 Block: DPS Scope

This block is used to display a distributed quantity during a running simulation, (Fig. 6.14).

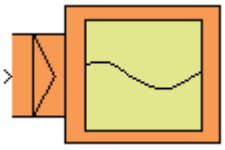


Fig. 6.14 Icon of the DPS Scope block.

The distributed quantity is displayed at discrete times along the user-defined mesh. The mesh is defined in the Cartesian coordinate system by using a matrix of nodes and the elements of the matrix. For simple meshes, it is sufficient to create these matrices by hand or by using a text file that describes a finite element mesh generated by an external software. At this time, the DPS Blockset does not support importing a mesh directly from external numerical software packages. The Matlab function *geom_32d* is a part of the DPS Blockset, it extracts the surface of mesh elements, which are then written in a matrix of elements for tetra- or hexahedrons. The mask of the block is shown in the Fig. 6.15.

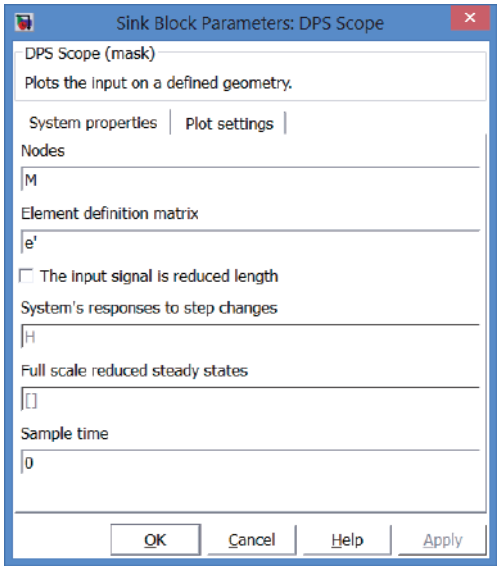


Fig. 6.15 Mask of the DPS Scope block.

Nodes – Matrix of coordinates of the nodal points. It forms a link between the node index and its coordinates in a Cartesian system. It holds that:

- number of matrix lines = total number of nodes,
- row position in the matrix = nodal index,
- 1st, 2nd, 3rd column of matrix = [*x*, *y*, *z*] coordinate of defined nodal point.

Element definition matrix – Matrix of elements. It forms a link between indices of the nodal points and peaks (edges) of planar elements, resp. edges of a bar element. It holds that:

- number of matrix rows = total number of nodes,
- row position in the matrix = nodal index.
- 1-th to *n*-th matrix column = 1. to *n*-th nodal point of defined element.

It is possible to combine different element types within the matrix, for example the bordering triangle and quadrilaterals, which are used by mesh surfaces to map curved edges. In this case, the bordered triangle is "expanded" to a quadrilateral, by the repeated entry of a particular node. This matrix cannot contain zeros or negative numbers.

Interpolation of the distributed quantity between element nodes works properly for quadrilateral, triangular and bar elements (2 nodes) only.

Example 6.2

We can define a matrix of 6 planar elements in the shape of a square and matrix of 8 nodal points in the Cartesian coordinate system for displaying a volume element in the shape of a cube (hexahedron with unit length edges), (Fig. 6.16).

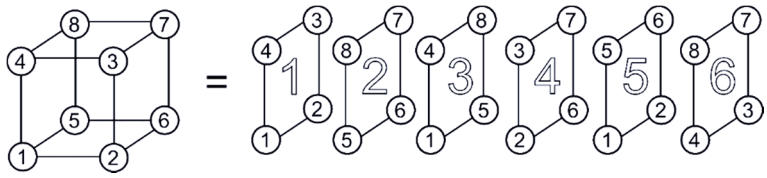


Fig. 6.16 Volume element with indices of nodal points and elements.

Each planar element is defined by the indices of 4 nodal points counter-clockwise, no matter what the position of the starting node is. Entering the two matrices into Matlab will involve creating:

- Matrix of elements by (6, 4):
- E = [1 2 3 4; 5 6 7 8; 1 5 8 4; 2 6 7 3; 1 2 6 5; 4 3 7 8].
- Matrix of nodal point coordinates (edge with unit length) by (8, 3):
- M = [0 0 0; 0 1 0; 0 1 1; 0 0 1; 1 0 0; 1 1 0; 1 1 1; 1 0 1].

To display the surface mesh consisting of two hexahedrons, only 10 planes are required, since two of those are continuous, (Fig. 6.17).

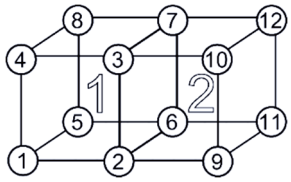


Fig. 6.17 Bordered volume elements.

The dimension of element matrix is (2, 8), for example E = [1 2 3 4 5 6 7 8; 2 9 10 3 6 11 12 7]. It is possible to remove bordered planes by using the *geom_32d* function which identifies the repeating nodal points [2 3 6 7]. The returned value of this function is a matrix of surface elements.

■

In case we solve the time synthesis only on a limited area, but want to display results on the whole area, we must use „*The input signal is reduced length*“ function. Missing values for nodal points defined by the matrix of nodal points are specified by a computation according to the steady-states on a full scale area (*Full scale reduced steady-states*). In this case, the number of rows of the matrix of nodal points is different than the number of elements of the displayed vector. The number of rows is

given by the type of elements. The rest of the parameters of the block are described in the block DPS Space Synthesis. In *Plot settings* one may set up the deformation of the rendered object in the direction of the selected axis of the coordinate system.

6.1.7 Block: Norm

This is a block to calculate the Euclidian vector norm, (Fig. 6.18).

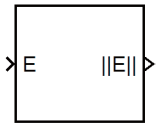


Fig. 6.18 Icon of Norm block.

The Euclidian norm of input vector $\|\boldsymbol{E}\| = \sqrt{\boldsymbol{E} \cdot \boldsymbol{E}}$ will be computed, where the operation $\boldsymbol{E} \cdot \boldsymbol{E} = \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}$ is a scalar multiplication of input vector $\boldsymbol{E}$. For this operation, the block of Dot Product from Simulink/Math Operations is used.

6.2 Co-simulation

Co-simulation is the concurrent run of Matlab & Simulink and ProCAST during the simulation of the HLDS block. At the layer of the Simulink model, the algorithm of co-simulation is implemented as a *Level-2 S-function in C++*. The C-MEX S-function is compiled in a *mexw32* format (a Matlab Executable on a 32-bit Microsoft Windows system). At the layer of the ProCAST model, this co-simulation algorithm is implemented with a special function named _externalcompute.c, which is run by the ProCAST solver (numerical solver) at the beginning and end of the computation and time step. Both mentioned features are part of the DPS Blockset with the relevant *include* files.

The start and end of the co-simulation procedure is controlled from the Simulink model window. Therefore, the time sequence of data exchange between the Simulink model and the ProCAST simulation is controlled by the above algorithm implemented as a C-MEX S-function. Each Matlab S-function must perform a set of operations referred to as callback methods. For the C-MEX S-functions, the methods *mdlInitializeSizes, mdlInitializeSampleTimes, mdlOutputs* and *mdlTerminate* are obligatory and are performed in the initialization of a simulation loop.

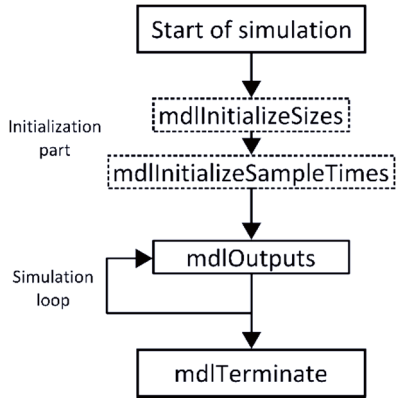


Fig. 6.19 The sequence of mandatory methods within C-MEX S-Functions.

Other (optional) methods can be included amongst the sequence of mandatory methods, depending on the type of the required algorithmic task. The solver of the Simulink model needs to be set to *discrete* with a *variable time step*. This type of solver is crucial because the ProCAST solver chooses a variable time step different from Simulink model time step. The algorithm C-MEX S-function should use the method *mdlGetTimeOfNextVarHit*. Utilizing this method, one can enter the next step by running the Simulink model. In case the ProCAST computation does not converge, the ProCAST time step will be repeated with a lower value. Since Simulink does not support such behavior, it must wait until the ProCAST will eventually converge and move into a new time step.

The method *mdlGetTimeOfNextVarHit* is called at initialization time, before the mandatory *mdlOutputs* method. If the Simulink model contains a number of S-functions, each method is sequentially triggered by the Simulink. The C-MEX S-function for co-simulation may be inserted into the model only once. The function is inserted from the Matlab / ProCAST co-simulation block with other input parameters entered in the mask block. A synchronized data exchange occurs during co-simulation at runtime between the processes at the level of the operating system through *Named Pipes*. The method is characterized by a simple deployment in the program code and balances the cost of memory and performance. Alternative solutions of inter-process communications are: socket, file system and shared memory. The following Table 6.1 shows the interaction of C-MEX S-Functions and the features ProCAST *_externalcompute.c* with regards to the different stages of the computation.

Simulink	C-MEX S-function	_externalcompute.c	ProCAST
Model Initialization	Creating a named pipe (server mode); The allocation of the required amount of memory from the data entered in the mask of the co-simulation block; Loading and updating the data file prefixp.dat; Starting the Datacast and ProCAST solver task called prefix.	Establish communication with the server via a named pipe (client mode).	funLoop0(&time) / Start calculation
Simulation loop / Major time step / mdlGetTimeOfNextVarHit.	Sending input vector for the block ProCAST user functions; Acceptance of a new time step size; Enter a new time step or a waiting.	Acceptance of vector data for ProCAST user functions; Send a new time step size.	funLoop1(&time) / Beginning of the current time step
Simulation loop / Major time step / mdlOutputs	Receiving and transferring data to the block of co-simulation output; Verifying the value of the current simulation time and, accordingly, return or exit from the simulation loop.	Filling data vector for output block co-simulation; Beginning of a new time step or acceptance of a request for termination of co-simulation.	funLoop2(&time) / End of the current time step
End Simulation / mdlTerminate	Sending a termination request to ProCAST simulation; Closure of named pipes and release of allocated memory.	Closure of communication with the server via a named pipe; Closure of ProCAST simulation.	funLoop3( )

Table 6.1 The interaction of C-MEX S-Functions and _externalcompute.c with respect to different phases of the calculation.

6.2.1 ProCAST Functions

The ProCAST software provides the option to use special features programmable in C. This handy extension allows implementing the idea of co-simulation. These functions can be divided into three categories according to the method used:

- *User functions*, which are intended for the definition of the conditions used in the calculation, instead of the constant value or a function of time or temperature.
- *External functions*, which return the value of the field quantities in the calculation and are called by the *User functions*.
- *External Computation function*, which calls ProCAST solver in various stages of calculation.

The algorithm of co-simulation is implemented on the ProCAST layer with a modification of the *External Computation function*. The modification consists in the function calls:

- funLoop0(double *)
- funLoop1(double *)
- funLoop2(double *)
- funLoop3( ).

The use of a specific *User function* depends on the type of the models. For example, the control of the temperature field by changing thermal conditions on the surface requires *HEAT Boundary Condition*. This condition can be entered by the following combination of *User functions*:

- Heat transfer coefficient on the surface of the object prefix_convehtransfer.c
- The emissivity of an object prefix_emissivity.c
- Ambient temperature of an object prefix_texternal.c
- The heat flux on the surface of an object prefix_heatflux.c

A unique index is assigned to the *HEAT boundary conditions* during calculation, which can be used within the specified *User functions* as a *numBC* variable of type *int*.

It is also possible to call any *External functions* within the *User function* as:

- The temperature of the node point usertemp1(node#)
- The proportion of solid phase in the node point userfs1(node#)

When using functions that link to a specific node with the ProCAST *parallel solver,* the *Distributed Memory Processing* architecture should be used with the *Global* extension.

6.2.2 Co-simulation Setup

Separate files for Matlab and ProCAST are placed into a special folder. The following figure shows the Simulink model folder (*Current Folder*), created as the ProCASTpart subfolder that contains all the files necessary in ProCAST.

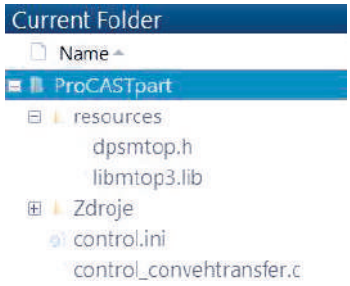


Fig. 6.20 Example of the arrangement of folders when setting up co-simulation.

The ProCASTpart folder contains a subfolder *resources*, copied from the location matlabroot / toolbox / dpstool. Typing *matlabroot* in the *Command Window* gets the location of the running Matlab installation. The sources subfolder is optional for users and may include the header files for the *User functions* of ProCAST. Other files within the folder ProCASTpart concern the computing tasks in ProCAST. The header file *dpsmtop.h* contains a declaration of functions

- double * getMemBuffer(int),
- long * getIndexes(int),
- long getSizeOf(int),

with the input integer argument named INPUTBUFFER, OUTPUTBUFFER type *int*. The significance of these arguments is related to the input / output port of the co-simulation block (MtoP). The term *buffer* refers to a common memory space reserved for data exchange within co-simulation. The following figure shows the co-simulation block included in the control system (CS), from which the inputs $U(k)$ are computed in the given time sequence. A vector of values of the distributed output $Y(\boldsymbol{x},k)$ at the output of the co-simulation block is provided at selected nodal points of the finite element mesh.

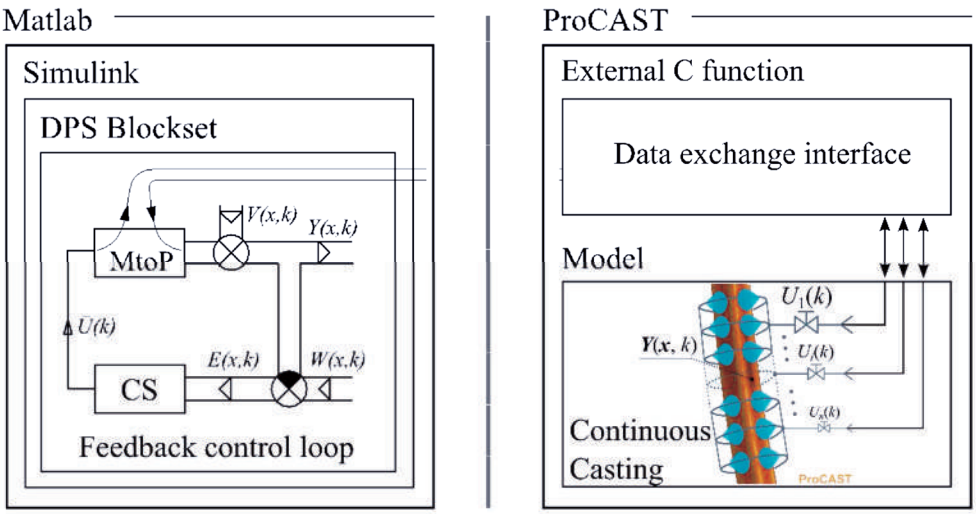


Fig. 6.21 Linking Matlab and ProCAST through block MtoP.

A table of return values of functions for both values of the input argument is given in accordance with the figure.

Function	**int = INPUTBUFFER**	**int = OUTPUTBUFFER**
getMemBuffer(int)	Pointer to the array (vector) of inputs $U(k)$ type *double*.	Pointer to the array (vector) of distributed output $Y(\boldsymbol{x},k)$ type *double*.
getIndexes(int)	Pointer to the array of inputs indices, resp. indices array of boundary conditions *numBC* type *long int*.	Pointer to the array of nodal points indices, which defines the distributed output type *long int*.
getSizeOf(int)	Numbers of elements by array of action inputs type *long int*.	Numbers of elements by array of distributed output type *long int*.

Table 6.2 Return values for both functions of the input argument.

Return values of these functions can be used for programming specific ProCAST *User functions*. *Input indices* and *Output indices* are first input parameters in co-simulation block mask, see example in Fig. 6.22



Fig. 6.22 Mask of the co-simulation block: Input parameter indices/Output indices.

In this case, the parameter *Input indices* consists of *inputs* and number 0. Within the variable *inputs* specific indices of boundary conditions *numBC* are entered, as they were established in the preparation phase of the computational task in ProCAST. If this variable is not used within the *User function*, but the values of inputs from the Simulink model are directly used (through getMemBuffer(INPUTBUFFER) function), elements of the *inputs* vector can be of any *n* indices on the boundary condition. This is true even if we use several inputs within a single *User function*. The *Output indices* parameter represents numbers of nodal points in finite element mesh.

Example 6.3

A ProCAST simulation of continuous casting on a curved casting strand uses one *HEAT* boundary condition, defined by the function *prefix_heatflux.c* and the transport condition *Solid transport* defined by functions *prefix_vxsolidtransport.c*, *prefix_vysolidtransport.c*. Simulink computes the inputs for five sections of the secondary cooling, which are passed to ProCAST by the function getMemBuffer (INPUTBUFFER) within a function *prefix_heatflux.c*. As the *numBC* is not used in *prefix_heatflux.c*, the Matlab variable *inputs* may include any five indices of ProCAST boundary conditions, for example *inputs* = [1 1 1 1 1]. The same applies to the condition of the casting speed, whose value is inserted as the 6-th element of the array.

■

At the path specified in *Full path to the ProCAST model*, the software is searching for text files *prefixd.dat* and *prefixp.dat*, where *prefix* is title of ProCAST simulation (case).

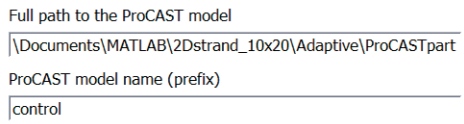


Fig. 6.23 The mask of the co-simulation block: Full path to the ProCAST model / prefix parameters.

In the file *prefixp.dat* checking and editing of parameters NSTEP and TFINAL is carried out. The parameter TFINAL must be less than the *Simulation stop time* of the Simulink model. The parameter NSTEP must respect the same principle. The program *Datacast* with the parameter *ProCAST model name (prefix)*. By this, all necessary files to run the ProCAST simulation will be created.

Remark: the program *Datacast* is started without the parameter *update option* and will overwrite all result files of the previous ProCAST simulation.

Co-simulation is started from the Simulink model by the *Run* button (Ctrl+T). A C-MEX S-Function (mexw32 format) is automatically inserted in the current folder,

along with a Simulink model title and *Simulink project folder* (slprj), which is managed by Simulink. Checking a Simulink model without the initialization is done by calling the *Update diagram* function (Ctrl+D).

6.3 Interactive Control

In an effort to offer an online support for the modelling and control of distributed parameter systems, the internet service Interactive Control was created for academic purposes. The service is available through the www.dpscontrol.sk web site in a client-server mode. The advantage of this service is that the client does not need to have Matlab / Simulink / DPS Blockset installed to evaluate benchmark problems.

The service offers model examples of technological processes, published within the dissertation theses discussing the distributed parameter control of processes like plastic extrusion, experimental smart foundry molds, induction heating and the continuous casting of steel. The outputs from these control processes are displayed after loading the results for the set-point, manipulated variable, input and the control error. Users of the service may specify or modify selected parameters of the control loops. It is possible to enter a distributed output and set-point after specifying the type of the controlled system. Furthermore, the service allows the tuning of PID controllers for specific control loops in an automatic spatial synthesis mode.

7. PREPARATION FOR THE APPLICATIONS

This chapter is devoted to the preparation steps that are necessary for the practical application of distributed parameter control, based on the formerly introduced methodical results and the possibilities implied by the software support for DPS. The control feedback loops suggested in the previous discussion must be adapted to use in the typical technical environment of technological and production processes, near the selected steady-state operating modes or, for example, transitioning in-between operating modes.

7.1 Control by Linearized Part of Controlled System Dynamics

The controlled technological or production process can be generally described by a nonlinear discrete lumped input and distributed parameter output system, let us denote this concept with the NHLDS abbreviation. Let us assume, that this system is working in a steady-state operating mode, using the input and output pair of $\bar{U}N(\infty)$ and $YN(\boldsymbol{x},\infty)$. We may express the input and output in the vicinity of this steady-state operating mode by $\bar{U}N(\infty)+\bar{U}(k)$ and $YN(\boldsymbol{x},t)=YN(x,\infty)+Y(\boldsymbol{x},t)$. Meanwhile, the relationship between $\bar{U}(k)$ and $Y(\boldsymbol{x},t)$ is described by the linearized part of NHLDS, represented by a linear discrete lumped input and distributed parameter output system, or HLDS (Fig. 7.1).

The desired changes (in other words the reference) on the output side $YN(\boldsymbol{x},t)$ may be ensured by a corresponding sequence of inputs $\bar{U}(k)$ on the level of the HLDS block. As it was formerly implied, the prescribed changes on the HLDS level may be generated by a distributed parameter control loop (Fig. 5.6). The blocks of the distributed parameter control system CS are designed and optimized based on the dynamic characteristics of the HLDS. As the dynamics of HLDS is a part of the dynamics of NHLDS, it is necessary to connect the control system CS of HLDS block (Fig. 5.6) directly to the NHLDS block (Fig. 7.2).

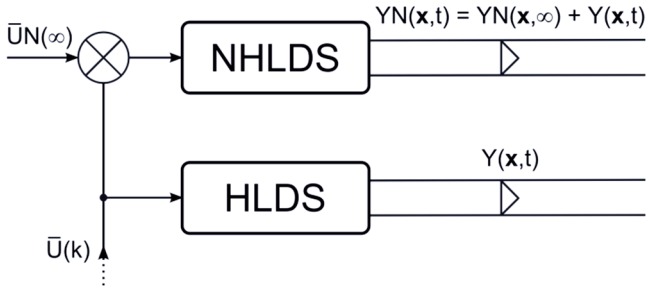


Fig. 7.1 Relationship between the nonlinear and the linearized part of the dynamics of the plant near a steady-state operating mode.

NHLDS - nonlinear discrete lumped input and distributed parameter output system with zero order hold units.

HLDS - linear discrete lumped input and distributed parameter output system with zero order hold units.

$\bar{U}N(\infty)$ - input vector of the NHLDS in steady-state.

$YN(\boldsymbol{x},\infty)$ - output of the NHLDS in steady-state.

$\bar{U}(k)$ - input vector of HLDS and NHLDS.

$YN(\boldsymbol{x},t)$ - output of the NHLDS.

$Y(\boldsymbol{x},t)$ - output of HLDS and linearized part of the NHLDS output.

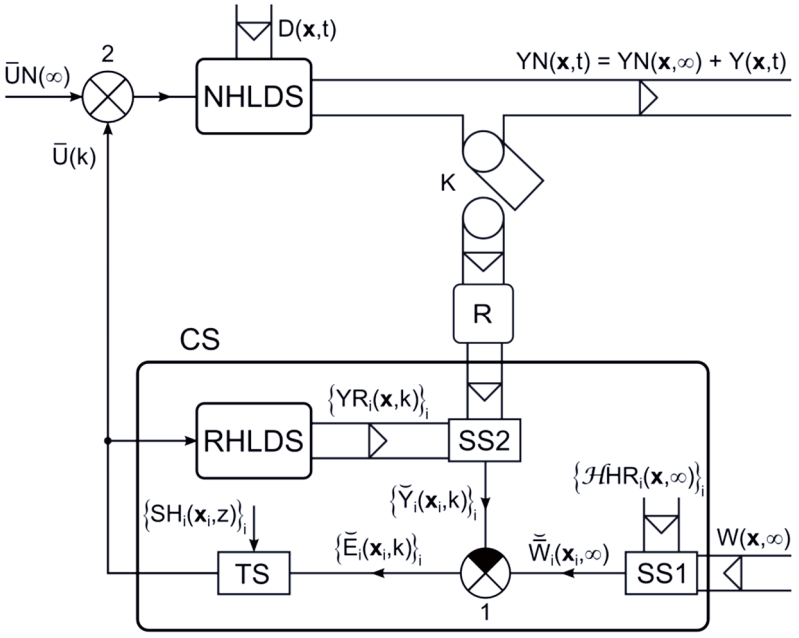


Fig. 7.2 Control of the NHLDS system near the linearized neighborhood of the selected operating mode.

NHLDS - nonlinear discrete lumped input and distributed parameter output system with zero order hold units.

RHLDS - generator of reduced characteristics.

CS - control system.

R - block of differences.

TS - temporal control synthesis.

SS1, SS2 - spatial control synthesis.

K - sampling.

$\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_i$ - spatial components of the dynamics of the controlled system.

$\{SH_i(\boldsymbol{x}_i,z)\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\bar{U}N(\infty)$ - vector of steady-state inputs.

$D(\boldsymbol{x},t)$ - continuous distributed disturbance of the NHLDS block.

$W(\boldsymbol{x},\infty)$ - overall distributed reference in the steady-state.

$\breve{W}_i(\boldsymbol{x}_i,\infty)$ - distributed reference vector in the steady-state.

$YN(\boldsymbol{x},t)$ - distributed manipulated variable of the NHLDS block.

$YN(\boldsymbol{x},\infty)$ - distributed manipulated variable of the NHLDS block in steady-state.

$Y(\boldsymbol{x},t)$ - manipulated variable on the HLDS level.

$\{Y_i(\boldsymbol{x}_i,k)\}_i$ - components of the partial distributed manipulated variables in locations $\{\mathbf{x}\}_i$ and samples k on the HLDS level.

$\{\breve{E}_i(\boldsymbol{x}_i,k)\}_i$ - components of control error.

$\{YR_i(\boldsymbol{x},k)\}_i$ - components of the reduced partial manipulated variables on the HLDS level.

We connect the blocks of the distributed parameter control system (CS) (Fig. 5.6) to the controlled plant, which is represented as a nonlinear system with lumped inputs and distributed outputs (the NHLDS block) working near the chosen operating mode. In order to ensure the changes that are required on the output side $YN(\boldsymbol{x},t)$, this control system generates a suitable sequence of inputs $\bar{U}(k)$.

The fixed steady-state operating mode is associated with an input of $\bar{U}N(\infty)$, producing outputs $YN(\boldsymbol{x},\infty)$. The block R evaluates the difference between

$$Y(\boldsymbol{x},k) = YN(\boldsymbol{x},k) - YN(\boldsymbol{x},\infty) \tag{7.1}$$

All inputs are zero, that is $\bar{U}(k) = 0$ and $\{\breve{W}_i(\boldsymbol{x}_i,\infty) = \breve{Y}_i(\boldsymbol{x}_i,\infty) = \breve{E}_i(\boldsymbol{x}_i,\infty) = 0\}_i$ in the steady-state, when $D(\boldsymbol{x},t) = 0$, $W(\boldsymbol{x},\infty) = 0$ along with $YN(\boldsymbol{x},t)_{t\to\infty} = YN(\boldsymbol{x},\infty)$, $Y(\boldsymbol{x},t)_{t\to\infty} = 0$.

- ***Disturbance rejection in the selected steady-state operating mode***

When disturbance affects the controlled plant operating in the neighborhood of a certain linearized steady-state mode, that is $D(\boldsymbol{x},t) \neq 0$; the block R will evaluate the difference $Y(x,k) = YN(\boldsymbol{x},k) - YN(\boldsymbol{x},\infty)$ that enters the feedback part of the control system (see Fig. 5.6), where an input sequence $\bar{U}(k)$ is generated to reject the disturbance according to Sect. *5.2*. (while initial values of RHLDS are set to non-zero level, for example $\{YR_i(\boldsymbol{x},1) = \mathcal{H}HR_i(\boldsymbol{x},1)\}_i$).

- ***Change of the steady-state manipulated variable in the linearized neighborhood of the selected operating mode***

The required change of the manipulated variable with respect to $YN(\boldsymbol{x},\infty)$ is formulated as the reference $W(\boldsymbol{x},\infty) \neq 0$ that is supplied to the SS1 block. The feedback part of the control system, Fig. 5.6, generates a sequence of inputs $\bar{U}(k)$ which will set the distributed manipulated variable on the $YN(\boldsymbol{x},\infty) + \breve{Y}(\boldsymbol{x},\infty)$ level in steady-state. Meanwhile $\breve{Y}(\boldsymbol{x},\infty)$ is the best approximation of $W(\boldsymbol{x},\infty) \neq 0$, according to the discussion presented in Sect. *5.2*.

- ***Transients between steady-state operating modes***

The nonlinear dynamics of the transients is decomposed into linearized segments with the corresponding linear dynamic characteristics and the sequence of references $\{W_m(\boldsymbol{x},\infty)\}_m$ for each segment. The transient between the operating modes is then realized by the sequence of inputs $\{\bar{U}_m(k)\}_m$, which are generated for the individual set-points $\{W_m(\boldsymbol{x},\infty)\}_m$ at the currently active dynamic characteristics defined by switching in blocks of the control feedback system in Fig. 5.6.

Near the operating mode it is possible to design a control system, noted as block CS, (Fig. 5.9), where $Y(\boldsymbol{x},t)_{t\to\infty} = \sum_{1}^{n} Y_i(\boldsymbol{x}_i,t)_{t\to\infty}\, \mathcal{H}HR_i(\boldsymbol{x},\infty)$ holds for the manipulated variable (Fig. 7.3). Processes such disturbance rejection, change of manipulated variable and transients between operating modes take place as in the control loop above.

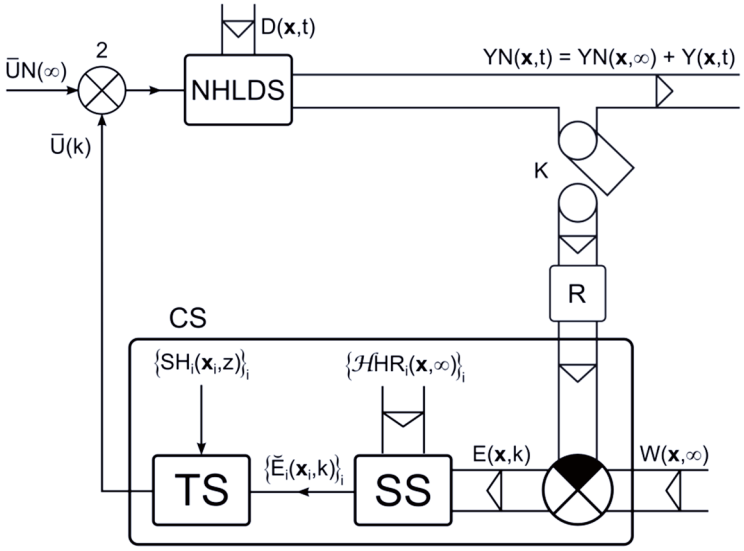


Fig. 7.3 Control of NHLDS near a certain steady-state operating mode according to scheme in Fig. 5.9.

NHLDS - nonlinear discrete lumped input and distributed parameter output system with zero order hold units.

CS - control system.

R - block of differences.

K - sampling.

TS - temporal control synthesis.

SS - spatial control synthesis.

$\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_i$ - spatial components of the dynamics of the controlled system.

$\{SH_i(\boldsymbol{x}_i,z)\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\bar{UN}(\infty)$ - vector of steady-state inputs.

$D(\boldsymbol{x},t)$ - continuous distributed disturbance of the NHLDS block.

$W(\boldsymbol{x},\infty)$ - overall distributed reference in the steady-state.

$\bar{\bar{W}}_i(\boldsymbol{x}_i,\infty)$ - distributed reference vector in the steady-state.

$YN(\boldsymbol{x},t)$ - distributed manipulated variable of the NHLDS block.

$YN(\boldsymbol{x},\infty)$ - distributed manipulated variable of the NHLDS block in steady-state.

$Y(\boldsymbol{x},t)$ - manipulated variable on the HLDS level.

$\{E_i(\boldsymbol{x}_i,k)\}_i$ - components of the partial control error.

The control feedback system designed this way will consider distributed parameters on the entire domain of definition (geometry). In engineering practice, these are e.g. cases where thermal fields may be directly assessed using thermal imaging cameras, such as; welding, forest fires, etc.

7.2 Control Synthesis by a Vector of Controlled Outputs of HLDS

In engineering practice we often encounter processes that are designed as closed systems, with their dynamics working isolated from the surrounding world. This also means, that the distributed manipulated variable can be measured only at certain locations $\{\boldsymbol{x}_i\}_i$. This scenario implies a good match between the dynamics of the controlled processes and the assigned dynamic models.

Elimination of random disturbances is important in every practical application, including the rejection of the measurement noise. We will assume that these random components are removed in the following discussion. Then, in the selected locations of the definition domain $\{\boldsymbol{x}_i\}_i$ (Fig. 7.2) we will obtain vectors that in the block R will be

$$\bar{YN}(\boldsymbol{x}_i,k) = \bar{YN}(\boldsymbol{x}_i,\infty) + \bar{Y}(\boldsymbol{x}_i,k) \tag{7.2}$$

and on the output of the block R will be

$$\bar{Y}\left(\boldsymbol{x}_i,k\right)=\bar{Y}N\left(\boldsymbol{x}_i,k\right)-\bar{Y}N\left(\boldsymbol{x}_i,\infty\right) \tag{7.3}$$

The control system in Fig. 7.2 is based on the distributed parameter control system of Fig. 5.6, using the components of the feedback control system (CS) of this setup. Therefore, our following discussion will concentrate on the analysis of the controller in Fig. 5.6, and the changes we must make. Instead of the distributed output $Y\left(\boldsymbol{x},t\right)$ we will consider the vector of outputs $\bar{Y}\left(\boldsymbol{x}_i,k\right)$ only at certain locations $\left\{\boldsymbol{x}_i\right\}_i$ of the definition domain of the controlled system.

7.2.1 Controller Design by SISO Loops

The following equation will hold between the terms $\bar{Y}\left(\boldsymbol{x}_i,k\right)=\left\{Y\left(\boldsymbol{x}_i,k\right)\right\}_i$ and $\bar{Y}_i\left(\boldsymbol{x}_i,k\right)=\left\{Y_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ (Fig. 7.3). In selected locations $\left\{\boldsymbol{x}_i\right\}_i$ according to Eq. (3.22), (3.26) we have

$$\bar{Y}\left(\boldsymbol{x}_i,k\right)=\bar{Y}R_i\left(\boldsymbol{x}_j,k\right)\bar{Y}_i\left(\boldsymbol{x}_i,k\right) \tag{7.4}$$

Then, instead of Eq. (5.11) the block SS3 will compute

$$\bar{Y}_i\left(\boldsymbol{x}_i,k\right)=\bar{Y}R_i\left(\boldsymbol{x}_j,k\right)^{-1}\bar{Y}\left(\boldsymbol{x}_i,k\right) \tag{7.5}$$

after the inversion of matrix $\bar{Y}R_i\left(\boldsymbol{x}_j,k\right)$, where the elements of the matrix are supplied to the SS3 block via the RHLDS block. Transformation between vectors $\bar{Y}\left(\boldsymbol{x}_i,k\right)$ and $\bar{Y}_i\left(\boldsymbol{x}_i,k\right)$ takes place in the block SS3.

The steady-state control error computed according to Sect. *5.2* $\left\{E_i\left(\boldsymbol{x}_i,k\right)=\breve{W}_i\left(\boldsymbol{x}_i,\infty\right)-Y_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ for $k\to\infty$ corresponding to the steady-state output of the manipulated variable $\breve{Y}\left(\boldsymbol{x},\infty\right)$ represents the best approximation of the distributed reference $W\left(\boldsymbol{x},\infty\right)$. In case a disturbance $V\left(\boldsymbol{x},\mathrm{t}\right)$ is active, it will be also rejected according to the Eqs. (5.9-10).

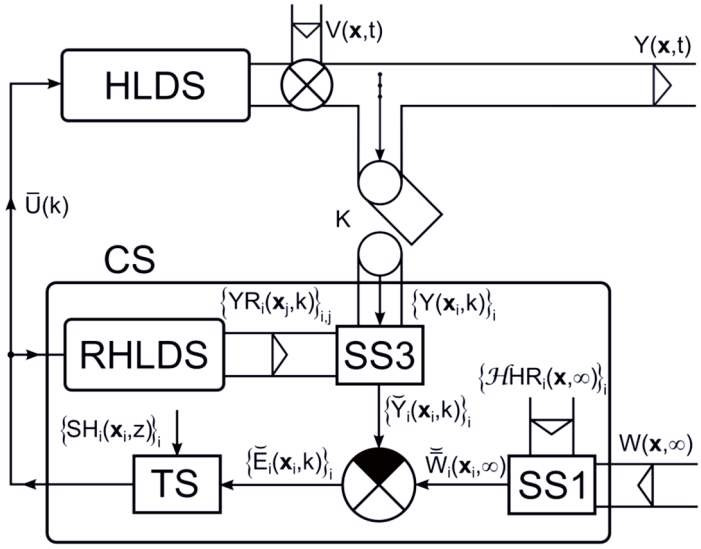


Fig. 7.4 Distributed parameter control feedback loop, in case the manipulated variable is measured at only certain locations of the definition domain.

HLDS - discrete lumped inputs and distributed output system with zero order hold units

RHLDS - generator of reduced characteristics.

CS - control system.

TS - temporal component of the control synthesis.

SS1 - spatial component of the control synthesis.

SS3 - transformation of vectors.

K - sampling.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{\mathcal{H}HR_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}\left(k\right)$ - vector of discrete lumped inputs.

$V\left(\boldsymbol{x},t\right)$ - distributed disturbance.

$W(\boldsymbol{x},\infty)$ - overall reference in steady-state.

$\bar{\bar{W}}_i(\boldsymbol{x}_i,\infty)$ - vector of the lumped reference in steady-state.

$Y(\boldsymbol{x},t)$ - distributed manipulated variable.

$\{Y(\boldsymbol{x}_i,k)\}_i$ - components of the continuous distributed manipulated variable in locations $\{\boldsymbol{x}_i\}_i$ and samples k.

$\{\bar{Y}_i(\boldsymbol{x}_i,k)\}_i$ - components of the reconstructed partial distributed manipulated variables in locations $\{\boldsymbol{x}_i\}_i$ and samples k.

$\{YR_i(\boldsymbol{x}_j,k)\}_{i,j}$ - components of the reduced partial manipulated variables in locations $\{\mathbf{x}_j\}_j$.

The responses $\{YR_i(\boldsymbol{x},k)\}_i$ will converge to the responses $\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_i$ due to the control process, in steady-state and $k\to\infty$ (see Eq. (3.25)). Therefore, when reconstructing the values of $\{Y_i(\boldsymbol{x}_i,k)\}_i$ based on $\{Y(\boldsymbol{x}_i,k)\}_i$ for $k\to\infty$ we may use the approximate relations (7.6-7), as

$$\begin{bmatrix} Y(\boldsymbol{x}_1,k)\\ \vdots\\ Y(\boldsymbol{x}_i,k)\\ \vdots\\ Y(\boldsymbol{x}_n,k)\end{bmatrix} \doteq \begin{bmatrix} \mathcal{H}HR_1(\boldsymbol{x}_1,\infty),...,\mathcal{H}HR_i(\boldsymbol{x}_1,\infty),...,\mathcal{H}HR_n(\boldsymbol{x}_1,\infty)\\ \\ \mathcal{H}HR_1(\boldsymbol{x}_i,\infty),...,\mathcal{H}HR_i(\boldsymbol{x}_i,\infty),...,\mathcal{H}HR_n(\boldsymbol{x}_i,\infty)\\ \\ \mathcal{H}HR_1(\boldsymbol{x}_n,\infty),...,\mathcal{H}HR_i(\boldsymbol{x}_n,\infty),...,\mathcal{H}HR_n(\boldsymbol{x}_n,\infty)\end{bmatrix} \begin{bmatrix} Y_1(\boldsymbol{x}_1,k)\\ \vdots\\ Y_i(\boldsymbol{x}_i,k)\\ \vdots\\ Y_n(\boldsymbol{x}_n,k)\end{bmatrix} \quad (7.6)$$

or in an abbreviated fashion, we may denote this as

$$\bar{Y}(\boldsymbol{x}_i,k) \doteq \bar{\mathcal{H}}HR_i(\boldsymbol{x}_j,\infty)\bar{Y}_i(\boldsymbol{x}_i,k) \quad (7.7)$$

then, instead of Eq. (5.11) we may use

$$\bar{Y}_i(\boldsymbol{x}_i,k) \doteq \bar{\mathcal{H}}HR_i(\boldsymbol{x}_j,\infty)^{-1}\bar{Y}(\boldsymbol{x}_i,k) \quad (7.8)$$

in the SS3 block after the inversion of the matrix $\bar{\mathcal{H}}HR_i(\boldsymbol{x}_j,\infty)$, where the control system (CS) provides the elements for this matrix.

Finally, we will obtain a control system with the measurement of the distributed manipulated variable in selected locations of the definition domain, shown in Fig. 7.5.

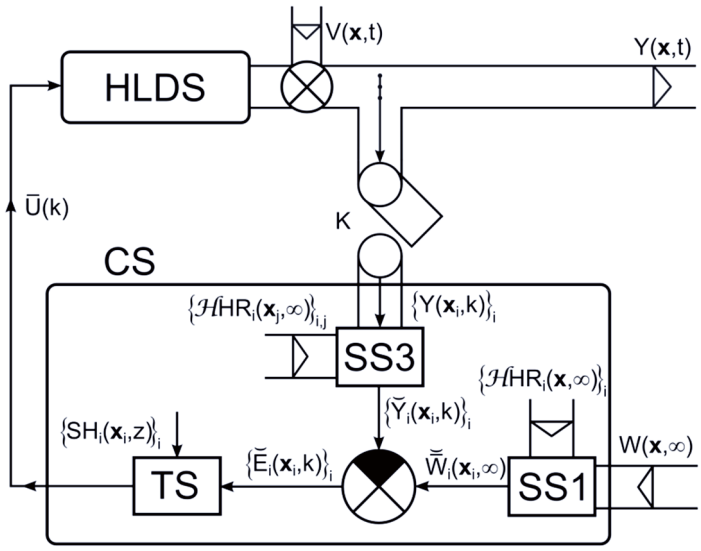


Fig. 7.5 Control system with distributed parameters, in case the manipulated variable is measured at selected points of the definition domain and the system operates near a certain steady-state operating mode.

HLDS - discrete lumped input and distributed output system with a zero order hold units.

CS - control system.

TS - temporal component of control synthesis.

SS1 - spatial component of control synthesis.

SS3 - transformation of vectors.

K - sampling.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{\mathcal{H}HR_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}\left(k\right)$ - discrete lumped input vector.

$V(\boldsymbol{x},t)$ - distributed disturbance.

$W(\boldsymbol{x},\infty)$ - overall distributed reference in the steady-state.

$\breve{\bar{W}}_i(\boldsymbol{x}_i,\infty)$ - lumped reference vector in the steady-state.

$Y(\boldsymbol{x},t)$ - distributed manipulated variable.

$\{Y(\boldsymbol{x}_i,k)\}_i$ - components of the distributed manipulated variable in locations $\{\mathbf{x}\}_i$ at samples k.

$\{\breve{Y}_i(\boldsymbol{x}_i,k)\}_i$ - components of the reconstructed partial distributed manipulated variable in locations $\{\mathbf{x}\}_i$ at samples k.

$\{\mathcal{H}\bar{H}R_i(\boldsymbol{x}_j,\infty)\}_{i,j}$ - components of the reduced characteristics in locations $\{\mathbf{x}\}_j$.

If the relation $Y(\boldsymbol{x},t)_{t\to\infty} = \sum_1^n Y_i(\boldsymbol{x}_i,t)_{t\to\infty}\,\mathcal{H}\bar{H}R_i(\boldsymbol{x},\infty)$ holds at the output of the control loop near a steady-state operating mode, then the control loop in Fig. 7.6 can be assembled. The regulators in the TS block $\{R_i(z)\}_i$ are tuned by the SISO control loops.

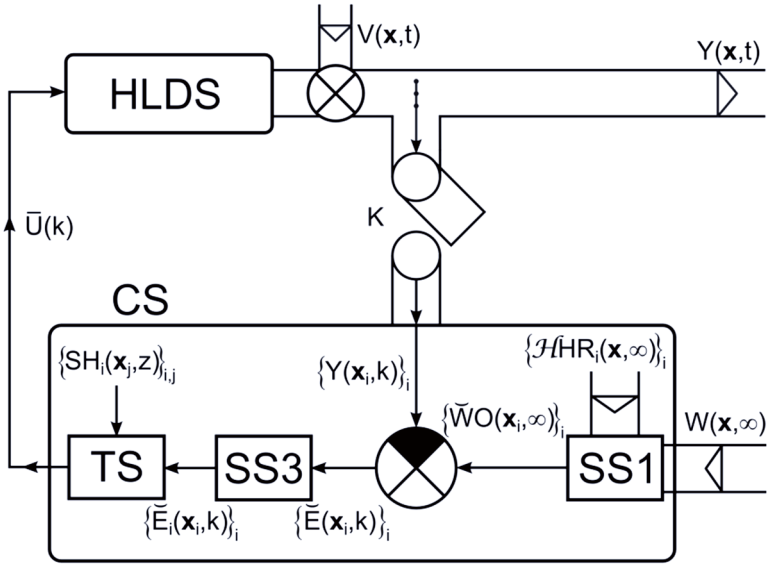


Fig. 7.6 Control system with distributed parameters, in case the manipulated variable is measured at selected locations of the definition domain and the system operates near a certain steady-state operating mode.

HLDS - discrete lumped input and distributed output system with a zero order hold units.

CS - control system.

TS - temporal component of control synthesis.

SS1 - spatial component of control synthesis.

SS3 - transformation of vectors.

K - sampling.

$\left\{\breve{E}\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the overall control error vector.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{\mathcal{H}\breve{H}R_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}\left(k\right)$ - discrete lumped input vector.

$V\left(\boldsymbol{x},t\right)$ - distributed disturbance.

$W\left(\boldsymbol{x},\infty\right)$ - overall distributed reference in the steady-state.

$\breve{\bar{W}}O\left(\boldsymbol{x}_i,\infty\right)$ - lumped reference vector in the steady-state.

$Y\left(\boldsymbol{x},t\right)$ - distributed manipulated variable.

$\left\{Y\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the distributed manipulated variable in locations $\{\mathbf{x}\}_i$ at samples k.

7.2.2 Controller Design by MIMO Loops

Let us now assume that the relationship between the manipulated variable $\{Y(\boldsymbol{x}_i,k)\}_i$ in locations $\{\boldsymbol{x}_i\}_i$ and the inputs $\{U_i(k)\}_i$ may be described according to Eq. (7.9)

$$\begin{bmatrix} Y(\boldsymbol{x}_1,z) \\ \vdots \\ Y(\boldsymbol{x}_i,z) \\ \vdots \\ Y(\boldsymbol{x}_n,z) \end{bmatrix} = \begin{bmatrix} SH_1(\boldsymbol{x}_1,z) & ,..., & SH_i(\boldsymbol{x}_1,z) & ,..., & SH_n(\boldsymbol{x}_1,z) \\ SH_1(\boldsymbol{x}_i,z) & ,..., & SH_i(\boldsymbol{x}_i,z) & ,..., & SH_n(\boldsymbol{x}_i,z) \\ SH_1(\boldsymbol{x}_n,z) & ,..., & SH_i(\boldsymbol{x}_n,z) & ,..., & SH_n(\boldsymbol{x}_n,z) \end{bmatrix} \begin{bmatrix} U_1(z) \\ \vdots \\ U_i(z) \\ \vdots \\ U_n(z) \end{bmatrix} \tag{7.9}$$

Only the diagonal elements of this matrix are considered when performing the control synthesis based on the decomposition of dynamics into its spatial and temporal components. Let us now assume a control process with a full dynamics matrix. According to Chapter *5* we may assign $\breve{W}O(\boldsymbol{x},\infty)$ to the distributed reference $W(\boldsymbol{x},\infty)$, so after all the relation $\breve{Y}(\boldsymbol{x},\infty)=\breve{W}O(\boldsymbol{x},\infty)$ will hold. In locations $\{\boldsymbol{x}_i\}_i$ this means that $\{\breve{Y}(\boldsymbol{x}_i,\infty)=\breve{W}O(\boldsymbol{x}_i,\infty)\}_i$, and for the steady-state we may express the following relation:

$$\begin{bmatrix} \breve{W}O(\boldsymbol{x}_1,\infty) \\ \vdots \\ \breve{W}O(\boldsymbol{x}_i,\infty) \\ \vdots \\ \breve{W}O(\boldsymbol{x}_n,\infty) \end{bmatrix} = \begin{bmatrix} \breve{Y}(\boldsymbol{x}_1,\infty) \\ \vdots \\ \breve{Y}(\boldsymbol{x}_i,\infty) \\ \vdots \\ \breve{Y}(\boldsymbol{x}_n,\infty) \end{bmatrix} = ...$$

$$... = \begin{bmatrix} \mathcal{H}H_1(\boldsymbol{x}_1,\infty),...,\mathcal{H}H_i(\boldsymbol{x}_1,\infty),...,\mathcal{H}H_n(\boldsymbol{x}_1,\infty) \\ \mathcal{H}H_1(\boldsymbol{x}_i,\infty),...,\mathcal{H}H_i(\boldsymbol{x}_i,\infty),...,\mathcal{H}H_n(\boldsymbol{x}_i,\infty) \\ \mathcal{H}H_1(\boldsymbol{x}_n,\infty),...,\mathcal{H}H_i(\boldsymbol{x}_n,\infty),...,\mathcal{H}H_n(\boldsymbol{x}_n,\infty) \end{bmatrix} \begin{bmatrix} \breve{U}_1(\infty) \\ \vdots \\ \breve{U}_i(\infty) \\ \vdots \\ \breve{U}_n(\infty) \end{bmatrix} \tag{7.10}$$

Moreover, based on Eqs. (3.7-3.10) it holds that

$$\breve{Y}(\boldsymbol{x},\infty)=\sum_{i=1}^{n}\breve{Y}_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}\breve{H}R_i(\boldsymbol{x},\infty)=$$
$$=\sum_{i=1}^{n}\breve{U}_i(\infty)\,\mathcal{H}H_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}\breve{H}R_i(\boldsymbol{x},\infty)= \tag{7.11}$$
$$=\sum_{i=1}^{n}\breve{U}_i(\infty)\,\mathcal{H}H_i(\boldsymbol{x},\infty)$$

When the reference on Fig. 7.7 is given as $\left\{\breve{W}O(\boldsymbol{x}_i,\infty)\right\}_i$, the control error will be $\left\{E(\boldsymbol{x}_i,k)=\breve{W}O(\boldsymbol{x}_i,\infty)-Y(\boldsymbol{x}_i,k)\right\}_i$. The time synthesis block contains the controllers $\left\{R_{i,j}(z)\right\}_{i,j}$, which are tuned by MIMO control loops for the particular requirements of the control task, so that the relations $\left\{E(\boldsymbol{x}_i,\infty)=\breve{W}O(\boldsymbol{x}_i,\infty)-Y(\boldsymbol{x}_i,\infty)=0\right\}_i$ hold in the steady-state for $k\to\infty$. Then, according to Eqs. (7.10-11) the distributed outputs of the control loop $\breve{Y}(\boldsymbol{x},\infty)$ will represent the best possible approximation of the reference $W(\boldsymbol{x},\infty)$. In case a disturbance $V(\boldsymbol{x},\mathrm{t})$ is present, then the controller will reject this as well, according to the discussion presented in Sect. *5.2*.

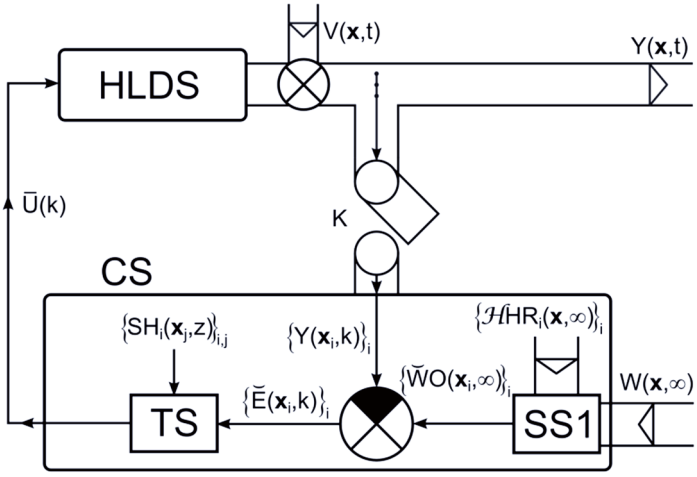


Fig. 7.7 Distributed parameter control feedback loop, in case the manipulated variable is measured at only certain locations of the definition domain based on MIMO synthesis of controllers in TS.

HLDS - discrete lumped input and distributed output with a zero order hold units.

CS - control system.

TS - temporal component of control synthesis.

SS1 - spatial component of control synthesis.

K - sampling.

$\left\{\breve{E}\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the control error vector.

$\left\{\mathcal{H}\breve{H}R_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}\left(k\right)$ - discrete lumped input vector.

$V\left(\boldsymbol{x},t\right)$ - distributed disturbance.

$W\left(\boldsymbol{x},\infty\right)$ - overall distributed reference in the steady-state.

$Y\left(\boldsymbol{x},t\right)$ - distributed manipulated variable.

$\left\{Y\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the distributed manipulated variable in locations $\{\mathbf{x}_i\}_i$ at samples k on the HLDS level.

$\left\{\breve{W}O\left(\boldsymbol{x}_i,\infty\right)\right\}_i$ - components of the best approximation of the continuous distributed reference at locations $\{\mathbf{x}_i\}_i$.

7.3 Control Synthesis by Vector of Controlled Outputs of the NHLDS

The measurement of the distributed outputs $\bar{Y}N\left(\boldsymbol{x}_i,t\right)$ on the definition domain is usually corrupted by noise in a practical technological process. So it will use an approximation $\hat{\bar{Y}}N\left(\boldsymbol{x}_i,k\right)$ of these values and then the difference $\hat{\bar{Y}}\left(\boldsymbol{x}_i,k\right)=\hat{\bar{Y}}N\left(\boldsymbol{x}_i,k\right)-\bar{Y}N\left(\boldsymbol{x}_i,\infty\right)$ will be computed in the block R, which enters the control system, see Fig. 7.8. When an ideal measuring process is considered difference is $\bar{Y}\left(\boldsymbol{x}_i,k\right)=\bar{Y}N\left(\boldsymbol{x}_i,k\right)-\bar{Y}N\left(\boldsymbol{x}_i,\infty\right)$.

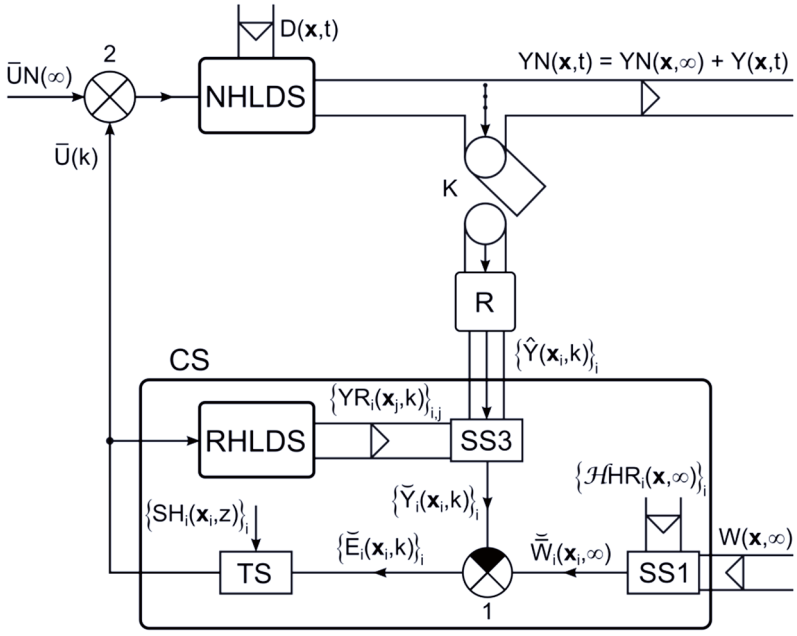


Fig. 7.8 Control of a NHLDS in the linearized neighborhood of the chosen steady-state operating mode, with measurements taken only at selected locations of the definition domain.

NHLDS - nonlinear discrete lumped input and distributed parameter output system with zero order hold units.

RHLDS - generator of reduced characteristics.

CS - control system.

TS - temporal control synthesis.

SS1 - spatial control synthesis.

SS3 - transformation of vectors.

K - sampling.

R - block of differences.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{\mathcal{H}\!HR_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\bar{U}N(\infty)$ - vector of steady-state inputs.

$D(\boldsymbol{x},t)$ - continuous distributed disturbance of the NHLDS block.

$W(\boldsymbol{x},\infty)$ - overall distributed reference in the steady-state.

$\breve{\bar{W}}_i(\boldsymbol{x}_i,\infty)$ - distributed reference vector in the steady-state.

$Y(\boldsymbol{x},t)$ - manipulated variable on the HLDS level.

$YN(\boldsymbol{x},\infty)$ - distributed manipulated variable in the steady-state.

$YN(\boldsymbol{x},t)$ - distributed manipulated variable of the NHLDS block.

$\{\breve{Y}_i(\boldsymbol{x}_i,k)\}_i$ - components of partial manipulated variables taken at locations $\{\boldsymbol{x}_i\}_i$ on the HLDS level.

$\{\hat{Y}(\boldsymbol{x}_i,k)\}_i$ - components of manipulated variable on the HLDS level.

$\{YR_i(\boldsymbol{x}_j,k)\}_{i,j}$ - components of the reduced partial manipulated variables on the HLDS level at locations $\{\boldsymbol{x}_j\}_j$.

Vector $\{\breve{Y}_i(\boldsymbol{x}_i,k)\}_i$ enters the algebraic member 1 and the control process will proceed according to the discussion in Sect. *7.2* (Sect. *5.2*), yielding a sequence of outputs $\bar{U}(k)$.

7.4 Development Systems for the Design of Control Loops

Before the process of control system implementation, it is necessary to set up a controller development system consisting of the blocks of the DPS Blockset, (Fig. 7.9). This development system then aids modelling, dynamics analysis and the optimization of the control system. The block MtoV represents a co-simulation block, meaning the cooperation between Matlab & Simulink and a software product used to model the

dynamics of the controlled system. This enables us to bring the specific characteristics and behavior of the controlled plant into the design procedure of the controller itself.

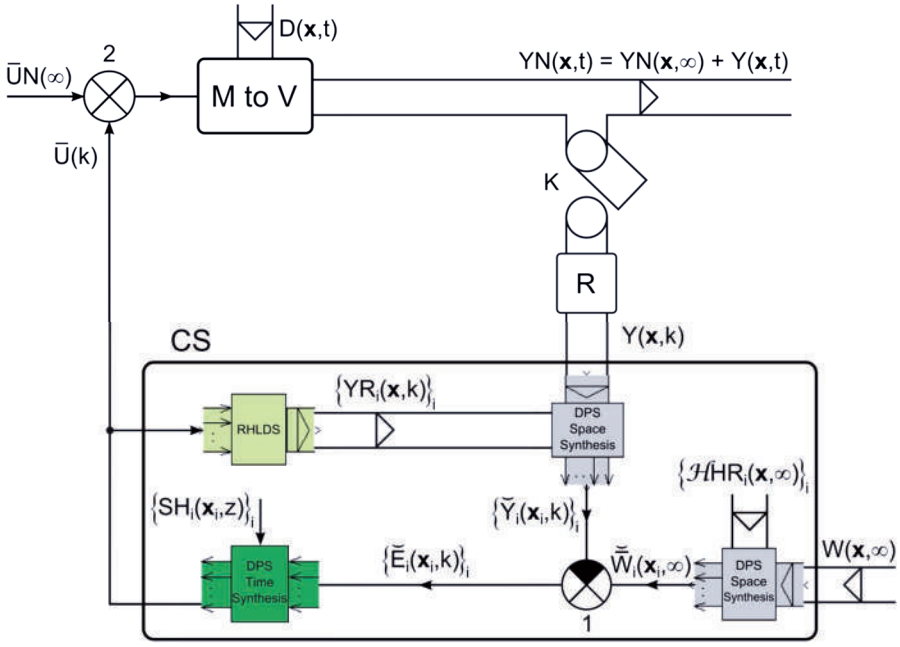


Fig. 7.9 Development system for the design and optimization of control system - CS.

MtoV - cosimulation scheme connecting Matlab & Simulink and the chosen numerical software environment.

CS - control system.

RHLDS - generator of reduced characteristics.

DPS Space Synthesis - spatial control synthesis block.

DPS Time Synthesis - temporal control synthesis block.

R - block of differences.

K - sampling.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{\mathcal{H}HR_i\left(\boldsymbol{x},\infty\right)\right\}_i$ - spatial components of the dynamics of the controlled system.

$\left\{SH_i\left(\boldsymbol{x}_i,z\right)\right\}_i$ - temporal components of the dynamics of the controlled system.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$\bar{U}N(\infty)$ - vector of steady-state inputs.

$D(\boldsymbol{x},t)$ - continuous distributed disturbance.

$W(\boldsymbol{x},\infty)$ - overall distributed reference in the steady-state.

$\breve{\bar{W}}_i(\boldsymbol{x}_i,\infty)$ - distributed reference vector in the steady-state.

$YN(\boldsymbol{x},t)$ - distributed manipulated variable of the NHLDS block.

$YN(\boldsymbol{x},\infty)$ - distributed manipulated variable of the NHLDS block in steady-state.

$Y(\boldsymbol{x},t)$ - manipulated variable on the HLDS level.

$Y(\boldsymbol{x},k)$ - discretized manipulated variable on the HLDS level.

$\{\breve{Y}_i(\boldsymbol{x}_i,k)\}_i$ - components of partial manipulated variables taken at locations $\{\boldsymbol{x}\}_i$ on the HLDS level.

$\{YR_i(\boldsymbol{x},k)\}_i$ - components of the reduced partial manipulated variables on the HLDS level.

Generally, only the existence of the unambiguity of the best approximation of the reference $W(\boldsymbol{x},\infty)$ by the manipulated variable $\breve{Y}(\boldsymbol{x},\infty)$ is mentioned, when solving control problems for distributed parameter feedback loops. The reference is of course approximated by the sum of partial distributed outputs

$$\breve{Y}(\boldsymbol{x},\infty)=\sum_1^n \breve{Y}_i(\boldsymbol{x},\infty)=\sum_1^n \breve{Y}_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}\!HR_i(\boldsymbol{x},\infty)=\sum_1^n \breve{U}_i(\infty)\,\mathcal{H}\!H_i(\boldsymbol{x},\infty) \quad (7.12)$$

at times when the reference changes or distributed disturbances are present.

It is necessary to optimize the number, placement and technical parameters of the actuators and generators of distributed inputs (Fig. 2.2, Fig. 4.2) using the presented design system (Fig. 7.9) in order to fulfil the term

$$\left\|W(\boldsymbol{x},\infty)-\breve{Y}(\boldsymbol{x},\infty)\right\|=\left\|\breve{E}(\boldsymbol{x},\infty)\right\|\le\delta \quad (7.13)$$

on the set of changes in the reference or presence of outside disturbances. Meanwhile it is also necessary to include the requirements for the quality of control on the NHLDS level, for the sequence of steady-state operating modes.

One must also clearly define the regions near the steady-state operating modes, where the assumptions on linearization may hold. Similarly, the extent of the linearized dynamics when segmenting otherwise nonlinear dynamics to describe the transition in-between steady-state operating modes must be unambiguous.

A choice for the particular control strategy for the SISO or MIMO control feedback loops that are parts of the temporal control synthesis must also be made before one attempts to use the formerly described method.

As it was previously mentioned, the design methods used to create control feedback loops are also suitable for a role as expert systems, where using the support of software environments these systems aid the decision process of personnel working on complex industrial systems.

8. MODELING AND CONTROL OF THE GRAVITATIONAL CASTING PROCESS

Casting plays an important role in manufacturing, mainly in the automotive industry, where bulky or complex-shaped parts such as engine blocks, cylinder heads and light-alloy wheels are being produced exclusively by this technology. The versatility of this technology is supported by the fact that, regardless of how large or small the series, casting sizes range from the tiniest components of high sensitivity mechanisms up to the enormous parts of the world's biggest and most amazing machines. This technology has been refined for millennia. Many methods have been developed, the procedures are well-standardized for a wide range of casting products. Each foundry focuses on particular materials and production methods, specific customer groups. In the present globalized world accompanied by the adoption of international quality control standards, markets are open practically anywhere. This fact is also facilitated a widespread usage of so-called CAE (*Computer-Aided Engineering*) software products, which changed the classical image of mechanical engineering forever.

8.1 Casting Technology

The product quality of casting depends on many factors, ultimately affecting the entire production process, starting with melting and conditioning of the cast alloy, mold filling, up to the final solidification of the product. This technology has developed a vast expert knowledge-base throughout time and has accumulated procedures to obtain a sound casting that meets all specifications given by the engineers (dimensional tolerances, surface quality, mechanical properties, pressure resistance, etc. (Kong (2001), Kosec (2007), Klobčar (2008), Becker (2009)). A critical phase of this process is casting solidification in the mold, when the metal/alloy undergoes a phase change from liquid to solid state, followed by cooling to room temperature (and eventual solid state phase changes). This issue is common to all casting technologies ranging from gravitational sand mold casting to high pressure die casting (HPDC), tilt casting, centrifugal casting as well as shell mold casting. The ability of a part design and/or material to form a casting is called *castability*. This has to be taken into account, regardless of the particular casting technology used, and depends on the alloy itself as well as heat transfer conditions at the casting–mold

interface. Tracking all phenomena occurring in the casting–mold system is in general very complicated, therefore, here we focus on the gravitational permanent mold casting of aluminium and describe the most important processes only.

During the process of pouring the superheated alloy into the mold cavity, intensive heat transfer occurs at the mold walls. Melt flow in the mold should be as laminar as possible, as turbulences may carry particles from the mold material away and include them into the casting, thereby deteriorating its quality. In metal molds, especially in HPDC, high flow velocities may cause cavitation resulting in reductions of the mold lifetimes. Wrong mold filling (and design) may result in the melt not reaching some sections of the mold or so-called *cold laps*, where two partially solidified and eventually oxidized flow fronts meet. Melt flow facilitates the equalization of temperature inside the casting and the dynamics of this process depends on the flow mode: the turbulent regime is accompanied by intensive convection and mixing, the laminar regime allows heat transfer in directions perpendicular to the main flow velocity only by means of free convection between layers of the melt. As the melt reaches the liquidus temperature T_{L} (actually, some undercooling is required, Maijer (2009)), the first crystallization nuclei form at mold walls, followed by their dendritic growth—solid phase grains form. The solid/liquid ratio at a given temperature depends on the chemical composition of the particular alloy. With decreasing temperature the solid fraction increases, accompanied by significant shrinking of the volume, which is compensated by the molten metal supply from hotter regions of the casting. In case the surrounding material enclosing the semi-solid region cools down to a critical temperature (close to the solidus T_{S}) and melt supply becomes impossible at the given pressure, cavities are formed inside this closed volume, Fig. 7.1, referred to as *shrinkage and porosity defects*.

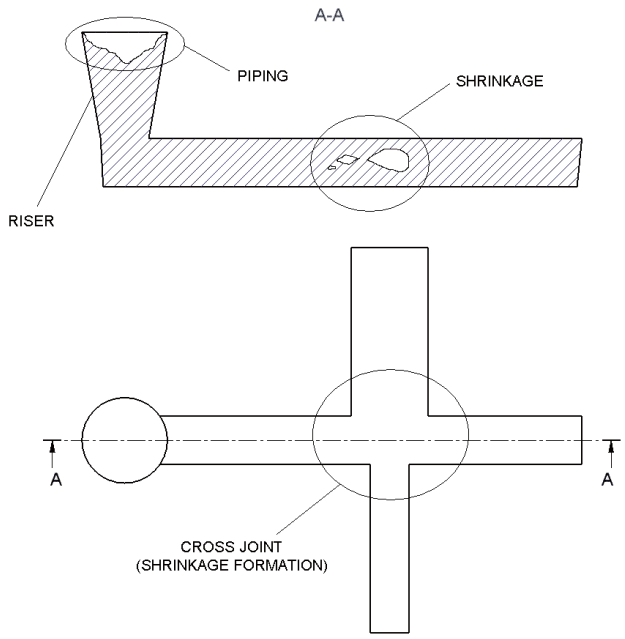


Fig. 8.1 Benchmark casting with a cross joint.

Shrinkage defects are cavities forming in the hot spots of castings: regions where heat removal is slower than in their surroundings e.g. cross-joints. They may also occur at the casting surface as it is common in risers, where piping occurs (this is foreseen as the riser is supposed to supply molten metal to functional parts of the casting, Fig. 8.1). If these are formed by many small voids distributed in some region (more or less uniformly), they are referred to as porosity defects. Hidden voids may cause significant weakening of mechanical strength and shorten life span of the product or result in insufficient pressure resistance and tightness.

Formation and distribution of shrinkage defects is determined by the casting shape and thickness of the mushy zone (Kong (2001)), which again depends on temperature gradient in the melt and thus, although indirectly, on the heat extraction from the casting-mold system. The orientation of the temperature gradient also determines the direction of crystal growth. The dendritic structure of grain growth follows the orientation of the steepest temperature antigradient.

Wherever the temperature gradient in the casting approaches zero $\frac{\partial}{\partial x}T(x,t) \to 0$, the risk of entrapment of not solidified volume is high, (Fig. 8.2). Since heat flux is proportional to the temperature gradient (convection being neglected), given by the heat transfer coefficient λ (W m^{-1} K^{-1}). Carefully controlled temperature difference between the mold and casting may ensure the right speed and direction of solidification

fronts. Directional solidification is needed in order to ensure the homogeneity of the solid and avoid shrinkage and porosity formation, (Fig. 8.2).

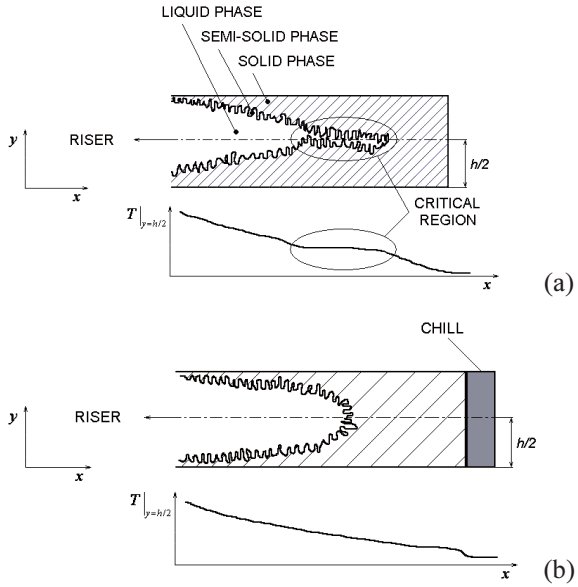


Fig. 8.2 Uncontrolled (a) and directional (b) solidification.

In gravitational casting, the so-called *risers* are used as hot reservoirs. Their purpose is to keep the temperature high in their respective location. Chillers are used as heat sinks where needed, in order to keep a low temperature at the location. These are metal blocks facilitating crystallization and, if cooled by some medium, they keep a constantly low temperature at their location, which ensures a nonzero temperature drop in the solidifying material.

8.1.1 Mathematical Model

The most general mathematical models of casting processes consider two physical phenomena: heat transfer and melt flow.

Heat Transfer

We start with the general heat equation

$$\rho\frac{\partial H}{\partial t}+\rho\nabla H\cdot\boldsymbol{v}-\nabla\cdot\left(\lambda\nabla T\right)-\sum_i U_i=0 \tag{8.1}$$

where H (J kg^{-1}) is specific enthalpy, ρ (kg m^{-3}) density, $\boldsymbol{x}$ (m) coordinates, $\boldsymbol{v}$ (m s^{-1}) melt flow velocity, T (K) temperature, U_i (W m^{-3}) heat sources and sinks. Specific enthalpy is defined as follows

$$H(T) = \int_0^T c\,\mathrm{d}\tau + L(1 - fs) = \int_0^T c\,\mathrm{d}\tau + Lfs^* \tag{8.2}$$

where c (J kg^{-1} K^{-1}) is specific heat, L (J kg^{-1}) latent heat of solidification, $fs = fs(T)$ solid fraction and $fs^*(T) = 1 - fs$(T) liquid fraction, (Yang (2006)).

For simple cases, this can be expressed as a rough approximation by (Xing (2008))

$$fs(T) = 1 - \frac{T - T_\mathrm{S}}{T_\mathrm{L} - T_\mathrm{S}} \tag{8.3}$$

Thermal conductivity also depends on fs

$$\lambda(T) = fs(T)\,\lambda_\mathrm{S} + (1 - fs(T))\,\lambda_\mathrm{L} \tag{8.4}$$

where λ_L is thermal conductivity of the liquid phase and λ_S thermal conductivity of the solid phase. In reality, these dependences are much more complicated, since conductivities of individual phases are not constant and vary with temperature.

Substituting Eq. (8.2) into Eq. (8.1) we get

$$\rho\left(c - L\frac{\partial fs}{\partial T}\right)\left(\frac{\partial T}{\partial t} + \nabla T \cdot \boldsymbol{v}\right) - \nabla \cdot (\lambda \nabla T) = \sum_i U_i \tag{8.5}$$

In case of a linear system, we obtain the well-known *Fourier-Kirchhoff* heat equation, omitting phase change

$$\frac{\partial T}{\partial t} + \nabla T \cdot \boldsymbol{v} = \frac{\lambda}{\rho c}\nabla^2 T + \sum_i U_i \tag{8.6}$$

This model, however, is applicable to the mold preheating process only, where one can neglect system parameter changes, thanks to the relatively narrow temperature range where the process takes place. On the other hand, at modelling solidification processes where the temperature range is considerably larger (20 – 700 °C), changing material parameters have to be considered along with latent heat release due to the occurring phase change.

Finally, the governing equation becomes

$$\rho(T)\left(c(T)-L\frac{\partial fs(T)}{\partial T}\right)\left(\frac{\partial T(\boldsymbol{x},t)}{\partial t}+\nabla T(\boldsymbol{x},t)\cdot\boldsymbol{v}(\boldsymbol{x},t)\right)- \\ -\nabla\cdot\left(\lambda(t)\nabla T(\boldsymbol{x},t)\right)=\sum_i U_i(\boldsymbol{x},t) \tag{8.7}$$

Latent heat may be included in the *effective heat capacity*

$$c_{\text{eff}}=c-L\frac{\partial fs}{\partial T}=c+L\frac{\partial fs^*}{\partial T} \tag{8.8}$$

which simplifies the equation to the form

$$\rho c_{\text{eff}}\left(\frac{\partial T}{\partial t}+\nabla T\cdot\boldsymbol{v}\right)-\nabla\cdot\left(\lambda\nabla T\right)=\sum_i U_i \tag{8.9}$$

In some cases, it is advantageous to consider latent heat release as a heat source $U_i(\boldsymbol{x},t)$ and include it that way. How to deal with latent heat depends on the particular problem and is chosen in order to make the computation as simple as possible.

In general, four basic types are considered:

Dirichlet boundary condition as prescribed temperature field

$$T(\boldsymbol{x},t)=T_0(\boldsymbol{x},t)\quad \boldsymbol{x}\in\Gamma_i \tag{8.10}$$

Neumann boundary condition as prescribed heat flux

$$-\lambda\nabla T(\boldsymbol{x},t)\cdot\boldsymbol{n}=-\lambda\frac{\partial T(\boldsymbol{x},t)}{\partial n}=q_{\text{N}}(\boldsymbol{x},t),\quad \boldsymbol{x}\in\Gamma_i \tag{8.11}$$

where $\boldsymbol{n}$ is the unit normal vector of the boundary surface and $q_{\text{N}}(\boldsymbol{x},t)$ the given heat flux through the surface Γ_i.

Then, the ***Fourier boundary condition*** is convection at the boundary

$$-\lambda\frac{\partial T(\boldsymbol{x},t)}{\partial n}=q_{\text{F}}(\boldsymbol{x},t)=h\left(T(\boldsymbol{x},t)-T_{\text{ext}}(\boldsymbol{x},t)\right),\quad \boldsymbol{x}\in\Gamma_i \tag{8.12}$$

where h (W m^{-2} K^{-1}) is the convection heat transfer coefficient, T_{ext} temperature of the cooling/heating medium, and ***radiation from the surface***

$$-\lambda\frac{\partial T(\boldsymbol{x},t)}{\partial n}=q_{\text{R}}(\boldsymbol{x},t)=\sigma\varepsilon\left(T^4(\boldsymbol{x},t)-T_{\text{ext}}^4(\boldsymbol{x},t)\right),\quad \boldsymbol{x}\in\Gamma_i \tag{8.13}$$

where σ is the Stefan-Boltzmann constant, T_{ext} ambient temperature and ε is the emissivity of the radiating surface. A more complex model, considering geometrical constraints between radiating surfaces, can be found e.g. in Nedjar (2002). One of the possible simplifications is including the radiation part into the convection as it has been proposed by Nejad (2010)

$$-\lambda\frac{\partial T(\boldsymbol{x},t)}{\partial n}=q_{\text{F}}(\boldsymbol{x},t)=\left(h_{\text{Conv}}+h_{\text{Rad}}\right)\left(T(\boldsymbol{x},t)-T_{\text{ext}}(\boldsymbol{x},t)\right),\quad \boldsymbol{x}\in\Gamma_i \tag{8.14}$$

while $h_{\text{Rad}}=\sigma\varepsilon\left(T^2+T_{\text{ext}}^2\right)\left(T+T_{\text{ext}}\right)$.

Melt Flow

The basic viscous fluid flow model is described by the *Navier-Stokes* equation and the equation of continuity

$$\frac{\partial(\rho\boldsymbol{v})}{\partial t}+\boldsymbol{v}\cdot\nabla(\rho\boldsymbol{v})=\boldsymbol{F}+\nabla\cdot\boldsymbol{s} \tag{8.15}$$

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\boldsymbol{v})=0 \tag{8.16}$$

where $\boldsymbol{F}$ (N m^{-3}) is the volume force, $\boldsymbol{s}$ (N m^{-2}) stress tensor, $\boldsymbol{v}$ (m.s^{-1}) flow velocity.
For a Newtonian incompressible fluid, the stress tensor becomes

$$\boldsymbol{s}=-p\boldsymbol{I}+2\mu\boldsymbol{d} \tag{8.17}$$

where p (Pa) is the pressure, $\boldsymbol{I}$ the identity tensor, μ (Pa s) dynamic viscosity and $\boldsymbol{d}$ is the symmetric velocity gradient tensor defined as follows

$$\boldsymbol{d}=\frac{1}{2}\left(\nabla\boldsymbol{v}+(\nabla\boldsymbol{v})^{\text{T}}\right) \tag{8.18}$$

Substituting Eq. (7.18) to Eq. (7.17), then to Eq. (7.15) and after some algebraic manipulation we have

$$\frac{\partial(\rho \boldsymbol{v})}{\partial t} + \boldsymbol{v} \cdot \nabla(\rho \boldsymbol{v}) = \boldsymbol{F} - \nabla p + \mu \nabla^2 \boldsymbol{v} \tag{8.19}$$

The boundary conditions are defined as a prescribed velocity field

$$\boldsymbol{v}(\boldsymbol{x},t) = \boldsymbol{v}_0(\boldsymbol{x},t) \quad \boldsymbol{x} \in \Gamma_i \tag{8.20}$$

and prescribed pressure field

$$p(\boldsymbol{x},t) = p_0(\boldsymbol{x},t), \quad \boldsymbol{x} \in \Gamma_i \tag{8.21}$$

The overall mathematical model of this process comprises of the Eqs. (8.5), (8.19) and (8.16), with the respective boundary conditions. For the sake of simplicity, here we only mentioned the fundamental principles. For a more in-depth study of the phenomena occurring during casting solidification, see e.g. Zhang (2007).

8.1.2 Casting Control Problem

Casting control, or also *crystallization control*, is a complex problem and can be handled from various points of view. Basically, two processes are to be taken care of:

Mold Preheating Control

The temperature field of the mold before the molten metal is cast in has a significant effect on the heat extraction and, consequently, the cast product quality as well as process cycle times. It is therefore desired to have the mold thermal state under control. Of course, this requires mold temperature field measurements in time and space.

Casting Solidification Control

In this case, the problem of feedback feasibility will arise. Measuring temperatures in the casting bulk by penetrating thermocouples is not acceptable, except for laboratory and testing purposes. However, in recent years, so-called *software sensors* attract increasing attention, which use a mathematical model of the running process and are based on a few industrially acceptable measurements. Software sensors

estimate the required quantities and these are then used by the control system instead of physically measured ones.

From the large number of works dealing with the modelling and control of batch casting processes, let us mention e.g. Aubry (2005), where an artificial neural network (ANN) is used for mold temperature field evaluation, which is sensed by a thermovision camera. The ANN judges the actual temperature field as good or bad, based on expert knowledge collected empirically during long term operation. Based on the ANN output, the required action is taken—heating or cooling of specific locations. The authors of the paper Metan (2009) are using thermovision combined with sub-surface thermocouple measurements, taken from the material in selected locations of the mold cavity. The feedback loop controls mold opening time and cooling water flow rate by a fuzzy controller, with the main concern being cycle time reduction and thus production efficiency, while still preserving an acceptable casting quality.

Authors of the work Zaidat (2007) investigated thermal processes in the low pressure die casting of aluminium parts, where a virtual 3D FEM model of the process is used, which was obtained by the CAE software ABAQUS and validated during long term operation. The model made virtual experiments possible without any risks to production. Maximal temperatures in three locations of the casting–mold system during steady-state casting cycles were controlled by a model predictive controller, based on a reduced MIMO model. The control objective was to compensate for disturbances in operation, like variable mold opening time and melt temperature. Thereby, a reduction in porosity was achieved. Nominal set-points were determined by operators, based on expert knowledge. The classical approach to the control of thermal processes in the casting–mold system is based on empirical knowledge, which is also applied in Li (2007), where feedback control of the temperature in the vicinity of chiller cooling channels is applied by means of a bang–bang controller. Such an approach is justified by the fact that real world processes often require keeping temperatures in some bounds rather than prescribing precise values, especially considering that these cycling processes exhibit large temperature fluctuations.

Part of the scientific community focuses on the control of solidification and crystallization down to the microstructure level by means of electromagnetic (EM) fields. For example, the authors of Ganapathysubramanian (2005) achieved a finer grain size and more homogeneous structure of Al-Si alloys, by putting it to a strong magnetic field, while the classical approach would count on various additives and faster cooling. In Liu (2006), the authors enforce directional solidification of an Al-Ni alloy by applying a moving EM field, which results in forced convection inside the melt and mushy zone during solidification, finally leading to a fine grain structure. This phenomenon is also being studied by the authors of Eskin (2006). Authors of Klobčar (2008) manipulate the magnetic field gradient, in order to reach a fine

structure and directional solidification. Finding the optimal gradient field is an optimization process, which is formulated in the paper and methods for its solution are proposed as well. Another approach is applied in the research presented in the paper Stefanescu (2008), where the authors investigated the effect of vibrations on acetone (C_3H_6O) and phenyl salicylate ($C_{13}H_{10}O_3$) solidification. Ultrasound vibrations also affect the flow inside molten metals and contribute to grain refinement. On the other hand, not only the casting quality is of concern. Several papers are dealing with fatigue life of casting molds. Authors of Schewchuk (1994) conclude that the temperature gradient is the main cause for tears and cracks in the mold material. Therefore, the temperature field is to be as uniform as possible, which limits maximum reachable temperatures and imposes constraints on actuator design, their number as well as power (regardless of whether cooling channels or heaters are needed).

In this chapter, we focus on the most fundamental parameter of a sound casting, thus the elimination of shrinkage and porosity defects, for which technological requirements are easily formulated. We approach these by classical means, i.e. heating elements for mold preheating and cooling channels for casting solidification control, which do not require the implementation of any new technologies and is feasible just by modifying existing systems.

8.2 Benchmark Experimental Casting Plant

Shrinkage and porosity are the most common defects that plague the gravitational casting processes. An experimental casting plant was built at the Institute of Automation, Measurement and Applied Informatics, FME STU in cooperation with the Institute of Technologies and Materials, FME STU; for the better understanding of these phenomena and for control system development, tuning and testing (see Fig. 8.3). It is equipped with instrumentation and power electronics for generating and tracking temperature fields of the mold as well as the casting. The human machine interface is set up in Matlab & Simulink, where real-time data exchange with the process is available.

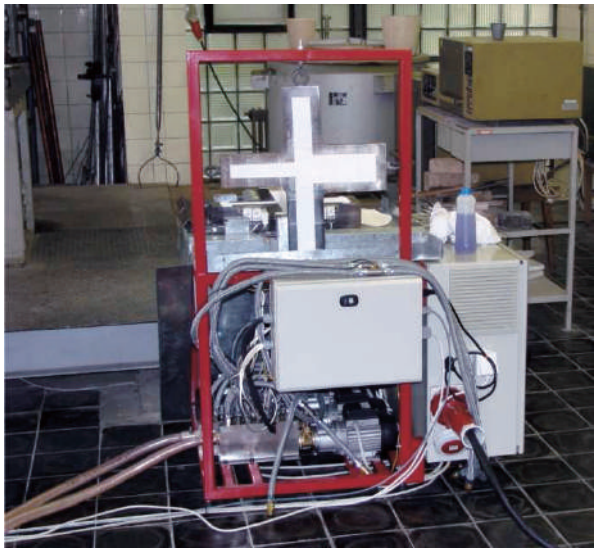


Fig. 8.3 Benchmark experimental casting plant.

It models a cross-joint, a very common, yet problematic feature due to the presence of a hot spot, which, if untreated, leads to shrinkage and porosity. The benchmark casting is a cross-shaped block with arms of different cross-sections, resulting in different solidification times, which poses another problem to be taken care of (Fig. 8.4). Finally, the riser is placed far away from the hot spot, which means a problematic feeding of molten metal to the cross joint.

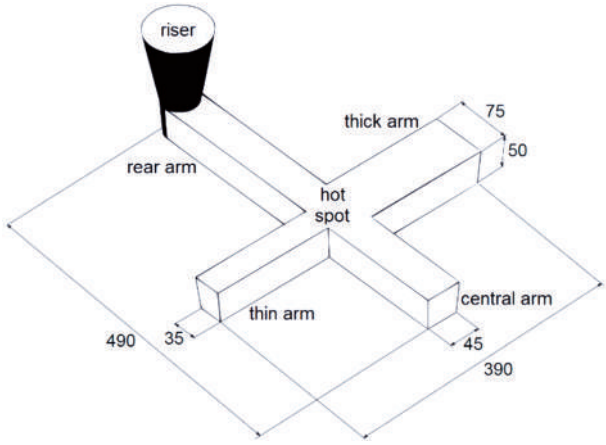


Fig. 8.4 Benchmark casting shape and dimensions.

The final size of the casting respects the capabilities of the foundry located at the laboratory and running costs for performing experiments. The casting shape determines all the remaining items of the experimental plant, whose “heart” is the steel mold with built in sensors and actuators.

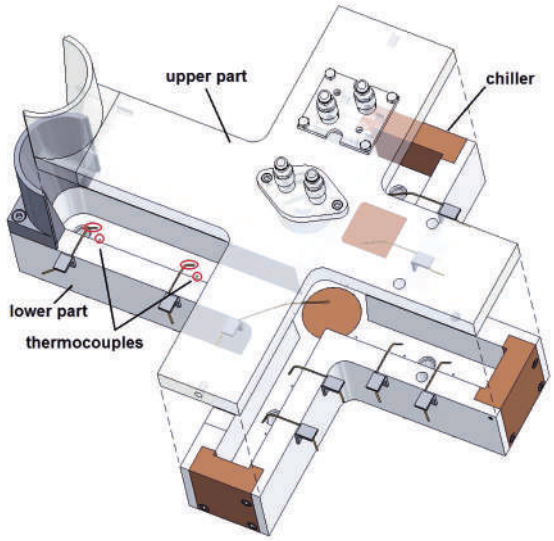


Fig. 8.5 The mold with built in sensors and actuators.

8.2.1 Plant Technology

Preheating System

The mold is equipped with 26 built-in electric resistance heating elements, each of 400 W nominal power, divided to 5 zones, Fig. 8.5 and Fig. 8.6:

- ZONE 1 Thin arm (35 mm), 1600 W
- ZONE 2 Cross joint – hot spot, 2400 W
- ZONE 3 Thick arm (75 mm), 1600 W
- ZONE 4 Center arm (45 mm), 1600 W
- ZONE 5 Rear arm (from the hot spot to the riser), 3200 W

Total installed power is 10.4 kW. Mold temperature is measured by thermocouple probes TE-P01 to TE-P11. The number and power of heaters was designed to make reasonable times (on a 10 minute scale) for mold temperature changes possible, especially keeping the large thermal inertia of the mold in mind.

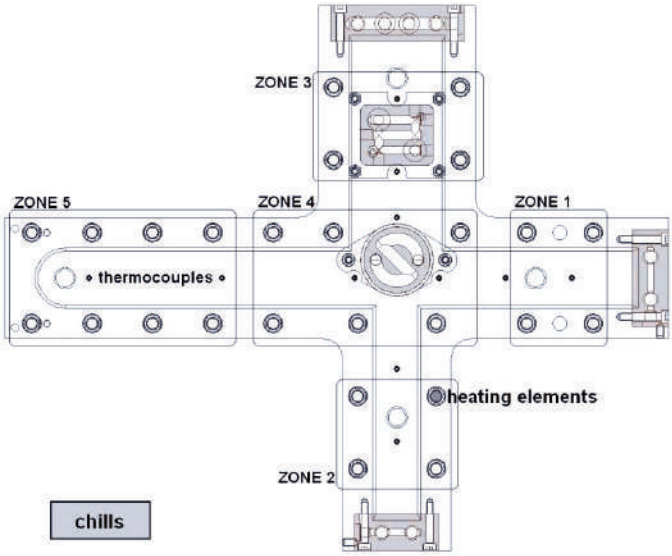


Fig. 8.6 Heating system layout.

Remark: Industrial devices usually use uniform temperature heating and/or cooling of steel molds, which is kept more or less constant. This is done in order to increase the lifetime of the mold and to improve casting quality, while providing a predictable heat removal from the system. *Therefore, this process is referred to as mold tempering.* Many possible configurations exist, particular mold tempering system designs depend on the complexity of the casting shape, where some parts need more intensive heat removal than others, which are sufficiently cooled by the mold walls without special treatment. All these are taken into account by the mold design engineer. In general, however, a single circuit is used as far as possible. Electrical heating is rather uncommon in casting technology, due to the high thermal loads the heaters are subject to, moreover to the fact that a liquid cooling system can be used for heating and cooling as well, thus being well off with one circuit used for both purposes.

Cooling System

The mold has seven built-in copper chillers serving five zones depicted in Fig. 8.6 and Fig. 8.7:

- CH1 Thin arm
- CH2 Center arm
- CH3 Thick arm
- CH4 Comprises the chillers CH4A and CH4B, provides additional cooling for the thick arm from below and above.
- CH5 Comprises the chillers CH5A and CH5B, cools the hot spot from below as well as above.

These are water-cooled (eventually substituted by a water-glycol solution. The cooling medium is pumped by pumps PA1 to PA5 from the tank TA-1 via the flow divider R-1, where the inlet temperature is measured by the thermoelement TE-S8. The fluid outlet temperature is measured separately in individual channels by thermoelements TE-S1 to TE-S7. All outlets are collected by the collector Z-1, where a check valve in each channel maintains system pressure in order to avoid cooler boiling inside the chillers due to low flow rates. Afterwards, the fluid flows to the collecting tank TA-2. Originally, an alternative with a closed circuit and heat exchanger was proposed, although not built due to higher prototyping costs. The overall pressure drop of individual circuits is in the range of 2–4 bar, caused by thin cooling channel cross-sections, hosing quick-couplings and the pressure keeping valves. Therefore, positive displacement pumps (vane pumps) are used, which were available in the 14 bar range. These are each protected by a built-in pressure relief valve. Temperature at the casting-mold interface is measured by thermocouple probes TE-P01 to TE-P11 and in the casting bulk thermocouple wires TE-W1 to TE-W9 are used.

Chiller layout is based on the recommendation of foundrymen and their experience and reflects the need of intensive heat removal from the thick arm as well as the hot spot. The number of measurements along the mold cavity and the casting bulk is kept minimal, while still being able to track the temperature field and gradients as well. The minimal number of measurements along the line of ensuring the tracking of gradient decrease due to latent heat release is three.

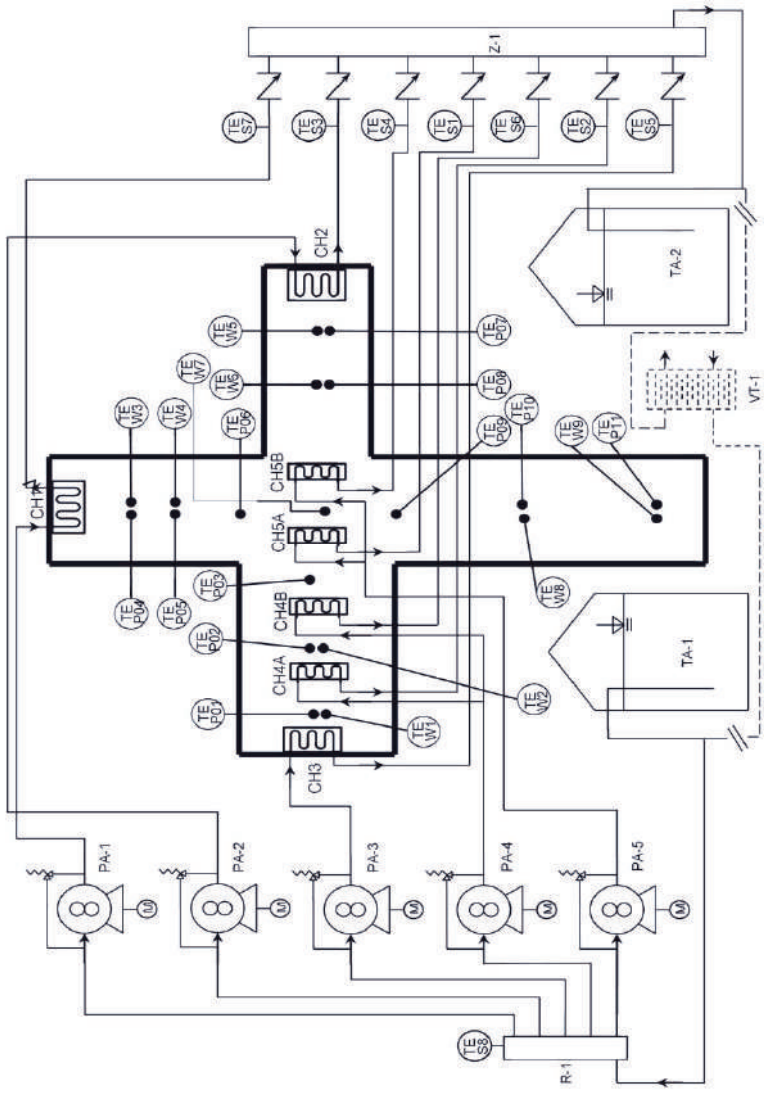


Fig. 8.7 Process diagram of the cooling system.

8.2.2 Measurement and Control Loops

Mold Temperature Measurement

Thermocouples TE-P01 to TE-P10, are of type K (NiCr-NiAl), manufactured by OMEGA, model TJ36. The signal is transformed and amplified via AUDON TCK-4 amplifiers, which transmit a voltage signal in the range 0 – 10 V to the measuring board terminals.

Cooling Medium Temperature Measurement

Thermoelements TE-S1 to TE-S8 are Pt1000 type, manufactured by SENSIT. They are connected to the RU-4PA-DC amplifier/converter of the same manufacturer, transmitting a 0-10V signal to the measuring board.

Casting Temperature Measurement

Thermocouples TE-W1 to TE-W9 are replaced by new ones in each experiment and are made of K-type sheathed thermocouple wire by OMEGA. The whole length, including the connectors, is made of the same wire materials to avoid disturbances and offsets and are wired to AUDON TCK-4 amplifiers which are transmitting the voltage signal to the measuring board. Each of these sensors is wired via an ADVANTECH PCLD-8710 terminal board to the ADVANTECH PCI-1720 measuring card.

Heating Elements

The BACKER ELEKTRO 2303/360 heating elements having the nominal power 400W are powered by Honeywell DC-3000S 230V thyristor power supplies controlled by a 0-10V voltage signal coming from the HUMUSOFT AD622 measuring board via the HUMUSOFT TB620 distribution board.

Cooling Pumps

The cooling medium is circulated by SPECK PUMPEN DS-450-MK.0042 type vane pumps, driven by asynchronous electromotors. These are fed by SEW EURODRIVE MOVITRAC 3 phase frequency converters, controlled by a 0-10V voltage signal from the HUMUSOFT AD622 and ADVANTECH PCI-1720 measuring boards via their respective distribution boards.

Software Interface

Human-Machine-Interface is realized via a Matlab & Simulink model, using the Simulink Desktop Real-Time framework, ensuring the real time functionality of the system, (Fig. 8.8).

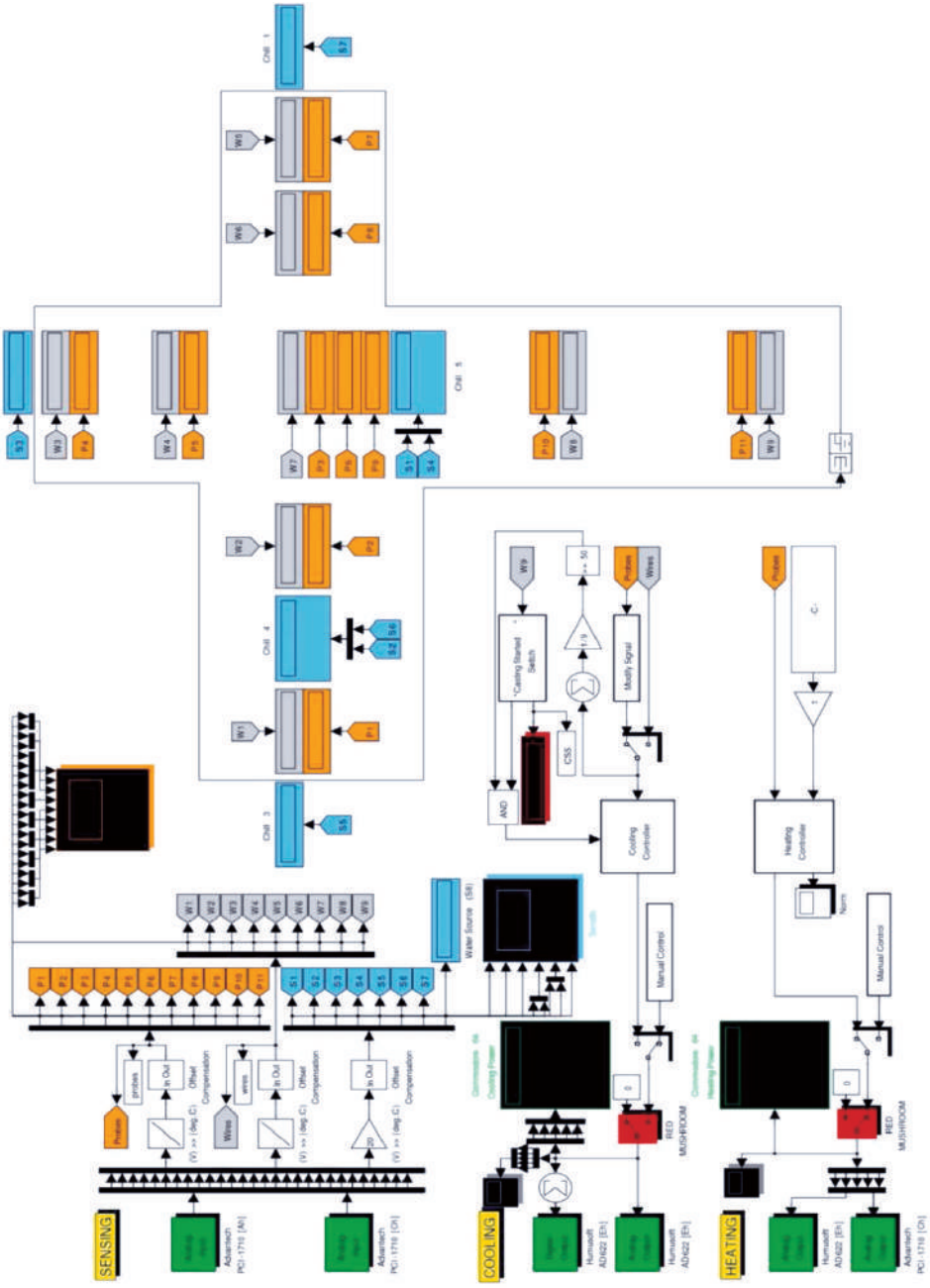


Fig. 8.8 Matlab & Simulink model of the control scheme.

8.2.3 Construction

The main parts of the mold, lower part, mold cover and two parts of the riser section are made of STN 11523 steel, which is not a usual material for aluminium casting mold production, nevertheless, it suits the requirements of our experiments and its low cost, high availability and good machinability were the main reasons of this choice. Fig. 8.9-8.13.

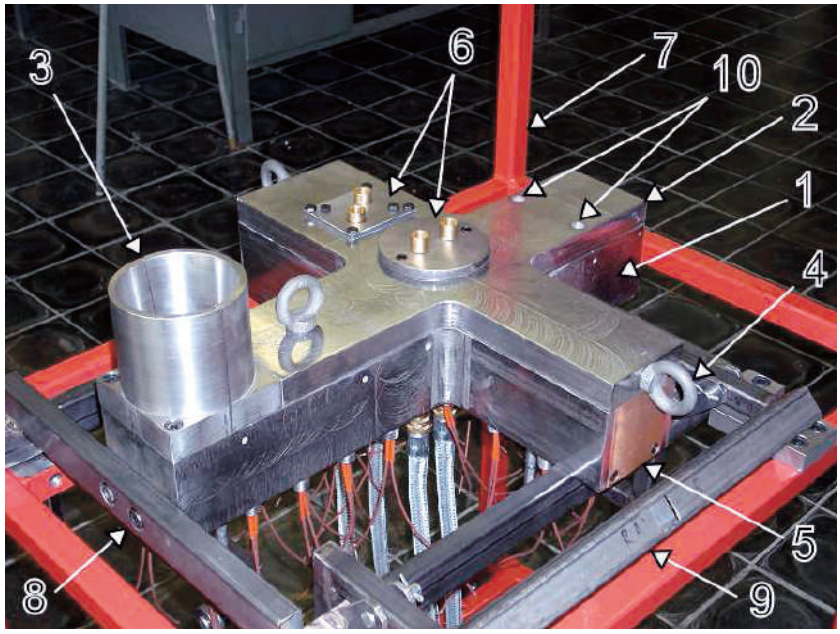


Fig. 8.9 Casting mold assembly: 1 – lower part with the mold cavity, 2 – upper part (cover), 3 – riser section, 4 – lifting eye, 5 – chiller CH2, 6 – quick couplings for the upper chillers CH4A and CH5A, 7 – main frame, 8 – auxiliary frame, 9 – ejector mechanism, 10 – conical pins.

The largest part of the mold assembly is the lower part with the mold cavity in shape of the benchmark casting. Cavity design follows general casting rules, including 2° tapering of the walls. The dividing plane is designed in a way that the whole cavity fits to one part of the mold; while the riser section consists of two separate parts, one bolted to the lower part, the other to the mold cover. This saved considerable machining costs. Around the cavity 26 blind Ø10 mm holes are drilled for electrical heater inserts, four through Ø15 mm holes for the ejectors in each of the arms and along the cross axes 11 through Ø1.5 mm holes to fit thermocouple probes. In each arm of the cross there is an ejector, which simplifies casting removal.

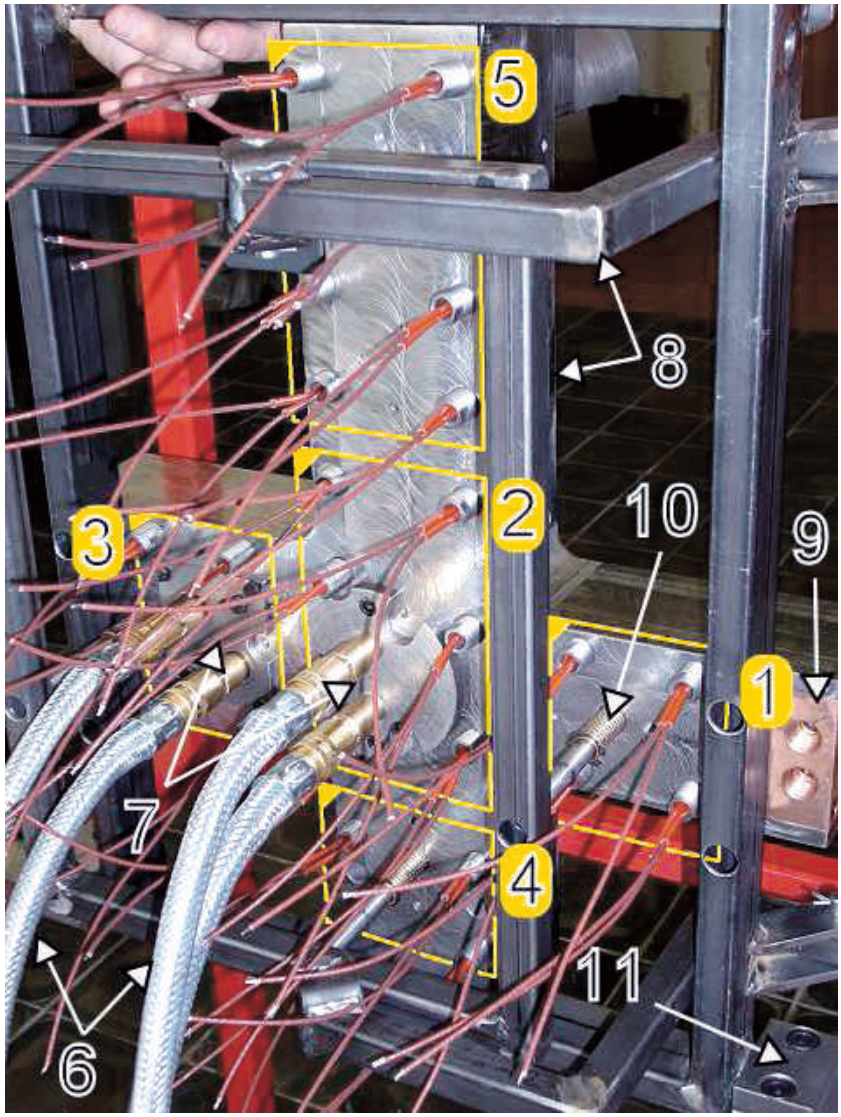


Fig. 8.10 The auxiliary frame tilted to vertical position: 1 – heating zone „1“, 2 – heating zone „2“, 3 – heating zone „3“, 4 – heating zone „4“, 5 – heating zone „5“, 6 – connecting hoses for cooling, 7 – flanges of chillers CH4A and CH5A, 8 – ejector mechanism, 9 – chiller CH1, 10 – ejector, 11 – pivot pin journal.

At the dividing plane, there are grooves for thermocouple wires and vents at the thin and thick arm which facilitate the venting of gases from the mold cavity during filling. Copper chillers are inserted into prepared openings at the end of the three arms. Auxiliary profiles for fastening ceramic connectors of thermocouple wiring are installed on the sides.

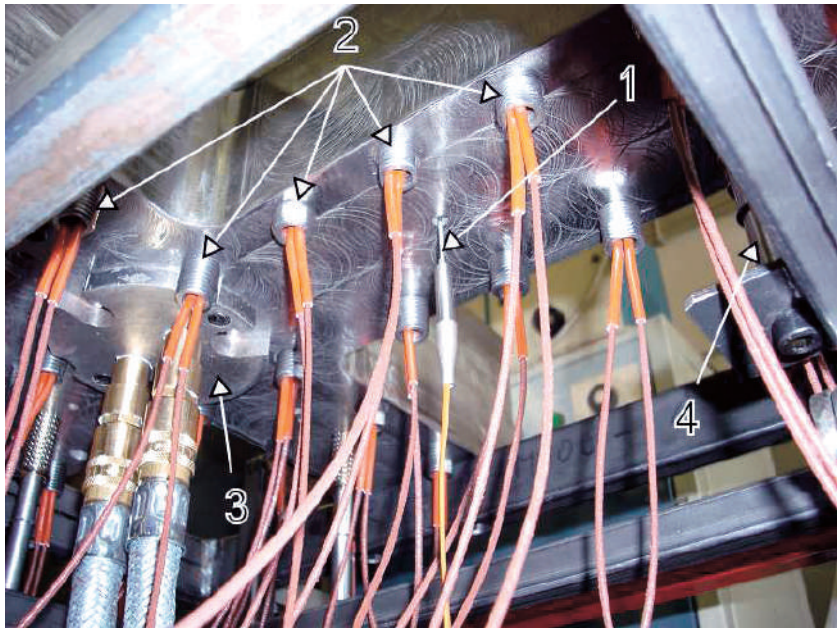


Fig. 8.11 Bottom view: 1 – thermocouple probe (TE-P05), 2 – electrical resistance heaters, 3 – flange of chiller CH5A, 4 – ejector.

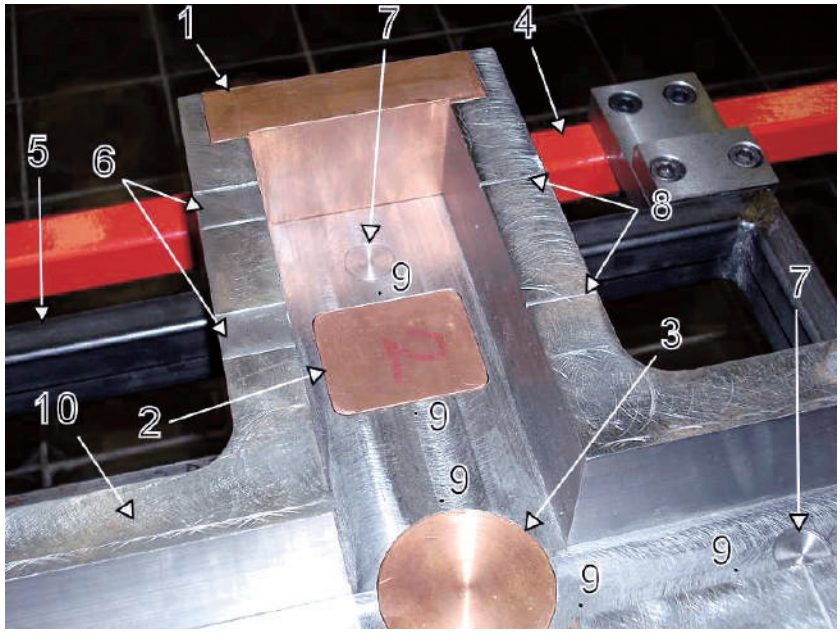


Fig. 8.12 Detail of the mold cavity: 1 – chiller CH3A, 2 – CH4A, 3 – CH5A, 4 – main frame, 5 – mold carrying frame, 6 – vents, 7 – ejector, 8 – grooves for thermocouple wire, 9 – thermocouple probes, 10 – mold dividing plane.

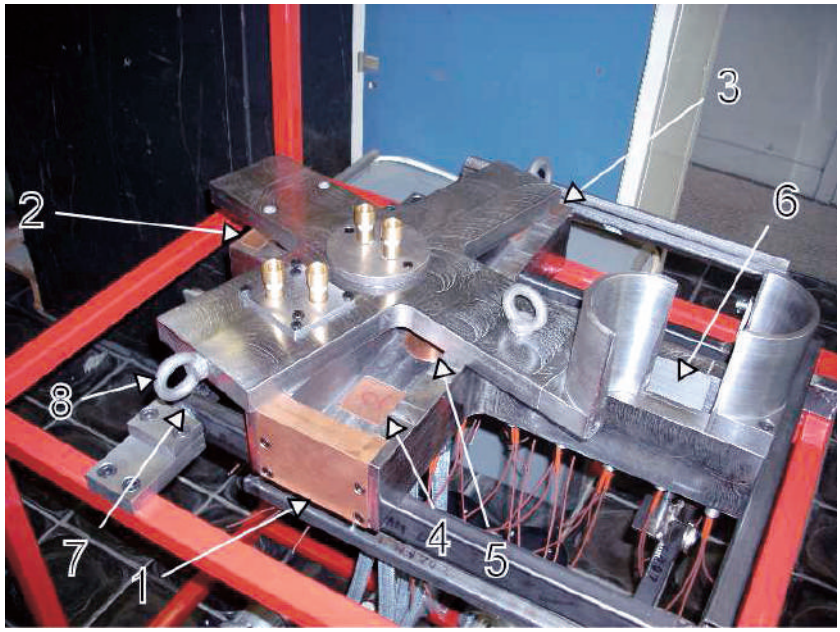


Fig. 8.13 View of the open mold: 1 – chiller CH3, 2 – CH2, 3 – CH1, 4 – CH4A, 5 – CH5A, 6 – mold cavity, 7 – pivot pin journal, 8 – lifting eye.

Chillers are made of copper blocks with drilled cooling channels. The chillers for zone 4, CH4A/B, are made of a copper part with the cooling channel milled in and the supporting steel flange, fused together by soldering. Chillers CH5A/B are also fastened to the mold by flanges and bolts. Components of the coolant distribution system, brass fittings, quick couplings and VITON® hoses, as well as the ejector bearings are from HASCO.

The upper part, mold cover, consists of two parts: a cross shape plate with built in chillers CH4B and CH5B and one half of the riser section. It is equipped with lifting eyes for manipulation and is kept in place by two tapered pins. The interface between the upper and lower parts is the casting dividing plane.

The mold assembly is fastened to an auxiliary frame with a manually operated integrated ejector mechanism connected to the ejectors in the mold. This is then hinged on a pivot, in order to be able to tilt the mold to facilitate gas release as well as for the maintenance of the installed instrumentation and electrical equipment.

8.3 Control of the Casting Process

8.3.1 Control System

The control problem formulation and its solution are based on the approach described in the previous chapters. We split the control problem into two independent parts: mold preheating and casting solidification control. Fig. 8.14. The preheating process is rather slow (unless we risk dangerous local overheating), it lasts tens of minutes. Casting solidification on the other hand is quite fast compared to the former, the whole process takes about 2 minutes. Simultaneous heating and cooling brings no improvement as the dynamics of these processes is significantly different, it only increases the power consumption. The control system is designed in order to reach the desired temperature field before casting and, after the metal is cast, to support the directional solidification of casting by means of the chillers. During mold preheating control, the control system evaluates the mold temperature field based on the data from the built-in thermocouple probes and adjusts heater powers accordingly via thyristor controlled power supplies. During casting solidification, it evaluates temperatures as well as gradients along the casting axis and controls coolant flow rates through chillers by means of frequency converters supplying pump motors. No flow rate measurement is taken since positive displacement pumps are used and it can be calculated from the drive frequency. We performed the control process in simulation as well as experimentally.

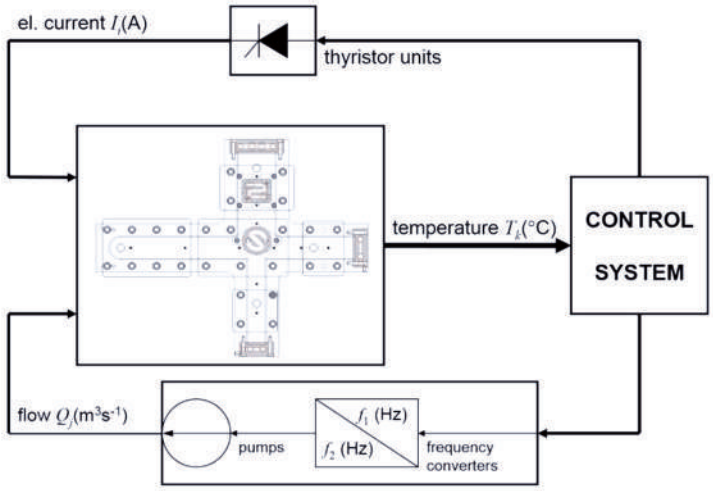


Fig. 8.14 Mold control system.

8.3.2 Numerical Model and Experimental Validation

We created a simulation model of these processes for the purpose of dynamical analysis and obtaining dynamical characteristics of the mold. This was done in order to be able to perform virtual experiments of mold preheating, as well as casting solidification. The software package ProCAST was utilized, which is specialized in casting. Computational methods and algorithms implemented there are based on principles and methods described in this book. Still, the main problem when developing a numerical model is the need of the knowledge of numerous unknown parameters: boundary conditions, material properties and others that define the system behavior. We used the ProCAST material database for the initial setup, which is experimentally validated. Boundary conditions were also defined based on experience and expert recommendations, which were adjusted according to the experimental data acquired later.

Mold Geometry, FEM Mesh Generation

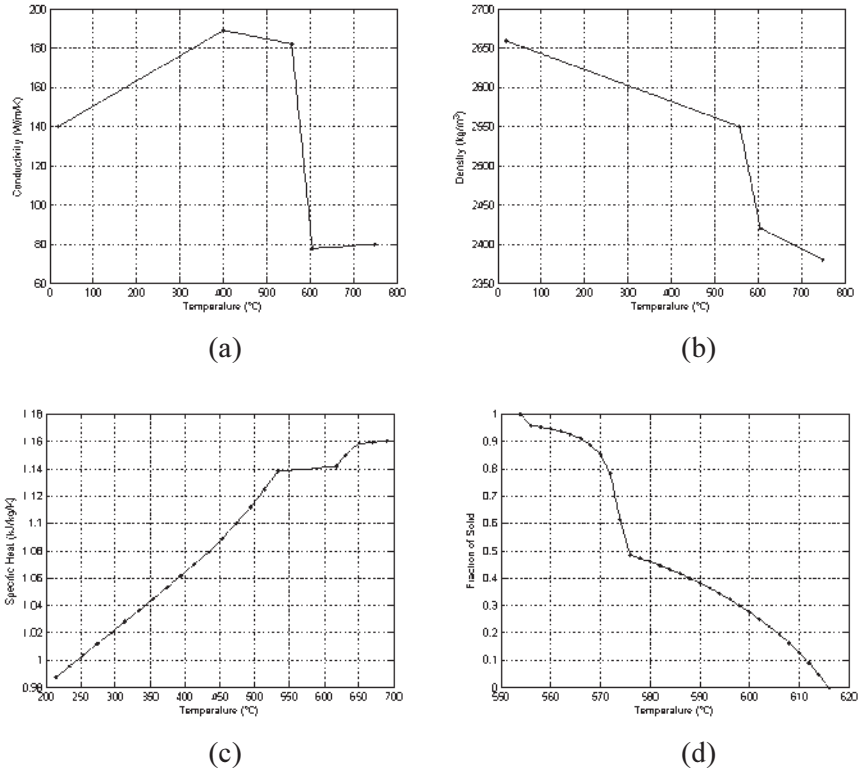
After the geometry is modelled by the CAD software, a computational mesh is generated by FEM software. When we imported the mold geometry into the ProCAST module called *MeshCAST*, we deleted the thermocouple probe holes, whose small dimensions either require immense local densening of the mesh or would most probably cause negative-Jacobian elements, see Section *4.2.2*. Furthermore, we neglected various details like chamfering, mounting holes etc., which are not significant in the modelled process, however, considering them in detail would require mesh densening that is undesirable as causes an unnecessary increase in computing power and memory requirements. The MeshCAST module generates FEM meshes automatically. The software lets the user adjust the mesh manually, especially when local densening is desirable, as it is in our case with the mold cavity surface. Alternatively this is needed when fast changing quantities are to be tracked, which also occurs in our case, where catching the moment when solidification starts is essential. The resulting model comprises 36512 nodes and 152581 elements.

Materials

The lower and upper parts of the mold as well as the riser section and chiller flanges are modelled as low-carbon steel taken from the ProCAST database, whose composition matches the specifications of STN 11523. Chillers were assigned the properties of pure copper and the casting is of AlSi7 alloy, which was used in the experiments.

Material property:	**Steel**	**Copper**
Thermal conductivity: λ (W m^{-1} K^{-1})	60.0	400.0
Density:ρ (kg m^{-3})	7809.0	8360.5
Specific heat capacity: c_p (J kg^{-1} K^{-1})	500.0	400.0

Table 8.1 Physical properties of casting mold construction materials.



(a) (b)

(c) (d)

Fig. 8.15 Temperature dependent material properties of AlSi7 alloy (a) thermal conductivity, (b) density, (c) specific heat capacity, (d) fraction of solid.

Interfaces

Conditions at the interface of two parts were modelled by assigning the heat transfer coefficient as follows:

- Bolted parts $h = 1000\ \mathrm{W\ m^{-2}\ K^{-1}}$
- Soldered parts $h = 5000\ \mathrm{W\ m^{-2}\ K^{-1}}$
- Free contact (no fastening, only gravity) $h = 500\ \mathrm{W\ m^{-2}\ K^{-1}}$
- Casting – mold interface $h = 2000\ \mathrm{W\ m^{-2}\ K^{-1}}$
- Riser – mold interface $h = 500\ \mathrm{W\ m^{-2}\ K^{-1}}$

The riser was insulated by a mineral wool layer.

Boundary Conditions

The riser liquid surface was considered as an adiabatic boundary (thermally insulated) and the remaining mold surface was assumed to be air cooled with the coefficient $h = 10\ \mathrm{W\ m^{-2}\ K^{-1}}$ and ambient temperature $T_{\mathrm{ext}} = 20\,^{\circ}\mathrm{C}$, (Fig. 8.16). Surface emissivity is $\varepsilon = 0.87$.

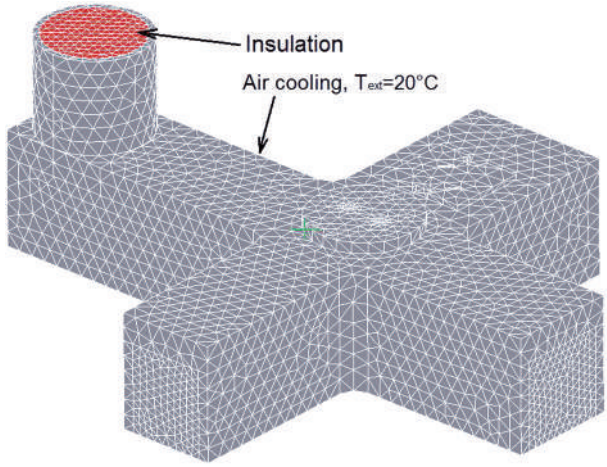


Fig. 8.16 Definition of boundary conditions on the finite element mesh.

The effect of heating inserts on the mold was modelled as a Neumann boundary condition (8.11) and, as these are to be controlled externally by a controller set up in Matlab & Simulink, we defined it by an external function. This function provides the communication between ProCAST and Matlab & Simulink. Electrical heaters of nominal power $P_n = 400\ \mathrm{W}$ are installed in the 26 uniformly distributed holes around

the mold cavity. These heaters have an average surface of $S = 2166\ \text{mm}^2$ and the maximal heat input at 10 V signal is $q = 184.6\ \text{kW m}^{-2}$. The layout of heaters is shown in Fig. 8.6.

Similarly, the effect of chillers was modelled by an external function, where we defined the convection boundary condition (8.12) on the cooling channel surfaces and the control input manipulates the convection heat transfer coefficient between the coolant and chiller.

Initial Conditions

The initial conditions for the preheating simulations were constant temperature settings of $T_0 = 20\,°\text{C}$ for all parts of the mold. For the simulation of the casting solidification, the casting initial temperature was set to $T_{0\text{casting}} = 680\,°\text{C}$ and mold initial temperature field was extracted from the preceding preheating simulation. The computation used the conjugate gradients method with diagonal preconditioning. We performed a series of simulations in order to obtain the mold characteristics, and for the investigation of effects of various preheating and cooling regimes on the final casting quality. This mathematical model was validated experimentally and, if required, some settings were revised, especially the heat transfer coefficients. Furthermore, the resulting validated model has been used in virtual experiments to generate and identify the dynamical characteristics needed for control system configuration. Special attention was given to the temperatures in nodes, located in places corresponding to the real measurement locations of the benchmark casting mold.

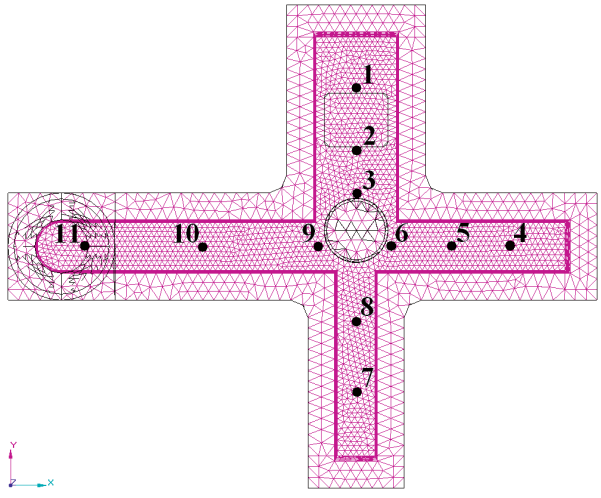


Fig. 8.17 Layout of measurement points $\{\boldsymbol{x}_j\}_{j=1,11}$ *at the mold-casting interface.*

8.3.3 Mold Preheating Control

The mold preheating process, was investigated using Matlab & Simulink. The full scale model was defined by the means of distributed impulse responses $\{Y_i(\boldsymbol{x},k)\}_i$ containing temperatures for each input of $\mathrm{NUT}=36512$ computational nodes in $\mathrm{Nsteps}=4001$ time steps, with the sampling time $T=10\,\mathrm{s}$. One distributed step response $\mathcal{H}H_i(\boldsymbol{x},k)$, in such a configuration assuming double precision data (8 B) takes $\mathrm{NUT}\times\mathrm{Nsteps}\times 8\ \mathrm{B}\approx 1{,}17\ \mathrm{GiB}$ of computer memory and all responses together, 5 of them require 5,84 GiB. This amount of data is not practical to work with on an industrial computer, which necessitates model reduction. Moreover, model reduction is supported by the fact that, in the real process, the desired temperature fields are usually given by discrete values at measurement points. This of course depends on the system complexity and technological requirements. In general, it happens rather rarely that data from each point of the full scale FEM model is required.

Thus, the desired temperature profile $W(\boldsymbol{x},\infty)$ is also defined in a few points of the definition domain and the remaining temperatures are constrained by material properties. This also reflects the reality in industry, where the process engineer prescribes values in measurement points. For our mold, we have a desired temperature profile along the cross arms. Here we do not define temperatures for the rest of the cross, or along other directions, as these are not of interest, moreover are strongly constrained by the material. Such a setup mimics a quasi-1D problem, (Fig. 8.18).

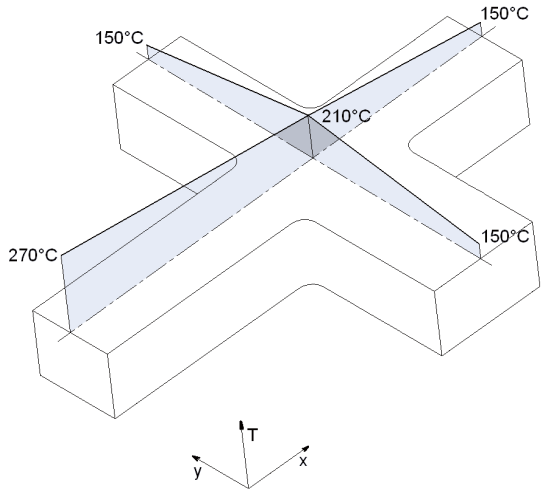


Fig. 8.18 Desired temperature profile in the casting arms.

Numerical dynamical analysis of the preheating process using ProCAST shows that a careful choice of nodes from which values are taken may reduce computational power requirements to obtain an adequate solution of space synthesis approximation problems. We picked the points from the computational mesh which correspond to measurement points of the real mold, i.e. thermocouple locations. By this procedure, we obtain the finite dimensional counterparts of distributed output and reference $\{Y_i(x_j,k)\}_{i=1,n}^{j=1,m}$ and $\{W_i(x_j,k)\}_{i=1,n}^{j=1,m}$. We took 11 outputs for each of the 5 inputs $\{Y_i(x_j,k)\}_{i=1,5}^{j=1,11}$ from the original FEM simulation result, for the points shown in Fig. 8.17. In other words, we are working on a subset of the full scale model

$$\{\mathcal{H}HR_i(x,\infty)\}_{i=1,5,\ x\in\{x_j\}_j} \subset \{\mathcal{H}HR_i(x,\infty)\}_{i=1,5,\ x\in\Omega} \tag{8.22}$$

Temperature field in other locations may be easily estimated using full scale profiles

$$Y(x,\infty)=\sum_{i=1}^{5}\mathcal{H}HR_i(x,\infty)Y_i(x_i,\infty) \tag{8.23}$$

while $Y_i(x_i,\infty)$ are identical for the reduced as well as the full scale model.

Identification of Mold Dynamics

The identification procedure of the mold dynamics is based on the approach described in Chapt. *4*. We investigated the responses of the mold temperature field to step changes in the range 0 - 10% of heating power in individual zones. Temperatures in measurement points are shown in Fig. 8.19 to Fig. 8.23, with respect to the initial condition. These data were collected to form a cell array $\mathrm{H}\{\ \}$ according to the requirements and specifications of the DPS Blockset, (Sect. *6.1.1*).

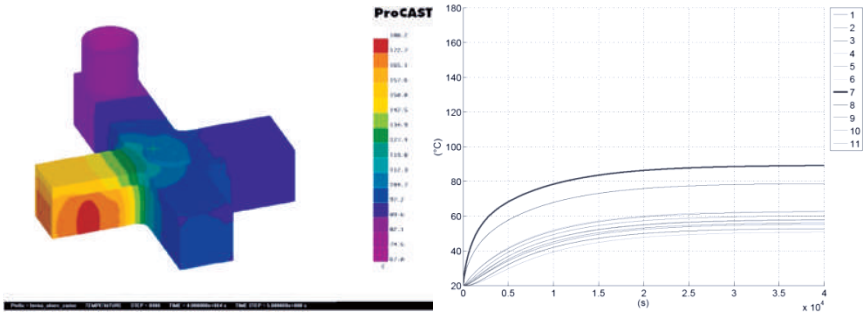


Fig. 8.19 Heating in Zone 1 "thin arm", steady-state temperature field and time plot of temperatures in measurement points.

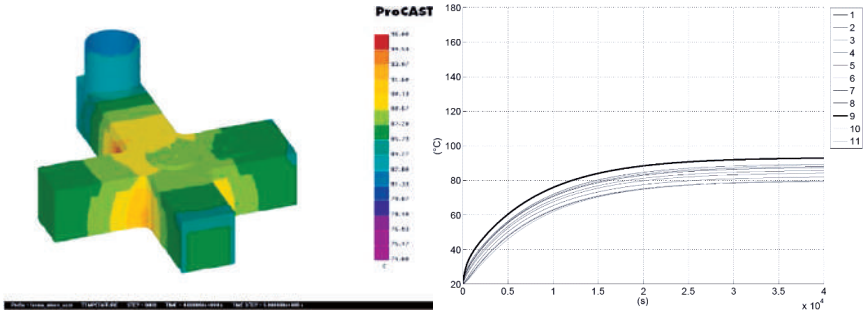


Fig. 8.20 Heating in Zone 2 "hot spot", steady-state temperature field and time plot of temperatures in measurement points (below).

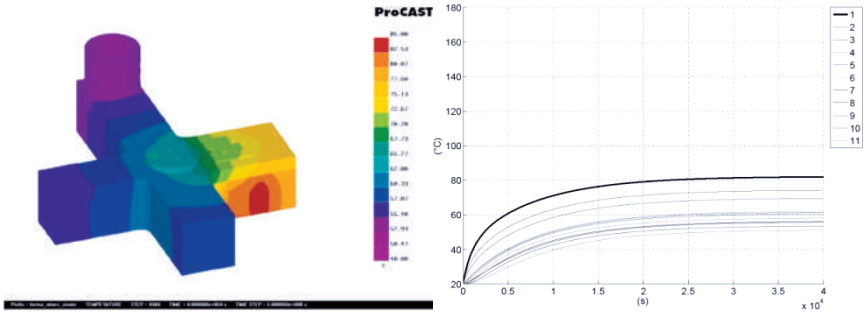


Fig. 8.21 Heating in Zone 3 "thick arm", steady-state temperature field and time plot of temperatures in measurement points.

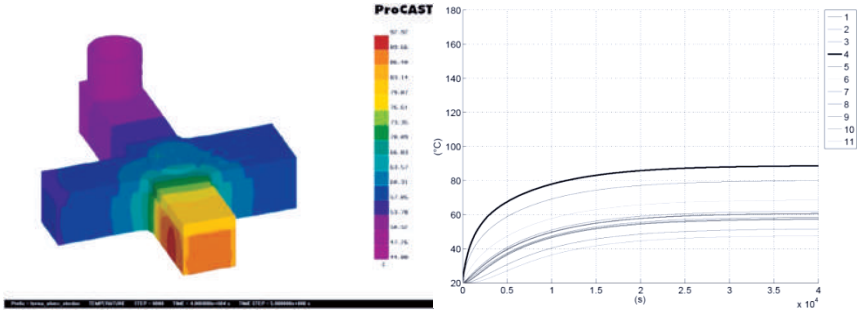


Fig. 8.22 Heating in Zone 4 "center arm", steady-state temperature field and time plot of temperatures in measurement points.

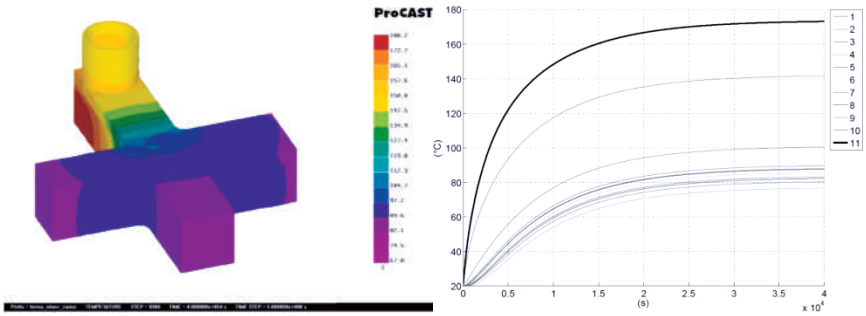
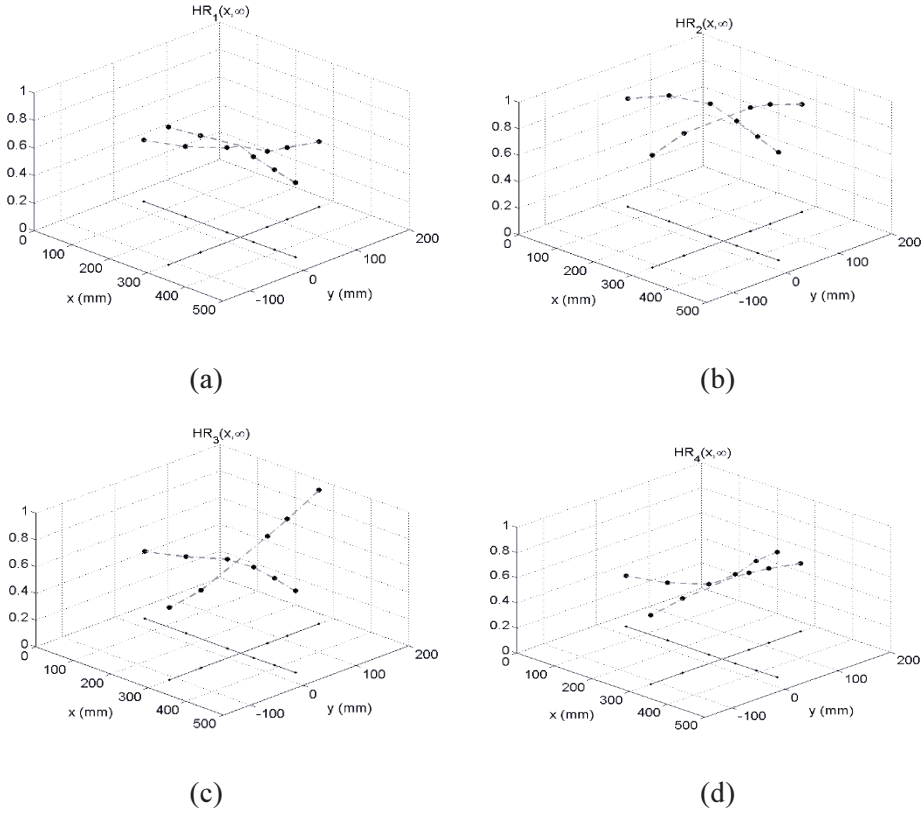
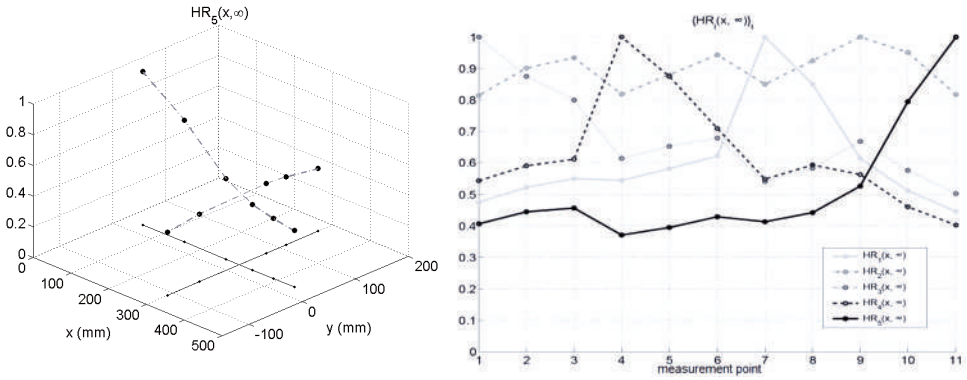


Fig. 8.23 Heating in Zone 5 "rear arm", steady-state temperature field and time plot of temperatures in measurement points.

We have gathered five steady-state profiles of partial distributed step responses (in locations corresponding to measurement locations) $\{\mathcal{H}H_i(\boldsymbol{x},\infty)\}_{i=1,5}$ from the measured data; in line with the procedure described in Chapt. *3*. We have obtained their reduced profiles, (Fig. 8.24).





(e) (f)

Fig. 8.24 Space components of virtual mold dynamics: zone 1 (a), zone 2 (b), zone 3 (c), zone 4 (d), zone 5 (e), quasi-1D display (f).

Remark: The measurement points are shown on a 1D interval and respective values are connected by lines. These lines serve solely the presentation, they do not indicate any physical connection (it is not a 1D problem).

Time domain components are obtained by the identification of SISO models fitted to the responses $\{Y_i(\boldsymbol{x}_i, k)\}_i$, where the points $\{\boldsymbol{x}_i\}_i$ are picked according to the location where the highest gain partial response is observed, Fig. 8.19 – Fig. 8.23. For heaters arranged in 5 zones, the highest gain response locations are listed in Table 8.2.

Zone 1	Zone 2	Zone 3	Zone 4	Zone 5
$\boldsymbol{x}_1 = 7$	$\boldsymbol{x}_2 = 9$	$\boldsymbol{x}_3 = 1$	$\boldsymbol{x}_4 = 4$	$\boldsymbol{x}_5 = 11$

Table 8.2 Highest gain points in individual zones.

Responses taken from these points were identified and Laplace transfer functions were obtained by the *System Identification Toolbox*. Responses of identified models were compared to the FEM simulation, Fig. 8.25. Process models were sought for in the form

$$G_S(s) = K_\mathrm{p} \frac{T_\mathrm{z} s + 1}{(T_{\mathrm{p}1} s + 1)(T_{\mathrm{p}2} s + 1)} \tag{8.24}$$

for individual inputs $\{U_i(k) = 1(k)\}_i$. The zero $-1/T_\mathrm{z}$ in the numerator of transfer function (8.24), was tuned semi-automatically by limiting the search interval. The acceptance criterion was a good fit of responses in Fig. 8.25 (a) to (e). Identified

models in form of *4-D idproc* transfer functions were obtained by the standard least squares method and the results are listed in Table 8.3.

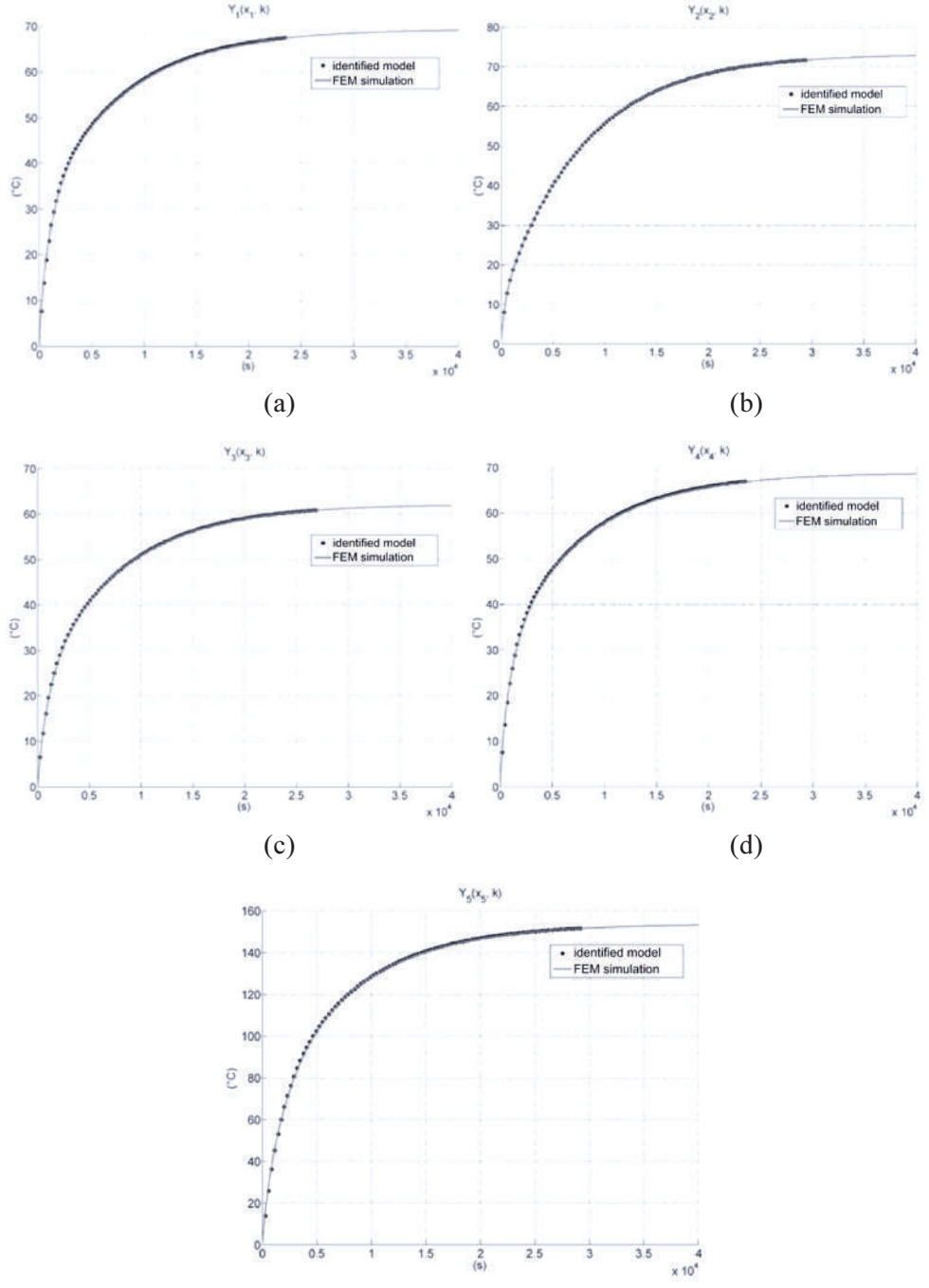


Fig. 8.25 (e) Identification of time domain dynamics – lumped parameter responses: zone 1 (a), zone 2 (b), zone 3 (c), zone 4 (d), zone 5 (e).

No. of zone:	1	2	3	4	5
K_p	69.29	73.15	62.08	68.67	153.42
T_{p1}	7625.30	7853.10	7682.30	7503.00	7500.30
T_{p2}	933.62	375.96	986.40	929.08	1520.20
T_z	3750.00	1450.00	3300.00	3600.00	3800.00

Table 8.3 Identified parameters of the virtual mold.

We identified the characteristics of our benchmark mold in the experimental plant by the same approach, Fig. 8.26, Table 8.4. By comparing these to the characteristics of the virtual mold, we observe differences in gains and time constants, which is attributed to the included radiation model (8.13) and to the fact that we have identified a linear model of a nonlinear system.

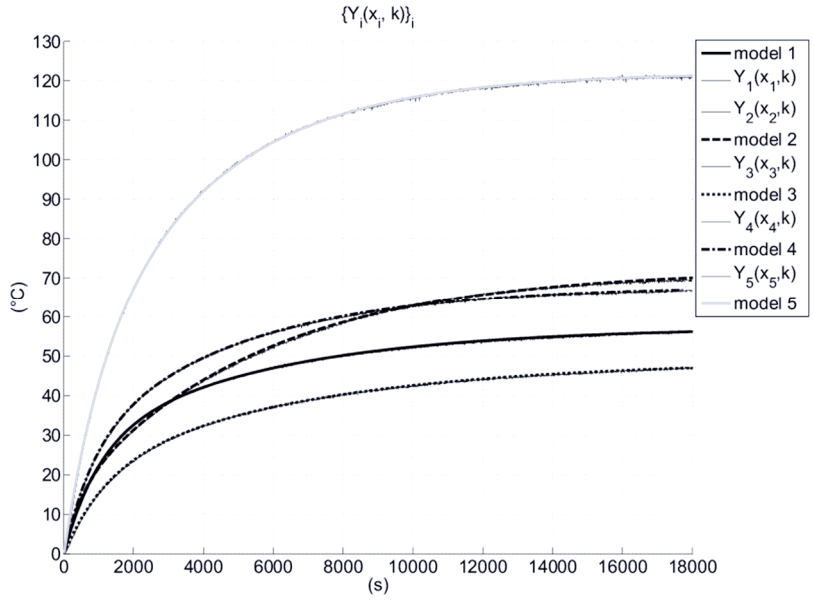


Fig. 8.26 Lumped parameter responses of the real mold taken from measurement probes (see, Fig. 8.6, Fig. 8.17) and respective SISO model outputs.

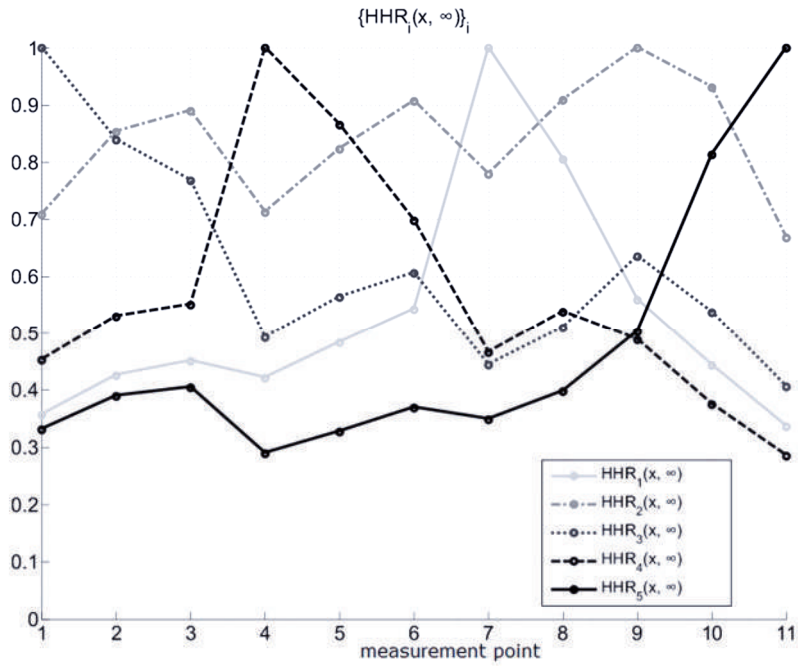


Fig. 8.27 Reduced steady-state profiles of the experimental mold (see Fig. 8.24 for comparison).

No. of zone:	**1**	**2**	**3**	**4**	**5**
K_p	57.17	72.07	48.68	67.74	121.91
T_{p1}	5524.16	5469.10	6486.30	4715.17	3901.26
T_{p2}	976.9	468.33	1189.18	843.27	1051.21
T_z	3100.00	1400.00	3300.00	2300.00	2000.00

Table 8.4 Identified parameters of the real mold (10% step of the total heating power).

Mold Preheating Controller Design

The control synthesis is based on the approach described in Chapt. *5 and 7,* Fig. 5.9 and Fig. 7.3. In the time domain, the individual SISO controllers are based on the P-P1 structure as outlined in Huba (2007), Fig. 8.28.

We have performed a programmed heating, first to a uniform temperature field $W(x,\infty) = 100°\text{C}$. After reaching steady-state, this was followed by a new set-point as shown in Fig. 8.29 with the reference values

$$W_{\text{field}} = [150 \quad 160 \quad 170 \quad 150 \quad 160 \quad 170 \quad 160 \quad 170 \quad 180 \quad 220 \quad 250].$$

Lumped, as well as distributed quantities of the control process are shown in Fig. 8.30 to Fig. 8.34.

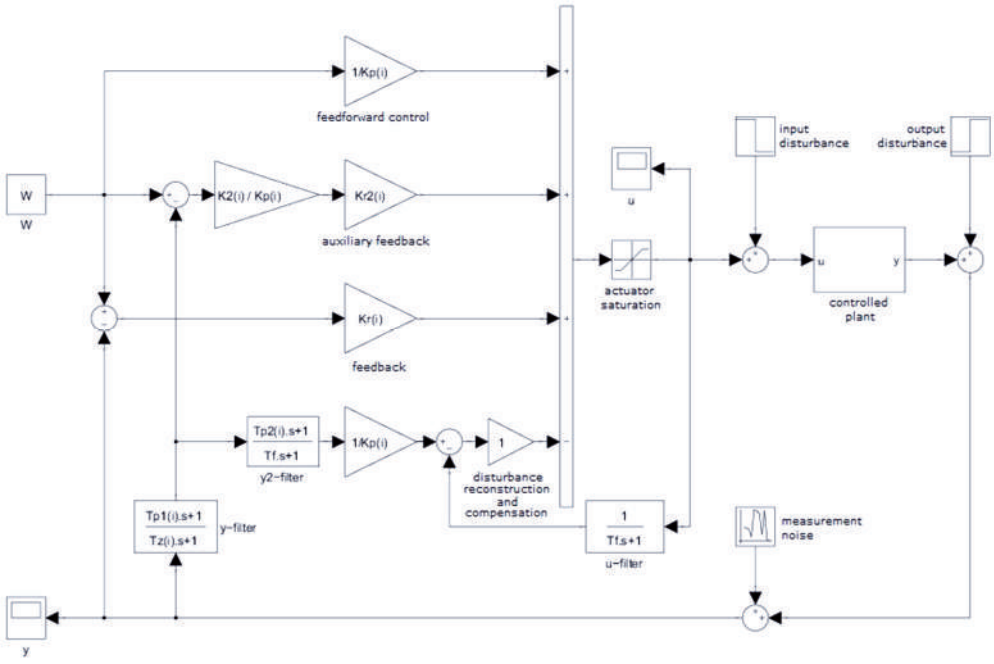


Fig. 8.28 P-P1 SISO Controller structure.

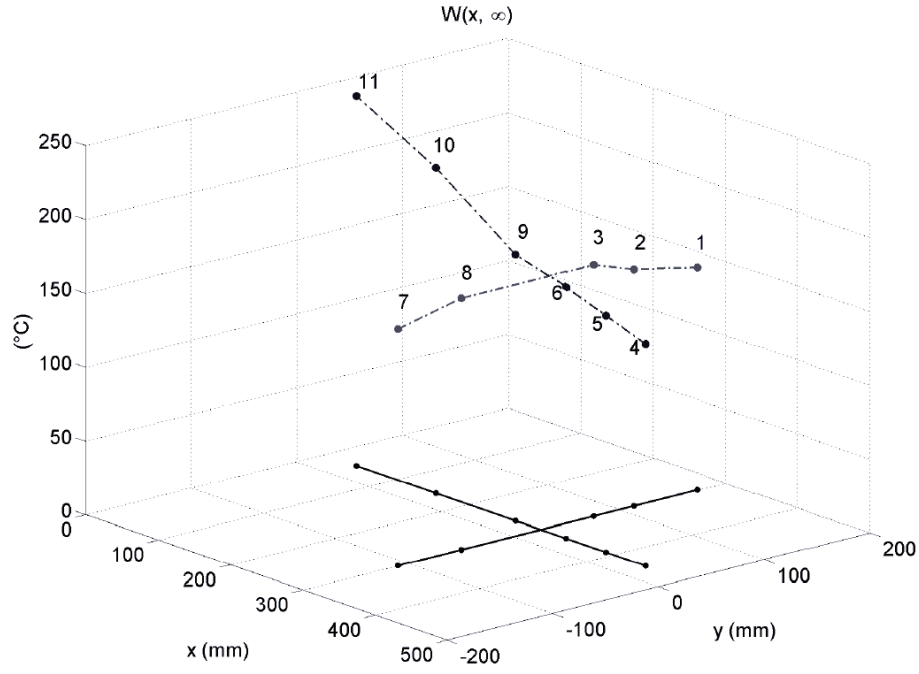


Fig. 8.29 Desired temperature field W_{field}.

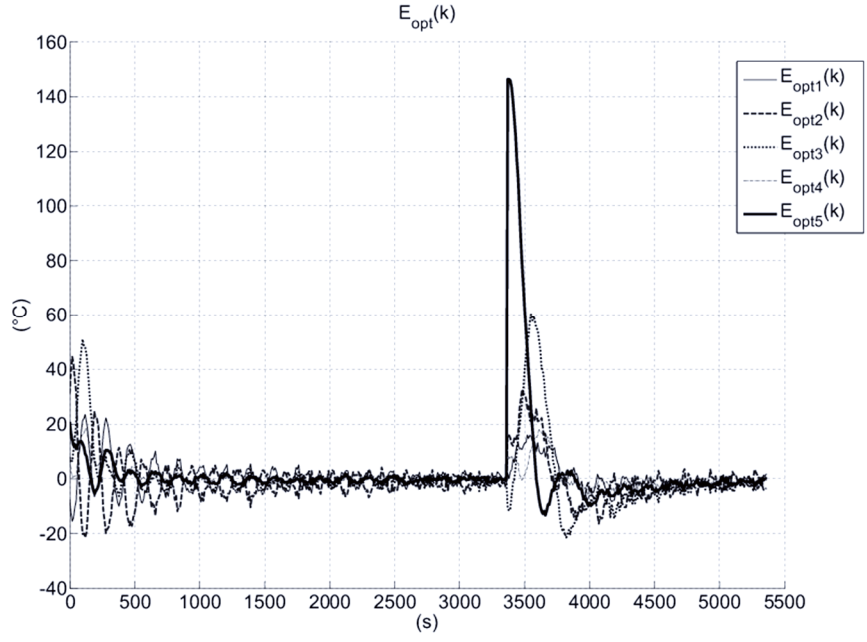


Fig 8.30 Controlled heating up to $W(\boldsymbol{x},\infty)=100\,^{\circ}\mathrm{C}$ *followed by* $W_{\text{field}}(\boldsymbol{x},\infty)$ *: optimal approximation coefficients of the distributed control error.*

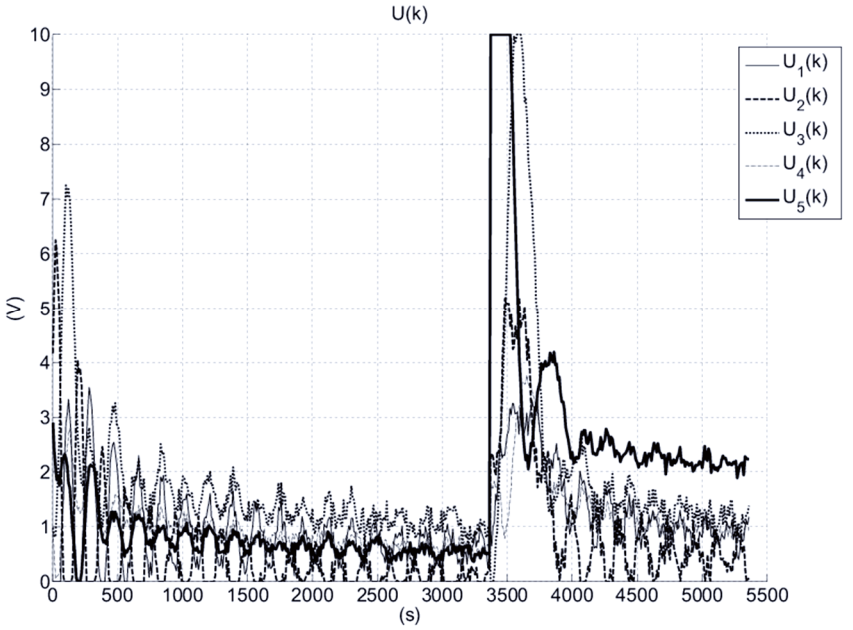


Fig. 8.31 Control input in individual zones.

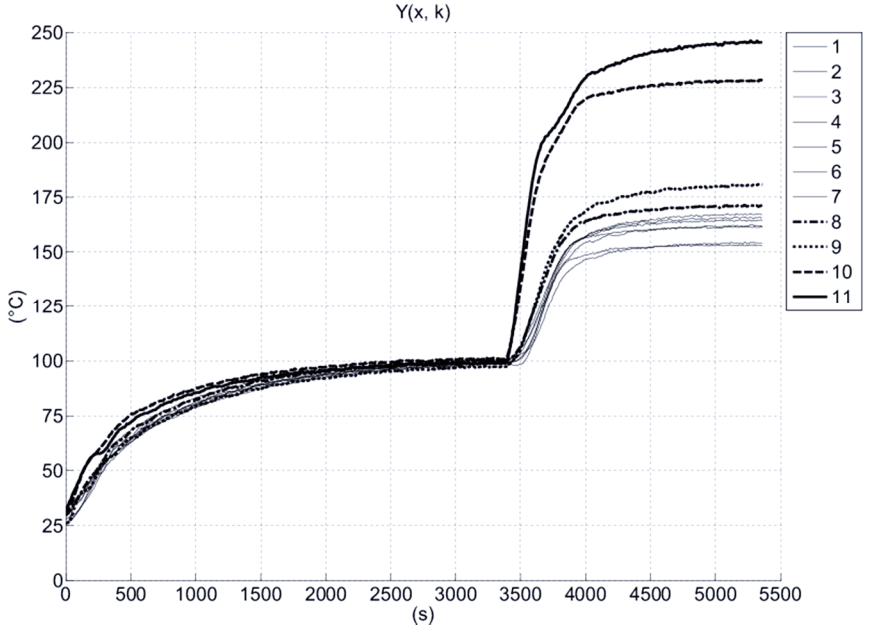


Fig. 8.32 Distributed manipulated variable.

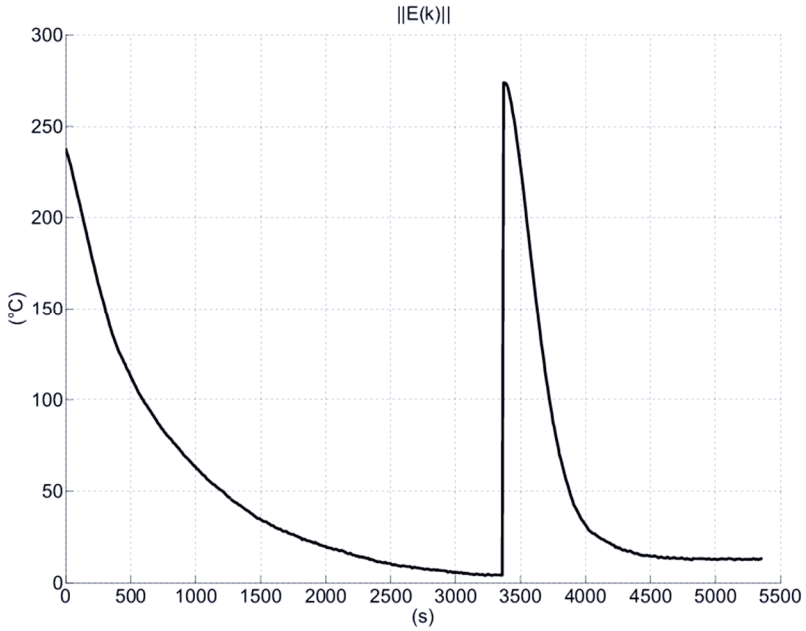


Fig. 8.33 Quadratic norm of control error.

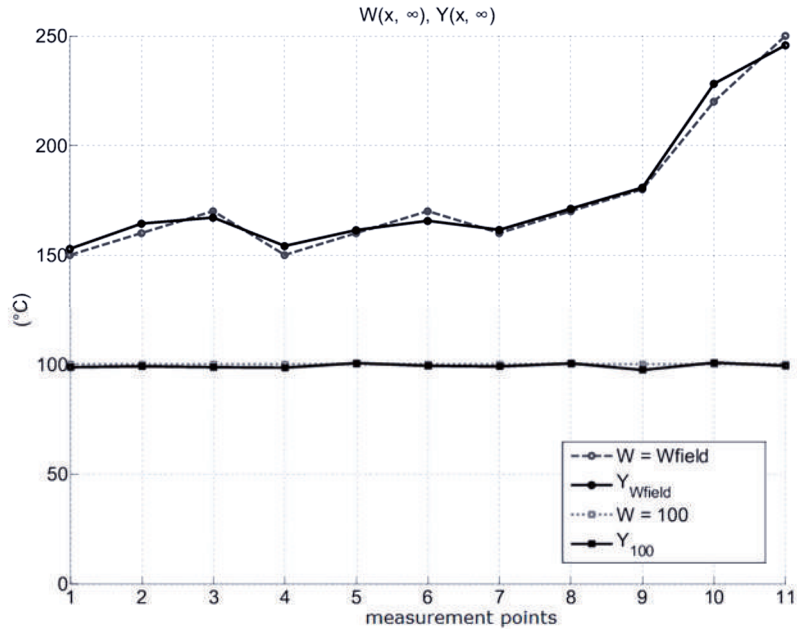


Fig. 8.34 Steady-state profiles reached (for better readability shown in 1D).

We can observe an oscillating nature of the set-point, nevertheless, the effect of peaks occurring on the manipulated variable as well as on the quadratic norm of the control error is practically negligible. The latter can be attributed to the considerable thermal inertia of the steel mold.

The focus was on the control of the temperature field in select locations of the definition domain, and the error between the set-point and the manipulated variable. It is, however, important to note that in the real process, the responses are present on the entire domain of definition—the same in the case of simulation—meaning that it is possible to evaluate the control error on the entire definition domain of the distributed manipulated variable, see Fig. 8.18 for details. In other words, it is possible to control the distributed parameter system in reality on its entire definition domain with the prescribed norm of the error evaluated by simulation, as the responses of the physical system are naturally given on the definition set of the real model of the casting mold.

Mold Preheating Optimization

In order to optimize the temperature field of the mold preheating procedure, we performed a study of the preheating temperature on temperature gradients along the casting axes. The freely solidifying casting was taken as a reference and the gradients compared to gradients resulting from other preheating temperatures. We optimized these gradients by modifying the preheating temperature field, (Fig. 8.35-39).

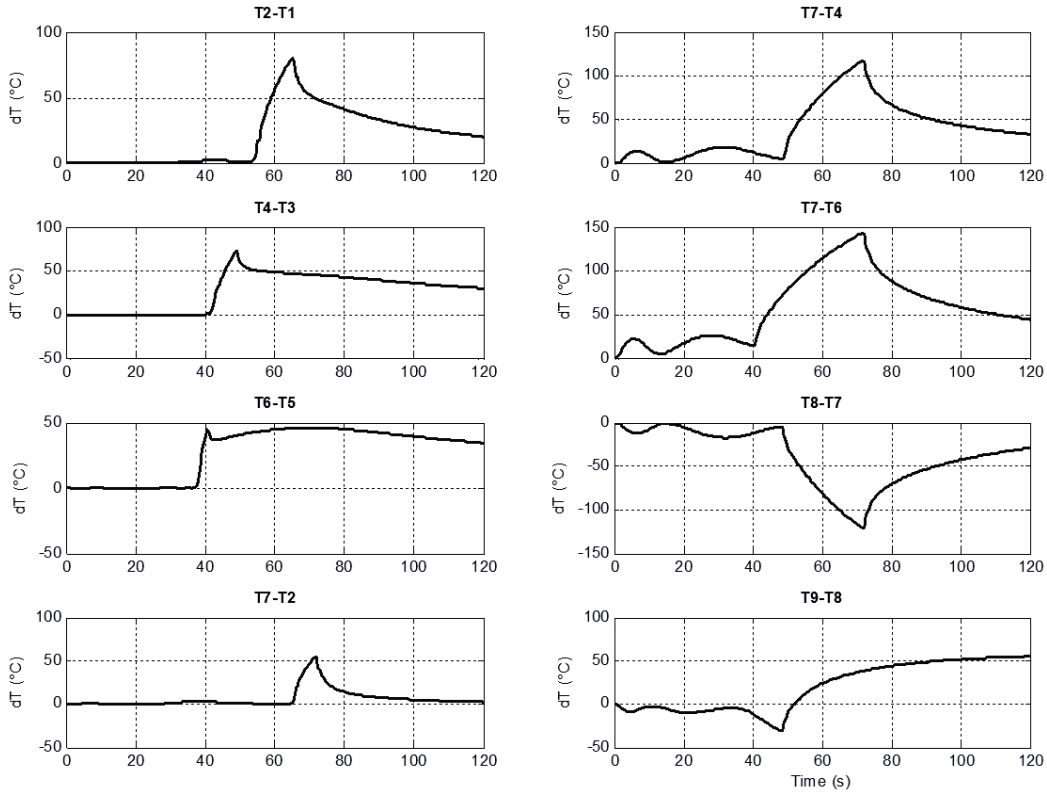


Fig. 8.35 Temperature gradients in the center of the casting cross-section. The mold was uniformly preheated to 150°C and chillers were not operational. (Layout of measurement points are shown in Fig. 8.17 and Fig. 8.5. For comparison see also Fig. 8.7, where the built in thermocouple probes are shown. Each one, except No. 3, 6 and 9, has a corresponding pair among wires inside the casting. In the hot spot, instead of 3 only one is present on the axis of chillers CH5A and CH5B in the middle.)



Fig. 8.36 Uncontrolled solidification results in cutting molten metal supply to the hot spot which has the consequence of a large shrinkage. The picture shows unsolidified parts of the casting where we can see the closed channel between the riser and the hot spot.

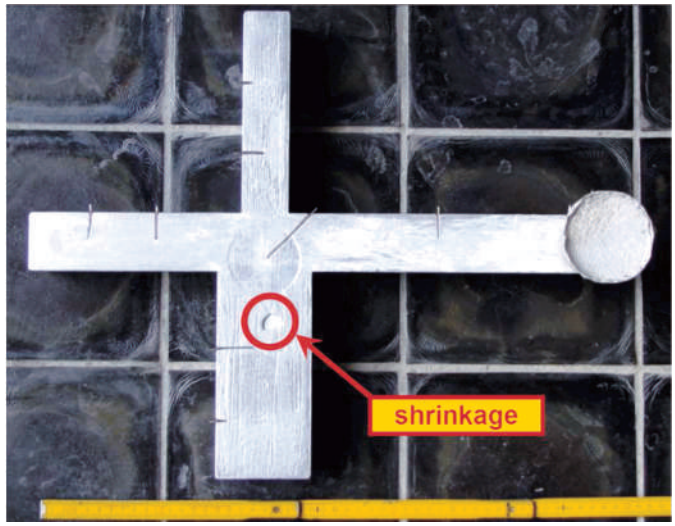


Fig. 8.37 Experiments confirm simulations. Forces due to the vacuum inside the shrinkage even damaged the solidified outer shell of the casting.

The plots in Fig. 8.35 clearly indicate that the worst situation is with the gradients along the casting axis from the hot spot to the riser, $T_8 - T_7$ and $T_9 - T_8$. At first, this problem is solved by mold preheating, thus increasing temperature in the problematic zones (we increased the riser zone and from the others just the hot spot zone to make

sure the field is feasible). This finally results in lower heat transfer from the casting to the mold, and in effect, increasing casting temperature in the problematic location.

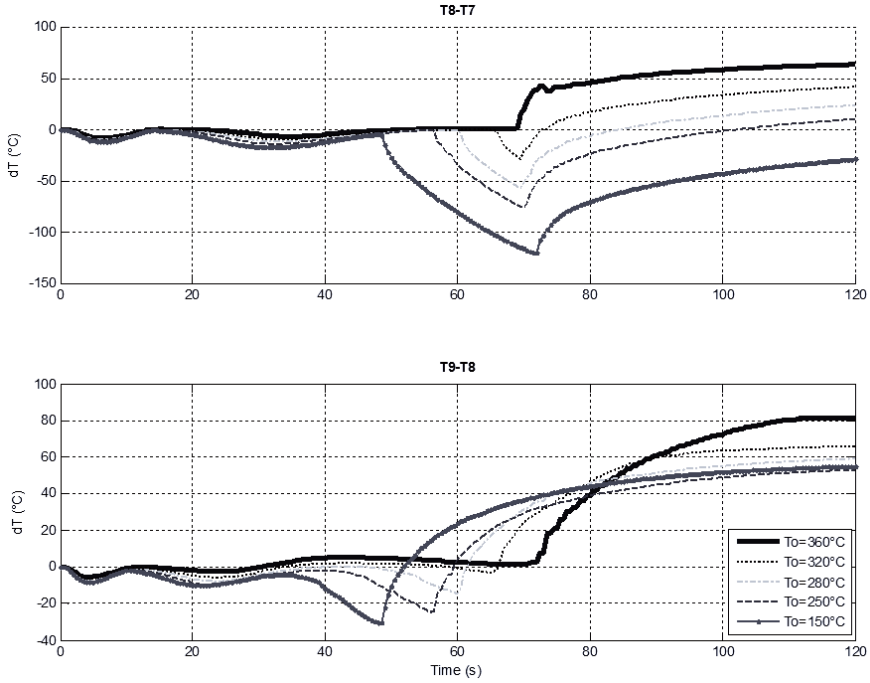


Fig. 8.38 Evolution of temperature gradients along the casting axis from the hot spot to the riser for various preheating temperatures. The remaining gradients are practically the same as shown in Fig. 8.35.

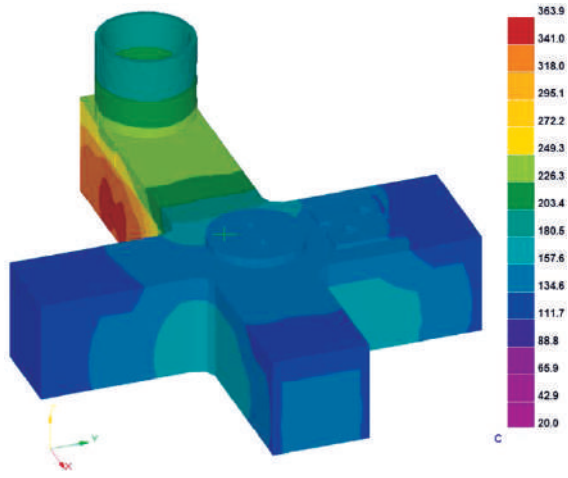


Fig. 8.39 Temperature field in the preheated mold.

We were able to reach a mold temperature field by gradually increasing the preheating temperature in the problematic zones

$$T_W = [150 \quad 170 \quad 210 \quad 150 \quad 170 \quad 210 \quad 150 \quad 170 \quad 210 \quad 270 \quad 360],$$

such that in connection with casting cooling it guarantees acceptable course of these gradients.

Remark: One might even achieve a positive gradient throughout the whole solidification process, however, the question arises whether this approach would lead to an overall process optimization and production cost savings. To guarantee a positive gradient during the process governed just by mold preheating, one would have to heat the mold to very high temperatures and the gradient between zones 5 and 2 would exceed allowable limits. A gradient with excessive temperatures has a deteriorating effect on mold lifetime, moreover, such an action would also be energetically inefficient – at high mold temperatures, casting solidification would take considerably more time, which in turn would cause additional economic losses.

8.3.4 Casting Solidification Control

The feedback control of the casting solidification is a much more complicated problem. Measurement of the casting temperature requires its penetration with thermocouples, which is unacceptable in general, but can be feasible for tuning and laboratory test purposes.

Therefore, a state observer or a software sensor is to be included in the control loop, which on the other hand requires a very precise mathematical description of the process and this may be very computation-intensive. Unfortunately, such a model is not easily implementable on common industrial control systems. Model reduction is therefore of utmost importance and research on the experimental plant was directed exactly in this direction along with the implementation of the reduced model to a standard industrial PLC. Fig. 8.42.

In principle, we are speaking about gradient control, thus the control objective is to keep positive gradients that are as high as possible in the direction of the chillers at the casting arm ends through the cross joint to the riser (similarly to Fig. 8.18). The controller follows temperature differences between neighboring thermocouples and increases coolant flow with decreasing gradients. However, the technology requires a minimum flow rate in order to avoid the boiling of the coolant. Moreover, one has to keep an eye on the neighboring gradients as well, so that e.g. maximization of the

gradient $T_9 - T_8$ and $T_8 - T_7$ by means of chillers CH5A&B does not cause negative gradients $T_7 - T_2$, $T_7 - T_4$ and $T_7 - T_6$ (see Fig. 8.40 - 8.42 d).

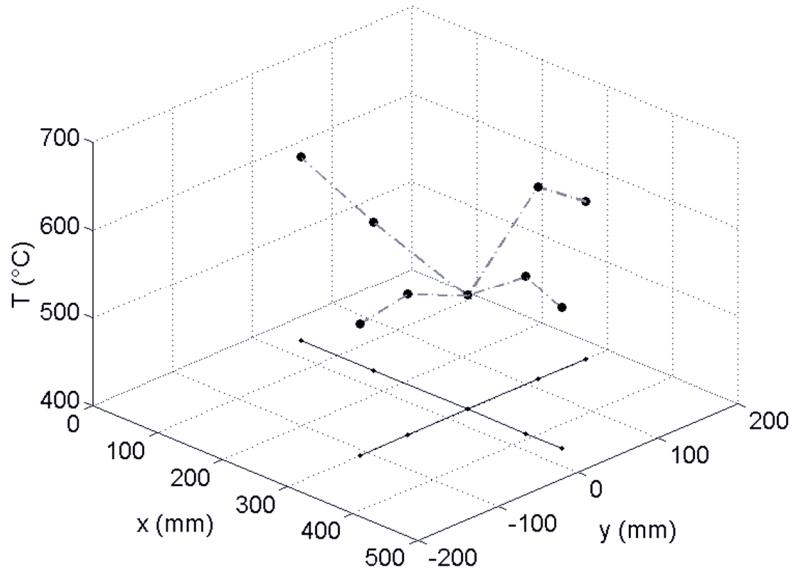


Fig. 8.40 Unacceptable temperature field as a result of overcooling of the cross joint.

Furthermore, the system identifies the moment when solidification starts in the particular measurement points (i.e. latent heat is released, Fig. 8.41) and sets the flow rate through the respective chiller to maximum in order to achieve the fastest possible heat removal.

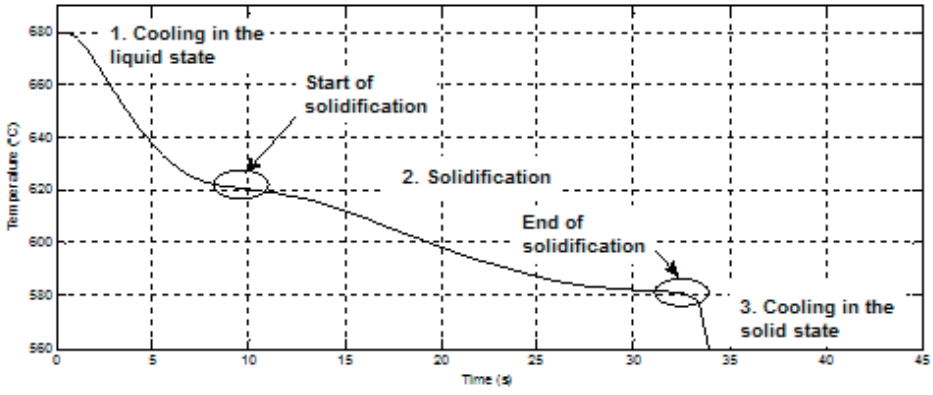


Fig. 8.41 Typical solidification curve. The moment when the solidification starts is indicated by the time derivative of temperature, which in this case of occurring phase change approaches zero.

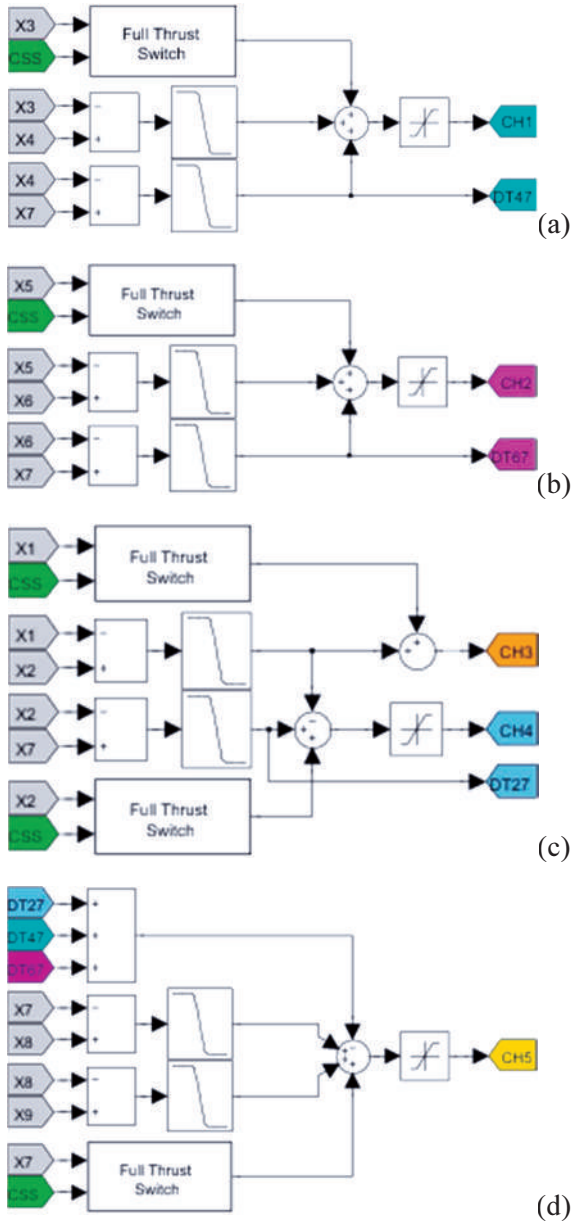
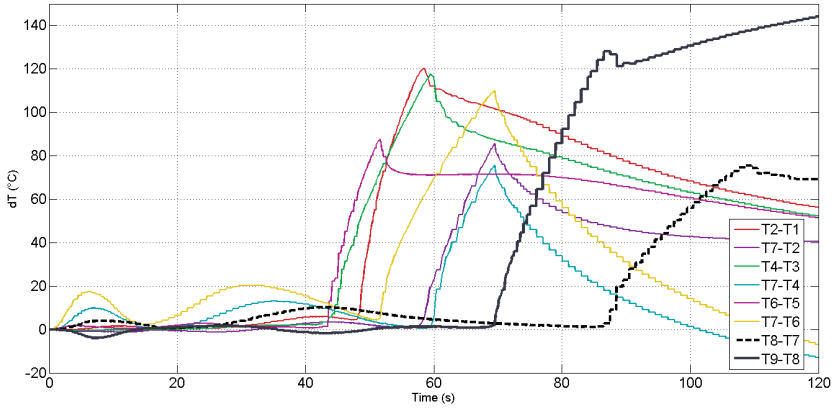


Fig. 8.42 Controller schematics for individual cooling zones: CH1 (a), CH2 (b), CH3 and CH4 (c), CH5 (d).

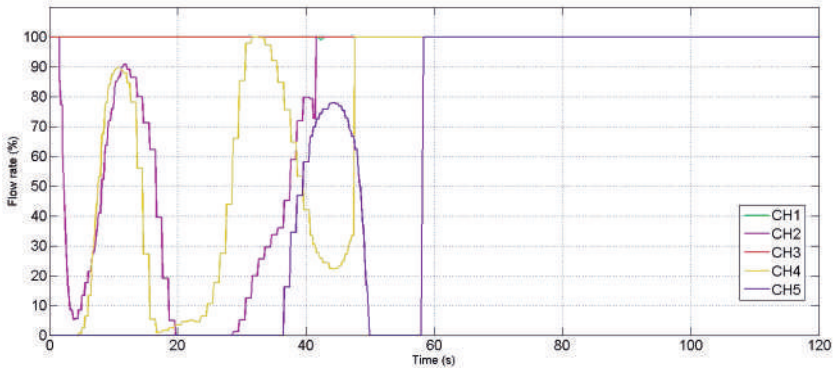
Controller Operation

Signals X1 – X9 are measured temperatures by thermocouples, DT*ij* represents temperature gradients between X*i* and X*j*. Each of these is transformed by a *Z-function* (which rises with decreasing gradient) in order to work with values between 0 and 1 (finally being the flow rate fraction). At the end, there is a saturation block which

limits the output to 100% power. The CSS signal becomes 1 in the moment the molten metal is poured into the mold cavity. This in fact triggers the controller operation. The "*Full Thrust Switch*" block identifies the moment when solidification starts and triggers full cooling medium flow through the respective chiller. Controller tuning parameters are the steepness of Z-functions and the threshold value for temperature change rate at which the controller assumes that the solidification started. These are tunable for each block separately. The control process is shown in Fig. 8.43, (Buček (2010), Noga (2011)).



(a)



(b)

Fig. 8.43 Temperature gradients during controlled solidification (highlighted the critical ones T8-T7 and T9-T8) (a), cooling flow rates (b).

Remark: negative temperature gradients at times 100-120 s are not of concern, as by that time the casting is already solidified at respective measurement positions.

9. ROBUST CONTROL OF THE CASTING MOLD TEMPERATURE FIELD

When implementing control systems based on the model of the controlled process in technical practice, the controllers must operate properly despite the differences between the model and the dynamics of the real process. This holds true for systems with lumped parameters as well as for distributed parameter systems. Robust control theory represents one possible approach to the design of the control loop with uncertainties. When designing a robust control system, the formulation of a nominal model of the controlled process is required, as well as the definition of uncertainties. Uncertainties are considered for the model along with an analysis of their impact on the model in the time and frequency domains. In this chapter, problems of robust control of distributed parameter systems will be discussed in relation to the discussion of Chapters *2*. - *5*. and in connection to Chapter *8*.

9.1 Sources of Uncertainties in the Control System

The control of technological processes, such as HLDS, is based on the decomposition of dynamics to space and time components, which are then used in control synthesis. The output of the RHLDS block in Fig. 5.6 is a block for spatial control synthesis, SS2 addresses the role of the approximation of the distributed output on the set of partial distributed outputs $\{YR_i(\boldsymbol{x},k)\}_{i,k}$ (see approximation problem (5.11)), while the outputs $\{\breve{Y}_i(\boldsymbol{x}_i,k)\}_i$ belong to transfer functions $\{SH_i(\boldsymbol{x}_i,z)\}_i$, that represent the time component of the plant with distributed parameters, see relation (3.4). Because for $k \to \infty$ the responses $\{YR_i(\boldsymbol{x},k)\}_{i,k}$ will converge to responses $\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_i$, see Eq. (3.25), reduced steady-state responses are often used in decomposing the distributed output in engineering practice, where control is usually realized in the close neighborhood of steady-state operating modes instead at $\{YR_i(\boldsymbol{x},k)\}_{i,k}$.

Responses obtained at the output block SS2 $\left\{\breve{Y}_i(k)\right\}_i$ are then partly modified due to the response $\left\{\breve{Y}_i(\boldsymbol{x}_i,k)\right\}_i$. This chapter will concentrate on solving control problems defined by this type of uncertainty but, of course, there are several other categories of uncertainties present in the control process of the cast temperature because of the physical properties of the process.

Generally, the uncertainty of the model includes all the differences, in which the model differs from the real process, as described by Veselý (2007). To take uncertainties into account, for this case we assume that the dynamic behavior of the controlled system is described not only by a simple linear time-invariant model, but as a set Ψ_i of simple linear time-invariant models, which can be expressed in various forms. This set of frequency domain models $S_i(\boldsymbol{x}_i, j\omega)$ is given by

$$\Psi_i = \left\{S_i : \left| \; S_i(\boldsymbol{x}_i, j\omega) \; - \; \tilde{S}_i(\boldsymbol{x}_i, j\omega) \; \right| \le \bar{L}_{ai}(\omega)\right\} \tag{9.1}$$

where $\tilde{S}_i(\boldsymbol{x}_i, j\omega)$ is the nominal model. Any model on the set Ψ_i satisfies the condition

$$S_i(\boldsymbol{x}_i, j\omega) = \tilde{S}_i(\boldsymbol{x}_i, j\omega) + \bar{L}_{ai}(\omega) \tag{9.2}$$

$$\left|L_{ai}(j\omega)\right| \le \; \bar{L}_{ai}(\omega) \; , \; \forall \; S_i \in \Psi_i \tag{9.3}$$

where $L_{ai}(j\omega)$ is the additive uncertainty and $\bar{L}_{ai}(\omega)$ is the boundary of the additive uncertainty.

To express the uncertainty of dynamic models, multiplicative uncertainty $L_{mi}(j\omega)$ can also be used, which in relation to the additive uncertainty is expressed in the form

$$L_{mi}(j\omega) = \frac{L_{ai}(j\omega)}{\tilde{S}_i(\boldsymbol{x}_i, j\omega)} \; ; \qquad \bar{L}_{mi}(\omega) = \frac{\bar{L}_{ai}(\omega)}{\left|\tilde{S}_i(\boldsymbol{x}_i, j\omega)\right|} \tag{9.4}$$

where $\bar{L}_{mi}(\omega)$ is the boundary of the multiplicative uncertainty.

9.1.1 Control Loop with Robust IMC Controllers

The general structure of the control loop with the internal model of the system, known as the Internal Model Control (IMC), (Morari (1989)) is shown in Fig. 9.1. This structure can be applied both to continuous and discrete-time control loops, and may be easily transformed to the classical feedback structure. The relationship between the feedback controller $R(z)$ and the IMC controller $Q(z)$ for the nominal model of the controlled system $\tilde{S}(z)$ is given by

$$Q(z) = \frac{R(z)}{1 + \tilde{S}(z)R(z)}; \qquad R(z) = \frac{Q(z)}{1 - \tilde{S}(z)Q(z)} \tag{9.5}$$

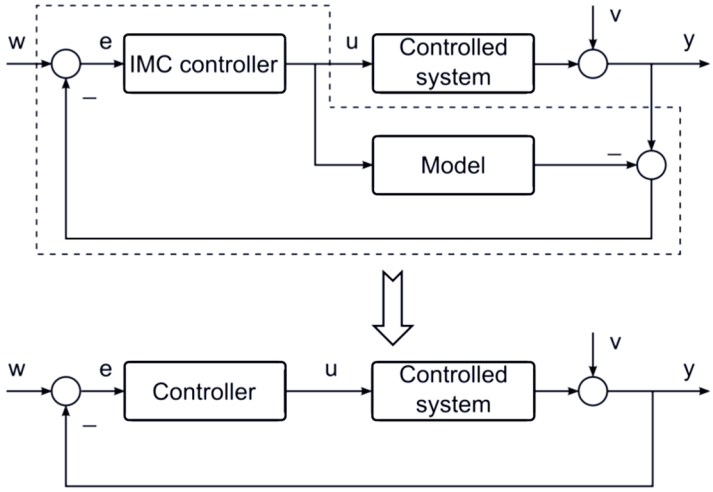


Fig. 9.1 Structure of the control loop with the internal model system and its transformation into a simple feedback control loop.

The IMC structure is generally characterized by good control loop stability, dynamic response and robustness. Several practices were outlined for the design IMC controllers according to different criteria, and their subsequent transformation to classical feedback controllers. These were used in applications in various areas of control of dynamic systems including DPS, (Wang (2001), Vítečková (2006), Vilanova (2008)).

Robust control for DPS, interpreted as LDS, is in this case carried out using the IMC structure control loop, Fig. 9.2. The IMC structure is included into the temporal control synthesis block TS for the overall DPS feedback control loop. As the nominal transfer function representing the model of the process consists of single control loops,

these are considered to be lumped transfer functions of the controlled system $\{SH_i(\boldsymbol{x}_i, z)\}_i$, by Eq. (3.4).

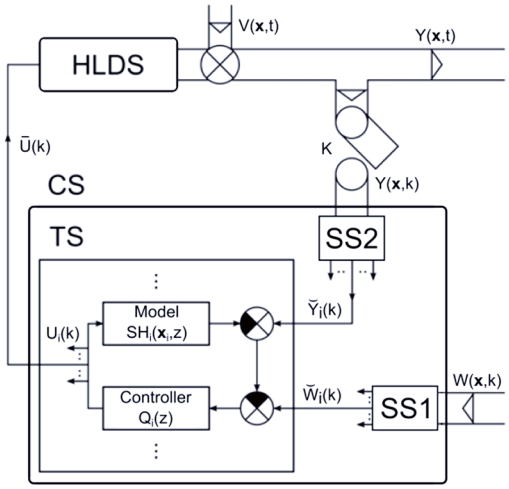


Fig. 9.2 Structure of the DPS robust control loop based on IMC.

HLDS - discrete lumped input and distributed output system with zero order hold units.

CS - control system.

TS - time component of the control synthesis.

K - sampling.

SS1,SS2 - space component of the control synthesis.

$\{Q_i(z)\}_i$ - Z-transform functions of the discrete transfer functions of the IMC controllers.

$\{SH_i(\boldsymbol{x}_i, z)\}_i$ - time components of the dynamics of the plant.

$\bar{U}(k)$ - vector of discrete lumped inputs.

$V(\boldsymbol{x}, t)$ - distributed disturbance.

$\{\tilde{W}_i(k)\}_i$ - components of the discrete lumped reference.

$W(\boldsymbol{x}, k)$ - overall distributed reference.

$\{\tilde{Y}_i(k)\}_i$ - approximating coefficients of the distributed manipulated variable.

$Y(\boldsymbol{x}, t)$ - distributed output.

In the space synthesis block SS2 when decomposing the manipulated variable (see approximation problem (5.11)) instead of using $\{YR_i(\boldsymbol{x}, k)\}_{i,k}$ responses

$\left\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\right\}_i$ will be utilized, while instead of obtaining $\left\{\breve{Y}_i(\boldsymbol{x}_i,k)\right\}_i$ the approximating coefficients $\left\{\breve{Y}_i(k)\right\}_i$ will be achieved.

H_2 optimal controllers $Q_i(z)$ for input $\breve{W}_i(k)$ in the form of a unit step function $\gamma(z)=\dfrac{z}{z-1}$ are determined by minimizing the criterion

$$\min_{Q_i(z)}\left\| E_i(\boldsymbol{x}_i,z)\right\|_2=\min_{Q_i(z)}\left\|\left(1-SH_i(\boldsymbol{x}_i,z)Q_i(z)\right)\gamma(z)\right\|_2 \tag{9.6}$$

In the first step, stable nominal transfer functions of the system $SH_i(\boldsymbol{x}_i,z)$ are factorized

$$SH_i(\boldsymbol{x}_i,z)=SH_{iN}(\boldsymbol{x}_i,z)SH_{iM}(\boldsymbol{x}_i,z) \tag{9.7}$$

where $SH_{iN}(\boldsymbol{x}_i,z)$ contains unstable zeros of the transfer function $SH_i(\boldsymbol{x}_i,z)$.

The discrete IMC controller $Q_i(z)$ is then in the form

$$Q_i(z)=SH_{iM}(\boldsymbol{x}_i,z)^{-1} \tag{9.8}$$

In the next step, the controller $Q_i(z)$ is augmented by the filter $F_i(z)$ with a tunable parameter α_i

$$F_i(z)=\frac{(1-\alpha_i)}{z-\alpha_i} \tag{9.9}$$

The final formulation of the discrete IMC controller with the filter $Q_{iF}(z)$ will be

$$Q_{iF}(z)=Q_i(z)F_i(z)=Q_i(z)\frac{(1-\alpha_i)}{z-\alpha_i} \tag{9.10}$$

The filter parameter α_i is the only adjustable parameter of the discrete IMC controller $Q_{iF}(z)$. We determine it by the solution of the optimization problem, so that the conditions of robust stability and robust performance are met

$$\left|F_i\left(e^{j\omega T}\right)\right| \;<\; \left[\left|S_i(\boldsymbol{x}_i,j\omega)Q_i\left(e^{j\omega T}\right)\right|\overline{L}_{mi}(\omega)\right]^{-1},\quad 0\le\omega\le\frac{\pi}{T} \tag{9.11}$$

$$\left| Q_i(j\omega) \right| \left| \bar{L}_{ai}(\omega) \right| + \left| 1 - S_i(\boldsymbol{x}_i, j\omega) Q_i(j\omega) \right| G_{wi}(\omega) < 1, \quad 0 \le \omega \le \frac{\pi}{T} \tag{9.12}$$

For $SH_i(\boldsymbol{x}_i, z) = SH_{iM}(\boldsymbol{x}_i, z)$ and the discrete Filter (9.9) controller $R_i(z)$ will be in the equivalent discrete-time simple feedback control loop of the time synthesis, in the form

$$R_i(z) = \frac{SH_{iM}(\boldsymbol{x}_i, z)^{-1} F_i(z)}{1 - SH_i(\boldsymbol{x}_i, z) SH_{iM}(\boldsymbol{x}_i, z)^{-1} F_i(z)} = \frac{1}{SH_{iM}(\boldsymbol{x}_i, z)} \cdot \frac{F_i(z)}{1 - F_i(z)} \tag{9.13}$$

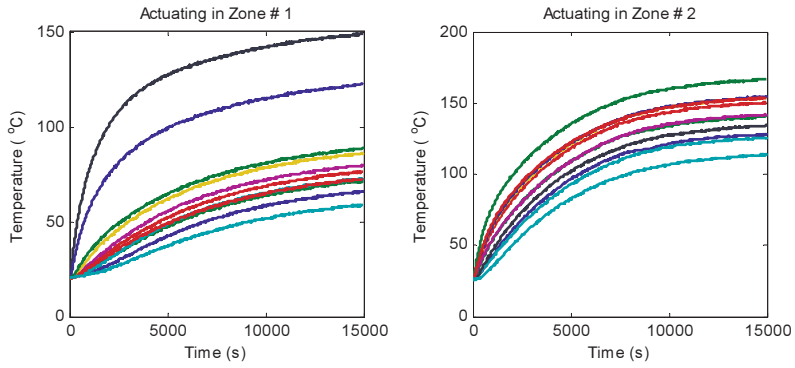
and after rearrangement

$$R_i(z) = \frac{1}{SH_{iM}(\boldsymbol{x}_i, z)} \cdot \frac{1 - \alpha_i}{z - 1}. \tag{9.14}$$

9.2 Identification of the Dynamical Characteristics of the Mold

Zonal heating was carried out in the range (0-25%) of the heating power on the experimental equipment described in Section *8.2*, in order to design a robust control of the temperature fields in the process of preheating. Compared to the 10% step change of the heating power (used in the identification of the dynamic characteristics in the previous chapter) a wide dispersion of the parameters is expected due to the impact of the nonlinear dynamics of the identified system.

The measured temperature in a step change of power heating elements in each zone is shown in Fig. 9.3 a-e and a steady-state temperatures in Fig. 9.4 a-e.



(a) (b)

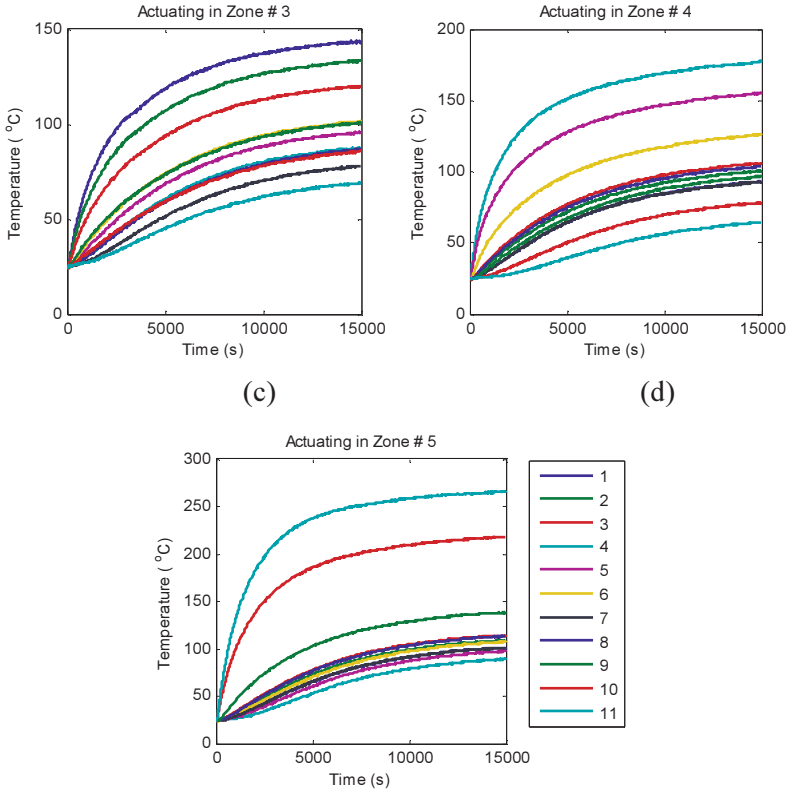
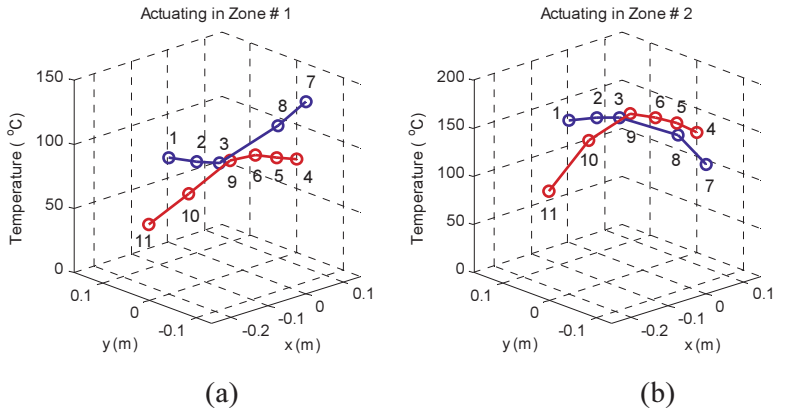


Fig. 9.3 Time responses of temperatures (Fig. 9.3 a-e), measured by thermocouples 1-11 at 25% step of the heating power in the individual zones of the mold.



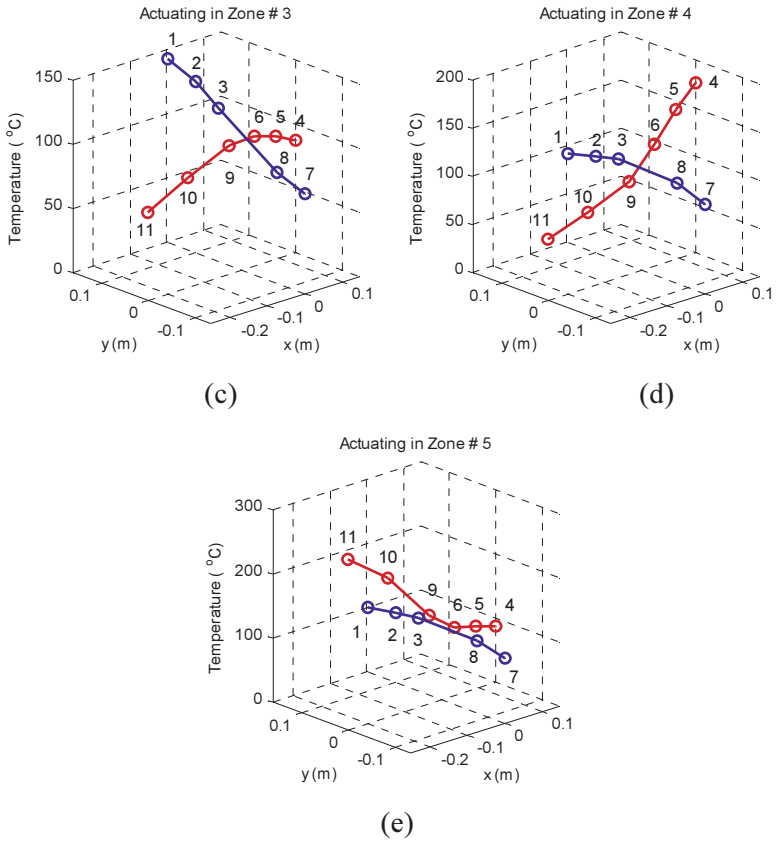


Fig. 9.4 Steady-state temperature profiles (Fig. 9.4 a-e), measured by thermocouples 1-11 at 25% step of the heating power in the individual zones of the mold.

Noisy temperature responses were smoothed using standard functions, such as *polyfit* and *polyval*. The above functions were implemented by translating the selected polynomial degrees through the measured values based on the criterion function, by the least squares method. Smooth dynamic characteristics were generated for the sampling period $T = 10$ s for the selected polynomial degrees. Distributed transient responses were further transformed by the standard Matlab function *mat2cell* into a cell array and stored in a data file.

Five transfer functions $\{S_i(\boldsymbol{x}_i, s)\}_{i=1,5}$ have been identified this way as a process model (8.24), the parameters of which are given in Table 9.1, whilst the highest gain response locations are identical with locations listed in Table 8.2. The discretization of transfer functions in the form of $\{SH_i(\boldsymbol{x}_i, z)\}_{i=1,5}$ was carried out with a sampling period $T = 10$ s.

No. of zone	1	2	3	4	5
K_p	53.40	58.16	49.36	62.56	97.87
T_{p1}	5881.68	4945.00	5489.58	5340.97	4578.71
T_{p2}	939.66	1958.20	1398.20	973.9	1214.73
T_z	3616.81	3301.80	3201.10	3195.90	3410.50

Table 9.1 Identified parameters of the real casting mold (25% step of the heat power).

9.3 Uncertainties in the Time and Space Domain

The HLDS block represents the dynamics of the controlled system based on a time-discrete space convolution (6.1). Here, discrete impulse characteristics $\{\mathcal{G}H_i(\boldsymbol{x},k)\}_{i=1,5}$ are determined according to Eq. (6.3). Distributed transient characteristics are measured at 11 points according to the locations shown in Fig. 8.17. The output of the block HLDS generates a time sequence of 11 transient responses to unit step inputs, representing the dynamics of the mold temperature in the corresponding positions. The selection of the characteristics is carried out by the block titled *Selector of output* in Fig. 9.5. This is done according to the analyzed area of Table 8.2. For example, the analyzed zone 1 is a response corresponding to the position of the measuring point no.7.

In the *DPS Space Synthesis* block, the spatial control synthesis task is solved as

$$\begin{aligned}&\min_{\breve{Y}_i}\left\| Y(\boldsymbol{x},k)-\sum_{i=1}^{n}Y_i(k)\,\mathcal{H}HR_i(\boldsymbol{x},\infty)\right\|=\\&=\left\| Y(\boldsymbol{x},k)-\sum_{i=1}^{n}\breve{Y}_i(k)\,\mathcal{H}HR_i(\boldsymbol{x},\infty)\right\|\end{aligned}\tag{9.15}$$

The approximated vector components $\{\breve{Y}_i(k)\}_{i=1,5}$ are analyzed depending on the number of zones, selected by the switch *Selector of approximation parameter*.

All zones were sequentially analyzed by this method to obtain approximations of distributed transient responses and lumped transient responses, near the locations of lumped inputs belonging to the transfer functions $\{SH_i(\boldsymbol{x}_i,z)\}_{i=1,5}$. The comparison of the approximating coefficients $\{\breve{Y}_i(k)\}_{i=1,5}$ on the level of distributed transient

characteristics and the responses of the lumped transient characteristics is shown in Fig. 9.6.

The difference in the dynamics of the approximation coefficients and lumped transient responses is time-dependent. Approximation error is minimal for $k \to \infty$ due to the uniqueness of the solution of the approximation problem in the quadratic norm on the set of reduced steady-state values of distributed transient responses $\left\{\mathcal{H}HR_i\left(\boldsymbol{x},\infty\right)\right\}_{i=1,5}$. The biggest difference is in the 2nd zone of the mold, what is from the physical point of view given by the geometry of the mold, placement of heaters and measuring points. The difference in dynamics may be considered as an uncertainty, and may be solved by robust control synthesis.

The identified transfer functions $\left\{S_i\left(\boldsymbol{x}_i,s\right)\right\}_i$ were further analyzed in terms of the uncertainty of their parameters, which was set via a change in percentage (%) as referred to the nominal parameters. For each zone, the file (*Family*) θ_i of transient responses was obtained, that offers a preview of the behavior of the system in the time domain at the given parameter uncertainty of the transfer function. An analysis was performed using the functions of the *Robust Control Toolbox*.

Using the function *ureal* it is possible to generate a range of uncertainties in 3 modes for the specified nominal parameter of the transfer function:

range	– specifies the interval changes of the nominal parameter.
percentage	– percentage deviation from the nominal parameter.
plusminus	– additive deviation from the nominal parameter.

A specified number of samples for a given mode can then be randomly generated using the function *usample,* or uniformly generated using the tool *gridureal.*

The graphical representation of the family of transient responses (*Samples*) for a set of variability parameters of transfer functions and nominal transient responses (*Nominal*) in Zone No. 1 are shown in Fig. 9.7. It is evident from the set of transient responses that the dominant influence on the dynamics of the transient response has a variability of the time constants Tp_1, Tp_2 in the denominator of the transfer function. The resulting uncertainty region in Fig. 9.6 is therefore feasible to be covered only by the variability of these parameters, for example: for zone no. 1, 3 and 5 of the intended variability given in Table 9.2.

The ranges of parameter changes Tp_1, Tp_2 (*Ranges*) are illustrated in Fig. 9.8, indicating the points corresponding to the nominal values *(Nominal).* The area of uncertainty, bounded by the course of the approximation coefficients *(Approximation)* and the nominal step response *(Nominal)*, is covered by the range of change *(Ranges)* parameters Tp_1, Tp_2.

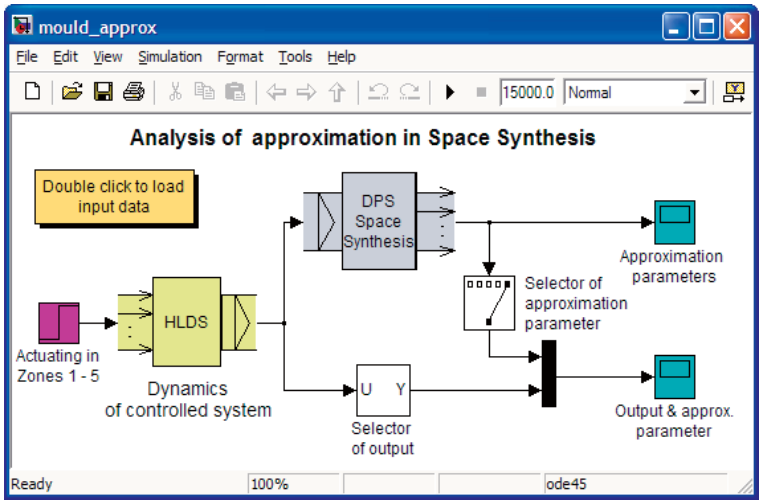


Fig. 9.5 Block scheme for the analysis of the approximation of the distributed transient responses and lumped transient responses.

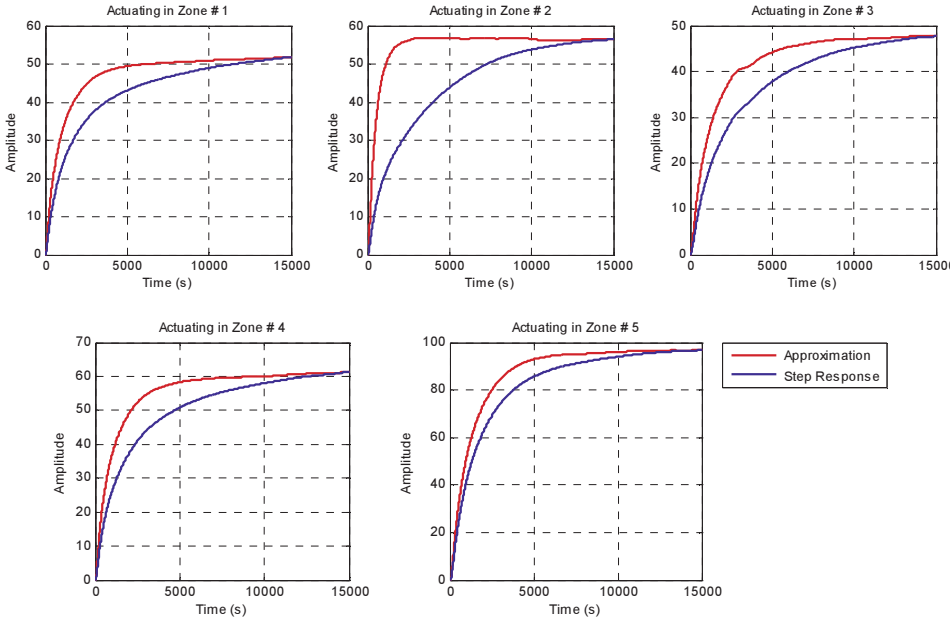


Fig. 9.6 Comparison of the approximation coefficients of distributed transient responses and lumped transient responses in different zones.

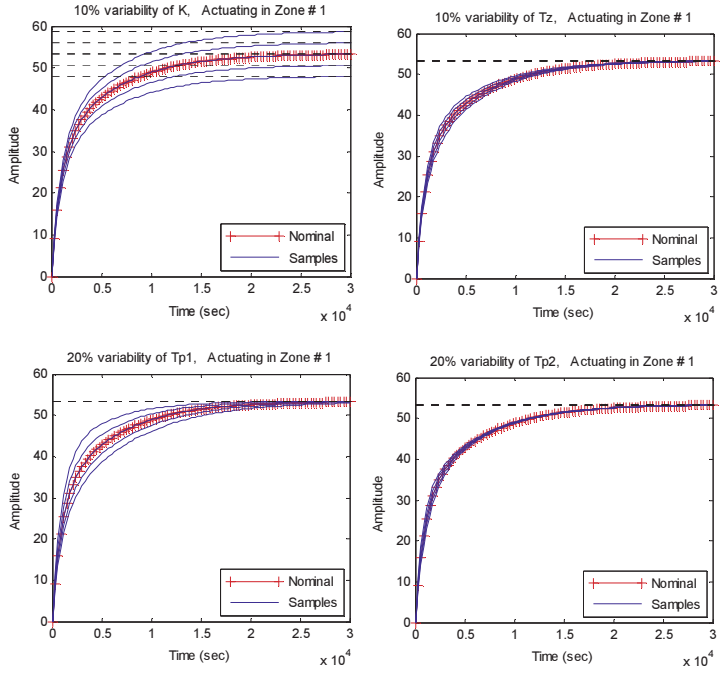


Fig. 9.7 Set (Family) of transient responses for variability parameters of the transfer function in Zone 1.

Zone	Parameter Tp1	Parameter Tp2
1	Name: 'Tp1' NominalValue: 5.8817e+003 Mode: 'Range' Range: [4.4113e+003 6.1758e+003] PlusMinus: [-1.4704e+003 294.084] Percentage: [-25 5.0000]	Name: 'Tp2' NominalValue: 939.6607 Mode: 'Range' Range: [751.7286 986.6438] PlusMinus: [-187.9321 46.9830] Percentage: [-20.0000 5.0000]
3	Name: 'Tp1' NominalValue: 5.4896e+003 Mode: 'Range' Range: [3.8427e+003 5.7641e+003] PlusMinus: [-1.6469e+003 274.478] Percentage: [-30.0000 5.0000]	Name: 'Tp2' NominalValue: 1.3982e+003 Mode: 'Range' Range: [1.0487e+003 1.4681e+003] PlusMinus: [-349.5500 69.9100] Percentage: [-25.0000 5.0000]

5	Name: 'Tp1' NominalValue: 4.5787e+003 Mode: 'Range' Range: [3.7545e+003 4.8076e+003] PlusMinus: [-824.1685 228.9357] Percentage: [-18.0000 5.0000]	Name: 'Tp2' NominalValue: 1.2147e+003 Mode: 'Range' Range: [1.0325e+003 1.2755e+003] PlusMinus: [-182.2101 60.7367] Percentage: [-15.0000 5.0000]

Table 9.2 Range of changes in parameters Tp_1, Tp_2 of the transfer function $S_i(\boldsymbol{x}_i, s)$ in Zone no. 1, 3 and 5.

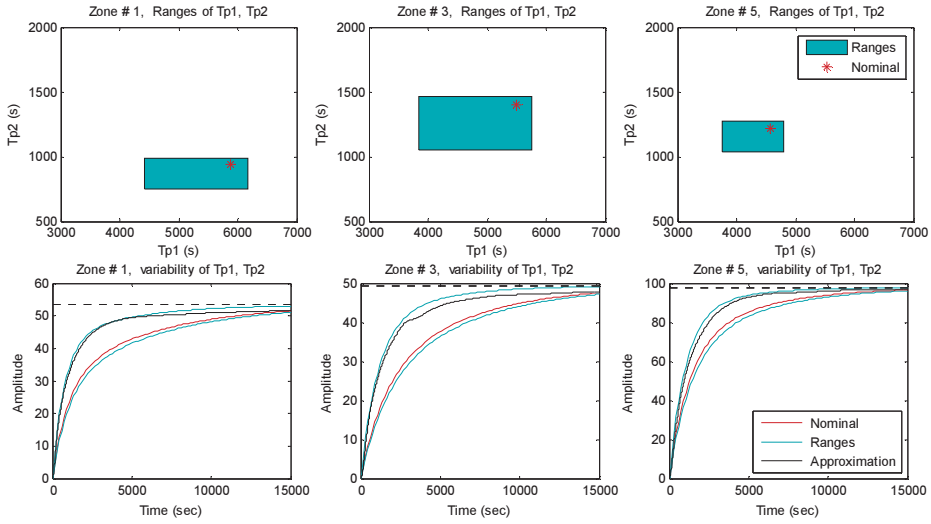


Fig. 9.8 Variability of the parameters Tp_1, Tp_2 in the denominator of the transfer functions $S_i(\boldsymbol{x}_i, s)$ and coverage of uncertainty region in Zone no. 1, 3 and 5.

9.4 Robust Control of the Mold Temperature Field

The proposed robust controllers and the simulation of the control of the temperature field in the casting mold undergoing a preheating process was implemented in the Matlab & Simulink and the DPS Blockset. At this stage, the control system was evaluated using SIL (Software-in-the-loop) simulation. Control methods are optimized and analyzed based on the model of the control process, respecting the required control performance criteria. At this development stage, a significant software support is given not only by the specialized libraries of Matlab & Simulink, but also by software products that take the specifics of the problem into account; such as the DPS Blockset for Matlab & Simulink in the area of distributed parameter control. The optimized control algorithms may be implemented in the real

process, based on the robust tuning procedure outlined above. A feedback loop was made with 5 discrete-time robust controllers $\left\{R_i^*(z)\right\}_{i=1,5}$ for the zone heating of the casting mold using the DPS approach, (Fig. 9.9).

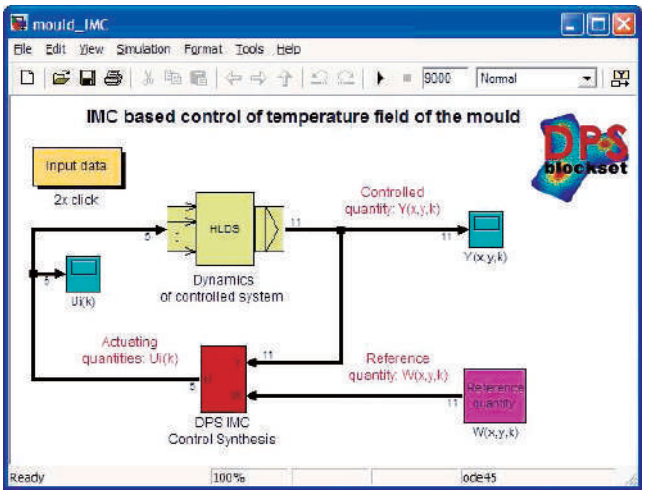


Fig. 9.9 Block scheme for the simulation of the control of the temperature field in the casting mold.

The internal structure of the *DPS IMC Control Synthesis* Simulink block is shown in Fig. 9.10. This includes solving a space synthesis for control, where the block *DPS Space Synthesis* solves the approximation task of the distributed output $\left\{Y(x_i,y_i,k)\right\}_{i=1,11}$ and the approximation task for the distributed reference $\left\{W(x_i,y_i,k)\right\}_{i=1,11}$; defined in 11 points of the casting mold body. The result of solving the approximation problems for the 5 control loops will be a set of approximation coefficients for the distributed manipulated variable $\left\{\breve{Y}_i(k)\right\}_{i=1,5}$ and vector of references $\left\{\breve{W}_i(k)\right\}_{i=1,5}$. The difference of these two, that is $\left\{\breve{E}_i(k)=\breve{W}_i(k)-\breve{Y}_i(k)\right\}_i$ will enter the block titled *Controllers based on IMC* of the temporal control synthesis as control error.

The internal structure of this block is shown in Fig. 9.11, where the discrete transfer functions of robust controllers $\left\{R_i^*(z)\right\}_{i=1,5}$ are expressed in the form of zeroes, poles and gain (ZPK representation), and are multiplied by the component $(1-\alpha)$ with the adjustable filter parameter α. The generated inputs are bounded in the block *Saturation* for the range (0-10). This boundary of inputs is introduced because of the limits on the control signal (0-10) V used in the physical model of the mold.

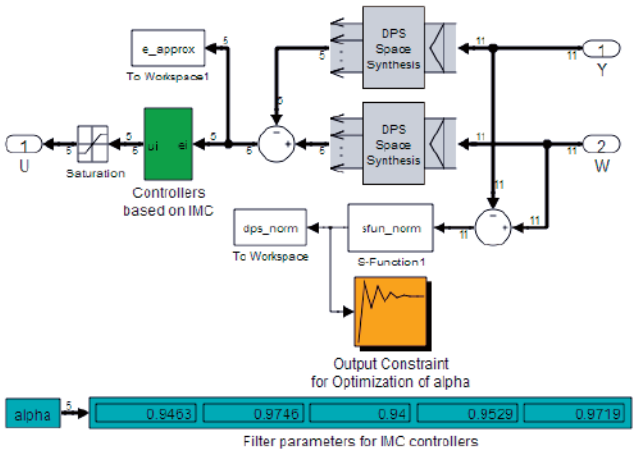


Fig. 9.10 The internal structure of the block robust control with space and time synthesis.

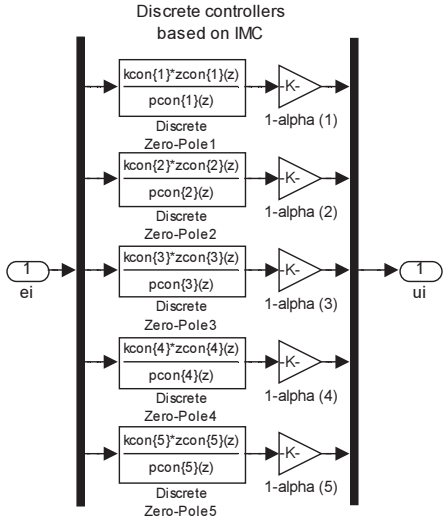


Fig. 9.11 The structure of the block containing discrete robust controllers.

Generally, the quality of control in systems with distributed parameters determines the amount of control error. This is expressed in a suitably chosen norm, for example the quadratic norm as

$$\left\|E(\boldsymbol{x},k)\right\| = \left\|W(\boldsymbol{x},k) - Y(\boldsymbol{x},k)\right\| \tag{9.16}$$

The quadratic norm of the control error $\left\|E(\boldsymbol{x},k)\right\|$ is an aggregate indicator of the control quality not only in the space dependency $\boldsymbol{x}$, but also in the time dependency k, which can be used to formulate the criterion function used for the optimization of the

controller parameters. For example, this can be expressed as the sum of the values of the quadratic norm of the control error for N steps, given by

$$J = \min_{\alpha_i} \sum_{k=0}^{N} \left\| W(\boldsymbol{x},k) - Y(\boldsymbol{x},k) \right\| \tag{9.17}$$

It is appropriate to include restrictive conditions in the optimization criterion, e.g. the amplitude of the time sequence of the quadratic norm of the control error. This is needed because, without restrictions, the control process may lead to an unacceptable course of inputs due to their inherent limitations.

Given the robust control problem solved by the SiL simulation method, it is useful to deal with the optimization of parameters $\{\alpha_i\}_{i=1,5}$ according to criterion (9.17).

The proper tool to solve this problem is the *Simulink Response Optimization Toolbox.* It is possible to formulate the optimization task, enter restrictive conditions, the choice of method and conditions of the solution. This is done by using the *Signal Constraint* block, which is meant to solve this task, expressed as *Output Constraint for Optimization of alpha*. Because the objective function is given by (9.17), this block has the internal structure of the block *IMC DPS Control Synthesis* (Fig. 9.10), this is connected to the output of the block *sfun_norm* computing the quadratic norm of the control error at each time step. Parameters are introduced for the optimization of $\{\alpha_i\}_{i=1,5}$ using the items *Optimization* and *Tuned parameters* inside the block *Output Constraint for Optimization of alpha.* These are allocated in the simulation as a 5 component vector called *alpha*.

The simulation *start time* 0 s, simulation *Stop time* 9000 s, step length *Variable-step* of the numerical solution was specified in *Simulation Options* and the default method for numerical solution *ode45 (Dormand Prince)* was also selected. The default parameter *auto* to determine the maximum step size *Maximum step size*, the minimum step size *Minimum Step size* and the initial step size *Initial Step size* was left at the default values. The gradient optimization method with a tolerance of 0.001 was chosen for the optimized parameters tolerance under *Optimization Options.* Furthermore, in order to limit the objective function, *Constraint tolerance* and *Function* tolerance was also set.

The gradient method is solved by the *fmincon.m* function, a component of the optimization library in the *Matlab Optimization Toolbox*. This function searches for the minimum of the scalar function of several quantities, generally nonlinear in the presence of constraints. It is therefore a nonlinear optimization problem with constraints or a nonlinear programming problem.

Furthermore, the option for finding a solution maximally compatible with the established limits was selected using the *Look for maximally feasible solution* option.

The dialog window of the block *Output Constraint for Optimization of alpha,* showing the upper and lower bounds for the time response of the quadratic norm of the control error $\|E(\boldsymbol{x},k)\|$ and the time response for each iteration of the optimization process from the initial values (blue curve) to find the optimal $\{\alpha_i\}_{i=1,5}$, (with a minimum amplitude of the curve) is shown in Fig. 9.12.

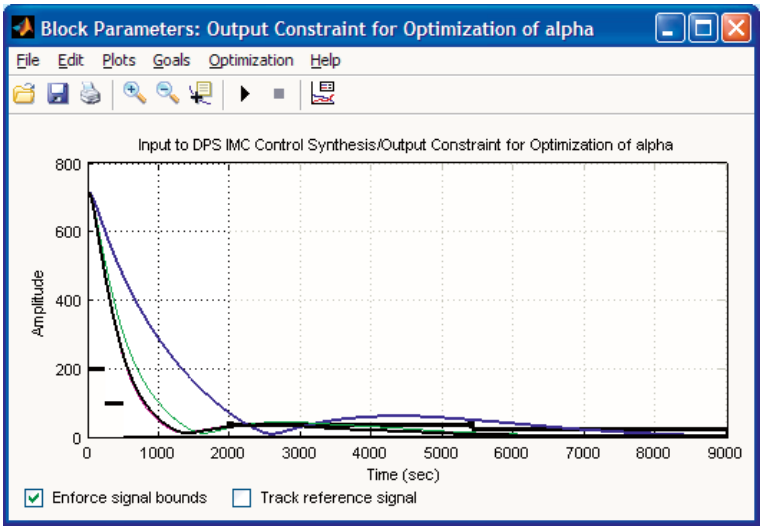


Fig. 9.12 Quadratic norm of the control error during the optimization process.

The optimization protocol of parameters $\{\alpha_i\}_{i=1,5}$ is also output during the procedure, and is shown in Fig. 9.13. In this case, the solution successfully converged and found the minimum of the objective function (9.17), a solution compatible with the established optimization conditions, including the limits on the time response of the quadratic norm of the control error. Optimal adjustable parameters of robust controllers $\{\alpha_i\}_{i=1,5}$ were found this way.

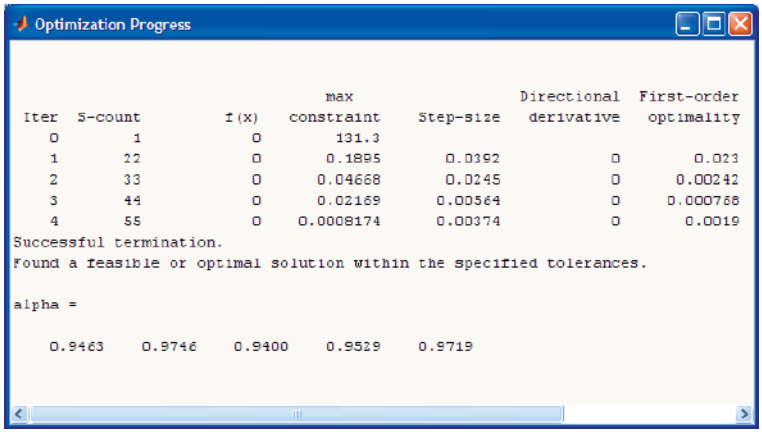


Fig. 9.13 Protocol of the optimization of parameters $\{\alpha_i\}_{i=1,5}$.

The simulation of the control process was performed for the distributed reference specified for the requirements of casting of aluminum alloys. The distributed reference is specified in 11 points, with the values given in Table 9.3 in relation to Fig. 8.18. The graphical representation of the distributed reference is shown in Fig. 9.14. The simulation results of control with optimum values $\{\alpha_i\}_{i=1,5}$ are shown in Fig. 9.15.

Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$W(\boldsymbol{x})$ (°C)	170	185	205	200	205	210	185	205	225	270	290

Table 9.3 Values of the distributed reference $W(\boldsymbol{x},\infty)$, $\boldsymbol{x}=\{x_i,y_i\}_{i=1,11}$.

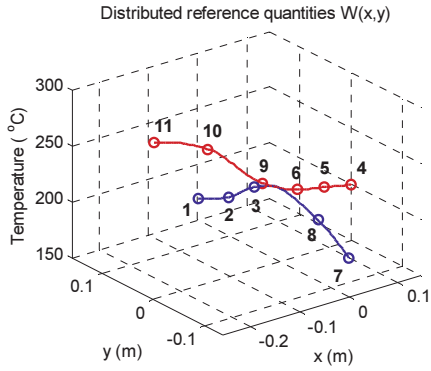


Fig. 9.14 Distributed reference $W(\boldsymbol{x},\infty)$, $\boldsymbol{x}=\{x_i,y_i\}_{i=1,11}$.

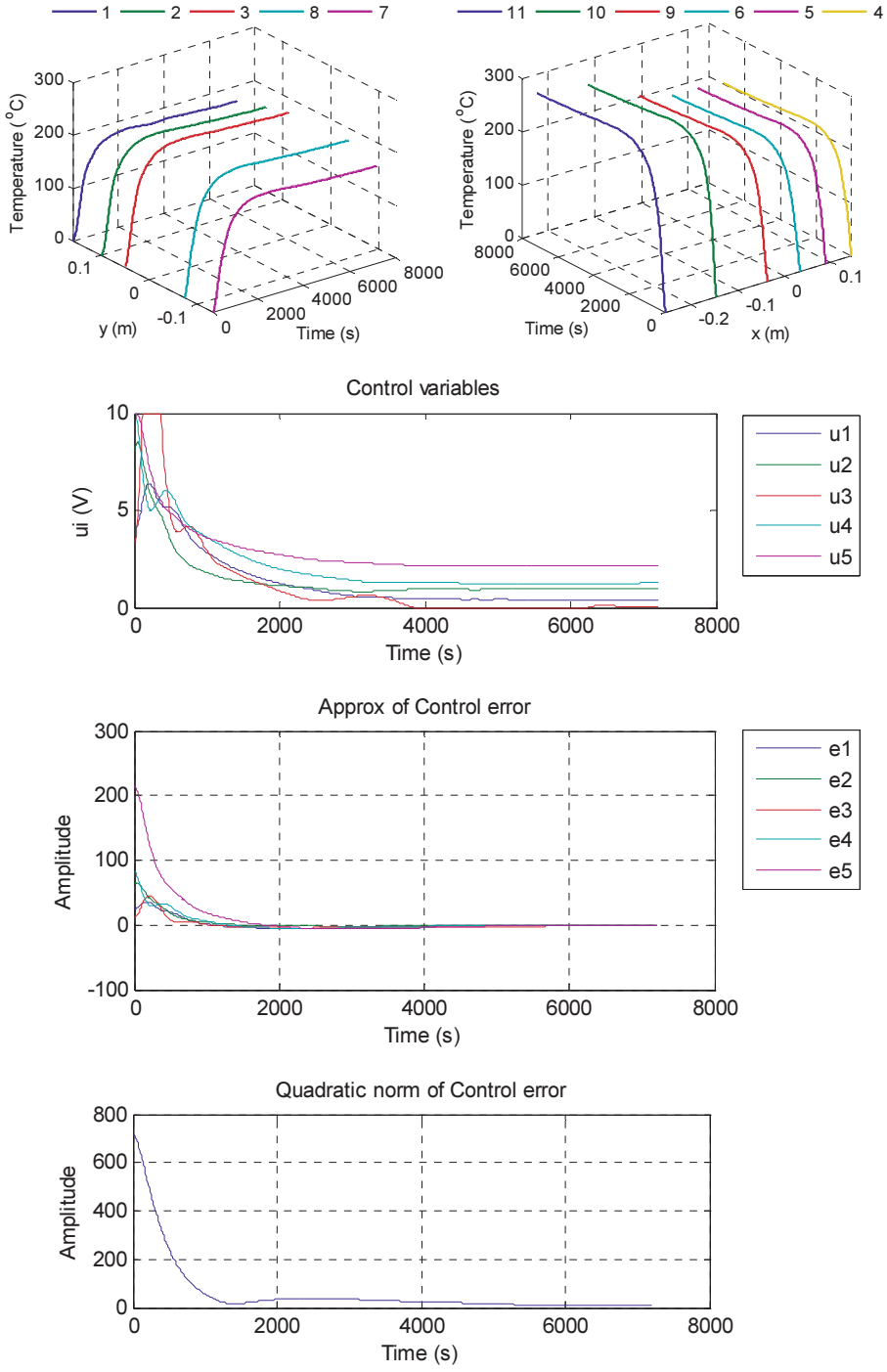


Fig. 9.15 Simulation of the control process: distributed output in locations 1-11, lumped inputs in zones 1-5, approximation of control error, quadratic norm of control error.

9.5 Control of the Temperature Fields on the Physical Model

After a simulation verification of the designed controllers, the next step in the development process was the implementation of the proposed solution to laboratory conditions. In this case, the problem is the control of the temperature field of the metal casting mold on a physical model of the controlled casting process. The control of the temperature fields in real-time was carried out by means of the scheme shown in Fig. 9.16.

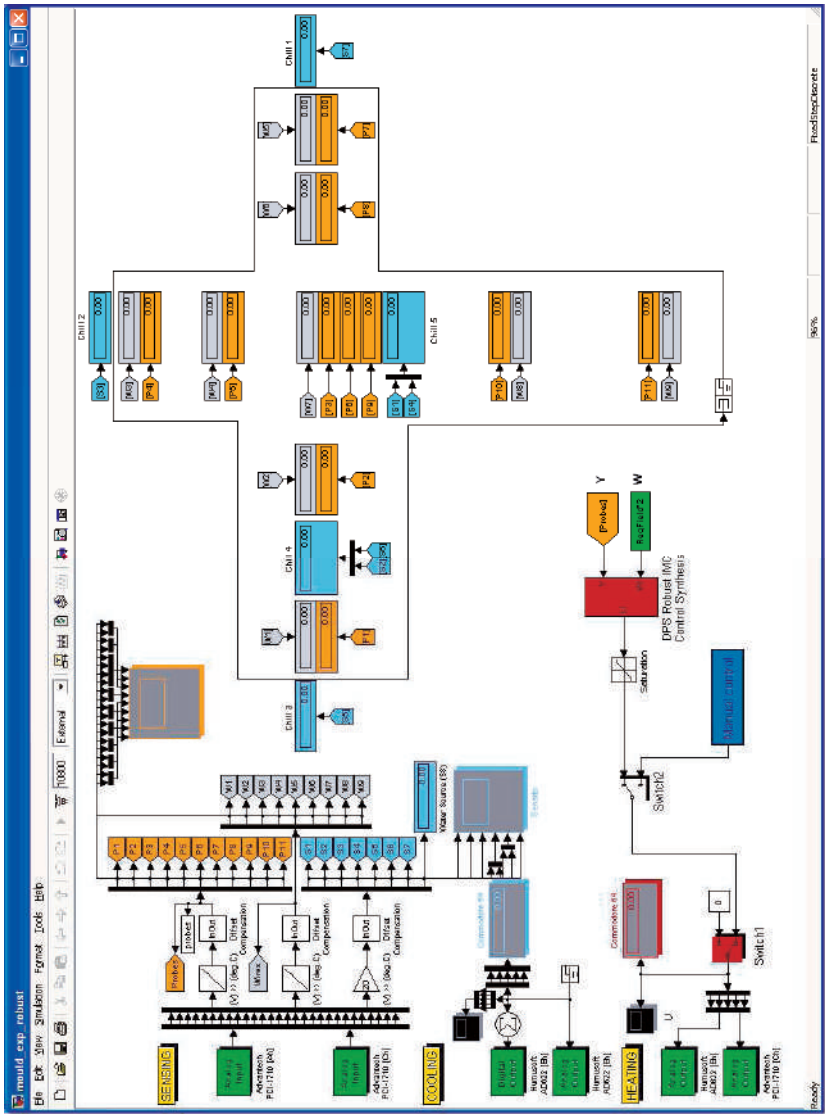


Fig. 9.16 Block scheme of the implementation of experiments

In control mode, *Switch1* is set to position 1, which is a zonal heating system HEATING, associated with the control system *DPS IMC Robust Control Synthesis*. The internal structure of this block is shown in Fig. 9.17.

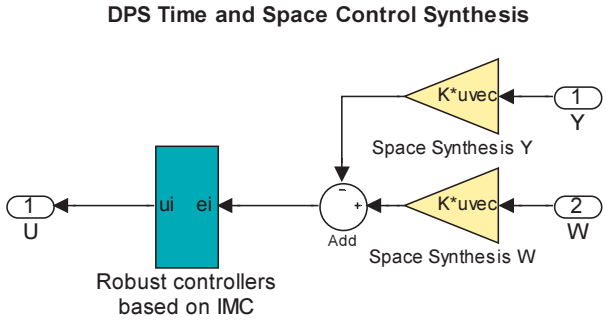


Fig. 9.17 Time and space control synthesis.

The laboratory control system includes the solution of the spatial synthesis in the block *Space Synthesis Y* (solving the approximation task for the distributed output $\{Y(x_i, y_i, k)\}_{i=1,11}$) and in the block *Space Synthesis W* solves the approximation task for the distributed reference $\{W(x_i, y_i, k)\}_{i=1,11}$ in accordance with the concept of DPS control based on LDS. The control error vector $\{\breve{E}_i(k) = \breve{W}_i(k) - \breve{Y}_i(k)\}_{i=1,5}$ is used as the input to the block of the time component of the control synthesis, realized within the *Robust Controllers based on IMC* block. The structure and parameters of robust controllers $\{R_i^*(z)\}_{i=1,5}$ are the same as in the simulation scheme. The results of the real-time robust control of the mold are shown in Fig. 9.18.

The controllers were optimized and validated by simulations in Matlab & Simulink using the DPS Blockset. The parameters of robust controllers based on IMC are established according to the identified discrete transfer functions in the form of zeros, poles and gain. The tuning parameter is optimized in the presence of bounds on the criterial function. Controllers have a simple structure and are easily implemented even in real-life conditions. The advantage of the DPS control design based on the principle of IMC is, that the single tuning parameter of the controller is a filter parameter that can be optimized using off-the-shelf software products.

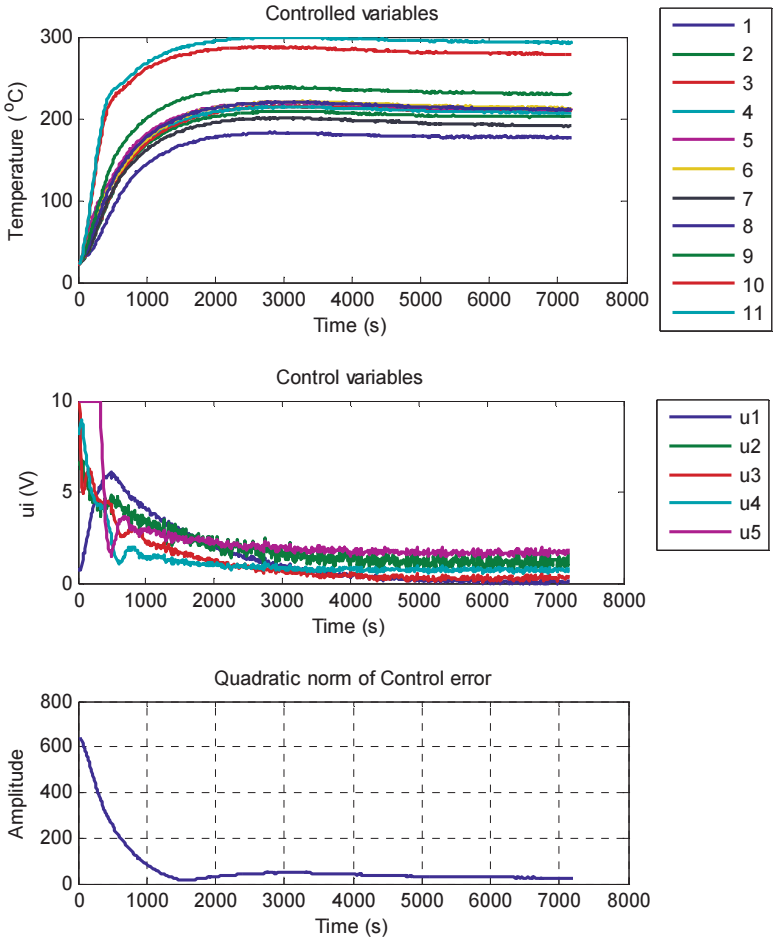


Fig. 9.18 Robust control of casting mold temperature field: distributed output in locations 1–11, lumped inputs in zones 1–5, quadratic norm of control error $\|E(x,k)\|$.

The analysis of the simulation results of the casting mold preheating process implies a high level of control performance, which is related to the optimized nature of the robust controllers. A similar conclusion holds for the case, where the control error is assumed in the sense of the simulation taken on the entire definition domain of the plan, for example over a profile similar to that shown in Fig. 8.18. The overall control quality is meanwhile not influenced by the solution of the approximation problem of the manipulated variable based on $\{\mathcal{H}HR_i(x,\infty)\}_{i=1,5}$ instead of $\{YR_i(x,k)\}_{i=1,5}$.

Similarly, the results of the control procedure on the physical model shows very good reproducibility of the proposed robust DPS control framework in laboratory conditions. It would be appropriate to use filters to smooth noisy signals in future work, as this is desirable for real-life sensors and industrial scenarios. The time

response of distributed output in 11 points of the casting mold is characterized by a short rise-time to the desired value, with practically negligible overshoot. This allows the reduction of the settling-time in the preheating phase and the synchronization of the various forms of technological requirements in the casting process. These results and the robust DPS control framework create excellent conditions for the shortening of the production cycle, cost reduction and quality improvements of the final product.

10. MODELING AND CONTROL OF THE INDUCTION HEATING PROCESS

Induction heating is a popular technology used for the heating of metallic materials, where electromagnetic fields generated by coils heat up metal semi-products. Since it is an energy-intensive technology, in order to make its usage more economical and efficient, an increased attention is paid to issues of the mathematical description of the process, its modelling and control. This mathematical description is predominantly given in the form of nonlinear partial differential equations. From the perspective of systems theory and control, these are distributed parameter systems.

The main advantages of induction heating are high reproducibility, high efficiency, very short heating time, temperature precision, good controllability, possibility of selective heating, controlling the heating depth and favorable environmental working conditions, (Sadegiphou (1996)).

Induction heating is a non-contact heating method, based on the principle of absorbing energy from an electromagnetic field generated by an inductor coil. Heating intensity and distribution depend on many physical factors and vary during the heating process. The majority of heat is produced by eddy currents. Heating by eddy currents works in all electrically conductive materials—magnetic or non-magnetic steels, stainless steel, copper, aluminium, graphite, etc. In recent years, modular multi-stage induction heaters with separated power supply circuits became more popular in the steel industry. Modular heaters are progressively replacing conventional heaters having one coil, powered by a single power supply.

In induction heating with a single coil, different materials and operating modes have their respective configuration tables. These are based on empiric knowledge but also on analytical computations. One of the examples is the operation of the heater made by the company Roboterm Inc. (Czech Republic) on the production line of the company HKS Forge Trnava Slovakia, co. ltd.

In the case of a modern modular multi-coil design, the process settings are established by the application of numerical simulation software. This offers better flexibility, temperature accuracy, controllability and energy efficiency, since each inductor module may be set to different power and frequency.

Induction heating is principally based on three fundamental physical phenomena—electromagnetic induction, skin effect and heat transfer. An electrically conductive material is inserted into a coil powered by alternating current. Alternating

current creates an electromagnetic field in the coil cavity. Electromagnetic energy starts to be absorbed by the material and is transformed into heat. As it has been mentioned previously, this phenomena can be mathematically described by partial differential equations. In our case, this provides the option to apply the control principles outlined in this book to induction heating processes, as described in Camber (2013), Kapusta (2013).

10.1 The Induction Heating Process

The principle of induction heating is based on the magnetic induction of current to a material to be heated, which is moving through the inductor coil. The current induced to the material has the same frequency, but opposite direction of the current flowing through the inductor coil. The induced current (also called an eddy current) flowing through the material is generating the heat due to the losses caused by the electrical resistance of heated material. In case of a ferromagnetic material, the hysteresis losses are also participating in the heating process of the material.

The energy from the electromagnetic field instigates movement in the free electrons inside the material structure. The free electrons boosted by the electromagnetic field achieve high speed on a very short distance, giving them a significant kinetic energy. The energy is transmitted to atoms during the collision between moving electrons and stationary atoms. This energy transmission increases the vibration of the material structure, which results in a rapid increase of heat inside the material. In short, the energy of the electromagnetic field is transferred into heat.

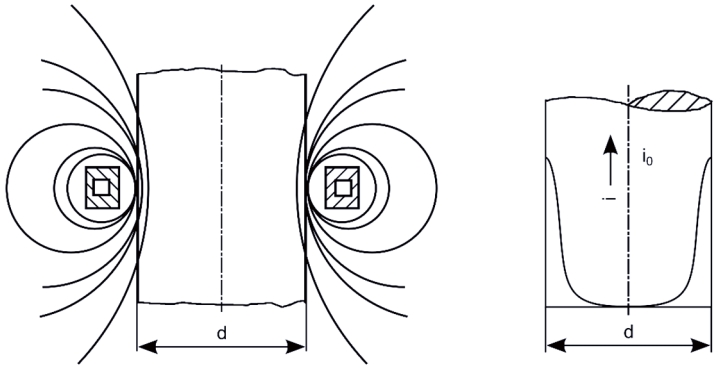


Fig. 10.1 Schematic diagram of induction heating and radial distribution of current density along the cross-section of a cylinder, (Langer (1979)).

The density of the high frequency eddy current is not uniform through the cross-section of the material. Maximal current density is located on the material surface and it is decreasing in the direction of the material core, (Fig. 10.1).

10.1.1 The Skin Effect

Induction heating is a result of several physical phenomena. The most important of them is a surface phenomena—the skin effect, where the current leaks to the surface of a heated material, which is an electrical conductor. The current flows through a fraction of the cross-section of the conductor. The current density exponentially decreases towards the center. This is influenced by frequency, specific electric resistivity and relative permeability. The skin effect also affects the coils of the inductor device, where it should be eliminated as it is an undesirable phenomena. The steady loss of the current density in the direction of the center of the conductor is known as *penetration depth* δ .

Penetration depth is the depth where the current density decreases to the value $1/e \approx 0.368$ (where e is Euler's number) of the current density present in the surface layer of a heated material, (Fig. 10.2). About 86% of the total induced energy to the conductor is transformed into heat. The rest heats the deeper layers.

Conducting materials are strongly ferromagnetic at lower temperatures and heat up quickly. After they cross the Curie point, the magnetic permeability of the conductor drops to $\mu = 1$ and specific electric resistivity increases, also increasing the penetration depth, (Davies (1990)).

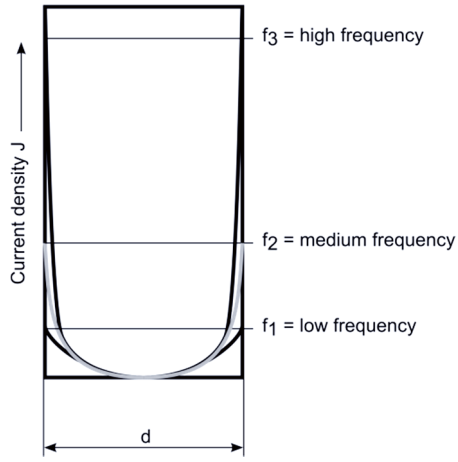


Fig. 10.2 Dependence of current density on the frequency in the conductor.

The penetration depth of the current is given by

$$\delta = \sqrt{\frac{\rho}{\pi.\mu.f}} \tag{10.1}$$

where f is current frequency, ρ is specific resistivity a μ is magnetic permeability. Eq. (10.1) shows that, with increasing frequency, the penetration depth decreases and the generated heat will be concentrated in the surface layers.

The concentration of heat in the surface layers (only) can overheat edges and thin cross-sections, which causes embrittlement, deformations, cracks or even fractures, (Sadegiphour (1996)).

The specific electric resistivity and permeability change with temperature. This means that penetration depth δ is a function of the temperature and the frequency of the current in the heated material, (Fig. 10.3 and Fig. 10.4).

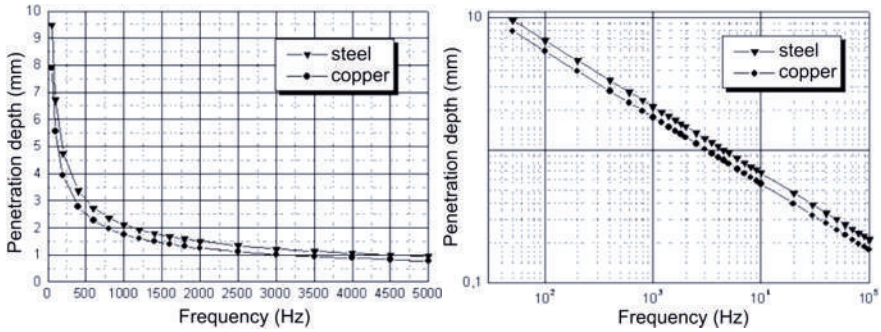


Fig. 10.3 Penetration depth dependence of the frequency for the steel and copper at 20 °C, (Langer (1979)).

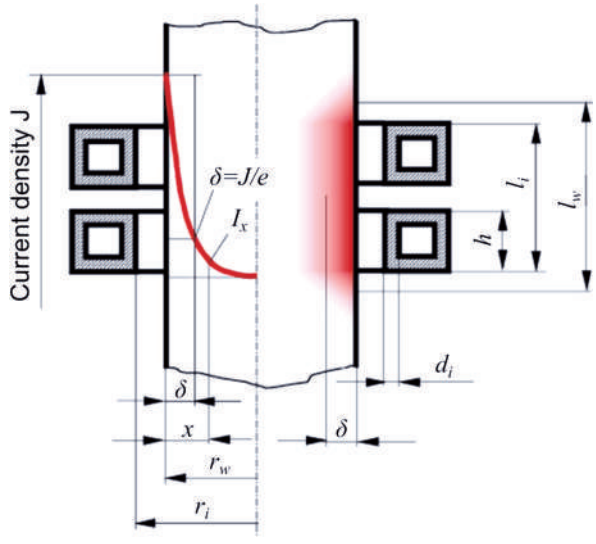


Fig. 10.4 Distribution of the current density affected by skin effect.

10.1.2 Electromagnetic Energy of Heating

Differential equations describing the electromagnetic radiation in a magnetic and/or polarizable material are derived from Maxwell's equations

$$\mathrm{rot}\,\boldsymbol{H} = \boldsymbol{J} + \varepsilon_0\varepsilon_\mathrm{r}\frac{\partial \boldsymbol{E}}{\partial t} \tag{10.2}$$

$$\mathrm{rot}\,\boldsymbol{E} = -\mu_0\mu_\mathrm{r}\frac{\partial \boldsymbol{H}}{\partial t} \tag{10.3}$$

$$\mathrm{div}\,\varepsilon_0\varepsilon_\mathrm{r}\boldsymbol{E} = \rho \tag{10.4}$$

$$\mathrm{div}\,\mu_0\mu_\mathrm{r}\,\boldsymbol{H} = 0 \tag{10.5}$$

where $\varepsilon_0 = 8.854...\times10^{-12}$ F m^{-1} is the physical coefficient for the permittivity of vacuum and $\mu_0 = 4\pi\times10^{-7}$ H m^{-1} expresses the permeability of vacuum. Relative permittivity $\varepsilon_\mathrm{r} = \varepsilon/\varepsilon_0$ is a dimensionless number, expressing the relation between the permittivity ε of the material and vacuum. The notation assumed for the volume charge density is ρ and current density is $\boldsymbol{J}$.

The Poynting vector S (W m^{-2}) determines the direction and density of electromagnetic energy in the space. This vector is given by the intensity of electric field $\boldsymbol{E}$ and the intensity of magnetic field $\boldsymbol{H}$ in an arbitrary point of the Euclidean space.

In a volume V, the amount of magnetic energy W_m and electric energy W_e are given by the equations

$$W_m = \frac{1}{2}\int_0^V \boldsymbol{BH}\,dV = \frac{1}{2}\mu_0\mu\int_0^V \boldsymbol{H}^2 dV \tag{10.6}$$

$$W_e = \frac{1}{2}\int_0^V \boldsymbol{DE}\,dV = \frac{1}{2}\varepsilon_0\varepsilon\int_0^V \boldsymbol{E}^2 dV \tag{10.7}$$

Vectors $\boldsymbol{B}$ and $\boldsymbol{D}$ denote magnetic and electric induction.

The density of the electromagnetic energy at each point is time dependent. The rate of energy loss in volume V per unit time is

$$-\frac{\partial}{\partial t}(W_{\mathrm{e}}+W_{\mathrm{m}})=-\int_0^V\frac{\partial}{\partial t}\left\{\frac{1}{2}\varepsilon_0\varepsilon\boldsymbol{E}^2+\frac{1}{2}\mu_0\mu\boldsymbol{H}^2\right\}dV \tag{10.8}$$

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\varepsilon_0\varepsilon\boldsymbol{E}^2\right)=\frac{1}{2}\varepsilon_0\varepsilon 2\boldsymbol{E}\frac{\partial\boldsymbol{E}}{\partial t}=\varepsilon_0\varepsilon\boldsymbol{E}\frac{\partial\boldsymbol{E}}{\partial t} \tag{10.9}$$

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\mu_0\mu\boldsymbol{H}^2\right)=\mu_0\mu\boldsymbol{H}\frac{\partial\boldsymbol{H}}{\partial t} \tag{10.10}$$

$$-\frac{\partial}{\partial t}\left(W_{\mathrm{e}}+W_{\mathrm{m}}\right)=-\int_0^V\left\{\varepsilon_0\varepsilon\boldsymbol{E}\frac{\partial\boldsymbol{E}}{\partial t}+\mu_0\mu\boldsymbol{H}\frac{\partial\boldsymbol{H}}{\partial t}\right\}dV \tag{10.11}$$

The equations above are used to derive Maxwell's first and second equations

$$\text{rot }\boldsymbol{H}=\nabla\times\boldsymbol{H}=\gamma\boldsymbol{E}+\varepsilon_0\varepsilon\frac{\partial\boldsymbol{E}}{\partial t};\ \varepsilon_0\varepsilon\frac{\partial\boldsymbol{E}}{\partial t}=\nabla\times\boldsymbol{H}-\gamma\boldsymbol{E} \tag{10.12}$$

$$\text{rot }\boldsymbol{E}=\nabla\times\boldsymbol{E}=-\mu_0\mu\frac{\partial\boldsymbol{H}}{\partial t};\ \mu_0\mu\frac{\partial\boldsymbol{H}}{\partial t}=-\nabla\times\boldsymbol{E} \tag{10.13}$$

Substituting these equations into equation (10.11) we get

$$\begin{aligned}-\frac{\partial}{\partial t}(W_{\mathrm{e}}+W_{\mathrm{m}})&=-\int_0^V\{(\nabla\times\boldsymbol{H}-\gamma\boldsymbol{E})\boldsymbol{E}-((\nabla\times\boldsymbol{E})\boldsymbol{H})\}dV=\\&=\int_0^V\{\gamma\boldsymbol{E}^2+((\nabla\times\boldsymbol{E})\boldsymbol{H})-((\nabla\times\boldsymbol{H})\boldsymbol{E})\}dV\end{aligned} \tag{10.14}$$

The second and third term in the brackets represents the dot product of vectors. Because the commutative law applies here, the sequence of terms is arbitrary. Relation (10.14) can be rewritten as

$$-\frac{\partial}{\partial t}(W_{\mathrm{e}}+W_{\mathrm{m}})=\int_0^V\{\gamma\boldsymbol{E}^2+(\boldsymbol{H}(\nabla\times\boldsymbol{E}))-(\boldsymbol{E}(\nabla\times\boldsymbol{H}))\}dV \tag{10.15}$$

Since the: $(\boldsymbol{H}(\nabla\times\boldsymbol{E}))-(\boldsymbol{E}(\nabla\times\boldsymbol{H}))=(\nabla(\boldsymbol{E}\times\boldsymbol{H}))=\mathrm{div}\,(\boldsymbol{E}\times\boldsymbol{H})$, then

$$-\frac{\partial}{\partial t}(W_{\mathrm{e}}+W_{\mathrm{m}})=\int_0^V\{\gamma\boldsymbol{E}^2+\mathrm{div}\,(\boldsymbol{E}\times\boldsymbol{H})\}dV \tag{10.16}$$

According to the Gauss-Ostrogradsky theorem, the volume integral is substituted by the surface integral

$$\int_0^V\mathrm{div}\,(\boldsymbol{E}\times\boldsymbol{H})dV=\oint_F(\boldsymbol{E}\times\boldsymbol{H})dF \tag{10.17}$$

$$-\frac{\partial}{\partial t}(W_{\mathrm{e}}+W_{\mathrm{m}})=\int_0^V\gamma\boldsymbol{E}^2dV+\oint_F(\boldsymbol{E}\times\boldsymbol{H})dF=\int_0^V\gamma\boldsymbol{E}^2dV+\oint_F\boldsymbol{S}dF \tag{10.18}$$

Equation (10.18) determines the loss of electromagnetic energy in the volume V in a unit of time. The first term on the right side represents the amount of electromagnetic energy in this volume, which is transferred into heat in one second. The second term describes the amount of electromagnetic energy leaving a volume unit into the surroundings (diverging) in one second. The amount of energy which diverges through an area of 1 cm^2 is the Poyting vector, (Langer (1979))

$$\boldsymbol{S}=\boldsymbol{E}\times\boldsymbol{H} \tag{10.19}$$

Based on the above discussion, one can see how electromagnetic radiation is related to the electromagnetic flux in space. The energy flux is in the direction of the vector $\boldsymbol{S}$.

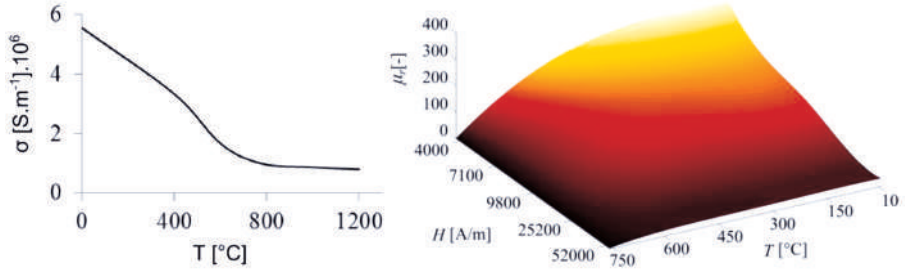
10.2 Dynamics of the Continuous Heating of Steel Semi-products

The problem of modelling the induction heating of steel semi-products is a multi-physics issue, which combines electromagnetism and heat transfer phenomena. The alternating magnetic field around coils induces eddy currents in ferromagnetic materials (steel semi-products) which generates Joule heating. This type of heating is governed by a system of two partial differential equations. The first one (Eq. (10.20)) governs the electromagnetic part, while the second one (Eq. (10.21)) governs heat transfer mechanisms (Rudnev (2003))

$$\left(j\omega\sigma(T)-\omega^2\varepsilon_0\varepsilon_{\mathrm{r}}\right)\boldsymbol{A}+\nabla\times\left(\mu_0^{-1}\mu_{\mathrm{r}}^{-1}(H,T)\boldsymbol{B}\right)-\sigma(T)\times\boldsymbol{B}=\boldsymbol{J} \tag{10.20}$$

where $\boldsymbol{A}$ represents the vector of magnetic potential, $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ magnetic induction (T), ω angular frequency, ε_0 vacuum permittivity, ε_r relative permittivity, $\boldsymbol{v}$ (m.s^{-1}) speed and $\boldsymbol{J}$ current density in the coil.

Electrical conductivity $\sigma = 1/\rho$ (S m^{-1}) is a temperature dependent material property, where ρ (Ω m) represents the specific electric resistivity, see Fig. 10.5. Permeability is a function of temperature and electrical intensity $\boldsymbol{H}$ (A m^{-1}). For a standard structural steel with the Curie Temperature T_C=775 °C the relative permeability is $\mu_r = 1$, (Davies (1990), Rudnev (2003)).



(a) (b)

Fig. 10.5 (a) electrical conductivity, (b) relative permeability of a standard structural steel as the function of temperature.

All three heat transfer mechanisms are present during induction heating: thermal conduction, convection and radiation. The temperature field in the semi-product during induction heating is governed by the Fourier-Kirchhoff equation

$$\rho(T)c_p(T)\left(\frac{\partial T}{\partial t} + \boldsymbol{v}\cdot\nabla T\right) - \nabla\cdot\left(\lambda(T)\nabla T\right) = Q_j \tag{10.21}$$

where the temperature dependent variables are: density, thermal conductivity and specific heat capacity, (see Sect. *8.1.1*). The Joule losses Q_j generated in the semi-product by eddy currents are given by equation

$$Q_j = \frac{\|\boldsymbol{J}_{in}\|^2}{\sigma(T)} \tag{10.22}$$

where $\boldsymbol{J}_{in}$ is the induced current density in the heated semi-product. The minimization of heat losses in the inductor is achieved with a refractory concrete lining. The thermal losses are governed by $q_{loss} = h\left(T - T_{amb}\right)$ (W m^{-2}), where T_{amb} is the ambient temperature and T is the surface temperature of the heated semi-product. The temperature dependent heat transfer coefficient h combines heat losses through thermal

radiation and natural convection. The temperature dependent material properties of the semi-product and the heat transfer coefficient between the semi-product and surroundings are depicted in Fig. 10.6 a-d.

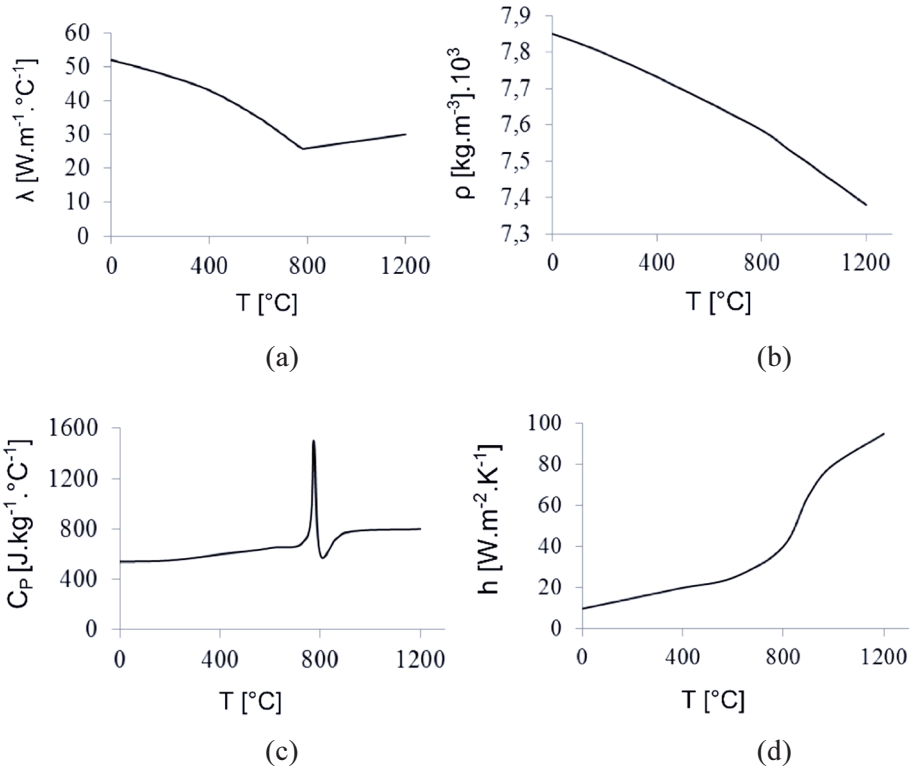
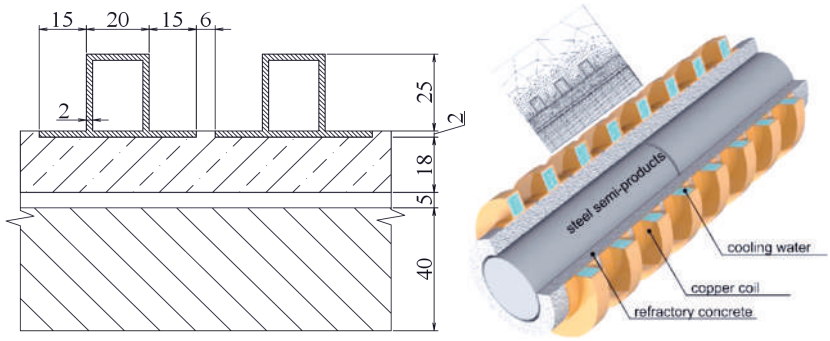


Fig. 10.6 (a) thermal conductivity, (b) density, (c) specific heat capacity of a standard construction steel; (d) temperature dependent heat-transfer coefficient.

10.3 Model in the COMSOL Multiphysics Software Environment

Based on the discussion above, the preheating of round (80 mm diameter) cross-section semi-products for forging is solved in the following discussion. The use of a segmented industrial inductor is assumed, where each segment is powered by individual currents of I_1 – I_4 (kA) (length 3 m, 55 turns of wire divided into 4 segments). The forging input temperature is approx. 300 K and (exit) temperature is 1450 K. The structural layout is given in Figs. 10.7 a, b.



(a) (b)

Fig. 10.7 (a) main dimensions (mm) of the inductor, (b) CAD model and FEM mesh detail.

Due to the skin effect discussed previously, a major part of the electromagnetic field passes through the surface of the heated semi-product. Therefore, a combination of a mapped and free mesh was utilized to model the system, Fig. 10.7 b. The final computation time is set to 800 s, which is the approximate settling time of the temperature field. Forging temperature at the exit of the coil was achieved in approx. 400 s, which meets the requirements for the industrial forging of steel semi-products. The steady-state temperature field of the steel semi-product achieved along the inductor coils is depicted in Fig. 10.8.

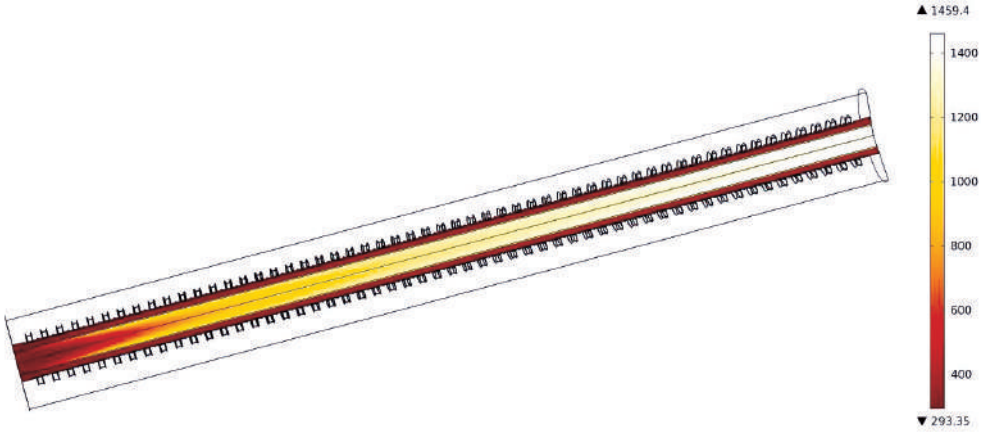


Fig. 10.8 Steady-state temperature profile of the steel semi product being heated in the inductor.

Four measurement points were selected according to the steady-state temperature field, each at the ends of the coils, where pyrometers can be installed. Also, 10 points were selected on the computational mesh, which will be used for the description of the dynamics of the heating process in the space domain.

10.4 Semi-product in the Segmented Induction Heater as DPS

Each coil in the segmented inductor $\{GU_i\}_i$, Fig. 10.9 is powered by a current $\{UNA_i(t)\}_i$, generated by actuators $\{SA_i\}_i$, causing a change of the temperature profile $YN(x,t)$ along the inductor coils. In terms of systems and control theory, this is a typical system with lumped input and distributed output (Hulkó (1998, 2009, 2011)). The output voltage of the source or input voltage of the coils is controlled by a programmable logic controller. The magnitude of currents that flow through the individual coils is determined by input voltages and resistances. These currents generate electromagnetic quantities. The model assumes the use of currents in individual coils, and these currents represent the lumped inputs of the controlled distributed parameter system, Fig. 10.9.

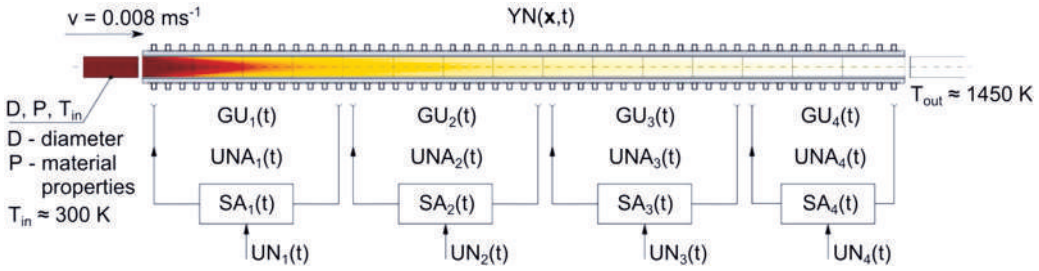


Fig. 10.9 Segmented 4 - coil inductor as a lumped input and distributed output system.

Let us consider single discrete step changes of lumped inputs in a linearized neighborhood of an operating mode of the inductor given by $\{UN_i(\infty)\}_i$ and $YN(x,\infty)$. Although these step changes are acting on the individual segments but they essentially effect the whole inductor resulting in the discrete distributed transient characteristics $\{\mathcal{H}H_i(x,k)\}_{i=1,4}$ of the inductor defined by the HLDS controlled system. We get a set of distributed impulse characteristics $\{\mathcal{G}H_i(x,k)\}_{i=1,4}$ by subtracting mutually shifted distributed transient characteristics, which describes the input-output relationships for the modelling of the dynamics by the HLDS block

$$Y(x,k)=\sum_{i=1}^{4}Y_i(x,k)=\sum_{i=1}^{4}\mathcal{G}H_i(x,k)\oplus U_i(k) \tag{10.23}$$

where $\oplus$ denotes discrete convolution and $\{Y_i(\boldsymbol{x},k)\}_i$ is a partial distributed output of the HLDS block.

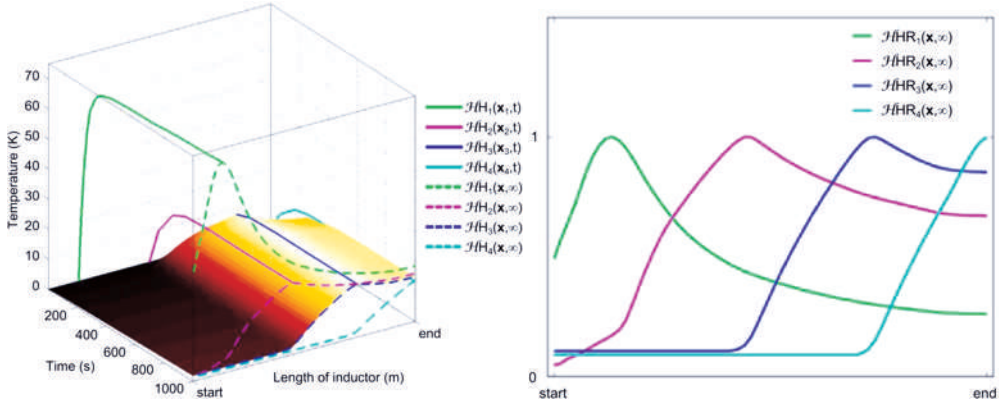
Let us analyze the distributed transient characteristics in terms of Chapt. *3. Analysis and decomposition of dynamics.* Let us select the partial responses with maximum amplitudes in the steady-state $\{\mathcal{H}H_i(\boldsymbol{x}_i,k)\}_{i=1,4}$ in the time domain; with points $\{\boldsymbol{x}_i\}_{i=1,4} \subseteq \Omega$ on the definition domain of the analyzed system, Fig. 10.10 a. Let us assign discrete transfer functions $\{SH_i(\boldsymbol{x}_i,z)\}_{i=1,4}$ to these partial transient characteristics. We may express the following relation in locations $\{\boldsymbol{x}_i\}_{i=1,4}$

$$\{Y_i(\boldsymbol{x}_i,z) = SH_i(\boldsymbol{x}_i,z)U_i(z)\}_{i=1,4} \tag{10.24}$$

between the lumped inputs $\{U_i(z)\}_{i=1,4}$ and the partial distributed outputs. Let us assign the reduced formulations to the distributed transient characteristics in the steady-state $\{\mathcal{H}H_i(\boldsymbol{x},\infty)\}_{i=1,4}$ by

$$\left\{\mathcal{H}HR_i(\boldsymbol{x},\infty) = \frac{\mathcal{H}H_i(\boldsymbol{x},\infty)}{\mathcal{H}H_i(\boldsymbol{x}_i,\infty)}\right\}_{i=1,4} \tag{10.25}$$

where $\{\mathcal{H}H_i(\boldsymbol{x}_i,\infty) \neq 0\}_{i=1,4}$, Fig. 10.10 b.



(a) (b)

Fig. 10.10 (a) distributed transient characteristics of the heater as HLDS b) reduced distributed transient characteristics - the space components of dynamics.

start-end (m) - overall length of the inductor.

0-1000 (s) - settling time interval of temperature field.

$\{\mathcal{H}H_i(\boldsymbol{x},t)\}_i$ - transient characteristics of temperature field.

$\{\mathcal{H}H_i(\boldsymbol{x}_i,t)\}_i$ - partial transient characteristics in locations $\{\boldsymbol{x}_i\}_i$.

Temperature (K) - changes of temperature field relative to the profile in the chosen steady-state mode.

The result is the decomposition of the dynamics of the controlled system in the neighborhood of the operating mode into:

time components

$\{SH_i(\boldsymbol{x}_i,z)\}_{i=1,4}$ – for given i and selected $\boldsymbol{x}_i$ – variable z, and

space components

$\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_{i=1,4}$ – for given i, $k\to\infty$ – variable $\boldsymbol{x}$.

The steady-state distributed output is given by the formerly introduced characteristics using

$$Y(\boldsymbol{x},\infty)=\sum_{i=1}^{4}Y_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{x},\infty) \tag{10.26}$$

10.5 Control of the Temperature Field During Induction Heating

Let us create a distributed parameter control system shown in Fig. 10.11, based on the feedback loop in Fig. 7.8 for the induction heating process in the linearized neighborhood of the selected operating mode given by $\{UN_i(\infty)\}_i$, $YN(\boldsymbol{x},\infty)$. The dynamics of the induction heating process is given by Eq. (10.23). The control task is to set a prescribed temperature profile for the heated semi products on the level $YN(\boldsymbol{x},\infty)+W(\boldsymbol{x},\infty)$, in the neighborhood of a given steady-state operating mode. In practice, this means to achieve $YN(\boldsymbol{x},\infty)+\breve{Y}(\boldsymbol{x},\infty)$, where $\breve{Y}(\boldsymbol{x},\infty)$ is an optimal approximation of $W(\boldsymbol{x},\infty)$ in steady-state, by using the reduced steady-state transient characteristics in the neighborhood of the selected operating mode.

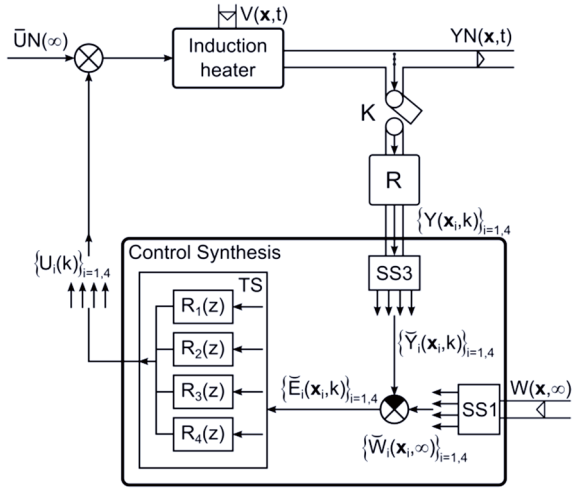


Fig. 10.11 Block diagram of the distributed parameter control system.

Induction heater - controlled system with lumped input and distributed output with zero order hold units (HLDS).

TS - time synthesis of the control problem.

SS1 - space synthesis of the control problem.

SS3 - transformation of vectors.

K - sampling.

R - block of differences and reconstruction.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_{i=1,4}$ - components of the partial control error vector.

$\left\{R_i\left(z\right)\right\}_{i=1,4}$ - Z- transfer functions of the controllers.

$\left\{U_i\left(k\right)\right\}_{i=1,4}$ - components of the lumped discrete input vector.

$\bar{U}N\left(\infty\right)$ - vector of steady-state inputs.

$V\left(\boldsymbol{x},t\right)$ - continuous distributed disturbance.

$W\left(\boldsymbol{x},\infty\right)$ - overall distributed reference in steady-state.

$\left\{\breve{W}_i\left(\boldsymbol{x}_i,\infty\right)\right\}_{i=1,4}$ - components of the vector of the discrete lumped reference in the steady-state.

$YN\left(\boldsymbol{x},t\right)$ - distributed manipulated variable.

$\left\{Y\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of measured manipulated variables on the HLDS level as output of block R.

$\left\{\breve{Y}_i\left(\boldsymbol{x}_i,k\right)\right\}_{i=1,4}$ - components of the partial distributed manipulated variable vector.

The SS1 block solves the approximation problem

$$\min_{W_i}\left\|W(\boldsymbol{x},\infty)-\sum_{i=1}^{4}W_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{x},\infty)\right\|= \\ =\left\|W(\boldsymbol{x},\infty)-\sum_{i=1}^{4}\breve{W}_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{x},\infty)\right\| \tag{10.27}$$

on the set of approximating functions of the distributed output in a quadratic norm. The solution of the approximation problem in SS1 gives the best approximation of the distributed reference, with a minimum value of the norm of the control error in the space domain

$$\breve{W}O(\boldsymbol{x},\infty)=\sum_{i=1}^{4}\breve{W}_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{x},\infty) \tag{10.28}$$

In the block SS3 by relation (7.5)

$$\bar{Y}_i(\boldsymbol{x}_i,k)=\bar{Y}R_i(\boldsymbol{x}_j,k)^{-1}\,\bar{Y}(\boldsymbol{x}_i,k) \tag{10.29}$$

Where $\bar{Y}(\boldsymbol{x}_i,k)$ is vector computed in block R based on measured temperatures. The solution of the approximation problem in the output gives

$$\breve{Y}(\boldsymbol{x},k)=\sum_{i=1}^{4}\breve{Y}_i(\boldsymbol{x}_i,k)\,YR_i(\boldsymbol{x},k) \tag{10.30}$$

and according to Eq. (3.25) we will have

$$\breve{Y}(\boldsymbol{x},\infty)=\sum_{i=1}^{4}\breve{Y}_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{x},\infty) \tag{10.31}$$

in the steady-state.

In terms of Eq. (10.24), the relations between input-output control variables $\{U_i(k)\}_{i=1,4}$ and $\{Y_i(\boldsymbol{x}_i,k)\}_{i=1,4}$ are given by transfer functions $\{SH_i(x_i,z)\}_{i=1,4}$. Then, the relations for the variables $\breve{Y}_i(\boldsymbol{x}_i,k)$, $\breve{W}_i(\boldsymbol{x}_i,\infty)$, $\breve{E}_i(\boldsymbol{x}_i,k)$ and $U_i(k)$ refer to the *i*-th control loop with lumped parameters, see Fig. 5.4. Controllers in the one-parameter control loops are designed by the means common to control synthesis in the time domain.

If, in the steady-state $\left\{\breve{E}_i(\boldsymbol{x}_i,\infty)=\breve{W}_i(\boldsymbol{x}_i,\infty)-\breve{Y}_i(\boldsymbol{x}_i,\infty)=0\right\}_{i=1,4}$, then $\left\{\breve{Y}_i(\boldsymbol{x}_i,\infty)=\breve{W}_i(\boldsymbol{x}_i,\infty)\right\}_{i=1,4}$, which means that in terms of Eq. (10.29) and Eq. (10.31) are $\breve{Y}(\boldsymbol{x},\infty)=\breve{W}(\boldsymbol{x},\infty)$. Thus, the control error between the distributed output and distributed reference in the quadratic norm is minimal.

10.5.1 Design of the Control System Using the DPS Blockset

The design of the control system is carried out in the Matlab & Simulink by using the DPS Blockset. The control system utilizing the blocks from the DPS Blockset library is shown in Fig. 10.12. The distributed manipulated variable is represented by a 10 component vector, whose components are defined by uniformly distributed points on the computational mesh within the definition domain given by body of inductor. The block HLDS models the dynamics of the controlled system around the linearized neighborhood of the chosen operating mode. Zero initial conditions are assumed in the modeling procedure.

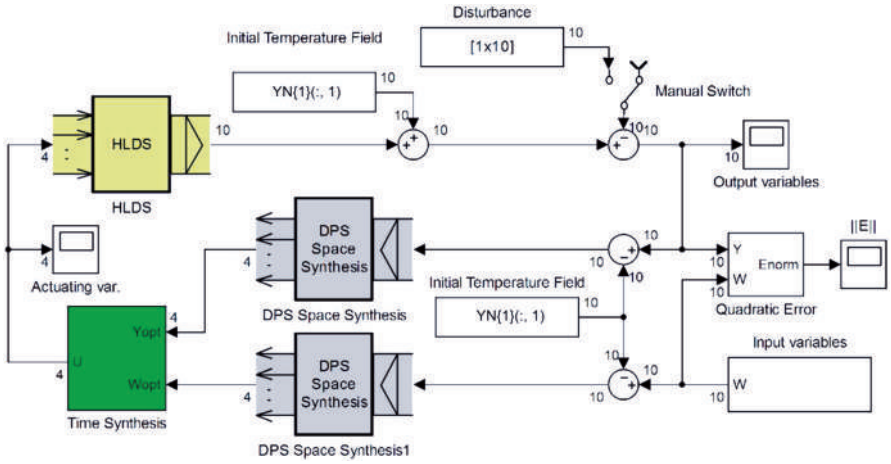


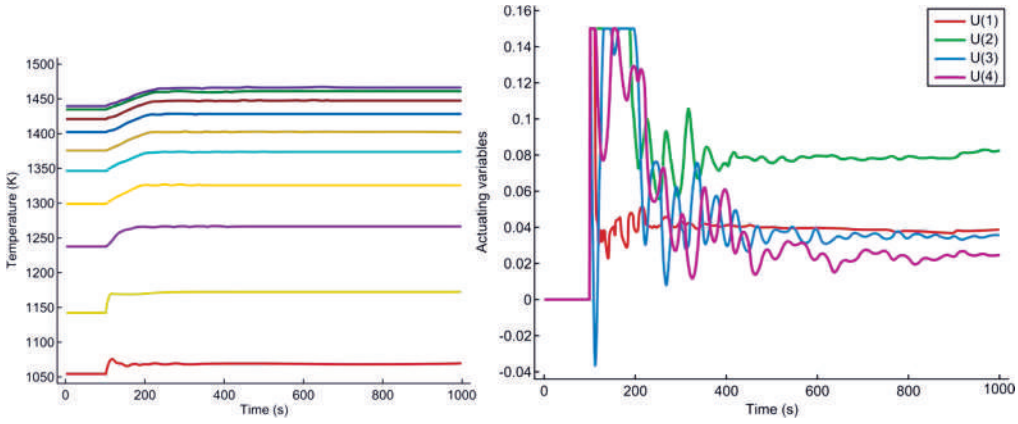
Fig. 10.12 Control loop in the Matlab & Simulink environment containing blocks from the DPS Blockset library.

The aim of the controller is to increase the temperature of the semi-products by 30 °C at the given steady-state temperature profile, along the length of the inductor. The second control task is a distributed disturbance $V(\boldsymbol{x},t)$ rejection problem, where the disturbance is a gradual decrease of 25 °C at the exit. Also, a small change of material properties may decrease the temperature at the output in the range of tens of degrees. The magnitude of the disturbance corresponds to a small change of material

properties. The solution of the control task in the time domain is accomplished by 4 robust PI1 IMC controllers, (Huba (2007)).

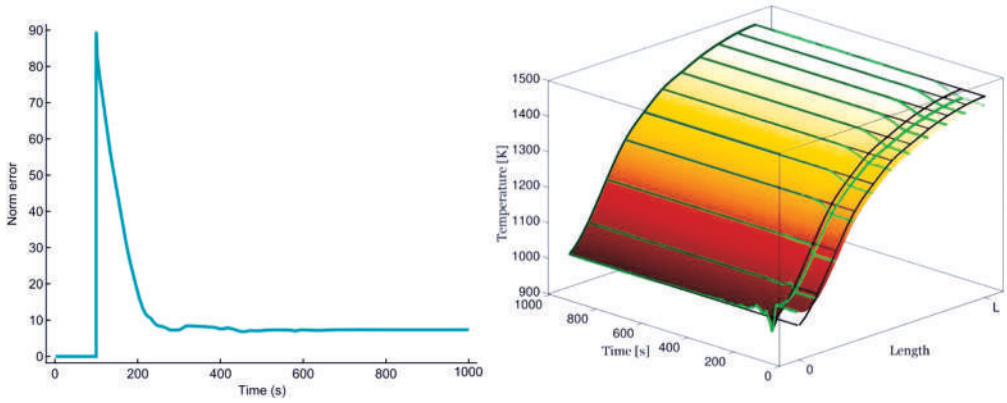
10.5.2 Change of the Distributed Reference

In the steady-state operating mode, at 100 s a delta change of distributed reference $\Delta W(\boldsymbol{x},\infty) = 30$ °C is introduced to the control system (CS). In the new steady-state it means $YN(\boldsymbol{x},\infty) + \Delta\breve{Y}(\boldsymbol{x},\infty)$, where $\Delta\breve{Y}(\boldsymbol{x},\infty)$ is the best approximation of $\Delta W(\boldsymbol{x},\infty)$.



(a) (b)

Fig. 10.13 (a) Time plot of the distributed output in 10 uniformly distributed points, (b) Time plot of the lumped inputs.



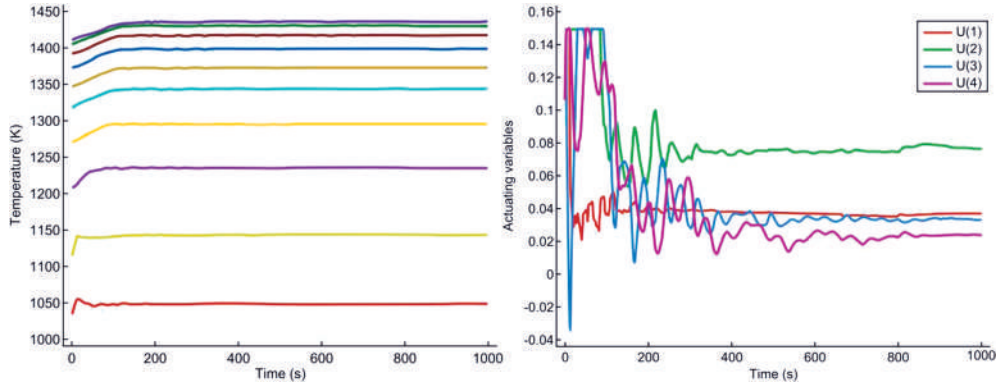
(a) (b)

Fig. 10.14 (a) Time plot of the quadratic norm of the control error, (b) Time and space plot of the control process.

Finally, changes of the temperature field were performed by 4 distributed actuators, the inductor heater coils. Lumped inputs are given as delta changes with respect to their steady-state values.

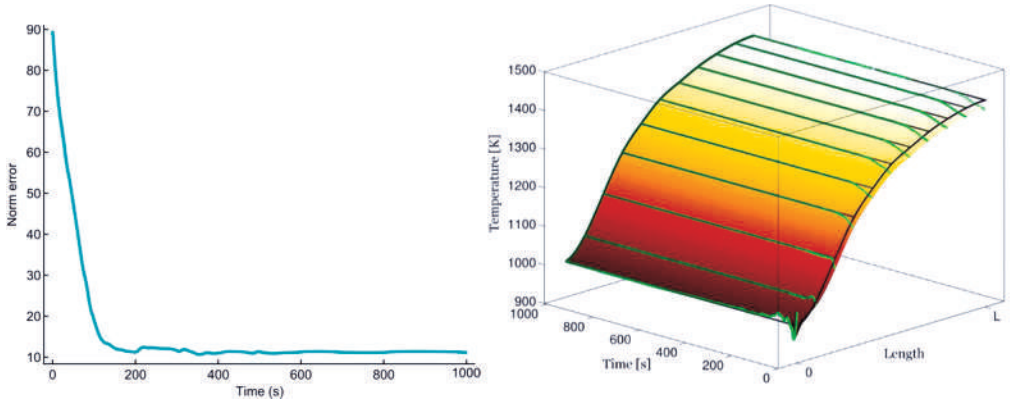
10.5.3 Distributed Disturbance Rejection

The simulation of disturbance rejection was performed under the formerly introduced circumstances, see Camber (2013), Kapusta (2013). After introducing the disturbance in the steady-state operating mode in the form of 25 °C temperature drop on the output of the heater, the compensation was carried out within a 250 s time horizon. (Fig. 10.15, Fig. 10.16)



(a) (b)

Fig. 10.15 (a) Time plot of the distributed output in 10 uniformly distributed points, (b) Time plot of the lumped inputs.



(a) (b)

Fig. 10.16 (a) Time plot of the quadratic norm of the control error, (b) Time and space plot of the control process in case of disturbance rejection.

One must note in relation to the figures presented here, that these were acquired in the development system featured in Fig. 7.9 with the SS3 block, that in reality uses a validated numerical model of induction heating, that was created in COMSOL Multiphysics.

11. MODELING AND CONTROL OF THE PLASTICS EXTRUSION PROCESS

Extrusion is one of the most common processing technologies for plastics and rubber compounds, since it is used in 80% of manufactured plastics. This technology is used either for the manufacture of products (bars, tubes, profiles, fibers) or may be a preparatory operation for subsequent processing operations (injection molding, blow molding), by which products with diverse shapes are made, (Rauvendaal (2001).

An experimental laboratory called "Controlled processing of plastics" has been established at the Faculty of Mechanical Engineering of the Slovak University of Technology for the purposes of modelling and control systems design for plastic extrusion. The experimental device is a single screw extruder with electric heater bands, this is described in the next section. (Fig. 11.1).

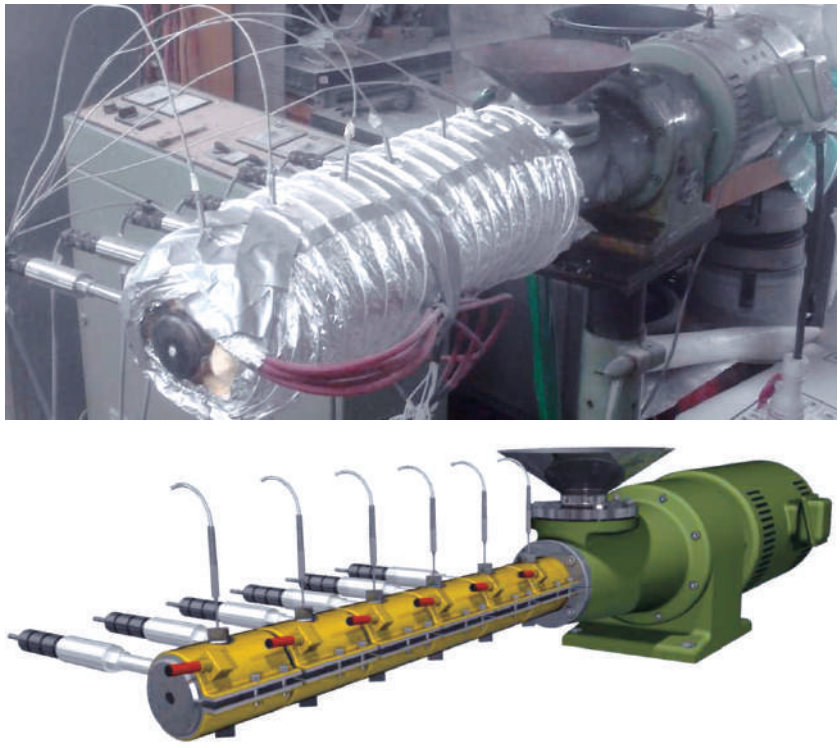


Fig. 11.1 Extruder at the experimental laboratory and its CAD model revealing the details hidden under the thermal isolation.

11.1 Extruder Operation

The hopper (1), Fig. 11.2, is filled with plastic pellets (8), which is then conveyed by the rotating screw (4). The most important process taking place inside is the melting of the plastic material, therefore every extruder has a heated part (5), which in our case is composed of 6 heater bands (6). A thermocouple is installed to measure the temperature in the middle of each band (7), close to the inner surface of the barrel (2). The plastic is molten, mixed, compressed and finally pushed through the die (3), which defines the outgoing profile and is changed depending on subsequent operations. Cooling channels (11) serve to prevent premature melting and as thermal protection of the screw drive (12) motor and gearbox.

It is important to mention that, before starting the rotation of the screw, the barrel has to be preheated to the desired temperature field, which is determined by the particular plastic material and profile. Insufficient preheating would cause the material not to melt at locations where it is supposed to do so, consequently rising pressure and eventually resulting in seizing the screw mechanism.

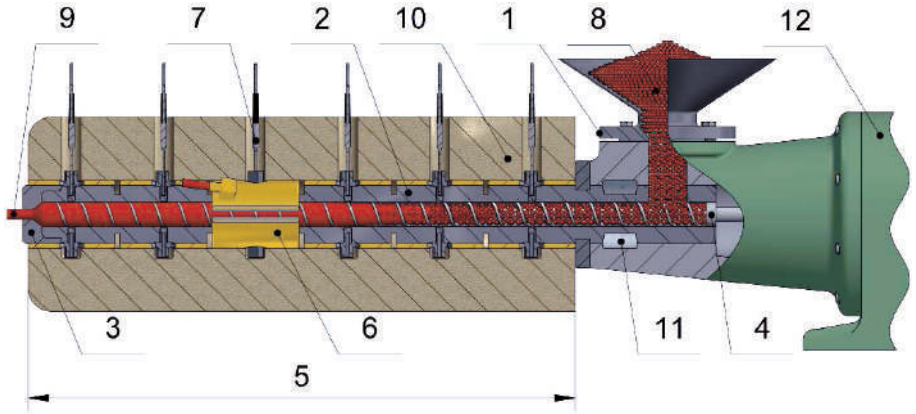


Fig. 11.2 Extruder assembly: 1 – Hopper, 2 – Barrel, 3 – Die, 4 – Screw, 5 – Heated part, 6 – Heater band, 7 – Temperature probe, 8 – Material feed, 9 – Extruded profile, 10 – Thermal isolation, 11 – Cooling channel, 12 – Screw drive motor.

11.2 Analysis of the Extrusion Process

The temperature field of the extruder barrel has a major influence on the product quality, as well as the process itself. The prescribed temperature field differs for various plastics and subsequent technological operations, as each material has different properties. The most important property is the melting temperature, which is in close

relation with the physical and chemical properties of the processed plastic (Rauvendaal (2001), Harold (2005)).

Higher process temperatures may be beneficial in order to facilitate material flow in the barrel and lowering production costs, however, overheating results in output material that is too soft, or worse, rendering it useless due to irreversible heat induced degradation.

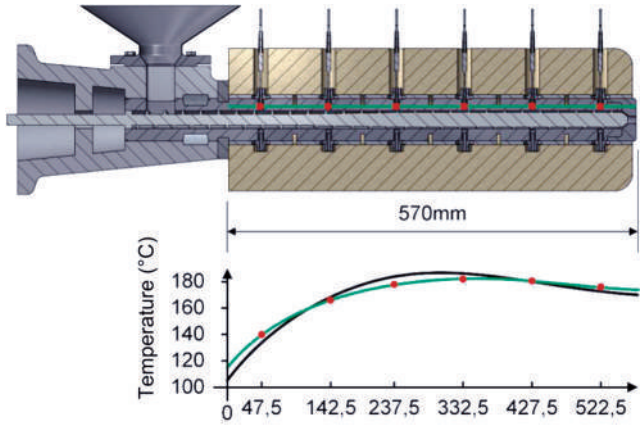


Fig. 11.3 An example of the desired temperature field along the longitudinal dimension of the device.

The temperature field along the barrel axis is prescribed by the process engineer, based on the processed product. However, the real extruder has only 6 barrel temperature measurements (red dots in Fig. 11.3, as only 6 thermocouple holes drilled to about 7 mm from the inner surface are available. This has the consequence that the measured temperature differs somewhat from the plastic temperature. Further along in this chapter, we will assume that the temperature field agrees to the measurements (green curve in Fig. 11.3.

11.3 Formulation of the Control Problem for the Extrusion Process

As we can conclude from the discussion above, the extruder system works in two main operating modes that need to be controlled:

a) The preheating of the extrusion barrel (no plastic present in the barrel cavity). The control system ensures the prescribed preheating speed, as well as the desired temperature profile as a pre-processing step. This happens before plastic processing starts, in case of die change, or when another material is to

be processed that requires a different temperature field. Such situations usually stop the production for some time, mainly due to the required change of temperature that is simply carried out by steady-state power settings. It is a system with slow dynamics, where the generated heat is only transferred to the surroundings via the outer surface. The properties of this system practically do not change throughout the operating temperature range, that is, 25°C – 300°C. Thus, the main control objective is to perform the temperature control of the extruder barrel in the range of 25°C – 300°C to shorten manufacturing downtimes.

b) Temperature regulation during the plastics extrusion process itself. The control objective is now keeping the desired temperature field during production, while it is subject to various disturbances, the most significant being: barrel hopper side cooling (fluctuating due to uncontrolled flow which depends on water supply pressure only), fluctuations in plastic pellet size, addition of additives to the base material, changes in friction (and fluid flow velocities), etc. This system has considerably faster dynamics than in the previous case, as heat is transferred to the extruded plastic, as well as to the cooling channels at the hopper side. Moreover, dynamic properties vary with the processed plastic, (Harold (2005), Previdi (2006)).

11.4 Modelling the Dynamics of the Extrusion Process

Plastics extrusion is one of those systems where one requires deep insight into the physics of the underlying phenomena. Experiments are very time consuming due to the slow dynamics, involving material and energy waste. The increasing computing power that is available for simulations enables the creation of sophisticated numerical models for analysis. Thanks to this, real experiments can be substituted by virtual ones, where the risk of damaging industrial machinery vanishes and no material or energy is wasted. Moreover, insight is gained in areas where it was not possible before experimentally, due to instrumentation and other technological limits.

From a mathematical perspective, the extrusion process is yet again a system governed by nonlinear partial differential equations. From the point of view of systems and control theory, these are distributed parameter systems. In this discussion, the software COMPUPLAST Virtual Extrusion Laboratory (2011) and ANSYS Polyflow (2010) are used for modelling and simulation. The simulation results obtained using these products are the distributed step responses of the extruder temperature field and the plastic material along the extruder, and eventually all

necessary parameters of the extrusion process in steady-state. Then, the dynamics decomposition and control synthesis into space and time components is introduced. The control system is designed in Matlab & Simulink utilizing the DPS Blockset.

11.5 Model in the COMPUPLAST Software Environment

The COMPUPLAST package is a simulation software tailored to the modelling of plastics extrusion processes, providing support to process engineers responsible for establishing process parameters and settings. It includes a polymer rheology model, as well as thermodynamics and fluid flow physics. (Fig. 11.4).

It serves for the computation of steady-state process parameters, such as the optimal temperature field, pressure distribution, extrusion mass flow rate based on screw speed, or the screw speed as a function of the desired mass flow rate defined by production requirements. One only needs to input the extruder parameters, characteristic dimensions, properties of the processed plastic and other standard process parameters.

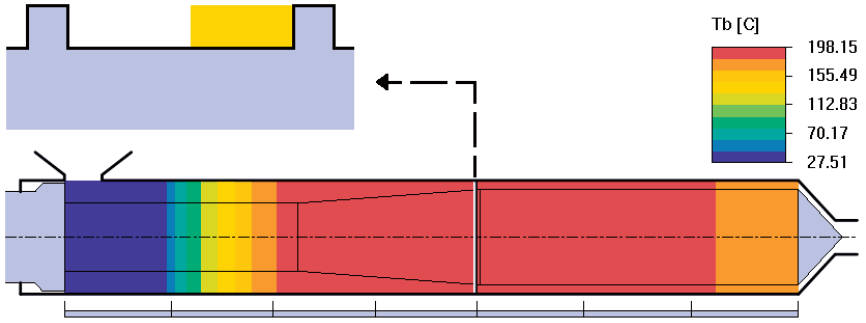


Fig. 11.4 The COMPUPLAST model: temperature field of the processed plastic with the possibility to investigate the amount of non-melted plastic in any location along the extruder axis.

11.6 Model in the ANSYS Polyflow Software Environment

ANSYS is a multiphysics simulation environment, based on the finite element method. Although it is not tailored to plastics extrusion technology specifically, it includes fluid dynamics as well as thermodynamical models, which are needed for the

modelling of the extrusion process. It is then possible to obtain time-dependent values of certain parameters, namely the temperature field of the processed plastic as well as the barrel, pressure and velocity fields throughout the whole definition domain. (Fig. 11.5, Fig. 11.6).

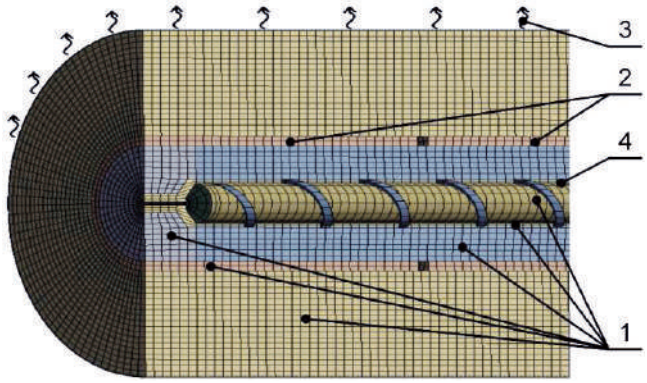


Fig. 11.5 Section view of the ANSYS virtual model.

Heat generated by heater bands dissipates in all directions, but due to the insulation, it predominantly transfers to the barrel and thereby heats the processed plastic. It is necessary to define the following parameters in order to create a thermal model of the process:

- material properties of the modelled parts (1),
- electrical power supplied to individual heater bands (2),
- heat dissipation to the surroundings (3) and
- presence and type of processed plastic as well as its motion (4).

The non-stationary barrel temperature field is given by the Fourier-Kirchhoff equation (10.21). The simplified model of the extrusion process focuses on the relation between the heater inputs and the barrel temperature field. Since the model is used to estimate only temperatures and heat flux, the screw rotation and rheology of the extruded plastic are not considered. These are a part of the COMPUPLAST model.

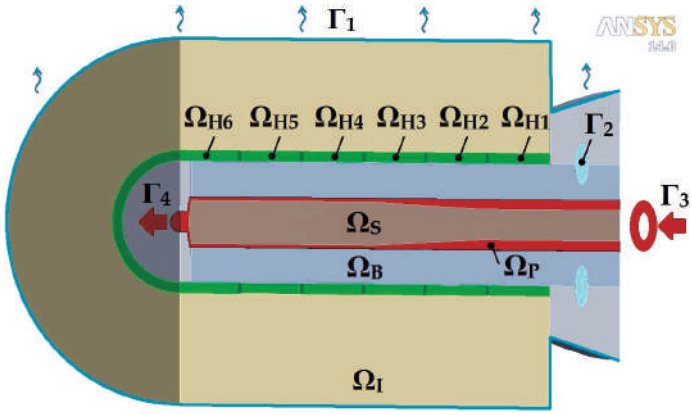


Fig. 11.6 The computational model comprises of the domains: Ω_B – the barrel, Ω_P – plastic, Ω_S – the screw, Ω_I – thermal isolation, $\Omega H1 \ldots 6$ – the heater bands; and the boundaries: $\Gamma 1$ – the outer surface, $\Gamma 2$ – the drive protection cooling channel, $\Gamma 3$ – cold plastic inlet, $\Gamma 4$ – hot plastic outlet.

The physical properties of the materials of which the extruder device consists are considered constant (ρ, c, λ = constant) within the temperature interval 25°C–300°C. Unfortunately, this simplification cannot be applied to the processed plastic, as its physical properties are subject to significant changes. Table 11.1.

	Steel	**Isolation**	**Plastic (Polypropylene)**			
	25-300°C	25-300°C	25-50°C	50-138°C	138-163°C	163-300°C
ρ	7750	18	400	400 – 500	500-735	735
c_p	450	1000	1700	1700	1700-2100	2100
λ	20	0,2	0,22	0,22	0,22-0,2	0,2

Table 11.1 Physical properties of the material domains.

Heat is generated in the heating bands and their output is controlled by thyristors. This is indirect resistance heating, when the electrical energy is transformed into heat according to Joule's law

$$Q_{\mathrm{el}} = RI^2 t = P_{\mathrm{el}} t \tag{11.1}$$

where R (Ω) is the heater wire resistance, I (A) current, P_{el} (W) electric power. The input heat generated in the heater band (per unit volume) is

$$\dot{q}_{\mathrm{H}i} = Q_{\mathrm{el},i} \big/ \left(t\, V_{\mathrm{H}i}\right)\big|_{i=1\ldots6} = P_{\mathrm{el},i} \big/ V_{\mathrm{H}i}\big|_{i=1\ldots6} \tag{11.2}$$

where $V_{Hi}\left(\mathrm{m}^3\right)$ is the i-th heater band volume.

Newton's law of cooling describes heat dissipation from the surface to the surroundings by natural circulation of air (free convection). The greater the temperature difference between surface and ambient air is, the more heat is dissipated to the surroundings

$$\dot{q}(\boldsymbol{x},t) = h\left(T(\boldsymbol{x},t) - T_{\mathrm{A}}\right), \quad \boldsymbol{x} \in \Gamma_1 \tag{11.3}$$

where $h = 10\ \mathrm{W\ m^{-2}\ °C^{-1}}$ is a heat transfer coefficient and $T_A = 25\,°\mathrm{C}$ is the temperature of the ambient air.

Water cooling is substituted by the Dirichlet boundary condition with the temperature of cooling water of $T_W = 15\,°\mathrm{C}$

$$T(\boldsymbol{x},t) = T_{\mathrm{W}}, \quad \boldsymbol{x} \in \Gamma_2 \tag{11.4}$$

Plastic inlet Γ_3 has a defined mass flow rate $Q_m\left(\mathrm{kg\ s^{-1}}\right)$ and inflow temperature $T_{in} = 25\,°\mathrm{C}$

$$T(\boldsymbol{x},t) = T_{\mathrm{in}}, \quad \boldsymbol{v}(\boldsymbol{x},t) = Q_m \big/ \left(\rho(\boldsymbol{x},T) S_{\Gamma 3}\right), \quad \boldsymbol{x} \in \Gamma_3 \tag{11.5}$$

where $S_{\Gamma 3}\left(\mathrm{m^2}\right)$ is the inlet cross-section surface. The mass conservation law requires the in- and outflow rate being equal, therefore the flow velocity is changing with fluid density throughout the domain. So, the plastic outlet boundary condition is as follows

$$\boldsymbol{v}(\boldsymbol{x},t) = Q_m \big/ \left(\rho(\boldsymbol{x},T) S_{\Gamma 4}\right), \quad \boldsymbol{x} \in \Gamma_4 \tag{11.6}$$

where $S_{\Gamma 4}\left(\mathrm{m^2}\right)$ is the cross-section surface of the outlet. The amount of heat extracted from the system is driven by the plastic flow rate and outlet temperature at the die, (ANSYS Polyflow (2010)).

11.7 Experimental Device

The experimental laboratory called "*Controlled processing of plastics*" is located in the Laboratory of Particulate Materials Processing at the Faculty of Mechanical Engineering of the Slovak University of Technology in Bratislava. The schematics of the experimental device in Fig. 11.1, Fig. 11.2 is shown in Fig. 11.7. The device consists of the following components:

- 6 heating bands of 6 × 500 W power, 6 thyristors controlled via a measurement board in the PC, by modulating electrical power in the full 0–100% range,
- 6 thermocouples built in the barrel for temperature sensing and 1 thermocouple measuring ambient temperature,
- 6 pressure sensors monitoring pressure inside the barrel cavity for safety reasons; motor drive of 3 kW power.

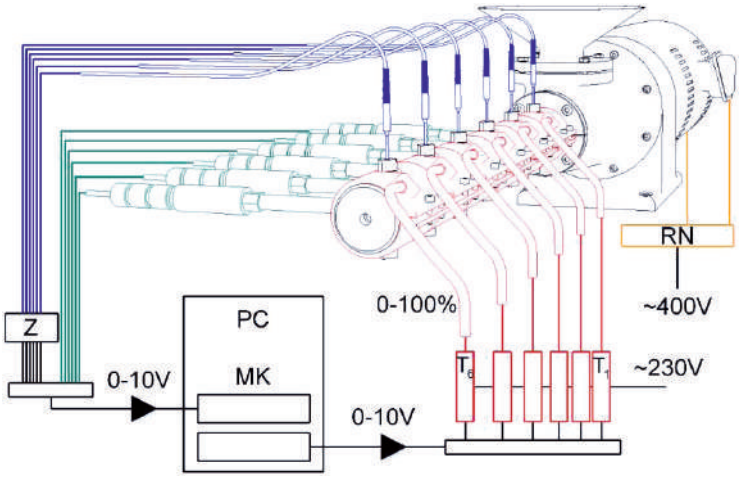


Fig. 11.7 Connection schematics of the controlled extrusion system: Z - amplifier, MK - laboratory measuring boards, RN - drive control.

11.8 Extruder Operation as DPS

The basic characteristics of operating modes and states of the extruder are given by the temperature fields inside. From the perspective of systems and control theory, this is a thermal system with lumped inputs and distributed outputs, (Hulkó (1998). In the following discussion, we will focus on the relationship between the performance of

individual heating bands $\{U_i(t)\}_i$ and the distributed output $Y(\boldsymbol{x},t)$, considered as the extruder barrel temperature field before and during the processing procedure.

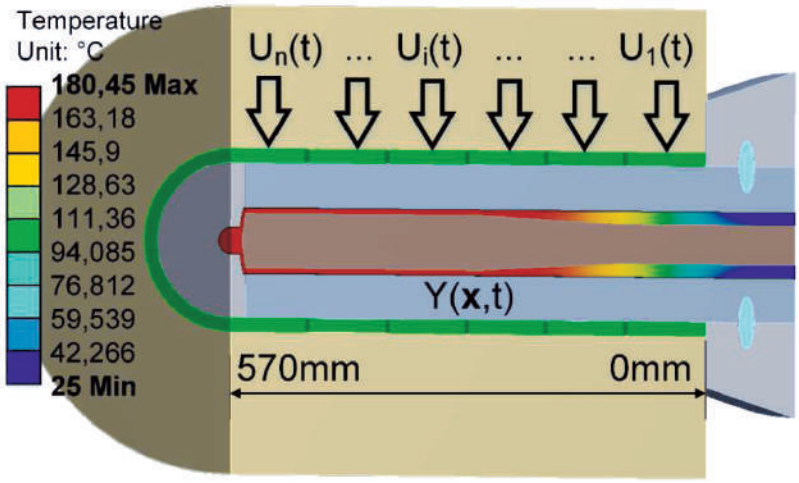


Fig. 11.8 Model of the extruder device as the system with lumped inputs (heating band powers) and distributed output (temperature field).

As it has been previously mentioned, there are two operating modes of concern. First, preheating of the extruder with no plastics being processed, second, temperature field control during plastics processing. Both exhibit nonlinear behavior, therefore the dynamical analysis is performed in a linearized neighborhood of these operating modes.

	Extruder preheating	**Heating during extrusion**
Operating mode	Plastic is not processed	Plastic is processed: 2.5 kg hr^{-1}
Cooling	off	on
Powers P1-6 (W)	64, 17, 21, 17, 17, 20	100, 50, 50, 50, 20, 20
Step changes	by 10W: 74, 27, 31, 27, 27, 30	by 50%: 150, 75, 75, 75, 30, 30

Table 11.2 Simulation parameters for the dynamical analysis of the different modes.

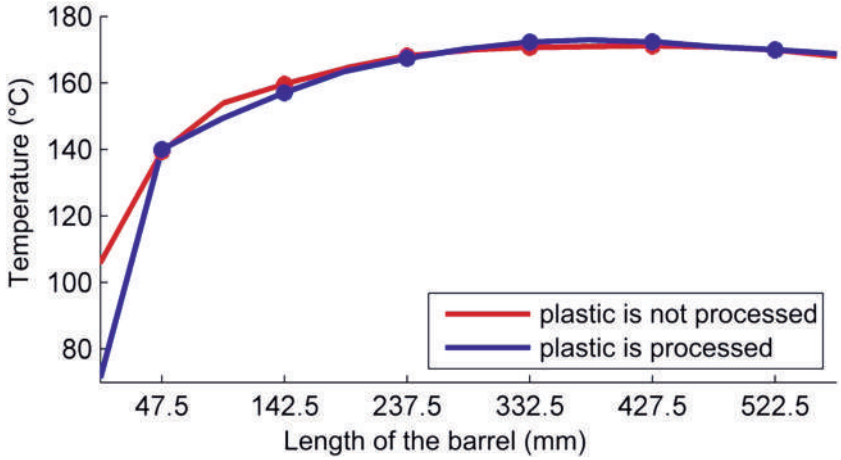


Fig. 11.9 Temperature distribution of the operating modes in which the analysis was performed.

We performed a simulation study of both models, in order to obtain their (discrete time) characteristics $\{\mathcal{H}H_i(\boldsymbol{x},k)\}_{i=1,6}$ in the form of responses to step changes in individual heater powers. These were evaluated on a computational mesh of the definition domain. Subsequently, we obtain the set of impulse responses by computing temperature differences between time steps, i.e. by subtracting the step responses shifted by one time step

$$\{\mathcal{G}H_i(\boldsymbol{x},k) = \mathcal{H}H_i(\boldsymbol{x},k) - \mathcal{H}H_i(\boldsymbol{x},k-1)\}_{i=1,6} \tag{11.7}$$

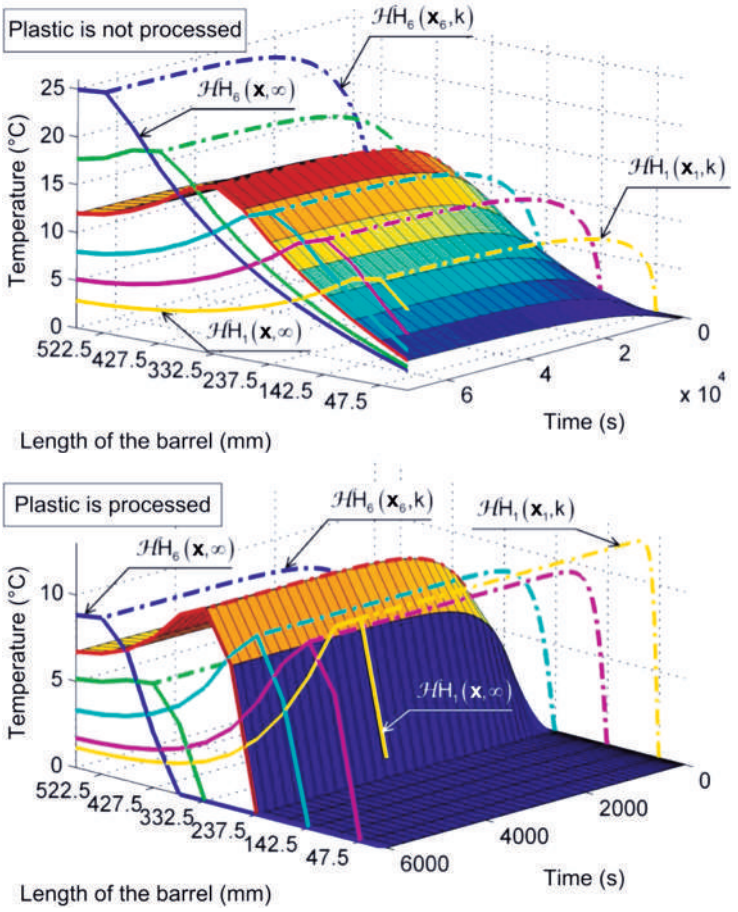


Fig. 11.10 Distributed responses to step changes in heating band powers $\{\mathcal{H}H_i(\boldsymbol{x},k)\}_{i=1,6}$ *and respective partial responses* $\{\mathcal{H}H_i(\boldsymbol{x}_i,k)\}_{i=1,6}$ *in locations* $\{\boldsymbol{x}_i\}_{i=1,6}$ *in linearized neighborhood of chosen steady-state mode.*

47.5,...,522.5 mm - locations of thermocouples $\{\boldsymbol{x}_i\}_i$ in the heated part of the extruder.

0-6000 s - settling time interval of temperature field.

$\{\mathcal{H}H_i(\boldsymbol{x},t)\}_i$ - transient characteristics of temperature field.

$\{\mathcal{H}H_i(\boldsymbol{x}_i,t)\}_i$ - partial transient characteristics in locations $\{\boldsymbol{x}_i\}_i$.

Temperature - changes of the temperature field relative to the profile in the chosen steady-state mode.

The overall discrete-time distributed response of the temperature field is

$$Y(\boldsymbol{x},k)=\sum_{i=1}^{6}Y_i(\boldsymbol{x},k)=\sum_{i=1}^{6}GH_i(\boldsymbol{x},k)\oplus U_i(k) \tag{11.8}$$

where $\oplus$ is the sign for discrete convolution. The distributed step responses with the highest gain are in the middle of the respective heated bands (measured along the barrel axis, emphasized by dashed lines) Fig. 11.10. The analyzed extruder has 6 such locations $\{\boldsymbol{x}_i\}_{i=1,6}$.

We can assign the discrete-time transfer functions $\{SH_i(\boldsymbol{x}_i,z)\}_{i=1,6}$ to the partial distributed transient characteristics in these locations $\{\mathcal{H}H_i(\boldsymbol{x}_i,k)\}_{i=1,6}$, (Fig. 11.11). These represent the time domain dynamics of the controlled DPS. Partial outputs in measurement points are

$$\{Y_i(\boldsymbol{x}_i,z)=SH_i(\boldsymbol{x}_i,z)U_i(z)\}_{i=1,6} \tag{11.9}$$

The dynamic characteristics of the controlled system are determined as reduced distributed transient characteristics in steady-state in the space domain

$$\{\mathcal{H}HR_i(\boldsymbol{x},\infty)=\mathcal{H}H_i(\boldsymbol{x},\infty)/\mathcal{H}H_i(\boldsymbol{x}_i,\infty)\}_{i=1,6} \tag{11.10}$$

where $\{\mathcal{H}H_i(\boldsymbol{x}_i,\infty)\neq 0\}_{i=1,6}$.

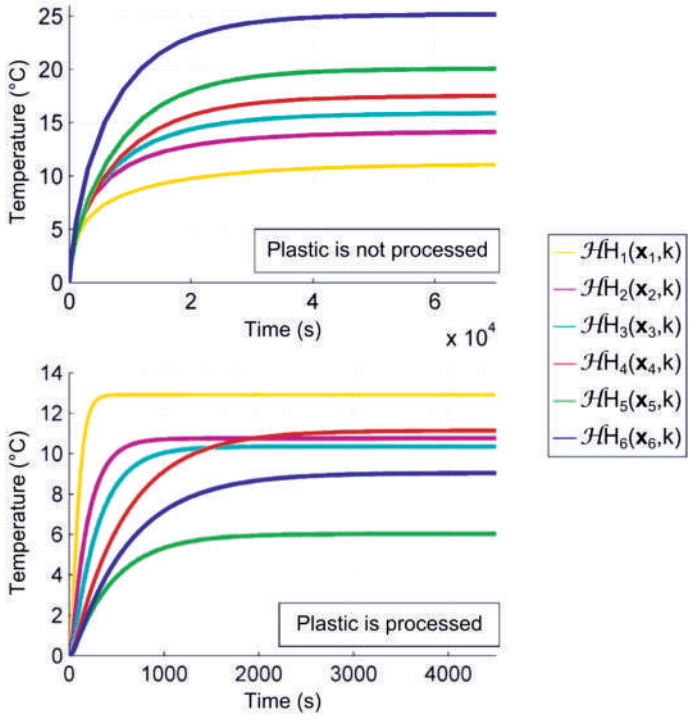


Fig. 11.11 Selected partial transient distributed characteristics to step changes in individual heater powers, in linearized neighborhood of the operating mode.

By using reduced steady-state profiles of partial distributed step responses we obtain partial steady-state distributed outputs as

$$\left\{Y_i\left(\boldsymbol{x},\infty\right)=Y_i\left(\boldsymbol{x}_i,\infty\right)\mathcal{H}\bar{H}R_i\left(\boldsymbol{x},\infty\right)\right\}_{i=1,6} \tag{11.11}$$

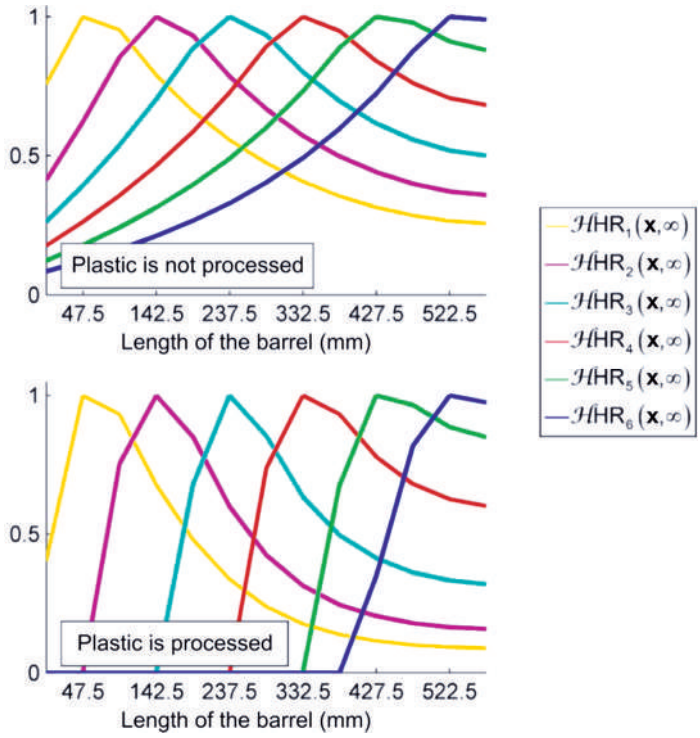


Fig. 11.12 Reduced steady-state profiles of partial distributed step responses.

For the overall distributed steady-state output we have

$$Y(\boldsymbol{x},\infty)=\sum_{i=1}^{6} Y_i(\boldsymbol{x}_i,\infty)\,\mathcal{H}\hat{H}R_i(\boldsymbol{x},\infty) \tag{11.12}$$

We express the reduced profiles of partial distributed outputs in each time step k by the same procedure

$$\{YR_i(\boldsymbol{x},k)\}_{i=1,6}=\{Y_i(\boldsymbol{x},k)/Y_i(\boldsymbol{x}_i,k)\}_{i=1,6}, \text{ where } \{Y_i(\boldsymbol{x}_i,k)\neq 0\}_{i=1,6} \tag{11.13}$$

which further yields

$$\{Y_i(\boldsymbol{x},k)=Y_i(\boldsymbol{x}_i,k)\,YR_i(\boldsymbol{x},k)\}_{i=1,6} \tag{11.14}$$

11.9 Control System Design Using DPS Blockset

We can proceed to the control of the temperature field in the linearized neighborhood of the selected operating mode, based on the obtained time/space domain

characteristics of the process dynamics during extruder barrel heating and extrusion. This will be carried out according to the distributed reference, in terms of the control system introduced in Section *7*, Fig. 7.8.

The control feedback loop will be constructed based on this scheme, where parallel to the controlled system, controllers created based on the verified model of the plant are used. The control system for the thermal field of the extruder is illustrated in Fig. 11.13, while the control system for the extrusion process temperature field is shown in Fig. 11.18.

The control process is carried out close to the steady-state operating mode of the extruder, to which we may assign the corresponding powers and the resulting temperature profiles. These values are present in the form of initial conditions in the HLDS block. Outputs in this case are then the temperatures measured at 6 equidistant locations on the extruder (see Fig. 11.13). The measured outputs are then compared in an algebraic member of the scheme with the outputs coming from the model. The difference of these enters the disturbance generator unit, which augments the result to 13 values using an approximation procedure, as shown in Fig. 11.14. The initial input/output conditions of the HLDS block are given by the steady-state operating mode of the extruder.

11.9.1 Synthesis of control by PI controllers

PI controllers are used to generate 6 inputs in the time synthesis block, inputs expressing the changes in power of the heating elements in the range of 0-500 W, in correspondence with the heating powers belonging to the particular operating mode (Huba (2007)).

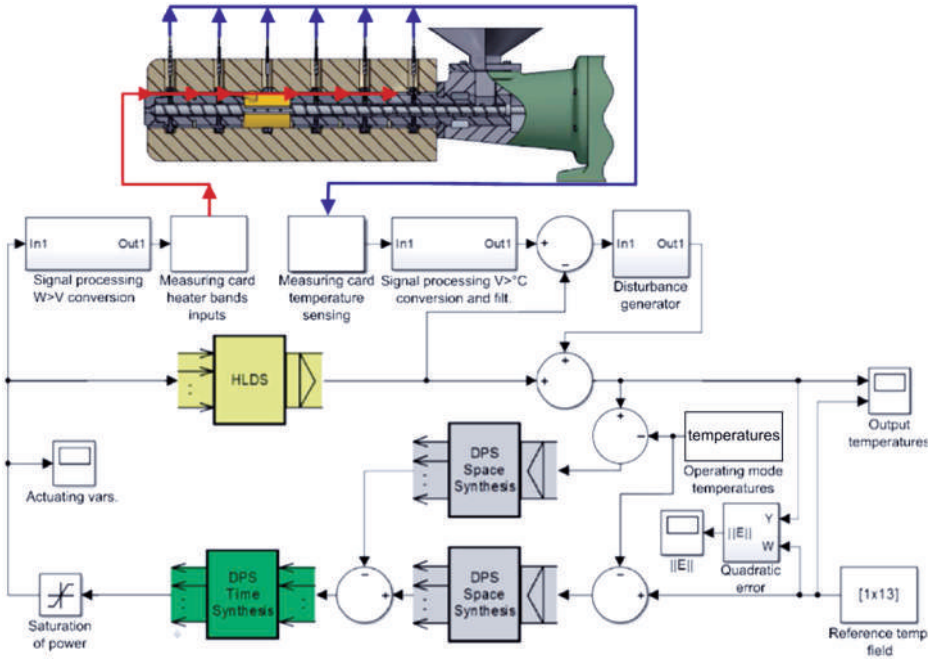


Fig. 11.13 Control loop for extruder barrel preheating temperature control.

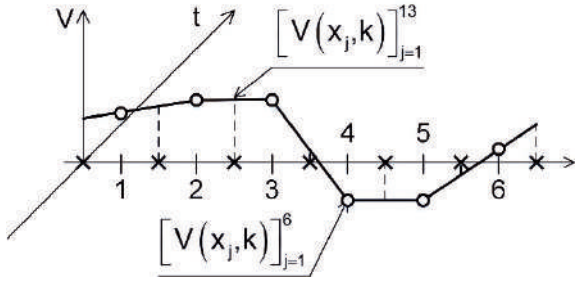


Fig. 11.14 Reconstruction of the disturbance.

The three dimensional graph shows the control process of the temperature field in reaction to a change in the reference temperature field. The blue line and surface denotes the initial steady-state temperature, while the green one shows the reference levels.

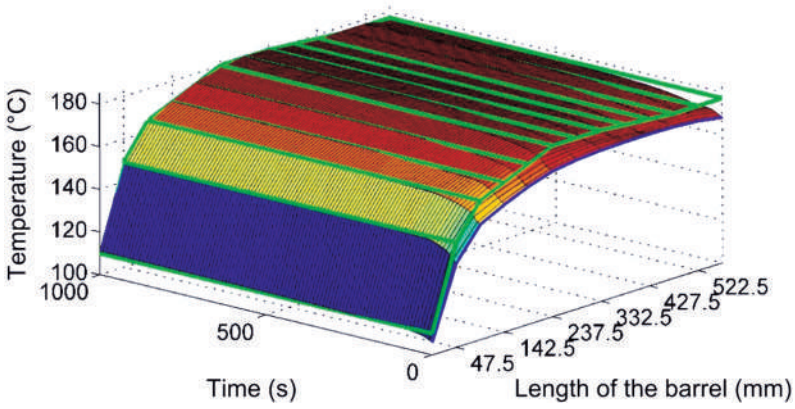


Fig. 11.15 Temperature field during the control process.

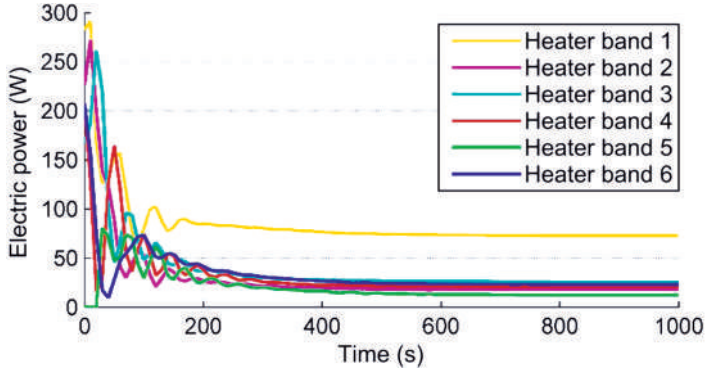


Fig. 11.16 Plots of the individual heating zone powers during the control process.

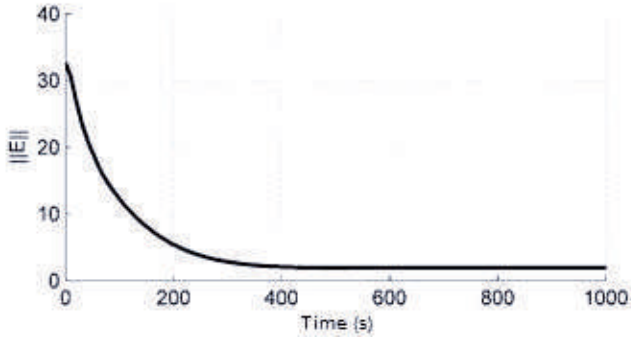


Fig. 11.17 Quadratic norm of control error.

11.9.2 Control synthesis by MPC

The following figure shows the schematics for the temperature field of the extruding machine during the process of extrusion, where plastics are processed. The

control synthesis here uses predictive control, designed using the MPC Toolbox in Matlab.

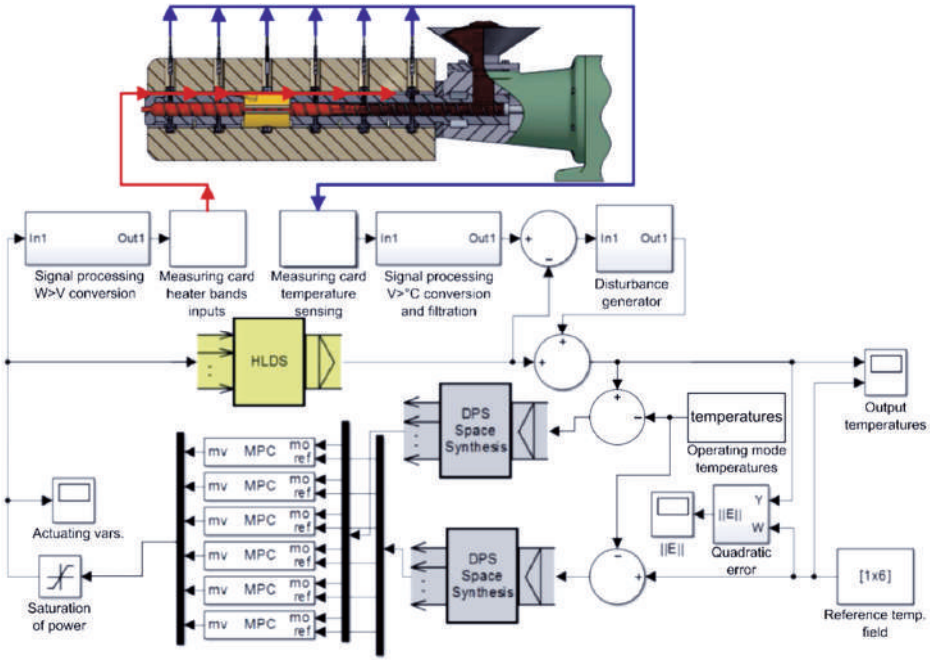


Fig. 11.18 Control loop for the extruder barrel preheating temperature with a parallel model.

The 3D plot below shows the temperature response to the step change of the desired operating mode of the extrusion process. The blue line represents the original operating point, the green the new desired one.

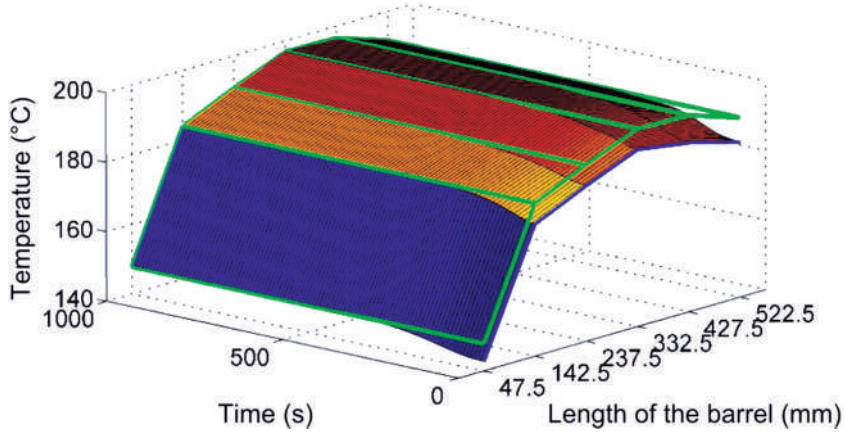


Fig. 11.19 Temperature field during the control process.

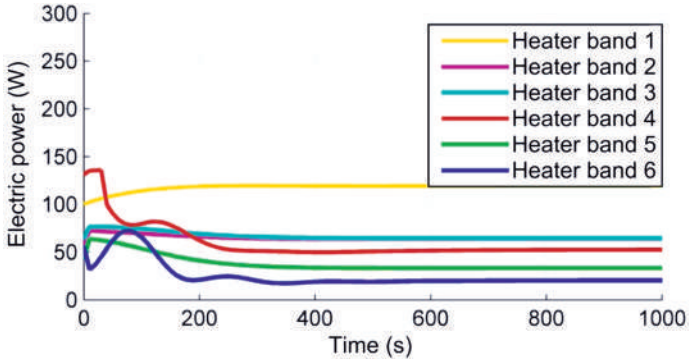


Fig. 11.20 Plots of individual heating zone powers during the control process.

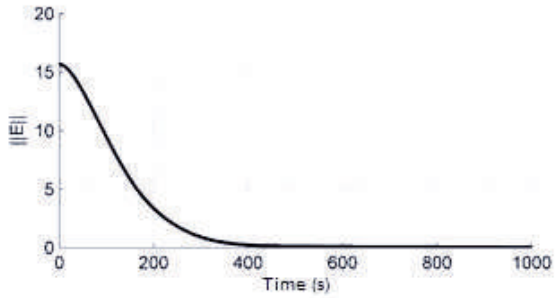


Fig. 11.21 Quadratic norm of control error.

The results of the numerical and physical simulations performed on the experimental device demonstrate novel co-simulation opportunities for the successful control of plastics extrusion processes as distributed parameter systems (Lipár (2014)).

Just as in the previous chapter, we may note that these results were obtained by the development system featured in Fig. 7.9. The system uses a validated numerical model of the plastics extrusion process, made in ANSYS Polyflow. As soon as the control system (CS) was connected to the extruder, it was possible to measure the temperature responses. The temperature fields are shown here with the precision of the validated numerical models, therefore both the measured and the simulated responses are practically identical in these locations.

12. MODELING AND CONTROL OF THE CONTINUOUS CASTING OF STEEL

The continuous casting (CC) of steel is one of the fundamental technologies of our modern world. The development of this technology has a long history, with the first patent acquired by Lord Henry Bessemer in 1857 for the continuous casting of malleable iron using counter-rotating cylinders. The following decades were abundant with numerous new patents that have significantly contributed to the further development of this technology. A major breakthrough came in 1934, when a German engineer Siegfried Junghans had patented a unique movement of the mold with a negative strip. He managed to overcome the major problems associated with the continuous casting of steel, especially the effect of sticking to the walls of the mold. The pilot operation using this method was launched in 1943, but the actual worldwide development of this technology occurred in the 50's and 60's of the last century. After all, one of the first continuous casting machines for steel in the world and the very first within Czechoslovakia, was launched in 1961 in Podbrezová Steel Mills.

The empirical skills that have been accumulated over the decades established optimal operating modes of the casting process. Today, thanks to the development of information technology, software based on the numerical solution of nonlinear partial differential equations ensure the ability to analyze the dynamics of the continuous casting process. The acquired dynamical characteristics open up new possibilities for creating input-output models with distributed parameters, as well as the formulation and problem solving process of the issues regarding the management of the continuous casting of steel. Research in the field of continuous casting is highly multidisciplinary. On the material level, it is a research of solidification, where the process is primarily dependent on the chemical composition of the cast material and thermodynamic conditions. The characteristic measure in the process of solidification is temperature, according to which we can determine when the material undergoes phase transformation. For a more accurate description of the casting temperature field, one can consider the flow of the melt, crystal growth of the solid phase, solidifying shell deformations, etc. Off-the-shelf software environments are prepared to address these complex, non-linear problems and provide a wide range of tools to analyze and interpret the results *(postprocessing).* Thanks to this, we are now able to understand the solidification process better and detect the possible errors in the design of the overall technological process. Another research area within the field of continuous casting deals with optimization in terms of economic costs and environmental impacts. This

mainly involves the design of new machines or production units. We are able to move the state of technology to the next level with the help of existing software products.

12.1 Continuous Casting Machine for Steel

The heart of the continuous casting technology is the continuous casting machine (CCM). There are several designs of these machines. Nowadays, the steel is usually cast on a radial type machine with a curved casting strand, (Fig. 12.1). Due to the curvature of the strand, it is not necessary to build high or deep hall structures, which reduces the initial construction cost and facilitates the handling of the castings. The radius of curvature depends on the size and type of the cast profile—it is different for a billet CCM, and different for a slab CCM. A central part of the radial CCM is a casting platform, on which a ladle turret and a tundish car is placed. A crew responsible for sub-tasks and supervising the operation of CCM is present on the casting platform during the process.

Liquid steel prepared for casting is delivered from the section of ladle metallurgy in a refractory container - ladle (1). The ladle filled with molten metal is placed on a ladle turret of the CCM. On this turret, the full ladle changes place with the empty one, which is then transferred for a refill. Liquid steel is poured in a controlled manner from the ladle, through the refractory shroud (tube) into the tundish (2). The tundish serves as an accumulator (buffer) of the molten steel with several possible outflow holes. The free surface of the steel in the tundish is covered with a thermal insulating slag that protects the steel against oxidation and binds non-metallic inclusions.

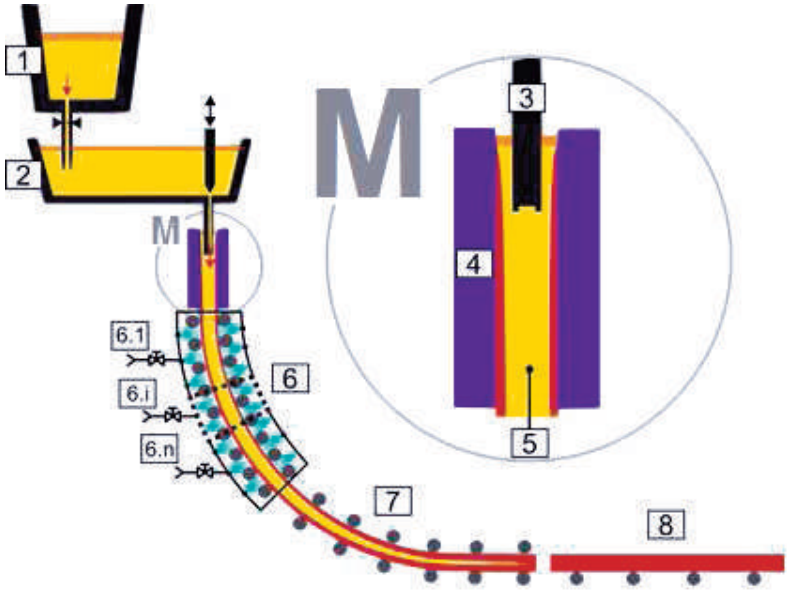


Fig. 12.1 Casting strand of the continuous casting machine – radial type, (Javurek (2008)).

Fixed, submerged entry nozzles are placed at the outflow holes (3). They protect the liquid steel against oxidation during the outflow from the tundish into the mold (M). The free surface of the mold is continuously filled with the casting powder, which acts as a thermal isolation barrier, inclusion binder and lubricant (it lowers the friction between mold walls and the casting). The level of the liquid steel is called meniscus and it is the beginning of the metallurgical length of the casting.

Solidification of the steel begins at the cold walls of the mold (4) just beneath the free surface. The casting is pulled out from the mold at constant speed, with the outer solidified surface (shell) containing a liquid core (5). The mold is oscillating in the vertical direction (amplitude in order of mm, frequency in order of Hz), which in combination with the melted casting powder significantly lowers the friction.

After the mold, the casting enters the secondary cooling area (6), where it is cooled with water sprays or water-air mist sprays. The secondary cooling area (SC) consists of independent water spray zones (6.1 – 6.n), which allows the setting the desired cooling profile. It is located in a closed chamber, because a direct spraying of the hot surface generates a large amount of steam.

Finally, the tertiary cooling area (TC) (7) follows, where the casting cools freely at ambient air temperature. In this section, the liquid core is almost closed and the casting is gradually straightened. The transition from the curved to the straight (horizontal) casting strand is driven in pinch roll units. It holds that, the more unbending points are present, the less is the risk of cold cracks (e.g. CCM in

Podbrezová Steel Mills this is a set of unbending points (radii) 9 m – 13 m – 21 m – ∞ m). The point where the liquid core ends is considered as the end of the metallurgical length. After the casting is unbended, it is cut into pieces of arbitrary lengths. The cut castings (8) are cooled down to the ambient temperature or reheated for further processing. The process is theoretically infinitely continuous and it is interrupted intentionally either for technological reasons (e.g. wear of components) or for economic-planning reasons (change of the chemical composition of the steel, change of casting profile). The unscheduled interruption of the casting process can be caused by technical failures of the machines, wear of individual components (submerged entry nozzle erosion), etc.

12.2 Continuously Cast Steel Casting Products

The portfolio of profiles and dimensions of the continuously cast steel products is very broad. This results in some degree of vagueness, especially near the limits of profile dimensions.



Fig. 12.2 Some basic profiles of the castings: Billets (□ square, ○ round), slab (rectangle), beam blank (I-profile).

The following Table 12.1 summarizes names of the basic profiles with description and approximate values of dimensions.

Profile name	**Description of the profile, *profile dimensions [mm]**
slab	rectangle, height > 100, width 500 – 3000
thin slab	rectangle, height < 100, width 500 – 1500
bloom	square or rectangle with a ratio of sides ≈ 1, side < 500
(round) bloom	round, diameter < 800
billet	square (□), side < 220
(round) billet	round (○), diameter < 220

beam blank	H profile / I profile, height < 500

*Table 12.1 Names of basic profiles of the continuously cast castings, *approximate values.*

The type of CCM installed in the steel mill is adapted to the main production portfolio. Flat products, e.g. metal sheets are produced by rolling of slabs, while narrow long products (pipes, wires and profiles) are produced by rolling of billets. Currently, there are three steel mills in Slovakia, which produce the steel by continuous casting technology. They all have CCMs of radial type and further process castings into final products.

Steel mill	**Product portfolio**
Slovakia Steel Mills	concrete reinforcement steel (bars, coils), rolled wire, threaded rods
U. S. Steel Košice	hot/cold rolled plates, electrotechnical plates, hot dip galvanized plates, organic coated products, tin plates, spiral welded pipes, radiators
Železiarne Podbrezová	seamless tubes, precision seamless tubes, cut and bent tubes, tube fittings

Table 12.2 Production portfolio of the steel mills in Slovakia.

Non-ferrous metals and their alloys such as copper, aluminium, brass and bronze are cast on horizontal or vertical CCMs. Besides the standard full section profiles, these CCMs enable the production of hollow tubular profiles, such as thick wall tubes. In comparison with the continuous casting of steel, these are distinct technologies, which conceptually fall into the extended family of continuous casting techniques.

12.2.1 CCM in Železiarne Podbrezová

Železiarne Podbrezová a.s. (Podbrezová Steel Mills) is a renowned producer of seamless steel tubes with a long tradition of continuous casting. The first CCM in the former Czechoslovakia was put into operation in the year 1961 in Podbrezová (it was a 2-strand vertical CCM of Soviet design). Since 1979, the main production portfolio of Železiarne Podbrezová consists of hot-rolled seamless steel tubes. Semi-products used in the production of seamless steel tubes are square cross-section steel blocks with a 205 or 160 millimeters long side. These are obtained by cutting continuously cast steel billets into smaller blocks (500 -1000 mm long). In addition to the main portfolio profiles (billets □205 or □160 mm), Železiarne Podbrezová produces another 6 types of square and round profiles. During the rolling process of steel tubes, steel is subject to

significant plastic deformations. The priority of steelmaking is therefore the highest possible quality of the material, the casting itself. In order to increase productivity and production quality, a new 3-strand CCM was put into operation in the year 2013, replacing the previous 4-strand CCM, which was in operation since 1982. Selected parameters of the previous and a present CCM are shown the Table 12.3 below.

CCM Manufacturer	**VÍTKOVICE (DEMAG licensing)**	**DANIELI**
Years in operation	1982 – 2013	2013 – present
Type of CCM	radial	radial
No. of strands / main radius (m)	4/9	3/9
Tundish capacity (metric tons)	8	12
Mold length (mm)	700	780
Oscillating bench	electro-mechanical	hydraulic
Electromagnetic stirring in mold	no	yes
No. of pinch roll units per 1 casting strand	2	5
Dummy bar	chain	rigid
No. of casting profiles	13	14
Main profile / radius of rounded edges	□ 200 / R8	□ 205 / R40
On-line model of casting strand	no	yes (by manufacturer)
No. of secondary cooling zones	3	4
Cooling medium in secondary cooling zones	water	water / water+air
Cutting system	torch	torch
Range of casting speeds ($m.min^{-1}$)	0.9 – 1.3 (□ 200 mm)	0.9 – 1.8 (□ 200 mm)

Table 12.3 Comparison of selected parameters of the present and previous CCM in Železiarne Podbrezová.

12.3 Dynamic Model of the Casting Temperature Field

Let us consider a steady-state operating mode of one strand in the CCM. The casting represents an open thermodynamic system, whose internal state is given by the temperature field at a certain instant in time. The input is the mass flow rate of the liquid steel $\dot{m}$ (kg s^{-1}) of some temperature and specific enthalpy H (J kg^{-1})

$$\dot{E}_{\mathrm{i}} = \dot{m} H_{\mathrm{i}} \tag{12.1}$$

where $\dot{E}_{\mathrm{i}}$ represents the thermal power of steel flowing into a mold. One part of this power is lost to the surroundings as a total loss power (W)

$$\dot{E}_{\mathrm{loss}} = \dot{E}_{\mathrm{mould}} + \dot{E}_{\mathrm{sc}} + \dot{E}_{\mathrm{tc}} \tag{12.2}$$

which accordingly to the particular heat transfer mechanism may be classified as power dissipated by

- conduction – copper mold area (M),
- forced convection and radiation – secondary cooling area (SC), and
- natural convection and radiation – tertiary cooling area (TC) .

The calculations may also include a contact with the support rollers, classified as heat transfer by conduction.

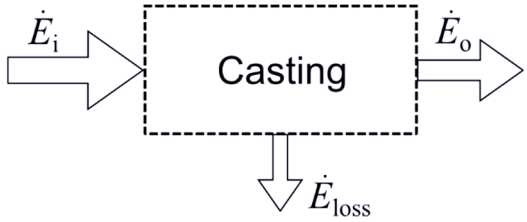


Fig. 12.3 The casting as a thermal dynamic system.

The thermal power dissipated from the casting is defined as

$$\dot{E}_{\mathrm{o}} = \dot{m} H_{\mathrm{o}} \tag{12.3}$$

while the mass flow rate remains constant.

The control task is to change the energy flow in the considered thermal system intentionally, in order to track the desired reference or to reject disturbances. From the viewpoint of the inputs, it is only possible to operate with a fraction of the total power

loss. This manipulable part enables us to change the surface temperature of the casting instantly. The intentional change of temperature in the deeper, undersurface layers of the casting is (theoretically) achievable with heavy surface undercooling, which unfortunately may initiate defects in the castings. Expressed in absolute numbers, the manipulable part of the effective cooling power is considerably lower compared to the total dissipated power $\dot{E}_{\text{loss}}$, where heat transfer by radiation dominates. For the sake of completeness, it is important to point out that the desired short-time change of the thermal power transferred by the mold $\dot{E}_{\text{mould}}$ during transient phenomena can impose undesirable oscillations in the cooling process. A technically feasible solution to control the temperature field of the casting is the manipulation of the reference, expressed as surface temperature. Alternatively, heat fluxes can be considered. Regarding the quality in relation to the reference, we may mention the temperature dependency of the steel ductility (Brimacombe (1999)).

The non-stationary temperature field $T(\boldsymbol{x},t)$ is governed by the Fourier-Kirchhoff equation (10.21), where liquid melt flow is not considered. The solution domain of the equation is a casting strand, starting at the meniscus. During solidification, a latent heat is released, whose effect is included in the source term.

The initial condition $T(\boldsymbol{x},0)=\overline{T}_{\text{ini}}$ represents, for example, the temperature field of the casting in some operating mode of the CCM. Boundary conditions represent heat fluxes on the casting in the direction of the normal vector $\boldsymbol{n}$.

Boundary conditions are defined separately for the

- mold $-\lambda\dfrac{\partial T}{\partial \boldsymbol{n}}=A^{-1}\left(\dot{m}c_{\text{p}}\Delta T\right)$,

where λ is the thermal conductivity of the casting, A (m^2) is the mold area in contact with the casting, c_{p} (J kg^{-1} K^{-1}) is the specific heat capacity of water and ΔT (°C) is temperature rise of cooling water in the mold (usually 4.5 – 5.5 °C) at volumetric flow rate 28 – 33 (dm^3 s^{-1}) for the billet □200.

- SC (under nozzles) $-\lambda\dfrac{\partial T}{\partial \boldsymbol{n}}=h_{\text{jet}}\left(T-T_{\text{jet}}\right)+\sigma\varepsilon\left(T^4-T_{\text{sec}}^4\right)$,
- SC (outside nozzles) $-\lambda\dfrac{\partial T}{\partial \boldsymbol{n}}=h_{\text{sec}}\left(T-T_{\text{sec}}\right)+\sigma\varepsilon\left(T^4-T_{\text{sec}}^4\right)$,
- SC (under rolls) $-\lambda\dfrac{\partial T}{\partial \boldsymbol{n}}=h_{\text{roll}}\left(T-T_{\text{roll}}\right)$,
- TC $-\lambda\dfrac{\partial T}{\partial \boldsymbol{n}}=h_{\text{ter}}\left(T-T_{\text{ter}}\right)+\sigma\varepsilon\left(T^4-T_{\text{ter}}^4\right)$,

where the quantities with approximate values are listed in Table 12.4

Quantity	Description	Approx. values	Unit
ε	Emissivity of the casting.	0.75 – 0.85	-
h_{jet}	Heat transfer coefficient of the „soft“ cooling water nozzle (minus thermal radiation), (Kotrbáček (2010)).	250 – 450	W m^{-2} K^{-1}
h_{jet}	Heat transfer coefficient of the „hard“ cooling water nozzle (minus thermal radiation), (Kotrbáček (2010)).	750 – 1250	W m^{-2} K^{-1}
h_{roll}	Heat transfer coefficient between the casting and support rollers (contact pressure dependent value) in SC, (Lin Kuan-Ju (2002)).	5000	W m^{-2} K^{-1}
h_{ter}	Heat transfer coefficient for natural convective cooling in TC.	10 – 15	W m^{-2} K^{-1}
T_{jet}	Temperature of cooling water sprayed through nozzles.	25 – 30	°C
T_{roll}	Surface temperature of the support roller in contact with the casting in SC.	500 – 700	°C
T_{sec}	Temperature in the chamber of SC.	200 – 300	°C
T_{ter}	Ambient temperature in TC.	20 – 30	°C

Table 12.4 Descriptions of quantities in the boundary conditions with approximate values.

12.4 Model in the ProCAST Software Environment

When using realistic boundary conditions and temperature dependent material properties, it is necessary to solve the mathematical problem numerically. Online monitoring systems of CCMs use numerical models to predict transverse and longitudinal temperature profiles, length and shape of the liquid core, etc. The computation algorithms are usually based on the explicit finite difference method.

In our case, one casting strand for a billet □200 in the CCM Vítkovice is governed by a 3D finite element model. The total length of the casting strand starting at the meniscus down to the cutting torches is approximately 19 m. Volume discretization of the casting strand uses about 800 000 finite volume elements of size approx.

1 × 1 × 1 cm. The computation of the transient response of the temperature field considering phase changes and no melt flow lasts for several hours in ProCAST, depending on the chosen approach (*solidtransport* or *travelling BC*), numerical solver settings and computer hardware configuration.

We need to achieve a significant reduction in the computational time for the identification of dynamics and temperature field control using the co-simulation approach. A significant simplification is the central symmetry plane of the casting, making possible the reduction of a full 3D computational domain down to 2D. Alternatively, it is feasible to reduce the computational domain up to the end of the SC area only and no further. These modifications may reduce the computation time of a new steady-state to about 25 minutes (Intel Xeon X5450 CPU @ 3.00 GHz, 8 GB RAM, Windows XP 64bit). It should be also noted that a symmetrical distribution of cooling intensity is assumed for the cooling nozzles.

Modelled length of the casting	Central symmetry	FEM mesh	Element type
14.17 m	yes	mapped	linear

Table 12.5 FEM mesh parameters used in numerical model.

12.4.1 Material Properties of Selected Steel Grades

The dynamics of the thermal system is given by heat conductivity, capacity and material density. The following Table 12.6 summarizes three steel grades of different carbon content, used in numerical models.

Steel / carbon content	Chemical composition (element wt.%)	T liq / T sol (°C)	Solidification interval ΔT = Tliq – Tsol
A / low	C 0.06, Si 0.01, Mn 0.38, S 0.35, Cr 0.022	1533 / 1509	24
B / moderate	C 0.18, Si 0.25, Mn 0.54, Cr 1.00, Ni 0.15	1515 / 1425	90

C / medium	C 0.38, Si 0.23, Mn 0.64, Cr 0.99, Ni 0.08	1498 / 1417	81

Table 12.6 Chemical composition of selected steel grades.

Liquidus and solidus temperatures were computed in the generator of thermodynamic properties in ProCAST, using the chemical composition. The results of this computation are: curve of specific enthalpy $H(T)$ or specific heat capacity $c_p(T)$, thermal conductivity $\lambda(T)$, density $\rho(T)$ and fraction of solid $fs(T)$.

It is evident that the solidification interval for higher carbon content steel is wider. Also, the casting speed and cooling intensity should be adjusted to respect this phenomenon. Modelling of solidification kinetics is performed by the *Scheil* or *Back-diffusion* approach with a known cooling rate. The following graph shows the difference between the computed thermal conductivity properties at equilibrium (Lever) and non-equilibrium (Scheil) solidification conditions.

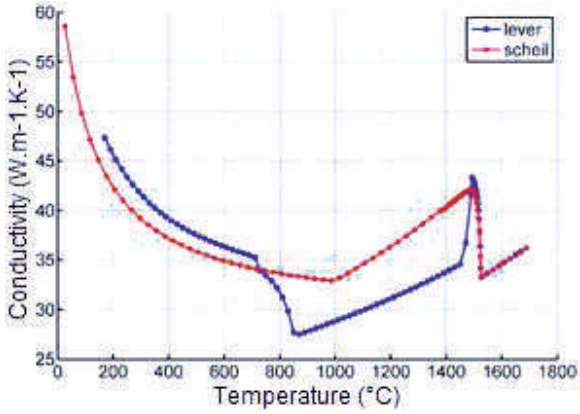


Fig. 12.4 Thermal conductivity of the steel grade „A / low" computed for different solidification conditions.

12.4.2 Heat Transfer Coefficient

Exact computations of the temperature profile on the casting surface require a knowledge of the heat transfer coefficient for different SC nozzle arrangements. Since the heat transfer coefficient is a function of several quantities (such as: position, temperature and shape of sprayed surface, temperature and speed of droplets, etc.) a general governing equation does not exist. The most exact method remains an inverse calculation from the measured heat fluxes, during an experiment. The main advantage

is the fidelity of the heat transfer coefficient distribution on the casting surface, however, the experiments are expensive and time consuming to carry out. Equations for approximate calculations can be found in different papers, e.g. in Petrus (2010)

$$h = A Q_{\mathrm{w}}^{c} \left(1 - b T_{\mathrm{w}}\right) \tag{12.4}$$

where the heat transfer coefficient is denoted as h, Q_{w} is surface density of cooling water flow rate ($\mathrm{dm^3\ s^{-1}\ m^{-2}}$) and T_{w} is water temperature. Constants A, b, c are determined by the actual nozzle arrangement. The following Table 12.7 shows one possible SC nozzle arrangement for a □200 billet on the CCM Vítkovice.

Type	**Cone angle**	**Distance from the casting surface**	**Nozzle position / Casting speed**	**Water flow rate**	**Water pressure**
full water cone	60°	180 mm	perpendicular / 1 m $\mathrm{min^{-1}}$	1.28 $\mathrm{dm^3\ min^{-1}}$	1.93 bar

Table 12.7 SC nozzle arrangement [With permission of ŽP Research-development center].

The following Figure 12.5 shows heat transfer coefficient distribution measured in experiment where a steel surface of temperature 1250 °C is sprayed with a nozzle. (in arrangement defined in Table 12.7)

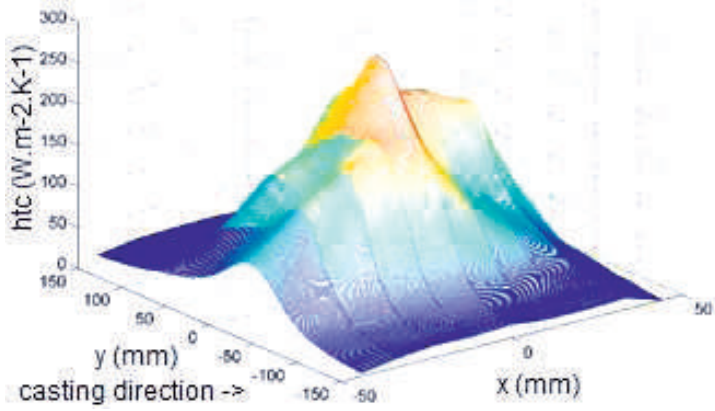


Fig. 12.5 Heat transfer coefficient distribution on the casting surface (minus thermal radiation) [With permission of ŽP Research-development center].

12.5 Measurement of the Surface Temperatures on the Casting

The constant movement of the casting naturally does not allow contact temperature measurements with thermocouples. Due to an enormous amount of water steam in the chamber of the SC and the varying emissivity of the casting, a standard (one color) thermovision may give erroneous results. This problem is partially eliminated by two color pyrometers. They are based on a ratio of emitted energies in two separated wavelength bands, assuming that the value of emissivity in both (narrow) bands is identical. Emissivity is not the only parameter that influences the reliability of non-contact temperature measurement. Detailed analyses of criteria on wavelength bands selection, band pass filters, interfering reflections and/or thermal radiation absorption can be found in technical papers on measurement devices, (Möllmann (2011)).

For the validation purposes of the numerical model of a given casting strand in CCM Vítkovice, a two-color pyrometer designed for continuous operation in demanding conditions has been utilized. Raytek Marathon 1 pyrometer series has a measuring range of 700–1800 °C. Some wavelength bands are out of the absorption spectrum of the water steam. It is necessary to insert the device into a protective shield with water cooling, when using this pyrometer in the chamber of SC. The protected device resists temperatures up to 315 °C. Also, an air purge collar can be used to keep the optics clean. The device is equipped with automatic signalization in case the optics get contaminated. Communication with the device is bi-directional using the RS485 interface. The device can be also used in one-color operating mode, when the measured value can be compared to a thermal imaging camera; or the emissivity of an object at a known temperature can be determined.

12.6 Casting in the Secondary Cooling Zone as DPS

The non-stationary surface temperature field $T(\boldsymbol{x},t)$ of the casting in SC represents the distributed output $YN(\boldsymbol{x},t)$. The surface is cooled by a system of nozzles–distributed input generators, whose lumped inputs are cooling water flow rates $\{UN_i(t)\}_i$ in individual zones of the SC. Dynamic relation between $\{UN_i(t)\}_i$ and

$YN(\boldsymbol{x},t)$ represents nonlinear LDS given on the spatial definition domain Ω defined by the casting.

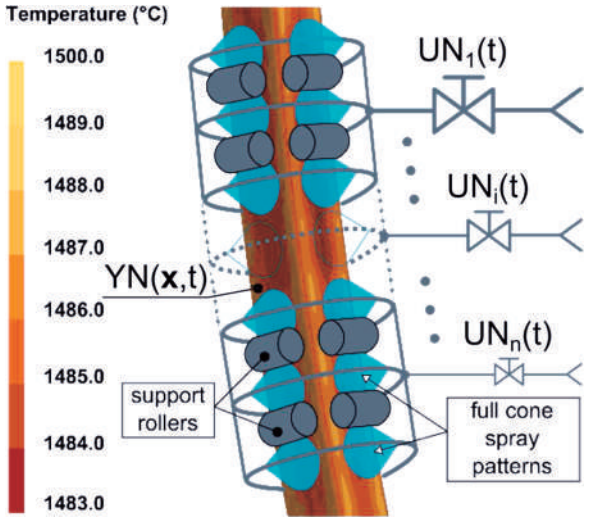


Fig. 12.6 The secondary cooling zone of continuous casting machine as a controlled lumped inputs $\{UN_i(t)\}_i$ *and distributed parameter output* $YN(\boldsymbol{x},t)$ *system.*

Lumped inputs $\{UN_i(t) = UN_i(\infty) + U_i(t)\}_{i=1,n}$ actuate in zones of the SC and generate changes of the boundary conditions on the surface of the casting $\partial\Omega$ (boundary of the computational domain) at chosen steady-state operation mode given by $\{UN_i(\infty)\}_i$ and $YN(\boldsymbol{x},\infty)$. Let us consider a linear response of the temperature field (as controlled system) to the lumped inputs in a linearized neighborhood of a selected operating mode for the purposes of designing a controller. The overall response of the system with respect to the selected operating mode can be expressed in terms of the superposition

$$Y(\boldsymbol{x},t) = \sum_{i=1}^{n} Y_i(\boldsymbol{x},t) \tag{12.5}$$

where $Y_i(\boldsymbol{x},t)$ is the *i*-th system response to $U_i(t)$ lumped input. The analysis of the discrete-time responses $\{Y_i(\boldsymbol{x},k)\}_{i=1,5}$ to the discrete step changes of the cooling water flow $\{U_i(k)\}_{i=1,5}$ was performed by a 2D finite element model in 5 locations. The distributed transient responses $\{\mathcal{H}H_i(\boldsymbol{x},k)\}_{i=1,5}$ to the discrete step changes of the inputs $\{U_i(k) = 1(k)\}_{i=1,5}$ are obtained as responses of the system with a zero-order hold H.

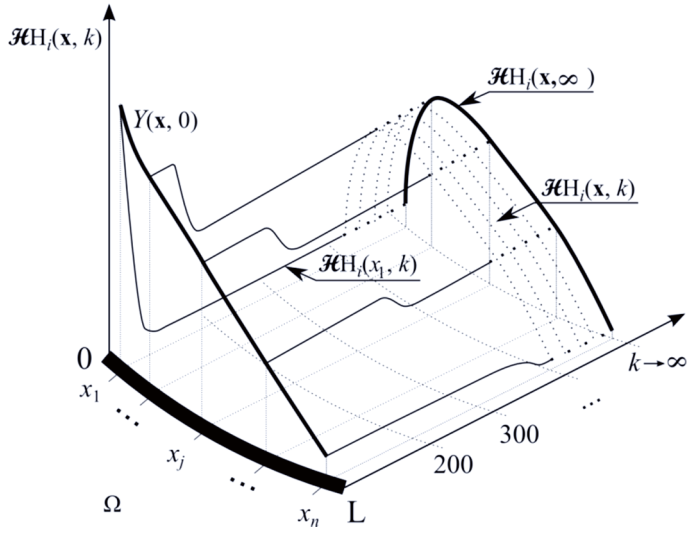


Fig. 12.7 Distributed transient response to the cooling water flow rate change in 1st. zone.

Let us decompose the dynamics of the identified system into time and space components, based on numerically computed transient and impulse characteristics:

- Let us choose locations $\{\boldsymbol{x}_i\}_{i=1,5} \subseteq \Omega$ from the distributed transient characteristics $\{\mathcal{H}H_i(\boldsymbol{x},k)\}_{i=1,5}$ such, that the partial transient characteristics at those locations will achieve their maximal amplitudes $\{\mathcal{H}H_i(\boldsymbol{x}_i,k)\}_{i=1,5}$ in steady-state. Let us introduce the reduced responses of these characteristics $\{\mathcal{H}HR_i(\boldsymbol{x},\infty)\}_{i=1,5}$ in steady-state, based on Eq. (3.6).

- Assign continuous or discrete transfer functions $\{S_i(\boldsymbol{x}_i,s)\}_{i=1,5}$, resp. $\{SH_i(\boldsymbol{x}_i,z)\}_{i=1,5}$, to this partial transient response, representing the time domain dynamics of the identified system.

The control task in the CC process is the rejection of disturbance, as well as setting the desired temperature field in the neighborhood of the selected operating mode.

The components of the vector of references expressing the desired thermal distribution of the cast are represented by five secondary cooling zones, where $\breve{W}_i(k) = \left\{\breve{W}_i(\boldsymbol{x}_i, k)\right\}_{i=1,5}$ is computed according to

$$\begin{aligned} &\min_{W_i} \left\| W(\boldsymbol{x}, \infty) - \sum_{i=1}^{5} W_i(\boldsymbol{x}_i, \infty)\, \mathcal{H}HR_i(\boldsymbol{x}, \infty) \right\| = \\ &= \left\| W(\boldsymbol{x}, \infty) - \sum_{i=1}^{5} \breve{W}_i(\boldsymbol{x}_i, \infty)\, \mathcal{H}HR_i(\boldsymbol{x}, \infty) \right\| \end{aligned} \tag{12.6}$$

The references obtained during the process of control synthesis are then compared with the values of the partial distributed outputs $\bar{Y}_i(\boldsymbol{x}_i, k) = \left\{\breve{Y}_i(\boldsymbol{x}_i, k)\right\}_{i=1,5}$, computed by the relation (7.5)

$$\bar{Y}_i(\boldsymbol{x}_i, k) = \bar{Y}R_i(\boldsymbol{x}_j, k)^{-1}\, \bar{Y}(\boldsymbol{x}_i, k) \tag{12.7}$$

which will yield the finite-dimensional control errors $\left\{\breve{E}_i(\boldsymbol{x}_i, k)\right\}_{i=1,5}$. The values of the thermal field $Y(\boldsymbol{x}_i, k)$ are obtained using the formerly mentioned pyrometers that are placed in locations $\{\boldsymbol{x}_i\}_{i=1,5} \subseteq \Omega$. In these locations the reduced partial distributed outputs $\left\{YR_i(\boldsymbol{x}_j, k)\right\}_i$ result in the elements of the matrix $\bar{Y}R_i(\boldsymbol{x}_j, k)$, based on relation (3.26).

12.7 Analysis of Temperature Field Dynamics

The SC was divided into 5 or 4 zones in the dynamics analysis and simulation tasks, each containing a number of nozzles. This is the SC used for the □200 billet in CCM Vítkovice. Finer division of zones used in modern CCMs allows one to set a desired cooling profile even more precisely. As a result, it is possible to achieve a higher quality, but on the other side, it can increase the complexity of SC control.

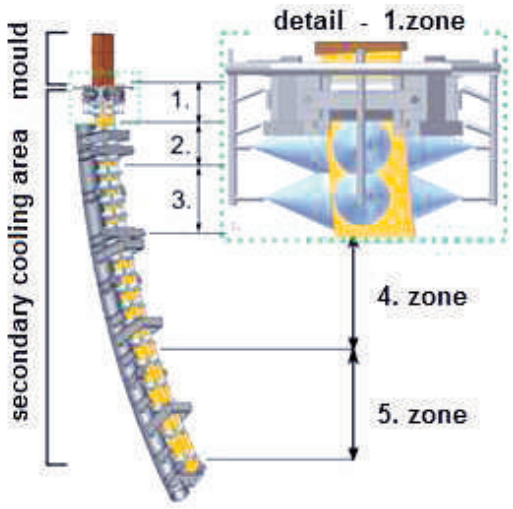


Fig. 12.8 Proposed division of SC into 5 independent zones.

Fig. 12.9 depicts the positions of virtual pyrometers. Due to the structural configuration of the SC, the pyrometers are placed right behind the last nozzle of each zone. Surface temperatures measured in locations $\{\boldsymbol{x}_i\}_{i=1,5} \subseteq \Omega$ will be denoted as {P1, P2, P3, P4, P5} in our discussion. The arc lengths between the points and meniscus on the inner radius are [0.98, 1.62, 2.80, 4.22, 6.08] meters.

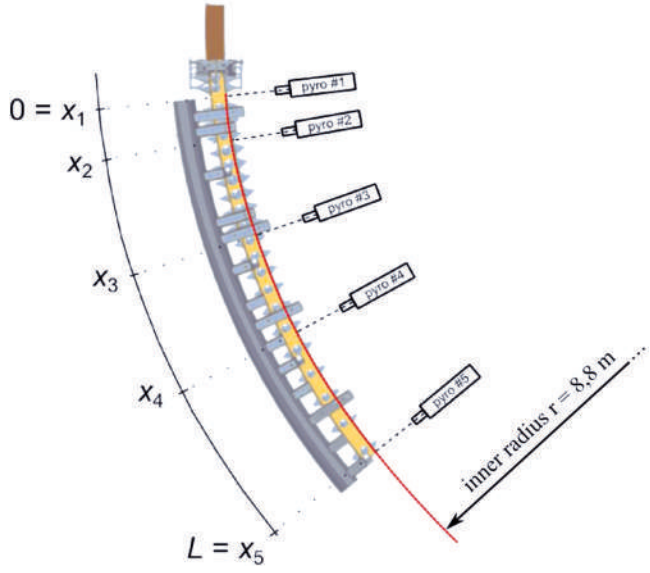


Fig. 12.9 Virtual pyrometers measuring temperatures on inner radius of the casting strand in locations P1 – P5.

The cooling intensity of the casting is given by the total amount of cooling water per 1 kilogram of steel ($dm^3\ kg^{-1}$), which corresponds to the cooling water flow rate,

divided into the zones. The division of the flow rate is unequal, because it takes the growing shell thickness of the casting into account. The flow rates are classified as process quantities, which, together with others, define an operating mode of the CCM. The water flow rates in individual zones are classified as inputs in the temperature field control of the SC.

12.7.1 The Operating Mode of the Continuous Casting Machine

A constant casting speed (1.07 m min^{-1}), cooling intensity and corresponding cooling water flow rates in the zones are prescribed within the narrow temperature interval (1548–1552 °C) in a tundish for a steel grade with low carbon content C < 0.10 wt. % of the profile □200 . In this operating mode, the rate of the tundish discharge is adapted to deliveries of new steel; overheating compensates heat losses and the length of the liquid core and quality of the castings are within the limits. From the viewpoint of control, we characterize this state as an optimal (or nominal) operating mode. In case any of the process quantities are out of the specified interval, it is necessary to react correctly. For example, if the temperature of the steel in the tundish (casting temperature) drops below 1548 °C, it is necessary to increase the casting speed to avoid the „freezing" of outflow holes. On the contrary, at higher temperatures, the solid shell of the casting becomes thinner, and the casting speed should be decreased, (Fig. 12.10).

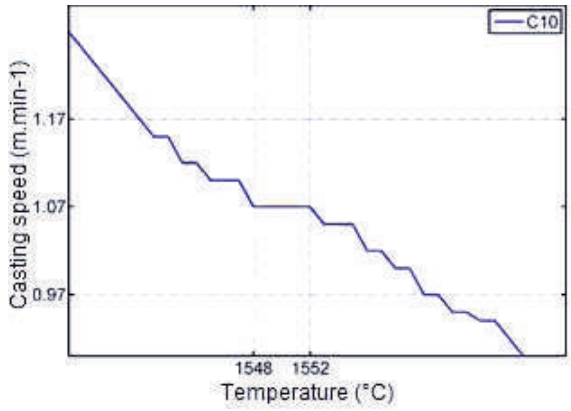


Fig. 12.10 Casting speed settings according to the casting temperatures for low carbon steel C < 0.1 wt. %, □200.

Proper settings of the process quantities rely on empirical knowledge of operators at a great extent. Numerical models help us establish an exact relation between process quantities, energy flows and the temperature field. Figure 12.11 shows heat fluxes in

three zones (operating mode according to Table 12.8.). The overall cooling performance is dominated by radiation.

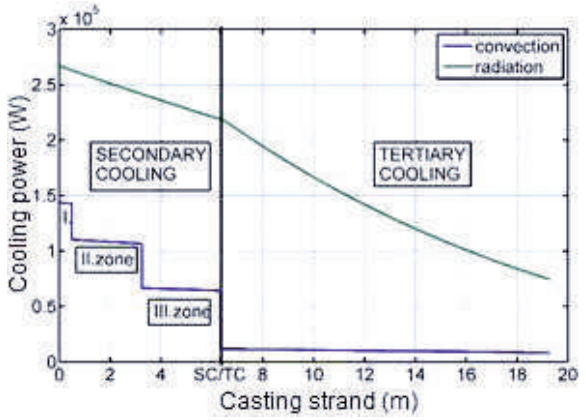


Fig. 12.11 Cooling performance ratios in the selected operating mode of CCM Vítkovice.

Control objectives are based upon analyses of achievable CCM operating modes, near which the process is controlled. There is a large number of theoretically achievable operating modes. We may assign a short-time min-max range of casting speeds to each casting temperature (vertical lines). We get more than a 1000 possible operating modes with the resolution of 0.01 m min^{-1} for □200 in CCM Vítkovice, (Fig. 12.12), where the smooth curve is a cubic fit of the casting speed settings in Fig. 12.10.

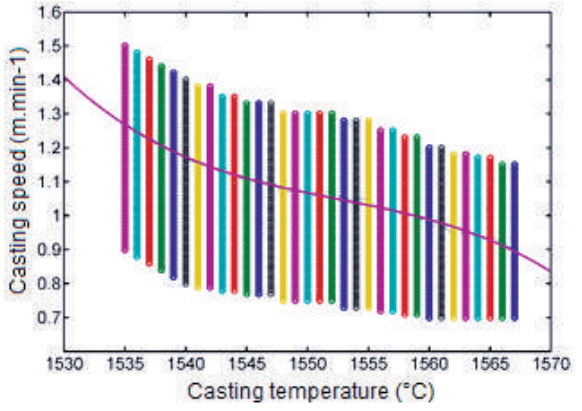


Fig. 12.12 Min-max ranges of the casting speeds for different casting temperatures.

The nominal operating mode for billet □200 in CCM Vítkovice is defined in the following Table 12.8.

Material	low carbon (C < 0.1 wt. %)

Profile / Rounding of an exterior corner	□ 200 / R8 (mm)
Casting temp. / Casting speed	1552 °C / 1.07 m min^{-1}
Cooling intensity in the SC	0.31 dm^3 kg^{-1}

Table 12.8 Nominal operating mode for □200 in CCM Vítkovice.

We can identify the

- mold area, characteristic of a sharp surface temperature decrease,
- SC characteristic of a „saw tooth“ profile caused by discontinuous cooling,
- TC characteristic of a monotonic temperature decrease

in the temperature profile on the inner radius referring to the nominal operating mode.

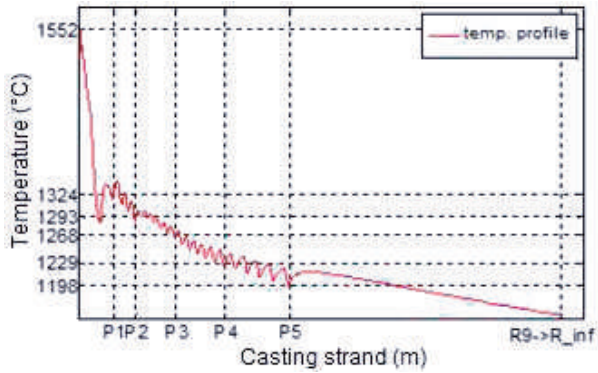


Fig. 12.13 Steady-state temperature profile on the inner radius in the nominal operating mode.

The desired change of the temperature field is defined with respect to the steady-state temperature profile on the inner radius.

12.7.2 Change of Casting Speed

The analysis of any CCM operating mode requires a quantification of changes in the temperature field with respect to the casting speed (VCAST). The following Fig. 12.14 shows the change of the temperature profile on the inner radius, after a positive step change of the casting speed.

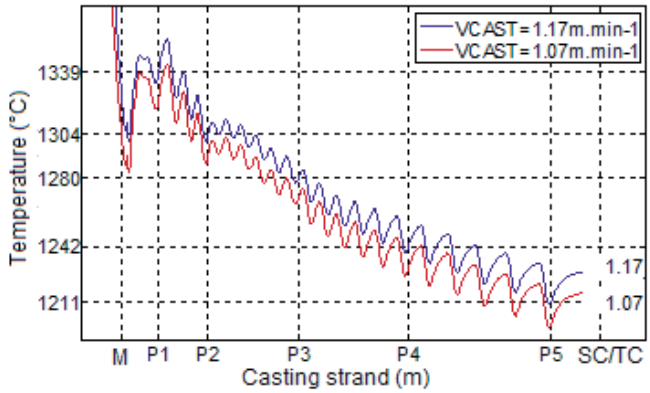


Fig. 12.14 Steady-state temperature profiles on the inner radius for casting speeds 1.07 m min^{-1} (down) and 1.17 m min^{-1} (up)

Temperature profile rises or falls along the length of the casting within the bounds of the changes in the casting speed. Non-linearity occurs in different magnitudes of change in the temperature field with respect to the direction of the casting speed change. The nonlinearity of the process is influenced mainly by the character of the technology. However, simulation studies imply that it is possible to assume linearity when the casting speed is a fixed and given parameter. The following Table 12.9 summarizes initial and final temperatures in locations P1–P5 before and after a positive change of the casting speed (1.07→1.17 m min^{-1}). Initial temperatures refer to the nominal operating mode in Table 12.8.

VCAST	P1 (°C)	P2 (°C)	P3 (°C)	P4 (°C)	P5 (°C)
1.07→1.17	1324→1339	1293→1304	1268→1280	1229→1242	1198→1211

Table 12.9 Initial and final temperatures in locations P1 – P5, before and after a positive change of the casting speed.

12.7.3 Change of Cooling Intensity

The analysis of the temperature field dynamics in response to the step changes of water flow rates is rather important, because proper flow rates in secondary cooling zones must be set during the change of casting speed. The following figure shows steady-state temperature profiles on the inner radius. Also, the positions of virtual pyrometers (P1 – P5) are marked.

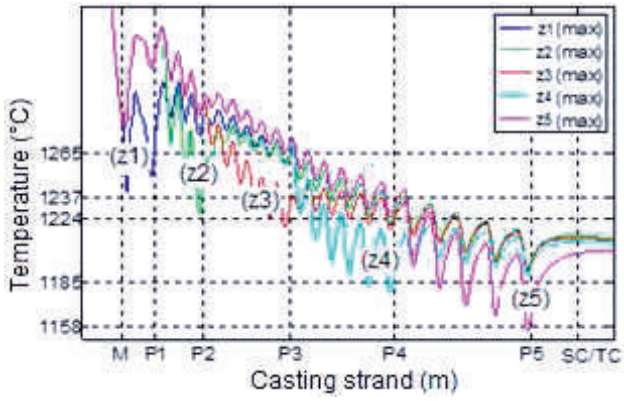


Fig. 12.15 Steady-state temperature profile after flow rate step changes.

Time dynamics of the response to the positive flow rate change in the first zone is shown in the Fig. 12.16.

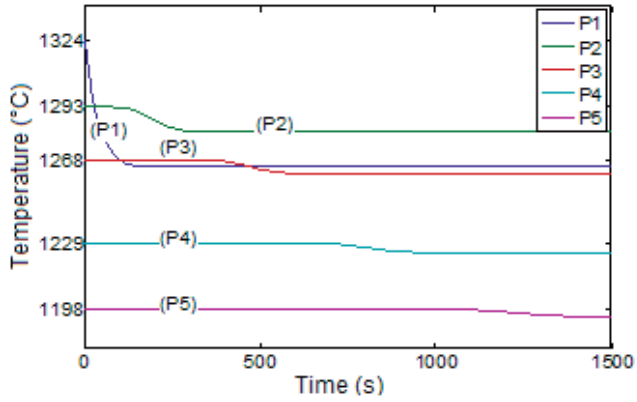


Fig. 12.16 Time dynamics of the temperature field in the points P1 – P5.

The initial and final temperatures in locations P1 – P5 after a positive change of the flow rates are listed in Table 12.10.

Zone	P1 (°C)	P2 (°C)	P3 (°C)	P4 (°C)	P5 (°C)
z1	1324→1265	1293→1281	1268→1261	1229→1224	1198→1195
z2	-	1293→1237	1268→1261	1229→1225	1198→1195
z3	-	-	1268→1224	1229→1221	1198→1193
z4	-	-	-	1229→1185	1198→1190
z5	-	-	-	-	1198→1158

Table 12.10 Initial and final temperatures in locations P1 – P5 after positive flow rate changes.

12.7.4 Identification of the System Dynamics

Let us get the distributed transient characteristics of the plant by changing the flow rate of cooling water in the individual sections of the secondary cooling zone, given a linearized neighborhood of the selected operating mode, or in other words at a fixed casting speed (Fig. 12.17).

Normed distributed transient characteristics are obtained from temperature field responses to the positive or negative flow rate changes in the zones of SC. These constitute the basis for a numerical structure H in the HLDS block, partial distributed transient characteristics (in locations P1 – P5) are selected from these as well. Depending on their shape, we assign them models of dynamic systems of the 1st to *n*-th order. Partial transient characteristic $\mathcal{H}H_1(P_1,t)$ in location P1 obtained from the step response in the 1st zone is represented by a dashed line in Fig. 12.17 (first from the right).

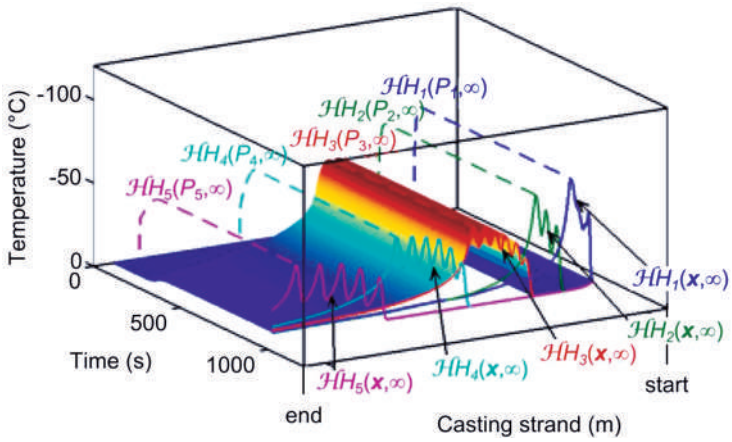


Fig. 12.17 Distributed transient characteristics.

start - end (m)	- overall length of the secondary cooling zone.
0-1200 (s)	- settling time interval of temperature field.
$\{\mathcal{H}H_i(x,t)\}_i$	- transient characteristics of temperature field at cooling of semi-products.
$\{\mathcal{H}H_i(P_i,t)\}_i$	- partial transient characteristics in points $\{P_i\}_i$.
Temperature	- changes of temperature field relative to the profile in the chosen steady-state mode.

Let us assign dynamic models in the form of 1st order transfer functions to the selected partial transient characteristics in locations P1 – P5

$$G(s) = \frac{K_p}{T_p s + 1} \tag{12.8}$$

Time constants T_p and gains K_p of these transfer functions are listed in the following table.

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$	$G_5(s)$
Gains	–59.4	–55.6	–43.7	–44.6	–40.4
Time constants	8.9	9.4	20.5	26.5	31.0

Table 12.11 Identified gains K_p and time constants T_p (negative sign due to cooling effect).

The spatial components of the temperature field dynamics $\{\mathcal{H}HR_i(x,\infty)\}_{1,5}$ are obtained from the steady-state transient characteristics using Eq. (3.6).

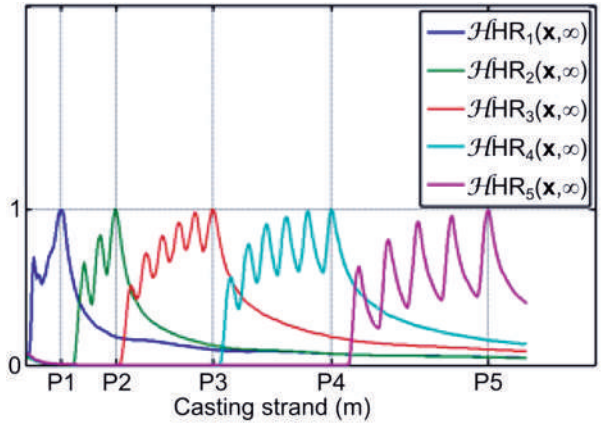


Fig. 12.18 Space components of the dynamics.

12.8 Control Tasks

In the case of the simulation study, feedback control was meant to solve two characteristic issues:

- Distributed output (temperature field) tracks changes of the distributed reference (*set-point tracking*).

- Distributed output (temperature field) is maintained constant during the presence of a disturbance (*disturbance rejection*).

For the purposes of our simulation study, a control system is assembled based on the control feedback loop in Fig. 7.8, in accordance with the diagram in Fig. 12.19. PI-1 controllers used in the *Time Synthesis* block are described by Huba (2007).

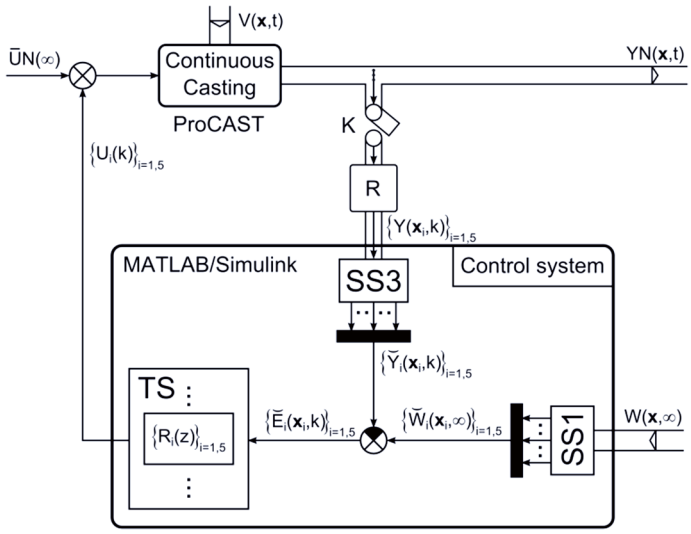


Fig. 12.19 Control loop diagram.

Continuous casting - controlled system with lumped input and distributed output, the block MtoV. (Fig. 7.9).

TS - temporal component of the control synthesis.

SS1 - spatial component of the control synthesis.

SS3 - transformation of vectors.

K - sampling.

R - block of differences and reconstruction.

$\left\{\breve{E}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$ - components of the partial control error vector.

$\left\{R_i\left(z\right)\right\}_i$ - Z-images of the discrete transfer functions of the controllers.

$\left\{U_i\left(k\right)\right\}_i$ - components of the discrete lumped input vector.

$\bar{U}N\left(\infty\right)$ - vector of steady-state inputs.

$V\left(\boldsymbol{x},t\right)$ - continuous distributed disturbance.

$W\left(\boldsymbol{x},\infty\right)$ - overall distributed reference in the steady state.

$\left\{\breve{W}_i\left(\boldsymbol{x}_i,\infty\right)\right\}_i$	- components of the discrete lumped reference in the steady-state.
$YN\left(\boldsymbol{x},t\right)$	- distributed manipulated variable.
$\left\{Y\left(\boldsymbol{x}_i,k\right)\right\}_i$	- component of measured manipulated variables on the HLDS level as output of block R.
$\left\{\breve{Y}_i\left(\boldsymbol{x}_i,k\right)\right\}_i$	- components of the partial distributed manipulated variable.

The control loop diagram above contains the co-simulation block (Matlab & Simulink and ProCAST) and it is launched after each ProCAST time step. Therefore, one needs to set in the *Configuration parameters* of the Simulink model:

- Solver: Discrete (no continuous states),
- Type: Variable step.

The adjustable parameter of the controllers in the time synthesis block is $\alpha = 0.27$.

12.8.1 Change of the Distributed Reference

The control task is to track the step change of the distributed reference $W\left(\boldsymbol{x},k\right)$ at the level $YN\left(\boldsymbol{x},\infty\right)+W\left(\boldsymbol{x},k\right)$ in some time instant *k,* by using the inputs $\bar{U}\left(k\right)$, while the steady-state operating mode is given by input – output pair $\bar{U}N\left(\infty\right)$ – $YN\left(\boldsymbol{x},\infty\right)$. Here, $W\left(\boldsymbol{x},k\right)$ refers to a negative step change of the constant temperature field by some value. The distributed output $YN\left(\boldsymbol{x},t\right)$ is the temperature field in the SC and the lumped inputs $\bar{U}\left(k\right)$ are cooling water flow rates changes in the zones of SC. At the end of the control process $YN\left(\boldsymbol{x},t\right)_{t\to\infty}=YN\left(\boldsymbol{x},\infty\right)+Y\left(\boldsymbol{x},\infty\right)$, where the quadratic norm of the distributed control error $\left\|E\left(\boldsymbol{x},\infty\right)\right\|=\left\|W\left(\boldsymbol{x},\infty\right)-Y\left(\boldsymbol{x},\infty\right)\right\|$ needs to be minimal.

Let us consider a uniform negative step change (cooling) of the steady-state temperature field by 30 °C (*const* → *const* – 30) at *t*=100 seconds. The following figures show the control process variables in co-simulation mode. Signals of lumped inputs are intentionally corrupted by a random noise.

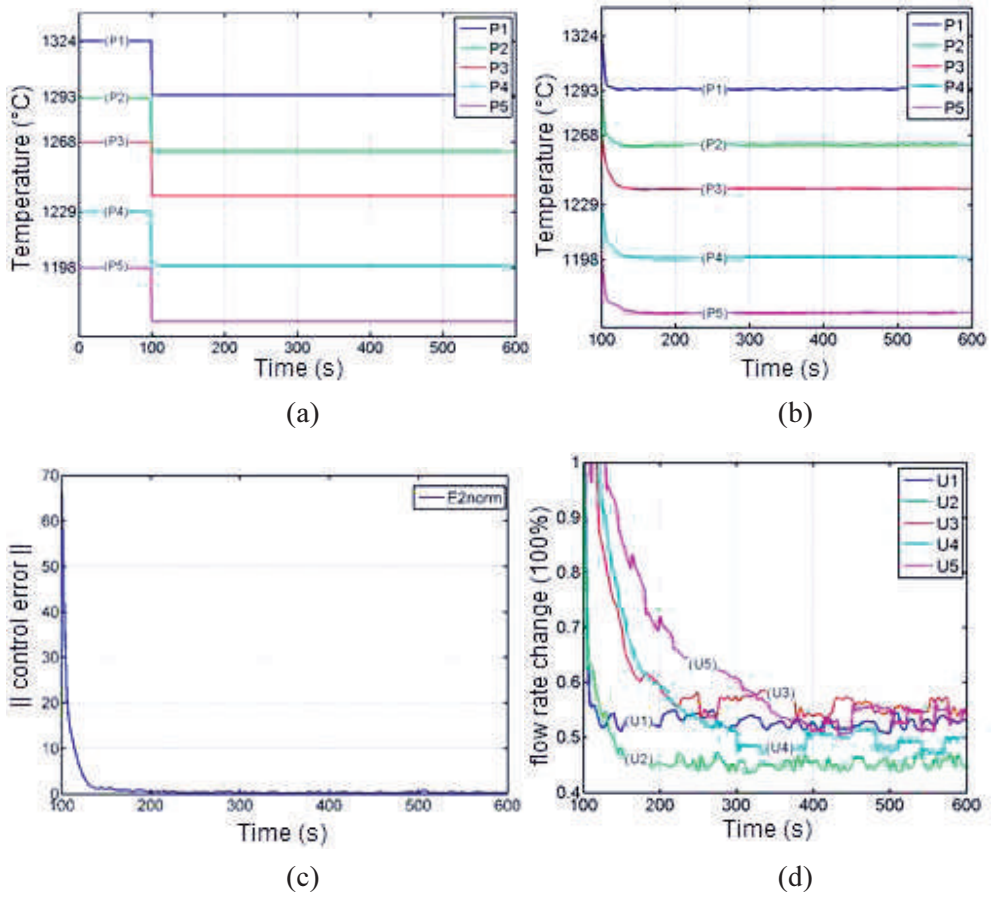


Fig. 12.20 Change of distributed reference scenario, (a) distributed reference, (b) distributed output, (c) quad. norm of control error, (d) lumped inputs (water flow rates).

The time-optimal control causes the saturation of lumped inputs on a single time interval. The hydraulic valves that control flow rates in the zones of SC react immediately. Due to surface undercooling by 30 °C, the center line mushy-zone in the casting shortens by 0.5 m (simulation result).

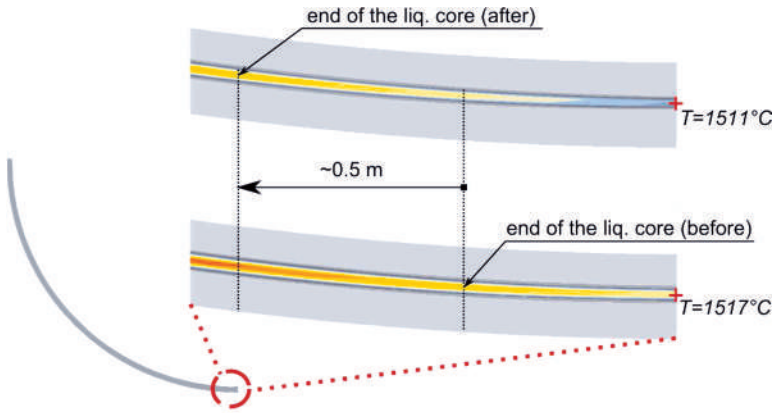


Fig. 12.21 Liquid core at the last unbending point (transition into infinite radius) before and after controlled surface cooling.

12.8.2 Distributed Disturbance Rejection

The control task here is to reject the effect caused by the distributed disturbance $V(\boldsymbol{x},t)$, a step change in a given time instant *k*, using the lumped inputs $\bar{U}(k)$. The distributed output $YN(\boldsymbol{x},t)=YN(\boldsymbol{x},\infty)+Y(\boldsymbol{x},t)$ is the temperature field in the SC, and the lumped inputs $\bar{U}(k)$ are cooling water flow rates in the zones of the SC. The quadratic norm of the distributed control error $E(\boldsymbol{x},\infty)=\left\|V(\boldsymbol{x},\infty)-\breve{Y}(\boldsymbol{x},k)\right\|$ should remain minimal during the control process.

Let us consider a positive casting speed change by 0.1 m min^{-1} (1.07→1.17 m min^{-1}) at t=100 seconds. The effect of this change is an overall increase of temperatures in the casting given by the distributed disturbance $V(\boldsymbol{x},t)$. The control process is shown in co-simulation mode in the following figures. Signals of lumped inputs are corrupted intentionally by random noise.

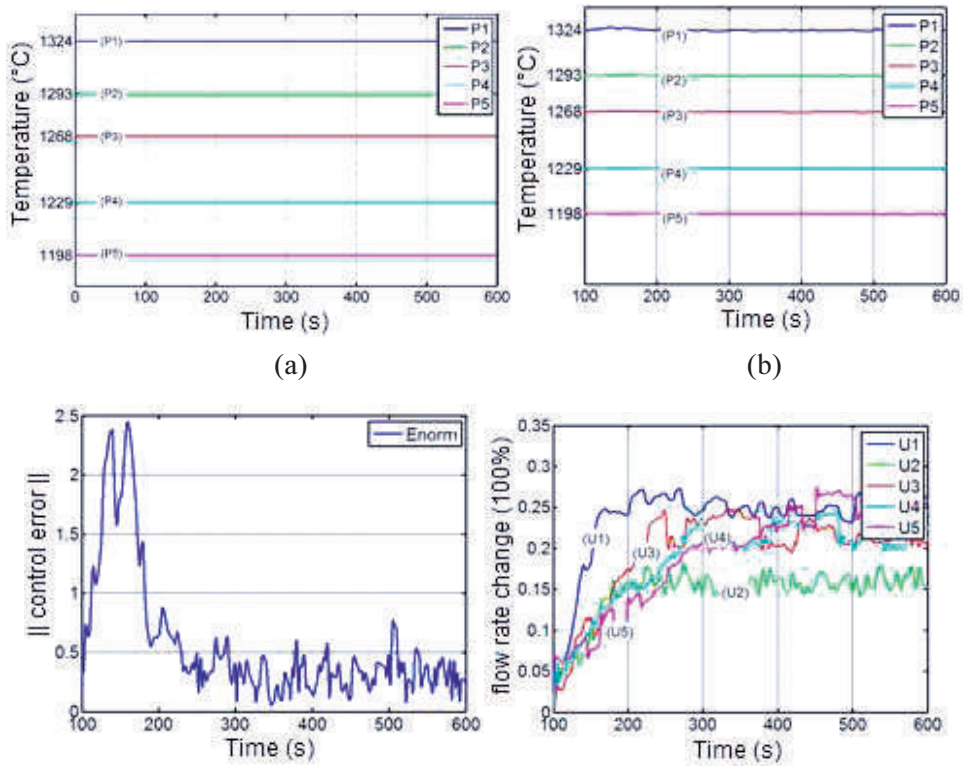


Fig. 12.22 Rejection of the distributed disturbance effect, (a) distributed reference, (b) distributed output, (c) quad. norm of control error, (d) lumped variables.

Due to the increase of casting speed by 0.1 m min^{-1}, the center line mushy-zone of the casting liquid core elongates by approx. 1.2 m (simulation result).

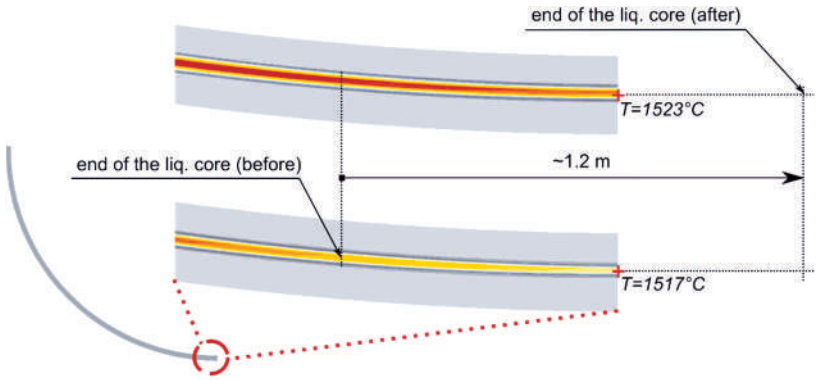


Fig. 12.23 Liquid core at the last unbending point (transition into infinite radius) before and after casting speed increase.

The following figure compares three different temperature transients from P1 (up) to P5 (down) after the step change in distributed error. The temperature responses vary upon the reactions to the change.

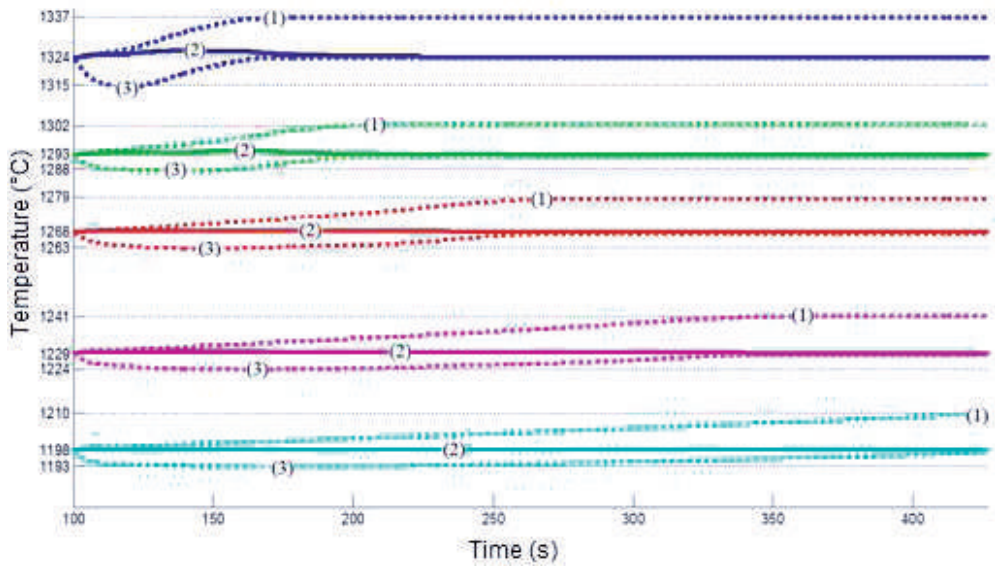


Fig. 12.24 Temperature responses in locations P1 – P5, (1) no reaction; (2) with control; (3) immediate reaction (manual setting).

The cooling intensity is increased when the casting speed is increased. In terms of process safety, it is a necessary reaction, since the weakening of the solid shell causes a breakout. In terms of the steel solidification process, we induce new thermodynamic conditions (Fig. 12.25 and Fig. 12. 26). Similarly to the previous case studies, let us note that the figures featured here were created using the development system featured in this book (Fig. 7.9) and show the course of the manipulated variable with the precision of the validated ProCAST numerical model.

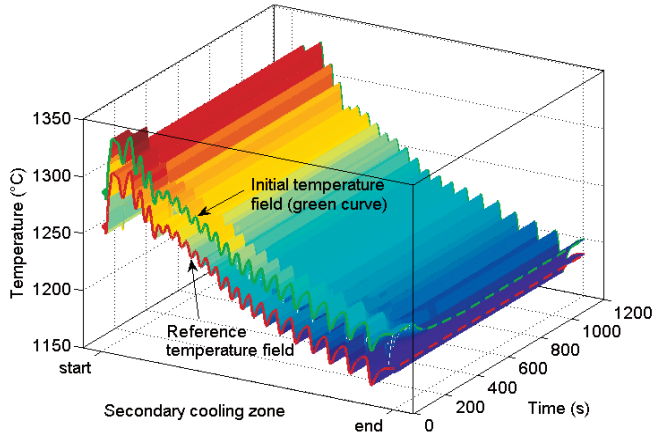


Fig. 12.25 Beginning of the control process, where the increase of casting speed causes an increase in the surface temperatures of the continuous cast in the secondary cooling zone.

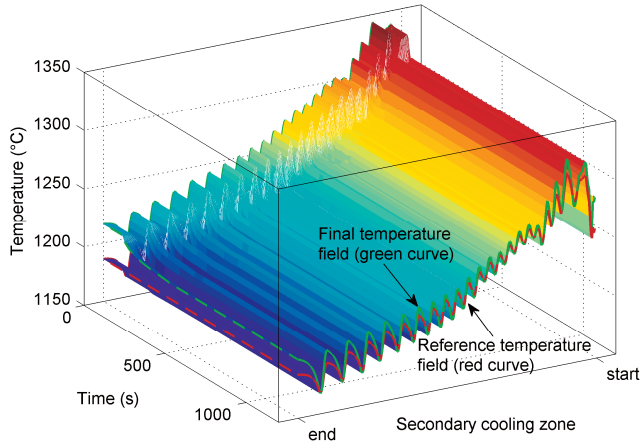


Fig. 12.26 Final result of the control process, where the thermal field achieves the reference level within a given precision for the steady-state operating mode of the CCM.

12.9 Adaptive Control

In *12.8.2 Distributed disturbance rejection* the control feedback loop is set to an operating mode with a casting velocity of 1.07 m min^{-1}. This means, that the dynamic characteristics in both the temporal and the spatial relation are acquired based on the distributed transient responses for this fixed casting velocity. Given a disturbance on the thermal field caused by a 0.1 m min^{-1} change in the casting velocity, the feedback loop kept the prescribed thermal profile of the secondary cooling zone successfully,

based on the characteristics that belong to a casting speed of 1.07 m min^{-1}. Accordingly, we will now segment the dynamics of the secondary zone to parts, on which the control quality is guaranteed according to Sect. *12.8.2.* We will adapt the control system by switching the dynamic characteristics based on the changes in the casting speed.

12.10 Segmentation of Dynamics

The segmentation of dynamics is carried out with a base step of 0.05 m min^{-1} between the interval of casting speeds of 0.70 m min^{-1} – 1.55 m min^{-1}. According to this, we obtained:

- 17×4 distributed transient characteristics to the positive step changes of the cooling water flow rates in all zones (at casting speeds 0.75; 0.80 through 1.55 m min^{-1}),
- 17×4 distributed transient characteristics to the negative step changes of the cooling water flow rates in all zones (at casting speeds 0.75; 0.80 through 1.55 m min^{-1}).

The identification procedure of the dynamics was performed according to Section *12.7* using four secondary cooling zones. Thus, at each casting speed, we obtained gains K_p and time constants T_p in agreement with the orientation of step changes in cooling intensity. Table 12.12 lists the gains and time constants at 0.75 m min^{-1}.

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
Gains (Q_m+)	–78.1	–70.8	–54.1	–56.02
Gains (Q_m-)	82.7	74.8	56.1	58.76
Time constants (Q_m+)	9.9	12.3	28.1	36.8
Time constants (Q_m-)	10.0	12.5	28.3	37.43

Table 12.12 Identified gains K_p and time constants T_p at the casting speed 0.75 m min^{-1}.

A gradual increase in casting speed resulted in a decrease of gains and time constants. The values of the gains K_p and the time constants T_p in Table 12.12 are highlighted by vertical rectangles in Fig. 12.25.

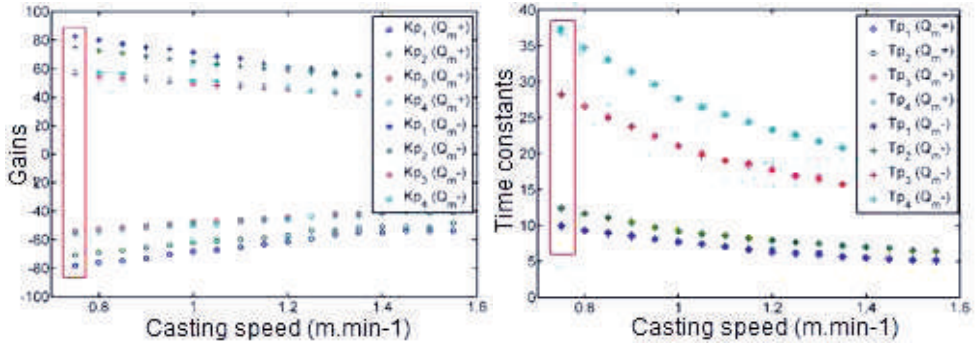


Fig. 12.27 Casting speed dependent gains and time constants.

We can conclude from the simulation results that the identified gains change their values depending on the direction of cooling intensities, however, their absolute value does not change significantly. The values of the time constants given by the direction of changes do not change at all.

The respective settings are changed in the control process, in other words, the temporal and spatial characteristics are varied based on the speed of casting and the orientation of the change (increase vs. decrease).

12.10.1 Adaptive Compensation of Changes in the Casting Speed

The control problem is defined by the compensation of a sequence of step changes in the casting speed, which will be represented by distributed disturbances $V(x,t)$ at individual samples. Their compensation is realized by using inputs $\bar{U}(k)$ in a way that the dynamic characteristics of the feedback loop are switched based on the actually used casting speed.

Let us consider a set of positive casting speed changes by $0.05\ \mathrm{m\ min^{-1}}$ ($0.70 \rightarrow 0.75$ then $0.75 \rightarrow 0.80$ and finally $0.80 \rightarrow 0.85\ \mathrm{m\ min^{-1}}$) at the time instants t=0, 1800, and 3600 seconds. The effect of this change is an overall increase of temperatures in the casting. The following figures show the control process quantities using the HLDS segmented dynamics model.

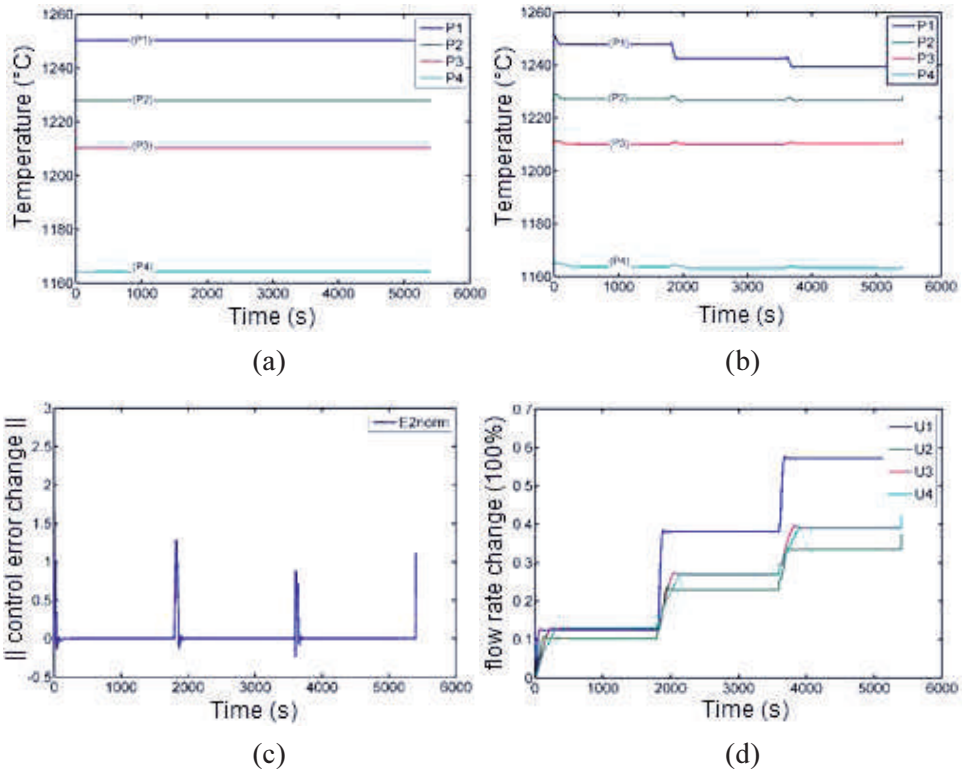


Fig. 12.28 Distributed disturbance rejection within the interval, (a) distributed reference, (b) distributed output, (c) change of the normed control error, (d) lumped inputs.

In contrast to the previous tasks, here the space synthesis is solved on the full resolution FEM mesh (i.e. 284 points). This causes a divergence of the distributed output displayed in the points P1 – P4. However, as it can be seen in the following Fig. 12.29, the quality of control in the space domain is in fact satisfactory.

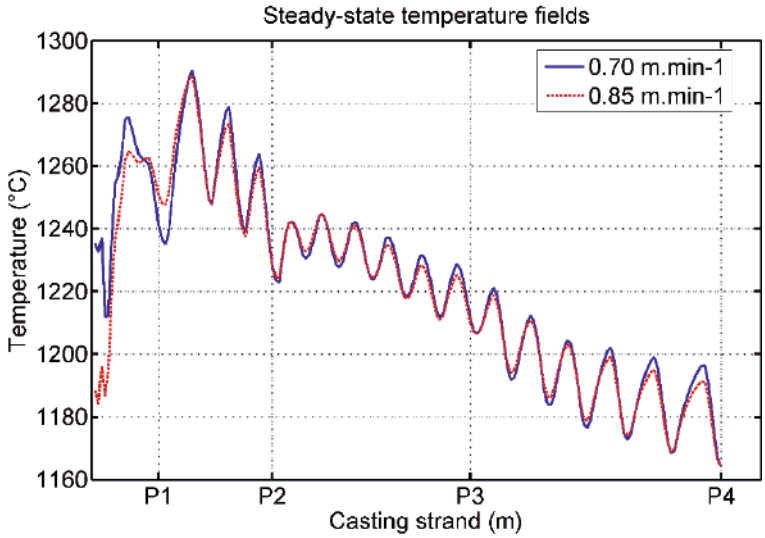


Fig. 12.29 Initial and final temperature fields after an overall casting speed change (0.70→0.85 m min^{-1}).

12.11 Modelling the Macroscopic Structure of the Casting

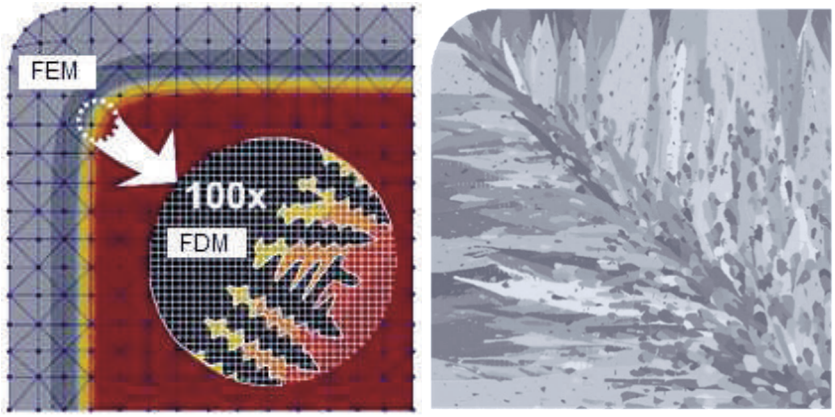
The temperature field transients itself cannot indicate the occurrence nor the location of surface or internal defects in the casting. If control should improve the quality of the casting, we need to include some specific rules into the design of control algorithms. The rules can be improved by numerical simulations of internal structures (e.g. ProCAST advanced modules) or physical simulations (e.g. Thermo-mechanical simulator Gleeble).

One may observe dendrite growth in the steel well below the standard resolution of the FEM mesh that is used to describe a temperature field in volume of the solidifying casting. Equally oriented dendrite groups form grains, which can be seen in metallographic etches by the naked eye. Their size and orientation determine the macroscopic structure of the casting. The structure is influenced by its thermal history—cooling mode. These simulations are important, when we search for links between cooling modes and the internal structures of the casting, which we want to include in the design of the control system.

The algorithm named *Cellular Automaton Finite Element* (CAFE) by Rappaz (2003) uses the thermal history computed by the finite element method. A great disproportion in sizes of individual dendrites and the volume of the casting does

not allow the use of a very fine mesh. On the level of dendrites, the algorithm uses the finite difference method with a very fine regular mesh of ~0.1 mm size.

The following figure displays the result of a CAFE simulation in ProCAST for the nominal operating mode (Table 12.8), with decreased casting speed. Only a quarter of the total cross-section is modelled utilizing the symmetry of the □200 casting.



(a) (b)

Fig. 12.30 (a) Dendrite growth at the solid/liquid interface, (b) Macroscopic structure on the quarter cut of the □200 casting (simulation result).

Dendrite growth kinematics is given by nucleation boundary conditions:

- Reference: Gaussian distribution of nucleation sites,
- Mean undercooling dT_m: 5.00,
- Standard deviation dTs: 1.0,

and by material properties:

- Grade 34CrMo4 (0.38 C, 0.23 Si, 0.64 Mn, 0.08 Ni, 0.99 Cr, 0.16 Mo),
- Liquidus temperature: 1499 °C (calculated from chemical composition),
- Growth kinetics coefficients: a_2=4.818e-7, a_3=1.968e-7,
- Thermal history: nominal operating mode, casting speed 0.75 m min^{-1}.

13. OPTIMIZATION OF THE SPATIAL DISTRIBUTION DESIGN OF THE CONTROLLED PLANT

The results presented in this book offer several further possibilities to utilize control loops for distributed parameter systems. Namely, the framework outlined here may come in handy at the design stage of technological and production processes, technical objects or equipment for these processes. The design procedure may be carried out analogously as the design of a real controlled distributed parameter system.

The design variable itself is the spatial distribution of the controlled object—geometry design; number, placement and parameters of actuators and other factors—so that the desired control quality is fulfilled both in the temporal and the spatial relations. Practically, this means that, one may carry out the optimization of the controller definition domain based on a model created in a software environment, selecting the minimal number of actuators and their parameters. All this is carried out while solving an approximation task for the control synthesis in the spatial relation, depending on the selected approximation tolerance δ, expressing the quality of control throughout the definition domain.

13.1 Groundwater remediation system design

Let us now show the basic steps of the design process for a groundwater remediation system, which can be considered as a distributed parameter system (see Fig. 13.1).

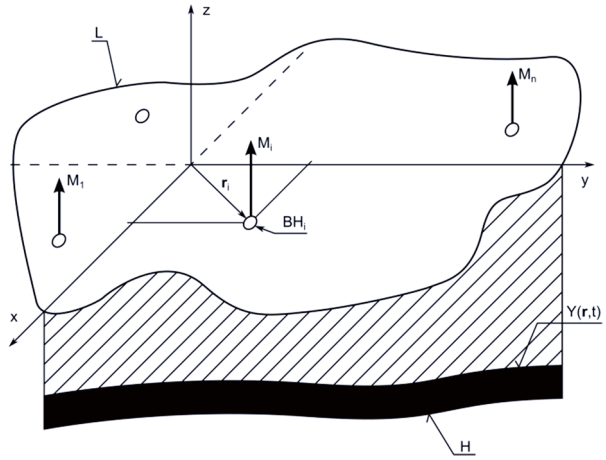


Fig. 13.1 Design of the spatial distribution of groundwater remediation system for a contaminated territory as a controlled system with distributed parameters.

H	- hydraulic water inlet.
L	- boundary of the locality with the polluted ground water.
$\{M_i(\boldsymbol{r}_i,t)\}_i$	- water outlet from the remediation locations.
$\{BH_i\}_i$	- remediation boreholes with specified parameters.
$Y(\boldsymbol{r},t)$	- contamination of the groundwater.

First, the pollution levels are assessed and the boundaries L of the polluted area are established based on a mathematic model of the groundwater pollution $Y(\boldsymbol{r},t)$ at the given steady-state operation mode. This model is created in a suitable software environment, for example the MODFLOW software suite created by the U.S. Geological Survey (USGS). Next, the software is used to create linearized distributed transient responses, their temporal $\{SH_i(\boldsymbol{r}_i,z)\}_i$ and spatial components $\{\mathcal{H}HR_i(\boldsymbol{r},\infty)\}_i$ in steady-state to step changes of groundwater pumping, with various configurations for the parameters of the remediation bore holes. This is done in a way that is possible to establish these transients for an arbitrary borehole location. Finally, the minimal number „*n*" of optimally distributed remediation boreholes is computed based on the solution of the approximation problem of the spatial control system synthesis, so that the remediation task is fulfilled in the steady-state

$$\begin{aligned}&\min\left\|Z(\boldsymbol{r},\infty)-\sum_{i=1}^{n}M_i(\boldsymbol{r}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{r},\infty)\right\|=\\&=\left\|Z(\boldsymbol{r},\infty)-\sum_{i=1}^{n}\breve{M}_i(\boldsymbol{r}_i,\infty)\,\mathcal{H}HR_i(\boldsymbol{r},\infty)\right\|\le\delta\end{aligned}\tag{13.1}$$

where $Z(\boldsymbol{r},\infty)$ is the acceptable pollution level, $\{\breve{M}_i(\boldsymbol{r}_i,\infty)\}_i$ is the steady-state amount of outgoing water pumped through the remediation borehole, while δ is the quadratic norm of the quality in the spatial relation. During the control synthesis in the temporal relation, it is possible to utilize well-established methods and software support known from classical control theory.

Finally, the design for the location of the definition domain L, and the location of the water outlets $\{M_i(\boldsymbol{r}_i,t)\}_i$ for the remediation borehole—representing the actuators for the remediation of polluted groundwater $Y(\boldsymbol{r},t)$—may serve as foundation for the technical implementation of an automated groundwater remediation system, designed as a distributed parameter control system.

14. CONCLUSION

Research in the field of control engineering for technological and production processes represented as distributed parameter systems has an interesting and long history. In the very first dedicated book titled *„Optimal Control Theory of Distributed Parameter Systems“* by Butkovskiy (1964)—one of the founders of distributed parameter system control theory—offers discussion and insight presented within lengthy chapters on the mathematical description, analog electronic modeling and control of the heating of metallic semi-products, based on the Pontryagin maximum principle. These were founded on the analytic solution of linear partial differential equations. The so-called mathematical theory of control for distributed parameter systems—known as the control of partial differential equations—is still being expanded up to this day. However, it soon became obvious that the analytical apparatus provides only modest possibilities to solve real-life engineering problems that require the control of distributed parameter systems.

Parallel to that, and partly thanks to the dramatic development of information technology, engineering practice adopted numerical methods for solving problems described by partial differential equations. Currently, all engineering issues described by partial differential equations are solved using numerical methods. Moreover, a whole array of software products were created lately, for analysis of dynamics of technological and production processes which are based on the numerical solution of partial differential equations. For example, there are more than ten sophisticated software packages available for the numerical analysis of dynamics of processes behind various metal casting technologies. These tools and numerical models offer novel possibilities to solve issues concerning the control of technological and production processes as distributed parameter systems. Using opportunities given by the numerical models and simulation tools, a new engineering approach for the control of technological and production processes based on linearized lumped input and distributed parameter output models - systems was proposed, designed and realized at the Institute of Automation, Measurement and Applied Informatics of the Faculty of Mechanical Engineering at the Slovak University of Technology in Bratislava, (Fig. 14.1).

The application phase of this engineering principle involves establishing validated numerical models, representing the plant within a distributed parameter control system. Control feedback loop optimization is realized in a way, that the effects of the input vector representing the actuators will govern the manipulated variable according to the

requirements of the technology. Meanwhile, control synthesis is decomposed into its spatial and temporal components. The temporal components are essentially SISO or MIMO control feedback loops with lumped parameters. Next, using these optimized SISO and MIMO feedback loops it is possible to compute the inputs, which will generate a distributed output of the manipulated variable that fits the reference as closely as possible. This approach does not necessitate an early or late discretization of the mathematical models, which is often the requirement in the practice of controlling distributed parameter systems. Instead, the procedure involves the decomposition of dynamics into its time and space components, based lumped input and distributed parameter output systems representing linearized models of the industrial process to be controlled.

The precision of introduced lumped input and distributed parameter output models is given by the precision of the numerical models that are created by various software products for numerical analysis of dynamics of technological and production processes to be controlled. The co-simulation option of the Matlab & Simulink software with different software products offers a range of novel possibilities. These may create development and expert systems to support the work of operating personnel servicing production equipment in everyday manufacturing.

The predominating view of the scientific and engineering community is that distributed parameter systems are too complicated and may offer little to the engineering practice. This may be so, since the problems begin right at the stage of modelling the dynamics of these processes. These were traditionally represented by nonlinear partial differential equations on definition domain consisting of complex three-dimensional geometric shapes. Currently, numerical methods offer a broad spectrum of possibilities to prepare lumped input and distributed parameter output models and to investigate the underlying dynamics of physical systems. These tools can establish the properties of the steady-state operating modes that are useful for the linearization of the dynamics, by which it is possible to create segmented dynamics to represent the nonlinear transition between different steady-state operating modes. It is then possible to create distributed parameter control systems by using the aforementioned numerical models. These can be used to establish the basic outlines for the engineering methodology presented in this book, enabling the control of distributed parameters systems that is no more complex than any other classical lumped input-output control system.

There is one more remaining question: what can be gained by the engineering practice by controlling plants as distributed parameter systems? In general, these systems offer an optimally controlled transfer of continuum in technological and production systems. The manufacturing practice may introduce novel innovative solutions, with the aim of increasing production quality and efficiency, optimize raw-

material use, minimize energy, maximize competitiveness and of course to minimize environmental impact.

How to use these new possibilities in engineering practice? Very simply, as it was demonstrated in several examples of this book such as metal casting, induction heating, extrusion of plastics, continuous casting of steel and groundwater remediation. A custom development system according to the scheme in Fig. 7.9 may be created for each technology or production process. Based on software for numerical simulation and its connection with Matlab & Simulink a cosimulation scheme may be created. The control system is then designed and optimized based on the results of this book. The design procedure for the implementation of control systems presented in Chap. *6* and *7* along with blocks of DPS Blockset for Matlab & Simulink make this easy enough. Finally, the optimized control system is ready to be implemented on the plant. The optimized development system also offers to support the work of operating personnel as a modern software-based expert system.

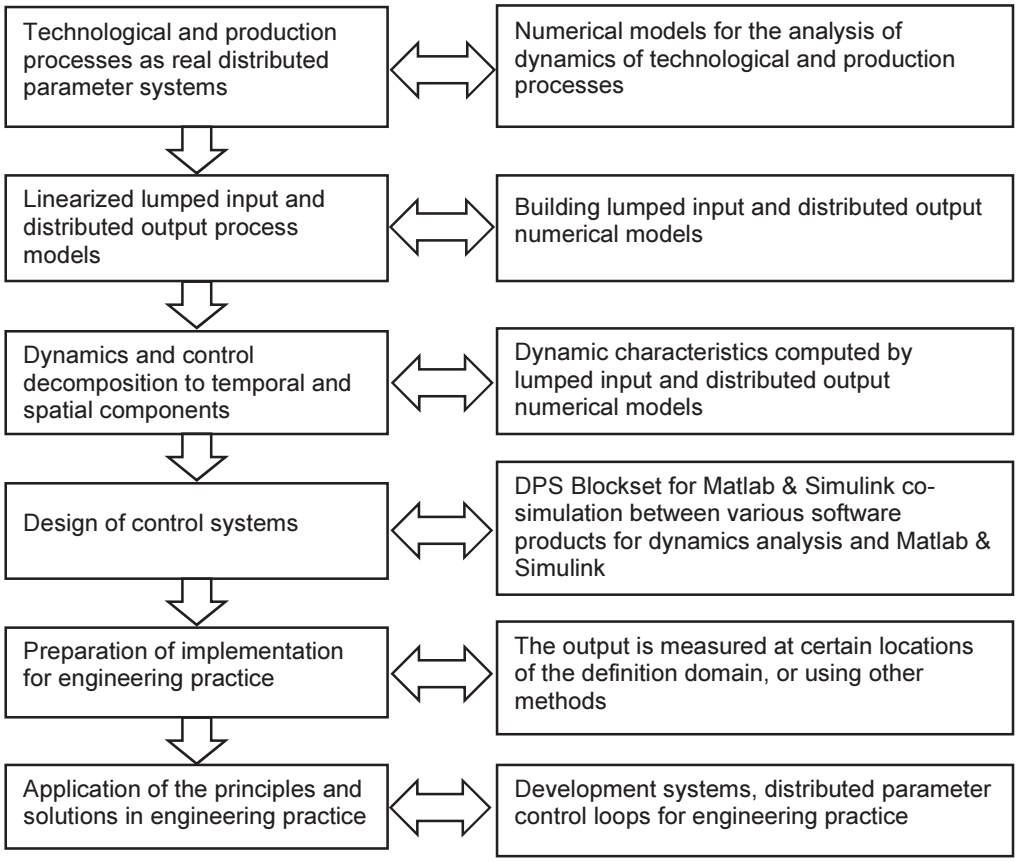


Fig. 14.1 Elements and interaction layers of the engineering approach for the control of technological and production processes as distributed parameter systems, supported by advanced numerical modeling.

BIBLIOGRAPHY

AGGELOGIANNAKI, E., SARIMVEIS, H. 2008. Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models. *Computers and Chemical Engineering* , vol. 32, pp. 1225–1237.

ALONSO RASGADO, M., T., DAVEY, K., CLARK, L., D., HINDUJA, S. 2006. Boundary element stress analysis for copper-based dies in pressure die casting. *Computers and Structures*, vol. 84, pp. 254-267.

ANSYS POLYFLOW. 2010. User's Guide, [online]. Available at: http://www.ansys.com/.

ANSYS Mechanical APDL and Mechanical Applications Theory Reference. 2010.[online]. Available at: http://www.ansys.com/.

AUBRY, R., IDELSOHN, S., R., ONATE, E. 2005. Particle finite element method in fluid-mechanics including thermal convection-diffusion. *Computers and Structures*, vol. 85, pp. 1459-1475.

BATHE, K., J., WILSON, E., L. 1976. Numerical Methods in Finite Element Analysis. New Jersey: Prentice-Hall, pp. 528. ISBN 0-13-627190-1.

BECKER, M. 2009. Quality Control of a Light Metal Die Casting Process Using Artificial Neural Networks. In *7th IEEE Conference on Computational Cybernetics*: Proceedings. Palma de Mallorca, pp. 113–117. ISBN 978-1-4244-5310-8.

BELAVÝ, C. et al. 2012. Robust Control of Distributed Parameter Systems with Demonstration in Casting Technology and Matlab/Simulink/DPS Blockset Software Support. In: Matlab - A Fundamental Tool for Scientific Computing and Engineering Applications. *InTech Open Access publisher of Scientific Books and Journals*, pp. 29-56. Rijeka. ISBN 978-953-51-0752-1.

BENTSMAN, J. et al. 2011. Hybrid Control of Continuous Casting for Whale and Crack Prevention and Resonance Control in Mold Oscillation System. NSF GRANT # DMI-0900138. NSF PROGRAM NAME: Control Systems Program. *Proceedings of NSF Engineering Research and Innovation Conference, Atlanta, Georgia.*

BRIMACOMBE, K., J. 1999. The Challenge of Quality in Continuous Casting Processes. In: *Edward DeMille Campbell Memorial Lecture (1996). Metallurgical and Material Transactions B*, vol. 30B, pp. 553-566.

BUČEK, P. 2010. Mechatronická zlievarenská forma ako systém s rozloženými parametrami. Doktorandská dizertačná práca, Bratislava.

BUTKOVSKIY, A., G., LERNER, Y., A. 1961. Discussion within the Optimal Systems of Control Session. *Proceedings of the First International Congress of the IFAC*, Moscow, 1960. Butterworths, London.

BUTKOVSKIY, A., G. 1965. Teorija optimaľnovo upravlenija sistemami s raspredeľonnymi parametrami. Nauka, Moskva.

BUTKOVSKIY, A., G. 1979. Charakteristiki sistem s raspredeľonnymi parametrami. Nauka, Moskva.

BUTKOVSKIY, A., G. 1982. Green's Functions and Transfer Functions Handbook. Ellis Horwood, Chichester.

CAMACHO, E., F., BORDONS, C. 2004. Model Predictive Control. London, Springer, ISBN 1-85233-694-3.

CAMBER, J. 2013. Riadenie procesov indukčného ohrevu ako systémov s rozloženými parametrami, Doktorandská dizertačná práca, Bratislava.

CHRISTOFIDES, P. D. 2001. Control of nonlinear distributed process systems: recent developments and challenges, *AIChE Journal* , vol. 47 (3), pp. 514–518.

CHRISTOFIDES, P. D.: WANG, X. Z. (Eds.). 2008. Special issue on 'control of particulate processes'. Chemical Engineering Science. Vol. 63 (5).

CIARLET, P., G. 2002. The finite element method for elliptic problems. (Classics in applied mathematics 40). Philadelphia : *SIAM*, ISBN 0-89871-514-8.

COMPUPLAST. 2011. Tutorial for Extruder MODULE, Version 6.4, COMPUPLAST INTERNATIONAL, Inc. Zlín.

COURANT, R. 1943. Variational Methods for Solution of Equilibrium and Vibration. *Bull. Amer. Math. Soc.* Volume 49, Number 1, pp. 1-23.

CURTAIN, R., F., ZWART, H., J. 1995. An Introduction to Infinite-Dimensional Linear Systems Theory. Springer, New York.

CURTAIN, R., F., MORRIS, K. 2009. Transfer Functions of Distributed Parameter Systems: A tutorial. *Automatica.* Vol. 45, pp. 1101-1116.

DAVIES, E., J. 1990. Conduction and Induction Heating, Stevenage: Peter Peregrinus Ltd.

DOCHAIN, D., BABARY, J. P., TALI-MAANAR, N. 1992. Modeling and adaptive control of nonlinear distributed parameter bioreactors via orthogonal collocation. *Automatic.* Vol. 28 (5), pp. 873–883.

DOCHAIN, D., TALI-MAANAR, N., BABARY, J. P. 1997. On modelling monitoring and control of fixed bed bioreactors. *Computers and Chemical Engineering*. Vol. 21 (11), pp. 1255–1266.

ESKIN, D., G., KATGERMAN, L. 2009. Solidification phenomena related to direct chill casting of aluminium alloys: fundamental studies and future challenges. *Materials Technology*. Vol. 24, 3, pp. 153-156.

ESKIN, G., I. 1998. Ultrasonic Treatment of Light Alloy Melts. Amsterdam, Gordon and Breach, ISBN 90-5699-042-X.

ProCAST. 2007. Užívateľská príručka k softvéru, Appendix C: Mathematical Formulations.

FERREAU, H., J., POTSCHKA, A., KIRCHES, C. 2007 – 2014. qpOASES webpage, http://www.qpOASES.org/.

FREY, P., J., GEORGE, P., L. 2008. Mesh generation: application to finite elements. Hoboken, Wiley, ISBN 978-1-84821-029-5.

GANAPATHYSUBRAMANIAN, B., ZABRAS, N. 2005. On the control of solidification using magnetic fields and magnetic field gradients. *International Journal of Heat and Mass Transfer.* Vol. 48, pp. 4174-4189.

GARGE, S., C., WETZEL, M., D., OGUNNAIKE, B., A. 2012. Control-Relevant Model Identification of Reactive Extrusion Processes. *Journal of Process Control*, vol. 22, pp. 1457–1467.

GREMAUD, M., RAPPAZ, M. 2001. Modeling of foundry processes: Diferences between various solutions. American Foundry Society.

HAGEN, H. 1992. Editor: Curve and Surface Design. Geometric Design Series. SIAM, Philadelphia.

HAN-XIONG Li, CHENKUN Qi, YONGGUANG, Yu. 2009. A spatio-temporal Volterra modeling approach for a class of distributed industrial processes. *Journal of Process Control*. Vol. 19, pp. 1126-1142.

HAN-XIONG Li, CHENKUN Qi. 2010. Modeling of distributed parameter systems for applications - A synthesized review from time–space separation. *Journal of Process Control*. Vol. 20, pp. 891-901.

HAROLD, F., G., WAGNER, J., R., MOUNT, E., M. 2005. Extrusion: The Definitive Processing Guide and Handbook. Norwich, Wiliam Andrew Inc.

HO-LE, K. 1988. Finite element mesh generation methods: a review and classification. *Computer-Aided Design*. Vol. 20, No. 1, pp. 27-38.

HRENNIKOFF, A. 1941. Solution of Problems in Elasticity by the Framework Method. *ASME J. Appl. Mech*. Vol. 8, pp. A619–A715.

HUBA, M. 2007. Nelineárne systémy: návrh riadenia s rešpektovaním obmedzení. 1. vyd. Vydavateľstvo STU v Bratislave, s. 218. ISBN 978-80-89316-05-2.

HULKÓ, G. a i. 1998. Modelovanie, riadenie a návrh systémov s rozloženými parametrami. Vydavateľstvo STU v Bratislave, s. 267. ISBN 80-227-1052-0.

HULKÓ, G. et al. 1998. *Modeling, Control and Design of Distributed Parameter Systems with Demonstrations in Matlab*. Publishing House of STU, Bratislava, pp. 265. ISBN 80-227-1083-0.

HULKÓ, G. et al. 2012. Distributed Parameter Systems Blockset for Matlab & Simulink. DPS Blockset for Matlab & Simulink. Program CONNECTIONS of The MathWorks - partner produkt spoločnosti The MathWork. 2003 - 2013, Bratislava – Natick.

HULKÓ, G. et al. 2009. Engineering Methods and Software Support for Modeling and Design of Discrete-time Control of Distributed Parameter Systems. – *Mini-tutorial*

- In: European Control Conference 2009. ECC '09. Budapest, Hungary. EUCA, pp. 272-281. ISBN 978-963-311-369-1.

HULKÓ, G. et al. 2009. Engineering Methods and Software Support for Modelling and Design of Discrete-time Control of Distributed Parameter Systems. In: *European Journal of Control.* Vol. 15. Iss. 3-4, Fundamental Issues in Control, pp. 407-417. ISSN 0947-3580.

HULKÓ, G. et al. 2011. Control of Distributed Parameter Systems - Engineering Methods and Software Support in the Matlab & Simulink Programming Environment. *Chapter in Matlab / Book 4 - Matlab for Engineers - Applications in Control, Electrical Engineering, IT and Robotics*, InTech Open Access publisher of Scientific Books and Journals, pp. 27-50, Vienna – Luknow, India. ISBN 978-953-307-914-1.

HULKÓ, G. et al. 2009. Engineering Methods and Software Support for Control of Distributed Parameter Systems. *ASC 2009: Proceedings of the 7th Asian Control Conference,* Hong Kong, China.

HULKÓ, G. a i. 2011. Adaptívne riadenie procesov kontiliatia v Železiarňach Podbrezová a. s., ako systémov s rozloženými parametrami – ARPKONTI. Podbrezová – Bratislava.

HULKÓ, G., ROHÁĽ-ILKIV, B., NOGA, P., LIPÁR, S. 2012. Control of Energy Systems as Distributed Parameter Systems with Software Support by Virtual Software Environments. *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, HI, pp. 2382-2387.

HULKÓ, G. et al. 2013. Modelling and control of extruder barrel temperature field. In Control of Systems Governed by Partial Differential Equations, Volume 1, 1st IFAC Workshop on Control of Systems Governed by Partial Differential Equations. Paris, France, IFAC, pp. 191-196. ISBN 978-3-902823-54-0.

HULKÓ, G. et al. 2010 – 2013. Interactive Control, www.dpscontrol.sk.

HUNTER, P., PULLAN, A. 2002. FEM/BEM Notes. Auckland. Dept. of Engineering Science, The University of Auckland.

KAPUSTA, J. 2013. Numerical modelling and control of progressive steel billet induction heater as distributed parameter system. Doktorandská dizertačná práca, Bratislava.

KAPUSTA, J., J. CAMBER, J., HULKÓ, G. 2013. Steel Billet Continuous Induction Heating – Numerical Model and Advanced Control, *Applied Mechanics and Materials*, Vol 378, pp. 520-528, Trans Tech Publications, Switzerland. doi:10.4028/www.scientific.net/AMM.378.520.

KLOBČAR, D., TUŠEK, J. 2008. Thermal Stresses in Aluminium Alloy Die Casting Dies. *Computational Materials Science*. Vol. 50, No. 2, pp. 1147–1154. ISSN 0927-0256.

KOMORNIK, V., LORETI, P. 2005. Fourier series in control theory. Springer – New-York. ISBN 0-387-2238.

KONG, L., X. et al. 2001. Thermal Control with Image Processing and Fuzzy Controllers. In Joint 9th IFSA World Congress and 20th NAFIPS International Conference: Proceedings. Vancouver. Vol. 3, pp. 1443–1449. ISBN 0-7803-7078-3.

KORNEJČUK, N., P. 1976. Ekstremaľnye zadači teorii približenija. Nauka, Moskva.

KORNEJČUK, N., P. 1987. Točnye konstanty v teorii približenija. Nauka, Moskva.

KOSEC, B., KOSEC, G., SOKOVIC, M. 2007. Temperature field and failure analysis of die-casting die. *Archives of Materials Science and Engineering*. ISSN 1897-2764, vol. 28, iss. 3, pp. 182–187.

KOTRBÁČEK, P., LUKS, T., HORSKÝ, J. 2010. Heat transfer tests (Meranie chladiacich účinkov trysiek KONTI ETAPA I.) pre ŽP VVC. Heat Transfer and Fluid Flow Laboratory, Brno.

KRSTIC, M., SHMYSLAYEV, A. 2008. Boundary control of PDEs - A Course on Backstepping Design, SIAM.

KUGI, A., SAWODNY, O. 2010. Steuerungs- und Regelungsansätze für Systeme mit verteilten Parametern. *at – Automatisierungstechnik.* Vol. 58, Issue 3 (Mar 2010). Published Online.

LANGER, E. 1979. Teorie indukčního a dielektrického tepla, Praha: Academica.

LASIECKA, I., TRIGGIANI, R. 2000. Control Theory for Partial Differential Equations. Vol. I.-III. Cambridge University Press.

LI, X., FAUTRELLE, Y., REN, Y. 2007. Effect of a high magnetic field on the microstructure in directionally solidified Al–12 Ni alloy. *Journal of Crystal Growth*. Iss. 306, pp. 187-194.

LIN KUAN-JU, THOMAS, G., B. 2002. Thermal stress analyses of bulging with roll missaligment for various slab cooling intensities. *China steel technical report.* No. 16, pp. 9 -14.

LIPÁR, S. 2014. Modelovanie a riadenie procesov extrúzie plastov ako systémov s rozloženými parametrami, Doktorandská dizertačná práca, Bratislava.

LIU, Y., C., YU, W., C., ROUX, B., LYUBIMOVA, T., P., LAN, C., W. 2006. Thermal–solutal flows and segregation and their control by angular vibration in vertical Bridgman crystal growth. *Chemical Engineering Science*. Vol. 61, No. 23, pp. 7766-7773.

LIONS, J., L. 1971. Optimal Control of Systems Governed by Partial Differential Equations. Springer-Verlag.

LOTOV, A., V., KAMENEV, G., K., BEREZKIN, V., E., MIETTINEN, K. 2005. Optimal control of cooling process in continuous casting of steel using a visualization-based multi-criteria approach. *Applied Mathematical Modeling*. Vol. 29, No. 7, pp. 653-672.

LOUHENKILPI, S., et al. 2008. Modelling as tool for improvement production efficiency and production quality. *Proceedings of 3rd International Conference on Process Development in Iron and Steelmaking*. Lulea, Sweden.

LOUHENKILPI, S., LAINE, J., MIETTINEN, J., VESANEN, R. 2013. New Continuous Casting and Slab Tracking Simulators for Steel Industry. *Materials Science Forum.* Vol. 762, pp 691-698. Trans Tech Publications, Switzerland. www.scientific.net/MSF.762.691.

MADELUNG, E. 1975. Príručka matematiky pre fyzikov. Bratislava. ALFA.

MAIJER, D., M., OWEN, W., S., VETTER, R., A. 2009. An investigation of predictive control for aluminum wheel casting via a virtual process model. *Journal of Materials Processing Technology.* Iss. 209, pp. 1965-1979. ISSN 0924-013.

MEURER, T., KUGI, A. 2009. Trajectory planning for boundary controlled parabolic PDEs with varying parameters on higher-dimensional spatial domains**.** *IEEE Transactions on Automatic Control*, Vol. 54, Issue 8, pp. 1854-1868.

MEURER, T., KUGI, A. 2011. Tracking control design for a wave equation with dynamic boundary conditions modeling a piezoelectric stack actuator. *International Journal of Robust and Nonlinear Control.* Vol. 21. Issue 5, pp. 542-562.

MIETTINEN, J., et al. 2008. IDS tool – theory and applications for continuous casting and heat treatment including modelling of microstructure and inclusions. *Proceedings of 6th ECCC* – Riccione, Italy.

METAN, V., EIGENFELD, K., RABIGER, R., LEONHARDT, M., ECKER, S. 2009. Grain size control in Al–Si alloys by grain refinement and electromagnetic stirring. *Journal of Alloys and Compounds.* Vol. 487, pp. 163-172.

MENDEL, M., KOVÁCS, T., HULKÓ,G. 2015. Control of Groundwater Remediation Process as Distributed Parameter System. *Proceedings of IEEE Process Control Conference.* Vysoké Tatry.

MOIN, P. 2010. Fundamentals of Engineering Numerical Analysis. New York. Cambridge University Press. ISBN 978-0-521-88432-7.

MORARI, M., ZAFIROU, E. 1989. Robust process control. Prentice Hall, New Jersey.

MÖLLMANN, P., K., PINNO, F., VOLLMER, M. 2011. Two-Color or Ratio Thermal Imaging – Potentials and Limits. FLIR Technical Series, Application Note for Research & Science.

NEDJAR, B. 2002. An enthalpy-based finite element method for nonlinear heat problems involving phase change. *Computers and Structures*. Vol. 80, pp. 9–21.

NEJAD, A., A., MAGHREBI, M., J., TABRIZI, H., B., HENG, Y., MHAMDI, A., MARQUORDT, W. 2010. Optimal operation of alloy material in solidification processes with inverse heat transfer. *International Communications in Heat and Mass Transfer*. Vol 37, pp. 711-716.

NOGA, P. 2011. Modelovanie a riadenie procesov zlievania ako systémov s rozloženými parametrami. Doktorandská dizertačná práca, Bratislava.

ONDREJKOVIČ, K. 2012. Modelovanie a riadenie procesov plynulého odlievania ocele ako systémov s rozloženými parametrami. Doktorandská dizertačná práca, Bratislava.

ONDREJKOVIČ, K., BUČEK, P, PYSZKO, R., HULKÓ, G. Control of Continuous Casting Processes as Distributed Parameter Systems. Proceedings of the VII. European Continuous Casting Conference. Düsseldorf.

ONDREJKOVIČ, K., BUČEK, P., NOGA, P., TKÁČ, L, HULKÓ, G. 2013. Modeling and Control of Temperature Field of the Secondary Cooling Zone in Continuous Casting of Steel as Distributed Parameter System. *Preprints of the 2013 IFAC Conference on Manufacturing Modelling, Management, and Control,* Saint Petersburg.

PETRUS, B., ZHENG, K., ZHOU, X., THOMAS, G., B., BENTSMAN, J. 2010. Real-Time, Model-Based Spray-Cooling Control System for Stell Continuous Casting. *Metallurgical and Materials Transactions B*. DOI: 10.1007/s11663-010-9452-7.

PETRUS, B., BENTSMAN, J., THOMAS, B., G. 2012. Enthalpy-based Feedback Control Algorithms for the Stefan Problem. *Proc. 51st IEEE Conference on Decision and Control*, Maui, Hawaii, USA, pp. 7037-7042.

PETRUS, B., BENTSMAN, J., THOMAS, B., G. 2014. Applications of Enthalpy-Based Feedback Control Methodology for the Two-Sided Stefan Problem. *Proc. 2014 American Control Conference*, Portland, OR, USA, pp. 1015-1020.

PREVDI, F., SAVARESI, S., M., PANAROTTO, A. 2006. Design of a Feedback Control System for Real-Time Control of Flow in a Single-Screw Extruder. *Control Engineering Practice*, vol. 14, pp. 1111–1121.

RAPPAZ, M., BELLET, M., DEVILLE, M. 2003. Numerical Modeling in Materials Science and Engineering. Heidelberg : Springer. Pp. 540. ISBN 978-3-540-42676-9.

RAUVENDAAL, C. 2001. Polymer Extrusion, 4th ed. München, Hanser Verlag.

REGER, M., VERO, B., CSEPELI, Zs., SZABO, Z., JOZSA, R., KELEMEN, T. 2013. Effect of supporting roll setting on the inner structure of cc slabs**.** *Materials Science Forum* Vol. 729, pp. 175-180. Online available since 2012/Nov/12 at www.scientific.net. Trans Tech Publications, Switzerland. doi:10.4028/www.scientific.net/MSF.729.175.

REGER, M., VERO, B., CSEPELI, Zs., JOZSA, R. 2014. Effect of mushy zone permeability on the formation of centerline segregation in slab casting. *Materials Science Forum Vols.* 790-791, pp 296-301.Online available since 2014/May/09 at www.scientific.net. Trans Tech Publications, Switzerland. doi:10.4028/www.scientific.net/MSF.790-791.296.

RÉGER, M., VERŐ, B., JÓZSA, R. 2014. Control of Centerline Segregation in Slab Casting. *Acta Polytechnica Hungarica.* Vol. 11, No. 4, pp. 119-137. ISSN 1785-8860.

REKTORYS, K. 1999. Varianční metody v inženýrských problémech a v problémech matematické fyziky. Praha, Academia, ISBN 80-200-0714-8.

ROBUST Control Toolbox. 2013. User's Guide, www.mathworks.com.

RUDNEV, V. 2003. Handbook of induction heating. Michigan USA. Marcel Dekker AG.

SADEGIPHOUR, K., DOPKIN, J., A., LI, K. 1996. Computer aided finite element/experimental analysis of induction heating process of steel. *Computers in industry.*

SHADRIN, A. 2005. Approximation Theory – Lecture notes for the course. DAMTP, University of Cambridge.

SHEWCHUK, J., R. 1994. Conjugate Gradients Without the Agonizing Pain. Pittsburgh: Carnegie Mellon University, [Online]. Available: http://www.cs.cmu.edu/jrs/jrspapers.html.

SHOUMEIA, X., LAUB., F., LEE, W., B. 2002. An efficient thermal analysis system for the die-casting process. *Journal of Materials Processing Technology*. Iss. 128, pp. 19-24.

SINGER, I. 1974. The Theory of Best Approximation and Functional Analysis. SIAM, Philadelphia.

SIDDIQI, A., H. 1993. Functional Analysis With Applications. New Delhi. Tata McGraw-Hill. ISBN 0074519204.

SMITH, G., D. 2004. Numerical solution of partial differential equations: finite difference methods. Oxford: Oxford University Press, ISBN 0 19 859650 2.

STEFANESCU, D. 2008. Science and Engineering of Casting Solidification, 2nd ed., New York: Springer.

THOMAS, B., G., et al. 2006. Online Dynamic Control of Cooling in Continuous Casting of Thin Steel Slabs. Grant: #DMI- 0500453. *Proceedings of NSF Design, Service, and Manufacturing Grantees and Research Conference*. St. Louis, Missouri.

Užívateľská príručka k softvéru Matlab, Optimization Toolbox User's Guide. 2010. [online] The Mathworks, Dostupné na Internete: http://www.mathworks.com/help/toolbox/optim/optim_product_page.html.

Užívateľská príručka k softvéru ProCAST. 2007. Appendix C: Mathematical Formulations. ESI-Group.

VESELÝ, V., HARSÁNYI, L. 2007. Robustné riadenie dynamických systémov. Vydavateľstvo STU, Vazovova 5, Bratislava.

VILANOVA, R. 2008. IMC based Robust PID design: Tuning guidelines and automatic tuning. *Journal of Process Control,* No 18, pp. 61-70.

VÍTEČKOVÁ, M. 2006. Jednoduché seřízení regulátorů metodou SIMC. *AT&P journal PLUS2*, pp. 50-54.

WANG, P., K., C. 1964. Control of Distributed Parameter Systems. Academic Press.

WANG, Q., G., et al. 2001. Single-loop controller design via IMC principles. *Automatica,* 37, pp. 2041-2048.

WROBEL, L., C. 2002. The boundary element method: Applications in thermo-fluids and acoustics. Chichester. Wiley, ISBN 0-471-72039-9.

XING, H., ZHAO, W., WANG, H. 2008. Feedback Control for Parabolic Distributed Parameter Systems with Time-Delay. In *Proceedings of the 7th World Congress on Intelligent Control and Automation.* Chongqing, pp. 4604–4609. ISBN 978-1-4244-2113-8,

YANG, T. et al. 2006. Thermal analysis of casting dies with local temperature controller. *The International Journal of Advanced Manufacturing Technology*. Vol. 33, No. 3–4, pp. 277–284. ISSN 1433-3015.

YEN, C., LIN, J., C., LIi, W. 2005. The selection of optimal cooling parameters for the multi-cavity die. *International Journal of Advanced Manufacturing Technology*. Vol. 26, pp. 228–235.

ZAIDAT, K., MANGELINK-NOEL, N., MOREAU, R. 2007. Control of melt convection by a travelling magnetic field during the directional solidification of Al–Ni alloys. *Comptes rendus Mecanique.* Iss. 335, pp. 330-335.

ZAVJALOV, JU., S., KVASOV, P., I., MIROŠPIČENKO, V., L. 1980. Metody splajn-funkcij. Nauka, Moskva.

ZHANG, H., T., QI, C., K, ZHOU, T., LI, H., X. 2007. Greatly Enhancing the Modeling Accuracy for Distributed Parameter Systems by Nonlinear Time/Space Separation. *Physica A.* Vol. 376, pp. 215-222.

ZHANG, B., MAIJER, D., M., COCKCROF, S., L. 2007. Development of a 3-D thermal model of the low-pressure die-cast (LPDC) process of A356 aluminum alloy wheels. *Materials Science and Engineering A.* Iss. 464, pp. 295-305.

ZHENG, K., PETRUS, B., THOMAS, B., G., BENTSMAN, J. 2007. Design and Implementation of a Real-time Spray Cooling. CoAISTech 2007, *Steelmaking Conference Proc.*, (May 7-10, Indianapolis, IN), AIST, Warrendale, PA, *Vol. 1, Control System for Continuous Casting of Thin Steel Slabs*.

ZIENKIEWICZ, O., C., TAYLOR, R., L., ZHU, J., Z. 2005. Finite Element Method: its basic and fundamentals. Oxford, Elsevier Butterworth-Heinemann, ISBN0750663200.

ZLÁMAL, M. 1968. On the finite-element method. *Numerische Mathematik*, Vol.12, pp. 394-409.

ZWART, H., J. 2013. Geometric Theory for Infinite Dimensional Systems (Lecture Notes in Control and Information Sciences). Springer , New Jersey.

CONTENTS

1. **INTRODUCTION** **7**

2. **DPS – DDS – LDS** **12**

3. **ANALYSIS AND DECOMPOSITION OF DYNAMICS** **15**

3.1 Decomposition of Dynamics Based on Transient Characteristics **15**

3.2 Decomposition of Dynamics Based on Impulse Characteristics **20**

3.3 Decomposition of Dynamics Based on Partial Outputs **23**

4. **SYSTEM IDENTIFICATION** **25**

4.1 Analytical System Identification **25**
4.1.1 Some Examples of Analytical Models in Standardized Forms 30
4.1.2 Models for the Control of Partial Differential Equations 34

4.2 Numerical Methods for Identification **38**
4.2.1 Finite Difference Method 38
Explicit FDM 39
Neumann Stability Condition 42
Implicit FDM 43
4.2.2 Finite Element Method 47
Basis Functions 48
Triangular and Tetrahedral Elements 51
General Problem Formulation 55
Error Estimation 56
Problem Solution, Galerkin Formulation 57

4.3 Experimental System Identification **62**

5. **CONTROL SYNTHESIS** **67**

5.1 Open Loop Control **67**

5.2 Closed Loop Control with the RHLDS Block **75**

5.3 Closed Loop Control **80**

5.4 Additional Control Loop Configurations **83**
5.4.1 Control Loop with Internal Model 84
5.4.2 Control of Partial Differential Equations 85

5.5 Robust and Adaptive Control 89

***6.* SOFTWARE SUPPORT 90**

6.1 DPS Blockset 91
6.1.1 Block: HLDS 92
6.1.2 Block: RHLDS 94
6.1.3 Block: DPS Space Synthesis 95
6.1.4 Block: DPS Time Synthesis 96
6.1.5 Block: MtoP 99
6.1.6 Block: DPS Scope 100
6.1.7 Block: Norm 103

6.2 Co-simulation 103
6.2.1 ProCAST Functions 106
6.2.2 Co-simulation Setup 107

6.3 Interactive Control 110

***7.* PREPARATION FOR THE APPLICATIONS 111**

7.1 Control by Linearized Part of Controlled System Dynamics 111

7.2 Control Synthesis by a Vector of Controlled Outputs of HLDS 117
7.2.1 Controller Design by SISO Loops 118
7.2.2 Controller Design by MIMO Loops 124

7.3 Control Synthesis by Vector of Controlled Outputs of the NHLDS 126

7.4 Development Systems for the Design of Control Loops 128

***8.* MODELING AND CONTROL OF THE GRAVITATIONAL CASTING PROCESS .. 132**

8.1 Casting Technology 132
8.1.1 Mathematical Model 135
Heat Transfer 135
Melt Flow 138
8.1.2 Casting Control Problem 139
Mold Preheating Control 139
Casting Solidification Control 139

8.2 Benchmark Experimental Casting Plant 141
8.2.1 Plant Technology 143
Preheating System 143
Cooling System 145
8.2.2 Measurement and Control Loops 146
Mold Temperature Measurement 146
Cooling Medium Temperature Measurement 147
Casting Temperature Measurement 147
Heating Elements 147
Cooling Pumps 147
Software Interface 147

8.2.3 Construction 149

8.3 Control of the Casting Process 153
8.3.1 Control System 153
8.3.2 Numerical Model and Experimental Validation 154
Mold Geometry, FEM Mesh Generation 154
Materials 154
Interfaces 156
Boundary Conditions 156
Initial Conditions 157
8.3.3 Mold Preheating Control 158
Identification of Mold Dynamics 159
Mold Preheating Controller Design 165
Mold Preheating Optimization 170
8.3.4 Casting Solidification Control 174
Controller Operation 176

***9.* ROBUST CONTROL OF THE CASTING MOLD TEMPERATURE FIELD 178**

9.1 Sources of Uncertainties in the Control System 178
9.1.1 Control Loop with Robust IMC Controllers 180

9.2 Identification of the Dynamical Characteristics of the Mold 183

9.3 Uncertainties in the Time and Space Domain 186

9.4 Robust Control of the Mold Temperature Field 190

9.5 Control of the Temperature Fields on the Physical Model 197

***10.* MODELING AND CONTROL OF THE INDUCTION HEATING PROCESS 201**

10.1 The Induction Heating Process 202
10.1.1 The Skin Effect 203
10.1.2 Electromagnetic Energy of Heating 205

10.2 Dynamics of the Continuous Heating of Steel Semi-products 207

10.3 Model in the COMSOL Multiphysics Software Environment 209

10.4 Semi-product in the Segmented Induction Heater as DPS 211

10.5 Control of the Temperature Field During Induction Heating 213
10.5.1 Design of the Control System Using the DPS Blockset 216
10.5.2 Change of the Distributed Reference 217
10.5.3 Distributed Disturbance Rejection 218

***11.* MODELING AND CONTROL OF THE PLASTICS EXTRUSION PROCESS 220**

11.1 Extruder Operation 221

11.2 Analysis of the Extrusion Process 221

11.3 Formulation of the Control Problem for the Extrusion Process 222

11.4 Modelling the Dynamics of the Extrusion Process 223

11.5 Model in the COMPUPLAST Software Environment..... 224

11.6 Model in the ANSYS Polyflow Software Environment..... 224

11.7 Experimental Device 228

11.8 Extruder Operation as DPS 228

11.9 Control System Design Using DPS Blockset..... 234
11.9.1 Synthesis of control by PI controllers..... 235
11.9.2 Control synthesis by MPC 237

***12.* MODELING AND CONTROL OF THE CONTINUOUS CASTING OF STEEL ... 240**

12.1 Continuous Casting Machine for Steel..... 241

12.2 Continuously Cast Steel Casting Products..... 243
12.2.1 CCM in Železiarne Podbrezová..... 244

12.3 Dynamic Model of the Casting Temperature Field 246

12.4 Model in the ProCAST Software Environment..... 248
12.4.1 Material Properties of Selected Steel Grades..... 249
12.4.2 Heat Transfer Coefficient..... 250

12.5 Measurement of the Surface Temperatures on the Casting 252

12.6 Casting in the Secondary Cooling Zone as DPS 252

12.7 Analysis of Temperature Field Dynamics 255
12.7.1 The Operating Mode of the Continuous Casting Machine 257
12.7.2 Change of Casting Speed..... 259
12.7.3 Change of Cooling Intensity..... 260
12.7.4 Identification of the System Dynamics..... 262

12.8 Control Tasks 263
12.8.1 Change of the Distributed Reference..... 265
12.8.2 Distributed Disturbance Rejection 267

12.9 Adaptive Control 270

12.10 Segmentation of Dynamics 271
12.10.1 Adaptive Compensation of Changes in the Casting Speed 272

12.11 Modelling the Macroscopic Structure of the Casting 274

***13.* OPTIMIZATION OF THE SPATIAL DISTR. DESIGN OF THE PLANT 276**

13.1 Groundwater remediation system design 276

***14.* CONCLUSION 279**

Prof. Ing. Gabriel Hulkó, DrSc., prof. Ing. Cyril Belavý, CSc.
Ing. Pavol Noga, PhD., Ing. Karol Ondrejkovič, PhD.

CONTROL OF TECHNOLOGICAL AND PRODUCTION PROCESSES AS DISTRIBUTED PARAMETER SYSTEMS
SUPPORTED BY ADVANCED NUMERICAL MODELING

Published by Slovak University of Technology in Bratislava in Publishing house of STU, Bratislava, Vazovova 5, in the year 2014.

Book series

Number of pages 295, 198 figures, 25 tables, 15,344 AH, 16,113 VH, 100 copies,
1. edition, editorial number 5814, printed in Publishing house of STU in Bratislava.

85 – 263 – 2014

ISBN 978-80-227-4290-0