Control Engineering Practice 66 (2017) 23–38

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Control of technological and production processes as distributed parameter systems based on advanced numerical modeling

G. Hulkó *, C. Belavý, K. Ondrejkovič, L. Bartalský, M. Bartko

Institute of Automation, Measurement and Applied Informatics, Mechanical Engineering Faculty, Slovak University of Technology, Bratislava Nám. Slobody 17, 812 31 Bratislava, Slovak Republic

A R T I C L E I N F O

Keywords:
Technological and production processes
Virtual software environments
Nonlinear distributed parameter systems
Control design
Secondary cooling in continuous casting of steel
Intelligent casting die

A B S T R A C T

The paper describes some practical control problems of technological and production processes as nonlinear distributed parameter systems. These are solved based on advanced numerical modeling in virtual software environments offered for the numerical dynamic analysis of technological and production processes with co-simulations. The controlled systems are interpreted as nonlinear lumped input and distributed parameter output systems. Synthesis of control in space relation is solved by approximation methods in temporal relation by methods of control of lumped parameter systems. Some results are demonstrated by the control of the secondary cooling in the continuous casting of steel, based on a software sensor. Furthermore, the control of a casting die preheating process is introduced in this framework using a programmable logic controller (PLC).

1. Introduction

Technological and production (TaP) processes often have a character of a distributed parameter system (DPS). The handling of such systems has a long history in control engineering, where the prevailing technical discussion is often on the control of systems governed by partial differential equations (PDE).

The design of control systems of DPS is often based on the early lumping approach in engineering practice. This means, that the process model is approximated prior to the control design using, e.g., finite-difference or finite-element techniques, proper orthogonal decomposition or the weighted-residual method. As a result, well-developed methods from linear and nonlinear finite-dimensional control theory can be applied for control and observer design; usually in the form of MIMO systems, see e.g. Jadachowski, Steinboeck, and Kugi (2016), Steinboeck, Wild, and Kugi (2013). In the late lumping approach, the control and observer design is directly based on the PDE description of the system dynamics. The results obtained this way are then adjusted for implementation into engineering practice (Jacob & Zwart, 2012; Meurer, 2013).

Thanks to the development of information technology, engineering practice frequently uses software products for the numerical analysis of the dynamics of various technical objects. These analyses are fundamentally based on the numerical solutions of the underlying nonlinear partial differential equations. The tasks in numerical analysis usually involve the investigation of time–spatial responses that are subject to manipulated input variables of the lumped character. A general lumped input and distributed parameter output system (LDS) can be formulated based on the results of the aforementioned analyses, in order to solve the control problems of TaP processes as DPS. The implementation stage then utilizes the lumping approach.

The underlying idea of this concept was published in the article by Hulkó et al. (2009) and successfully applied in the fields of energy systems (Hulkó, Rohal'-Ilkiv, Noga, & Lipár, 2012), extrusion of plastics (Lipár, Noga, & Hulkó, 2013), induction heating (Kapusta, Camber, & Hulkó, 2013), metallurgy (Ondrejkovič, Buček, Noga, & Hulkó, 2013; Ondrejkovič, Pyszko, & Hulkó, 2015) as well as in groundwater remediation control (Mendel, Kovács, & Hulkó, 2015). The present article that is based on the extended conference paper (Hulkó et al., 2016), this approach is further elaborated and generalized to the control of TaP processes as DPS.

Distributed outputs are sometimes at our disposal on the whole definition domain of the controlled system; however, often there are difficulties with the reliable distributed sensing of outputs. In these cases, control synthesis is solved by a software sensor, where modeled manipulated outputs are available in all locations of the definition domain. Frequently, technological and production processes take place in closed structures, isolated from external disturbances, where the outputs are measured only at selected locations of the definition domain.

* Corresponding author.
E-mail address: gabriel.hulko@stuba.sk (G. Hulkó).

http://dx.doi.org/10.1016/j.conengprac.2017.05.010
Received 26 November 2016; Received in revised form 31 May 2017; Accepted 31 May 2017

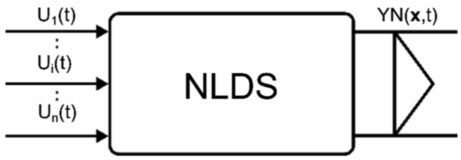

Fig. 2.1. Nonlinear lumped input and distributed parameter output system — NLDS.

Here, the aim of the control synthesis is to ensure the minimization of the distributed control error, based on measurements only in selected locations. Technological and production processes operate very often in linearized surroundings of steady state regimes. At transitions between these steady state regimes, the control problems are solved by the segmentation of the nonlinear transition dynamics into linearized segments. In the linearized surroundings of given steady state regimes dynamics of controlled systems is decomposed into temporal and spatial components. Control synthesis in the spatial relation is solved as approximation problem. For control synthesis in the temporal domain methods of lumped parameter systems control are used.

Some results of this approach will be demonstrated by the problem of controlling the secondary cooling in continuous casting of steel (Petrus, Zheng, Zhou, Thomas, & Bentsman, 2010; Mauder, Šandera, & Štetina, 2015; Hulkó et al., 2016). Furthermore, the casting die preheating process, using the framework introduced above, will be controlled by a programmable logic controller (PLC) (Yang et al., 2006; Belavý et al., 2012). The controller featuring a software sensor will use a Kalman filter-based estimation to provide state estimates for a constrained model predictive controllers (MPC). The demonstration example for the casting die preheater solves temporal feedback synthesis by robust PI controllers. Due to the nonlinear character of controlled systems, nonlinear dynamics will be turned into linearized segments. At solution of these control problems co-simulations of virtual software environments (ProCAST, 2010; COMSOL, 2012) and MATLAB (2014) are used.

2. Basic dynamical relations

Technological and production processes (TaP) as distributed parameter systems (DPS) between lumped manipulatable quantities $\{U_i(t)\}_i$ and distributed outputs $YN(\boldsymbol{x},t)$ actually represent nonlinear lumped input and distributed parameter output systems (NLDS), Fig. 2.1. Let us consider step changes of manipulatable input quantities $\{U_i(t)\}_i$ in the linearized surroundings of the chosen steady state of the NLDS. Here for simplicity the distribution of NLDS is given on the one-dimensional interval $[0, L]$, but all results are valid also for 3D. Responses can be considered as discrete distributed parameter transient characteristics with a unit sampling period $\{\mathcal{H}H_i(x,k)\}_i$, Fig. 2.2.

The distributed parameter impulse characteristics are given as subtractions of the shifted transient characteristics $\{\mathcal{G}H_i(x,k) = \mathcal{H}H_i(x,k) - \mathcal{H}H_i(x,k-1)\}_{i,k}$.

Then linear discrete convolution model with lumped inputs $\{U_i(k)\}_i$ and distributed parameter output $Y(x,k)$ gives linearized part of nonlinear dynamics of NLDS along with particular outputs $\{Y_i(\boldsymbol{x},k)\}_i$ in the form

$$Y(x,k) = \sum_{i=1}^{n} Y_i(x,k) = \sum_{i=1}^{n}\sum_{q=0}^{k} \mathcal{G}H_i(x,k-q)U_i(q) \tag{2.1}$$

which can be interpreted as linear discrete lumped input and distributed parameter output system with zero order hold units — HLDS, Fig. 2.3.

Inputs $\{U_i(k)\}_i$ act in the locations $\{x_i\}_i$ of the definition domain of HLDS and generate distributed transient characteristics. In order to simplify the explanation, let us consider in the following that distributed transient characteristics in the steady state $\{\mathcal{H}H_i(x,\infty)\}_i$ reach their

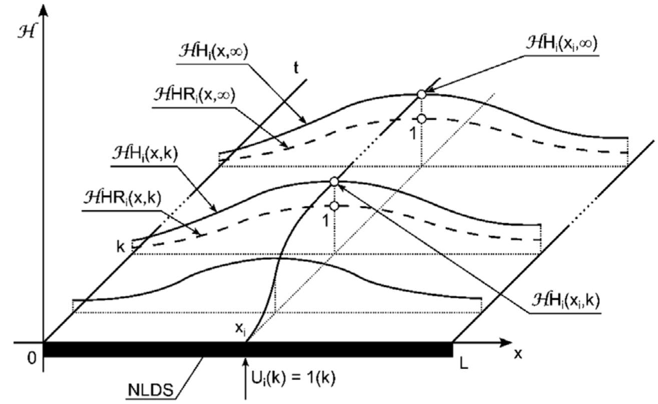

Fig. 2.2. i-th distributed parameter step response $\mathcal{H}H_i(x,k)$ with reduced profiles $\{\mathcal{H}HR_i(x,k)\}_k$.

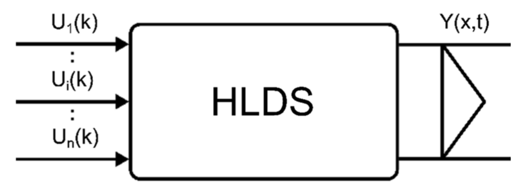

Fig. 2.3. Discrete linear HLDS with zero order holds.

maximal amplitudes in locations $\{x_i\}_i$: $\{\mathcal{H}H_i(x_i,\infty)\}_i$, Fig. 2.2. Moreover, let us introduce the reduced courses of these characteristics in steady state as

$$\{\mathcal{H}HR_i(x,\infty) = \mathcal{H}H_i(x,\infty)/\mathcal{H}H_i(x_i,\infty)\}_i \tag{2.2}$$

for $\{\mathcal{H}H_i(x_i,\infty) \neq 0\}_i$

Similarly we can generate reduced courses of particular distributed output quantities

$$\{YR_i(x,k) = Y_i(x,k)/Y_i(x_i,k)\}_{i,k} \tag{2.3}$$

for $\{Y_i(x_i,k) \neq 0\}_{i,k}$. Then

$$Y(x,k) = \sum_i Y_i(x,k) = \sum_i YR_i(x,k)\,Y_i(x_i,k). \tag{2.4}$$

Furthermore, for $k \to \infty$ the courses $\{YR_i(x,k)\}_{i,k}$ will converge to $\{\mathcal{H}HR_i(x,\infty)\}_i$, because

$$\left\{\begin{aligned} &YR_i(x,\infty) = Y_i(x,\infty)/Y_i(x_i,\infty) \\ &\quad = U_i(\infty)\mathcal{H}H_i(x,\infty)/U_i(\infty)\mathcal{H}H_i(x_i,\infty) = \mathcal{H}HR_i(x_i,\infty)\end{aligned}\right\}_i. \tag{2.5}$$

Moreover, according to (2.4) in locations $\{x_i\}_i$ we will have the following relation:

$$\begin{bmatrix} Y(x_1,k) \\ \vdots \\ Y(x_i,k) \\ \vdots \\ Y(x_n,k) \end{bmatrix} = \begin{bmatrix} YR_1(x_1,k) & ,\ldots, & YR_i(x_1,k) & ,\ldots, & YR_n(x_1,k) \\ YR_1(x_i,k) & ,\ldots, & YR_i(x_i,k) & ,\ldots, & YR_n(x_i,k) \\ YR_1(x_n,k) & ,\ldots, & YR_i(x_n,k) & ,\ldots, & YR_n(x_n,k) \end{bmatrix} \times \begin{bmatrix} Y_1(x_1,k) \\ \vdots \\ Y_i(x_i,k) \\ \vdots \\ Y_n(x_n,k) \end{bmatrix} \tag{2.6}$$

that is obtained by simple breakdown of previous results. This can be expressed in the following abbreviated form

$$\{Y(x_i,k)\} = \boldsymbol{YR}_i(x_i,k)\{Y_i(x_i,k)\} \tag{2.7}$$