UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SVM PERFORMANCE ASSESSMENT FOR THE CONTROL OF INJECTION MOULDING PROCESSES AND PLASTICATING EXTRUSION

Davide Anguita, Andrea Boni and Luca Tagliafico

January 2002

Technical Report # DIT-02-0035

Also: accepted by International Journal of Systems Science

SVM Performance Assessment for the Control of Injection Moulding Processes and Plasticating Extrusion

Davide Anguita[*], Andrea Boni[†], and Luca Tagliafico[‡]

Corresponding author: Andrea Boni

University of Trento, Via Mesiano, 77

I-38050 Povo di Mesiano (Trento) – Italy

Phone/Fax: +39-0461-882440/+39-0461-881977

e–mail: andrea.boni@ing.unitn.it

January 22, 2002

Abstract

This paper presents the application of a new and promising learning algorithm based on kernel methods, i.e., support vector machines (SVMs), for the control of injection moulding processes and plasticating extrusion. In particular, the main purpose of this work is to assess the effectiveness of the method when applied to such kinds of industrial processes, characterised by a large number of variables and strictly correlated by nonlinear relationships. First, we analyse the injection process by developing a simplified model, then we identify it by using a support vector machine. The reference of the control system is tracked through the design of a control block based on the structure of the SVM.

[*] Davide Anguita is with DIBE, University of Genova, Italy, e–mail: anguita@dibe.unige.it

[†] Andrea Boni is with the University of Trento, Italy, e–mail: andrea.boni@ing.unitn.it

[‡] Luca Tagliafico is with DITEC, University of Genova, Italy, e–mail: tgl@ditec.unige.it

1 Introduction

The extrusion process of molten polymeric materials is a well–established technique for the production of a large number of end products or components commonly found in a multitude of consumer goods. The wide variety of polymer blends and production techniques makes it quite difficult to define theoretically all the possible aspects of the control of the large number of process variables involved (thermal and mechanical), for a system with a highly nonlinear behaviour. Each variable (temperature, flow rate, pressure, etc.), has a great impact on both the final quality of a product and the production rate. Therefore, the choice of the control strategy and the development of new regulation tools become crucial to have more efficient, energy saving and environmentally safe extrusion processes.

In this paper, we propose the use of a new learning algorithm to develop a framework (based on intelligent modules), to be inserted in the regulation chain of an extrusion process. Our main goal is to verify its actual validity on the basis of experiments carried out by using a simplified thermal and fluid dynamic model of the injection process.

The injection moulding process (IMP) is one of the most important and widely used extrusion processes (Rosato 1995), and several new control approaches have been proposed in the past few years (Agrawal *et al.* 1987, Tsai and Lu 1998). In the IMP, the molten material is injected into the mould cavity at high speed and high pressure by means of an axially moving screw, activated by a piston moving at a given axial speed (Figure 1).

The quality of the extrudate is highly dependent on the temperature uniformity inside the polymer during the injection, on the working pressure, on the screw translation velocity, on the homogeneity of the physical properties obtained by the mixing process in the metering section, and on the temperature profile along the barrel. The temperature,in turn, is a complex combination of external heat and heat generated inside the polymer due to viscous dissipation. Each of these parameters must be carefully controlled to achieve a satisfactory quality of extrudates. One of the main operating parameters is the polymer temperature at the nozzle exit; this temperature is a very complicated function of material properties, process control parameters, screw geometry, durations of the different stages, residence time, and so on. It is indeed rather cumbersome to develop a global theoretical model able to describe the extruder behaviour in terms of the input-output system parameters. Therefore, the application of modern identification techniques based on neural networks (in a general way), in particular, on support vector machines (SVMs), seems to be quite attractive, especially for regulation and control purposes. The SVM module we propose in this paper acts as an identification block in the control loop of the IMP process.

In general, an identification problem can be defined as the process through which an unknown function, $\phi : \Re^d \rightarrow \Re$, describing the behaviour of a dynamic system, is estimated on the basis of some of its samples, $Z = \{(\boldsymbol{x}_i, t_i)\}_{i=1 \dots n_p}$. Usually, this problem can be tackled when the input/output signals of the system considered can be observed, but the system dynamics (i.e., the structure of ϕ) is not known. Typical applications include: time–series forecasting, identification and control of nonlinear systems, signal and image processing, and others. Support Vector Machines (SVMs) (Cristianini and Shawe–Taylor 2000) are a new paradigm that have recently been proposed to accomplish pattern–recognition and function–approximation tasks, hence they represent an attractive approach to solve the injection moulding problem just described. There is a long list of theoretical and practical advantages of SVMs over other connectionist regression methods (e.g., Multi-Layer Perceptrons - MLPs (Bishop 1995)). One of the most appealing properties is certainly the possibility of solving a quadratic programming problem subject to linear constraints (without local minima), instead of using a difficult nonlinear optimisation algorithm in the MLP case. Following the early theoretical studies of the properties of this new learning approach, different applications are starting to emerge, especially in the field of optimal control of nonlinear systems, as described in recent works (e.g., Suykens *et al.* 2001).

The first step in developing a control system based on an SVM is to assess if SVMs are capable to identify and reproduce the dynamic behaviour of the IMP. The basic idea of the present paper is to develop a simplified, thermal, fluid dynamic model of the injection stage, studying the SVM behaviour during the identification of the system for dynamic control purposes. The paper is organised as follows: the next section describes the thermal, fluid and dynamic model used to test the SVM method, briefly outlined in Section 3. In Section 4, we present the identification–control models and scheme that we have built to control the polymer temperature. The designs of the SVM and control blocks are also extensively discussed. In Section 5, we deal with the numerical experiments carried out on each building block of the whole control system; the performances of the dynamic model, the SVM and each control block are reported and discussed. Finally, in Section 6, we summarise the obtained results and propose further lines of research.

In the following text, we shall indicate the vectors and the matrices with lowercase and uppercase boldface letters, respectively; Tables 1, 2 and 3 give the variables and symbols used in this paper.

2 Thermal, fluid and dynamic model of the injection process

The injection process can be summarised as shown in Figure 2. In the first stage, the polymer is fed by using powder or pellets in the hopper: then it is pushed forward by rotating the screw, while the nozzle exit is closed to allow the "charge" material to fill the injection chamber. The polymer is molten inside the compression section by electric heating (barrel heaters) and viscous dissipation. Finally, it reaches a uniform temperature and thermophysical properties in the metering section. While the molten polymer fills the injection chamber, the screw goes slightly back in order to allow the desired quantity of polymer to accumulate in the nozzle, and new material is fed from the hopper. When the right polymer charge is ready to be injected, the piston pushes the screw (which behaves like a ram) axially along the barrel, thus allowing the mould to be filled with the polymer melt. A holding pressure is kept on the back of the piston until the whole mould is filled and the material inside it starts to solidify owing to external refrigeration systems. After the polymer at the nozzle exit (i.e., mould entrance) has solidified, the mould is removed and the cycle starts again. The whole process can be subdivided into three main stages, after the die removal:

1. Charge filling

2. Injection (translating – screw)

3. Mould filling and cooling (holding – screw)

A further, short stage can be added, which includes the removal of the filled mould and the insertion of a new, empty mould ready for the next cycle.

Neglecting, as a first approximation, the fluid compressibility and keeping in mind that the time-pressure profile at the nozzle exit is mainly driven by the cavity–pressure pattern inside the mould, the outlet pressure has been assumed to be an independent, given boundary condition, which may vary during the injection stage. For the same reason, the polymer introduced into the chamber by rotating the screw is assumed to be at a given, uniform temperature at the screw/polymer interface. The operating parameters are the volumetric flow rate of the screw, the polymer and cylinder working temperatures, the imposed displacement velocity of the screw during the injection stage, and the heat–transfer coefficients between the polymer and the inner cylinder walls, which can be heated and kept at the desired temperature.

For the present simplified approach, a one–dimensional, thermal, fluid dynamic model has been developed, making an explicit marching–time finite–volume analysis to calculate, at each time step, the temperature and pressure distributions inside the screw tip/nozzle assembly. All the different contributions to the energy balance have been introduced: conduction, convection, viscous–dissipation effects and dynamic behaviour. Furthermore, all the thermophysical properties have been assumed to vary with the temperature, the shear–stress and shear–rate, as required by the study of the flow behaviour of polymer melts, which are strongly non–Newtonian fluids.

The flow channel has been subdivided into small finite–volume elements, with variable cross sections (flow passages), depending on the channel geometry, as shown in Figure 3.

The mass, momentum, energy, and constitutive equations have been integrated over time, using different particular solutions in each of the three main steps of the extrusion moulding process, i.e., injection, holding and mould removal, and filling of a new charge. In the injection stage, the flow behaviour has been simulated by the momentum equation, applied to each volume in the form:

$$p_{i+1}^{j} = p_{i}^{j} + \Delta p_{i}^{j}, \quad i = 1, \ldots, N^{j}; \quad j = 0, \ldots, J \tag{1}$$

where p_i^j is the pressure on the left side in the element i, and Δp_i^j is the variation in pressure along the element i during the time interval j, as shown in Figure 4. N^j is the number of space elements, and ΔX^j is chosen over the time interval j. Depending on the kind of study being developed (simulation, control, design or optimisation), the pressure boundary condition can be given at the nozzle exit (cavity mould pressure) or on the basis of the screw–tip working conditions (interfacial surface area and pushing force during the injection stage). If the axial force F_s^j on the screw is given for each time interval j of the process (as usually happens in given conterpressure–control approaches), the pressure on the first polymer element is:

$$p_1^j = \frac{F_s^j}{A_s} \tag{2}$$

A_s stands for the frontal surface area of the screw. The evaluation of Δp_i^j has been performed by the simplified method of "representative viscosity," which allows the relationships derived for Newtonian material flows to be applied also in the case of polymer melts. As a result, the pressure drop in the channel is given by:

$$\Delta p_i^j = -\dot{V}^j \frac{\eta_i^j}{\tilde{K}_i^j f_{pi}} \tag{3}$$

where f_{pi} is a cross-section coefficient that is 1 for a circular cross-section, $\dot{V}^j$ is the volumetric flow rate, η_i^j is the local viscosity, and $\tilde{K}_i^j$ is the so-called die conductance, which takes on the form:

$$\tilde{K}_i^j = \frac{A_i^3}{2\Delta X^j P_i^2} \tag{4}$$

for constant cross-section flow passages, and the form:

$$\tilde{K}_i^j = \left(\frac{A_i^3}{2\Delta X^j P_i^2}\right)\left(\frac{3\left(\frac{R_{\max}}{R_{\min}} - 1\right)}{1 - \left(\frac{R_{\min}}{R_{\max}}\right)^3}\right) \tag{5}$$

for converging dies. A and P are the local cross–section area $[m^2]$ and the wetted perimeter $[m]$, respectively. To take into account the possibility that the screw axial velocity, $\dot{X}_s$, may be varied during the injection for control purposes, following a given velocity–profile in time, the volumetric flow rate is variable over each time interval j, according to the equation:

$$\dot{V}^j = \dot{X}_s^j A_s \cong \dot{X}_s^j \frac{\pi D_s^2}{4} \tag{6}$$

For the same reason, the axial discretisation too can be varied at each time step j, according to:

$$\Delta X^j = \Delta\tau \dot{X}_s^j \tag{7}$$

The local viscosity can be computed, as described in (Vinogradov and Malkin 1966), by using the universal temperature and pressure equation calculated as:

$$\eta\left(\dot{\gamma}\right) = \left(\frac{1 + a_1\left(\dot{\gamma}\eta_0\right)^{a_3} + a_2\left(\dot{\gamma}\eta_0\right)^{2a_3}}{\eta_0\left(T,p\right)}\right)^{-1} \tag{8}$$

where $\dot{\gamma}$ $[sec^{-1}]$ is the shear rate, the empirical coefficients a_1, a_2 and a_3 are, to a great extent, constant for each polymer over wide ranges of temperature and pressure values, and $\eta_0\left(T,p\right)$ is the reference viscosity in the extrapolated viscosity limit at a zero shear rate. For the particular polymer considered, the following numerical expressions can be given for the viscosity and shear rate:

$$\begin{aligned} & T_s = 130^{\circ}C \\ & T_s\left(p\right) = T_s + 0.03 p_b [\circ C] \\ & \eta_0\left(T_s\right) = 4.48 \cdot 10^6 [Pa \cdot sec] \end{aligned} \tag{9}$$

$$\eta_0\left(T,p\right) = \eta_0\left(T_s\right) \cdot e^{\frac{-8.86\left(T_c - T_s\left(p\right)\right)}{101.6 + \left(T_c - T_s\left(p\right)\right)}} [Pa \cdot sec] \tag{10}$$

where p_b is the pressure in bar units, T is the actual polymer temperature in $^{\circ}C$, and:

$$\dot{\gamma}_i^j = \frac{4\dot{V}^j}{\pi R_i^3} 0.815 \tag{11}$$

The energy equation has been used in its general form as a balance between the internal variation in the energy of the control volume (on the left) and all possible heat–transfer and mechanical–work contributions (on the right):

$$M_i^j c_{vi} \Delta T_i^j = \left[\psi_i^j - \xi_i^j\right] \Delta\tau - \zeta_i^j + \upsilon_i^j \tag{12}$$

$$
\begin{aligned}
& M_i^j = A_i \Delta X \rho_i^j \\
& \psi_i^j = h_c P_i \Delta X^j \left(T_{pi}^j - T_i^j \right) \\
& \xi_i^j = \tfrac{\tilde{k}(T_i^j)}{2\Delta X^j} \left[(A_i + A_{i-1}) \left(T_i^j - T_{i-1}^j \right) + (A_i + A_{i+1}) \left(T_i^j - T_{i+1}^j \right) \right] \\
& \zeta_i^j = A_i \Delta X^j \Delta p_i^j \\
& \upsilon_i^j = A_i \Delta X^j \rho_i^j c_{vi} \left(T_{i-1}^j - T_i^j \right) \\
& i = 1, \ldots, N^j - 1 \\
& j = 0, \ldots, J
\end{aligned}
\tag{13}
$$

where, for $i = N^j$, the adiabatic boundary condition at the exit gives:

$$
\xi_i^j = \frac{\tilde{k}\left(T_i^j\right)}{2\Delta X^j} (A_i + A_{i-1}) \left(T_i^j - T_{i-1}^j \right) \tag{14}
$$

The variation in the temperature of the element i over the time interval j–$j+1$ is expressed by the explicit marching–time schema:

$$
\Delta T_i^j = T_i^{j+1} - T_i^j \tag{15}
$$

In the above equations, M $[kg]$ is the mass, c_{vi} $\left[J(kg^\circ K)^{-1}\right]$ is the constant–volume specific heat, A_i $[m^2]$ is the local duct cross–section, ρ $\left[kg(m^3)^{-1}\right]$ is the polymer density, ψ_i $[W]$ is the mean convective–heat flow rate, ξ_i $[W]$ is the net conductive–heat flow rate inside the polymer in the axial direction, ζ_i is the compression work, and υ_i is the axial energy flow due to the polymer motion. In general, the barrel temperature T_p can be variable in space and time according to temperature–control strategies; therefore, a generic T_{pi}^j has been used in equation 13.

The proposed simplified approach is based on the idea of assuming that the finite volume ΔX^j plays a double role, that is, as both a space interval and a screw displacement over the time interval $\Delta\tau$. Assuming a constant, uniform time interval for the entire flow simulation, the flow channel is discretised every time with a different space size, depending on the actual screw axial velocity. The use of an explicit marching–time algorithm allows us to calculate a spline approximation for the temperature profile at the time j and to re-evaluate the initial temperature distribution in the new discretised spatial domain over the time interval $j + 1$. The solution of the set of equations (1)–(15) follows a calculation schema of the type illustrated in Figure 5, i.e., by giving the temperature and pressure profiles along the channel during an arbitrary number of injection cycles.

3 An overview of SVMs for function approximation

Let us outline here the SVM approach to function approximation. Usually, a typical function–approximation task is defined as follows: a set of training points of the unknown function $\phi(\cdot)$ to be estimated are given after, for example, a design of experiments. Let us indicate such a set as: $Z = \{(\boldsymbol{x}_i, t_i)\}_{i=1\ldots n_p}$, where $\boldsymbol{x} \in \Re^d$ is an input vector and $t_i \in \Re$ is the corresponding output of $\phi(\cdot)$. This is also indicated as a regression problem, and different classic or advanced techniques can be used (most applications adopt B–splines, polynomial functions or different flavours of neural networks, like, for example, the Multilayer Perceptron). The task in which a function is estimated on the basis of some of its samples is also called the *learning problem*. At the end of the '90s, a new paradigm for learning by examples, called Support Vector Machines, was obtained thanks to the research work by the Russian mathematician V. Vapnik and his group. SVMs are based on two different theories developed in the '60s and '70: Statistical Learning Theory, by Vapnik and Chervonenkis (Vapnik 1998), and the theory for Reproducing Kernel Hilbert Space (RKHS) (Wahba 1999). The RKHS framework plays a basic role in the efficiency of SVMs because it makes possible to obtain nonlinear approximation functions. Basically, a non–linear SVM can be built by mapping each input pattern $\boldsymbol{x}$ into a different high–dimensional *feature space* (in the RKHS framework) through a nonlinear transformation $\Phi : \Re^d \rightarrow \Re^D, D >> d$. In that space, the use of Mercer's theorem permits the interpretation of kernels as inner products, thus bypassing the need for a direct knowledge of $\Phi(\cdot)$.

SVMs have recently provided very good performances in accomplishing classification and function–approximation tasks. The advantage of using SVMs over other methods is twofold. The structure of the optimisation problem, which consists in the resolution of a constrained quadratic problem (CQP), overcomes many typical drawbacks of classical neural–network approaches; for example, the plague of local minima that affects the back-propagation algorithm is completely avoided in SVM learning. Furthermore, the intrinsic properties of the method for controlling the complexity of the model (i.e., the Structural Risk Minimisation inductive principle (Vapnik 1998)), guarantee considerable generalisation capacity.

Briefly, the main goal of ε-SV regression is to find a function $\hat{\phi}(\cdot)$ that has at most an ε deviation from the targets t_i for all the points and, at the same time, that is as flat as possible (for more details on the use of SVMs for regression purposes see (Cristianini and Shawe–Taylor 2000, Schölkopf and Smola 1998)). The function $\hat{\phi}(\cdot)$ is given, in general form, by:

$$\hat{t} = \hat{\phi}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{n_p} (\beta_i - \beta_i^*) K\left(\boldsymbol{x_i}, \boldsymbol{x}\right) + b \qquad (16)$$

where the free parameters are found by solving the dual formulation:

$$\min E(\boldsymbol{\alpha}) \quad = \quad \frac{1}{2}\boldsymbol{\alpha}^t \boldsymbol{H} \boldsymbol{\alpha} + \boldsymbol{c}^t \boldsymbol{\alpha} \qquad (17)$$

under the constraints $0 \leq \boldsymbol{\alpha} \leq C$ and $\boldsymbol{\alpha}^t \boldsymbol{y} = 0$. C measures the tradeoff between the deviation of each target from the function and the flatness of the function itself, and all the other matrices and vectors are defined as follows:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{Q} & -\boldsymbol{Q} \\ -\boldsymbol{Q} & \boldsymbol{Q} \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\beta}^* \end{bmatrix} \quad \boldsymbol{c} = \begin{bmatrix} \boldsymbol{\varepsilon} - \boldsymbol{t} \\ \boldsymbol{\varepsilon} + \boldsymbol{t} \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} \tag{18}$$

The threshold b is found through the Karush-Kuhn-Tucker conditions at optimality; $q_{ij} = q_{ji} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\varepsilon_i = \varepsilon, \forall i$, $\boldsymbol{\alpha}$, $\boldsymbol{c}$, $\boldsymbol{y} \in \Re^{2n_p}$, $\boldsymbol{\beta}$, $\boldsymbol{\beta}^* \in \Re^{n_p}$; $\mathbf{1}$ is a vector with all ones of n_p elements, whereas $\boldsymbol{t}$ is the vector of the targets; $K(\cdot, \cdot)$ is a *kernel function* in the RKHS. The use of different kernel functions changes the mapping from the input to the feature space; therefore, it modifies the SVM structure and the corresponding approximation surface. Among available kernels, the most widely used are linear ($K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$, where "$\cdot$" is a simple dot–product), polynomial and Gaussian of the form:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}} \tag{19}$$

where σ controls the amplitude of the RBF and the generalisation ability of the SVM. The described CQP can be solved by using traditional optimisation techniques (Bertsekas 1995), or special oriented algorithms (Platt 1999, Joachims 1999).

The models developed so far require the setting of two kinds of different parameters:

1. the vector $\boldsymbol{\alpha}$ and b, called *functional variables*, which can be selected through the solution of the CQP;

2. a set of *structural variables* (C, σ and ε[1]).

The process by which one searches for optimal values of the structural variables is also known as a model–selection task, and different approaches, based on statistical methods, have recently been suggested (see, for examples, the method described in (Chapelle and Vapnik 2000). It is known that the most critical parameter is the variance σ; here we use the standard bootstrap technique discussed in (Anguita *et al.* 1999) for the selection of σ.

[1] Note that, in equation (16), we have indicated the collection of the functional and structural variables with the vector $\boldsymbol{w}$.

4 Identification and control models by using SVMs

In this section, we show how the SVM algorithm described previously can be used for:

1. reproducing the behaviour of the thermal, fluid, dynamic model defined in Section 2 (design of the SVM–block).

2. generating the command–input to the model (design of the control–block).

Here we use a typical *identification + control* approach, as sketched in the schema of Figure 6, where:

- t is the actual temperature (at the nozzle) that, in our case, is generated by the model (in Section 2 we have indicated it with the symbol $T^j_{N^j}$, the temperature of the last element at the time j); usually, this is also referred to, in system control theory, as the output of a *plant*;

- $\hat{t}$ is the output of the SVM (equation 16);

- $\hat{e}$ is the error between the output of the model and the output of the SVM (it is the identification error, and depends on the approximation capability of the SVM itself);

- r (reference) is the desired temperature at the nozzle;

- e is the control error;

- u is the control signal or the command input to the system; it is the temperature of the barrel heater at the injection stage (in Section 2, we have indicated it, in a general way, as T^j_{pi}).

The control–block provides the command u to the system, on the basis of the structure of the SVM–block, in order to track the reference temperature r through the minimisation of the error e. Note that the actual system is supposed to be unknown; therefore, the only way of building the control–block is to refer to the structure of the SVM–block. In the following, we describe how such a block has been designed.

The thermal, fluid, dynamic model presented in Section 2 is a discrete–time description ($\Delta\tau$ is the corresponding sample time) of the temporal evolution of the injection moulding process. In general, the output of a discrete–time system at the time $k+1$ can be represented as a function of n previous samples of the output itself and m previous samples of the signal input. The relationship is:

$$\begin{aligned} t(k+1) &= \phi\left(t(k), t(k-1), \ldots, t(k-n+1), u(k), u(k-1), \ldots, u(k-m+1)\right) = \phi\left(\boldsymbol{x}(k)\right) \\ \boldsymbol{x}(k) &= \left[t(k), t(k-1), \ldots, t(k-n+1), u(k), u(k-1), \ldots, u(k-m+1)\right]^T \end{aligned} \tag{20}$$

$\boldsymbol{x} \in \Re^d$, $d = n + m$. At the time $k+1$, the SVM gives an estimate of $t(k+1)$, called $\hat{t}(k+1)$, on the basis of equation (16), that is, $\hat{t}(k+1) = \phi^*\left(\boldsymbol{x}(k), \boldsymbol{w}\right)$; this is also known, in the neural–network literature as an *identification–serial–parallel model* (Narendra and Parthasarathy 1990). Note that, as in an on–line process the exact measurement of the output of a plant, at each sample time, is often a difficult task, a *parallel identification model* can be alternatively used. In this case, the input to the SVM depends on the previous samples of its output, rather than on the output of the actual system:

$$\hat{t}\left(k+1\right) = \phi^*\left[\hat{t}\left(k\right), \cdots, \hat{t}\left(k-n+1\right), u\left(k\right), \cdots, u\left(k-m+1\right), \boldsymbol{w}\right] \tag{21}$$

The former identification model is also preferred to the latter because it guarantees a more accurate system stability, as stated in (Narendra and Parthasarathy 1990).

In order to design the SVM block (i.e., the setting of $\boldsymbol{w}$), the following steps have to be executed in sequence:

1. building a training set Z and setting the parameter vector $\boldsymbol{w}$, as described in the previous section (note that functional and structural variables can be selected by using the same Z, as shown by the procedure used in (Anguita *et al.* 1999));

2. building a separated validation set V, to compute the actual error of the SVM.

T build Z, we choose the curves in Figure 7; in this way, we are able to collect different samples, representing the evolution of the process when different operating parameters are applied. In other words, we assume a given initial temperature (470 K), and then we set several values of the barrel heater (i.e., the signal u in our control system schema). Subsequently, we collect the output t, given by the model, at each sample time: $t(0), t(1), t(2), \ldots$, etc. The pairs $\{(\boldsymbol{x}_i, t_i)\}$ to be given to the training algorithm have been simply designed in the following way:[2]

$$\begin{aligned}
&[(t(n-1), \cdots, t(0), u(n-1), \cdots, u(n-m)), t(n)] = (\boldsymbol{x}_1, t_1) \\
&[(t(n), \cdots, t(n-n+1), u(n), \cdots, u(n-m+1)), t(n+1)] = (\boldsymbol{x}_2, t_2) \\
&[(t(n+1), \cdots, t(2), u(n+1), \cdots, u(n-m+2)), t(n+2)] = (\boldsymbol{x}_3, t_3) \\
&\vdots
\end{aligned} \tag{22}$$

and so on.

The job of the controller is to provide the signal control u in such a way to minimise the control error e on the basis of the SVM–block. The idea is quite simple and based on the following criteria (given in (Noriega and Wang 1998)). A cost Ψ measuring the quadratic error between the reference and the output of the SVM is defined as:

$$\Psi = \frac{1}{2} e^2 (k+1) = \frac{1}{2} \left[r(k+1) - \hat{t}(k+1) \right]^2 \tag{23}$$

The goal is to minimize Ψ by finding a simple control law for u:

$$u(k+1) = u(k) - \mu \frac{\partial \Psi}{\partial u(k)} \tag{24}$$

where μ is the control descent step to be properly set in order to avoid undesired oscillations. If a linear and a Gaussian kernel are supposed to be used, it is possible to obtain,

[2] Note that, in fact, n and m are unknown parameters; their optimal values have to be established through experiments.

by applying equation (24) to equation (16), the following control laws:

Controller 1 (linear kernel):

$$u\left(k+1\right)=u\left(k\right)+\mu e\left(k+1\right)\sum_{i}\left(\beta_i-\beta_i^*\right)x_{i,n+1} \tag{25}$$

Controller 2 (Gaussian kernel)

$$u\left(k+1\right)=u\left(k\right)+\frac{\mu}{\sigma^2}e\left(k+1\right)\sum_{i}\left(\beta_i-\beta_i^*\right)\left(x_{i,n+1}-u(k)\right)e^{-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}(k)\|^2}{2\sigma^2}} \tag{26}$$

by $x_{i,n+1}$ we denote the component $n+1$ of the vector $\boldsymbol{x}_i$.

To complete this section, we provide some considerations about the stability of the proposed control system. Let us observe that the control algorithm just described is based on an intelligent learning methodology, which can be qualitatively analysed by using the Lyapunov theory. Basically, nonlinear control systems like the one described in this section guarantee stable points as long as the following conditions hold true:

- the SVM–block converges to the actual physical system;
- the control law stabilises the SVM–block.

We can proceed as in (Noriega and Wang 1998). This is confirmed by the fact that the case of a Gaussian kernel is a special case of RBF neural networks, discussed in that work. Furthermore, the universal approximation properties of SVMs have been extensively demonstrated by Vapnik (Vapnik 1998).

5 Numerical results and SVM performances

The experiments described in this section concerned each block in figure 6. In particular, here we show the behaviour and the performance of the thermal, fluid, dynamic model defined in Section 2, the procedure adopted to train the SVM-block, and the way we chose the parameters of the control–block.

A number of calculations were performed with reference to the converging duct shown in Figure 3; its thermophysical properties, geometrical and operating parameters are given in Tables 4 and 5. The time history of the outlet temperature is presented in Figure 8 and some axial temperature profiles inside the injection chamber are shown in figure 9. The time analysis shows clearly that each injection stage was characterised by a sharp increase in temperature, due to viscous dissipation, while the global trend was a slow decrease in temperature due to the over–warmed polymer (initially, at $470K$), as compared with to the heater–barrel temperature ($T_p = u = 460K$). The axial temperature profile of the polymer inside the chamber points out that, as far as the polymer flowed slowly in contact with a lower cylinder temperature, the axial temperature decreased steadily, while only in the last sharp convergence section did the temperature increase.

In the design of the SVM–block our main goal has been to assess the performances of the parallel and serial–parallel models, as described in the previous section. In the first experiment, a uniform initial temperature (fixed at $470K$) of the polymer inside the charge was assumed. Then we generated two different profiles of the outlet polymer temperature to build the training set Z (training 1 and training 2 respectively, as shown in Figure 10). In the first realisation (training 1), the temperature of the barrel heater was increased from $462K$ to $464K$, and from $462K$ to $464K$ in the second one (training 2). The test was performed by setting the value of the command input at 463K. Figure 10 shows the output of two different SVMs (parallel and serial–parallel), and the output of the model–plant. Note that we used $n = 7$ samples of the output and $m = 1$ sample of the input. We used a Gaussian kernel with $\sigma^2 = 0.05$ and $\varepsilon = 10^{-3}$, following the bootstrap technique described in (Anguita et. al 1999), to minimise the generalisation error of the SVM. From Figure 10, it is possible to observe that the output of the model–plant reached a stationary temperature of $466.3K$, and was well–identified by the serial–parallel SVM.

However, when the SVM was completely uncoupled from the model (parallel SVM), a worst performance was obtained. In this case, the SVM–block produced a significant error. Thanks to these experiments, we deduced that only two curves used as a training set are insufficient information about the system behaviour, so the range of operating temperatures was extended, as shown in Figure 7.

Once the SVM has been designed, it is possible to set up the controller by using controller 1 or controller 2, as described in the previous section. Figure 11 gives an example of application of controller 2 based on a parallel–SVM; we set the reference value to 466.3K and 468K, respectively; we used a sample time of 0.2 $[sec]$ and a descent step of $\mu = 0.8$. We carried out several numerical experiments also for controller 1. From the obtained results, we deduced that the thermal, fluid, dynamic model presented in section 2 can be identified and controlled, to a good approximation, also by using a linear kernel. Figure 12 shows the responses of the system to different values of μ (the reference value was fixed at $466K$), which controls the speed of convergence of the closed loop to the reference value. Our choice is a good tradeoff between the speed of convergence to a stationary temperature and the magnitudes of the oscillations.

6 Discussion

In this paper we have reported on a first preliminary study of the applicability of machine–learning methods for the identification and control of injection moulding processes and plasticating extrusion. The obtained results have shown the efficiency of the SVM approach when applied to such kinds of industrial processes. However, it is worth noting that the actual use of the whole framework must be preceded by the design of effective experiments in order to collect real samples $(\boldsymbol{x}_i, t_i)$ of the process considered.

Further work has to be done for each of the building blocks in Figure 6. In the following, we address some critical points that need to be investigated in detail.

The thermal, fluid, dynamic model discussed in Section 2 has been designed under two basic assumptions: first, in all the experiments, we have supposed a uniform initial temperature (fixed at 470K) of the polymer inside the charge; in other words, we have not theoretically modelled the very first stage in which the solid–state polymer is pushed inside the extruder and reaches the desired plasticating temperature. This stage is very difficult to define analytically, even by using approximate model. Second, the basic hypothesis for a correct behaviour of the model is incompressibility property of the polymer. Only under this strong assumption does a volume variation in the first space element, due to the axial movement of the screw in $\Delta\tau$, produce an equal volume variation in the polymer at the nozzle exit. Both assumptions can be obviously revised by analysing the behaviour of the injection moulding process in more detail. This would inevitably cause a significant change in the structure of the whole set of equations and, accordingly, an increase in their complexity. As a consequence, a simple simulation and an easy interpretation of the results, which have been our main goals for the development of a general framework based on an SVM, would be more difficult to obtain.

As a final remark, related to the SVM–block, we note that, if a reference value not included in the pre-defined range is imposed, the general stability conditions considered in the previous sections will not be verified any more. This is a typical behaviour of control systems based on neural approaches. For example, Figure 13 has been obtained by setting the reference value to 475K. In this case, to improve the system one could proceed in two different ways: 1) designing a fault–tolerant block (which helps to recover a correct functionality) through the measurement of the identification error $\hat{e}$, as described in (Noriega and Wang 1998); 2) training the SVM–block by an on–line recursive algorithm, as described in recent works (e.g., Cauwenberghs and Poggio 2001). Note that the latter approach needs further investigations because up to now many points have not been clarified; for example, it is not clear how to keep track of previous samples when a new one will arrive.

References

AGRAWAL, A. R. , et. al., 1987, Injection molding process control. *Polymer Engineering and Science*, **27**, (18), 17–32.

AIZERMAN, M., BRAVERMAN, E., and ROZONOER, L., 1964, Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, **25**, 821–837.

ANGUITA, D., BONI, A., and RIDELLA, S., 1999, Evaluating the generalization ability of support vector machines through the bootstrap. *Neural Processing Letters*, **11**, 1–8.

BERTSEKAS, D. P.,1995, *Nonlinear Programming* (Belmont, Mass., Athena Scientific).

BISHOP, C., 1995, *Neural Networks for Pattern Recognition* (Oxford, Clarendon Press).

CHAPELLE, O., and VAPNIK, V., 2000, Model selection for support vector machines. In Solla, S. A., Leen, T. K., and Müller, K. R. (Eds.) *Advances in Neural Information Processing Systems 12* (MIT press).

CAUWENBERGHS, G., and POGGIO, T., 2001, Incremental and decremental support vector machine learning. In Solla, S. A., Leen, T. K., and Müller, K. R. (Eds.) *Advances in Neural Information Processing Systems 13* (MIT press).

TSAI, C–C , and LU, C–H, 1998, Multivariable self-tunig temperature control for plastic molding process. *IEEE Transactions on Industry Applications*, **34** (2).

CRISTIANINI, N., and SHAWE–TAYLOR, J., 2000, *An Introduction to Support Vector Machines* (Cambridge University Press, Cambridge, UK).

JOACHIMS, T., 1999, Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., and Smola, A. (Eds.) *Advances in Kernel Methods – Support Vector Learning* (MIT Press, MA, USA).

NARENDRA, K. S., and PARTHASARATHY, K., 1990, Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, **1**(1), 4–27.

NORIEGA, J. R., and WANG, H.,1998, A direct adaptive neural–network control for unknown non-linear system and its application. *IEEE Transactions on Neural Networks*, **9** (1).

PLATT, J., 1999, Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C., and Smola, A. (Eds.) *Advances in Kernel Methods – Support Vector Learning* (MIT Press, MA, USA).

ROSATO, D. V., 1995, *Injection Molding Handbook: the Complete Molding Operation Technology, Performance, Economics*(2nd ed., New York).

SCHÖLKOPF, B., and SMOLA, A., 1998, A tutorial on support vector regression. NeuroCOLT2 Technical Report Series, NC2–TR–1998–030.

SUYKENS, J. A. K., VANDEWALLE, J., and DE MOOR, B., 2001, Optimal control by least squares support vector machines. *Neural Networks*, **14**, 23–35.

VAPNIK, V.,1998, *Statistical Learning Theory* (Wiley–Interscience Pub.).

VINOGRADOV, G. V., MALKIN, A. Ya., 1966, Rheological properties of polymer melts. *Journal of Polymer and Science, Part A-2*, **4**, 135–154.

WAHBA, G., 1999, Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In Schölkopf, B., Burges, C., and Smola, A. (Eds.) *Advances in Kernel Methods – Support Vector Learning* (MIT Press, MA, USA).

List of Tables

1 Nomenclature (1). 25

2 Nomenclature (2). 26

3 Nomenclature (3). 27

4 Duct geometry (Figure 3). 28

5 Polymer thermophysical properties and values of the operating parameters. 28

List of Figures

1 The extruder. 29

2 The injection process: injection (or mould filling) - holding (or packing) - cooling and die removal - filling of the charge chamber. 30

3 Screw-channel assembly and finite volume discretization: i is the discrete axial co-ordinate; j is the discrete–time co-ordinate. 31

4 Pressure variation in the element i. 31

5 Calculation schema for the thermal, fluid, dynamic model. 32

6 A typical identification–control scheme. 33

7 Outlet polymer temperature, used for training, at the nozzle exit for different settings of the barrel heater. 34

8 Outlet polymer temperature at the nozzle exit. 35

9 Temperature distribution in the chamber for successive injection cycles of the polymer extrusion temperature (at the chamber inlet) $T_0 = 470K$ and the uniform barrel temperature $T_p = 460K$. 36

10 Training profiles (training 1 and training 2) and SVM output for the parallel and serial–parallel models. 37

11 Closed–loop outlet polymer temperature at the nozzle exit (plant) and at the SVM–block exit (SVM) for different reference values. 38

12 Closed–loop outlet polymer temperature for different values of the descent step parameter μ. 39

13 Example of system instability; outlet polymer temperature at the SVM–block exit for a reference value beyond the range. 40

Table 1: Nomenclature (1).

A	cross section
C	regularisation constant
D_{max}/D_{min}	max. and min. diameters
D	feature–space dimension
E	quadratic function
$\boldsymbol{H}$	quadratic function Hessian matrix
F	axial force
J	No. of discrete–time elements
$K(\cdot,\cdot)$	kernel function (inner product in feature space)
$\tilde{K}$	die conductance
$L_{conv/end/tot}$	screw lengths (Figure 3)
N	No. of axial elements
P	wetted perimeters
$\boldsymbol{Q}$	quadratic function Hessian matrix block
$R_{max/min}$	max. and min. radii
$T/T_p/T_0$	polymer/barrel/polymer–extrusion temperatures
$\dot{V}$	volumetric flow rate
X_s	screw axial co-ordinate
$\dot{X}_s$	screw axial velocity
Z	training set

Table 2: Nomenclature (2).

$a_{1/2/3}$	empirical coefficients
b	threshold
c	quadratic function linear term
c_v	constant–volume specific heat
d	input space dimension
e	control error
$\hat{e}$	identification error
f_p	cross–section coefficient
h_c	polymer/wall transfer coefficient
$i/j/k$	discrete–axial/discrete–time(model)/discrete–time(control loop) counters
$\tilde{k}$	mean polymer thermal conductivity
m	No. of signal–control samples
n	No. of signal output (plant/SVM) samples
n_p	No. of training patterns ($n_p = dim(Z)$)
p	pressure
r	control loop reference
t	plant output
$\hat{t}$	SVM output
$\boldsymbol{t}$	targets vector
u	control–signal in the control loop
$\boldsymbol{w}$	SVM free parameters vector
$\boldsymbol{x}$	SVM input vector
$\boldsymbol{y}$	vector of ones and minus ones

Table 3: Nomenclature (3).

$\Phi(\cdot)$	the mapping $\Re^d \rightarrow \Re^D$
Ψ	control–loop functional cost
$\boldsymbol{\alpha}$	quadratic function free parameters
$\boldsymbol{\beta}$	n_p–dimension component vector of $\boldsymbol{\alpha}$
$\boldsymbol{\beta}^*$	n_p–dimension component vector of $\boldsymbol{\alpha}$
ε	loss function insensitivity parameter
$\phi(\cdot)$	plant I/O relationship
$\hat{\phi}(\cdot)$	SVM I/O relationship
$\dot{\gamma}$	shear rate
η	local viscosity
μ	control–block descent step
ρ	local density
ψ	mean convective heat flow rate
ξ	net conductive heat flow rate
ζ	compression work
υ	axial energy flow
σ^2	variance of the Gaussian kernel

Table 4: Duct geometry (Figure 3).

D_{max}	$0.05\ [m]$	max. diameter
D_{min}	$0.008\ [m]$	min. diameter
L_{tot}	$0.39\ [m]$	total charge length
L_{conv}	$0.15\ [m]$	converging section length
L_{end}	$0.02\ [m]$	termination duct length

Table 5: Polymer thermophysical properties and values of the operating parameters.

c_v	$2000\ [J/(kgK)]$	constant–volume specific heat
ρ	$780\ [kg/m^2]$	polymer density
k	$0.16\ [W/mK]$	mean polymer thermal conductivity
τ_I	$2\ [s]$	injection
τ_M	$2\ [s]$	holding
τ_R	$6\ [s]$	filling
$\Delta\tau$	$0.2\ [s]$	time step interval
T_0	$470\ [K]$	inlet temperature (extrusion screw output)
T_p	$465\ [K]$	barrel temperature
$\dot{X}_s$	$0.11\ [m/s]$	screw axial velocity
p_{init}	$1.5\ 10^8 [Bar]$	polymer/screw interface pressure
h_c	$100\ [W/m^2K]$	polymer/wall heat transf. coeff. (injection)
h_{cm}	$30\ [W/m^2K]$	polymer/wall heat transf. coeff. (holding)
h_{cr}	$30\ [W/m^2K]$	polymer/wall heat transf. coeff. (filling)

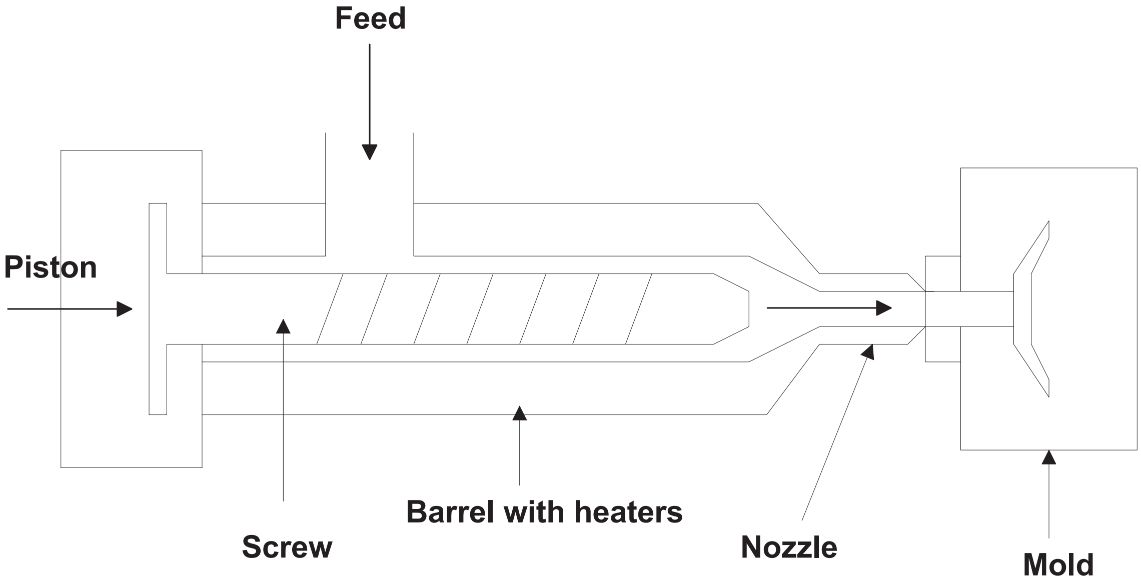

Figure 1: The extruder.

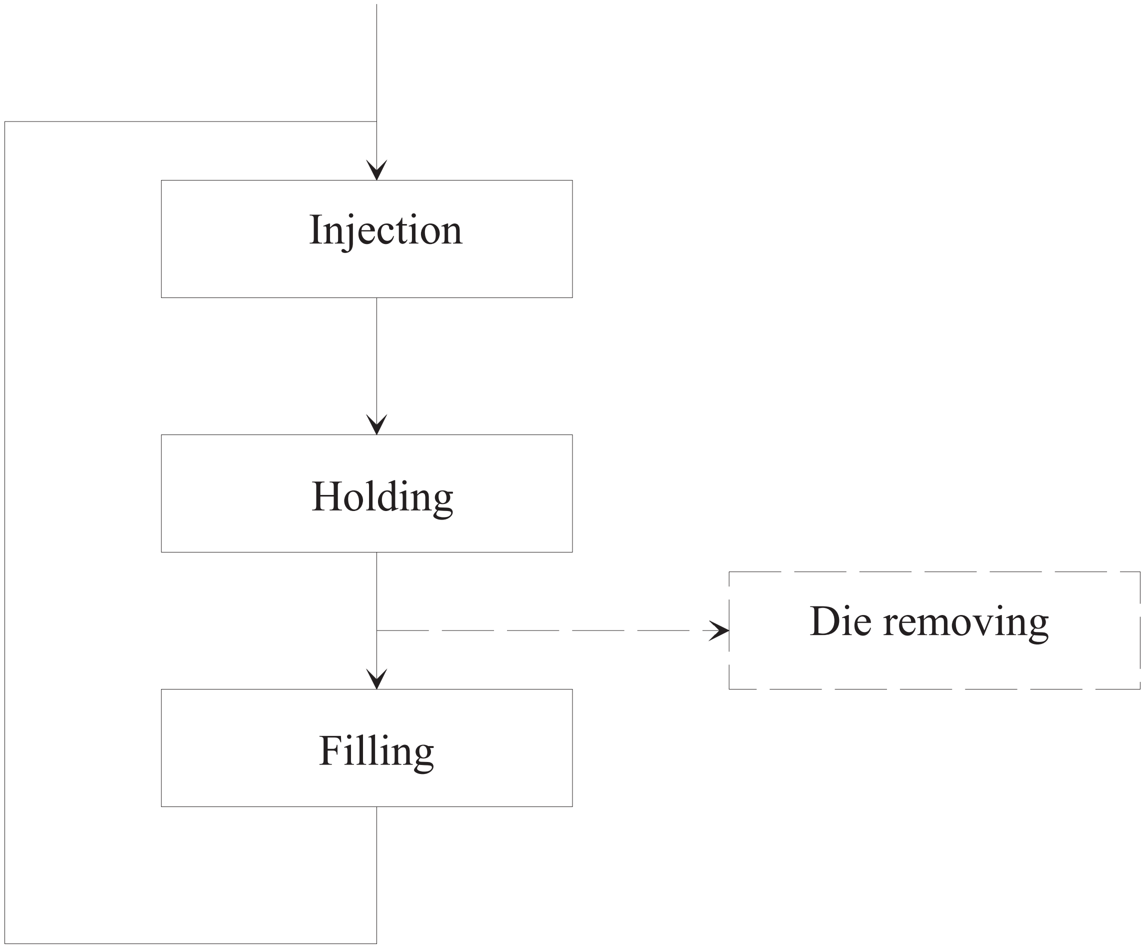

Figure 2: The injection process: injection (or mould filling) - holding (or packing) - cooling and die removal - filling of the charge chamber.

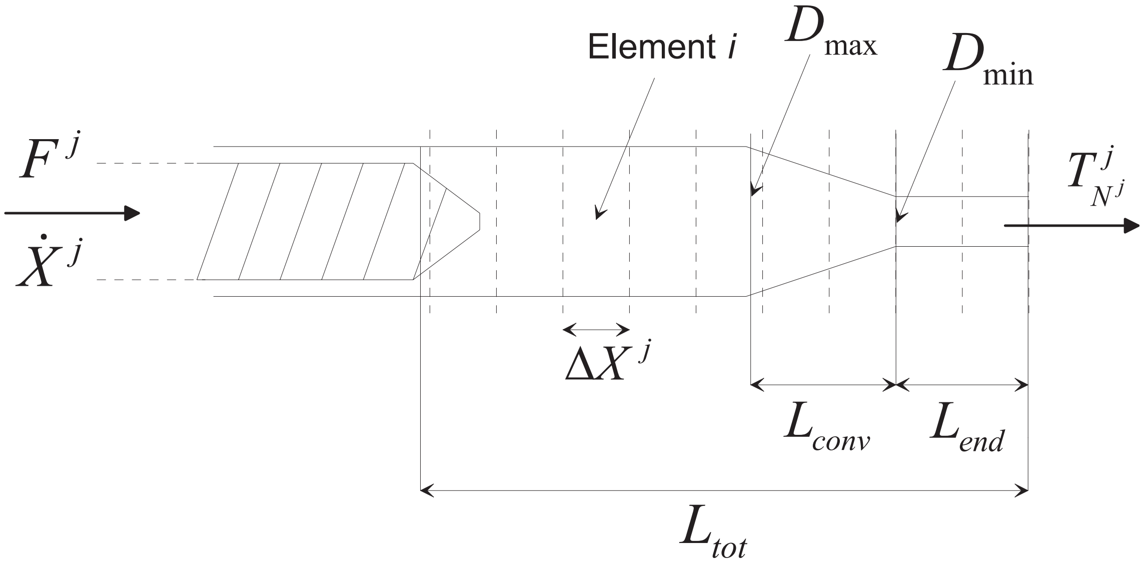

Figure 3: Screw-channel assembly and finite volume discretization: i is the discrete axial co-ordinate; j is the discrete–time co-ordinate.

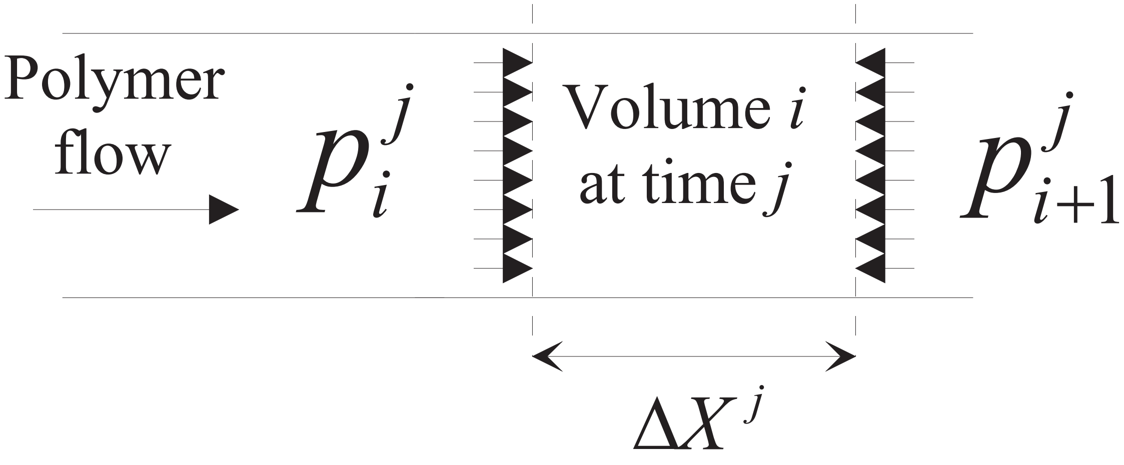

Figure 4: Pressure variation in the element i.

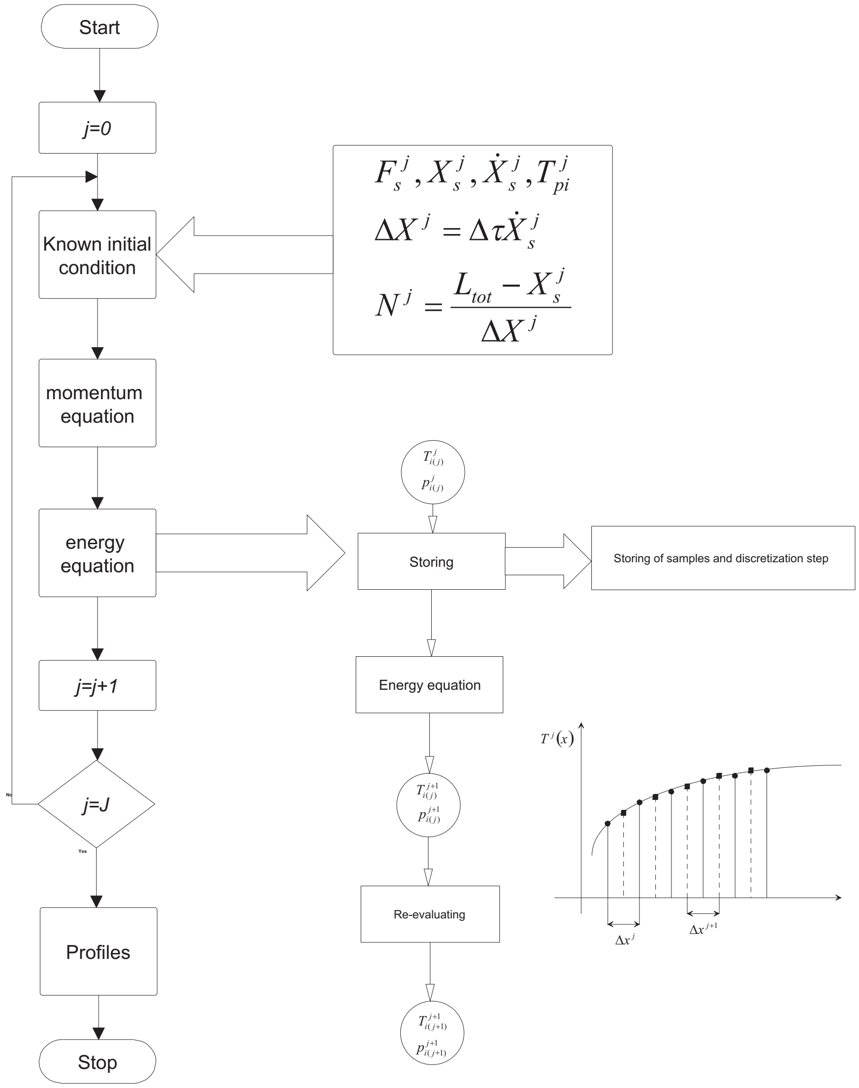

Figure 5: Calculation schema for the thermal, fluid, dynamic model.

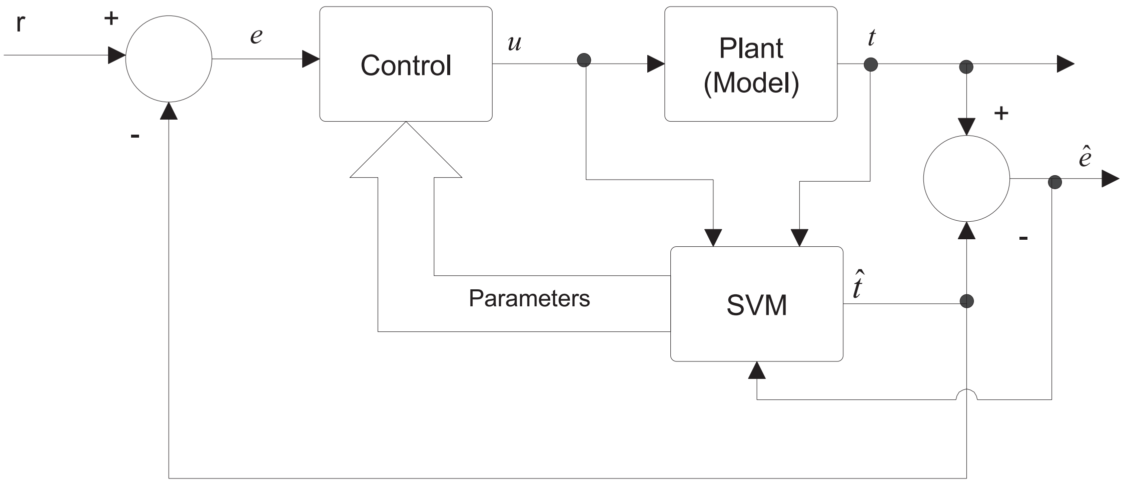

Figure 6: A typical identification–control scheme.

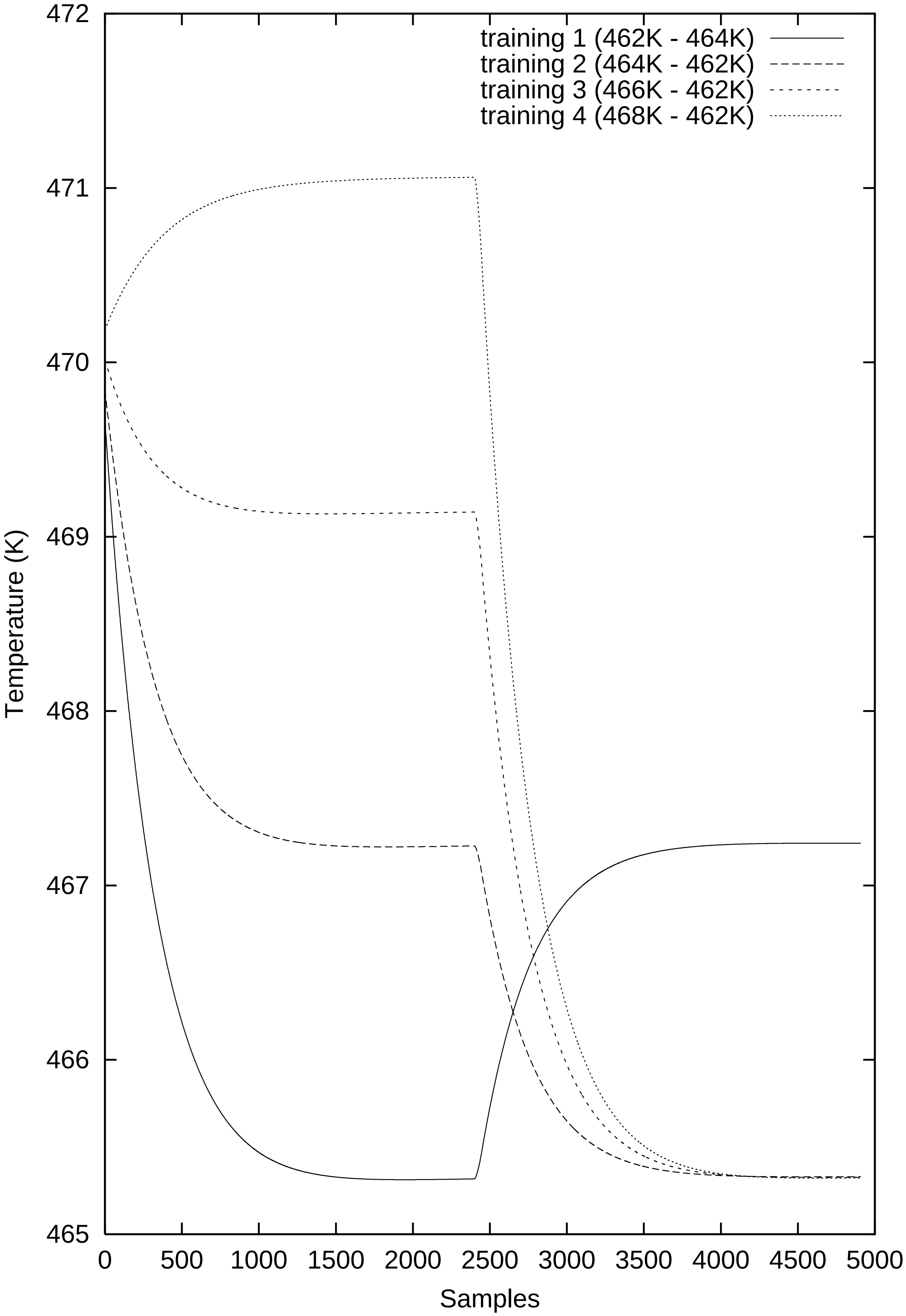

Figure 7: Outlet polymer temperature, used for training, at the nozzle exit for different settings of the barrel heater.

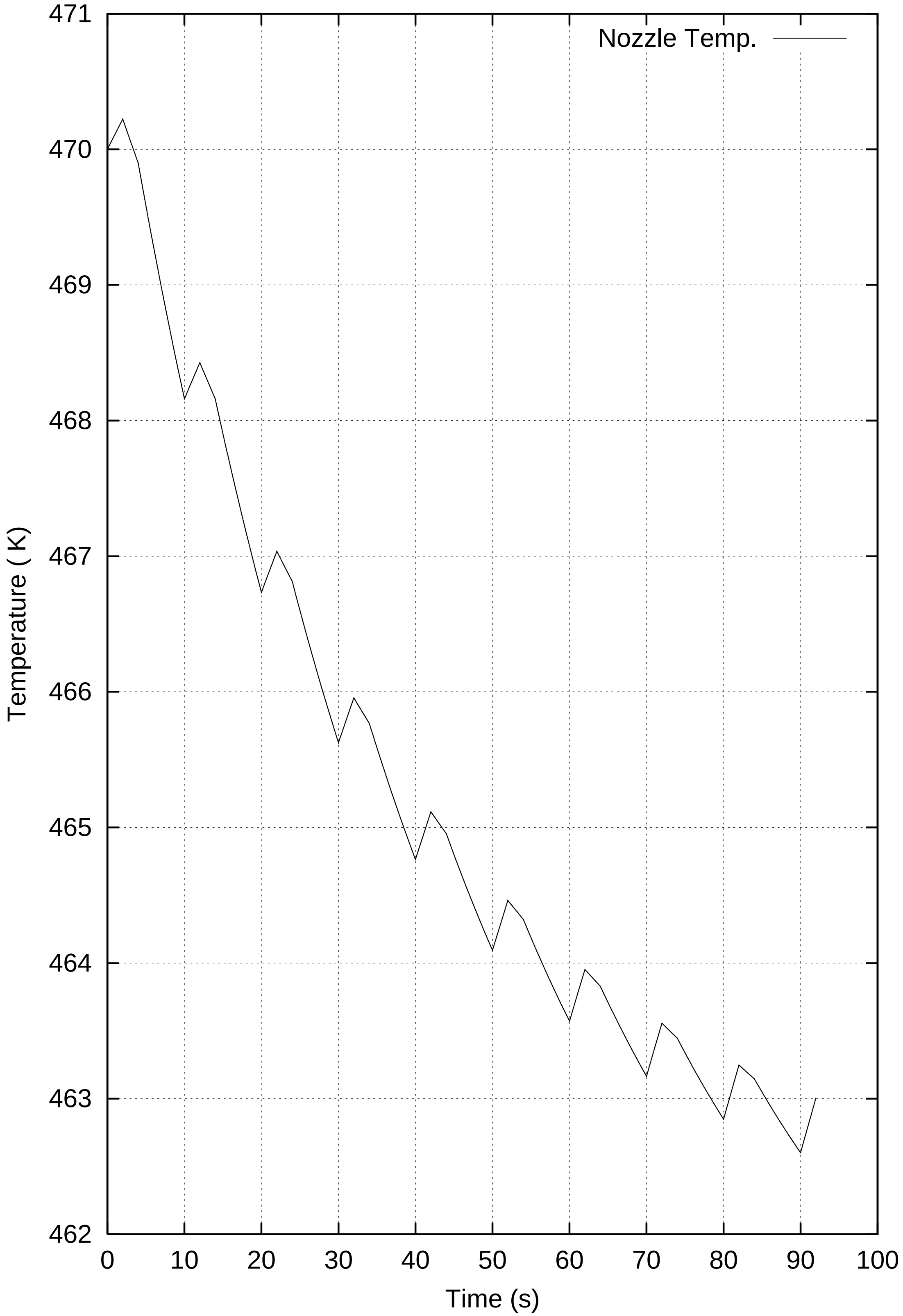

Figure 8: Outlet polymer temperature at the nozzle exit.

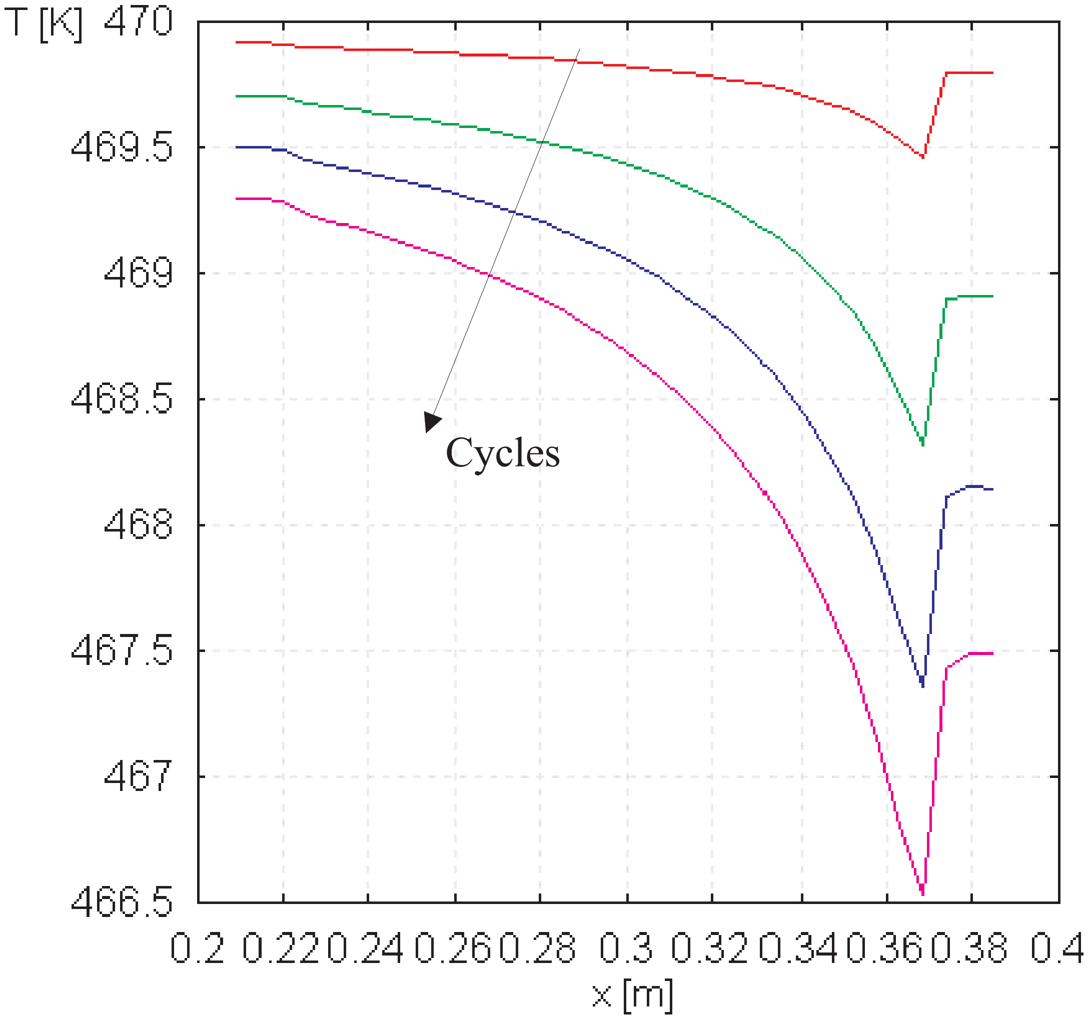

Figure 9: Temperature distribution in the chamber for successive injection cycles of the polymer extrusion temperature (at the chamber inlet) $T_0 = 470K$ and the uniform barrel temperature $T_p = 460K$.

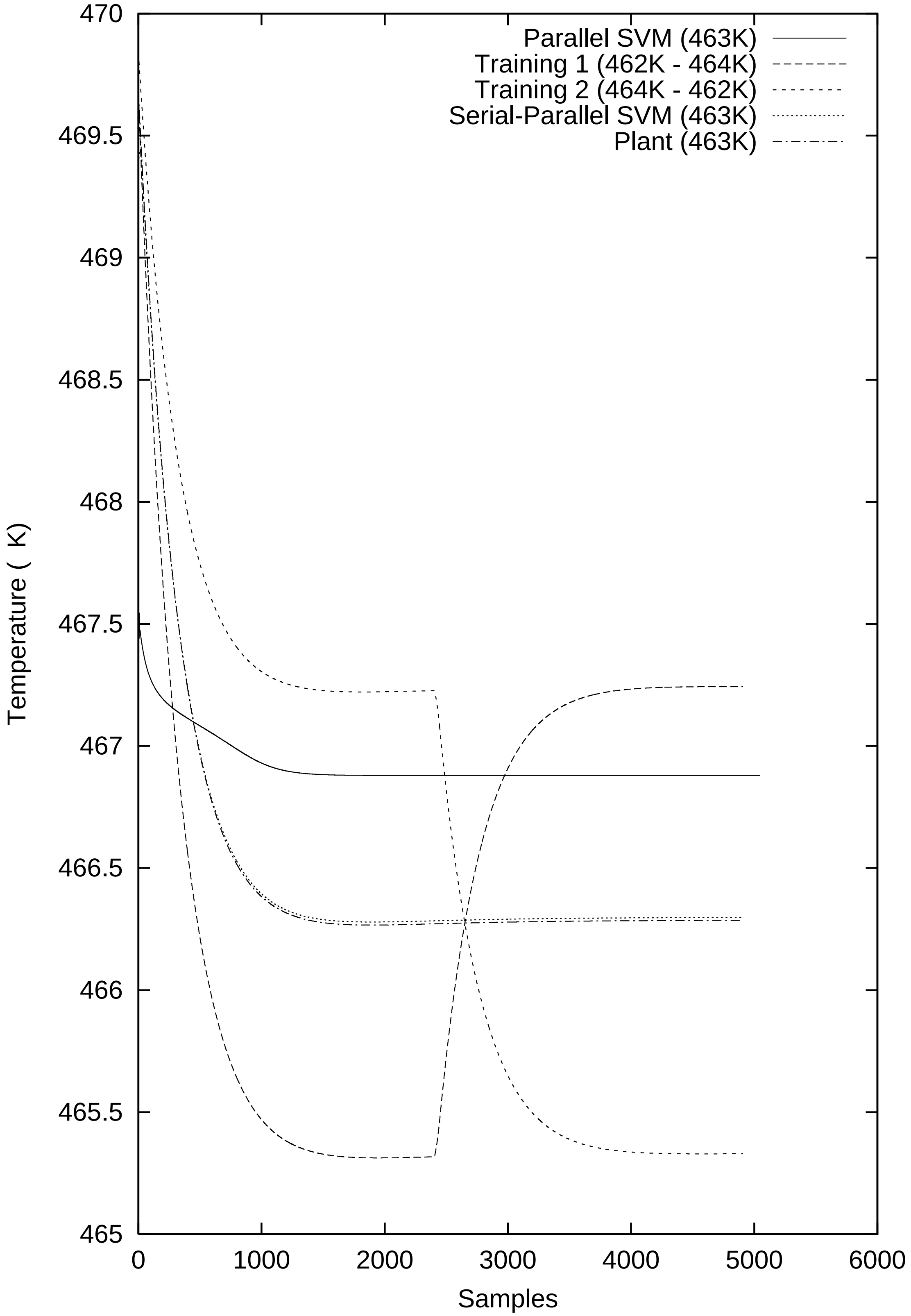

Figure 10: Training profiles (training 1 and training 2) and SVM output for the parallel and serial–parallel models.

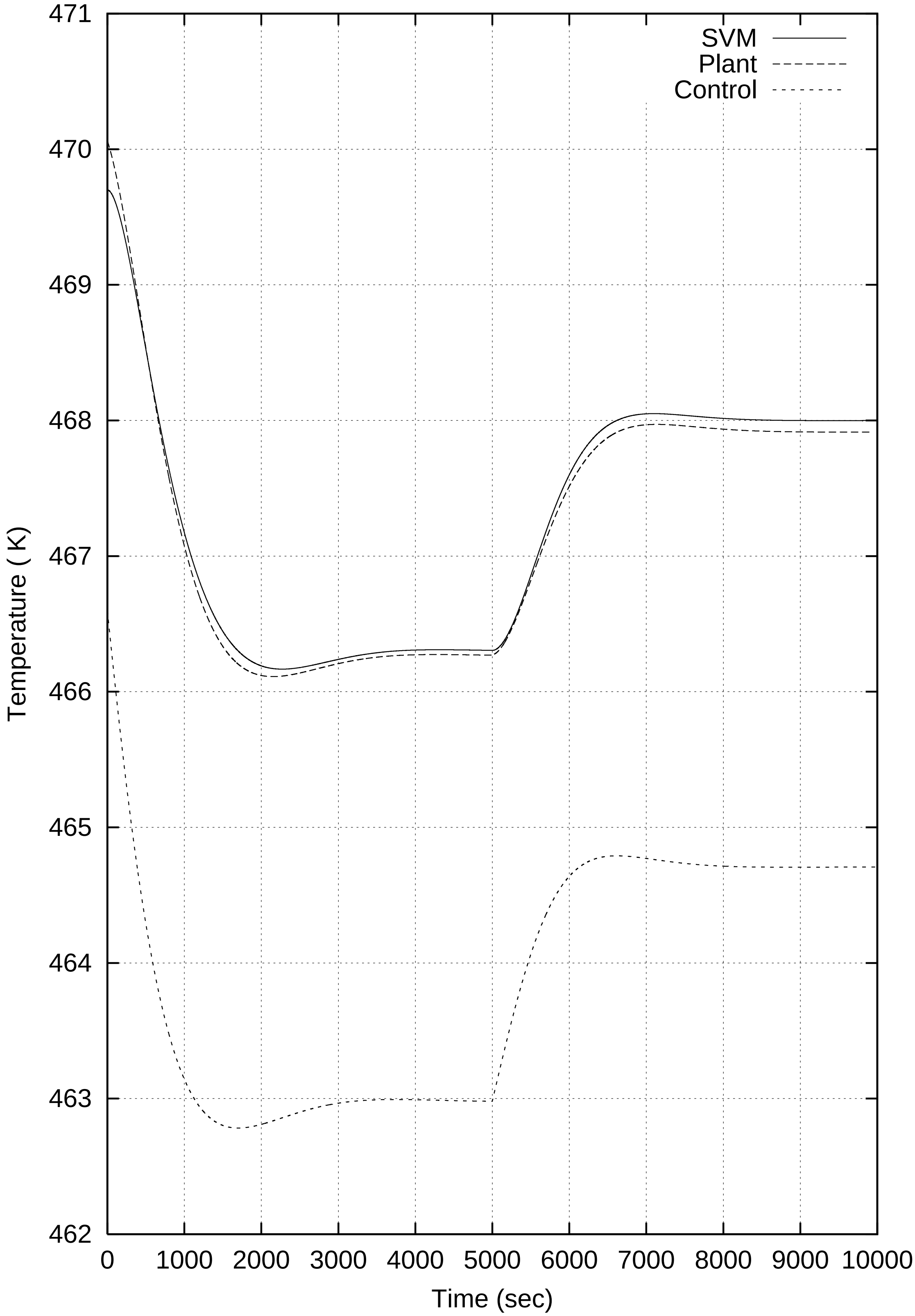

Figure 11: Closed–loop outlet polymer temperature at the nozzle exit (plant) and at the SVM–block exit (SVM) for different reference values.

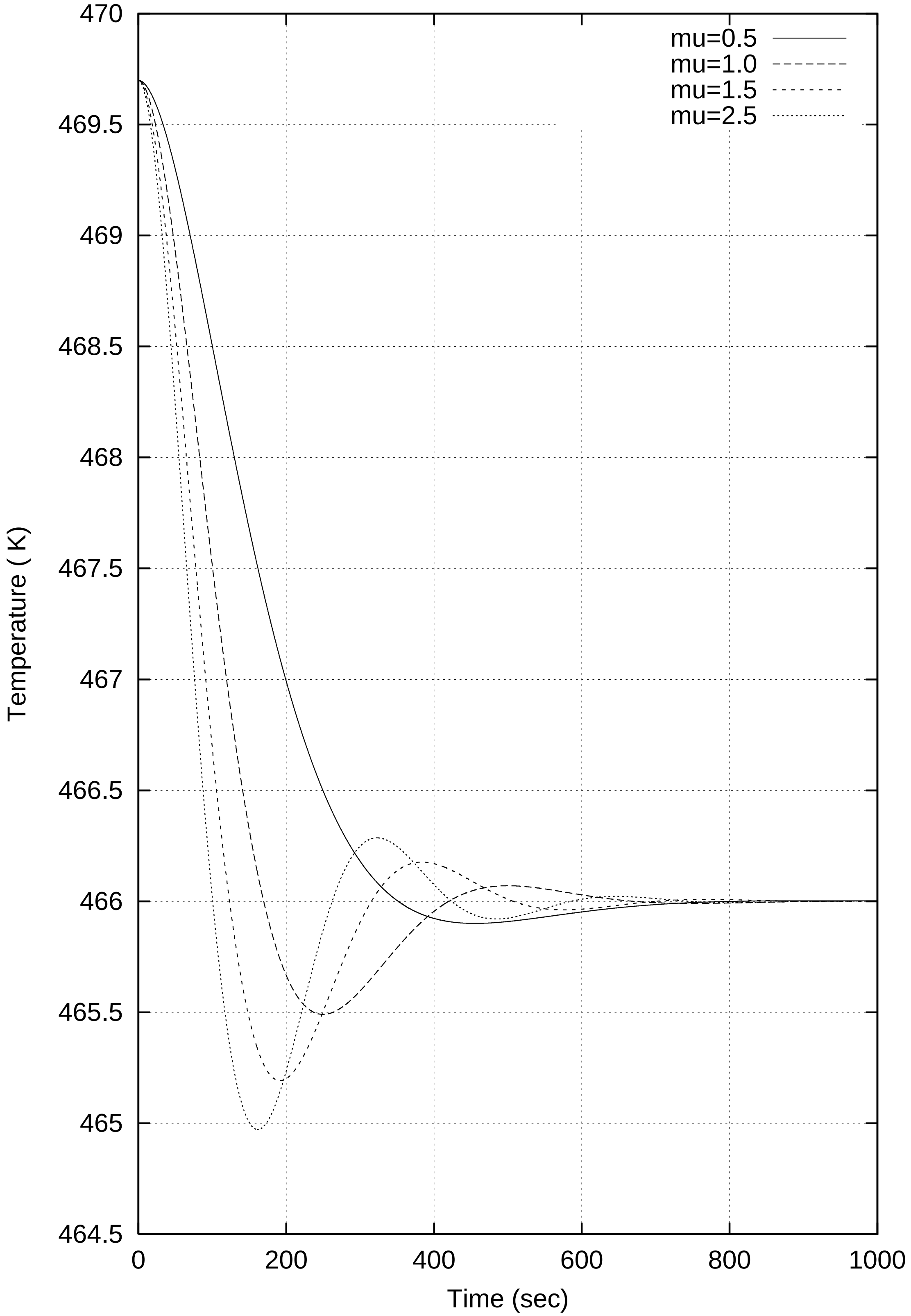

Figure 12: Closed–loop outlet polymer temperature for different values of the descent step parameter μ.

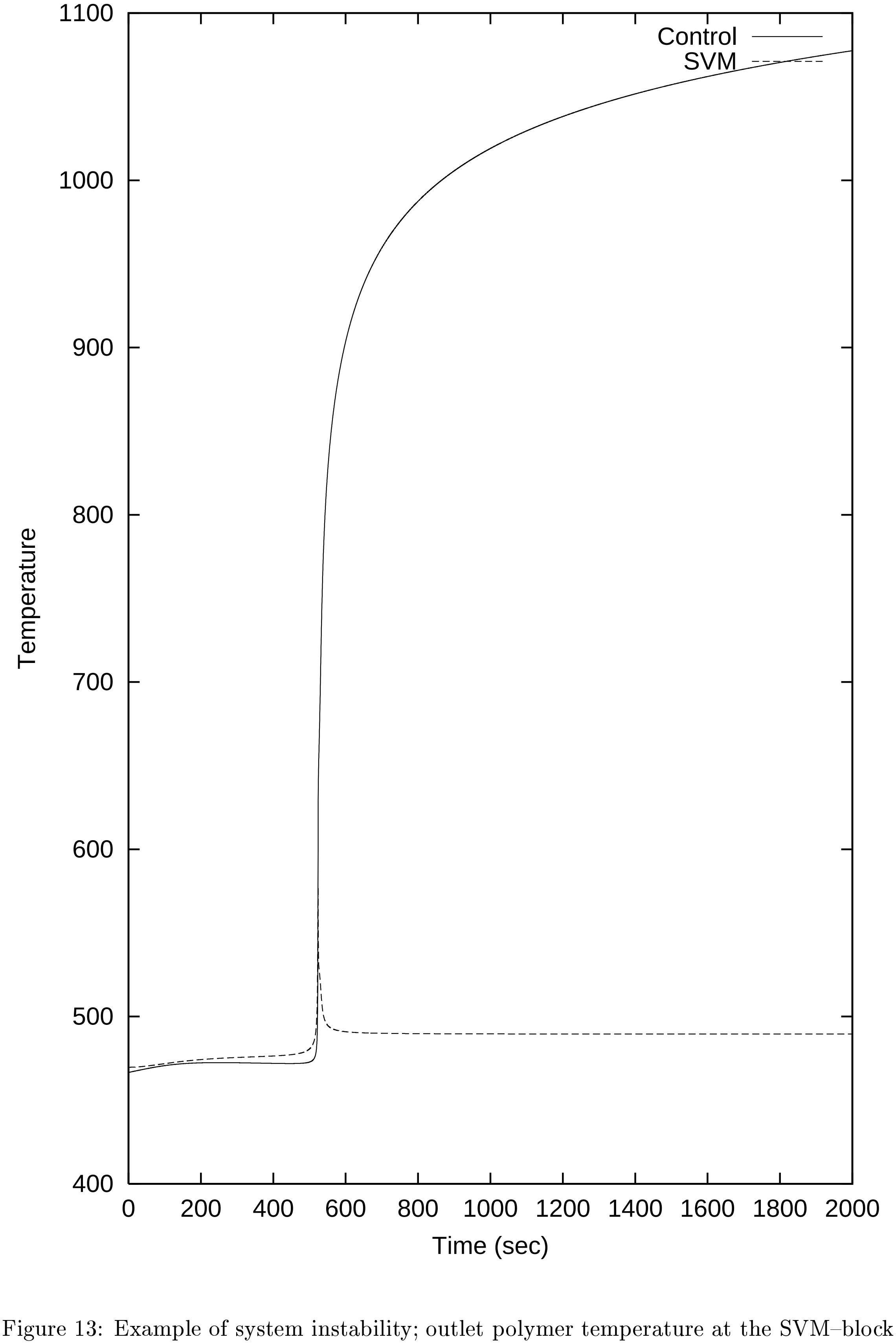

Figure 13: Example of system instability; outlet polymer temperature at the SVM–block exit for a reference value beyond the range.