2009 International Conference on Adaptive and Intelligent Systems - ICAIS'09.
September 24-26, 2009, Klagenfurt - Austria

Intelligent Predictive Control – Application to Scheduled Crystallization Processes

Luis Alberto Paz Suárez[1], Petia Georgieva[2], Sebastião Feyo de Azevedo[1]

[1] Department of Chemical Engineering, Faculty of Engineering, University of Porto, Portugal;
e-mail: lpaz@fe.up.pt; sfeyo@fe.up.pt
[2] Signal Processing Lab, IEETA, University of Aveiro, Portugal, e-mail: petia@ua.pt

***Abstract*—The purpose of this paper is twofold. On the one hand, we propose a modification of the general Model Predictive Control (MPC) approach where a prespecified tracking error is tolerated. The introduction of error tolerance (ET) in the MPC optimization algorithm reduces considerably the average duration of each optimization step and makes the MPC computationally more efficient and attractive for industrial applications. On the other hand a challenging scheduled crystallization process serves as a case study to show the practical relevance of the new intelligent predictive control. Comparative tests with different control policies are performed: i) Classical MPC with analytical or Artificial Neural Network (ANN) process model; ii) ET MPC with analytical or ANN process model; iii) Proportional-Integral (PI) control. Besides the computational benefits of ET MPC, the integration of ANN into the ET MPC brings substantial improvements of the final process performance measures and further relaxes the computational demands.**

Keywords - model predictive control; artificial neural networks; error tolerant optimization; sugar crystallization

I. INTRODUCTION

The model-based predictive control (MPC) introduced in late seventies, nowadays has evolved to a mature level and became an attractive control strategy implemented in a variety of process industries [2]. The main difference between the MPC configurations is the model (linear or nonlinear) used to predict the future behavior of the process or the implemented optimization procedure.

The MPC controllers can cope with process constraints, nonlinear or unstable processes and consider multiple process objectives. Despite these facts, the MPC (particularly the nonlinear case) still remains a challenging control system to design and maintain, so its industrial use is often limited to processes with severe nonlinearities or complicated dynamics. Even for such processes, where the standard control approach (e.g. PID) is not the best solution, the implementation of MPC is impeded due to high computational costs. These are normally related with a heavy optimization procedure or a complex process model that has to be recalled at each optimization step.

How to relax the computational aspects of MPC and make it more attractive for the industry? These academic questions became a real implementation issue when we faced the problem of improving the final product quality of a scheduled sugar crystallization process [6]. The batch or fed-batch mode of operation is a typical production scheme for a large group of pharmaceutical, biotechnological, food and chemical processes. The specificity of such type of processes is related with economic and performance objectives focused at the end of the process [4]. For example, the sugar quality is evaluated by the final average size of sugar particles (termed AM) and the respective variation of this size (CV). The main challenge of the batch production is the large batch to batch variation of AM and CV [6]. This lack of process repeatability is caused mainly by improper control policy and results in product recycling and loss increase.

MPC seems a promising alternative of the traditional proportional-integral-derivative (PID) control that has the potential to overcome the problem of the lack of repeatability. However, on-line execution of MPC with predictions running on a large number of empirical and analytical algebraic differential equations (the process model) make this alternative computationally more involved or even unfeasible for processes with fast nonlinear dynamics (as partially is the crystallization) .

These problems constitute the main motivation for the present work. Our main contribution is the modification of the general MPC approach where a prespecified tracking error is tolerated. The introduction of error tolerance (ET) in the MPC optimization algorithm reduces considerably the average duration of each optimization step. Our second contribution is the integration of the MPC with a data-based process model that only represents the input-output behavior. This black box model is realized by an Artificial Neural Network (ANN). The intuition behind is to build an internally less transparent model but more suitable for fast process output predictions and rapid adaptation. This control paradigm is repeated in the four sequential stages of sugar production and compared with alternative solutions.

The crystallization phenomenon is typical for a great number of industrial processes such as for example in the pharmaceutical and food engineering. Therefore, the ANN-based ET MPC strategy tested successfully on the present industrial case can be further extended and easily applied to other scheduled crystallization based processes.

The reminder of this paper proceeds as follows: In Section II the general MPC problem is stated and the error tolerant (ET) MPC is introduced. In section III the ANN models are discussed. In section IV short description of the

978-0-7695-3827-3/09 $25.00 © 2009 IEEE
DOI 10.1109/ICAIS.2009.34

IEEE computer society

sugar crystallization is given. Control tests and results are discussed in section V. Some future work lines are addressed in section VI.

II. NONLINEAR MODEL PREDICTIVE CONTROL (NLMPC)

A. General MPC Framework

NMPC is an optimization-based multivariable constrained control technique that uses a nonlinear dynamic model for the prediction of the process outputs [1]. At each sampling time the model is updated on the basis of new measurements and state variables estimates. Then the open-loop optimal manipulated variable moves are computed over a finite (predefined) prediction horizon with respect to some performance index, and the manipulated variables for the subsequent prediction horizon are implemented. Then the prediction horizon is shifted or shrunk by usually one sampling time into the future, and the previous steps are repeated. The optimal control problem in the NMPC framework can be mathematically formulated as:

$$\min_{u_{\min} \le u(t) \le u_{\max}} J = \varphi\big(x(t), u(t), P\big), \tag{1}$$

subject to:

$$\dot{x} = f\big(x(t), u(t), P\big),\ 0 \le t \le t_f,\ x(0) = x_0 \tag{2.1}$$

$$y_p(t) = h\big(x(t), P\big) \tag{2.2}$$

$$g_j(x) = 0, \quad j = 1,2,....p \tag{3.1}$$

$$v_j(x) \le 0, \quad j = 1,2,....l \tag{3.2}$$

Where (1) is the performance index, (2) is the process model, function f is the state-space description; function h is the relationship between the outputs and the states, P is the vector of possibly uncertain parameters and t_f is the final time. $x(t) \in \mathbf{X}, u(t) \in \mathbf{Z}, y_p(t) \in \mathbf{Y}$ are the state, the manipulated input and the measured output vectors, respectively. $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{Y}$ are convex and closed subsets R^n , R^m and R^p . g_j and v_j are the equality and inequality constraints with *p* and *l* dimensions respectively.

B. Error-Tolerance MPC - Main Contribution

Considering the usually discrete nature of the online control, the continuous time optimization (1) involved in the MPC is solved by a discrete approximation where the time horizon $t = \lfloor t_0, t_f \rfloor$ is divided into equally spaced time intervals: $t_k = t_0 + k \cdot \Delta t,\ \ k = 0,1,....., N$. The process model (2) is discretised as follows:

$$x(k+1) = f\left[x(k), u(k), P\right] \tag{4.1}$$

$$y_p(k) = h\left[x(k), P\right] \tag{4.2}$$

We propose a modification of the general MPC formulation (1), where the discretized optimization is performed based on the following performance index

$$u(t+k) = \begin{cases} u : \begin{cases} \displaystyle\min_{[u(t+k), u(t+k+1), \dots u(t+H_c)]} J = \\ \displaystyle = \lambda_1 \sum_{k=1}^{H_p} \big(e(t+k)\big)^2 - \lambda_2 \sum_{k=1}^{H_c} \big(\Delta u(t+k)\big)^2 \\ \qquad \text{if } E_\Sigma > \alpha,\ \alpha \in R^+ \end{cases} \\ u^* \qquad \text{if } E_\Sigma \le \alpha \end{cases} \tag{5}$$

where

$$E_\Sigma = \frac{1}{H_p} \sum_{k=1}^{H_p} \big|e(t+k)\big|, e(t+k) = ref(t+k) - y_p(t+k), \\ \Delta u(t+k) = u(t+k-1) - u(t+k-2) \tag{6}$$

subject to the process input constraints

$$u_{\min} \le u(t+k) \le u_{\max}, k = 1,2....H_c \\ \Delta u_{\min} \le \Delta u(t+k) \le \Delta u_{\max} \tag{7}$$

and process output constraints

$$y_{\min} \le y_p(t+k) \le y_{\max}, \quad k = 1,2....H_p \tag{8}$$

$ref(.)$ is the desired response, y_p is the prediction model response. The prediction horizon H_p is the number of time steps over which the prediction errors are minimized and the control horizon H_c is the number of time steps over which the control increments are minimized.

$u(t+k), u(t+k+1), ...u(t+H_c)$ are tentative values of the future limited control signal.

Eq. (5) is a particular discrete form of the general performance index defined by (1). We denote it as an error tolerant (ET) MPC because the optimization is performed only when the error function E_Σ is bigger than a predefined real positive value α. In order to reduce the computational burden when the error is smaller than α the control action is equal to u^* which is the last value of u, computed before the error enters the α strip. Note that E_Σ in (5) is defined as the mean value of the future errors, between the predicted output and its reference along the next H_p steps.

Remark: Before the present formulation of the error function, $E_\Sigma = |ref(t) - y_p(t)|$ (i.e. the absolute value of the current error) was assumed [7]. The choice of E_Σ according to (5) improved the tracking and the batch end point specifications; however, the price to be paid is computational time increase. Therefore, alternative formulations of the error function and the error tolerance can be considered, which we assume is a problem dependent issue and interesting topic for further study.

α is a design parameter and its choice is decisive for achieving a reasonable compromise between lower computational costs and acceptable tracking error. While a formal procedure for its selection is still missing, the error tolerance is chosen based on common sense consideration of 1-5 % error around the set point.

III. ANN Process Model

Over the last 20 years, the Artificial Neural Networks (ANNs) became a well-established methodology not only as a reliable classifier with countless applications but also as dynamical regressor mainly for time-series prediction and identification. In the context of the present work we are mainly interested in studying the ability to project an efficient ANN-based controller for a nonlinear system. This issue received an increasing attention [5,7] with ANNs being applied to design robust neural controllers with guaranteed stability and reference tracking. The neural control problem can be approached in direct or indirect control design framework. Direct ANN control means that the controller has an artificial neural network structure, while in the indirect ANN control scheme, first an ANN is used to model the process to be controlled, and this model is then employed in a more conventional controller design. The implementation of the first approach is simple but the design and the tuning are rather challenging. The indirect design is very flexible, the model is typically trained in advance and the controller is designed on-line, therefore it is the chosen scheme for the present work. The development of suitable ANN training algorithms, as for example the Levenberg-Marquard (LM) algorithm, contributed for the increasing interest to the ANNs in the control community [3]. Though the LM algorithm requires a lot of memory, the trained ANN model is robust against noise and exhibits remarkable generalization properties. The problem with multiple local minima, typical for all derivative based optimization algorithms, is solved by repeating trainings starting with different initial weights.

The most popular ANN structures for modeling reasons are Feedforward Networks (FFNN) and Recurrent (RNN) ones. Due to the memory introduced by delayed inputs the RNN appears to be more suitable for dynamical system modeling and that is why in the present work RNNs were chosen as process models. A linear activation function is located at the single output layer, while tangent sigmoid hyperbolic functions are chosen as processing units in the hidden layer. Though other alternatives can be considered for hidden node functions (for example log-sigmoid function), our choice was determined by the symmetry of the *tansig* output into the interval (-1, 1).

The ANN model was trained with real industrial data. Different regression models were obtained based on data of two, four and six batches. The ANN trained with six batches exhibits the best performance; therefore only results with this model are reported in section 5. In order to extract the underlying nonlinear process dynamics a prepossessing of the initial industrial data was performed. From the complete time series corresponding to the input signal of one stage only the portion that really excites the process output of the same stage is extracted. Hence, long periods of constant (steady-state) behavior are discarded. Since, the steady-state periods for normal operation are usually preceded by transient intervals, the data base constructed consists (in average) of 60-70% of transient period data

IV. Scheduling Process Description

Sugar crystallization occurs through the mechanisms of nucleation, growth and agglomeration. The typical process operation is scheduled and divided into the following sequential phases [3].

<u>Charging:</u> During the first phase the pan is partially filled with a juice containing dissolved sucrose (termed liquor). The charge is usually performed by complete opening of the feeding valve. Therefore, no special control policy is required at this stage.

<u>Concentration:</u> The next phase is the concentration. The liquor is concentrated by evaporation, under vacuum, until the supersaturation reaches a predefined value. At this value seed crystals are introduced into the pan to start the production of crystals. This is the beginning of the crystallization phase.

<u>Crystallization (main phase):</u> At this phase as evaporation takes place further liquor is added to the pan in order to guarantee crystal growth at a controlled supersaturation level and to increase the sugar content of the pan. Near to the end of this phase and for economical reasons, the liquor is replaced by other juice of lower purity (termed syrup).

<u>Tightening:</u> The fourth phase consists of tightening which is principally controlled by evaporation capacity. The pan is filled with a suspension of sugar crystals in heavy syrup, which is dropped into a storage mixer. At the end of the batch, the massecuite undergoes centrifugation, where final refined sugar is separated from (mother) liquor that is recycled to the process.

Sugar production is still a very heuristically operated process, with classical PID controllers being the most typical solution. However, the industrial partners (Sugar Refinery RAR, Portugal, Company 30 de Noviembre, Pinar del Río, Cuba) agreed that an optimized operation policy might result in reduction of the recycled batches and thus in reduction of energy and material loss. These problems motivated the selection of the sugar crystallization as the case study in order to test the computational efficiency of the new ANN-ETMPC control.

The different phases of the sugar production are comparatively independent, thus a single controller can hardly be effective for the complete process. Instead, individual controllers for each stage where active control is required, was the adopted framework. See Table I for more details on the operation strategy.

V. Experimental Results

A. Special case of AM Reference in the 4th loop

The traditional practice of sugar production is to build a sequence of control loops where a measurable variable has to be kept constant over a certain period. This intuitive strategy leads to simple error-correction based control system easy to implement and maintain. However, the controlled variable is often not the one that directly determines the process performance but is the one that is possible to measure. Meanwhile, recent advances in software sensor research demonstrated that it is more plausible to develop a strategy to estimate the unmeasurable variable that directly determine the process performance and take control corrections based on these estimations. Therefore we extend our study by formulating a reference for the mean crystal size AM over the last process stage. AM is the main sugar production performance measure and since it is not directly measurable we estimate it. Laboratory samples taken over the 4th control loop show that the crystal growth follows an exponential curve similar to the one depicted on Fig.1.

At each sampling time k the AM reference is on-line recomputed according to the following empirical expression

$$AM_{ref} = (1-\gamma)AM_{end\_ref} + \gamma \cdot AM_{ref}(k-1) \qquad (9)$$

where AM_{end_ref} is the reference for the crystal size at the batch end. γ belongs to the interval [0,1] and determines how smooth is the curve on Fig. 1. For the present tests $\gamma = 0.9$ was selected.

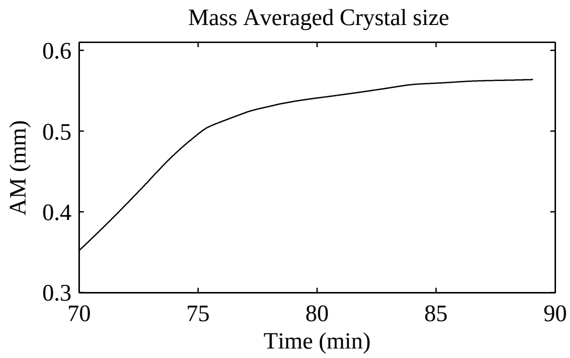

Figure 1. Crystal growth over the last crystallization stage (4th control loop).

B. Discussion of results

The operation strategy, summarized in Table I and implemented by a sequence of ET MPC, classical MPC or PI controllers is tested. The output predictions are provided either by a simplified discrete model (with the main operation parameters kept constant) or by a trained ANN model. A process simulator was developed based on a detailed phenomenological model [3]. Realistic disturbances and noise are introduced substituting the analytical expressions for the vacuum pressure, brix and temperature of the feed flow, pressure and temperature of the steam with original industrial data (without any preprocessing). The choice of the MPC design parameters for each control loop are summarized in Table II. α-strip values were chosen as about 1% error around the specified set points. The set-points are chosen empirically based on the process operator experience. See [7] for more details on the choice of the MPC parameters.

On Figs. 2-4 are summarized the results with respect to the CPU time required by the MPC optimization at each iteration. First the classical and the ET MPC are compared (Fig. 2) integrating a discredited analytical model (Georgieva et al., 2003). Next on Fig. 3 the classical and the ET MPC are compared with ANNs as predictive models. On Fig. 4 are depicted the CPU time results of ANN-ET-MPC and ANN-MPC for the special case of variable reference (9) in the 4th loop. The results demonstrated that the ET MPC reduces significantly the optimization time, which is further relaxed if the controller integrates an ANN (inputo-utput) predictive model. Due to the small tracking error tolerated the ET MPC leads naturally to worse set point tracking compared with the classical MPC and PI control. However, the end point process characteristics, that really matter, are improved (see Table III). Particularly, when in the last loop the policy is not a set point tracking of the crystal fraction but a variable reference for AM, there is a tendency of getting final crystals with better distribution and size.

VI. Future Work

Research on the computational efforts-tracking tolerance compromise in the choice of parameter α is now in progress. An open question is also the implementation of ET MPC for the most challenging (from computational point of view) MPC scheme, namely the batch MPC, where the performance index explicitly accounts for the final specifications and therefore the prediction horizon is equal to the batch duration or at least to the batch stage. Finally, the choice of the error tolerance function is also an interesting issue.

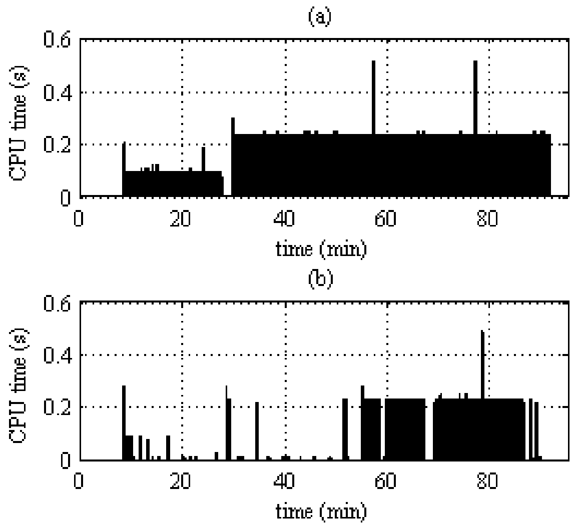

Figure 2. CPU time per iteration along the process duration. (a)-Error Tolerant (ET) MPC; (b) Classical MPC

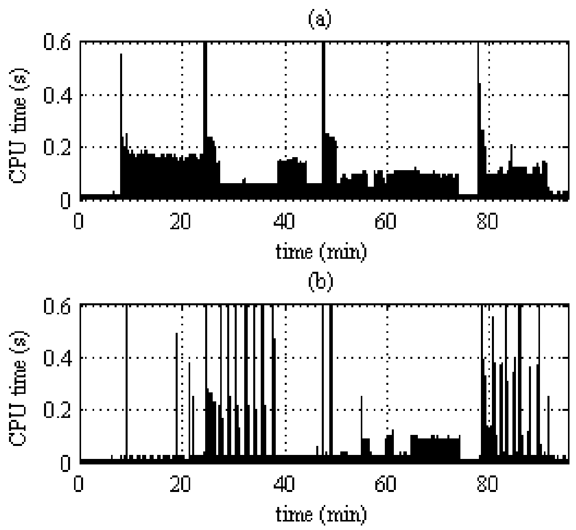

Figure 3. CPU time per iteration along the process duration. (a) ANN-MPC (b) ANN ET MPC

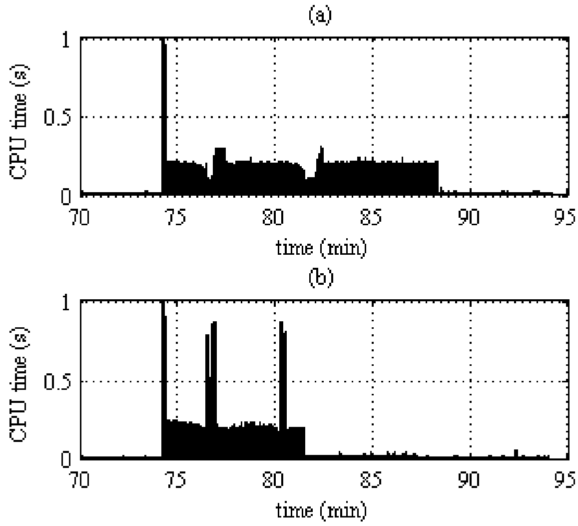

Figure 4. CPU time per iteration along the 4th control loop. (a) ANN-MPC (b) ANN- ET MPC

ACKNOWLEDGMENT

Several institutions contributed for this study: 1) The Institute of Electronic Engineering and Telematics of Aveiro (IEETA); 2) Laboratory for Process, Environmental and Energy Engineering (LEPAE), Department of Chemical Engineering, University of Porto; 3) Foundation of Science and Technology of Portugal, which financed the scholarship of investigation of doctorate SFR/16175/2004; 4) Sugar refinery RAR, Portugal; 5) Company 30 de Noviembre, Pinar del Río, Cuba; The authors are thankful to all of them.

REFERENCE S

[1] Allgöwer, F., Findeisen, R. & Nagy, Z. K. (2004). Nonlinear model predicitve control: From theory to application. Journal of Chinese Institute of Chemical Engineers, 35 (3), 299-315.

[2] Camacho, E. F., Bordons, C. (2004). Model predictive control in the process industry. London: Springer-Verlag.

[3] Georgieva, P., Meireles, M. J. & Feyo de Azevedo, S. (2003). Knowledge Based Hybrid Modeling of a Batch Crystallization When Accounting for Nucleation, Growth and Agglomeration Phenomena. Chemical Engineering Science, 58, 3699-3707.

[4] Nagy, Z. K., Braatz, R. D. (2003). Robust nonlinear model predictive control of batch processes. AIChE J, 49 (7), 1776-1786.

[5] Norgaard, M., Ravn, O., Poulsen, N. K. & Hansen, L. K. (2000). Neural networks for modelling and control of dynamic systems. London: Springer-Verlag.

[6] Simoglou, A., Georgieva, P., Martin, E. B., Morris, J. & Feyo de Azevedo, S. (2005). On-line Monitoring of a Sugar Crystallization Process. Computers & Chemical Engineering, 29 (6), 1411-1422.

[7] Suárez, L. A. P., Georgieva, P. & Feyo de Azevedo, S. (2009). Computationally efficient process control with neural network-based predictive models. International Joint Conference on Neural Networks (IJCNN), Atlanta, Georgia, USA (accepted).

TABLE I. SUMMARY OF THE SUGAR CRYSTALLIZATION OPERATION STRATEGY

Stage	Actions	Control
Charge	The steam valve is closed. The stirrer is off. The vacuum pressure changes from 1 to 0.23 bar. The vacuum pressure reaches 0.5 bar, feeding starts with max rate. Liquor covers 40 % of the vessel height.	No control. The feed valve is completely open
Concen tration	The vacuum pressure stabilizes around 0.23 bar.The stirrer is on. The volume is kept constant. The steam flowrate increases to 2 kg/s . The supersaturation reaches 1.06, the feeding is closed, the steam flowrate is reduced to 1.4 kg/s	*Control loop 1*. Controlled variable: Volume. Manipulated variable: liquor feed flowrate
Seeding and setting the grain	The supersaturation reaches 1.11. Seed crystals are introduced. The steam flowrate is kept at the minimum for two minutes.	No control. The feed valve is closed
Crystallization with liquor (phase 1)	The steam flowrate is kept around 1.4 kg/s. The supersaturation is controlled at the set point 1.15.	*Control loop 2* Controlled variable: supersaturation Manipulated variable: liquor feed flowrate
Crystallization with liquor (phase 2)	The volume of crystallizer reaches _ 22 m3. The feed valve is closed. The supersaturation is controlled at the set point 1.15. The stirrer power reaches 20.5 A	*Control loop 3* Controlled variable: supersaturation Manipulated variable: steam flowrate
Crystallization with syrup	The steam flowrate is kept around the maximum of 2.75 kg/s. (hard constraint). The volume fraction of crystals is kept at the set point 0.45. The volume reaches its maximum value (30 m3). Feed valve closed.	*Control loop 4* Controlled variable: volume fraction of crystals. Manipulated variable: syrup feed flowrate
Tightening	The stirrer power reaches the maximum value of 50 A (hard constraint). The steam valve is closed. The stirrer and the barometric condenser are stopped.	No control

TABLE II. MPC DESING PARAMETERS FOR THE CONTROL LOOPS DEFINED IN TABLE I

Control loop (CL	t_s **(s) Settling time**	Δt **(s) Sampling period**	H_p **Prediction horizon**	H_c **Control horizon**	λ_2	**Controlled variable**	**Set-point**	α **-strip (1%)**
CL1	40	4	10	2	1000	Volume	12.15	0.15
CL2	40	4	10	2	0.1	Supersaturation	1.15	0.01
CL3	60	4	15	2	0.01	Supersaturation	1.15	0.01
CL4	80	4	20	2	10000	Fraction of crystals	0.43	0.004

TABLE III. BATCH END POINT PERFORMANCE MEASURES

a) Batch 1

Performance measures	**Constant set point references in all control loops**					**Variable reference in the 4th loop**		
	Classical MPC	***ET-MPC***	***ANN-MPC***	***ET ANN-MPC***	***PI***	***ANN-MPC***	***ET ANN-MPC***	***PI***
AM (mm) (reference 0.56)	0.586	0.588	0.584	0.583	0.590	0.559	0.550	0.589
CV (%)	32.17	31.39	31.13	31.26	32.96	30.14	30.15	30.24
Average CPU time (s)	0.166	0.074	0.091	0.061	-------	0.203	0.151	-----

a) Batch 2

Performance measures	**Constant set point references in all control loops**					**Variable reference in the 4th loop**		
	Classical MPC	***ET-MPC***	***ANN-MPC***	***ET ANN-MPC***	***PI***	***ANN-MPC***	***ET ANN-MPC***	***PI***
AM (mm) (reference 0.56)	0.615	0.603	0.609	0.605	0.613	0.573	0.550	0.611
CV (%)	29.36	30.26	30.28	30.42	31.14	29.34	30.15	30.83
Average CPU time (s)	0.166	0.088	0.102	0.078	-------	0.214	0.144	-----

a) Batch 3

Performance measures	**Constant set point references in all control loops**					**Variable reference in the 4th loop**		
	Classical MPC	***ET-MPC***	***ANN-MPC***	***ET ANN-MPC***	***PI***	***ANN-MPC***	***ET ANN-MPC***	***PI***
AM (mm) (reference 0.56)	0.636	0.625	0.631	0.625	0.626	0.581	0.578	0.622
CV (%)	28.74	29.85	29.42	30.76	29.23	28.39	28.67	28.64
Average CPU time (s)	0.167	0.103	0.096	0.067	-------	0.209	0.140	-----