

# On batch process control using feature extraction with application to a gravimetric blender

Simone Formentin<sup>1</sup>, Alberto L. Cogni<sup>1</sup>, Fabio Previdi<sup>1</sup>, Sergio M. Savaresi<sup>2</sup>

**Abstract**—In this paper, a data-driven control approach for batch processes based on feature extraction is proposed. In this strategy, a model of the system dynamics is not needed and the issue of adapting to different operating conditions is recast into a simple controller design problem. Throughout the paper, the proposed algorithm will be described in detail with the help of an experimental gravimetric blender example.

## I. INTRODUCTION

According to the definition of the Instrument Society of America (ISA), a batch process is defined as “a process that leads to the production of finite quantities of material by subjecting quantities of input materials to an ordered set of processing activities over a finite period of time using one or more pieces of equipment” (ISA, 1995).

This definition reflects most of the processes included in the modern industry and, therefore, control of such processes may critically affect the performance of a company within the market.

Unlike continuous systems, control of batch processes can exploit the fact that operations are repeated along the batches. This observation led to the development of different *ad-hoc* strategies, most of them based on a model of the process to control. However, as recalled in [4], “although model-based solutions are available, process models in the batch area tend to be poor”, especially in modern factories manufacturing products of high complexity.

It follows that, along with the increase of computational power of electronic control units, numerical and optimization methods based on measurements, instead of physical modeling, have become more and more interesting from an industrial (and, subsequently, scientific) perspective. Among them, iterative learning control and repetitive control [1], [2] have been the basic components for the development of a complete framework, where the control objective is achieved by adaptation along standard operating conditions and only few priors about the process are used.

In this manuscript, a different approach to the above problem will be presented. The main assumption behind it is that, for each batch, input and output trajectories can be compressed into a finite set of features, which describe the variations of the time-series with respect to a pre-determined pattern (or set of patterns). The advantages of the use of features instead of time-trajectories can be assessed by

looking at Fig. 1, which visually shows the above rationale. By moving from time-domain ( $t$ ) to batch-domain ( $T$ ), the control problem is dramatically simplified, as it can be reformulated as a classical tracking problem of a multivariable system, where  $t$  is replaced by  $T$  and, as far as a stabilizing controller in the batch-domain is designed, the batch process surely converges to the desired operating conditions.

In this paper, a Principal Component Analysis will be employed to compress input and output trajectories into suitable features from a large dataset. To make the whole procedure only depending on measurements, the controller will be designed based on batch-domain data without deriving a model between input features and output features. Specifically, the data-driven design procedure in [7] will be adopted. This choice will make the overall design procedure simple and reliable against uncertainty.

The above procedure will be illustrated throughout the paper with the help of an experimental case study concerning a gravimetric blender, a significant element in the plastic industry [15] [16].

A control algorithm for such a device should have two main objectives: to ensure a dosage as close as possible to the defined ratios and to guarantee a low adaptation time after a change in the recipe, since all the material produced with a wrong recipe (during the transient) is plastic waste. For this reason, a reduction of the number of adaptation cycles yields a significant reduction of costs and time.

Control of such systems has seldom been investigated in the scientific literature. Although a few contributions can be found on control of continuous blenders, *e.g.* [13], as far as the authors are aware, no studies have been presented on the batch architecture. It follows that, so far, the state-of-the-art controllers are the empirically tuned rules embedded in the off-the-shelf products. Hence, the case study presented in this paper also represents a novel way to deal with this important control problem in the plastic industry.

The remainder of the paper is as follows. In *Section II*, the PCA-based approach for the batch system description and control is introduced and described qualitatively. The gravimetric blender case study is illustrated in *Section III*, where the technical features of the proposed strategy are discussed in more detail. Some concluding remarks end the paper.

## II. THE METHODOLOGY

Let the nonlinear dynamical process described, for each batch, by

$$f(y(t), \dot{y}(t), \dots, u(t), \dot{u}(t), \dots) = 0, \quad (1)$$

<sup>1</sup>Dipartimento di Ingegneria, Università degli Studi di Bergamo, Viale Marconi 5, 24044 Dalmine (BG), Italy. Corresponding author: simone.formentin@unibg.it

<sup>2</sup>Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

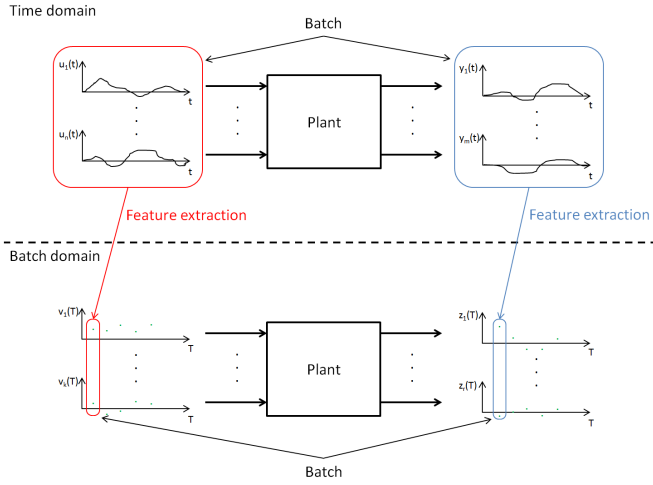


Fig. 1. From time-domain to batch-domain using data.

where  $y(t) \in \mathcal{R}^n$ ,  $u(t) \in \mathcal{R}^m$  are continuous functions of time and  $f$  depends on  $u$ ,  $y$  and their derivatives. Moreover, let the time-length of the batch  $N$  be fixed.

Since (1) is a batch process, the system will “move” from one batch to the other, depending on the user requirements, over a variety of operating conditions corresponding to a related variety of input/output (I/O) trajectories of the same length  $N$ .

However, in batch processes, the operating conditions are often similar to each other, since in most of the cases only some features (e.g. the amplitude) of the I/O signal changes, depending on some external parameters, like, e.g. the temperature or the reference behaviour. In these cases, it is intuitive that a pattern underlying the set of trajectories can be highlighted and one can think to characterize the input signal as a variation with respect to that pattern, instead of a simple time-series.

The advantage of such an approach is that few numbers for each batch (or *features*, from now on) may be sufficient to fully describe  $u(t)$  or  $y(t)$ , regardless of how complicated the system dynamics is or how rich the frequency content of the signals is.

Fig. 1 illustrates the above concept. Once the I/O patterns have been defined, the new representation of the batch process becomes the relationship between the I/O features (namely,  $v(T)$  and  $z(T)$ ) in the domain of “batches”, thus substituting Equation (1) linking the trajectories in the  $t$  domain (namely  $u(t)$  and  $y(t)$ ). Notice that such a relationship may be simple even in case of very complex I/O dynamics in time-domain.

The compression of  $u(t)$  and  $y(t)$ , for  $t = 1, \dots, N$  into  $v(T)$  and  $z(T)$  for a given  $T$  can be performed by collecting a large number of experiments (i.e. batch data) over the whole operating region. Then, Principal Component Analysis (PCA) can be applied to easily obtain such a transformation.

PCA, also known as *Karhunen-Loève transform* (see [9]), is a method widely used in dimensionality reduction and feature extraction. PCA projects the data onto a lower

dimensional subspace, such that the mean squared distance between the data points and their projection is minimized.

Consider for instance the matrix of output trajectories  $Y$  built using the profiles characterizing  $r$  different batches of the same duration.  $Y$  can be rewritten as

$$Y = A\Sigma B = \sum_{i=1}^r \alpha_i a_i b_i^T,$$

where  $a_i$  and  $b_i$  are suited orthogonal vectors,  $A$  and  $B$  are matrices composed by joining  $a_i$  and  $b_i$ , respectively, and  $\Sigma$  is a diagonal matrix composed by the  $\alpha_i$  terms,  $i = 1, \dots, r$ . The singular values  $\alpha_i$ 's give an idea of the relevance of each component in  $Y$  and the singular value decomposition (SVD) approach guarantees that the versors are sorted according to their relevance. It is then possible to reconstruct  $Y$  with  $m < r$  terms, with accuracy bounded by

$$\varepsilon_j = \sum_{i=m+1}^r \alpha_i^2.$$

Those  $m$  terms represent the elements of the vector  $z(T)$  and their numerical values may vary for each batch. The same philosophy can obviously be applied to the input trajectory.

The most important point in such an approach is that, if also the reference output trajectory is given in terms of variations with respect to the output pattern, *the issue of making the system to adapt to any change in the reference signal can be recast into a tracking control problem in the batch-domain*.

In this paper, the problem of designing a controller to follow a given reference trajectory  $z^o(T)$  in the batch-domain will be dealt with without resorting to a model of the relationship between  $v(T)$  and  $z(T)$ , whose structure would be hard to guess. More specifically, the data-driven method in [7] will be employed.

In [7], a linear time-invariant desired behavior  $M$  for the closed-loop system and a controller structure  $K(\rho)$ , parameterized with  $\rho$ , are supposed to be given by the user. In particular, in this paper, the PID controller

$$v(T) = v(T-1) + \sum_{k=0}^2 B_k e_z(T-k), \quad (2)$$

will be employed, where  $e_z(T) = z^o(T) - z(T)$  and  $B_k \in \mathcal{R}^{2 \times 2}$  are matrices containing the unknown parameters, such that  $\rho$  is defined as

$$\rho = [\text{vec}^T(B_0) \dots \text{vec}^T(B_2)]^T. \quad (3)$$

The regression form of the controller is then

$$\begin{aligned} v(T) &= v(T-1) + B_0 e_z(T) + \dots + B_n e_z(T-n) \\ &= v(T-1) + [e_z^T(T) \otimes I \dots e_z^T(T-n) \otimes I] \rho \\ &= v(T-1) + \varphi^T(T) \rho, \end{aligned}$$

where the last equality defines  $\varphi(t)$  and  $\otimes$  denotes the Kronecker matrix product. The above form clearly shows the linearity in the parameters of the so-built multivariable PID.

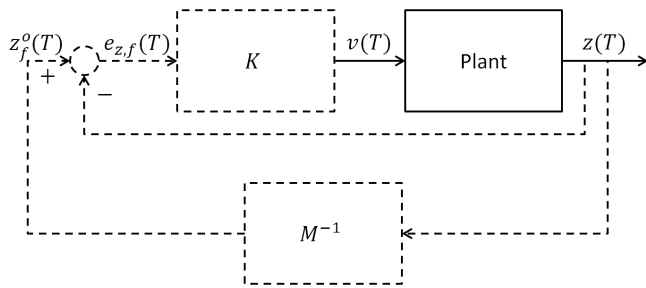


Fig. 2. Tuning scheme of the data-driven control approach in [7].

The main idea of method [7] for control design is very simple. The basis observation is that, if the available  $r$  input and output batches collected in the open-loop experiments used for PCA were instead generated within the “ideal” closed-loop system  $M$ , the closed-loop complementary sensitivity function in the batch-domain would be exactly equal to  $M$ . Furthermore, the reference signal, referred to as “fictitious reference” signal from now on, could be computed following Fig. 2 as

$$z_f^o(T) = M^{-1}z(T),$$

where  $M^{-1}$  denotes the inverse of  $M$ . The corresponding “fictitious error” signal is then  $e_{z,f}(T) = z_f^o(T) - z(T)$ . It is easy to argue that the ideal controller is the one that generates  $v(T)$  when fed by  $e_{z,f}(T)$ . Following this rationale, the control design issue turns out to be a simple identification problem, where the optimal controller is the one that best approximates the ideal one in the given PID class. Practically, in order to get the optimal PID, the cost function

$$J_{VR}^r(\rho) = \frac{1}{r} \sum_{T=1}^r \|v(T) - K(\rho)e_{z,f}(T)\|_2^2 \quad (4)$$

is minimized with respect to  $\rho$  using batch-domain data. Notice that, with the above parameterization, the optimization procedure is convex and the global minimum is guaranteed to be reached. In [7], it has also been proven that, provided that  $v(T)$  and  $z(T)$  are suitably prefiltered by  $L = M$ , the desired closed-loop behaviour can be achieved. See [7], [8] for more details.

The procedure can then be (qualitatively) summarized as in the box at the end of the section.

Notice that the proposed approach allows to feed the closed-loop system with any reference time-trajectory (unlike many ILC strategies) and to adapt to different operating conditions. On the other hand, a large dataset exploring the whole operating space is required. It should also be remarked that such a method is meaningful (and works) only under the assumption that a common pattern over all the I/O time-trajectories exists.

In the next section, a case study on a batch gravimetric blender for the plastic industry will be presented in detail, to better delineate the technical features of the proposed strategy.

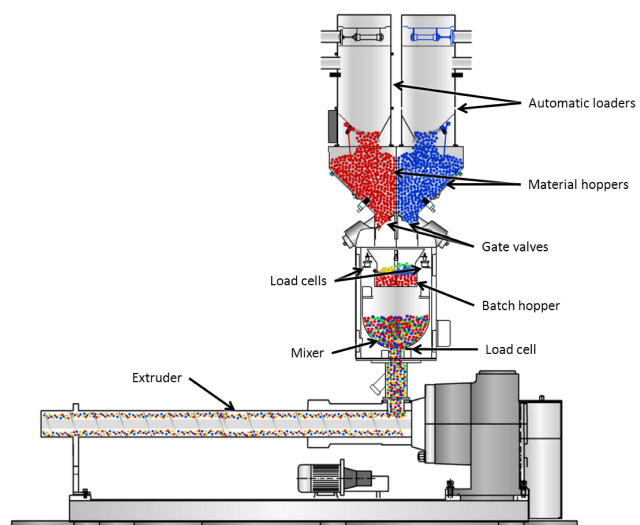


Fig. 3. Schematics of the employed experimental setup with 4 gate valves.

#### Batch-domain control strategy

- A. **Data collection:** perform a set of experiments scattered over the whole operating range, so as to have some (partial) information on the possible I/O trajectories.
- B. **PCA decomposition:** perform a PCA on the matrix of input data to derive the input pattern and a PCA on the matrix of output data to derive the output pattern.
- C. **PCA validation:** project the available I/O trajectories on the respective patterns to derive  $v(T)$  and  $z(T)$ . Then, use the finite dimensional  $v(T)$  and  $z(T)$  to reconstruct the original I/O trajectories. The reconstruction error gives a measure of the approximation quality. In case of a bad approximation, the number of singular values needed to reconstruct the trajectories must be increased.
- D. **Data-driven controller design:** take the batch-domain trajectories  $v(T)$  and  $z(T)$ . These signals represent the input and the output of a static system, since each batch is independent from the previous one. Define a reference behavior for the closed-loop system and use the data to design a MIMO PID controller using the method in [7].

### III. THE GRAVIMETRIC BLENDER CASE STUDY

The blender is an important element in the plastic extrusion process. Usually, by means of gravimetric or volumetric blenders, from two to six different polymeric components can be blended in the feeding section of the extruder in form of granulate, pellets or irregular small bits. Then, the polymer is transported along the barrel by means of a rotating screw. During the process, the polymer undergoes very complex thermo-mechanical transformations inducing strong changes in the physical properties of the material [14], [17]. The final

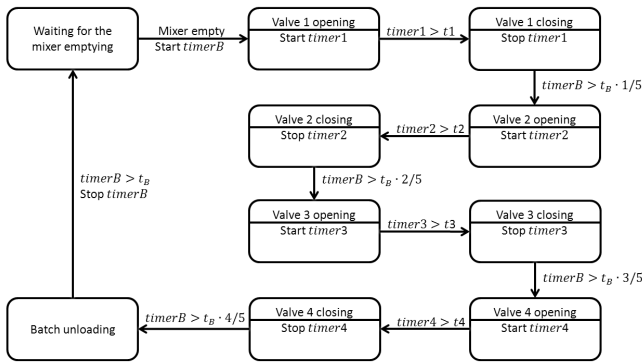


Fig. 4. Working cycle.

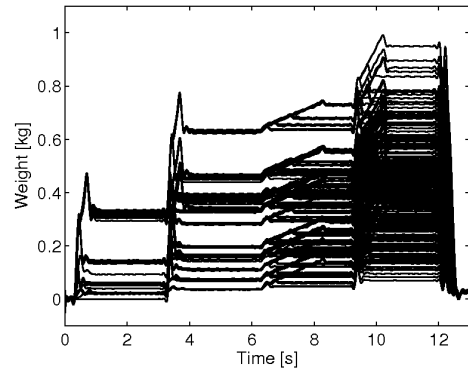


Fig. 5. Spread of working cycles.

product quality in extrusion is essentially characterized by a precisely-regulated volumetric flow of the polymer through the extruder. This can be achieved by fine regulation of the mass delivered from the blender to the extruder and by exact distribution of the different materials.

Closed-loop control of gravimetric blending provides many advantages with respect to volumetric blenders, *e.g.* metering is independent of material density variations, no frequent calibrations are necessary; the increased accuracy considerably reduces the incidence of raw material costs. More specifically, a *batch* gravimetric blenders sequentially doses each material, by weight, into a common dynamic mixer assembly. Every batch is captured and mixed by the mixer before being released to the extruder. The mechanical mixing of a batch blender works well for materials that are of similar particle shape, size, and density.

The structure of the plant considered for the case study, whose schematics is illustrated in Fig. 3, is characterized of three main parts: a set of 4 gate valves necessary to dispense the material, a batch hopper that collect the material outgoing from the gate valves and a mixer devoted to make the components uniform in a homogeneous mixture. The operating principle is very simple: each gate valve releases the material to the weighted batch hopper with a predefined order (from component 1 to component 4); at the end of the last metering the batch hopper is unloaded and the material falls into the mixer. Gate valves and batch hopper are controlled by ON/OFF actuators: from the ECU (Electronic Control Unit) is possible to define, for each  $T_s = 4 \text{ ms}$  sampling period, the state of the actuators.

The typical working cycle is defined as a time scheduling of basic operations (see Fig. 4). In the first part, every meter releases the material into the batch hopper; the gate valves opening is executed on a regular basis every  $2.4 \text{ s}$ . The last part is devoted to the emptying of the batch, when the material is released into the mixer. The total time for 4 valves opening and the mixer release is  $t_B = (4 + 1) \cdot 2.4 = 12 \text{ s}$ . The request of a batch starts from the mixer weight analysis: when its mass signal descends under a defined value, a new batch is created. For each batch the control must define the opening times of the gate valves  $v_i(T)$ , where  $i$  is the gate index (from 1 to  $m$ ,  $m = 4$ ) and  $T$  identifies the  $T^{\text{th}}$  batch.

The user can define the total weight of the batch in terms of  $kg$  and the percentages of each component: in this way, once the opening order is defined, it is possible to generate a trajectory of weight reference against time.

In the rest of the section, the procedure illustrated in Section II will be applied to the present example. Each step of the proposed strategy will be then discussed in detail.

#### A. Experiments

The first operation, necessary to design the data driven control, is the data acquisition from a set of tests exploring different operating conditions.

As shown in Fig. 5 a set of  $r$  cycles scattered over the set of possible recipes as exemplified ( $r = 256$ ). Notice that the weight profile varies a lot from one point to another (Fig. 5), due to the dynamics of different valves (that also determine different over-shoots caused by falling material on the batch hopper).

#### B. PCA decomposition and validation

As previously defined, for each batch, the control inputs are constant values  $v_{1..m}(T)$  while the output is represented by a weight trajectory defined in  $13 \text{ s}$  (the number of samples is 3250). As shown in Section II, at the end of the acquisitions, the subsequent step of the algorithm is to elaborate the input/output data through a PCA decomposition.

Is easy to observe that, under the assumption of constant input values during a batch, a PCA decomposition of the inputs provides an obvious result: each input can be represented with a singular value  $z_1$  and the corresponding eigenvector is a vector of constant numbers. This fact means that, in this case, a PCA decomposition of the inputs is not necessary: the principal component of an input is the opening time of the valve (a total of 4 components, one for each valve).

The PCA decomposition, applied on the output signals extracted from the experiments (Fig. 5), generates a SVD map represented in Fig. 6: the most important elements in determining the weight profile are the first 4 patterns, illustrated in Fig. 7.

Employing only the first 4 eigenspaces (or “eigenweights”, see again Fig. 7), the reconstruction performance from the

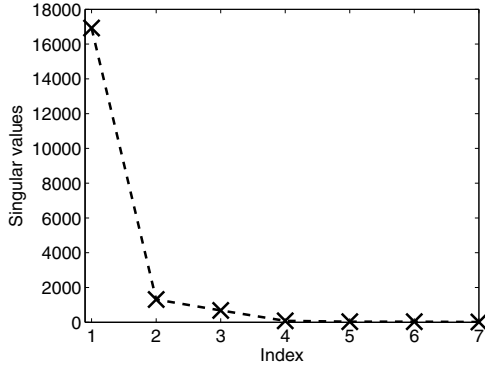


Fig. 6. Singular values.

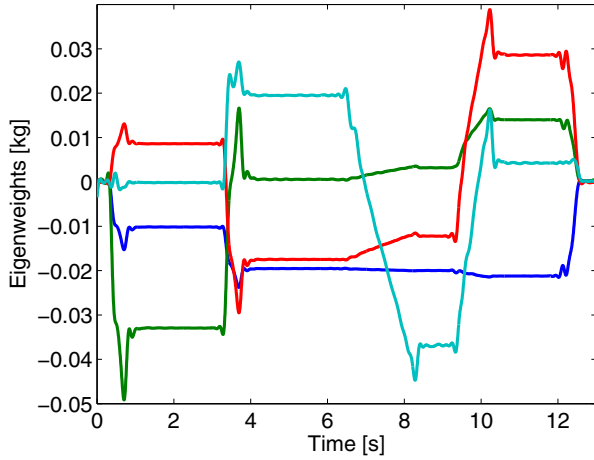


Fig. 7. Eigenspaces associated to the first 4 singular values in Fig. 6.

extracted features can be very accurate, as illustrated in Fig. 8 for 3 randomly chosen operating points. This fact means that, once the eigenweights are fixed, the weight description contained in a batch of length 3250 samples can be “compressed” in only 4 numerical features  $z_i$ ,  $i = 1, 2, 3, 4$ , of fixed value for each batch, without substantial loss of information. The 4 main features are then well-suited to represent the weight as output of the gravimetric blender fed by 4 opening times. As a consequence, the blender can be represented as a static (there is no effect of one batch on the following)  $4 \times 4$  system.

For completeness (even if not required by the algorithm), the identification of a model of the overall system is provided (in this case, a neural network [3]), in order to evaluate the structure of the overall system, including the PCA decomposition. The performance of the trained network on the identification data-set can be evaluated according to the percentage data-fitting error

$$err_i(T) = \frac{z_i(T) - \hat{z}_i(T)}{z_i(T)} \cdot 100,$$

where  $z_i(T)$  is the numerical value of the  $i^{th}$  feature at the  $T^{th}$  batch and  $\hat{z}_i(T)$  is the estimated  $i^{th}$  feature at the same

TABLE I

IDENTIFICATION DATA-SET: ASSESSMENT OF FITTING PERFORMANCE.

output feature	$mean[err_i(T)]$	$std[err_i(T)]$
1	5.99 %	$\pm 1.0236$ %
2	4.79 %	$\pm 0.7984$ %
3	3.21 %	$\pm 0.4697$ %
4	5.45 %	$\pm 0.4354$ %

TABLE II

IDENTIFICATION DATA-SET: ASSESSMENT OF PROFILE RECONSTRUCTION PERFORMANCE OVER 256 BATCHES.

$mean[err_y(T)]$	$std[err_y(T)]$
0.58 %	$\pm 0.972$ %

batch. In Table I, the mean value and the related standard deviation over all the tests are reported for each feature.

The mean and standard deviation of the normalized mean reconstruction error between the weight trajectory  $y(t, T)$  over the period of all 256 tests and the estimated weight  $\hat{y}(t, T)$ , namely

$$err_y(t, T) = \frac{y(t, T) - \hat{y}(t, T)}{y(t, T)} \cdot 100,$$

is instead shown in Table II. Notice that, as expected, a good fitting of the first 4 features representing the weight profile  $y$  implies the accurate matching of the weight data, too.

### C. Data-driven controller tuning

The required controller  $K$  has a dimension of  $4 \times 4$  and can be found using [7] and the signals already available and employed for the PCA. Specifically, the error vector  $e(T)$  is defined as the difference between the reference features  $z^o(T)$  and the measured features  $z(T)$ , whereas the output vector of the controller is made of 4 elements representing the opening times of the valves throughout the measured batches ( $v_i(T)$ ,  $i = 1 \dots 4$ ).

Specifically, every sampling period (every batch), the controller will take the weight measurement, compute the corresponding 4 features by projecting the weight profile on the eigenweights of Fig. 7 and change the opening times such that the next profile resembles the reference one.

In Fig. 10, 3 random reference recipes are proposed. With the controller computed using [7], the given specifications in terms of recipes are always respected; note that the weight signal, due to the force of the dropping material on the batch hopper, is affected by oscillations (in this type of machines the transient of the signal is uncontrollable: the most important part is the final value).

## IV. CONCLUSIONS

In this paper, a data-driven approach for the control of batch processes with fixed batch duration has been proposed. The main idea is to convert the time-domain trajectories into batch-domain trajectories using a PCA for both input and output signals. In this way, the adaptation to different requirements along the batches has been recast into a standard

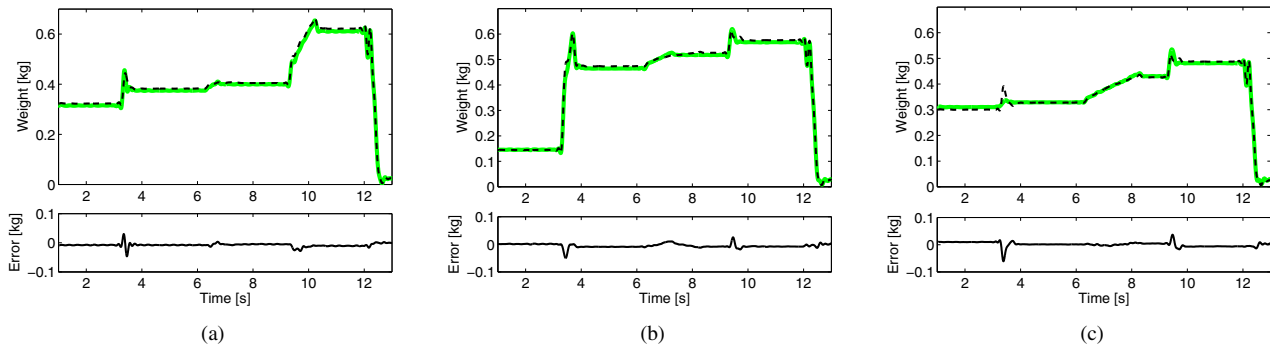


Fig. 8. Weight profiles (thick solid: measurements, thin dashed: reconstruction) and error plots for 3 random operating points.

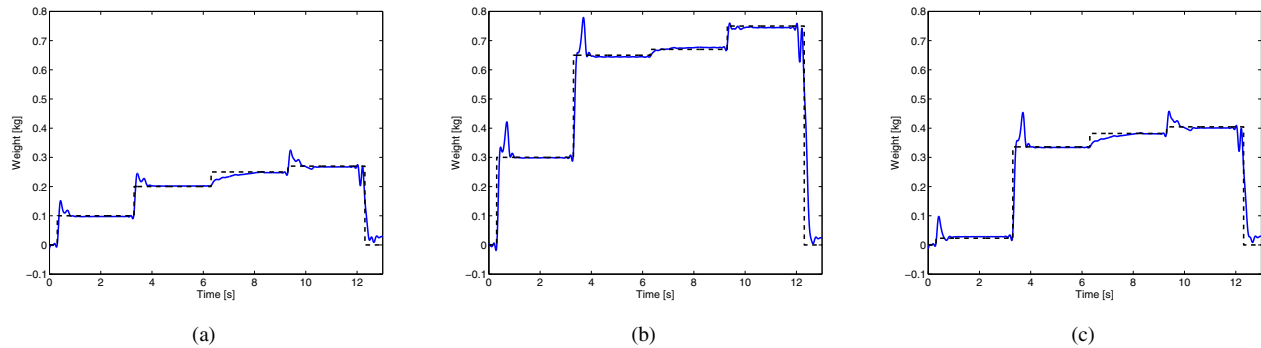


Fig. 10. Control performance on 3 random working points (dashed: reference, solid: output of the closed-loop system).

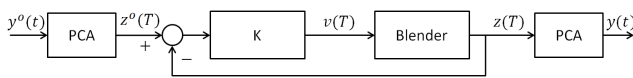


Fig. 9. Feature-based control architecture.

control problem. The controller design part can be dealt with using data only by employing the method introduced in [7].

The data-driven strategy showed to be an effective approach for the case of a batch gravimetric blender used in the plastic extrusion processes.

Future work will be devoted to the optimization of the batch time for different configurations of the batch blender.

## REFERENCES

- [1] Escobar, G. and Martinez, P.R. and Leyva-Ramos, J. and Mattavelli, P., *A negative feedback repetitive control scheme for harmonic compensation*, vol. 53, num. 4, pp. 1383–1386, IEEE Transactions on Industrial Electronics, IEEE, 2006.
- [2] Abidi, K. and Xu, J., *Iterative learning control for sampled-data systems: From theory to practice*, vol. 58, num. 7, pp. 3002–3015, IEEE Transactions on Industrial Electronics, IEEE, 2011.
- [3] Fukuda, T. and Shibata, T., *Theory and applications of neural networks for industrial control systems*, vol. 39, num. 6, pp. 472–489, IEEE Transactions on Industrial Electronics, IEEE, 1992.
- [4] Bonvin, D., *Control and optimization of batch processes*, vol. 26, num. 6, pp. 34–45, Control Systems, IEEE, 2006.
- [5] Wang, Y. and Gao, F. and Doyle III, F., *Survey on iterative learning control, repetitive control, and run-to-run control*, vol. 19, num. 10, pp. 1589–1600, Journal of Process Control, 2009.
- [6] Lee, J.H. and Lee, K.S., *Iterative learning control applied to batch processes: An overview*, vol. 15, num. 10, pp. 1306–1318, Control Engineering Practice, 2007.
- [7] Formentin, S. and Savaresi, SM and Del Re, L. *Non-iterative direct data-driven controller tuning for multivariable systems: theory and application*, vol. 6, num. 9, pp. 1250–1257, IET, 2012.
- [8] Passenbrunner, TE and Formentin, S. and Savaresi, SM and del Re, L. *Direct data-driven control of internal combustion engine test benches using closed-loop experiments*, pp. 3765–3770, 51th IEEE Conference on Decision and Control (CDC) IEEE, 2012.
- [9] Jolliffe, I.T. *Principal component analysis*, Wiley, 2005.
- [10] Sjöberg, Jonas and Zhang, Qinghua and Ljung, Lennart and Benveniste, Albert and Delyon, Bernard and Glorennec, Pierre-Yves and Hjalmarsson, Håkan and Juditsky, Anatoli *Nonlinear black-box modeling in system identification: a unified overview*, vol. 31, num. 12, pp. 1691–1724, Elsevier, 1995.
- [11] Hunt, K Jetal and Sbarbaro, D. and Żbikowski, R. and Gawthrop, PJ *Neural networks for control systems: a survey*, vol. 28, num. 6, pp. 1083–1112, Elsevier, 1992.
- [12] Miller III, W Thomas and Sutton, Richard S and Werbos, Paul J *Neural networks for control*, MIT press, 1995.
- [13] Previdi, F. and Belloli, D. and Cologni, A. and Savaresi, SM and Cazzola, D. and Madaschi, M. *Control System Design for a Continuous Gravimetric Blender*, vol. 18, num. 1, pp. 1025–1030, IFAC World Congress, 2011.
- [14] Fenner, R. T. and Cox, A. P. D. and Isherwood D. P. *On the dynamics and control of a plasticating extruder*, vol. 20, num. 6, pp. 733–736, Surging in screw extruders, 1979.
- [15] Costin, M. H. and Taylor, P. A. and Wright J. D. *A critical review of dynamic modelling and control of plasticating extruders*, vol. 22, num. 7, pp. 393–401, Polymer Engineering and Science, 1982.
- [16] Kochhar, A. K. and Parnaby, J. *Dynamical modelling and control of plastics extrusion processes*, vol. 13, pp. 177–183, Automatica, 1977.
- [17] Mudalamanan, R. and Bigio, D. I. *Process variations and the transient behavior of extruders*, vol. 49, num. 12, pp. 3150–3160, AIChE Journal, 2003.