A Data-Driven Approach to Control of Batch Processes With an Application to a Gravimetric Blender

Simone Formentin, *Member, IEEE*, Alberto L. Cologni, Fabio Previdi, *Member, IEEE*, and Sergio M. Savaresi, *Member, IEEE*

***Abstract*—In this paper, a data-driven controller design approach for batch processes is proposed. In this strategy, based on a suitable transformation of the input and output signals, a mathematical description of the process dynamics is not needed, and the working cycle is guaranteed to adapt to the desired operating conditions. Throughout this paper, all the steps of the above algorithm are described in detail with the help of an experimental case study dedicated to a gravimetric blender, which is a key element in the plastic extrusion process.**

***Index Terms*—Batch processes, data-driven control, dosing, polymers, principal component analysis (PCA).**

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 [8], "although model-based solutions are available, process models in the batch area tend to be poor," particularly in modern factories manufacturing products of high complexity.

It follows that, along with the increase in computational power of electronic control units (ECUs), 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 are iterative learning control and repetitive control [5], [6], [9], [10] 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 paper, a novel methodological framework will be proposed to deal with control of batch processes. Specifically, the first contribution of this paper is to show that the use of large data sets is as effective as the derivation of an accurate model of the system to enforce a desired behavior. The main assumption behind the proposed approach is that, for each batch, input and output time-domain trajectories can be "compressed" into a finite set of features, which describe the variations of the time series with respect to a predetermined pattern (or set of patterns). The advantages of the use of features instead of time trajectories can be assessed by looking at Fig. 1. By moving from time domain (t) to batch domain (T), the control problem can be 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 the time domain. Specifically, a principal component analysis (PCA) will be employed to compress input and output trajectories into suitable features from a large data set.

A second contribution of this paper is the application of the data-driven procedure in [11] for the design of the feedback controller in the batch domain. This choice will make the overall design procedure simple and still based on data only.

The above procedure will be illustrated throughout this paper with the help of an experimental case study concerning a gravimetric blender, which is a significant element in the plastic extrusion process [21], [23].

Traditionally, metering of plastic is performed by continuous volumetric blenders, which are not equipped to measure the plastic quantity actually delivered to the extruder, and those, therefore, are always controlled in an open loop with limited performance [19], [20], [25]–[28]. Gravimetric blenders, which are equipped with a load cell to measure the plastic weight fed into the extruder, can instead give the chance of a considerable increase in performance by means of closed-loop control [22].

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

Manuscript received August 22, 2013; revised November 28, 2013; accepted December 25, 2013. Date of publication February 25, 2014; date of current version June 6, 2014.

S. Formentin, A. L. Cologni, and F. Previdi are with the Laboratorio di Automatica, Dipartimento di Ingegneria, Università degli Studi di Bergamo, 24044 Dalmine, Italy.

S. M. Savaresi is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mail: simone.formentin@unibg.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2014.2308158

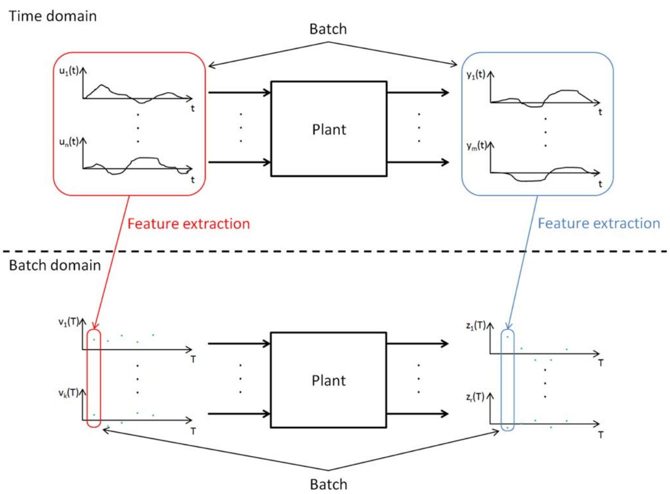

Fig. 1. From time domain to batch domain using data.

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., [17], 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.

For the sake of completeness, it should be said that some other important industrial problems have been addressed in the field of batch processes from a data-driven perspective, see, e.g., [3], [4].

The remainder of this paper is as follows. In Section II, the PCA-based approach for the batch system description and control is introduced and qualitatively described. 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 this paper.

II. Methodology

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

$$f\left(y(t), \dot{y}(t), \ldots, u(t), \dot{u}(t), \ldots\right) = 0 \qquad (1)$$

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, e.g., the temperature or the reference behavior. In these cases, it is intuitive that a pattern underlying the set of trajectories can be highlighted and it is then possible 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 (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 $t = 1, \ldots, N$ 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, PCA can be applied to easily obtain such a transformation.

PCA, also known as *Karhunen-Loéve transform* (see [2]), 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 Σ is a diagonal matrix composed by the α_i terms, $i = 1, \ldots, r$. The singular values α_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 vector $z(T)$ and their numerical values may vary for each batch. The same philosophy can be obvioulsy applied to the input trajectory.

The most important point in such an approach is that, if the reference output trajectory is given in terms of variations with respect to the output pattern, *the issue of making the system 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 [11] will be employed.

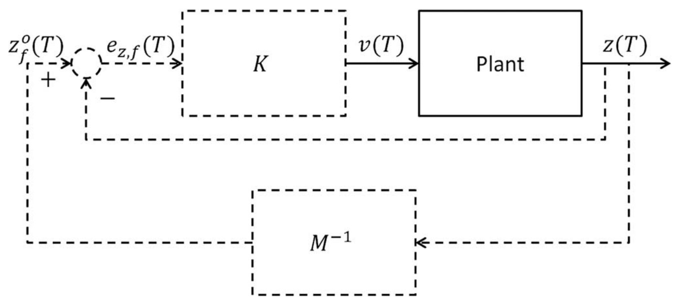

Fig. 2. Tuning scheme of the data-driven control approach in [11].

In [11], a linear time-invariant desired behavior M for the closed-loop system and a controller structure $K(\rho)$, parameterized with ρ, are supposed to be given by the user. In particular, in this paper, the PID controller

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

will be employed, where $B_k \in \mathcal{R}^{2\times 2}$ are matrices containing the unknown parameters, such that ρ is defined as

$$\rho = \left[\mathrm{vec}^T(B_0)\ldots\mathrm{vec}^T(B_2)\right]^T. \tag{3}$$

The regression form of the controller is then

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

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.

The main idea of method [11] 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

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

is minimized with respect to ρ 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 [11], it has been also proven that, provided that $v(T)$ and $z(T)$ are suitably prefiltered by $L = M$, the desired closed-loop behavior can be achieved. See [11]–[13] 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 one 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 data set exploring the whole operating space is required. It should be also 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.

Batch-domain control strategy

A. **Data collection**: perform a set of experiments scattered over the whole operating range, 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 multivariable PID controller using the method in [11].

III. Gravimetric Blender Case Study

As already mentioned, the blender is an important element in the plastic extrusion process. Usually, by means of gravimetric or volumetric blenders, 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 thermomechanical transformations inducing strong changes in the physical properties of the material [18], [24], [27]. The final product quality in extrusion is essentially characterized by a precisely regulated 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

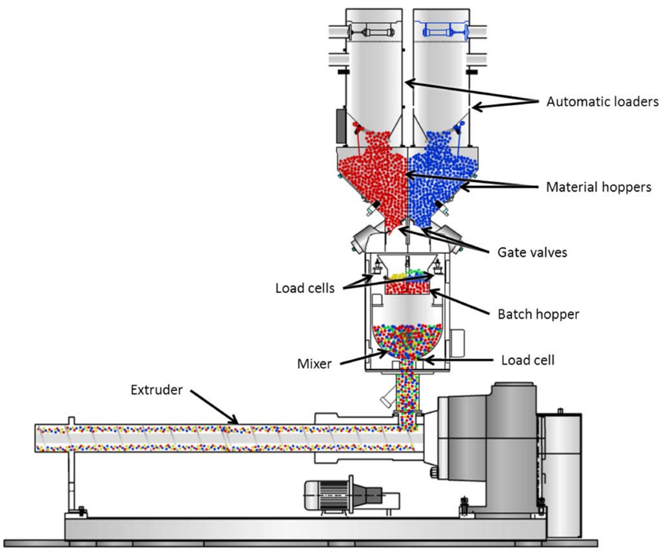

Fig. 3. Schematics of the employed experimental setup with four gate valves.

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 blender 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 four gate valves necessary to dispense the material;
- a batch hopper that collect the material outgoing from the gate valves;
- 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, it is possible to define, for each $T_s = 4$ ms sampling period, the state of the actuators.

The typical working cycle is defined as a time scheduling of basic operations.

In the first part, every meter releases the material into the batch hopper; the gate valve's opening is executed on a regular basis every 2.4 s. The last part is devoted to the emptying of the batch, when the material is released into the mixer. The total time for four valves opening and the mixer release is $t_B = (4 + 1) \cdot 2.4 = 12$ 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 T, the control must define the opening times of the gate valves v_i, where i is the gate index (from 1 to m, $m = 4$). The user can define the total weight of the batch in terms of kg and the percentages of each component. This way, once the opening order is defined, it is possible to generate a trajectory of weight reference against time.

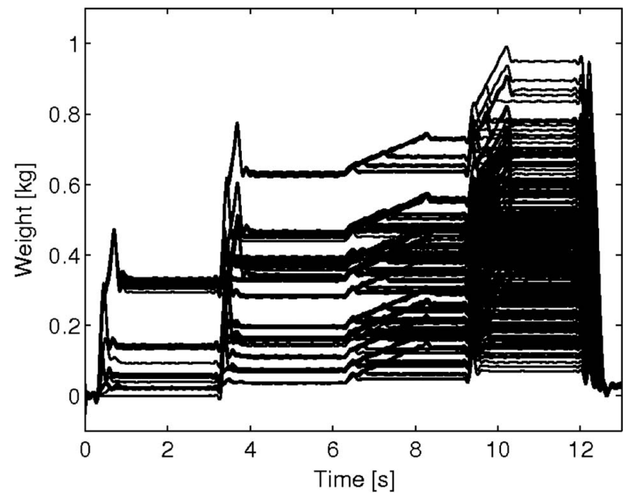

Fig. 4. Spread of working cycles.

TABLE I
SINGULAR VALUES

Singular value	1	2	3	4	5	6	7
Value	1692	1307	677.5	80.8	34.9	32	13.4

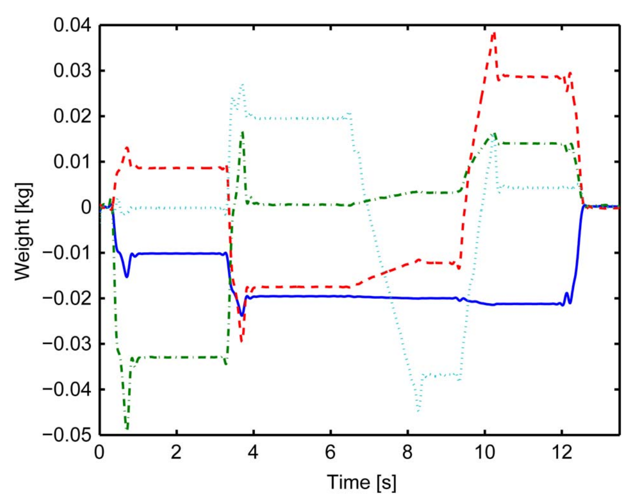

Fig. 5. Eigenspaces associated to the first four singular values in Table I: (solid) first, (dashed-dotted) second, (dashed) third, and (dotted) fourth.

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 an example, a set of r cycles scattered over the set of possible recipes (with $r = 256$) is exemplified in Fig. 4. Notice that the weight profile varies a lot from one point to another, due to the dynamics of different valves (that also determine different over-shoots caused by falling material on the batch hopper).

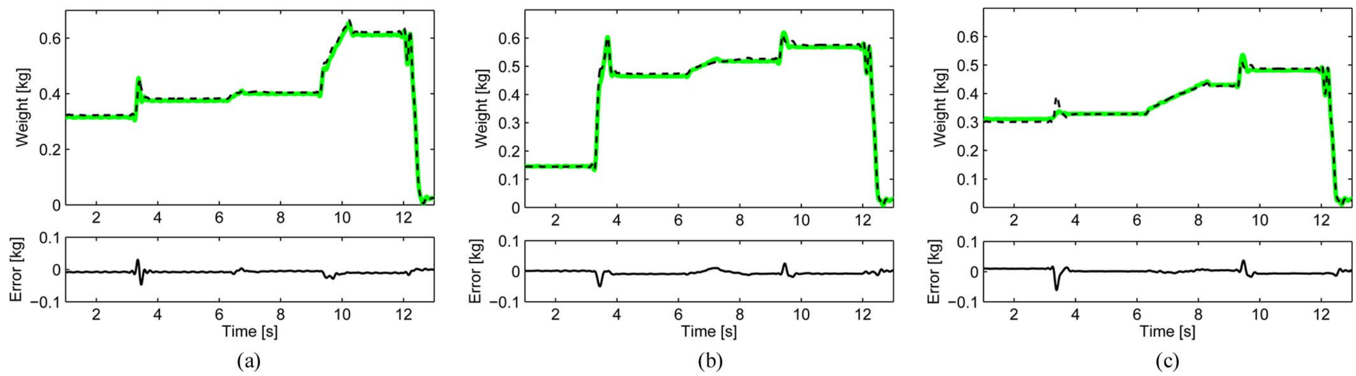

Fig. 6. Weight profiles (thick solid: measurements, thin dashed: reconstruction) and error plots for three random operating points.

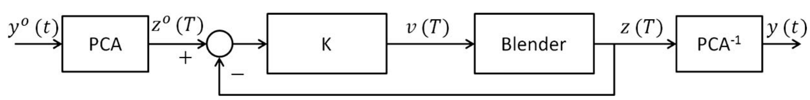

Fig. 7. Feature-based control architecture.

B. PCA Decomposition

As previously defined, for each batch T, the control inputs are constant values $v_{1\ldots m}$ whereas the output is represented by a weight trajectory defined in 13 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 I/O data through a PCA decomposition.

It 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 four components, one for each valve).

The PCA decomposition, applied on the output signals extracted from the experiments (see Fig. 4), generates an SVD map described in Table I: the most important elements in determining the weight profile are the first four patterns, illustrated in Fig. 5.

C. PCA Validation

Employing only the first 4 eigenspaces (or "eigenweights", see again Fig. 5), the reconstruction performance from the extracted features can be very accurate, as illustrated in Fig. 6 for three 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 four numerical features z_i, $i = 1, 2, 3, 4$, of fixed value for each batch, without substantial loss of information. The four main features are then well-suited to represent the weight as output of the gravimetric blender fed by four opening times. As a consequence, the blender can be represented as a static (there is no effect of one batch on the following) 4×4 system (see Fig. 7).

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 [7]), 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 ith feature at the Tth batch and $\hat{z}_i(T)$ is the estimated ith feature at the same batch. In Table II, the mean value and the related standard deviation over all the tests are reported for each feature.

TABLE II
IDENTIFICATION DATA SET: ASSESSMENT OF FITTING PERFORMANCE

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

TABLE III
IDENTIFICATION DATA SET: ASSESSMENT OF PROFILE RECONSTRUCTION PERFORMANCE OVER 256 BATCHES

$mean[err_y(t,T)]$	$std[err_y(t,T)]$
0.58 %	± 0.972 %

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 III. 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.

D. Data-Driven Controller Tuning

The required controller K has a dimension of 4×4 and can be found using [11] 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 $z^o(T)$

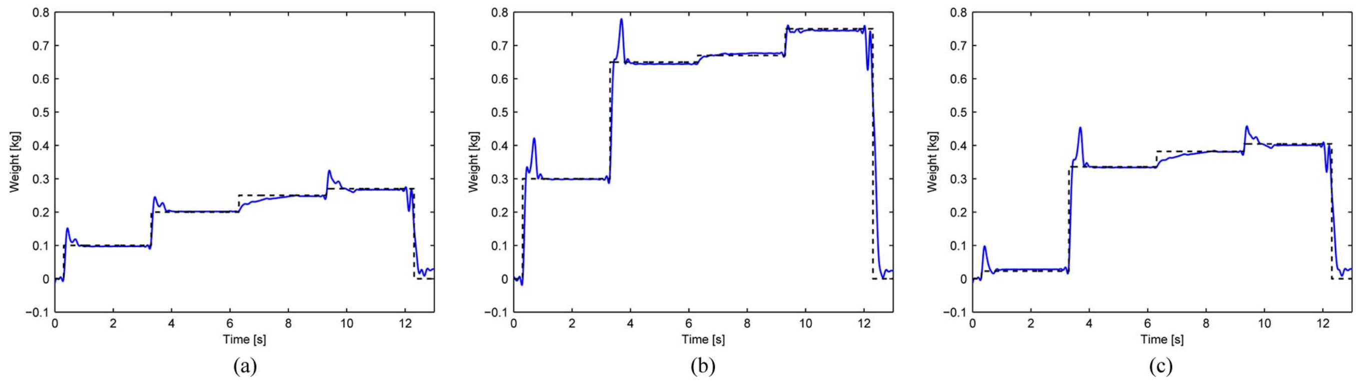

Fig. 8. Control performance on three random working points (dashed: reference, solid: output of the closed-loop system).

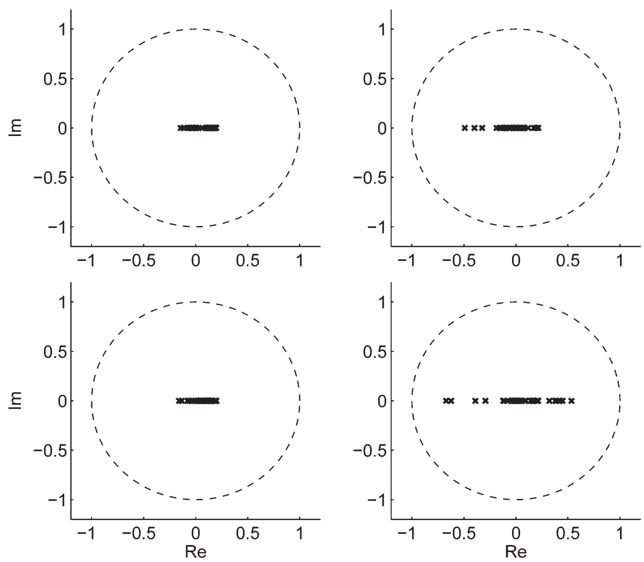

Fig. 9. Spread of the dominant poles (crosses) of the 4 closed-loop system over 256 Monte Carlo runs: first hopper (top left), second hopper (top right), third hopper (bottom left) and fourth hopper (bottom right). Notice that the closed-loop stability is always guaranteed with a good margin (the maximum absolute value of a pole is less than 0.65).

and the measured features $z(T)$, whereas the output vector of the controller is made of four elements representing the opening times of the valves throughout the measured batches $(v_i(T), i = 1 \ldots 4)$.

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. 5 and change the opening times of the valves such that the next profile resembles the reference one.

The control law is computed here using the approach in Section II in the new batch-domain framework. The desired reference model M is selected, for every hopper, as a simple one step delay system. As an example, in Fig. 8, three random reference recipes are used to excite the closed-loop system with the above controller. Notice that the given specifications in terms of recipes are always respected. Moreover, even if the reference behavior M cannot be fully achieved, the four closed-loop systems are always stable, as shown in Fig. 9, where the dominant pole of each closed-loop system is plotted over 256 Monte Carlo tests with different reference excitation. The dominant pole of the closed-loop system has been found by means of linear identification (see [1]) of the closed-loop system using reference and output data.

Notice that, using the approach in [11], the closed-loop stability in case of not fully achievable M can be verified from data only this way, since a model of the multivariable system is not available. However, this fact is not in contrast with the framework of the proposed method, where experiments are supposed to be cheap, whereas it is hard to undertake a full modeling study. Notice also that, even in a model-based approach, stability can be proven only for the closed-loop system with the model of the plant, and not for the real system.

Finally, it should be remarked here that the weight signal in Fig. 8, due to the force of the dropping material on the batch hopper, is affected by oscillations. However, using a constant opening time, the transient of the signal is intrinsically not controllable and therefore the only important part is the final value.

The behavior of the closed-loop system (with the obtained controller) will be now compared with the state-of-the-art open-loop controllers used in the industry.

Specifically, a test is performed during which the density of the material is suddenly changed, going from 0.35 kg/L (Material 1) through 0.46 kg/L (Material 2) to 0.53 kg/L (Material 3). As it can be observed in Fig. 10, the changes of material corresponds to step disturbances in the output features. The proposed controller obviously shows to be more effective against such a change in the system operating conditions. Moreover, the dosing performance of the closed-loop system and that of the open loop control system can be compared also in terms of weight tracking. Fig. 11 shows three different tests where the density of the material changes. Finally, notice that, at the end of a transient phase (after three batches in the examples), the designed controller is able to guarantee the accurate tracking of the desired recipe.

IV. Conclusion

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. This way, the adaptation to different requirements along the batches has been recast into a standard control problem.

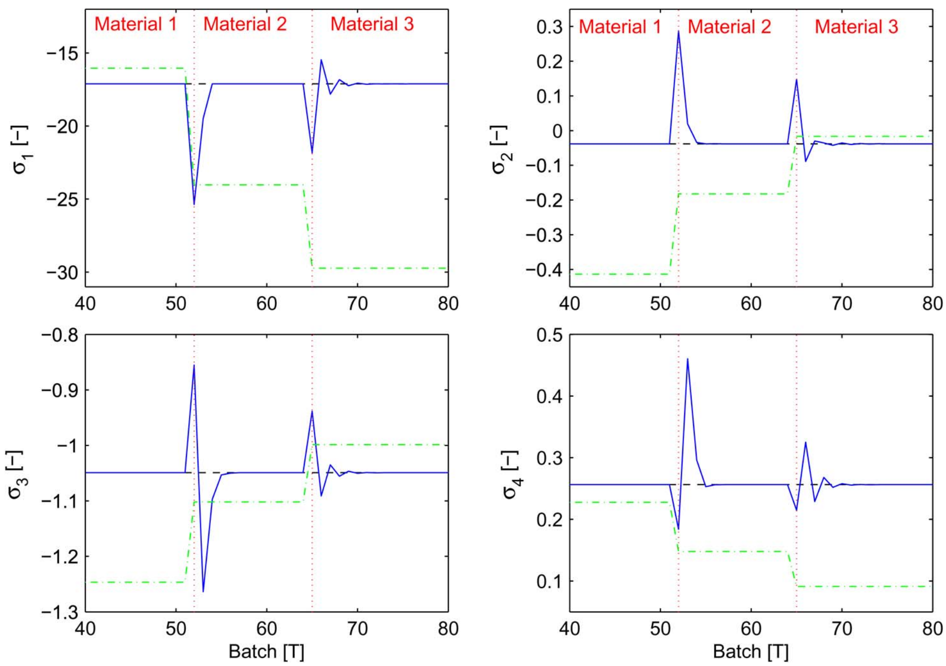

Fig. 10. Control performance in case of material changes (dashed: constant reference, solid: system with the closed-loop controller, dashed-dotted: system with the open-loop controller). The diagram represents the batches (x-axis) versus the features (y-axis), computed according to the definition of z in Section II.

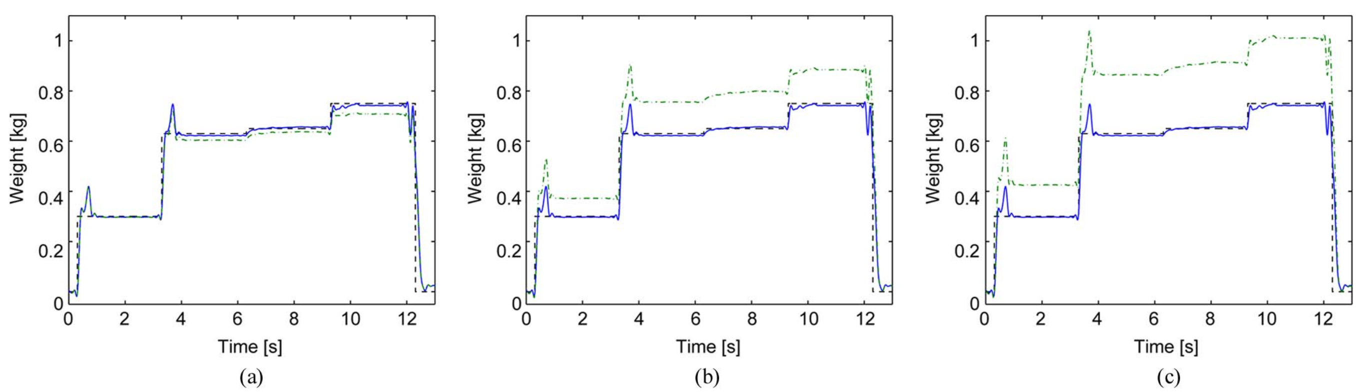

Fig. 11. Three examples of weight trajectory tracking in case of a change of material (dashed: reference, solid: output of the closed-loop system, dashed-dotted: output of the open-loop system).

The controller design part can be dealt with using data only by employing the method introduced in [11].

The data-driven strategy has been first presented qualitatively and then discussed in detail on an experimental case study. Specifically, the regulation of batch gravimetric blenders used in the plastic industry has been coped with and satisfactory results have been obtained. From a practical point of view, the desired recipes have been always achieved and the time for change of recipe has been quantified in one cycle only.

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

REFERENCES

[1] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2005.

[2] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[3] S. Yin, S. X. Ding, S. Abandan, H. Adel, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *Int. J. Syst. Sci.*, vol. 44, no. 7, pp. 1366–1376, Jul. 2013.

[4] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May 2014.

[5] G. Escobar, P. R. Martinez, J. Leyva-Ramos, and P. Mattavelli, "A negative feedback repetitive control scheme for harmonic compensation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1383–1386, Aug. 2006.

[6] K. Abidi and J. Xu, "Iterative learning control for sampled-data systems: From theory to practice," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 3002–3015, Jul. 2011.

[7] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Trans. Ind. Electron.*, vol. 39, no. 6, pp. 472–489, Dec. 1992.

[8] D. Bonvin, "Control and optimization of batch processes," *IEEE Control Syst. Mag.*, vol. 26, no. 6, pp. 34–45, Dec. 2006.

[9] Y. Wang, F. Gao, and F. Doyle, III, "Survey on iterative learning control, repetitive control, run-to-run control," *J. Process Control*, vol. 19, no. 10, pp. 1589–1600, Dec. 2009.

[10] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Eng. Pract.*, vol. 15, no. 10, pp. 1306–1318, Oct. 2007.

[11] S. Formentin, S. M. Savaresi, and L. Del Re, "Non-iterative direct data-driven controller tuning for multivariable systems: Theory and application," *IET Control Theory Appl.*, vol. 6, no. 9, pp. 1250–1257, Jun. 2012.

[12] S. Formentin and S. M. Savaresi, “Noniterative data-driven design of multivariable controllers,” in *Proc. 50th IEEE CDC-ECC*, 2011, pp. 5106–5111.

[13] T. E. Passenbrunner, S. Formentin, S. M. Savaresi, and L. Del Re, “Direct data-driven control of internal combustion engine test benches using closed-loop experiments,” in *Proc. 51st IEEE CDC*, 2012, pp. 3765–3770.

[14] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky, “Nonlinear black-box modeling in system identification: A unified overview,” *Automatica*, vol. 31, no. 12, pp. 1691–1724, Dec. 1995.

[15] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems: A survey,” *Automatica*, vol. 28, no. 6, pp. 1083–1112, Nov. 1992.

[16] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*. Cambridge, MA, USA: MIT Press, 1995.

[17] F. Previdi, D. Belloli, A. Cologni, S. M. Savaresi, D. Cazzola, and M. Madaschi, “Control system design for a continuous gravimetric blender,” in *Proc. IFAC World Congr.*, 2011, vol. 18, no. 1, pp. 1025–1030.

[18] R. T. Fenner, A. P. D. Cox, and D. P. Isherwood, “On the dynamics and control of a plasticating extruder,” *Surging Screw Extruders*, vol. 20, no. 6, pp. 733–736, 1979.

[19] T. O. Broadhead, W. I. Patterson, and J. M. Dealy, “Closed loop viscosity control of reactve extrusion with an in-line rheometer,” *Polymer Eng. Sci.*, vol. 36, no. 23, pp. 2840–2851, Dec. 1996.

[20] S. H. Chiu and S. H. Pong, “In-line viscosity fuzzy control,” *Polymer Eng. Sci.*, vol. 79, no. 7, pp. 1249–1255, Feb. 2000.

[21] M. H. Costin, P. A. Taylor, and J. D. Wright, “A critical review of dynamic modelling and control of plasticating extruders,” *Polymer Eng. Sci.*, vol. 22, no. 7, pp. 393–401, 1982.

[22] R. Isermann and U. Raab, “Intelligent actuators ways to autonomous actuating systems,” *Automatica*, vol. 29, no. 5, pp. 1315–1331, Sep. 1993.

[23] A. K. Kochhar and J. Parnaby, “Dynamical modelling and control of plastics extrusion processes,” *Automatica*, vol. 13, no. 2, pp. 177–183, Mar. 1977.

[24] R. Mudalamane and D. I. Bigio, “Process variations and the transient behavior of extruders,” *AIChE J.*, vol. 49, no. 12, pp. 3150–3160, Dec. 2003.

[25] M. Del Pilar Noriega and C. Rauwendaal, *Troubleshooting the Extrusion Process: A Systematic Approach to Solving Plastic Extrusion Problems.* Cincinnati, OH, USA: Hanser, 2001.

[26] F. Previdi, S. M. Savaresi, and A. Panarotto, “Design of a feedback control system for real time control of flow in a single screw extruder,” *Control Eng. Pract.*, vol. 14, pp. 1111–1121, 2006.

[27] C. Rauwendaal, *Polymer Extrusion*. Cincinnati, OH, USA: Hanser, 2001.

[28] P. E. Wellstead, W. P. Health, and A. P. Kjaer, “Identification and control of web processes: Polymer film extrusion,” *AIChE J.*, vol. 6, no. 3, pp. 321–331, Mar. 1998.

Simone Formentin (M’13) was born in Legnano, Italy, in 1984. He received the B.Sc. and M.Sc. degrees *(summa cum laude)* in control system engineering from the Politecnico di Milano, Milan, Italy, in 2006 and 2008, respectively, and the Ph.D. degree in automatic control within a joint program between the Politecnico di Milano and Johannes Kepler University, Linz, Austria, in 2012.

From December 2011 to August 2012, he was a Postdoctoral Fellow with the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He is currently a Postdoctoral Fellow with the Università degli studi di Bergamo, Dalmine, Italy. His research interests include data-driven controller tuning approaches and automotive control.

Alberto L. Cologni was born in Ponte San Pietro, Italy, in 1985. He received the M.S. degree *(cum laude)* in computer engineering in 2009 from the Università degli Studi di Bergamo, Dalmine, Italy, where he is currently working toward the Ph.D. degree. His M.S. thesis concerned modeling and identification of an electrohydrostatic actuator.

In January 2010, he joined as a Researcher of the Cometha project with Intellimech, Bergamo, Italy. His main research interests include the development and implementation of control algorithms.

Fabio Previdi (M’10) was born in Milan, Italy, in 1968. He received the M.Sc. degree in electronic engineering (control systems) and the Ph.D. degree in computer science and control systems engineering from the Politecnico di Milano, Milan, Italy, in 1993 and 1999, respectively.

From 1999 to 2000, he was a Research Assistant with the Centre for Systems and Control of the Department of Mechanical Engineering, University of Glasgow, Glasgow, U.K. From 2000 to 2002, he was a Research Assistant with the Dipartimento di Elettronica e Informazione, Politecnico di Milano. From 2002 to 2006, he was an Assistant Professor with the Engineering Faculty, Università degli Studi di Bergamo, Dalmine, Italy, where, in 2006, he became an Associate Professor. His research interests include fault diagnosis, system identification, control of biomedical systems, and application of control systems methods in physics.

Sergio M. Savaresi (M’00) was born in Manerbio, Italy, in 1968. He received the M.Sc. degree in electrical engineering and the Ph.D. degree in systems and control engineering from the Politecnico di Milano, Milan, Italy, 1992 and 1996, respectively, and the M.Sc. degree in applied mathematics from the Catholic University, Brescia, Italy, in 2000.

After receiving the Ph.D. degree, he was a Management Consultant with McKinsey&Co, Milan. He is currently a Full Professor of automatic control with the Politecnico di Milano. He was a Visiting Researcher with Lund University, Lund, Sweden; University of Twente, Enschede, The Netherlands; Canberra National University, Canberra, Australia; Stanford University, Stanford, CA, USA; University of Minnesota, Minneapolis, MN, USA; and Johannes Kepler University, Linz, Austria.

Prof. Savaresi is an Associate Editor of the *European Journal of Control* and *IET Transactions on Control Theory and Applications*. He is a Member of the Editorial Board of the IEEE Control Systems Society. He is a Director of the Milan-Singapore Executive master “ICT and Design for Innovation”. His main interests are in the areas of vehicles control, automotive systems, data analysis and system identification, nonlinear control theory, and control applications.