Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Enhanced Big Data Approximating Control of an Industrial Paste Thickener

Saúl Langarica * **Felipe Núñez** *

* *Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Santiago, Chile 7820436. (e-mail: salangarica@uc.cl, fenunez@ing.puc.cl).*

Abstract: The Big Data revolution refers to using a large amount of data to improve decision making. In process control applications, the use of big data techniques has been restricted to complementing classical control schemes as model-based or PID approaches. This work focuses on a model-free purely data-driven control strategy known as Big Data approximating control (BDAC), which was recently introduced in the context of process control. In particular, this work proposes two modifications to the classical BDAC formulation and presents a real implementation of the enhanced BDAC technique to a real industrial paste thickener.

Keywords: Big data, Cyber-physical systems, Process control, Industrial automation.

1. INTRODUCTION

Process control is a mature area of control engineering that aims at enhancing the performance of an industrial process by applying automatic control concepts. Among the existing control techniques, model-based control has become the state-of-the-art tool in process control applications, showing great success in oil, paper and pulp, and mining processes (Mayne, 2014; Qin and Badgwell, 2003).

Recently, industrial facilities have seen an important upgrade in connectivity, driven by the Industrial Internet paradigm, which has resulted in the availability of an enormous amount of process data, typically in the form of time-series, which represents an opportunity to develop data-driven algorithms and applications for supervision, modeling and control (Ge, 2017). This is known as the big data revolution (Mayer-Schonberger and Cukier, 2013), whose main purpose is to use large amounts of data to enable knowledge discovery and better decision making.

The big data revolution has bolstered the fast development of technologies and techniques for handling enormous amounts of data. Machine learning and deep learning algorithms are examples of data-driven techniques that have had impressive success in extracting hidden patterns and rich information from big data in different areas.

As expected, the use of big data techniques has also impacted process control. However, although there are many applications based on these new techniques, most of them have been developed following a classical control scheme, replacing some of its components but without altering the underlying structure. For example, Lu et al. (2017) presents an adaptive PID with a neural network to regulate its gains. The work by Núñez et al. (2020) presents a Model Predictive Control (MPC) scheme using a neural network as a model and a nonlinear optimizer, and Liang et al. (2017) presents an optimal controller using a support vector machine as the model. Despite the fact that these techniques have proven to be effective, classic control schemes were not designed for getting the most out of big data. Therefore, designing a controller architecture tailored specifically for exploiting big data in the context of process control is of great interest.

A first step in this direction was taken in Stanley (2018), where a new control paradigm, model-free and purely based on a big data approach, called Big data approximating control (BDAC) is introduced. In Stanley (2018), it was shown that BDAC is capable of controlling different systems with almost no modifications. However, all the evaluations were performed in a simulated environment and its performance in a real process is yet to be seen. In this work, we first propose two key modifications to the original formulation of the BDAC in what we call an enhanced BDAC scheme. We then study the application of BDAC to a real industrial process: a paste thickener, and we show that this new control paradigm is capable of controlling this challenging process, in a real-life scenario, without the use of an explicit model and only relying on archived data. Consequently, the contributions of this article are twofold, namely, the formulation of the enhanced BDAC scheme and its application to a real industrial process.

1.1 Notation and Basic Definitions

In this work, $\mathbb{R}$ denotes the real numbers, $\mathbb{Z}_{\geq 0}$ the non-negative integers, $\mathbb{R}^n$ the Euclidean space of dimension n, and $\mathbb{R}^{n\times m}$ the set of $n \times m$ matrices with real coefficients. For $a, b \in \mathbb{Z}_{\geq 0}$ we use $[a; b]$ to denote their closed interval in $\mathbb{Z}$. For a vector $v \in \mathbb{R}^n$, v^i denotes its ith component. For a matrix $A \in \mathbb{R}^{n\times m}$, A_i denotes its ith column, A^i its ith row, and $||A||_{2,1}$ its $L_{2,1}$ norm, given by $||A||_{2,1} = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}}$, where a_{ij} is the ijth entry of A. For an n-dimensional real-valued sequence $\alpha : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$,

⋆ This work was partially funded by ANID under grant ANID PIA ACT192013.

$\alpha(t)$ denotes its tth element, and $\alpha_{[a;b]}$ denotes its restriction to the interval $[a;b]$, i.e., a sub-sequence. For an n-dimensional sub-sequence $\alpha_{[a;b]}$, $M(\alpha_{[a;b]}) \in \mathbb{R}^{n\times(b-a+1)}$ is a matrix whose ith column is equal to $\alpha(a+i-1)$, with $i \in [1; b-a+1]$.

2. BDAC BACKGROUND

Consider a generic non-linear time-invariant system Σ described by

$$\Sigma : \begin{cases} \dot{x} = f(x,u), & x \in \mathcal{X} \subseteq \mathbb{R}^K, u \in \mathcal{U} \subseteq \mathbb{R}^M \\ y = h(x,u), & y \in \mathcal{Y} \subseteq \mathbb{R}^N \end{cases}, \tag{1}$$

where f and h are smooth non-linear mappings. Assume that the system operates over a set of steady-states $\mathcal{I}$, such that for all $(y^i, x^i, u^i) \in \mathcal{I}$, $f(x^i, u^i) = 0$ and $h(x^i, u^i) = y^i$ holds. Moreover, for each $(y^i, x^i, u^i) \in \mathcal{I}$, Σ is locally asymptotically controllable to (y^i, x^i, u^i). BDAC is a discrete-time control strategy that solves the problem of regulating the system Σ to an operating point $(y^i, x^i, u^i) \in \mathcal{I}$ by using only historical data in the form of sampled time series, i.e., sequences $\bar{y}$ and $\bar{u}$ such that $\bar{y}(k)$ and $\bar{u}(k)$ represents the output and input of Σ at time instant kT respectively, where T is the sampling period. BDAC works in two stages: i) a training phase where the controller "learns" the system dynamics by gathering sequences that contain rich input-output information; and ii) an estimation and control phase, where given a sequence representing the current operating condition of Σ and a reference trajectory, the controller retrieves a control sequence from which the first input is applied, following the receding horizon control principle (Kwon and Han, 2005).

2.1 Training phase

During the training phase, BDAC works with an observation sequence $\bar{\Sigma}$, with $\bar{\Sigma}^T(k) = [\bar{y}^T(k)\ \bar{u}^T(k)]$, to generate a database S such that each $s \in S$, a trajectory, is a matrix $s = M(\bar{\Sigma}_{[k-2n_h;k]})$ where k is the time instant at which the trajectory s was acquired. We can think of S as a subset of a set Ω that contains all possible system trajectories of length $2n_h + 1$. As it might be thought, Ω could be very large and intractable. In contrast, we want S to remain of a manageable size with as much information as possible. Therefore, in addition to recording only trajectories with enough dynamic information, similar trajectories are merged, which also helps S to be adaptive in case the system is non-stationary.

Rejection: Consider a database S for Σ is being formed. When a new candidate trajectory s_{new} is available, it goes a first screening process to check whether the system is in steady-state or not. To this end, a simple steady state detector is created just by analyzing the difference in mean and variance at different points of all the constituent signals of s_{new}. If the differences are lower than a threshold, then the system is declared in steady state and the trajectory s_{new} is rejected for lacking dynamic content.

Filtering: If the trajectory s_{new} was not rejected, then it can enter S. However, there are three possible options for the new trajectory to enter S, depending on: a given threshold d_{th}, the maximum size for S denoted as S_{max}, and the closest trajectory to s_{new} already existing in S, s_{close}, where $s_{\text{close}} = argmin_s(||s - s_{\text{new}}||_{2,1}), s \in S$.

- $||s_{\text{new}} - s_{\text{close}}||_{2,1} > d_{th}$: In this case, s_{new} is incorporated to S as a new trajectory.
- $||s_{\text{new}} - s_{\text{close}}||_{2,1} \le d_{th}$: s_{new} is incorporated recursively through real time exponential filter clustering (RTEFC) given by (Stanley, 2018):
  $$s_{\text{close}} = \alpha s_{\text{close}} + (1-\alpha)s_{\text{new}}. \tag{2}$$
  This filtering helps to attenuate noise and the impact of unmeasured disturbances and, additionally, helps S to slowly adapt to process changes.
- Size of S is greater than S_{max}: In this case, forced filtering occurs using RTEFC independently of $||s_{\text{new}} - s_{\text{close}}||_{2,1}$.

It should be noted that, as stated in Stanley (2018), to account for causality, the definition of the norm is modified. Here $||s_{\text{new}} - s_{\text{close}}||_{2,1}$ could be thought of as a weighted $L_{2,1}$ norm where all the weight values for process outputs are set to zero, so the distance calculation is only over process inputs. By that, we are clustering together trajectories of similar inputs but not necessarily similar outputs, thus letting S to adapt to slow system changes that cause outputs to same inputs to be different.

Table 1. Target calculation summary

Type of signal	Values for $s^{\text{virtual}}_{\text{target}}$
Controlled variables	Setpoint
Manipulated variables	Last value
Disturbances	Last measurement

2.2 Control and Estimation phase

After the training phase is finished, hence S is formed by a large number of different trajectories, the online control and estimation phase starts. However, it should be noted that training can continue online and more trajectories should be incorporated to S during this phase to allow the controller to adapt.

Target trajectory: The first step is to construct a target trajectory for the system, s_{target}. To this end, at each control instant k, s_{target} is generated by concatenating two sub-trajectories as $s_{\text{target}} = [s^{\text{real}}_{\text{target}}\ s^{\text{virtual}}_{\text{target}}]$, where $s^{\text{real}}_{\text{target}} = M(\bar{\Sigma}_{[k-n_h;k]})$ is a trajectory that considers the system dynamics during the last $n_h + 1$ sampling instants, and $s^{\text{virtual}}_{\text{target}}$ is a virtual trajectory of size n_h constructed depending on the three types of signals that it could contain. 1) First for controlled variables, $s^{\text{virtual}}_{\text{target}}$ contains the desired setpoints. 2) In the case of manipulated variables, since we would like to minimize control variations, $s^{\text{virtual}}_{\text{target}}$ should hold the last value applied to the process. 3) for process inputs acting as disturbances, since their behavior is uncertain, $s^{\text{virtual}}_{\text{target}}$ holds the last measured value. Table 1 presents the values in $s^{\text{virtual}}_{\text{target}}$ for each type of variable.

BDAC approximation problem: When generating s_{target} we were careful to construct it based on control objectives, such as reaching the desired setpoint under a minimum control effort. However, consistency with process dynamics was not considered at all. The BDAC approximation problem consists in approximating s_{target}, in the best possible way trying not to violate the system dynamics. Mathematically, this amounts to finding

$$s^* = argmin_s ||s - s_{\text{target}}||_{2,1}, \quad s \in \Omega, \tag{3}$$

where Ω is the set of all possible trajectories of the system. Since s_{target} may not be feasible, and we do not have access to the set Ω, the BDAC approximation problem is solved using trajectories stored in S. In Stanley (2018) several solutions to this problem were proposed. Here we revisit those solutions and propose novel solutions to the BDAC approximation problem

- *Direct Matching*: This is the simplest method and it consists in approximating s^* just by the closest trajectory in S. Mathematically:
$$s^* = argmin_s ||s_{\text{target}} - s||_{2,1}, \quad s \in S \tag{4}$$
Intuitively, as S tends to Ω direct matching should have better results and in the limit where $S = \Omega$ this simple method would achieve the best performance. However, when S is small relatively to Ω, this method achieves poor results as we will see later.
- *Orthogonal decomposition*: This method consists in constructing an orthonormal set of basis vectors e_i from the rows of S, using the Gram-Schmidt procedure (Leon et al., 2013). Then, the solution is given by the orthogonal projection of s_{target} onto the space created by the basis vectors. Mathematically,
$$s^* = \sum_i < s_{target}, e_i > e_i, \tag{5}$$
where $< \cdot, \cdot >$ denotes the inner product of two vectors. This linear technique was proposed in Stanley (2018) and used in its simulations.
- *Inverse distance weighting* (IDW): Also proposed in Stanley (2018), IDW is a nonlinear multivariate interpolation method popular in image processing that is based on normalized weights calculated as the inverse of the L-distance between s_{target} and the stored trajectories in S. Mathematically,
$$s^* = \frac{\sum_i w_i s_i}{\sum_i w_i}, \quad w_i = \frac{1}{||s_{\text{target}} - s_i||^L} \tag{6}$$
- *PCA-approximation*: We now propose a new candidate method based on Principal component analysis (Wold et al., 1987). This method is a linear technique similar to orthogonal decomposition, but here the basis vectors are the eigenvectors of the covariance matrix of the rows of S, ordered based on the magnitude of their corresponding eigenvalues. In this case s^* is calculated by taking the projection of s_{target} to the principal components subspace, and then transforming back to the original space using only the first n ($n \leq dim(S)$) components. The idea behind using only the first n principal components for reconstruction is to avoid the effect of undesired outliers or noise that might be present in S (Langarica et al., 2020).
- *PCA-approximation with KNN*: This novel candidate method is a nonlinear extension of the PCA-approximation. In this case, the PCA transformation is not done with the entire S but only with the k nearest neighbors of s_{target}. Even when PCA is used with a linear kernel, the locality of this method helps to approximating nonlinearities of the space generated by the vectors in S in a better way. PCA-approximation with KNN can be implemented with both linear and nonlinear kernels.

After solving the BDAC approximation problem with any of the methods explained above, the manipulated variables are extracted from s^* and applied to the process. The whole control and estimation procedure must be repeated each time step following the receding horizon principle.

Finally, it should be noted that s^* can be used not only for control purposes, but also for prediction since it includes an estimation of future values for each variable based on historical information.

Weighting and constraint management in BDAC: An important aspect for any high performing controller is prioritization of control objectives and constraint management for both controlled and manipulated variables. These two aspects are of great importance when implementing a controller in a real industrial process, where reaching the setpoint is more important for some variables and a violation of constraints may have catastrophic consequences.

For prioritizing control objectives, a weighting strategy was suggested in Stanley (2018) consisting in normalizing all signals and then weighting them depending on their type: disturbance, manipulated or controlled variables. A method for handling constrains was not proposed.

To account for constraints, a simple yet effective approach is proposed here, which consists in using only a subset of the database $S_{res} \subseteq S$, that contains only those trajectories in S that do not violate any of the constraints when solving the BDAC approximation problem. Since the solution to the BDAC approximation problem, solved with any of the methods introduced above, is an interpolation over a finite set of trajectories, then if none of the trajectories violate any of the constraints, s^* neither does.

Integral Action: Since S (or S_{res}) may not contain information of a given setpoint, it could be the case that s^* cannot regulate the system to that setpoint, yielding a permanent error. To rule out this behavior, a reactive pseudo-integral term is added to the BDAC formulation.

This pseudo-integral action consists in replacing the setpoint SP for controlled variables by $SP - K_i e(k)$, where K_i is the integral constant and $e(k)$ is given by $e(k) = y(k) - SP$ with $y(k)$ being the measured controlled variable at time k. What this integral action does is to move the real setpoint in the opposite direction of the error, trying to compensate it when retrieving signals from S.

3. BDAC OF A SIMULATED PLANT

To test the proposed enhancements to the BDAC formulation, we first apply BDAC to a simulated plant. The well-known quadruple tank system (Johansson, 2000), challenging for controllers because of its nonlinear, MIMO (multi-input multi-output) and coupled nature, is chosen as toy example. Figure 1 shows the quadruple-tank system. The four levels are measured and only the inferior tanks levels (Tank 1 and Tank 2) are controlled by the two pumps.

To construct the database S, the system was excited with a set of steps of different magnitudes and the resulting

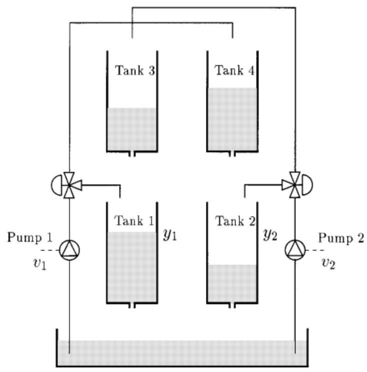


Fig. 1. Quadruple-tank system (Johansson, 2000). The control objective to control Tank 1 and Tank 2 levels using both pumps.

Table 2. Parameters used to control Quadruple-Tank system (W stands for weights).

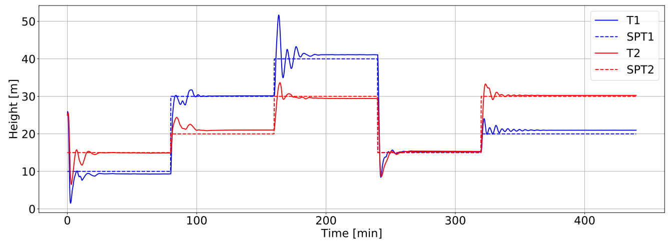
Parameter	Value
n_H	50
d_{th}	20
α	0.98
S_{max}	500
K_i	2
W controlled	2
W manipulated	1

trajectories were saved following the procedure described in the previous section. Once S is constructed, BDAC was applied for setpoint tracking. The parameters used in all the simulations are shown in table 2.

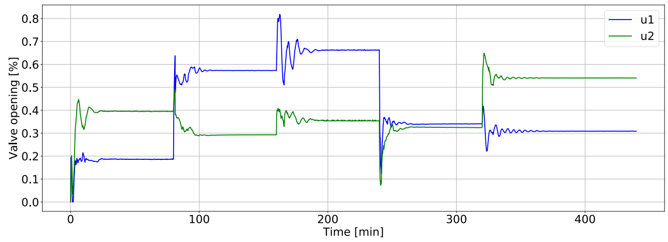
To evaluate performance, the different methods for solving the BDAC approximation problem were compared against using the ISE (integral squared error) and the total energy (TE) indices, defined as

$$ISE = \sum_{t=0}^{T}(r_t - y_t)^2, \quad TE = \sum_{i=0}^{T-1}(u_{t+1} - u_t)^2. \qquad (7)$$

The trajectory used for setpoint tracking is shown in Figure 2 along the closed-loop response of the BDAC controller using local PCA with 30 neighbors. The results of all the tested methods are presented in Table 3. From these results, it can be seen that, in general, when the search of trajectories is restricted to a smaller space (fewer components or fewer neighbors) the behaviour of the controller is more aggressive, having a lower ISE but a higher TE. This trade-off can be adjusted depending on the requirements of the specific application. Another interesting result observed in Table 3 is that methods that use nonlinear kernels have much worse results, this may be due to the fact that the Gaussian kernel is impossible to invert exactly and it has to be estimated numerically, which may lead to poor results.



(a) Controlled variables



(b) Manipulated variables

Fig. 2. Setpoint tracking results using the local PCA technique with 30 neighbors.

Table 3. Setpoint tracking results for the four tanks system.

Method	ISE	TE
Orthogonal decomposition	38947	0.031
IDW	55957	0.672
Min dist	149836	2.044
PCA (all components)	40275	0.027
PCA (200 components)	13495	0.292
Local PCA (100 components, 30 neighbors)	12423	0.577
K. PCA (200 components)	397078	0.571
K. Loc. PCA (100 components, 30 neighbors)	354280	0.304

4. APPLICATION TO AN INDUSTRIAL PASTE THICKENER

To evaluate the performance of the BDAC technique in a real industrial process, we applied it to control an industrial paste thickener in a mineral processing facility.

4.1 Thickening Basics

Thickening is the primary method for producing high density tailings slurries. The most common method generally involves a large thickener tank with a slow turning raking system. Typically, the tailings slurry is added to the tank after the ore extraction process, along with a sedimentation-promoting polymer known as flocculant, which increases the sedimentation rate to produce thickened material discharged as underflow. In this context, the main control objectives are: 1) to stabilize the solids contents in the underflow; 2) to improve the clarity of the overflow water, and 3) to reduce the flocculant consumption (Núñez et al., 2020). In this work the focus is on the control of a paste thickener, a taller type of thickener that produces a discharge with a solids content near 70%. Figure 3 presents the process and instrumentation diagram of the thickener under study.

Regarding operation, it is customary in thickening plants to control the solids content in the discharge and conducting an stabilizing control over the internal states around

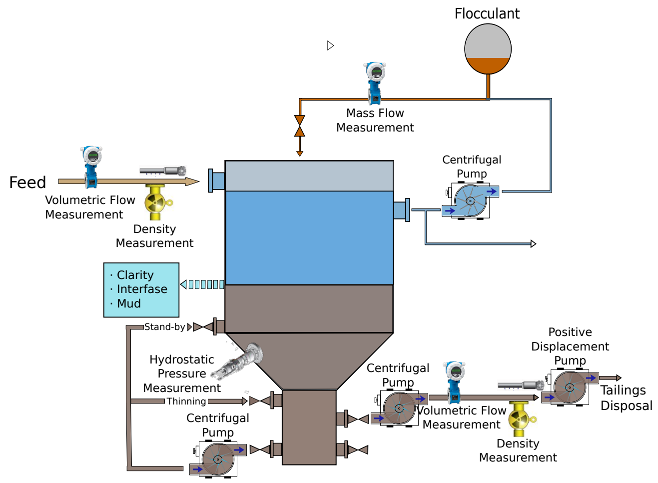


Fig. 3. Process and instrumentation diagram of the thickener under study.

Table 4. Variables used in the first test

Type of variable	Variables
Controlled variables	Output solids concentration
Manipulated variables	Flocculant flow, Output Flow
Measured disturbances	Input Flow, Input solids concentration

Table 5. Variables used in the second test

Type of variable	Variables
Controlled variables	Bed, Rake Torque, Hydrostatic Pressure, Output solids concentration
Manipulated variables	Flocculant flow, Output Flow
Measured disturbances	Input Flow, Input solids concentration

desirable ranges. These internal states are the mud, interface, bed and clarity levels, which are measured online using the "smart diver" sensor, the hydrostatic pressure and the rake torque, which are indicators of the amount of solids inside the thickener. It should be noted that thickening is a slow process, with response times in the order of several hours and in addition to this, is a highly nonlinear multi-input multi-output (MIMO) process, highly autoregressive and subject to multiple disturbances as mineral hardness and granulometric distribution. Hence, traditional mathematical modeling based on differential equations is very difficult and inaccurate. In the same line, synthesizing a first-principles-based controller for automatic regulation is unfeasible given the intricate dynamics and strong disturbances. In this context, data-driven models and controllers, as BDAC, are an appealing option.

4.2 Controller design

Given the control objectives mentioned above, two BDAC configurations were tested in the mining facility for controlling the thickener. The first configuration aims at tracking a setpoint only for the output solids concentration, while the second configuration aims at tracking a setpoint for the output solid concentration and all the internal states. Tables 4 and 5 show the variables and their role in S for both configurations, respectively.

Based on historical data spanning the period from April 2018 to March 2019, yielding around 96.500 training sequences with $T = 5$ minutes, it was determined that

Table 6. Parameters used to control the Thickener (vector's order is consistent with Table 5)

Parameter	Value
n_H	60
d_{th}	80
α	0.99
S_{max}	1500
K_i	[1, 0, 0, 2]
W controlled	[2, 1, 1, 10]
W manipulated	[1, 1]
W perturbations	[0.2, 0.2]
Manipulated V. limits	[[0.2-1.8], [50-180]]
Controlled V. limits	[[3-5], [80-100], [10-30], [65-75]]
Setpoints	[4.2, 89, 20, 70]
BDAC method	Local PCA with 30 Neighbors

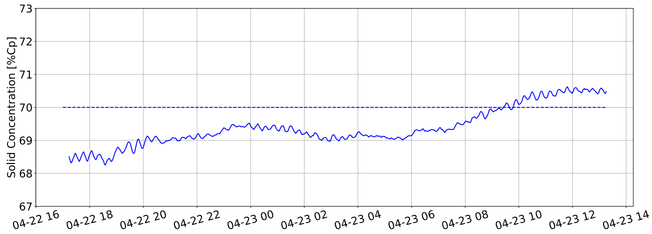
the response time of the system ranges from 4 to 5 hours, therefore, n_H was set to 60 in both configurations. Furthermore, since the thickener is highly nonlinear, local linear approximations of S are expected to perform better than linear approximations of the entire space. Hence, the local PCA method was chosen for solving the BDAC approximation problem in both configurations.

Another important parameter to set precisely is d_{th}, which if set too small forces every trajectory to be introduced in S and reach S_{max} with not necessarily sufficient information, on the other hand, if d_{th} is set to a very large number it forces every trajectory to be filtered with others and novel information is lost. Therefore, to select a good d_{th}, we sampled some trajectories from historical data and set a desired percentage of these trajectories to be stored in S as novel cases. Then by an iterative method we moved d_{th} until the desired number of cases were stored. Table 6 shows the parameters used in the second configuration. For the first configuration, the corresponding subset of parameters was used.

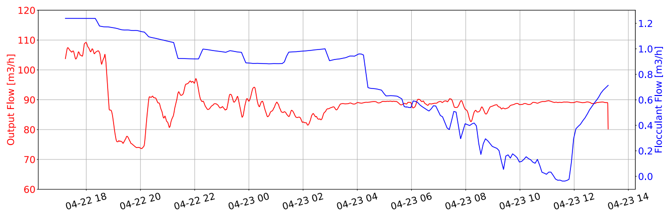
4.3 Results

Test 1 : Figure 4 shows the results obtained for the first configuration, over a 19 hours test. It can be seen that BDAC succeeds in the task of disturbance rejection and setpoint tracking of the output solids concentration, the only controlled variable in this configuration, with a minor permanent error.

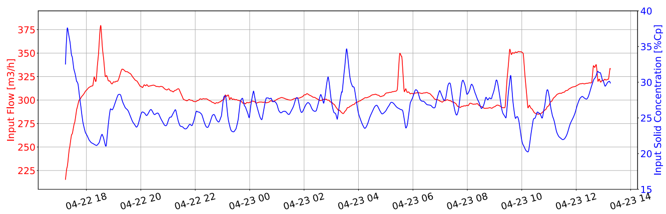
Test 2 : Figure 5 shows the results obtained for second configuration over a 10 hours test. In this case, in addition to the output solid concentration, the rake torque is plotted as well to show how the controller succeeds in controlling secondary variables at their setpoins despite the strong disturbances. It is also interesting to see how the controller is able to manage long delays between manipulated and controlled variables. The flocculant, which mostly affects the rake torque, acts drastically around at 19:00 hours to reduce the torque value and has effect only after almost three hours at 22:00, then the controller stabilizes flocculant at 22:00 having the effect of stabilizing the torque at 02:00. On other hand the output flow, that mostly affects the output solid concentration acts around 18:30 to increase the solid concentration value and then stabilizes around 19:30, stabilizing the controlled variable at 22:00 with only a minor permanent error.



(a) Controlled variables

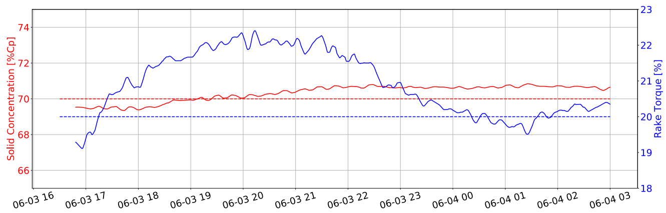


(b) Manipulated variables

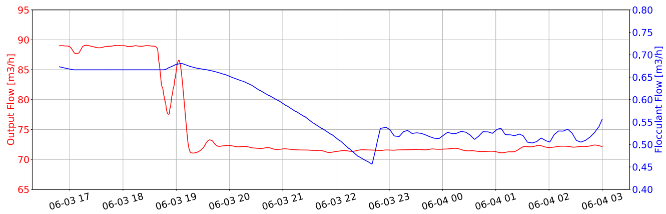


(c) Disturbances

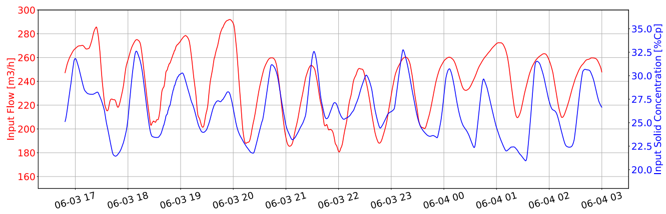
Fig. 4. Setpoint tracking of BDAC controller using the local PCA technique with 30 neighbors.



(a) Controlled variables



(b) Manipulated variables



(c) Disturbances

Fig. 5. Setpoint tracking of BDAC controller using the local PCA technique with 30 neighbors.

5. CONCLUSIONS AND FUTURE WORK

An enhanced formulation of Big Data Approximating Control (BDAC) is presented and evaluated in controlling both a simulated plant and an industrial paste thickener. A first evaluation in a simulated four tanks system indicates that the novel local linear approximation based on PCA outperforms previously proposed methods for solving the BDAC approximation problem. Evaluation in the industrial paste thickener shows that BDAC successfully regulates the system to the setpoint despite the presence of strong disturbances.

Future work includes developing new methods for solving the BDAC approximation problem using advanced nonlinear techniques, such as neural networks.

REFERENCES

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 171, 16 – 25.

Johansson, K.H. (2000). The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8(3), 456–465.

Kwon, W. and Han, S. (2005). *Receding Horizon Control*. Springer, London.

Langarica, S., Rüffelmacher, C., and Núñez, F. (2020). An industrial internet application for real-time fault diagnosis in industrial motors. *IEEE Transactions on Automation Science and Engineering*, 17(1), 284–295.

Leon, S.J., Bjrck, ., and Gander, W. (2013). Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20(3), 492–532.

Liang, D., Xu, F., Chen, H., and Yu, S. (2017). Support vector machine based model predictive control for engine idle speed control. In *2017 Chinese Automation Congress (CAC)*, 2772–2777.

Lu, H., Shu-yuan, W., and Sheng-nan, L. (2017). Study and simulation of permanent magnet synchronous motors based on neuron self-adaptive PID. In *2017 Chinese Automation Congress (CAC)*, 2668–2672.

Mayer-Schonberger, V. and Cukier, K. (2013). *FBig Data: A Revolution That Will Transform How We Live, Work and Think*.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967 – 2986.

Núñez, F., Langarica, S., Díaz, P., Torres, M., and Salas, J.C. (2020). Neural network-based model predictive control of a paste thickener over an industrial internet platform. *IEEE Transactions on Industrial Informatics*, 16(4), 2859–2867.

Qin, S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733 – 764.

Stanley, G. (2018). Big data approximating control (BDAC) a new model-free estimation and control paradigm based on pattern matching and approximation. *Journal of Process Control*, 67, 141 – 159.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1), 37–52.