

Non-parametric Noise Filtering in PID Control Loops using Singular Spectrum Analysis

Emerson Alves da Silva* Leonardo Amaral Mozelli**
Michel Carlo Rodrigues Leles***

* *Graduate Program in Electrical Engineering - Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, MG, Brazil*

** *Department of Electronics Engineering - UFMG*

*** *Department of Technology - Campus Alto Paraopeba - Universidade Federal de São João del-Rei, Ouro Branco, MG, Brazil
(eas2011@ufmg.br, lamoz@ufmg.br, mleles@ufs.edu.br)*

Abstract: PID controllers are still widely used in industrial processes, and their efficient operation requires the design of a filter. This filter makes the derivative action feasible and attenuates the measurement noise, reducing the variability of the control action. However, it influences the closed-loop performance and robustness. In this context, we analyze the effects of introducing a non-parametric noise filter based on the Singular Spectrum Analysis (SSA) technique. The SSA decomposes a signal into a set of additive components, including the measurement noise, in an adaptive manner. The filter was tested in First-Order Time Delay (FOTD) models, typical in industrial processes, for PI and PID controllers, designed by a optimized method, in scenarios with and without measurement noise. The Integral Absolute Error (IAE) metric measured the performance and the Total Variance (TV) the signal variability. The lag-dominated dynamics showed high sensitivity to changes in the filter attenuation degree in comparison with other processes. In contrast, for balanced and delay-dominated dynamics, the filter could improve both TV and performance. The PID achieved better performance than the PI for all processes and scenarios considered, but for higher TV.

Keywords: PID control; Non-Parametric Filtering; Singular Spectrum Analysis; Measurement Noise; FOTD model

1. INTRODUCTION

PID controllers are still the majority of controllers used in the industry, due to their simplicity, few parameters, and cost-effectiveness. According to Åström and Hägglund (2001), they are used in process control, motor drive, optical and magnetic memories, automobiles, flight control, and instrumentation. PID controllers require the design of a low-pass filter to make the derivative action realizable. This filter is also conveniently used to attenuate high-frequency measurement noise, inherent in sensor devices in closed-loop systems.

Excessive noise leads to undesirable deviations in the control action, causing frequent and excessive actuator motion. As a mechanical device, this variation can lead to the wear of its internal components and malfunction. On the other hand, the filter dynamics impact the performance and robustness of the control system. Thus, an attempt to make the system more robust to the measurement noise leads to a conflict: higher noise attenuation versus better performance and robustness.

In this context, this study aims to investigate the effects of using a non-parametric noise filter based on the Singular Spectrum Analysis (SSA) algorithm in a PID control system. This work investigates the the impact of filtering on the closed-loop system, in terms of the performance

and variability of the control action and system output, for different degrees of measurement noise attenuation. For processes with balanced, lag, and delay-dominated dynamics, as well as PI and PID controllers and scenarios with and without measurement noise.

This paper is organized as follows. The noise filtering problem in PID control and the SSA main aspects are reviewed in section 2. Section 3 details the SSA filter, the processes and model approximations, controller structure and synthesis, as well as the performance metrics adopted. The simulation results of the closed-loop system with the non-parametric noise filter are presented in section 4. Finally, the conclusions are given in section 5.

2. LITERATURE REVIEW

A disadvantage of feedback is that measurement noise is introduced into the system (Segovia et al., 2013). Measurement noise is typically high frequency and is assumed to enter the system additively at the process output (Segovia et al., 2014c). It causes high variations in the control signal (Åström and Hägglund, 2006), and, consequently, undesired movements of the actuator, leading to wear and failure. One way to reduce these effects is to use a filter, in this case, a parametric one, characterized by a time constant (Hägglund, 2012; Segovia et al., 2014b).

However, this filter introduces additional dynamics in the control loop (Segovia et al., 2014b). According to Segovia et al. (2014a), the filter changes the system dynamics, and if poorly designed, it can deteriorate robustness and load disturbance rejection. That is one of the reasons why the derivative action is rarely used in industrial process control (Isaksson and Graebe, 2002; Huba, 2015; Šekara and Matausek, 2009).

According to Isaksson and Graebe (2002), the simpler structure of the PI, compared to the PID, which has different configurations (ideal, series, and parallel), is also one of the reasons. In addition, it highlights the lack of tuning methods for PID controllers that also take into account the filter parameters. However, the main reason, according to Isaksson and Graebe (2002); Åström and Hägglund (2006); Kristiansson and Lennartson (2006); Šekara and Matausek (2009), is the fact that the derivative action amplifies the measurement noise, resulting in high variations in the control signal, making the PID more sensitive to it than a PI controller.

Isaksson and Graebe (2002) point out that, although PI can deal satisfactorily with first-order process dynamics, a well-tuned PID controller can achieve significant improvements. As noted in Kristiansson and Lennartson (2006), in which the use of the derivative action significantly improved performance compared to PI, for equal stability margins and only a moderate increase in control activity, for all processes, including those with significant time delay.

In this context, a well-designed filter can even improve the closed-loop performance (Hägglund, 2013). Huba (2015) emphasizes that the success of PID controllers depends heavily on this. This leads to the following compromise: choosing a filter with a given order and structure, seeking to attenuate the measurement noise without impacting too much the robustness and performance of the system (Segovia et al., 2013).

The SSA is a technique for time series analysis and prediction that includes classic elements of series analysis, statistics, multivariate geometry, dynamic systems, and signal processing (Golyandina et al., 2001; Alexandrov, 2009). It decomposes the original time series in terms of its internal components, which are additive and interpretable, such as trend, periodical components, and noise (Golyandina and Zhigljavsky, 2013).

This technique can be applied in trend analysis, detection, and extraction of quasi-periodic components, noise attenuation, and point change detection (Alexandrov, 2009). According to Hassani (2007), other applications include identification of trends and patterns with different resolutions and complexity, smoothing, extraction of seasonal components, cycles, and periodic signals with different amplitudes.

It does not require any statistical assumptions to be fulfilled for the time series or its components (Golyandina et al., 2001). It is not necessary also a parametric model that characterizes its statistical distribution, nor non-stationarity conditions (Zhigljavsky, 2010; Hassani, 2007). It is simple to use, requires few parameters to be adjusted,

and allows trend extraction in the presence of noise and oscillating components (Alexandrov, 2009).

As Alexandrov (2009) points out, it is similar to the Principal Component Analysis (PCA) technique, which is applied to a multidimensional series, decomposed by the Singular Value Decomposition (SVD) method, and returns a new representation, with a lower dimension and different physical meaning. The SSA, nonetheless, preserves its dimension and physical meaning throughout the decomposition process.

Although relatively new, it is a powerful technique being applied to many practical problems in different areas of science (Hassani and Thomakos, 2010). Concerning control process, in Zhang and Wang (2016) attitude regulation is performed jointly by Active Disturbance Rejection Control and SSA.

3. METHODS

The block diagram in Fig. 1 was considered to analyze the effects of the non-parametric noise filter based on the SSA method in the performance of the closed-loop system.

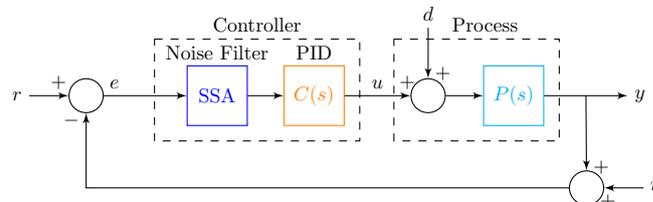


Figure 1. Closed-loop system investigated. The blocks SSA, $C(s)$, and $P(s)$ are, respectively, the noise filter, the ideal PID controller, and the process. The signals are: r , reference input, e , error, u , control action, d , disturbance, y , system output, and η , noise.

It consists of a SSA noise filter in series with an ideal PID controller, that aims to manipulate the process $P(s)$, subjected to a disturbance d , by comparing the measured value of the output y , mixed with noise η , to its set-point r . The design of each block in this system is explained in the following sections.

3.1 Non-parametric Filter SSA

The non-parametric SSA filter, in its original form, is applied to a real-valued time series X_N of length N :

$$X_N = (x_0, x_1, \dots, x_n, \dots, x_{N-2}, x_{N-1}) \quad (1)$$

The filtering process consists of four steps, divided in two stages. The first is the Decomposition stage, which comprises the Embedding and SVD steps. The second is the Reconstruction stage, which performs the Grouping and the Diagonal Averaging steps. These steps are described as follows.

1. Decomposition

1.1. Embedding: This step maps the original 1-dimensional time series, given in (1), into a K -dimensional series \mathbf{X}_m , called the trajectory matrix, shown in (2).

$$\mathbf{X}_m = [\mathbf{X}_1, \dots, \mathbf{X}_K], \quad \mathbf{X}_k = [x_{k-1}, \dots, x_{k+L-2}]^T \quad (2)$$

This trajectory matrix is made of a set of K column vectors of length L that are subsets of the original

time series, with $K = N - L + 1$. It is a Hankel matrix, given that the elements in its anti-diagonals are the same. The filter parameter $L \in \mathbb{Z}$ is called the window length or the embedding dimension and is constrained to $2 \leq L \leq N/2$. It represents the maximum number of components into which the original time series is decomposed.

1.2. Singular Value Decomposition:

The second step aims to decompose the trajectory matrix in terms of its singular vectors $\mathbf{U} \in \mathbb{R}^{L \times L}$ and $\mathbf{V} \in \mathbb{R}^{K \times K}$, and singular values $\mathbf{\Sigma} \in \mathbb{R}^{L \times K}$, by the use of the SVD method, resulting in:

$$\mathbf{X}_m = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3)$$

where the singular vectors and values are in a decreasing order of magnitude: $\sigma_{11} \geq \sigma_{22} \geq \dots \geq \sigma_{LK}$. From the representation in (3), the trajectory matrix is then splitted into $q = \max\{i|\sigma_{ii} > 0\} \leq L$ elementary matrices, as illustrated in:

$$\mathbf{X}_m = \sum_{i=1}^{q \leq L} \sigma_i U_i V_i^T \quad (4)$$

resulting in a set of matrices $X_{\mathcal{D}}$ that represents the trajectory matrix of each component:

$$X_{\mathcal{D}} = \{\mathbf{X}_{m_1}, \mathbf{X}_{m_2}, \dots, \mathbf{X}_{m_q}\}, \quad \mathbf{X}_{m_i} \in \mathbb{R}^{L \times K} \quad (5)$$

1.3. Eigendecomposition:

An alternative to the SVD method is the eigendecomposition of the covariance matrix, given in (6).

$$\mathbf{C}_m = (\mathbf{X}_m \mathbf{X}_m^T) / K, \quad \mathbf{C}_m \in \mathbb{R}^{L \times L} \quad (6)$$

which results in a representation of the original covariance matrix in terms of its eigenvalues $\mathbf{S} \in \mathbb{R}^{L \times L}$ and eigenvectors $\mathbf{U} \in \mathbb{R}^{L \times L}$, as shown in:

$$\mathbf{C}_m = \mathbf{U}\mathbf{S}\mathbf{U}^T \quad (7)$$

From this representation, the original trajectory matrix can be decomposed into $q = \max\{i|\lambda_{ii} > 0\} \leq L$ matrices, by the projection of each column in \mathbf{X}_m onto the respective eigenvector in \mathbf{U} , as shown in:

$$\mathbf{X}_m = \sum_{i=1}^{q \leq L} U_i (X_i^T U_i)^T = \sum_{i=1}^{q \leq L} U_i X_{pc_i}^T \quad (8)$$

where $\mathbf{X}_{pc_i} \in \mathbb{R}^{K \times L}$ are the time series principal components. This process results in the same set of decomposed matrices $X_{\mathcal{D}}$ as the one obtained in (5).

2. Reconstruction

2.1. Grouping:

In this step, the set of matrices $X_{\mathcal{D}}$, from the decomposition stage, are merged into disjoint groups. Defining $\mathcal{I} = \{I_1, I_2, \dots, I_i, \dots, I_p\}$ such that a given set of indices $I_i = \{i_1, i_2, \dots, i_{p_i}\}$ corresponds to the matrix indices that belongs to the i -th group. Then, the grouping step is performed as in (9), by summing up all the matrices that belongs to that group.

$$\mathbf{X}_{I_i} = \sum_{i \in I_i} \mathbf{X}_{m_i} = \mathbf{X}_{m_{i_1}} + \mathbf{X}_{m_{i_2}} + \dots + \mathbf{X}_{m_{i_{p_i}}} \quad (9)$$

Extending to all sets of indices, results in (10), a set of grouped matrices.

$$\mathbf{X}_{\mathcal{I}} = \{\mathbf{X}_{I_1}, \mathbf{X}_{I_2}, \dots, \mathbf{X}_{I_p}\} \quad (10)$$

This step relies on prior knowledge about the separability of the components, the magnitude of the

eigenvalues, the existence of patterns between the eigenvectors, among others. Roughly speaking, at this step the filtering effectively occurs, by dismissing some matrices.

2.2. Diagonal Averaging:

This step seeks to map each matrix resulting from the grouping step, back into a 1-dimensional time series, \tilde{X}_N , with the same length N as the original one.

$$\tilde{X}_N^{(k)} = (\tilde{x}_0^{(k)}, \tilde{x}_1^{(k)}, \dots, \tilde{x}_n^{(k)}, \dots, \tilde{x}_{N-2}^{(k)}, \tilde{x}_{N-1}^{(k)}) \quad (11)$$

Let $\mathbf{X}_{I_i} = \mathbf{A} = (a_{ij})_{i,j=1}^{L,K}$ represent one of the matrices obtained from the grouping step. The reconstructed time series in (11) is obtained by performing an average operation on each anti-diagonal of \mathbf{A} . This operation is shown in (12), and it is repeated for $n = 0, 1, \dots, N - 1$.

$$\tilde{x}_n^{(k)} = \begin{cases} \frac{1}{n+1} \sum_{i=1}^{n+1} a_{ij}^*, & 0 \leq n < L^* - 1 \\ \frac{1}{L^*} \sum_{i=1}^{L^*} a_{ij}^*, & L^* - 1 \leq n < K^* \\ \frac{1}{N-n} \sum_{i=n-K^*+1}^{N-K^*+1} a_{ij}^*, & K^* \leq n < N \end{cases} \quad (12)$$

where $j' = n - i + 2$, $L^* = \min(L, K)$, $K^* = \max(L, K)$, $N = L + K - 1$, $a_{ij}^* = a_{ij}$ for $L < K$, and $a_{ij}^* = a_{ji}$ for $L \geq K$.

Extending this procedure for each component $k = 1, 2, \dots, q \leq L$, the original time series is approximated by a reconstructed one, \tilde{X}_N^R , as shown in (13).

$$\tilde{X}_N^R = \sum_{k=1}^R \tilde{X}_N^{(k)} \quad (13)$$

where R is a parameter that controls the number of components used to approximate the time series, with $1 \leq R \leq L$. If $R = L$, then $\tilde{X}_N^R = X_N$, the reconstructed series is equal to the original one, if $R < L$, then \tilde{X}_N^R is an approximation of X_N considering the first R most important components.

The SSA filtering steps are summarized in the pseudo-algorithm shown in 1.

Algorithm 1: SSA Algorithm

Input: $X_N \in \mathbb{R}^N$: original time series
 $L \leq N/2 \in \mathbb{Z}$: embedding dimension
 $R \leq L \in \mathbb{Z}$: components for reconstruction

Output: $\tilde{X}_N^R \in \mathbb{R}^N$: reconstructed time series

Function SSA(X_N, L, R):

$\mathbf{X}_m \leftarrow$ Embedding(X_N, L) # Trajectory matrix
 $X_{\mathcal{D}} \leftarrow$ Decomposition(\mathbf{X}_m) # SVD
 $X_{\mathcal{I}} \leftarrow$ Grouping($X_{\mathcal{D}}, \mathcal{I}$)
 $\tilde{X}_N^R \leftarrow$ DiagonalAveraging($X_{\mathcal{I}}, R$)

end

The SSA filter requires all the samples of the time series to be available when applying the method. In order to use it in a real-time application, a modified version of it,

proposed in Leles et al. (2017, p. 4), called Causal Singular Spectrum Analysis (CSSA), was adopted.

The method collects samples from a continuous time signal $x_n = x(nT_s)$ with T_s the sampling interval, to form a time series X_N of fixed length N . After that, as new samples arrive, the old ones are discarded, as in a sliding window.

Besides, for each incoming sample, x_n , instead of performing the diagonal averaging on each component's matrices, its reconstruction, \tilde{x}_n , is done directly from the updated eigenvectors U of the decomposition step, according to (14), which performs the same operation as (8).

$$\tilde{x}_n^{(k)} = \sum_{i=1}^L u_L^{(k)} u_i^{(k)} x_{n-L+i} \quad (14) \quad \tilde{x}_n^R = \sum_{k=1}^R \tilde{x}_n^{(k)} \quad (15)$$

where $u^{(k)}$ are the eigenvectors of the k -th component.

The reconstructed sample considering then the first R components of the time series is obtained by (15). The pseudo-algorithm of this filter is presented in 2.

Algorithm 2: CSSA Algorithm

Input: $x_n \in \mathbb{R}$: sample of the original time series
 $N \in \mathbb{Z}$: minimum number of samples
 $L \leq N/2 \in \mathbb{Z}$: embedding dimension
 $R \leq L \in \mathbb{Z}$: components for reconstruction

Output: $\tilde{x}_n^R \in \mathbb{R}$: reconstructed sample

Function CausalSSA(x_n, N, L, R):

```

n ← 0, XN ← 0
while new sample to process do
  if n < N then
    XN ← AddSample(XN, xns)
     $\tilde{x}_n^R \leftarrow x_n$ 
  else if n == N then
     $\tilde{X}_N^R \leftarrow \text{SSA}(X_N, L, R)$  # SSA filter
  else if n > N then
    XN ← UpdateTimeSeries(XN, xn)
    Xm ← Embedding(XN, L)
    U ← Eigendecomposition((XmXmT)/K)
     $\tilde{x}_n^R \leftarrow \text{Reconstruction}(X_N, U)$  # (14)-(15)
  end
  n ← n + 1
end
end

```

In the CSSA algorithm, n and X_N represent global variables (maintain its state between function calls) that are initialized once. The filter executes as long as new samples arrive. For $n < N$, the filter is inactive. When $n = N$, the minimum number of samples has been reached, then the SSA filter is executed on X_N . After that, for $n > N$, the series is updated with the new sample, the eigendecomposition is performed on the new covariance matrix and the operations in (14) and (15) are executed to compute the reconstructed sample.

3.2 Process Dynamics and Model

The processes $P(s)$ used in Fig. 1 were based on Segovia et al. (2013, 2014b). They were selected to cover a significant range of dynamics encountered in process control.

They were: (i) lag-dominated; (ii) balanced, and (iii) delay-dominated dynamics. Its Transfer Functions (TFs) are given in (16), (17) and (18), respectively.

$$P_1(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)} \quad (16)$$

$$P_2(s) = \frac{1}{(s+1)^4} \quad (17) \quad P_3(s) = \frac{e^{-s}}{\left(\frac{5}{100}s+1\right)^2} \quad (18)$$

For the controller design, the processes were approximated by the First-Order Time Delay (FOTD) model in (19).

$$G(s) = \frac{k_p e^{-\tau_d s}}{\tau s + 1} \quad (19)$$

where k_p , τ_d and τ are the steady-state gain, the dead-time, and the time constant, respectively.

These model parameters were determined by the reaction curve of each process to a step input. The gain k_p were calculated by the step and process output steady-state gains. The time delay τ_d was obtained through the intersection of the tangent line at the inflection point of the response curve with the time axis, and the time constant was given by $\tau = t_{63.2} - \tau_d$, where $t_{63.2}$ is the instant when the response reaches $\approx 63.2\%$ of its steady-state value.

3.3 PID Controller and Synthesis

The Proportional-Integral (PI) and PID controllers shown in the block diagram in Fig. 1 are given by the TFs in (20) and (21).

$$C_{PI}(s) = k_c \left(1 + \frac{1}{T_i s} \right) = k_c + \frac{k_i}{s} \quad (20)$$

$$C_{PID}(s) = k_c \left(1 + \frac{1}{T_i s} + T_d s \right) = k_c + \frac{k_i}{s} + k_d s \quad (21)$$

where k_c , $k_i = k_c/T_i$, and $k_d = k_c T_d$ are the controller parameters. Which were set by the Approximated M-constrained Integral Gain Optimization (AMIGO) tuning rules, proposed in Segovia et al. (2013) and reproduced in Tab. 1.

Table 1. AMIGO tuning rules for PI and PID controllers for process modeled by a FOTD.

	PI	PID
k_c	$\frac{0.15}{k_p} + \frac{\tau}{k_p \tau_d} \left(0.35 - \frac{\tau_d \tau}{(\tau_d + \tau)^2} \right)$	$\frac{1}{k_p} \left(0.2 + 0.45 \frac{\tau}{\tau_d} \right)$
T_i	$0.35 \tau_d + \frac{13 \tau_d \tau^2}{\tau^2 + 12 \tau_d \tau + 7 \tau_d^2}$	$\tau_d \frac{0.4 \tau_d + 0.8 \tau}{\tau_d + 0.1 \tau}$
T_d	-	$\frac{0.5 \tau_d \tau}{0.3 \tau_d + \tau}$

3.4 Performance Measurement

As in Segovia et al. (2014b), the Integral Absolute Error (IAE) metric in (22) was chosen to measure the performance of the controller in the closed-loop system of Fig. 1.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (22)$$

where e is the error due to a unit step input.

To measure the degree of the measurement noise present in the control signal u and the system output y , the Total

Variance (TV) metric was chosen. This metric was defined in Huba (2015) and is given in (23) and (24).

$$TV(u) = \sum_k |u(k+1) - u(k)| - |2u_m - u_\infty - u_0| \quad (23)$$

$$TV(y) = \sum_k |y(k+1) - y(k)| - |2y_m - y_\infty - y_0| \quad (24)$$

where k is the sample index, $y_m \notin (y_0, y_\infty)$ and $u_m \notin (u_0, u_\infty)$ are extreme values of y and u , y_0, u_0, y_∞ and u_∞ the initial and steady-state values of y and u , respectively.

4. RESULTS AND DISCUSSION

The simulations of the system in Fig. 1 aimed to investigate the behavior of the closed-loop system performance, measured by the IAE, and the amount of noise present in the signal of interest, measured by the TV, as the values of L and R were changed, considering both the presence and absence of measurement noise. The error signal was decomposed into a fixed number of components, constant value of L , while its reconstruction was made with a different number of components, varying R for each L .

The PI and PID controller parameters used in the simulations are shown in Tab. 3. Which were obtained applying the AMIGO tuning rules in Tab. 1 on the FOTD model parameters of the processes, presented in Tab. 2, for a sensitivity of $M = 1.4$.

Table 2. FOTD model approximation parameters for each process.

	P_1	P_2	P_3
k_p	1	1	1
τ_d	0.08	1.43	1.01
τ	1.04	2.92	0.09

Table 3. PI and PID controller parameters for each process from the AMIGO tuning rules.

	P_1		P_2		P_3	
	PI	PID	PI	PID	PI	PID
k_c	4.13	6.44	0.41	1.1	0.18	0.24
k_i	7.67	17.86	0.16	0.47	0.47	0.51
k_d	0	0.24	0	0.69	0	0.03

The simulation lasted 15s, with sampling period of $T_s = 0.01$ s, for a unit step reference input r applied at $t = 5$ s, with no disturbance, $d = 0$. The measurement noise η added at the process output y was generated from a Gaussian distribution with zero mean and Signal-to-Noise Ratio (SNR) of 0.01.

For the CSSA, the non-parametric noise filter, the size of sliding window for the time series was $N = 200$ samples. The values of L and R for each scenario were: (i) $L = [15, 25, 35, \dots, 85, 95]$, and R was varied from 1 to 15 components, for each L ; and (ii) $R = [1, 2, \dots, 10]$, and L was varied from 10 to 100, for each R .

These values of L and R satisfy the filter constraints, $2 \leq L \leq N/2$ and $1 \leq R \leq L$, and cover a wide range of L , given the choice of N . The combination of L and R results in different noise attenuation degrees. Since the reconstruction with a few but significant components dispose the high-frequency ones, that are less significant in spectral terms, usually where the noisy components are located.

4.1 Lag-dominated dynamics

The results for P_1 , lag-dominated dynamics, are shown in Figs. 2 and 3. Each point in Fig. 2 represents the value of the IAE and TV of the control action u and system output y , obtained from the simulation of the closed-loop system for a given combination of L and R . Where $R_i, i = 1, 2, \dots, 15$, means that the error signal e was reconstructed with the first i components.

The results in Fig. 2, in the absence of measurement noise, as the attenuation degree increases, by increasing L and reducing R , both the performance and the signal variability deteriorates, i.e. larger IAE and TV, of u and y , for both PI and PID controllers. The best result is achieved when the filter is not activated, which happens for $L = R = 15$. This suggests that, in the absence of noise, the filter is discarding important information present in the signal when disregarding some of its components.

Meanwhile, in the presence of measurement noise, the higher the attenuation degree, the lower is the TV of u and y , as expected, since the suppression of the noise results in less signal variability. But at the expense of worse performance, higher IAE, for both controllers. At some point, as can be seen for the $TV(y)$, greater noise attenuation results in an increase of the TV as well. Moreover, the PID controller performed better than the PI, for most simulations, but for higher values of the TV.

Some of these results can be seen by the time behavior of the control action u and system output y , for the case where noise is added to the output and for some values of L and R , as shown in Fig. 3.

The worsening of the system's performance can be seen clearly for $R = 1$ as L increases from 15 to 95, equivalent to increasing the degree of noise attenuation. It deteriorates to the point that the system is no longer stable, for the PI controller, and marginally stable for the PID, see $L = 95$ and $R = 1$. Also from Fig. 3, is clearly seen the greater variability of the control action for the PID compared to the PI controller. Which reflects on the TV metric and is caused by the amplifying effect of high-frequency components by the derivative part of the controller.

4.2 Balanced dynamics

The simulation results for process P_2 , of balanced dynamics, are presented in Fig. 5. For better viewing, some results are enlarged. For this process, in the absence of measurement noise and for any value of L and R , the TV of the output y were negligible, as can be seen from the magnitude values on the x -axis. In contrast, they were significant for the control action u and show a larger gap between the PI and PID controllers results.

Moreover, in the absence of measurement noise, the filter deteriorates both the performance and the TV, even for small values of L , as can be seen from the points labeled with R_1 . As the degree of noise attenuation is decreased, by making R closer to L , the performance converges to ≈ 4.8 (PI) and ≈ 2.25 (PID), while the TV continues to decrease.

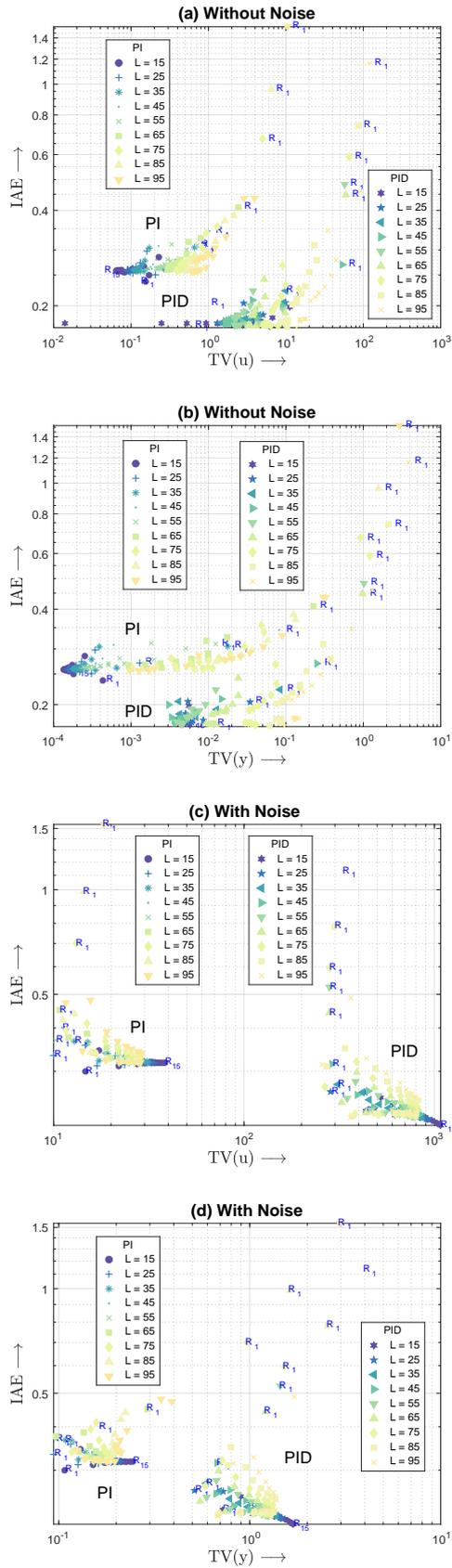


Figure 2. IAE versus TV of u and y for lag-dominated dynamics, for a unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

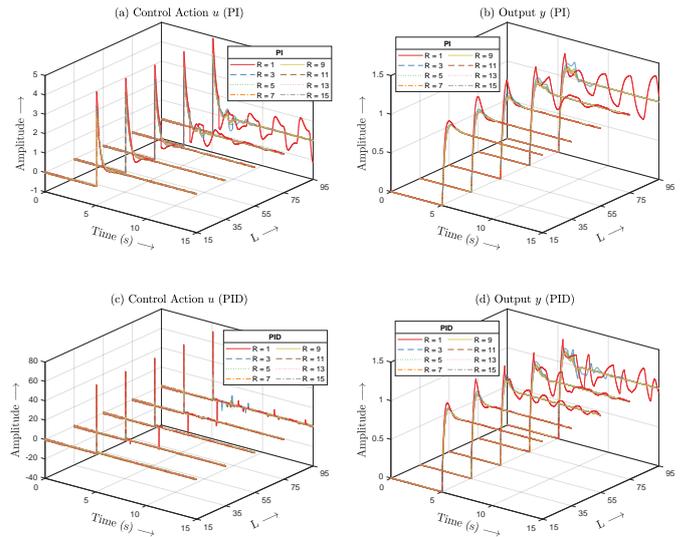


Figure 3. Closed-loop performance of the lag-dominated dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

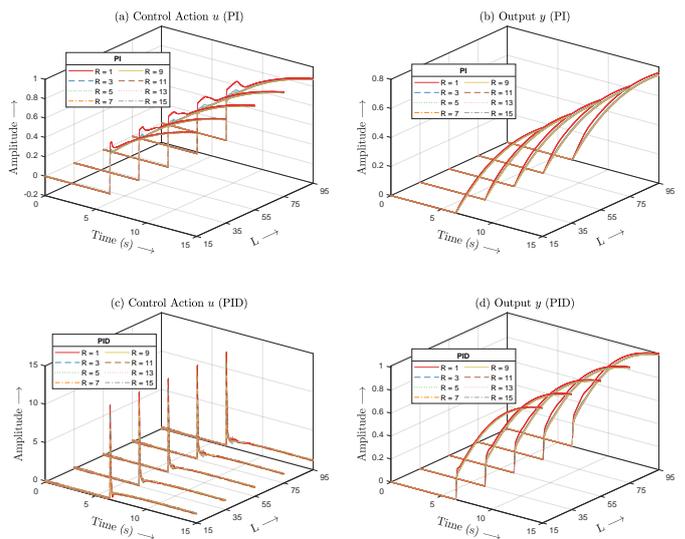


Figure 4. Closed-loop performance of the balanced dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

In the presence of measurement noise, the pattern changes. Although the gap between PI and PID remains large, now, as the attenuation degree increases, moving from right to left, the results split into two curves, one directed upwards and the other downwards, for both the control action u and output y . This suggests that for this type of dynamics, some values of L and R can achieve, simultaneously, better performance and greater noise attenuation.

From the time behavior of the system output, y in Fig. 4, the PI response is slower than that of the PID, which explains its smaller IAE. Besides, as L and R vary, for this type of process, there are no major differences between the response curves, except for $R = 1$ for the control

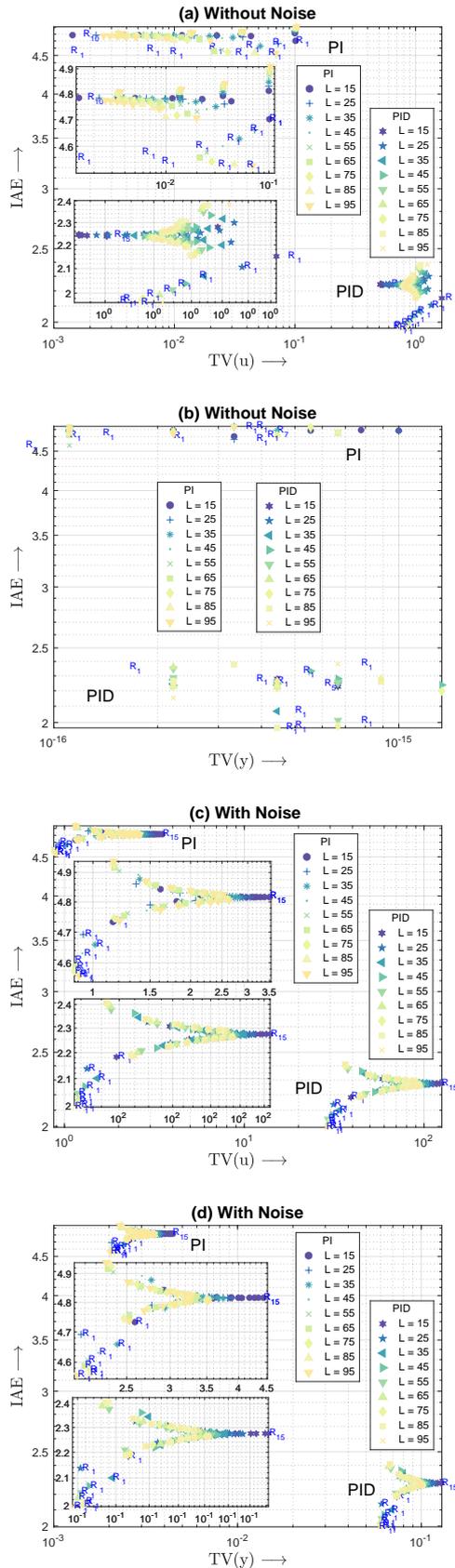


Figure 5. IAE versus TV of u and y for balanced dynamics, for a unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

action and of the PI controller. In addition, the magnitude of the control action for the PID was smaller than that of the process P_1 . It is also worth mentioning that no combination of L and R values resulted in an unstable closed-loop system.

4.3 Delay-dominated dynamics

The simulation results for the process P_3 , of delay-dominated dynamics, are shown in Fig. 6. The results for the absence of measurement noise were not significant and therefore are not presented. Meanwhile, in the presence of measurement noise, the patterns are similar to those obtained for process P_2 , but with PI and PID performances closer. As the noise attenuation degree decreases, the two curves converge to ≈ 2.1 (PI) and ≈ 1.95 (PID). Also, the IAE values obtained are much lower than those of process P_2 and slightly higher than of P_1 .

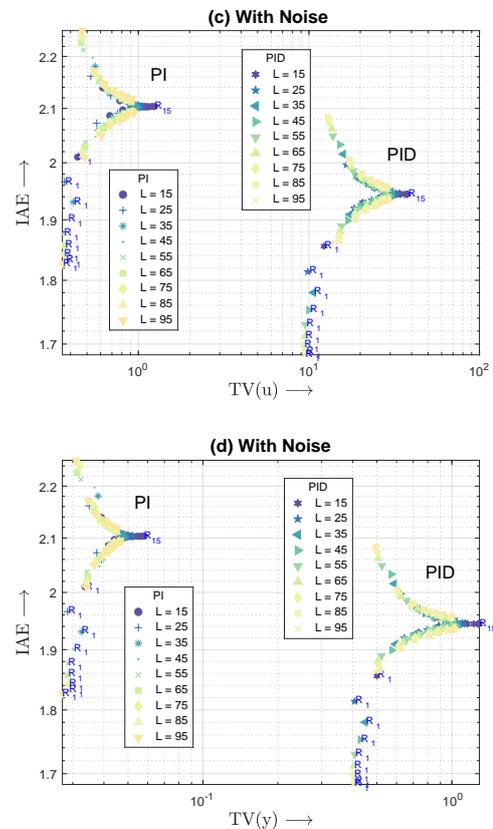


Figure 6. IAE versus TV of u and y for delay-dominated dynamics, for a unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

The time behavior for the system output y in Fig. 7 shows a similar performance for PI and PID. Likewise process P_2 , no combinations of L and R destabilized the closed-loop system.

5. CONCLUSION

This paper investigated the effects of using a non-parametric noise filter based on the SSA method in a Proportional-Integral-Derivative (PID) control loop. The simulations consisted of varying the filter attenuation degree for processes with balanced, lag, and delay-dominated

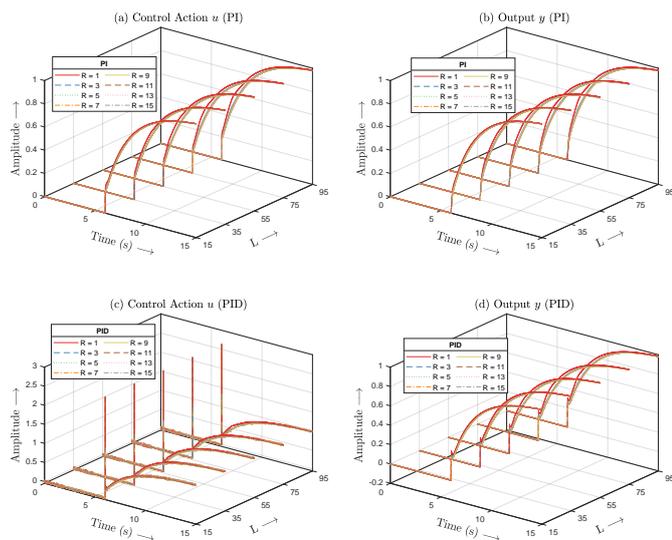


Figure 7. Closed-loop performance of the delay-dominated dynamics, for a unit step input, in the presence of measurement noise with $\text{SNR} = 0.01$, $N = 200$, R varying from 1 to 15 for each L , and $T_s = 0.01$ s.

dynamics, for PI and PID controllers, both in the presence and absence of measurement noise. The trade-off between performance and measurement noise attenuation was analyzed through the IAE and TV of the control action and system output.

In the absence of measurement noise, no filtering achieved the best performances for both PI and PID controllers, regardless of the process dynamics. Since there is no noise, the SSA filters relevant components of the error signal, causing the control action and output performance to deteriorate, with an unnecessary increase in variability. Meanwhile, the trade-off between performance and noise attenuation was noticeable in the presence of measurement noise. For the lag-dominated dynamics, a higher attenuation degree resulted in less signal variability, to a certain extent, at the expense of worse performance. On the other hand, for the balanced and delay-dominated dynamics, for some values of L and R , a higher attenuation degree simultaneously improved the performance and the TV.

In addition, the PID achieved better performance than the PI across all processes, independent of the presence or absence of measurement noise. The largest discrepancy between PI and PID was obtained for the balanced dynamics, followed by the delay and lag-dominated dynamics. Furthermore, the balanced dynamics process showed greater sensitivity to changes in the filter attenuation degree. Resulting in an unstable closed-loop system for the PI and marginally stable for the PID in the extreme case. In contrast, the delay and lag-dominated dynamics showed robustness to changes in L and R , resulting in stable systems with similar response curves.

The SSA is an adaptable filter that uses the time series' spectral composition to identify its components. It is more general than a low-pass filter since it can identify and extract oscillatory elements and trends. The drawbacks are the computational cost due to the SVD step and the need to adjust the parameters N , L , and R for each problem.

Suggestions for future work include simulations for a more diverse set of dynamics and the use of other tuning methods. Given the plausibility of SSA as a noise filter in PID loops, an open avenue of research is to consider an automatic algorithm for selection in the grouping step, which can render optimal/sub-optimal trade-offs.

ACKNOWLEDGMENTS

This work has been supported by the Brazilian agencies CAPES and CNPq.

REFERENCES

- Alexandrov, T. (2009). A method of trend extraction using singular spectrum analysis. *Revstat-Statistical Journal*, 7(1), 1–22.
- Åström, K.J. and Hägglund (2006). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society, Research Triangle.
- Åström, K.J. and Hägglund, T. (2001). The future of pid control. *Control engineering practice*, 9(11), 1163–1175.
- Golyandina, N., Nekrutkin, V., and Zhigljavsky, A.A. (2001). *Analysis of time series structure: SSA and related techniques*. Chapman and Hall/CRC, Boca Raton.
- Golyandina, N. and Zhigljavsky, A. (2013). Ssa for forecasting, interpolation, filtration and estimation. In *Singular Spectrum Analysis for Time Series*, 71–119. Springer.
- Hägglund, T. (2012). Signal filtering in pid control. *IFAC Proceedings Volumes*, 45(3), 1–10.
- Hägglund, T. (2013). A unified discussion on signal filtering in pid control. *Control engineering practice*, 21(8), 994–1006.
- Hassani, H. (2007). Singular spectrum analysis: Methodology and comparison. *Journal of Data Science*, 5, 239–257.
- Hassani, H. and Thomakos, D. (2010). A review on singular spectrum analysis for economic and financial time series. *Statistics and its Interface*, 3(3), 377–397.
- Huba, M. (2015). Filter choice for an effective measurement noise attenuation in pi and pid controllers. In *2015 IEEE International Conference on Mechatronics (ICM)*, 46–51. IEEE.
- Isaksson, A. and Graebe, S. (2002). Derivative filter is an integral part of pid design. *IEE Proceedings-Control Theory and Applications*, 149(1), 41–45.
- Kristiansson, B. and Lennartson, B. (2006). Evaluation and simple tuning of pid controllers with high-frequency robustness. *Journal of Process Control*, 16(2), 91–102.
- Leles, M.C.R., Mozelli, L.A., and Guimarães, H.N. (2017). A new trend-following indicator: Using ssa to design trading rules. *Fluctuation and Noise Letters*, 16(02), 1750016.
- Segovia, V.R., Hägglund, T., and Åström, K.J. (2014a). Measurement noise filtering for common pid tuning rules. *Control Engineering Practice*, 32, 43–63.
- Segovia, V.R., Hägglund, T., and Åström, K.J. (2014b). Measurement noise filtering for pid controllers. *Journal of Process Control*, 24(4), 299–313.
- Segovia, V.R., Hägglund, T., and Åström, K.J. (2014c). Design of measurement noise filters for pid control. *IFAC Proceedings Volumes*, 47(3), 8359–8364.
- Segovia, V.R., Hägglund, T., and Åström, K.J. (2013). Noise filtering in pi and pid control. In *2013 American Control Conference*, 1763–1770. IEEE.
- Šekara, T.B. and Matausek, M.R. (2009). Optimization of pid controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise. *IEEE Transactions on Automatic Control*, 54(1), 184–189.
- Zhang, H. and Wang, Z. (2016). Attitude control and sloshing suppression for liquid-filled spacecraft in the presence of sinusoidal disturbance. *Journal of Sound and Vibration*, 383, 64–75.
- Zhigljavsky, A. (2010). Singular spectrum analysis for time series: Introduction to this special issue. *Statistics and its Interface*, 3(3), 255–258.