

Article

An Interface to Monitor Process Variability Using the Binomial ATTRIVAR SS Control Chart

João Pedro Costa Violante ¹, Marcela A. G. Machado ^{1,*}, Amanda dos Santos Mendes ² and Túlio S. Almeida ¹

¹ Department of Production Engineering, São Paulo State University, Guaratinguetá 05508-070, Brazil; jp.violante@unesp.br (J.P.C.V.); tulio.almeida@unesp.br (T.S.A.)

² Department of Administration and Public Administration, Fluminense Federal University, Volta Redonda 27213-145, Brazil; amanda_mendes@id.uff.br

* Correspondence: marcela.freitas@unesp.br

Abstract: Control charts are tools of paramount importance in statistical process control. They are broadly applied in monitoring processes and improving quality, as they allow the detection of special causes of variation with a significant level of accuracy. Furthermore, there are several strategies able to be employed in different contexts, all of which offer their own advantages. Therefore, this study focuses on monitoring the variability in univariate processes through variance using the Binomial version of the ATTRIVAR Same Sample S^2 (B-ATTRIVAR SS S^2) control chart, given that it allows coupling attribute and variable inspections (ATTRIVAR means attribute + variable), i.e., taking advantage of the cost-effectiveness of the former and the wealth of information and greater performance of the latter. Its Binomial version was used for such a purpose, since inspections are made using two attributes, and the Same Sample was used due to being submitted to both the attribute and variable stages of inspection. A computational application was developed in the R language using the Shiny package so as to create an interface to facilitate its application and use in the quality control of the production processes. Its application enables users to input process parameters and generate the B-ATTRIVAR SS control chart for monitoring the process variability with variance. By comparing the data obtained from its application with a simpler code, its performance was validated, given that its results exhibited striking similarity.

Keywords: control chart; ATTRIVAR; variability; variance; application; interface; R; Shiny package



Citation: Violante, J.P.C.; Machado, M.A.G.; Mendes, A.d.S.; Almeida, T.S. An Interface to Monitor Process Variability Using the Binomial ATTRIVAR SS Control Chart. *Algorithms* **2024**, *17*, 216. <https://doi.org/10.3390/a17050216>

Academic Editors: Zixin Huang, Li Jin, Xiongbo Wan and Sheng Du

Received: 5 April 2024
Revised: 28 April 2024
Accepted: 28 April 2024
Published: 16 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Control charts were initially introduced by [1] and are the primary and most technically sophisticated tools of statistical process control [2]. They are widely employed in process monitoring and serve to detect nonconformities and special causes of variation, thus enabling the early identification of disturbances aimed to minimize negative adverse financial impacts [3].

Among control charts, some utilize variable inspection for monitoring continuous quality characteristics, while others employ attribute inspection, which was initially designed for monitoring non-continuous quality characteristics.

One of which is grounded on the numerical measurements of quality characteristics, and another is based on the attributes defined by [4] as quality characteristics measured in a nominal scale or categorized according to a predetermined scheme of labels. For instance, classifying fruits as being good or rotten, and nails or screws as defective or not defective.

Although the primary purpose of attribute inspection was outlined about 70 years ago, researchers have made contributions by suggesting their use for controlling continuous quality characteristics, and more recent studies have proposed better means for such [5].

There are numerous comparisons between variable and attribute charts in the literature, such as [5–10]. These authors emphasize the superior efficiency and informativeness

of variable inspection due to its reliance on measurements, as opposed to a simpler, faster, and cheaper application of attribute inspection.

Several researchers explore the use of attribute inspections for continuous characteristics, such as [8,9,11], in an attempt to continuously seek to improve the efficiency of resource utilization and improve the performance of control charts.

Ergo, what once meant having to use a sample 6.667 times larger in attribute charts to achieve the same level of performance as variable charts [2] now requires a sample less than twice as large using the np_{S^2} proposed by [9].

The strategy of using variable and attribute inspections combined into a single chart had initially been proposed by [12] and is similar to one which was later on called as ATTRIVAR (ATTRibute + VARiable) by [5]. Thenceforth, these mixed charts combine the advantages of both forms of inspection initially called as such.

A version of [12] consisted in designing the np_x (attribute) and \bar{X} (variable) charts separately and dividing the collected sample into two subsamples. Then, the former was submitted to np_x chart evaluation, and the latter was submitted to \bar{X} chart evaluation if the former had been rejected.

Ref. [7] also proposed a mixed chart without subdividing the sample and used the same sample in both stages of inspection (as in [13,14], and to the proposal herein). Ref. [5] proposed ATTRIVAR-1 chart, which uses the same sample in both stages, in addition to ATTRIVAR-2 chart, which makes use of different samples. In this work, the terminologies SS (same sample) and DS (different sample) refer to these strategies, respectively.

Unlike these proposals, which addressed only mean monitoring, Ref. [10] proposed a mixed chart to monitor variability through variance. The authors coupled the np_{S^2} chart from [9] in the stage of attribute inspection with the S^2 chart in the stage of variable inspection using different samples.

Afterwards, Ref. [15] proposed the Trinomial version of the ATTRIVAR (T-ATTRIVAR) chart aimed at mean monitoring but using three attributes in the first inspection stage instead of two as in other strategies.

Thus, this paper explores the Binomial version of ATTRIVAR Same Sample S^2 (B-ATTRIVAR SS S^2) chart that makes use of the first inspection stage with two attributes ("binomial") using the same sample in both inspection stages ("same sample") to monitor the process variance (S^2).

Although there is a certain number of works on control charts coupling attribute and variable inspections in the literature, research on monitoring the variability of univariate processes through variance, i.e., B-ATTRIVAR SS S^2 , is still lacking. The novelty of this study lies in proposing an interface capable of receiving input data from a user thereof, to obtain control limits through simulations and generate a (B-ATTRIVAR SS S^2) chart so as to monitor the variability of a univariate process.

The control chart strategy proposed herein is similar, particularly to those from: Ref. [9], as both make use of attribute inspections to monitor the process variance, Ref. [10], given that the attribute and variable data were also used to monitor process variance, and [15], who proposed the most similar strategy but used it to monitor process mean using three attributes in the first monitoring stage (instead of two as in this study).

Even though in-control process parameters are usually unknown in practice and require estimates using historical data, this paper focuses on monitoring processes where these parameters are known [16].

In addition to advances in statistical process control strategies, an integration of computer interfaces is also crucial to increase the manufacturing system's efficiency. As discussed by [17], the convergence between digital models and physical industrial environments using data is of paramount relevance to smart manufacturing. This is due to the fact that it can provide interaction and information exchange between software, the computational interfaces, and the physical systems of manufacturing processes.

In this context, the application of computational interfaces offers new possibilities for monitoring and controlling industrial processes in real time. Although this work mainly

focuses on introducing an interface developed using R and Shiny to monitor process variability, it is also worth mentioning the potential of integrating it with industrial digital systems, such as manufacturing execution systems (MES). Despite not being directly discussed in this study, the role of MES in facilitating quality monitoring and process optimization is also worth being mentioned. The authors of [18] explain that MES can provide a useful platform in quality monitoring, as it allows direct data acquisition and the monitoring of quality parameters' variability. Ref. [19], for instance, explores the possibilities of using control charts to analyze the data stored in MES and provide feedback to the systems aiming to optimize production parameters.

This manuscript is organized as follows: Section 2 describes the B-ATTRIVAR SS S^2 chart and draws a comparison with more usual charts; Section 3 explains how its application was conceived and the manner in which the results have been achieved; Section 4 presents its interface and operation, as well as a validation of its results; and Section 5 draws its conclusion and proposes suggestions for future works.

2. Chart B-ATTRIVAR SS S^2

After sample collection, all n items of the B-ATTRIVAR SS S^2 chart are inspected based on attribute, similarly to the np_{S^2} from [9]. These inspections make use of some device (such as a go/no-go gauge, for example) configured using the discriminant control limits LDL (lower discriminant limit) and UDL (upper discriminant limit). Each item outside these limits is classified as nonconforming. After classifying all items in the sample, Y_D is recorded as the number of rejected items. If $Y_D \geq CL_Y$ (attribute control limit), the process is defined as being out of control. Conversely, it is assessed whether $Y_D < W_Y$ (warning attribute control limit). If so, it is found that the process is in control. If otherwise, the variable inspection process is initiated using the same sample.

The stage of variable inspection is performed similarly to that in the classic S^2 chart described by [2,4], in which quality is measured based on all n items in the sample, and variance S^2 is calculated. If S^2 is outside the acceptance interval defined by LCL (variable lower control limit) and UCL (variable upper control limit), the process is defined as being in control. Otherwise, it is out of control. This entire process is depicted in Figure 1.

It is worth mentioning that it is common to use only the upper control limit to inspect variable S^2 , as a process should have the least possible variability, as mentioned by [20], given that this strategy makes the chart more effective at detecting increases in process variance. In such a case, UCL is denoted as CL_{S^2} .

In this paper, both the attribute and variable inspection limits are optimized to achieve acceptable performance values determined by ARL_0 and ARL_1 , or by α and β , considering that ARL refers to the average run length or the average number of samples collected until the chart signals for the first time, in addition to the fact that [4]:

$$ARL = \frac{1}{(\text{Probability of sample rejection by the chart})} \quad (1)$$

$$ARL_0 = \frac{1}{\alpha} \quad (2)$$

$$ARL_1 = \frac{1}{(1 - \beta)} \quad (3)$$

where α is the probability of type I error, β is the probability of type II error, the index "0" indicates a process in control, and index "1" indicates a process being out of control.

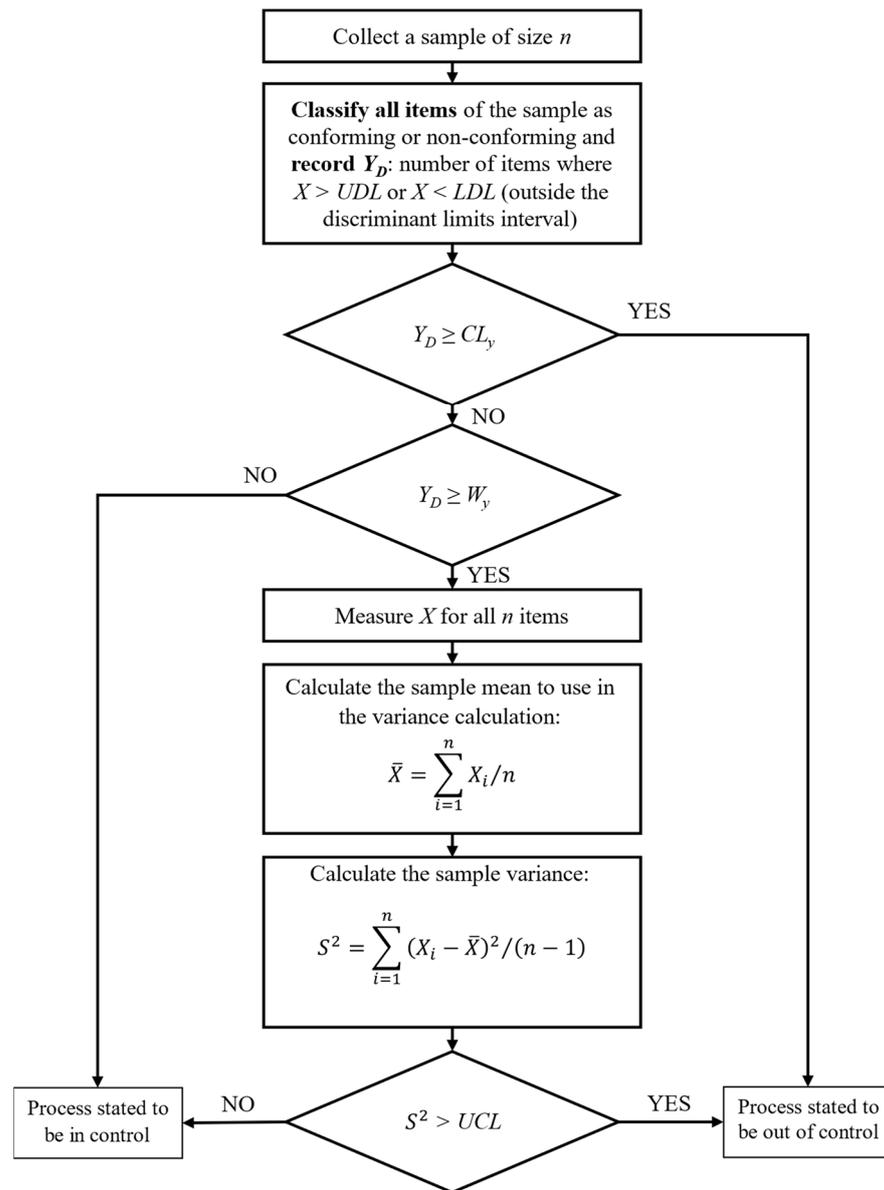


Figure 1. Flowchart of the B-ATTRIVAR SS S^2 Chart.

The performance of the B-ATTRIVAR SS S^2 chart (mixed inspection) was compared to that of S^2 (variable inspection) and np (attribute inspection) charts, as in Table 1, which shows simulated ARL values with sample sizes of 5 ($n = 5$) by varying the standard deviation of simulated samples through λ in each row. The results demonstrate a similar performance of the studied chart to that of the np chart. Nevertheless, it is still worth mentioning its advantages over the np charts, mainly on account of the fact that the quality characteristic measurements required in the stage of variable inspection of the B-ATTRIVAR SS S^2 are capable of providing valuable information to enhance the process analysis and make improvements. Although the performance of the ATTRIVAR proved inferior to that of the S^2 chart, it still offers notable advantage over it, i.e., its operational simplicity [14].

Table 1. Comparison of the performance of charts for monitoring S^2 with $n = 5$.

λ (Lambda)	S^2	ARL np	B-ATTRIVAR SS S^2
1.0	367.28	371.71	370.71
1.1	107.93	118.87	120.12
1.2	42.79	50.26	51.54
1.3	21.21	26.72	26.31
1.4	12.32	15.61	15.48
1.5	8.10	10.19	10.22
1.6	5.72	7.18	7.33
1.7	4.33	5.44	5.51
1.8	3.52	4.37	4.34
1.9	2.92	3.54	3.56
2.0	2.51	3.00	3.04

3. Application Development

The application was developed using the RStudio development environment, which is free and offers various packages and functionalities available for installation using a computational programming based on the R language. According to [21], R is ranked among the 10 most popular programming languages used globally and has become a fundamental computational tool for research in various areas, such as statistics, mathematics, physics, chemistry, medicine, among others.

Additionally, the Shiny package was used for interface development, which, according to [21], was launched in 2012 and has continuously gained popularity in developing interactive websites using R language functionalities.

The average run length (ARL) was used for measuring the efficiency or performance of control charts. Ref. [22] considers it as the best-known and most widely used method to measure and analyze control chart performance.

3.1. Inputs and Outputs Definition

Regarding the development of a B-ATTRIVAR SS control chart application aimed to monitor the variability of univariate processes through variance, the code should be capable of generating control limits for the attribute (W_Y and CL_Y) and variable (LCL and UCL , or just CL_{S^2} if $LCL \leq 0$) inspections. Based on these four outputs, the application can effectively perform the primary function of a control chart: either accepting or rejecting samples to classify the process as being in control or out of control.

For such a purpose, the initial necessary inputs would be sample size (n), mean (μ_0), and standard deviation (σ_0) of the process, given that they are essential parameters for the program to perform normal sample simulations.

The user would have to provide the maximum desired probability of a type I error (α_{max}) and the maximum desired probability of a type II error (β_{max}) for a given variation (λ) in standard deviation to obtain the attribute and variable control limits. Thus, α_{max} would define the min ARL_0 and β_{max} for a given λ , which would define the max ARL_1 . Then, it would be possible to determine the acceptance intervals, as ARL_0 and ARL_1 are directly related to the control limits.

To start the first stage of monitoring with attribute inspection, The user would have to determine the discriminant limits LDL and UDL , which were also defined as outputs. Thus, these values could be configured using a go/no-go gauge device, for instance. Afterwards, they could conduct inspections and input the number of nonconforming items in each sample into the application to proceed to the next stage.

Furthermore, the user would have to provide the maximum percentage of samples to be submitted to variable inspection ($\%S^2_{max}$) to define the discriminant limits (LDL and UDL).

3.2. Obtaining Control Limits

The process of obtaining control limits was expected to be performed through simulations and trial-and-error iterations. Its logical is that the program makes the first iteration using lower values for the control limits, and it comes to %S², α or β in each iteration. Then, the program verifies whether these values are lower than the maximum values established by the user, and it changes the limits and proceeds to the next iteration if otherwise.

To enhance code organization, it was segmented into four sections, namely, A, B, C1, and C2. Each having a beginning and an end, and their own iterations able to calculate and record the results in the form of a matrix.

In section A, it explores the possible combinations of W_Y and CL_Y to derive the smallest discriminant limits (LDL and UDL) satisfying the %S²_{max} condition:

$$\%S^2_{Simulated} = \frac{NV}{TN} \leq \%S^2_{max} \tag{4}$$

where NV is the number of samples, which would be submitted to variable inspection, and TN is the total number of simulated samples.

For such a purpose, it simulates one million samples from a normal distribution based on user-defined parameters (*n*, *μ*, and *σ*) for in-control conditions, and calculates how many would undergo variable inspection, i.e., $W_Y \leq Y_D \leq CL_Y$. Then, it performs the test described in Equation (4). This entire process repeats until the condition is satisfied.

In section B, it accesses each combination of W_Y and CL_Y, but using the values of LDL and UDL for each, and proceeds to obtain the smallest variable control limits (LCL and UCL) meeting the requirement of α_{max}:

$$\frac{1}{ARL_0} \leq \alpha_{max} \tag{5}$$

For such, it simulates samples under control and submits them to attribute inspections and to variable inspections whenever necessary, until it reaches ten thousand signals. Then, it records the number of simulated samples until each signal's emission, thus allowing it to calculate ARL₀ by computing the arithmetic mean of these ten thousand recorded numbers. Afterwards, it assesses the condition of Equation (5) and repeats this until it is satisfied by incrementing the upper limit and decrementing the lower limit during each iteration, as shown in the Equations (6) and (7). For each combination of W_Y, CL_Y, LDL, and UDL, it assigns the following values to limits by ever incrementing the absolute value of parameter *L*:

$$LCL = [\sigma^2 - L\sigma^2] \tag{6}$$

$$UCL = [\sigma^2 + L\sigma^2] \tag{7}$$

Within section C1, it already has recorded the various combinations of W_Y, CL_Y, LDL, UDL, LCL, and UCL, and it assesses whether each of them meets the β_{max} requirement set by the user:

$$1 - \frac{1}{ARL_1} \leq \beta_{max} \tag{8}$$

For such, it simulates normal samples, albeit out of control, by altering only the standard deviation to σ₁ = λσ (where λ and σ are defined by the user). These samples undergo the entire B-ATTRIVAR SS S² chart flow using the parameters from each obtained combination. Consequently, ARL₁ is calculated for each combination (similarly to ARL₀) and it is tested whether Equation (8) is satisfied. This process is carried out only once for each parameter combination, and it records which combinations can meet this requirement.

In section C2, it accesses combinations meeting the β_{max} requirement and selects the best solution as the one with the lowest calculated β.

Definite results can be seen by the user and are used by the program to perform other steps and ensure the functioning of the control chart itself. The displayed parameters are: n , μ , σ , W_Y , CL_Y , LDL , UDL , $\%S^2$, LCL , UCL , α , λ , and β .

4. Results

Given the objective of developing an interface to enable the utilization of the B-ATTRIVAR SS S^2 chart, it is necessary to initially understand the expected usage dynamics through which the application was conceived.

At first, the application user is expected to input the initial parameters required for obtaining the control limits (as described earlier). Once the program finishes obtaining them, the user should be able to visualize a chart showing horizontal lines representing the control limits and should input the Y_D of their sample, which can be plotted on the chart. If Y_D is between W_Y and CL_Y , something users can verify on the chart, they should input the quality characteristic measurements for all items in the sample. Therefore, the program should calculate and plot sample variance on the chart, and the signal if it somewhat indicates that the process is out of control.

4.1. The Interface

The main layout of its interface was configured as depicted in Figure 2. In the area marked with the number one, there is a sidebar with fields where the user can input values or keep the predefined values in the program (shown in Figure 2). Below this sidebar, in the area marked with the number two, there is the RUN button which the user should click on to execute the code and obtain the results. Adjacent to this, in the area marked with the number three, there are the two tabs of the application, between which the user can switch by clicking on one of them. At the bottom, the area marked with the number four represents where the results of calculations and the chart itself are displayed in the RESULTS AND CHART tab; and the editable table to insert sample data in the INSERT DATA tab.

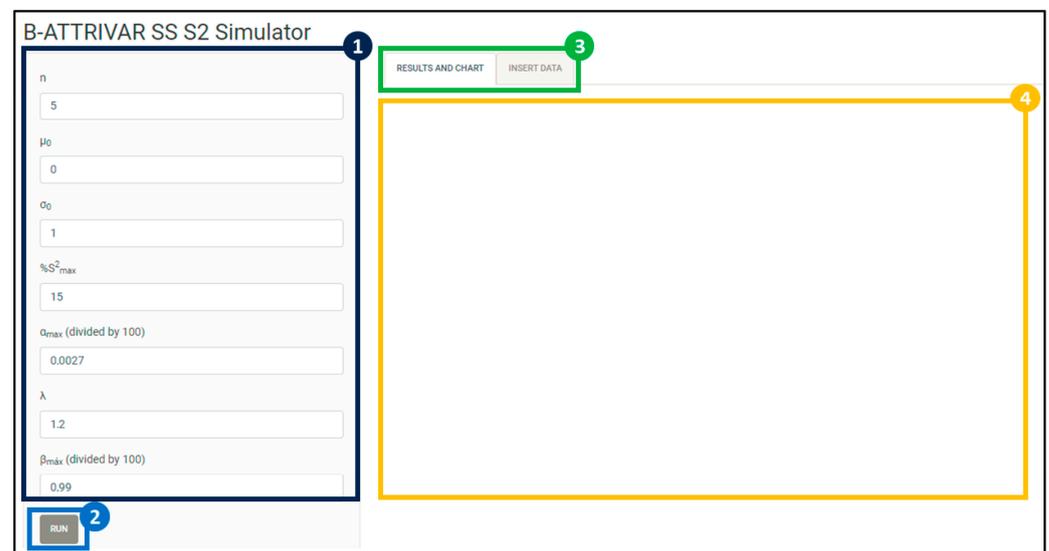


Figure 2. Main layout of the developed interface.

After pressing the RUN button, the end results and the chart are displayed in the RESULTS AND CHART tab, as shown in Figure 3.

Some parameters entered by the user (n , μ_0 , σ_0 e λ) are shown below the tab names and above the graph, along with the results found for other parameters through simulations and calculations (W_Y , CL_Y , LDI , LDS , $\%S^2$, LCL , UCL , α , β).

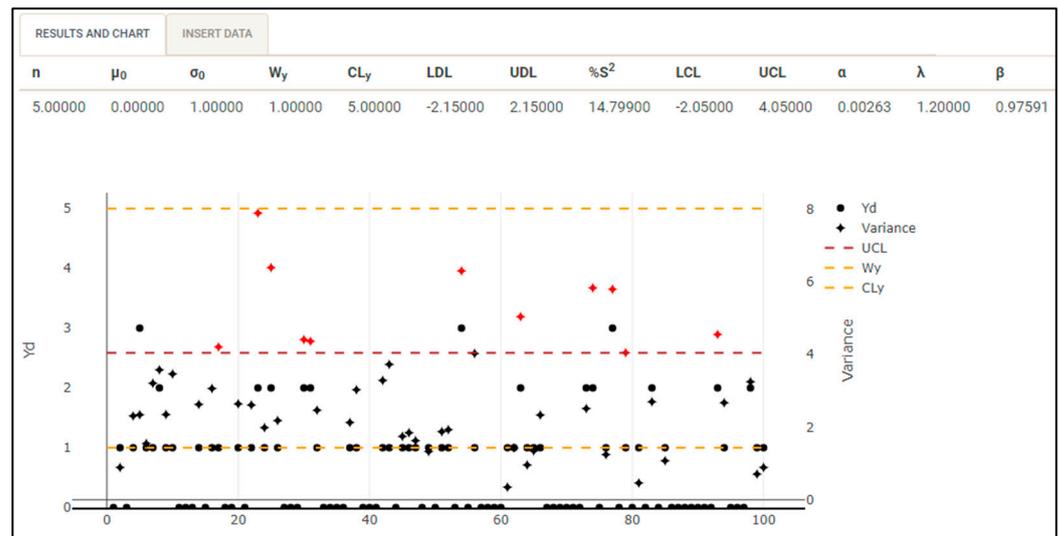


Figure 3. Results and chart displayed in the interface.

4.2. The Chart

Bearing a resemblance to the charts presented by [13–15], Figure 3 shows an x-axis representing the sample number, and two y-axes: one for Y_D on the left side and another for sample variance on the right side.

The chart displays the limits of W_Y and CL_Y on the Y_D axis, and UCL on the variance axis, all represented as dashed lines. The circular points represent the Y_D data series, while the differently shaped points represent the calculated sample variance data series. It is crucial to highlight that the points representing sample variance will be plotted only if $W_y \leq Y_D < CL_y$, due to the fact that samples satisfying $Y_D < W_y$ are accepted and those within $Y_D \geq CL_y$ are rejected in the B-ATTRIVAR SS chart, and thus do not need to be submitted to variable inspection in both cases.

Points are standardly plotted in black. However, they are shown in red to represent signals (when samples are rejected for instance) to display if they occur in the attribute inspection (circular points) or variable inspection (non-circular points).

The chart is rather dynamic, as any changes in the editable table data lead to automatic adjustments in the plotted points on the chart. Additionally, users can press and hold the mouse button on either of the two abscissa axes to drag them upwards or downwards, thus moving both their labels and associated points. Furthermore, users can zoom in and out by pressing and holding the mouse button over any point in the plot area, then dragging it to form a rectangle defining the area to be zoomed.

4.3. The Editable Table

The data plotted on the chart are from the editable table illustrated in Figure 4 and located in the INSERT DATA tab. It contains some columns dependent on the user-inputted n value, although the first and last columns always represent Y_D and the calculated variance of each sample, respectively, regardless of the inputted n value. Other columns between the first and the last ones are intended for inputting quality characteristic measurements, and thus depend on the sample size ($n = 5$ in the case of Figure 4, and therefore there are 5 columns for measurements).

All the columns are editable, except for the last one. Although the user can attempt to change its value, the table automatically recalculates it, since the variance values cannot be arbitrary, but are rather calculated from the quality characteristic measurements.

Yd	X1	X2	X3	X4	X5	S2
0	-0.8953456742028633	-0.4461941560425836	-0.6245747507318891	-1.179058397682084	0.7601485157026023	0.5552907134352891
1	2.404150423227579	1.353620018689438	2.016358394461732	0.2052821720304423	0.5049039702948307	0.8914320570455969
0	-0.4253181916568534	0.08347554522386486	0.5043969876402552	-0.3445119768716455	-0.9816668263607279	0.3134584836014375
1	0.3234663374612783	1.792112920234225	-2.423649859302055	0.2235440255939575	-0.07925427570630209	2.310316542238614
3	-3.101570026720071	-2.418643787760713	-0.9118016340442792	0.6616733997124288	-2.539398528784959	2.344238876263868

Figure 4. Editable table on the interface.

Before users start monitoring any process, the application offers a preview of its functionality. It generates data for 100 out-of-control samples from a normal distribution, then it submits them to inspections with the definite results of the control limits and plots them on the chart. After clicking on the RUN button at the sidebar, once the results are obtained, the application automatically displays data about the 100 samples in both the table and the chart.

The editable table standardly displays only five rows, but just below the Editable Table title, the number of rows can be changed by clicking on Show 5 entries and altering them to the desired number.

Below the table, there are the DELETE DATA and RECALCULATE Y_D s buttons. All the values recorded in the table are deleted by pressing the first button, leaving all fields empty for new data input and, consequently, erasing all the plotted points on the chart. The second button recalculates all Y_D s in the table's first column based on the values of the monitored quality measurements.

Naturally, as previously explained, according to the logic of monitoring a process through an ATTRIVAR chart, the user can start inspections via attribute without conducting any measurements, which is one of the main advantages of this strategy compared to variable control to avoid excessive measurements. These can often be expensive and/or excessively time-consuming; thus, it would be irrelevant to calculate Y_D through measurements.

Moreover, it would also be unreasonable if the user altered any measurement data and if Y_D remained the same, even though the number of nonconforming items might have changed. Furthermore, automatically calculating this column would be inappropriate as well, since it would not allow the user to input Y_D regardless of the measurements, which would contradict the inherent logic of the ATTRIVAR strategy.

Therefore, the Y_D fields are editable so that the user can input the number of nonconforming items for each sample. Additionally, a button is provided for the user to click upon at any time to recalculate all Y_D s in the table based on the measurements.

4.4. Results Validation

Theoretical calculations of the parameters of this chart have not been identified. Therefore, a simpler code was developed to simulate the ARLs of the B-ATTRIVAR $SS S^2$ chart aiming to validate the application's results and ensure the consistency of simulations and control limit calculations. Control limits are set in it, and it calculates ARLs for different values of λ .

Hence, to validate the application results, the control limits generated by the application were utilized as input parameters for the code. Subsequently, the results obtained from both methods were compared in terms of $\%S^2$, ARL_0 , and ARL_1 . The reasons for selecting these parameters are as follows:

- Analyzing whether $\%S^2$ verifies if the chart submits the expected percentage of samples in the stage of variable inspection;
- Analyzing whether ARL_0 verifies if the chart commits to the type I error after the expected number of tested samples is calculated;
- And analyzing whether ARL_1 verifies if the chart detects a signal at the expected speed, or after the expected number of tested samples is calculated.

As shown in Table 2, several scenarios were tested, all being $\mu_0 = 0$ and $\sigma_0 = 1$, and the columns labeled as “reference” correspond to the values obtained through the simpler code.

Table 2. Results comparison and validation ($\mu_0 = 0, \sigma_0 = 1$).

<i>n</i>	<i>W_y</i>	<i>CL_y</i>	$\%S^2_{max}$	UDL	UCL	$\%S^2$	$\%S^2$ Reference	ARL_0	NMA ₀ Reference	Δ ARL ₀	λ	ARL_1	ARL ₁ Reference	Δ ARL ₁
5	1	3	15%	2.15	4.09	14.81%	14.81%	379.221	368.596	2.88%	1.2	41.9332	41.8994	0.08%
5	1	4	15%	2.15	4.07	14.82%	14.81%	385.416	383.675	0.45%	1.2	42.0345	43.2341	2.77%
5	1	5	15%	2.15	4.07	14.80%	14.80%	392.364	375.318	4.54%	1.2	42.8538	42.4327	0.99%
5	2	4	15%	1.47	4.50	14.74%	14.74%	377.595	362.952	4.03%	1.2	48.7047	49.3499	1.31%
5	2	5	15%	1.47	4.03	14.95%	14.93%	377.669	370.022	2.07%	1.2	43.0950	42.1248	2.30%
5	1	3	10%	2.31	4.06	9.99%	10.01%	374.161	373.848	0.08%	1.2	41.8840	42.1619	0.66%
5	1	4	10%	2.32	4.05	9.75%	9.73%	378.737	371.406	1.97%	1.2	42.7068	42.6622	0.10%
5	1	5	10%	2.32	4.06	9.74%	9.75%	375.552	378.126	0.68%	1.2	43.2554	42.3157	2.22%
5	2	4	10%	1.59	4.12	9.84%	9.84%	376.320	372.540	1.01%	1.2	44.0206	43.6339	0.89%
5	2	5	10%	1.59	4.01	9.96%	9.93%	378.703	367.040	3.18%	1.2	44.0304	43.0009	2.39%
6	1	3	10%	2.38	3.63	9.94%	9.94%	388.136	367.356	5.66%	1.5	6.4295	6.4782	0.75%
6	1	4	10%	2.38	3.63	9.96%	9.93%	386.227	367.750	5.02%	1.5	6.5638	6.5563	0.11%
6	1	5	10%	2.38	3.64	9.97%	9.95%	394.030	376.513	4.65%	1.5	6.6395	6.6466	0.11%
6	1	6	10%	2.38	3.64	9.96%	9.94%	387.306	379.715	2.00%	1.5	6.5732	6.6439	1.06%
6	2	4	10%	1.68	3.74	9.94%	9.98%	377.044	371.606	1.46%	1.5	6.8422	6.7470	1.41%
6	2	5	10%	1.69	3.62	9.70%	9.72%	391.299	377.701	3.60%	1.5	6.8597	7.0055	2.08%
6	2	6	10%	1.69	3.61	9.72%	9.70%	388.899	376.739	3.23%	1.5	6.9539	6.8811	1.06%
6	3	5	10%	1.28	3.89	9.77%	9.79%	383.502	368.495	4.07%	1.5	8.0894	8.1119	0.28%
6	3	6	10%	1.28	3.54	9.94%	9.93%	380.937	365.824	4.13%	1.5	7.1517	7.0933	0.82%

The first five rows on the table represent the execution of simulations with default values set on the application interface, as in Figure 2. In the next five rows, the application was executed only by changing the $\%S^2_{max}$ parameter from 15% to 10%. The next nine rows contained changes not only in $\%S^2_{max}$ but also in *n* and λ . This approach enables the analysis of whether alterations in one or more parameters affect the validity of its results.

Its results demonstrate a strong consistency with those obtained through the reference code, i.e., $\%S^2$, ARL_0 , and ARL_1 show notably similar values. For instance, the largest percentage differences observed in the ΔARL_0 and ΔARL_1 columns were 5.66% and 2.77%, respectively.

4.5. Example

To exemplify how the proposed application functions, the code was run using the default values for inputs (the same values set in the interface fields shown in Figure 3). After having calculated control limits as described in Section 3.2, the application is ready to be used for process monitoring.

In this example, the simulations converged to the following values for control limits: $W_Y = 1$; $CL_Y = 3$; and $UCL = 4.07$.

Although the application was designed with the purpose of allowing the user to start monitoring via the attribute inspections, the following example is a simulation. Therefore, as shown in Table 3, 25 random (out of control) samples of measurements were generated and Y_D s were calculated from them. For samples where Y_D is between W_Y and CL_Y limits, the variances were calculated (S^2).

Table 3. Example data.

	X_1	X_2	X_3	X_4	X_5	Y_D	S^2
1	-0.068	-1.419	0.512	1.599	-1.242	0	-
2	-2.434	0.416	-0.550	-0.157	2.045	1	2.628
3	-0.595	-2.908	-2.326	-0.192	-0.627	2	1.453
4	0.964	2.562	-0.595	-0.552	0.193	1	1.716
5	-1.033	-0.648	-0.040	-0.714	1.571	0	-
6	-0.604	0.862	1.348	0.560	1.623	0	-
7	0.362	0.478	2.527	2.959	1.228	2	1.399
8	0.250	-3.543	-1.195	0.386	-0.528	1	2.546
9	-0.399	0.227	-0.220	2.194	1.783	1	1.420
10	1.010	-0.340	0.551	-0.488	1.335	0	-
11	1.177	0.091	1.738	-1.572	0.261	0	-
12	-2.305	0.427	-0.083	0.396	-0.686	1	1.279
13	-1.511	2.279	-1.349	-0.571	-0.322	1	2.323
14	-1.945	-0.888	1.219	1.238	1.557	0	-
15	-2.502	-0.879	-1.618	-0.047	1.013	1	1.857
16	0.327	0.213	0.075	-0.544	-2.475	1	1.356
17	1.332	-1.233	-1.886	-2.049	0.597	0	-
18	0.577	-0.307	0.463	0.723	-0.726	0	-
19	0.554	-0.112	2.413	0.950	0.743	1	0.865
20	-0.954	2.335	-1.239	-0.935	-0.466	1	2.168
21	-2.057	-1.459	-1.045	0.483	0.569	0	-
22	-0.364	-1.582	0.092	-0.601	0.154	0	-
23	-3.362	1.516	0.590	-2.660	0.043	2	4.506
24	-1.749	-1.797	0.928	-0.830	-0.913	0	-
25	-2.969	-1.621	-2.386	-1.518	0.492	2	1.719

Figure 5 shows how the interface plots data from Table 3. As aforementioned, the Y_D s of all samples are plotted as circular points, and S^2 are plotted as non-circular points whenever necessary.

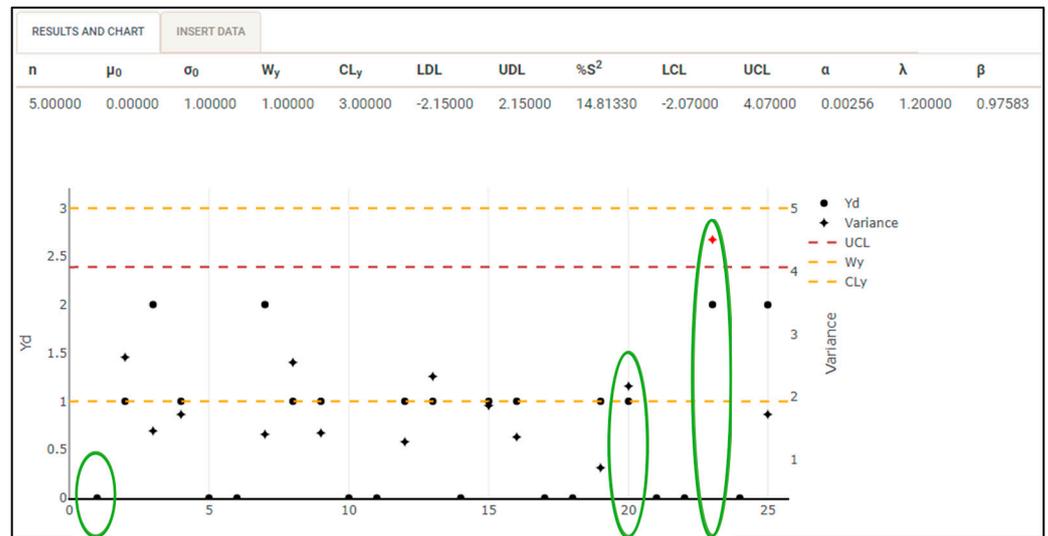


Figure 5. Example chart.

The first sample highlighted in the chart is a case where $Y_D \leq W_Y$ and S^2 does not need to be calculated thereof (as shown in Table 3), thus not being plotted in the chart (as in Figure 5).

The twentieth sample, also highlighted in the chart, is a case where $CL_Y < Y_D \leq W_Y$. Thereby, S^2 is calculated and plotted, since $S^2 \leq UCL$, and the variance is plotted in black in this example.

Similarly, the twenty-third sample, also circled in the chart, is a case where Y_D is also within the range of attribute control limits, and the variance is calculated and plotted. However, $S^2 > UCL$ and the chart signals plotting the variance are in red.

Likewise, in cases where $Y_D > CL_Y$, the chart signals that the process is out of control, and Y_D is plotted in red.

5. Conclusions

In this study, an interface was developed to generate and display control limits for the B-ATTRIVAR SS S^2 chart, enabling the real-time monitoring of process variance. It was achieved through inputting process data and user-defined requirements, and the program uses these data to compute the control limits and render a chart showing them and sample data. Once users have their process data, they can input it and use the control chart to monitor process variability. Although the application has not been tested in actual processes, its results have been validated in this study through comparisons with reference values, thus revealing low percentage deviations.

As an opportunity for improvement and further research, it is imperative to explore methods to reduce its execution time. Due to multiple iterations aimed at generating more precise results, the program might last a significantly long execution time depending on the device on which it is run. One potential approach to face this challenge is leveraging artificial intelligence tailored for optimizing code performance.

Furthermore, investigating the feasibility of adapting the application to integrate it with business and industrial systems, such as manufacturing execution systems (MES), could enhance its versatility and potential for automation across various industrial applications, including real-time production monitoring and control.

Moreover, future research should analyze the influence of variations in input parameters on results, in addition to conducting extensive studies and interface development aimed to consider the application of the proposed control strategy using estimated parameters, thereby extending its applicability. Furthermore, exploring methods for theoretically and mathematically deriving the ARLs and control limits of the proposed strategy would be of significant interest.

Author Contributions: Conceptualization, J.P.C.V. and M.A.G.M.; methodology, A.d.S.M. and T.S.A.; software, J.P.C.V. and A.d.S.M.; validation, M.A.G.M. and T.S.A.; formal analysis, J.P.C.V., M.A.G.M., A.d.S.M. and T.S.A.; investigation, J.P.C.V. and M.A.G.M.; resources, A.d.S.M. and T.S.A.; data curation, J.P.C.V., M.A.G.M., A.d.S.M. and T.S.A.; writing—original draft preparation, J.P.C.V., M.A.G.M., A.d.S.M. and T.S.A.; writing—review and editing, J.P.C.V., M.A.G.M., A.d.S.M. and T.S.A.; visualization, J.P.C.V., M.A.G.M., A.d.S.M. and T.S.A.; supervision, M.A.G.M.; project administration, M.A.G.M.; funding acquisition, M.A.G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, and the APC was funded by São Paulo State University under process EDITAL PROPG 16/2024.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: Thanks to PIBIC/CNPq-EDITAL 2020 and CNPq process 307945/2019-7.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shewhart, W.A. *Economic Control of Quality of Manufactured Product*, 1st ed.; D. Van Nostrand Company: Nova York, NY, USA, 1931.
2. Montgomery, D.C. *Introduction to Statistical Quality Control*, 6th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2009.
3. Ali, S.; Abbas, Z.; Nazir, H.Z.; Riaz, M.; Zhang, X.; Li, Y. On developing sensitive nonparametric mixed control charts with application to manufacturing industry. *Qual. Reliab. Eng. Int.* **2021**, *37*, 2699–2723. [[CrossRef](#)]
4. Aslam, M.; Saghir, A.; Ahmad, L. *Introduction to Statistical Process Control*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2020.
5. Ho, L.L.; Aparisi, F. Attrivar: Optimized control charts to monitor process mean with lower operational cost. *Int. J. Prod. Econ.* **2016**, *182*, 472–483. [[CrossRef](#)]

6. Aparisi, F.; Ho, L.L. M-ATTRIVAR: An attribute-variable chart to monitor multivariate process means. *Qual. Reliab. Eng. Int.* **2018**, *34*, 214–228. [[CrossRef](#)]
7. Aslam, M.; Azam, M.; Khan, N.; Jun, C.H. A mixed control chart to monitor the process. *Int. J. Prod. Res.* **2015**, *53*, 4684–4693. [[CrossRef](#)]
8. Bezerra, E.L.; Ho, L.L.; Quinino, R.d.C. GS2: An optimized attribute control chart to monitor process variability. *Int. J. Prod. Econ.* **2018**, *195*, 287–295. [[CrossRef](#)]
9. Ho, L.L.; Quinino, R.C. An attribute control chart for monitoring the variability of a process. *Int. J. Prod. Econ.* **2013**, *145*, 263–267.
10. Ho, L.L.; Quinino, R.d.C. Combining attribute and variable data to monitor process variability: MIX S 2 control chart. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 3389–3396. [[CrossRef](#)]
11. Wu, Z.; Khoo, M.B.C.; Shu, L.; Jiang, W. An np control chart for monitoring the mean of a variable based on an attribute inspection. *Int. J. Prod. Econ.* **2009**, *121*, 141–147. [[CrossRef](#)]
12. Sampaio, E.S.; Ho, L.L.; de Medeiros, P.G. A combined $np_x - \bar{X}$ control chart to monitor the process mean in a two-stage sampling. *Qual. Reliab. Eng. Int.* **2014**, *30*, 1003–1013. [[CrossRef](#)]
13. Leoni, R.C.; Costa, A.F.B. The performance of the truncated mixed control chart. *Commun. Stat.-Theory Methods* **2019**, *48*, 4294–4301. [[CrossRef](#)]
14. Costa, A.F.B.; Neto, A.F. Monitoring the process mean with an ATTRIVAR chart. *Commun. Stat.-Theory Methods* **2022**, *51*, 4903–4920. [[CrossRef](#)]
15. Simões, F.D.; Costa, A.F.B.; Machado, M.A.G. The Trinomial ATTRIVAR control chart. *Int. J. Prod. Econ.* **2020**, *224*, 1–8. [[CrossRef](#)]
16. Faraz, A.; Heuchenne, C.; Saniga, E. An exact method for designing Shewhart and S2 control charts to guarantee in-control performance. *Int. J. Prod. Res.* **2018**, *56*, 2570–2584. [[CrossRef](#)]
17. Xia, K.; Sacco, C.; Kirkpatrick, M.; Saidy, C.; Nguyen, L.; Kircaliali, A.; Harik, R. A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *J. Manuf. Syst.* **2021**, *58*, 210–230. [[CrossRef](#)]
18. Gendre, Y.; Waridel, G.; Guyon, M.; Demuth, J.-F.; Guelpa, H.; Humbert, T. Manufacturing execution systems: Examples of performance indicator and operational robustness tools. *Chimia* **2016**, *70*, 616–620. [[CrossRef](#)] [[PubMed](#)]
19. Li, B.; Chen, R.-S.; Liu, C.-Y. Using intelligent technology and real-time feedback algorithm to improve manufacturing process in IoT semiconductor industry. *J. Supercomput.* **2021**, *77*, 4639–4658. [[CrossRef](#)]
20. Aparisi, F.; Mosquera, J.; Epprechet, E.K. Simultaneously guaranteeing the in-control and out-of-control performances of the S2 control chart with estimated variance. *Qual. Reliab. Eng. Int.* **2018**, *34*, 1110–1126. [[CrossRef](#)]
21. Giorgi, F.M.; Ceraolo, C.; Mercatelli, D. The R Language: An Engine for Bioinformatics and Data Science. *Life* **2022**, *12*, 648–673. [[CrossRef](#)] [[PubMed](#)]
22. Chakraborti, S.; Graham, M.A. *Nonparametric Statistical Process Control*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2019.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.