



ELSEVIER



CrossMark

Procedia Manufacturing

Volume 5, 2016, Pages 122–131

44th Proceedings of the North American Manufacturing  
Research Institution of SME <http://www.sme.org/namrc>



# A Process Capability Analysis Method Using Adjusted Modified Sample Entropy

Siim Koppel<sup>1\*</sup> and Shing I Chang<sup>1</sup>

<sup>1</sup>*Department of Industrial and Manufacturing Systems Engineering,  
Kansas State University  
Manhattan, KS 66506, USA*

## Abstract

The evolution of sensors and data storage possibilities has created possibilities for more precise data collection in processes. However, process capability analysis has become more difficult. Traditional methods, such as process capability ratios, cannot handle large volumes of process data over time because these methods assume normal process distribution that is not changing. Entropy methods have been proposed for process capability studies because entropy is not dependent on distribution and can therefore provide accurate readings in changing distribution environments. The goal of this paper is to explore the use of entropy-based methods, specifically modified Sample Entropy to identify process variations over time. A study based on simulated data sets showed that the proposed method provides process capability information.

*Keywords:* Entropy, SampEn, mSampEn, AdmSE, AdSEn, Process Capability Analysis, Complex Manufacturing Processes

## 1 Introduction

Current production processes are becoming increasingly complex because the amount of data received from the production line continues to grow. Although data storage possibilities have evolved with the introduction of Big Data, the growth of recorded data has increased inherent statistical variability. Process capability analysis is often used to ensure process stability for quality assurance. However, these analyses are often performed retrospectively. In addition, these practices do not proactively detect process changes in real time; rather they compute process variation generated from a production period with the assumption that one steady-state process distribution exists in the period in question (Montgomery, 2009)

\* Corresponding author. E-mail: [siim@ksu.edu](mailto:siim@ksu.edu)

Process capability analysis must be able to detect changes and be free of distribution assumption. (Montgomery, 2009) asserted that process capability is a vital part of any quality-improvement process program. A process capability study includes two objectives: measure variability of process output and compare that variability to product tolerance (Montgomery, 2009)

Traditional process capability analysis has used process capability indexes that are measurable properties between manufacturing process variations and specification limits. Over the years there have been many studies and proposed methods developed to assess the process capabilities. (Yum and Kim, 2011) have assembled approximately 530 journal papers and books on process capability indices from 2000-2009 alone. After 2009 there have been proposals to use fuzzy methods (Kaya & Kahraman, 2011), different variations of classifications of rates of defects (for example real defective rate (Lupo, 2015)) etc in process capability analysis.

However, indices such  $C_p$ ,  $C_{pk}$ , and  $C_{pc}$  (Montgomery, 2009) are uniformity measurements for univariate processes, and they require that the quality characteristic be normally distributed and statistically in control. Unfortunately, modern production processes are so complex that fixed-process assumptions may not reflect the reality. This study utilized an entropy-based algorithm as a measure of process changes over time.

In general, entropy-based algorithms quantify the orderliness of a complex time series. When the degree of disorder in a process or system is high, the entropy output increases, allowing entropy to measure chaos of data signal or quantify the regularity of a time series when the distribution is unknown. Entropy-based algorithms have recently been used extensively in physiological and biomedical time-series analyses. Extensive review of various types of entropy-based algorithms is presented by (Humeau-Heurtier, 2015). (Kong et al., 2015) were one of the first to propose the use of entropy methods for process capability analysis. They proposed the use of transformed time series and Sample Entropy (SampEn) to detect mean shift and variance change.

The goal of this paper is to explore the possibilities of entropy methods for process capability analysis. The main contribution of this paper is a method based on Adjusted modified Sample Entropy (AdmSEn) that can be used as an univariate process capability study without the assumption of fixed-process distribution.

The rest of this paper is organized as follows. Section 2 presents the definition and characteristics of Adjusted Sample Entropy (AdSEn). Section 3 explains the method of Modified Sample Entropy (mSampEn), and Section 4 proposes an AdmSEn. A simulation study is presented in Section 5, and conclusions and future studies are presented in Section 6.

## 2 Adjusted Sample Entropy

In this report we use entropy to detect changes in time series. Entropy is defined as the average amount of information contained in each message received. A message refers to an event, sample, or character drawn from a distribution or data stream. Entropy also measures chaos in a data signal.

Andrei Kolmogorov derived the fundamental algorithm for entropy in 1963 (Kolmogorov, 1998) according to Shannon's theory published in 1948 (Shannon, 1948). The base of the algorithm required the calculation of probabilities of chances that a particular message was actually transmitted, and the entropy of the message system was a measure of the average amount of information in a message. Several modifications derived from Kolmogorov's algorithm have computed approximate entropy for finite input data using heuristics. Two most commonly used are Approximate Entropy (ApEn) and SampEn (Grassberger, 1988; Grassberger & Procaccia, 1983). Because these studies showed that SampEn is more accurate of the two when the data length varies, SampEn was used in this application.

For an  $N$ -points normalized time series  $\{u(i): 1 \leq i \leq N\}$  with mean zero and unit standard deviation, the following vector was formed:

$$Xi_m = \{u(i), u(i + 1), \dots, u(i + m - 1)\} \quad 1 \leq i \leq N - m + 1 \quad (1)$$

where  $X_i^m$  represents  $m$  consecutive  $u$  values starting from the  $i$ th point. Distance  $d_{ij}^m$  between  $X_i^m$  and  $X_j^m$  is defined as

$$d_{ij}^m = d[X_i^m, X_j^m] = \max|u(i+k) - u(j+k)|, k \in (0, m-1), i \neq j \quad (2)$$

The probability within threshold  $r$  for vector length  $m$  is defined as

$$C_r^m(i) = \frac{1}{N-m-1} \sum_{j=1, i \neq j}^{N-m} \theta(d_{ij}^m - r) \quad (3)$$

where  $\theta$  is the Heaviside step function

$$\theta(z) = \begin{cases} 1, & \text{if } z \leq 0 \\ 0, & \text{if } z > 0 \end{cases} \quad (4)$$

The cumulative probability within threshold  $r$  for vector length  $m$  is defined as

$$C_r^m = \frac{1}{N-m} \sum_{i=1}^{N-m} C_r^m(i) \quad (5)$$

We then formed  $(m+1)$ -dimensional embedding vectors  $X_i^{m+1} = \{u(i), u(i+1), \dots, u(i+m)\}$  and identically defined  $C_r^{m+1}$  using  $X_i^{m+1}$ .

SampEn is defined as

$$SampEn(m, r) = -\ln \frac{C_r^{m+1}}{C_r^m(i)} \quad (6)$$

where  $SampEn(m, r)$  is a negative natural logarithm of the conditional probability that two similar sequences for  $m$  points remain similar within a tolerance of  $r$  at the next point.

In the definition of SampEn, vector similarity is based on the Heaviside function, which primarily provides a step function that converts input into activity equal to 0 or 1. This function is discontinuous because it contains a ‘‘break’’ when the value goes from 0 to 1. Contributions of all data points inside the boundary are treated as equal, while data points outside the boundary are ignored. As a result, SampEn values may be discontinuous and dramatically rise or fall if threshold  $r$  changes slightly.

SampEn detects variance change by counting the number of data points that fall within the threshold. This counting mechanism, which is based on the overall variance of a time series, enables SampEn to handle variance change detection. Unfortunately, however, the algorithm does not contain an element to react to mean level shifts. Therefore, AdSEn was derived [5], with a core component of transformation based on input in order to synthesize mean shift and variance changes. The mean shift in a sample dataset must be converted into variance change via input transformation. The AdSEn algorithm transforms the original time series of data into a new time series as follows:

$$y_{ij} = x_{ij} \left( \left| \frac{\bar{x}_i - \mu}{\sigma} \right| + 1 \right) \quad (7)$$

where  $\bar{x}_i$  is the estimated mean of  $i^{\text{th}}$  segment set;  $\mu$  is the desired mean of the variable of interest, and  $\sigma$  is the desired standard deviation of the variable of interest.

### 3 Modified Sample Entropy

SampEn, however, contains limitations. For example, SampEn does not typically produce an output with a small sample size; relevant studies suggest a sample size of  $20^m$  to obtain readings. In addition, tolerance  $r$  cannot be small; the SampEn algorithm gives outputs until  $r = 0.2$  if the sample size is around 100. If the sample size is 100–200, the tolerance must be approximately  $0.6 \times$  standard deviation. Both of these limitations are due to the fact that SampEn contains a Heaviside function, as discussed in Section 2. When the algorithm compares data points, it has an output of 1 (if the data point is less than the threshold) or 0 (else). It is really strict distinction, and the function is not continuous. Therefore, (Xie et al, 2010) proposed mSampEn as an alternative to the Heaviside function with use of a fuzzy membership function, resulting in an outcome that would represent the similarity degree between two data points.

Calculation of mSampEn is performed in the following steps. For  $N$ -points normalized time series  $\{u(i): 1 \leq i \leq N\}$ , the vector sequence is similar to the definition of SampEn:

$$X_i^m = \{u(i), u(i+1), \dots, u(i+m-1)\} - u_0(i) \quad 1 \leq i \leq N - m + 1$$

where  $X_i^m$  is generalized by removing a baseline

$$u_0(i) = \frac{1}{m} \sum_{j=0}^{m-1} u(i+j) \quad (8)$$

Then distance  $d_{ij}^m$  between vector  $X_i^m$  and  $X_j^m$  is defined as

$$d_{ij}^m = d[X_i^m, X_j^m] = \max_k |u(i+k) - u_0(i) - u(j+k) - u_0(j)|, \quad k \in (0, m-1), i \neq j \quad (9)$$

Similarity degree  $D_{ij}^m$  between  $X_i^m$  and  $X_j^m$  is determined by fuzzy membership function

$$D_{ij}^m = u(d_{ij}^m, r) \quad (10)$$

Any fuzzy membership function, such as Gaussian or Sigmoid bell shape, can be used as the fuzzy membership function as long as it is continuous to prevent abrupt changes in similarity and convex to maximize self-similarity. In this report we use the following Sigmoid function:

$$u(d_{ij}^m, r) = \frac{1}{1 + \exp\left(\frac{d_{ij}^m + 0.5}{r}\right)} \quad (11)$$

where  $u(d_{ij}^m, r)$  is the similarity output,  $d_{ij}^m$  is the distance between  $x_i$  and  $x_j$ , and  $r$  is the threshold.

Similar to the definition of SampEn, averaging all similarity degrees of neighboring vectors  $X_j^m$  of each vector  $X_i^m$  results in

$$C_r^m(i) = \frac{1}{N-m-1} \sum_{j=1, i \neq j}^{N-m} D_{ij}^m \quad (12)$$

The cumulative probability is

$$C_r^m = \frac{1}{N-m} \sum_{i=1}^{N-m} C_r^m(i) \quad (13)$$

We then form  $(m+1)$ -dimensional embedding vectors  $X_i^{m+1} = \{u(i), u(i+1), \dots, u(i+m)\}$  and identically defined  $C_r^{m+1}$  using  $X_i^{m+1}$ .

For finite data sets, mSampEn can be estimated from

$$mSampEn = -\ln \frac{C_r^{m+1}}{C_r^m(i)} \quad (14)$$

## 4 The Proposed Method

### 4.1 Input Transformation of the Proposed Adjusted Modified Sample Entropy Algorithm

Similar to SampEn, mSampEn can detect variance change but not mean shift. Therefore, we propose the following transformation of the original time series, similar to AdSEn.

Let  $x$  be a vector of a time series of univariate observations based on an historic data series. Vector  $x$  can be segmented into  $k$  equal length subsets so that  $x = \{x_1, x_2, \dots, x_k\}$ . Each subset  $x_i$  has  $n$  observations. Denote  $x_{ij}$  as the  $j^{\text{th}}$  observation within subsets  $x_i$ ; thus,  $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ . Transformed variable  $y$  is based on the original input variable  $x$ . AdSEn detects the anomaly of recorded process data from the original state. Therefore, the proposed method requires a predefined template dataset  $x_0$  in order to represent this original state. If a predefined template is unavailable,  $x_0$  can be set as the first segment of  $x$ , namely  $x_1$ . Any segment  $i$  deemed suitable, as in the control example, can also be used as the template. Denote mean and standard deviation of the template segment  $x_i$  as  $\mu_0$  and  $\sigma_0$ , respectively. The proposed transformation, shown in Equation (15), is an

adjustment based on multiplying  $\frac{\bar{x}_i - \mu_0}{\sigma_0}$  to the original input  $x_{ij}$ . The addition of a constant 1 to the absolute value of the multiplied value ensures that when  $\frac{\bar{x}_i - \mu_0}{\sigma_0} = 0$ , then  $y_{ij} = x_{ij}$ .

$$y_{ij} = x_{ij} \left( \left| \frac{\bar{x}_i - \mu}{\sigma} \right| + 1 \right) \quad (15)$$

For example, when a process has a mean shift of one sigma  $\sigma_0$ , then the numerical value in the parenthesis is 2 (i.e.,  $y_{ij} = 2 x_{ij}$ ), thereby amplifying the original observations.

## 4.2 Implementation of the Proposed Adjusted Modified Sample Entropy

The following steps describe how to use AdmSEn to identify process mean and variance changes.

1. Import data from collection devices, database, or real-time sensor.
2. Estimate process mean and standard deviation. If there is a known target and control standard deviation, use the existing  $\mu_0$  and  $\sigma_0$ . Otherwise, select a template segment within the HDS that is representative of the process under study and calculate the sample mean and sigma in the transformation function. To avoid overestimating within-sample variation, we used the sample size of  $n$  observations of this segment in order to compute a range of  $X_{\max} - X_{\min}$  and then compute  $\sigma \approx R / d_2$ , where  $R$  is the average of all ranges in this representative sub dataset and  $d_2$  is a constant related to  $n$  [8]. For example, consider a segment with 100 observations. If sample size  $n = 5$  is used to compute the ranges, then 20 ranges are available to compute  $R$ . Therefore,  $d_2 = 2.326$ , the process mean estimate  $\hat{\mu}$  is the sample mean, and the process standard deviation is  $\hat{\sigma} \approx R / d_2$ .
3. Input transformation transforms standardized input  $x$  into  $y$  according to the transformation formula.
4. Set parameters for mSampEn. Users can define parameters, including threshold  $r$ , length  $m$  of comparing vector, delay  $\tau$ , and multiscale resolution  $k$ , which determines how many segments the HDS is divided into. Threshold  $r$  sets the boundary distance between vectors,  $m$  is the dimension of the vector,  $\tau$  sets the mode of vector selection (for a full-size dataset,  $\tau$  is recommended to be 1), and scale  $k$  is the number of segments of the HDS. In practice, if a user does not have a specific concept of the parameters, he/she only has to define scale number  $k$ ; the rest of the parameters can be set as  $r = 0.2$ ,  $m = 2$ , and  $\tau = 1$  as the default. The choice of  $k$  depends on the number of observations in an HDS. For a small historic data series,  $k$  should be chosen so that the number of observations within a segment is large enough.
5. Perform mSampEn calculations on inputs  $x$  and  $y$ .
6. Plot both outputs on one graph.

The following chapter describes simulations for SampEn, mSampEn, AdSEn, and AdmSEn to determine how each alternative can predict mean shifts and variance changes.

## 5 Simulations

Our first investigation determined if the AdmSEn performs as well as AdSEn for detecting mean and variance shift. We generated a time series using a random number generator. The time series consisted of 1200 data points as shown in Table 1.

Segment	1 (1–300)	2 (301–600)	3 (601–900)	4 (901–1200)
Mean	0	1	0	1
Variance	1	1	3	3

Table 1. Time series generating values

Each segment had 300 data points. The first segment with no change was the template segment, and the second segment was the mean shift segment. The third segment had variance change only, and the fourth segment contained both mean and variance shifts. We used threshold of  $r = 0.6$ .

For comparison, we also used the SampEn algorithm and mSampEn algorithm to observe the performance of the algorithms. The time series were generated 100 times, and the statistics of the outcomes are shown in Figure 1.

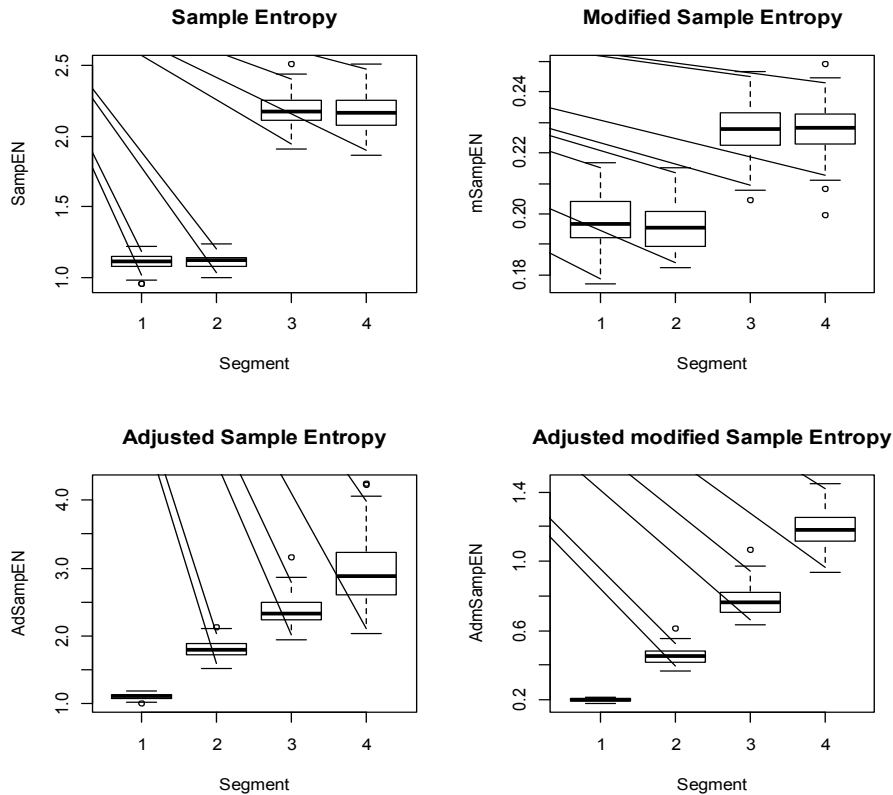


Figure 1. Box-plots of algorithm outcomes

As shown in the Figure 1, adjusted entropies caught mean shift and variance change, while the Sample Entropies caught only variance change.

Next we attempted to determine if the AdmSEn performs more consistently than AdSEn when the segment lengths are 50 and 100 points. We also wanted to identify where the threshold becomes a factor, so we generated two sets of numbers: one with a length of 50 and the other with a length of 100. We changed the threshold value  $r$  from 0.05 to 0.6 with an interval of 0.05.

Data with length of 50 points produced can be seen in Figure 2 and Table 2

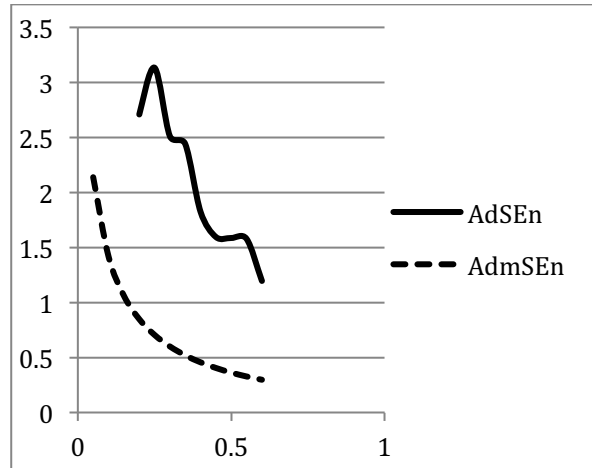


Figure 2. Comparison of AdSEn and AdmSEn outputs with 50 data points

r	AdSEn	AdmSEn
0.05	NA	2.141139
0.1	Inf	1.419322
0.15	Inf	1.062442
0.2	2.70805	0.849621
0.25	3.135494	0.705721
0.3	2.512306	0.600625
0.35	2.442347	0.520104
0.4	1.824549	0.456453
0.45	1.596015	0.405015
0.5	1.586965	0.362735
0.55	1.578979	0.327498
0.6	1.196251	0.297782

Table 2. Numerical output of AdSEn and AdmSEn algorithms with 50 data points

Data with length of 100 points can be seen in Figure 3 and Table 3:

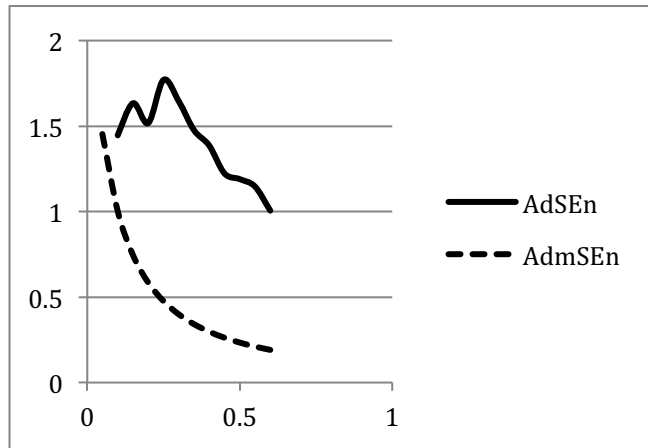


Figure 3: Comparison of AdSEn and AdmSEn outputs with 100 data points

r	AdSEn	AdmSEn
0.05	Inf	1.453439
0.1	1.446919	1.001401
0.15	1.634131	0.744896
0.2	1.517871	0.584493
0.25	1.770706	0.476331
0.3	1.645806	0.399147
0.35	1.476541	0.341636
0.4	1.386294	0.297321
0.45	1.223107	0.262254
0.5	1.189962	0.2339
0.55	1.146656	0.210561
0.6	1.005875	0.191062

Table 3. Numerical output of AdSEn and AdmSEn algorithms with 100 data points

As shown in Figure 2 and Figure 3, AdmSEn was much more consistent and smoother when the data length was small. In addition, AdmSEn had an output with really small threshold of  $r$ , while the AdSEn method did not provide consistent results.



## 6 Conclusions and Future Work

In this paper, an AdmSEn method was proposed to help detect process pattern changes. AdmSEn is based on mSampEn with time series transformation. A simulation study was conducted to compare characteristics of proposed method to SampEn and AdSEn. In the first simulation, a dataset of 1200 observations was created using four segments to show how the proposed method detects mean shift and variance change. The simulations were repeated 100 times with random seeds. Figure 1 of boxplots of AdmSEn values showed that the proposed method detected mean shift and variance change. A second simulation compared AdmSEn to AdSEn. The following conclusions were made regarding the proposed AdmSEn:

1. AdmSEn is more stable and smoother than AdSEn.
2. AdmSEn performs well with small sample sizes and very low threshold levels

AdmSEn was better than AdSEn in Statistical Process Control applications.

The proposed work, however, was limited to univariate cases. Future work should extend the AdmSEn to multivariate cases for process capability studies. An investigation should also be made into whether AdmSEn could be used in a control chart environment for quality assurance and fault detection.

## References

- Grassberger, P. Finite sample corrections to entropy and dimension. *Physics Letters A* 1988; 128((6-7)): 369-373.
- Grassberger, P, & Procaccia, I. Estimation of the Kolmogorov entropy from a chaotic signal. *Physical Review A* 1983; 28(4): 2591-2593.
- Humeau-Heurtier, A. The multiscale entropy algorithm and its variants: A review *Entropy* 2015; 17(5): 3110-3123.
- Kaya, I. & Kahraman, C. Process capability analyses with fuzzy parameters. *Expert Systems with Applications* 2011; 38(9): 11918-11927
- Kolmogorov, A. On Tables of Random Numbers. *Theoretical Computer Science* 1998, 207: 387-395.
- Kong, X., Chang, S.I. & Zhang, Z. A Novel Method Based on Adjusted Sample Entropy for Process Capability Analysis in Complex Manufacturing Processes. In: *ASME 2015 International Manufacturing Science and Engineering Conference, MSEC 2015*.
- Lupo, T. The new Nino capability index for dynamic process capability analysis. *Quality and Reliability Engineering International* 2015; 31(2): 305-312
- Montgomery, D. *Introduction to Statistical Quality Control* Sixth Edition, Arizona State University, 2009.
- Shannon, C.A. Mathematical Theory of Communication. *Bell SysTetch* 1948, 27: 379-423
- Yum, B.J & Kim K.WA Bibliography of the Literature on process Capability Indices: 2000-2009. *Quality and Reliability Engineering International* 2011; 27(3): 251-268
- Xie, H.B, Guo, J.Y, & Zheng, Y.P. (2010). Using the modified sample entropy to detect determinism. *Physics Letters A* 2010; 374(38): 3926-3931.