Improving CUSUM Control Chart for Monitoring a Change in Processes Based on Seasonal ARX Model

C. Chananet and S. Phanyaem*

***Abstract*—The Cumulative Sum (CUSUM) control chart is a widely used statistical quality control tool, especially in manufacturing processes. It serves to give an alarm when the process is out of control. Control Charts are frequently used to monitor and evaluate product quality in the manufacturing process. In real-world situations where random processes are related in sequence, such as hospitalizations, stock prices, or daily rainfall. The main purpose of this paper is to develop the explicit formula for the average run length (ARL) of the CUSUM control chart based on a seasonal autoregressive model with one exogenous variable ($SARX(1,1)_L$) using the Fredholm integral equation. The findings of the explicit formula were also compared to the results of a numerical integral equation using the Gaussian rule, the Midpoint rule, and the Trapezoidal rule. The results show that the analytical solution of the ARL is sufficiently accurate and easy to calculate in comparison with the numerical integration techniques. In addition, evaluating the proposed explicit formulas on the CPU time takes less than a second, whereas the numerical integration takes 13–18 minutes.**

***Index Terms*—Statistical Process Control, Analytical solution, Numerical integration, Average Run Length**

I. Introduction

TWO of the most prevalent types of control charts are Xbar-R charts and individual charts. Shewhart control charts were first introduced by Walter Shewhart [1]. Shewhart control charts are useful for detecting somewhat large changes in the mean of the process. The Cumulative Sum (CUSUM) control chart is an alternate control chart that is primarily used to detect small shifts in the mean of the process. Page [2] introduced the CUSUM control chart in 1954. It's been used in a variety of industries, but it's most famous for its use in the chemical industry. The CUSUM control charts are used to detect changes in the output of the manufacturing process. In terms of detecting small changes in the process means, Hawkins and Olwell [3] and Lucas [4] show that the CUSUM chart is more efficient than the Shewhart chart.

Control charts are currently utilized in medicine to ensure the quality of drug manufacture. The cumulative sum control chart has recently been introduced to detect process changes in dichotomized or quantitative data. It's a simple and effective chart for detecting small changes, with the additional benefit of demonstrating the robustness to non-normality (See for detail Borror et, al. [5]). The observations from the process are usually assumed to be independent and identically distributed (*i.i.d.*) with a normal distribution when control charts are created and evaluated. In real applications, there are several instances where process data is derived from non-normal distributions, such that the data is autocorrelated, which necessitates the use of appropriate control charts to monitor the data.

Average Run Length (*ARL*) is a common criterion for comparing the performance of control charts. ARL_0 denotes the expected number of observations taken from an in-control process before the control chart inappropriately signals out-of-control. An ARL_0 is considered appropriate if it is large enough to keep the false alarm rate under control. The expected number of observations obtained from an out-of-control process until the control chart alerts that the process is out-of-control, which is denoted by ARL_1, is a second common characteristic.

Many approaches for evaluating the ARL have been proposed in the literature, including Monte Carlo simulations (MC), Markov Chain Methodology (MCA), and the Numerical Integral Equation approach (NIE). Busaba et al. [6] studied the ARL for the CUSUM control chart by using the Fredholm Integral Equation to derive the ARL. Phanyaem et al. [7] used the integral equation technique to derive the explicit formula for the ARL of the CUSUM control chart for an autoregressive and moving average process with exponential white noise. Petcharat et al. [8] established an analytical equation for the ARL of the CUSUM control chart when the random observations are modeled as a moving average of an order q process. Later, Paichit [9] presented an analytical solution for the ARL of a CUSUM control chart for an autoregressive process with one explanatory variable. Subsequently, Phanyaem [10] used an Integral Equation approach for the seasonal autoregressive moving average; $SARMA(1,1)_L$ model to evaluate the ARL of the CUSUM control chart. Recently, Piyapatr and Lili [11] compared numerical integration to explicit formulas for the

Manuscript received December 30, 2021; revised May 10, 2022. This research was funded by Thailand Science Research and Innovation Fund, and King Mongkut's University of Technology North Bangkok, with Contract no. KMUTNB-BasicR-64-19.

C. Chananet is an Assistant Professor of Applied Statistics Department, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand (phone: 662-555-2000 e-mail: chanaphun.c@sci.kmutnb.ac.th).

*Corresponding author: S. Phanyaem is an Assistant Professor of Applied Statistics Department, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand (phone: 662-555-2000 e-mail: suvimol.p@sci.kmutnb.ac.th).

ARL of the CUSUM control chart for an autoregressive integrated moving average model with exponential white noise. Furthermore, Peerajit et al. [12] showed the explicit formulas for the ARL of CUSUM control chart for ARMA with an exogenous variable; $ARMAX(p,q,r)$ model. Finallly, Phanyaem [13] derived the explicit formula of the ARL for the Exponentially Weighted Moving Average chart by using an integral equation when observations are described by an autoregressive integrated moving average; $ARIMA(p,d,q)$ model.

The analytical formula for ARL_0 and ARL_1 of the CUSUM chart when observations are seasonal autoregressive with a single exogenous variable; $SARX(1,1)_L$ model with exponential white noise which is derived in this research. In addition, we approximate the ARL using the Gaussian rule, the Midpoint rule, and the Trapezoidal rule, and compare the results obtained using explicit formulae. The following is the structure of this paper: The materials and methods are given in Section 2. In Section 3, we compare the numerical results obtained from the explicit formula for the ARL of $SARX(1,1)_L$ model with the numerical solution obtained from a numerical integral equation. The final section contains the conclusions.

II. Materials and Methods

The statistical process control chart is investigated in this study under the assumption that sequential data are a seasonal autoregressive model with an exogenous variable and exponential white noise. The "in-control state" is defined as $\lambda = \lambda_0$ before a change-point time $\theta \le \infty$, and the "out-of-control state" is defined as $\lambda = \lambda_1$ after a change-point time θ. The stopping times of the control charts are used to construct the statistical process control charts. The CUSUM control chart's stopping time is given by

$$\tau_b = \inf\{t > 0; C_t > b\};\ t = 1,2,.... \tag{1}$$

where C_t is the CUSUM statistics

b is the upper control limit of the CUSUM chart.

The following are the general criteria for selecting the stopping times τ_b:

$$E_\infty(\tau_b) = T \tag{2}$$

where T is a constant, and $E_\infty(.)$ is the expectation that the change point will occur at θ under distribution $F(x,\lambda_0)$. The average run length for the in-control state is defined in this paper as $E_\infty(\tau_b)$. The average run length for an in-control process is known as ARL_0 and is calculated as

$$ARL_0 = E_\infty(\tau_b) = T \tag{3}$$

The constraint, on the other hand, is to minimize the quantity to as small as possible.

$$Sup_\theta E_\infty(\tau_b - \theta + 1 | \tau_b \ge \theta) \tag{4}$$

where $E_\theta(.)$ is the expectation under distribution $F(x,\lambda_1)$ and the value of a parameter after a change-point occurs is λ_1. The average run length is called ARL_1 and it is given by $\theta = 1$ for out-of-control processes.

$$ARL_1 = E_1(\tau_b) \tag{5}$$

Page [2] proposed recursive CUSUM statistics, which are defined as

$$C_t = \max(0, C_{t-1} + Y_t - a)\ ;\ t = 1,2,.... \tag{6}$$

where Y_t is a $SARX(1,1)_L$ model with exponential white noise sequence

C_0 is the starting value of the CUSUM statistics;

a is the reference value of CUSUM control chart.

In this paper, we consider the seasonal autoregressive model with one exogenous variable. The $SARX(1,1)_L$ model is defined as

$$Y_t = \mu + \varphi Y_{t-L} + \varepsilon_t + \beta X_t\ ;\ t = 1,2,.... \tag{7}$$

where μ is a process mean

X_t is an exogenous variable

β is an exogenous variable coefficient

φ is an autoregressive coefficient

ε_t is an exponential white noise.

Let $\mathbb{P}_C$ is the probability measure and $\mathbb{E}_C$ is the expectation corresponding to the initial value of CUSUM statistics $C_0 = u$. The CUSUM control chart's average run length is defined as

$$L(u) = \mathbb{E}(\tau_b) < \infty\ ;\ u \in [0,b] \tag{8}$$

The average run length is solved using the Fredholm Integral Equations of the second kind:

$$\begin{aligned} L(u) &= \mathbb{E}_C[I\{C_1 \ge b\}] + \mathbb{E}_C[I\{C_1 < b\}(\tau_b)] \\ &= 1 + \mathbb{E}_C[I\{0 \le C_1 \le b\}]L(C_1) \\ &= 1 + \mathbb{E}_C[I\{0 < C_1 < b\}L(C_1)] + \mathbb{P}_C\{C_1 = 0\}L(0) \end{aligned} \tag{9}$$

Consequently, we get the following formula for the ARL of the CUSUM control chart:

$$\begin{aligned} L(u) = 1 + \lambda e^{\lambda(u-a+\mu+\varphi Y_{t-L}+\beta X_t)} \int_0^b L(y) e^{-\lambda y} dy \\ + (1 - e^{-\lambda(a-u-\mu-\varphi Y_{t-L}-\beta X_t)} L(0). \end{aligned} \tag{10}$$

Let d be a constant of the function $L(u)$; $d = \int_0^b L(y)e^{-\lambda y}dy.$

So, the corresponding function of ARL can be written as

$$L(u) = 1 + \lambda e^{\lambda(u-a+\mu+\varphi Y_{t-L}+\beta X_t)}d + (1 - e^{-\lambda(a-u-\mu-\varphi Y_{t-L}-\beta X_t)})L(0). \quad (11)$$

For the case $u = 0$ we obtain the function $L(0)$ as follow:

$$\begin{aligned} L(0) &= 1 + \lambda e^{\lambda(u-a+\mu+\varphi Y_{t-L}+\beta X_t)}d \\ &\quad + (1 - e^{-\lambda(a-u-\mu-\varphi Y_{t-L}-\beta X_t)})L(0) \\ &= e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} + \lambda d. \end{aligned} \quad (12)$$

We get the function $L(u)$ by substituting $L(0)$ into (11).

$$\begin{aligned} L(u) &= 1 + \lambda e^{\lambda(u-a+\mu+\varphi Y_{t-L}+\beta X_t)}d \\ &\quad + (1 - e^{-\lambda(a-u-\mu-\varphi Y_{t-L}-\beta X_t)})e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} + \lambda d; \\ &= 1 + e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} + \lambda d - e^{\lambda u}. \end{aligned} \quad (13)$$

Next, we consider finding the constant d as follows:

$$\begin{aligned} d &= \int_0^b L(y)e^{-\lambda y}dy. \\ &= \int_0^b (1 + \lambda d + e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} - e^{\lambda u})e^{-\lambda y}dy \\ &= \int_0^b (1 + \lambda d + e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} - e^{\lambda u})dy - \int_0^b e^{\lambda y-\lambda y}dy \\ &= \frac{e^{\lambda b}}{\lambda}(1 - e^{-\lambda b})(1 + e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)}) - be^{-\lambda b} \end{aligned} \quad (14)$$

As a result of solving the Integral Equation, the following explicit formulas are obtained:

$$L(u) = e^{\lambda b}(1 + e^{\lambda(a-\mu-\varphi Y_{t-L}-\beta X_t)} - \lambda b) - e^{\lambda u} \; ; u \geq 0 \quad (15)$$

The value of the exponential white noise parameter is set to $\lambda = \lambda_0$ while the process is in the control state. Hence, we can derive the explicit formula of ARL_0 for the CUSUM control chart as follows:

$$ARL_0 = e^{\lambda_0 b}(1 + e^{\lambda_0(a-\mu-\varphi Y_{t-L}-\beta X_t)} - \lambda_0 b) - e^{\lambda_0 u} \; ; u \geq 0 \quad (16)$$

While the process is out-of-control state, the value of exponential white noise parameter $\lambda = \lambda_1$; $\lambda_1 = \lambda_0(1+\delta)$. Thus, we obtain the explicit formula of ARL_1 for CUSUM control chart as follows:

$$ARL_1 = e^{\lambda_1 b}(1 + e^{\lambda_1(a-\mu-\varphi Y_{t-L}-\beta X_t)} - \lambda_1 b) - e^{\lambda_1 u} \; ; u \geq 0 \quad (17)$$

III. Numerical Methods for Solving the Integral Equation

Quadratures of the Gaussian type are particularly useful for calculating integrals of the average run length. The Gauss–Legendre quadrature formula is used to achieve accurate numerical integration. Champ and Rigdon [14] introduce the numerical integral equation, or NIE method.

In this paper, we study at how to find an approximation of an integral equation using Gaussian rules, Midpoint rule and Trapezoidal rule.

The scheme for numerically evaluating the solutions of the integral equation is presented in this section. Firstly, the integral equation of the ARL in equation (10) can be rewritten as follows:

$$\begin{aligned} \tilde{L}(u) &= 1 + L(0)F(a-u-\mu-\varphi Y_{t-L}-\beta X_t) \\ &\quad + \int_0^b L(y)f(y+a-u-\mu-\varphi Y_{t-L}-\beta X_t) \end{aligned} \quad (18)$$

If y is a random variable with an exponential distribution and $f(y) = \lambda e^{-\lambda y}$ is the probability density function, and $F(y) = 1 - e^{-\lambda y}$ is the cumulative density function. This numerical approximation method divides the interval [0, b] into the same m subintervals and uses specific formulas to calculate the area of the function for each subinterval.

In Gaussian rules, the integration interval can be infinite, the weight function $W(y)$ might not equal 1, and the set of points $\{y_k, k = 1, 2, ..., n\}$ is equally spaced. Gauss-Legendre quadrature is a form of the Gaussian quadrature rule that is used to approximate the integral equation.

The Gaussian quadrature rule has the following form for integrating over the interval [0, b]

$$\int_0^b W(y)f(y)dy \approx \sum_{k=1}^m w_k f(a_k)$$

where a_k is a set of point, $0 \leq a_1 \leq a_2 \leq ... \leq a_m \leq b$ and w_k is a set of constant weight, $w_k = b/m \geq 0$.

The Midpoint rule approximation for integrating over the interval [0, b] is given by:

$$\int_0^b f(y)dy \approx h\sum_{k=1}^m f\left(a + \left(k - \frac{1}{2}\right)h\right).$$

where h is the width of subinterval, $h = b/m$

m is the subinterval.

The Trapezoidal rule approximation for integrating over the interval [0, b] is given by:

$$\int_0^b f(y)dy \approx \frac{h}{2}(f(0) - f(b)) + h\sum_{k=1}^m f(y_k).$$

where h is the width of subinterval, $h = \frac{b-0}{2m}$

m is the subinterval.

$\tilde{L}(a_i)$ denotes the ARL's numerical approximation to the numerical integral equation for the $SARX(1,1)_L$ model.

$$\tilde{H}(a_i) = 1 + \tilde{H}(0)\,\mathrm{F}(a - a_i - \mu - \varphi Y_{t-L} - \beta X_t) + \sum_{j=1}^{m} w_j \tilde{H}(a_j) f(a_j + a - a_i - \mu - \varphi Y_{t-L} - \beta X_t).$$

We can approximate the function $\tilde{L}(u)$ to solve this set of equations for the approximate values of $\tilde{L}(a_1), \tilde{L}(a_2),..,$ and $\tilde{L}(a_m)$, which can be written as

$$\tilde{L}(a_1) = 1 + \tilde{L}(a_1)[\mathrm{F}(a - a_1 - \mu - \varphi Y_{t-L} - \beta X_t) + w_1 f(a - \mu - \varphi Y_{t-L} - \beta X_t)] + \sum_{j=2}^{m} w_j \tilde{L}(a_j) f(a_j + a - a_1 - \mu - \varphi Y_{t-L} - \beta X_t)$$

$$\tilde{L}(a_2) = 1 + \tilde{L}(a_1)[\mathrm{F}(a - a_2 - \mu - \varphi Y_{t-L} - \beta X_t) + w_1 f(a_1 + a - a_2 - \mu - \varphi Y_{t-L} - \beta X_t)] + \sum_{j=2}^{m} w_j \tilde{L}(a_j) f(a_j + a - a_2 - \mu - \varphi Y_{t-L} - \beta X_t)$$

$$\vdots$$

$$\tilde{L}(a_m) = 1 + \tilde{L}(a_1)[\mathrm{F}(a - a_m - \mu - \varphi Y_{t-L} - \beta X_t) + w_1 f(a_1 + a - a_m - \mu - \varphi Y_{t-L} - \beta X_t)] + \sum_{j=2}^{m} w_j \tilde{L}(a_j) f(a_j + a - a_m - \mu - \varphi Y_{t-L} - \beta X_t)$$

The numerical integral equation can be rewritten in matrix form as follows:

$$\mathbf{L}_{m\times1} = \mathbf{1}_{m\times1} + \mathbf{R}_{m\times m}\mathbf{L}_{m\times1}$$

where $\mathbf{L}_{m\times1} = \begin{pmatrix} \tilde{L}(a_1) \\ \tilde{L}(a_2) \\ \vdots \\ \tilde{L}(a_m) \end{pmatrix}$, $\mathbf{1}_{m\times1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

and $\mathbf{I}_m = diag(1,1,\ldots,1)$ is the unit matrix of order m. If there exists an $(\mathbf{I}_m - \mathbf{R}_{mxm})^{-1}$, the solution of the matrix equation is as follows:

$$\mathbf{L}_{m\times1} = (\mathbf{I}_m - \mathbf{R}_{m\times m})^{-1}\mathbf{1}_{m\times1}\,.$$

As a result, for the CUSUM chart based on the $SARX(1,1)_L$ model, the numerical integration of ARL is as follows:

$$\tilde{L}(u) = 1 + \tilde{L}(a_1)[\mathrm{F}(a - u - \mu - \varphi Y_{t-L} - \beta X_t) + w_1 f(a_1 + a - u - \mu - \varphi Y_{t-L} - \beta X_t)] + \sum_{j=2}^{m} w_j \tilde{L}(a_j) f(a_j + a - u - \mu - \varphi Y_{t-L} - \beta X_t) \quad (19)$$

IV. Results

In this section, we compute the explicit formula values for ARL_0 and ARL_1 from equation (16) and equation (17) with the parameters (a and b) and compare these results with values obtained from the numerical integration approach. $L(u)$ denotes the explicit formula, while $\tilde{L}(u)$ denotes the numerical integral equation. The computational time (CPU) required to compute the numerical values for ARL_0 and ARL_1 is also compared, the values in parentheses denote the computational times. The absolute percentage difference is calculated as follows:

$$Diff\,(\%) = \frac{|ARL_{EF} - ARL_{NIE}|}{ARL_{EF}} \times 100.$$

where ARL_{EF} is the ARL from explicit formula

ARL_{NIE} is the ARL from NIE method.

Table I and Table II shows the ARL_0 and ARL_1 on the $SARX(1,1)_4$ model with $\varphi = 0.10$, $\beta = 0.1$ by using the explicit formula and the Gaussian rule NIE methods on the CUSUM control chart. The ARL are computed for an in-control parameter value $\lambda_0 = 1$ and the numerical values for ARL_1 are computed for a range of out-of-control parameter values $\lambda = \lambda_1 = \lambda_0(1+\delta)$ where δ = 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, and 3.0. The reference values (a) were 2.50 and 3.00, respectively and an initial value (u) was 1.

In Table I, we set the parameter values for the $SARX(1,1)_4$ model with φ = 0.10, β = 0.10 then the parameter values of CUSUM control chart are a = 2.50 and b = 3.976. The CPU times have been obtained for each computation. For the case λ_0 = 1 and ARL_0 = 370, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for Gaussian rule NIE method runs for 13-14 minutes.

In Fig 1, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_4$ model by using the explicit formula and the Gaussian rule NIE methods, given ARL_0 = 370, a = 2.50, and b = 3.976. The results show that the ARL from the explicit formula method and the Gaussian rule NIE method differ only slightly.

In Table II, we set the parameter values for the $SARX(1,1)_4$ model with φ = 0.10, and β = 0.10, then the parameter values of CUSUM control chart are a = 3.00 and b = 3.270. The results show that the explicit formula takes less than a second to compute, whereas the Gaussian rule NIE approach takes 13-15 minutes on the CPU.

In Fig 2, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_4$ model by using the explicit formula and the Gaussian rule NIE methods, given ARL_0 = 370, a = 3.00, and b = 3.270. The results show that the ARL from the explicit formula is close to the Gaussian rule NIE methods.

In Table III, we set the parameter values for the $SARX(1,1)_4$ model with φ = 0.10, β = 0.10 then the parameter values of CUSUM control chart are a = 2.50 and b = 4.326. The CPU times have been obtained for each computation. For the case λ_0 = 1 and ARL_0 = 500, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for Gaussian rule NIE method runs for 12-13 minutes.

In Fig.3, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_4$ model by using the explicit formula and the Gaussian rule NIE methods, The parameter values used for the CUSUM control chart were a = 2.5 and b = 4.326. The results suggest that the explicit formula's ARL is comparable to the Gaussian rule NIE approaches.

In Table IV, we set the parameter values for the $SARX(1,1)_4$ model with φ = 0.10, β = 0.10 then the parameter values of CUSUM control chart are a = 3.00 and b = 3.592. The CPU times have been obtained for each computation. For the case λ_0 = 1 and ARL_0 = 500, the CPU time required for the Gaussian rule NIE method is 13-14 minutes, but the calculation time based on the explicit formula is less than 1 second.

In Fig 4, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_4$ model by using the explicit formula and the Gaussian rule NIE methods, the parameter values used for the CUSUM control chart were a = 3.00 and b = 3.592. The results suggest that the explicit formula's ARL is comparable to the Gaussian rule NIE approaches.

Table V and VI show the ARL_0 and ARL_1 on the $SARX(1,1)_{12}$ model with φ = 0.30, β = 0.50 by using the explicit formula and the Midpoint rule NIE methods on the CUSUM control chart. The ARLs are computed for an in-control parameter value $\lambda_0 = 1$ and numerical values for ARL_1 are computed for a range of out-of-control parameter values $\lambda = \lambda_1 = \lambda_0(1+\delta)$ where δ = 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, and 3.0. The CUSUM control chart was set the reference values (a) is 4.50 and 5.00, respectively, and an initial value (u) is 1.

In Table V, we set the parameter values for the $SARX(1,1)_{12}$ model with φ = 0.30, β = 0.50 then the parameter values of CUSUM control chart are a = 4.50 and b = 2.253. The CPU times have been obtained for each computation. For the case $\lambda_0 = 1$ and ARL_0 = 370, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for Midpoint rule NIE method runs for 14-15 minutes.

In Fig 5, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_{12}$ model by using the explicit formula and the Midpoint rule NIE methods, given ARL_0 = 370 and a = 4.50. The results show that the ARL from the explicit formula and the Midpoint rule NIE methods differ only slightly.

In Table VI, we set the parameter values for the $SARX(1,1)_{12}$ model with φ = 0.30, β = 0.50 then the parameter values of CUSUM control chart are a = 5.0 and b = 1.732. The CPU times have been obtained for each computation. For the case $\lambda_0 = 1$ and ARL_0 = 370, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for Midpoint rule NIE method runs for 14-15 minutes.

In Fig 6, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_{12}$ model by using the explicit formula and the Midpoint rule NIE methods, given ARL_0 = 370 and a = 5.00. The results show that the ARL from the explicit formula and the Midpoint rule NIE methods differ only slightly.

Table VII and VIII presents the ARL_0 and ARL_1 on the $SARX(1,1)_{12}$ model with φ = 0.50, β = 0.80 by using the explicit formula and the Trapezoidal rule NIE methods on the CUSUM control chart. The ARLs are computed for an in-control parameter value λ_0 = 1 and numerical values for ARL_1 are computed for a range of out-of-control parameter values $\lambda = \lambda_1 = \lambda_0(1+\delta)$ where δ = 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, and 3.0. The CUSUM control chart was set with the reference values (a) at 4.50 and 5.00, respectively, and an initial value (u) at 1.

In Table VII, we set the parameter values for the $SARX(1,1)_{12}$ model with φ = 0.50, β = 0.80 then the parameter values of CUSUM control chart are a = 4.5 and b = 3.110. The CPU times have been obtained for each computation. For the case $\lambda_0 = 1$ and ARL_0 = 500, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for the Trapezoidal rule NIE method runs for 15-18 minutes.

In Fig 7, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_{12}$ model by using the explicit formula and the Trapezoidal rule NIE methods, given ARL_0 = 500 and a = 4.50. The results show that the ARL from the explicit formula and the Trapezoidal rule NIE methods differ only slightly.

In Table VIII, we set the parameter values for $SARX(1,1)_{12}$ model with φ = 0.50, β = 0.80 then the parameter values of the CUSUM control chart are a = 5.0 and b = 2.560. The CPU times have been obtained for each computation. For the case $\lambda_0 = 1$ and ARL_0 = 500, the computing time based on the explicit formula takes less than 1 second, while the CPU time required for Trapezoidal rule NIE method runs for 15-18 minutes.

In Fig 8, we compare the ARL of the CUSUM control chart based on the $SARX(1,1)_{12}$ model by using the explicit formula and the Trapezoidal rule NIE methods, given ARL_0 = 500 and a = 5.00. The results show that the ARL from the explicit formula and the Trapezoidal rule NIE methods differ only slightly.

TABLE I
COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST GAUSSIAN RULE NUMERICAL INTEGRAL EQUATION FOR $SARX(1,1)_4$ MODEL GIVEN ARL_0 = 370, φ = 0.10, β = 0.10, a = 2.5 AND b = 3.976

Shift size δ	Explicit Formula Method		Gaussian Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	370.31	0.01	370.00	13.46	0.0834
1.50	7.922	0.01	8.025	12.95	1.3002
1.60	7.266	0.01	7.351	13.35	1.1698
1.70	6.712	0.01	6.784	13.16	1.0727
1.80	6.240	0.01	6.300	13.72	0.9615
1.90	5.834	0.01	5.885	13.72	0.8742
2.00	5.481	0.01	5.526	13.24	0.8210
2.10	5.173	0.01	5.211	13.62	0.7346
2.20	4.901	0.01	4.935	13.47	0.6937
2.30	4.661	0.01	4.691	13.83	0.6436
2.40	4.446	0.01	4.473	13.09	0.6073
2.50	4.254	0.01	4.278	13.91	0.5642
3.00	3.534	0.01	3.548	13.07	0.3962

TABLE II

COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST GAUSSIAN RULE NUMERICAL INTEGRAL EQUATION FOR $SARX(1,1)_4$ MODEL GIVEN $ARL_0 = 370$, $\varphi = 0.10$, $\beta = 0.10$, $a = 3.0$ AND $b = 3.270$

Shift size δ	Explicit Formula Method		Gaussian Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	370.24	0.01	370.00	14.04	0.0637
1.50	8.705	0.01	8.752	13.88	0.5399
1.60	7.950	0.01	7.989	14.52	0.4906
1.70	7.313	0.01	7.346	14.89	0.4513
1.80	6.770	0.01	6.799	14.85	0.4284
1.90	6.304	0.01	6.329	13.12	0.3966
2.00	5.900	0.01	5.922	14.23	0.3729
2.10	5.547	0.01	5.566	13.57	0.3425
2.20	5.237	0.01	5.254	13.64	0.3246
2.30	4.963	0.01	4.978	14.03	0.3022
2.40	4.719	0.01	4.733	14.12	0.2967
2.50	4.501	0.01	4.513	13.93	0.2666
3.00	3.690	0.01	3.697	14.27	0.1897

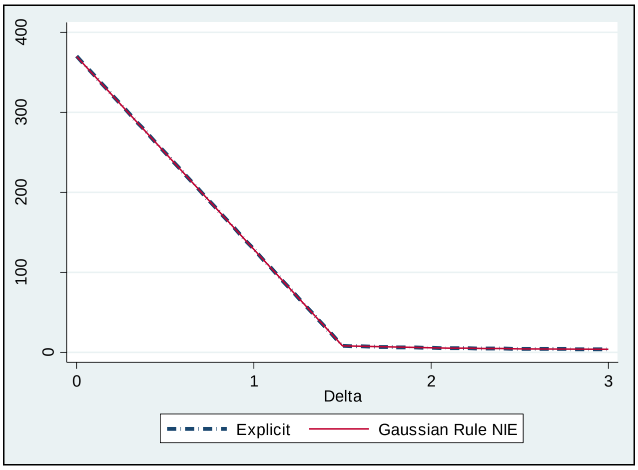

Fig 1 Comparison of ARL of the explicit formula and the Gaussian rule NIE methods for $SARX(1,1)_4$ model with $ARL_0 = 370$, $a = 2.5$ and $b = 3.976$

TABLE III

COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULAS AGAINST GAUSSIAN RULE NUMERICAL INTEGRAL EQUATION FOR $SARX(1,1)_4$ MODEL GIVEN $ARL_0 = 500$, $\varphi = 0.10$, $\beta = 0.10$, $a = 2.5$ AND $b = 4.326$

Shift size δ	Explicit Formula Method		Gaussian Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	500.16	0.01	500.00	13.01	0.0316
1.50	8.546	0.01	8.698	13.19	1.7786
1.60	7.814	0.01	7.938	13.49	1.5869
1.70	7.198	0.01	7.301	13.55	1.4310
1.80	6.675	0.01	6.762	13.63	1.3034
1.90	6.227	0.01	6.300	13.44	1.1723
2.00	5.839	0.01	5.902	13.46	1.0790
2.10	5.500	0.01	5.555	13.05	1.0000
2.20	5.203	0.01	5.250	13.67	0.9033
2.30	4.940	0.01	4.981	13.14	0.8300
2.40	4.706	0.01	4.728	13.08	0.4675
2.50	4.497	0.01	4.529	13.98	0.7116
3.00	3.717	0.01	3.736	12.54	0..5112

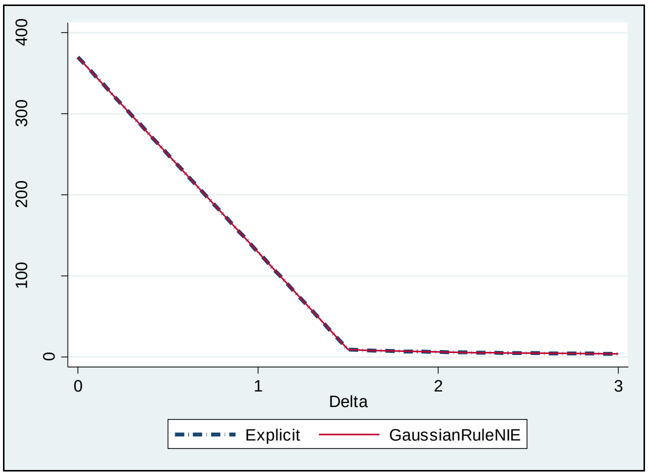

Fig 2 Comparison of ARL of the Explicit formula and the Gaussian rule NIE methods for $SARX(1,1)_4$ model with $ARL_0 = 370$, $a = 3.0$ and $b = 3.270$

TABLE IV

COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULAS AGAINST GAUSSIAN RULE NUMERICAL INTEGRAL EQUATION FOR $SARX(1,1)_4$ MODEL GIVEN $ARL_0 = 500$, $\varphi = 0.10$, $\beta = 0.10$, $a = 3.0$ AND $b = 3.592$

Shift size δ	Explicit Formula Method		Gaussian Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	500.22	0.01	500.00	13.08	0.0448
1.50	9.565	0.01	9.635	13.32	0.7318
1.60	8.699	0.01	8.757	13.84	0.6667
1.70	7.972	0.01	8.021	13.81	0.6147
1.80	7.355	0.01	7.397	13.69	0.5710
1.90	6.827	0.01	6.863	13.86	0.5273
2.00	6.371	0.01	6.402	13.72	0.4866
2.10	5.975	0.01	6.001	13.75	0.4351
2.20	5.627	0.01	5.651	13.78	0.4265
2.30	5.321	0.01	5.342	13.00	0.3947
2.40	5.049	0.01	5.068	13.13	0.3363
2.50	4.807	0.01	4.823	13.08	0.3328
3.00	3.909	0.01	3.919	13.02	0.2558

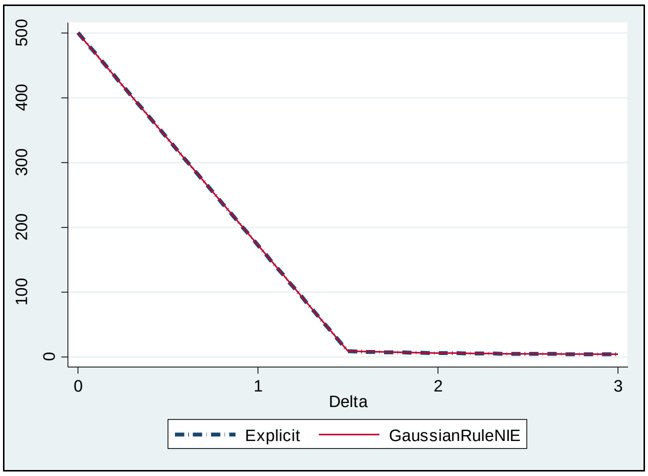

Fig 3 Comparison of ARL of the Explicit formula and the Gaussian rule NIE methods for $SARX(1,1)_4$ model with $ARL_0 = 500$, $a = 2.5$ and $b = 4.326$

TABLE V
COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST MIDPOINT RULE NUMERICAL INTEGRAL EQUATION FOR SARX(1,1)$_{12}$ MODEL GIVEN $ARL_0 = 370$, $\varphi = 0.30$, $\beta = 0.50$, $a = 4.5$ AND $b = 2.253$

Shift size δ	Explicit Formula Method		Midpoint Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	370.26	0.01	370.99	15.35	0.1756
1.50	9.569	0.01	9.559	15.14	0.1045
1.60	8.720	0.01	8.710	14.36	0.1147
1.70	8.002	0.01	7.994	14.73	0.1000
1.80	7.389	0.01	7.383	14.47	0.0812
1.90	6.863	0.01	6.857	15.11	0.0874
2.00	6.406	0.01	6.401	14.90	0.0781
2.10	6.008	0.01	6.003	14.78	0.0832
2.20	5.657	0.01	5.653	15.04	0.0707
2.30	5.348	0.01	5.344	15.08	0.0748
2.40	5.072	0.01	5.069	15.18	0.0591
2.50	4.826	0.01	4.823	15.19	0.0622
3.00	3.912	0.01	3.910	14.96	0.0511

TABLE VI
COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST MIDPOINT RULE NUMERICAL INTEGRAL EQUATION FOR SARX(1,1)$_{12}$ MODEL GIVEN $ARL_0 = 370$, $\varphi = 0.30$, $\beta = 0.50$, $a = 5.0$ AND $b = 1.732$

Shift size δ	Explicit Formula Method		Midpoint Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	370.05	0.01	370.91	14.53	0.2324
1.50	9.850	0.01	9.841	14.99	0.0914
1.60	8.973	0.01	8.965	15.20	0.0892
1.70	8.231	0.01	8.224	14.55	0.0850
1.80	7.598	0.01	7.592	14.72	0.0790
1.90	7.053	0.01	7.048	15.17	0.0709
2.00	6.581	0.01	6.576	14.54	0.0760
2.10	6.168	0.01	6.164	14.80	0.0649
2.20	5.805	0.01	5.801	14.95	0.0689
2.30	5.484	0.01	5.481	14.90	0.0547
2.40	5.199	0.01	5.196	15.41	0.0577
2.50	4.944	0.01	4.941	15.23	0.0607
3.00	3.996	0.01	3.995	14.82	0.0250

TABLE VII
COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST TRAPEZOIDAL RULE NUMERICAL INTEGRAL EQUATION FOR SARX(1,1)$_{12}$ MODEL GIVEN $ARL_0 = 500$, $\varphi = 0.50$, $\beta = 0.80$, $a = 4.5$ AND $b = 3.110$

Shift size δ	Explicit Formula Method		Trapezoidal Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	500.02	0.01	500.50	15.29	0.0960
1.50	10.140	0.01	10.126	15.32	0.1381
1.60	9.206	0.01	9.194	16.80	0.1303
1.70	8.422	0.01	8.412	16.92	0.1187
1.80	7.756	0.01	7.748	14.67	0.1031
1.90	7.186	0.01	7.179	16.12	0.0974
2.00	6.694	0.01	6.688	16.06	0.0896
2.10	6.266	0.01	6.261	16.18	0.0798
2.20	5.892	0.01	5.886	16.20	0.1018
2.30	5.561	0.01	5.557	17.70	0.0719
2.40	5.268	0.01	5.264	17.56	0.0759
2.50	5.007	0.01	5.004	18.10	0.0599
3.00	4.043	0.01	4.041	15.41	0.0495

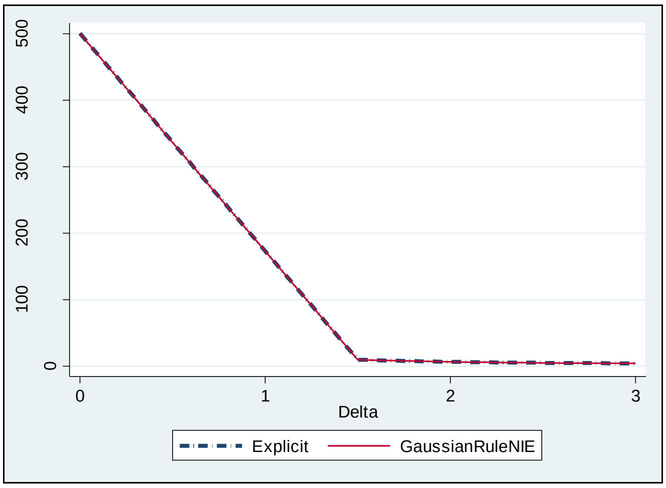

Fig 4 Comparison of ARL of the Explicit formula and the Gaussian rule NIE methods for SARX(1,1)$_4$ model with $ARL_0 = 500$, $a = 3.0$ and $b = 3.592$

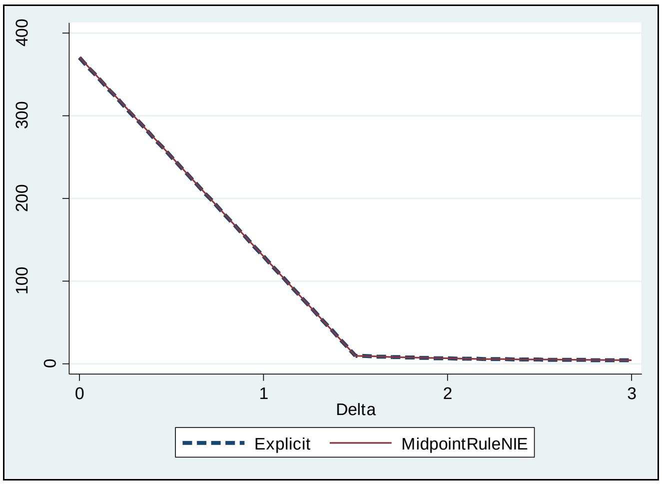

Fig 5 Comparison of ARL of the Explicit formula and the Midpoint rule NIE methods for SARX(1,1)$_{12}$ model with $ARL_0 = 370$, $a = 4.5$ and $b = 2.253$

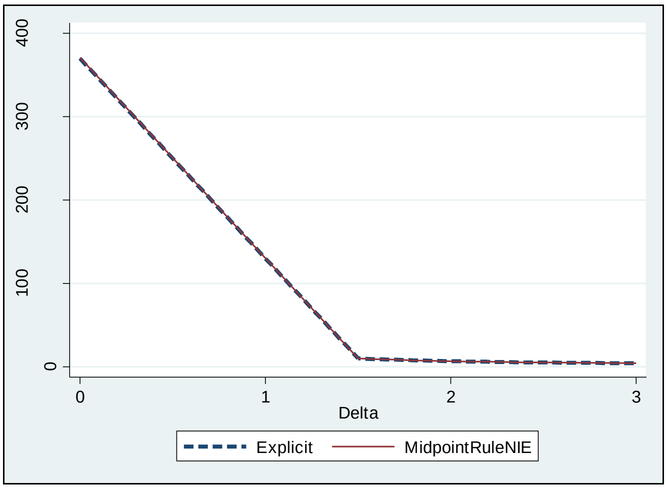

Fig 6 Comparison of ARL of the explicit formula and the Midpoint rule NIE methods for SARX(1,1)$_{12}$ model with $ARL_0 = 370$, $a = 5.0$ and $b = 1.732$

TABLE VIII
COMPARISON OF ARL COMPUTED USING EXPLICIT FORMULA AGAINST TRAPEZOIDAL RULE NUMERICAL INTEGRAL EQUATION FOR SARX(1,1)$_{12}$ MODEL GIVEN $ARL_0 = 500$, $\varphi = 0.50$, $\beta = 0.80$, $a = 5.0$ AND $b = 2.560$

Shift size δ	Explicit Formula Method		Trapezoidal Rule NIE Method		*Diff* (%)
	ARL_{EF}	CPU_{EF}	ARL_{NIE}	CPU_{NIE}	
0.00	500.32	0.01	500.23	18.23	0.0180
1.50	10.673	0.01	10.659	17.83	0.1312
1.60	9.680	0.01	9.669	16.42	0.1136
1.70	8.846	0.01	8.836	17.38	0.1130
1.80	8.138	0.01	8.129	16.62	0.1106
1.90	7.531	0.01	7.523	17.46	0.1062
2.00	7.007	0.01	7.000	16.94	0.0999
2.10	6.551	0.01	6.545	15.64	0.0916
2.20	6.151	0.01	6.146	17.27	0.0813
2.30	5.799	0.01	5.794	16.13	0.0862
2.40	5.487	0.01	5.482	16.25	0.0911
2.50	5.208	0.01	5.204	14.80	0.0768
3.00	4.181	0.01	4.179	15.41	0.0478

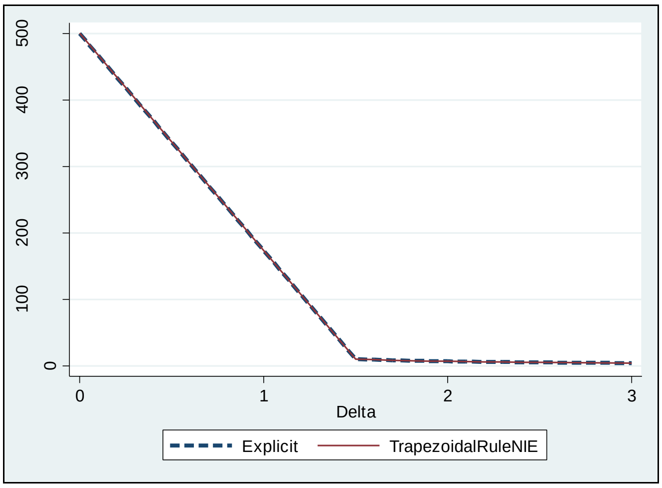

Fig 7 Comparison of ARL of the explicit formula and the Trapezoidal rule NIE methods for SARX(1,1)$_{12}$ model with $ARL_0 = 500$, $a = 4.5$ and $b = 3.110$

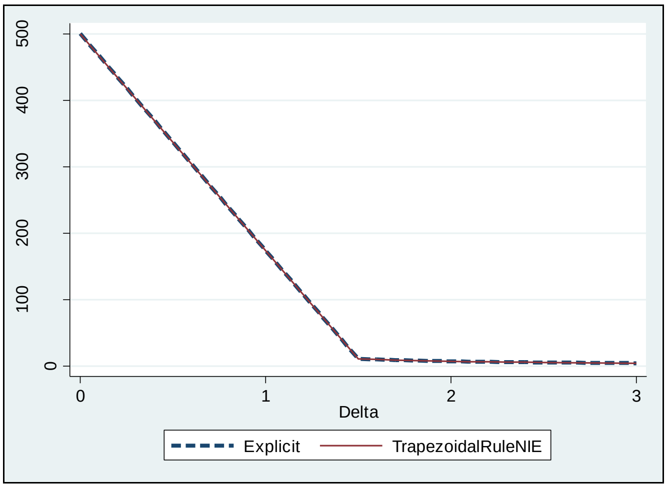

Fig 8 Comparison of ARL of the explicit formula and the Trapezoidal rule NIE methods for SARX(1,1)$_{12}$ model with $ARL_0 = 500$, $a = 5.0$ and $b = 2.560$

V. CONCLUSION

This paper presents the explicit formula and a numerical integral estimation formula for ARL_0 and ARL_1 of CUSUM control chart when observations are seasonal autoregressive models with one exogenous variables (SARX(1,1)$_L$). Our explicit formulas have been demonstrated to be extremely accurate and simple to calculate. We also proposed a numerical integration approach which can be used to approximate the ARL of the CUSUM control chart. In conclusion, the ARL from explicit formulas is close to the numerical integration with an absolute percentage difference of less than 2%. The CPU time for evaluating the proposed explicit formulas is less than 1 second, whereas the Gaussian rule numerical integral equation approach takes over 13-15 minutes, the Midpoint rule numerical integral equation approach takes over 14-15 minutes, and the Trapezoidal rule numerical integral equation approach takes over 15-18 minutes.

REFERENCES

[1] W.A. Shewhart, *Economic Control of Quality of Manufactured Product*. Van Nostrand, New York, 1993.
[2] E.S. Page, "Continuous Inspection Schemes," *Biometrika*, vol. 41, pp 100-114, 1954.
[3] D.M. Hawkins, and D.H. Olwell, *Cumulative Sum Charts and Charting for Quality Improvement*, Springer New York, 1998.
[4] J.M. Lucas, and M.S. Saccucci, "Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements," *Technometrics*, vol. 32, pp1-29, 1990.
[5] C.M. Borror, D.C. Montgomery, and G.C. Runger, "Robustness of the EWMA Control Chart to Non-normality," *Journal of Quality Technology*, vol. 31, pp309-316, 1999.
[6] J. Busaba, S. Sukparungsee, Y. Areepong, and G. Mititelu "Numerical Approximations of Average Run Length for AR(1) on Exponential CUSUM," *Proc. The International Muti Conference of Engineers and Computer Scientists*, Hong Kong, pp1268-1273, March 7-10, 2012.
[7] S. Phanyaem, Y. Areepong, and S. Sukparungsee, "Explicit Formulas of Average Run Length for ARMA(1,1) Process of CUSUM Control Chart," *Far East Journal of Mathematical Sciences*. vol. 90 no. 2, pp 211-224, 2014.
[8] K. Petcharat, S. Sukparungsee, and Y. Areepong, "Exact Solution of the Average Run Length for the Cumulative Sum Chart for a Moving Average Process of Order q," *ScienceAsia*. vol.41, pp141-147, 2015.
[9] P. Paichit, "Average Run Length of Control Chart for ARX(1) Process with Exponential White Noise," *Globle Journal of Pure and Applied Mathematics.* vol. 12 no.3, pp2143-2153, 2016.
[10] S. Phanyaem, "Average Run Length of Cumulative Sum Control Charts for SARMA(1,1)$_L$ Models," *Thailand Statistician.* vol. 15 no. 2, pp184-195, 2017.
[11] L. Zhang, and P. Busababodhin, "The ARIMA(p,d,q) on Upper Sided of CUSUM Procedure," *Lobachevskii Journal of Mathematics*, 2018, vol. 39, no. 3, pp424-432, 2018.
[12] W. Peerajit, and Y. Areepong, "The Performance of CUSUM Control Chart for Monitoring Process Mean for Autoregressive Moving Average with Exogenous Variable Model," *Applied Science and Engineering Progress.* DOI: 10.14416/j.asep.2020.11.007, 2020.
[13] S. Phanyaem, "The Integral Equation Approach for Solving the Average Run Length of EWMA Procedure for Autocorrelated Process," *Thailand Statistician.* vol. 19 no. 3, pp627-641, 2021.
[14] C.W. Champ, and S.E. Rigdon. A comparison of the Markov chain and the integral equation approaches for evaluating the run length distribution of quality control charts," *Communications in Statistics: Simulation and Computation*, vol.20, pp191-204, 1991.

C. Chananet is a Lecturer of Applied Statistics Department, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand.
S. Phanyaem is a Lecturer of Applied Statistics Department, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand.