Journal of Statistical Software

January 2013, Volume 52, Issue 9. http://www.jstatsoft.org/

edcc: An R Package for the Economic Design of the Control Chart

Weicheng Zhu
Chung-Ang University

Changsoon Park
Chung-Ang University

Abstract

The basic purpose of the economic design of the control charts is to find the optimum control charts parameters to minimize the process cost. In this paper, an R package, **edcc** (economic design of control charts), which provides a numerical method to find the optimum chart parameters is presented using the unified approach of the economic design. Also, some examples are given to illustrate how to use this package. The types of the control chart available in the **edcc** package are $\bar{X}$, CUSUM (cumulative sum), and EWMA (exponentially-weighted moving average) control charts.

Keywords: economic design, control charts, optimization, R.

1. Introduction

According to the different definitions of efficiency, the design of control charts can be classified into the statistical design and the economic design. The statistical design aims to minimize the out-of-control (OC) average run length (ARL) when the in-control (IC) ARL is fixed a constant. On the other hand, the economic design aims to minimize the expected cost per hour (ECH) of the control chart where the time and the cost parameters, for example, the IC time, the OC cost, and the false alarm cost, are considered.

Duncan (1956) first introduced the optimum methodology in the $\bar{X}$ chart to obtain the chart parameters, namely subgroup size (`n`), sampling interval (`h`) and control-limit width ($\pm$ `k` standard deviations) for minimizing the process cost. The cost items consist of the sampling and testing costs, the increased cost under the OC process, the false alarm cost and the search and repair costs. Taylor (1968) followed with the economic design of the CUSUM (cumulative sum) charts. Torng, Montgomery, and Cochran (1994) and Ho and Case (1994) independently developed the procedures for economically optimal design of the EWMA (exponentially-weighted moving average) chart.

In this article we introduce and provide an overview of a new R (R Core Team 2012) package, **edcc**, which contains a suite of functions useful when finding the optimum chart parameters of the economic design of the control charts based on Lorenzen and Vance (1986)'s approach. The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=edcc`. In Section 2, a brief review of the theory of economic design of the control charts is provided. In Section 3, the functions contained in our package **edcc** (version 0.2-1) are described. In Section 4, the validity of the functions is confirmed. In Section 5, a classical example was used to show the usage of the functions. Furthermore, we demonstrated how to solve practical problems by means of economic design of the control chart using our package in Section 6.

2. Economic design of the control chart

There are various kinds of parameters considered in the economic design of the control chart. All such parameters are presented by symbols, which are explained as follows:

- h: Sampling interval.
- n: Sample size.
- σ: Standard deviation of observations.
- μ_0: IC process mean.
- μ_1: OC process mean.
- δ: Shift in process mean in standard deviation units when assignable cause occurs ($\delta = |\mu_1 - \mu_0|/\sigma$).
- s: Expected number of samples taken while IC.
- τ: Expected time of occurrence of the assignable cause, given it occurs between the i-th and *(i+1)*-st samples.
- λ: We assume the IC time follows an exponential distribution with mean $\frac{1}{\lambda}$.
- $ARL1$: ARL when the process is IC.
- $ARL2$: ARL when the process has shifted to an OC state.
- T_0: Time to sample and chart one item.
- T_c: Expected time to discover the assignable $c$ause.
- T_f: Expected search time when $f$alse alarm occurs.
- T_r: Expected time to $r$epair the process.
- d_1: Flag for whether the production continues during searches (1 = yes, 0 = no).
- d_2: Flag for whether the production continues during repairs (1 = yes, 0 = no).
- a: Fixed cost per sample.

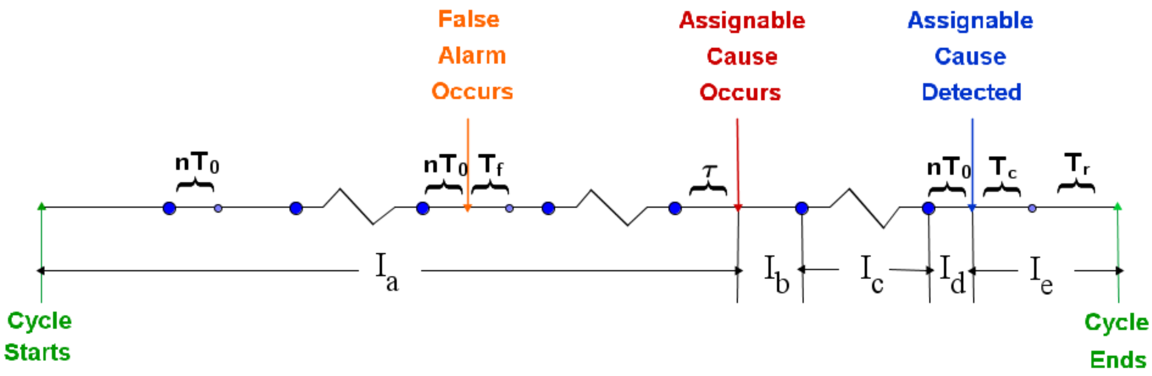

Figure 1: Process cycle.

- b: Cost per unit sampled.
- C_r: Cost for searching and $r$epairing the assignable cause, including any downtime.
- C_f: Cost per $f$alse alarm, including the cost of searching for the cause and the cost of downtime if production ceases during search.
- C_0: Cost per hour due to nonconformities produced while the process is IC.
- C_1: Cost per hour due to nonconformities produced while the process is OC ($C_1 > C_0$).
- P_0: Profit per hour earned by the process operating IC.
- P_1: Profit per hour earned by the process operating OC ($P_0 > P_1$).

The efficiency of the economic design of the control chart is represented by the ECH. For calculation of the ECH, a cycle of the continuous process is defined first as the time from the start of the process until the end of the repair of the process when an assignable cause has occurred. A cycle of the process is illustrated in Figure 1.

The ECH is defined as

$$ECH = \frac{ECC}{ECT}. \tag{1}$$

where ECC stands for the expected cycle cost and ECT stands for the expected cycle time.

2.1. The expected cycle time

As shown in Figure 1, a cycle is partitioned into five sub-time intervals, I_a, I_b, I_c, I_d, and I_e.

I_a: The time until the assignable cause occurs from the start of the cycle.

I_b: The time until the next sample is taken from the end of I_a.

I_c: The time until the chart gives an OC signal from the end of I_b.

I_d: The time to analyze the sample and chart the result from the end of I_c.

I_e: The time to discover the assignable cause and repair the process from the end of I_d.

For each part the expected time is calculated as follows:

$$\mathrm{E}(I_a) = \frac{1}{\lambda} + (1-d_1)\frac{s}{ARL1}T_f \qquad \text{where } s = \frac{e^{-\lambda h}}{1-e^{-\lambda h}}, \tag{2}$$

$$\mathrm{E}(I_b) = h - \tau \qquad \text{where } \tau = \frac{1-(1+\lambda h)e^{-\lambda h}}{\lambda(1-e^{-\lambda h})}, \tag{3}$$

$$\mathrm{E}(I_c) = h(ARL2 - 1), \tag{4}$$

$$\mathrm{E}(I_d) = nT_0, \tag{5}$$

$$\mathrm{E}(I_e) = T_c + T_r. \tag{6}$$

Finally combining Equation 2 to 6, we get the ECT as

$$ECT = \frac{1}{\lambda} + (1-d_1)\frac{s}{ARL1}T_f - \tau + nT_0 + h(ARL2) + T_c + T_r. \tag{7}$$

2.2. The excepted cycle cost

The ECC is mainly incurred due to the following three parts

$C_{\mathbf{(A)}}$**:** nonconformities produced while IC and OC,

$C_{\mathbf{(B)}}$**:** false alarms and locating and repairing of the assignable cause,

$C_{\mathbf{(C)}}$**:** sampling and inspection.

For each part the expected cost is calculated as follows:

$$\mathrm{E}(C_{(\mathrm{A})}) = \frac{C_0}{\lambda} + C_1(-\tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r), \tag{8}$$

$$\mathrm{E}(C_{(\mathrm{B})}) = \frac{sC_f}{ARL1} + C_r, \tag{9}$$

$$\mathrm{E}(C_{(\mathrm{C})}) = (a+bn)\frac{1/\lambda - \tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r}{h}. \tag{10}$$

Adding up Equation 8 to 10 gives the ECC as

$$\begin{aligned} ECC = & \frac{C_0}{\lambda} + C_1(-\tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r) + \frac{sC_f}{ARL1} + C_r \\ & + (a+bn)\frac{1/\lambda - \tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r}{h}. \end{aligned} \tag{11}$$

Sometimes it would be easier to calculate the 'profit' rather than 'cost'. So when using P_0 and P_1 instead of C_0 and C_1, the Expected Cycle Profit (ECP) can be calculated as

$$\begin{aligned} ECP = & \frac{P_0}{\lambda} + P_1(-\tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r) - \frac{sC_f}{ARL1} - C_r \\ & - (a+bn)\frac{1/\lambda - \tau + nT_0 + h(ARL2) + d_1T_c + d_2T_r}{h}. \end{aligned} \tag{12}$$

Thus ECH can be defined using Equation 7 and 12 as

$$ECH = P_0 - \frac{ECP}{ECT}. \tag{13}$$

It should be noted that the ECH is a function of the control chart parameters, because ARL1 and ARL2 are functions of the control chart parameters too. So the problem of optimization is the search for the optimum values of the control chart parameters for which the ECH is the minimum.

2.3. Implementation in R

We will take advantage of R package **spc** (Knoth 2012) to calculate ARL1 and ARL2 for CUSUM and EWMA control charts and use the `stats::optim` function to calculate the chart parameters which minimize the ECH.

Besides the optimization algorithm, this package also provides the so-called 'grid method' in optimizing ECH. When the domain and design points of all chart parameters are given, the ECH for each design point within the domain is calculated, and then the design point with the minimum ECH is treated as the optimum point. More details are presented in next section.

3. The R package edcc

The R package **edcc** comes with a namespace. The index of the package is listed as follows:

- `ecoXbar`: Economic design for $\bar{X}$ control chart.
- `ecoCusum`: Economic design for CUSUM control chart.
- `ecoEwma`: Economic design for EWMA control chart.
- `edcc-class`: Class '`edcc`'.
- `update`: S3 method, update for an '`edcc`' class object.
- `contour`: S3 method, contour plot of an '`edcc`' class object.

More details about them are described below.

3.1. Function `ecoXbar`

The control procedure of the $\bar{X}$ chart for detecting positive and negative shifts is to give an out-of-control signal if $|\frac{\bar{X}-\mu_0}{\sigma/\sqrt{n}}| \geq L$. The function `ecoXbar` calculates the optimum parameters, `n` (sample size), `h` (sampling interval) and `L` (number of standard deviations from control limits to the center line) for the economic design of the $\bar{X}$ control chart. It is used as:

```
ecoXbar(h, L, n, lambda = 0.05, delta = 2, P0 = NULL, P1 = NULL,
  C0 = NULL, C1 = NULL, Cr = 25, Cf = 50, T0 = 0.0167, Tc = 1,
  Tf = 0, Tr = 0, a = 1, b = 0.1, d1 = 1, d2 = 1, nlevels = 30,
  sided = "two", par = NULL, contour.plot = FALSE, call.print = TRUE,
  ...)
```

The arguments of the function are described as follows:

- `h`: Sampling interval. It can be a numeric vector or left undefined.
- `L`: Number of standard deviations from control limits to the center line. It can be a numeric vector or left undefined.
- `n`: Sample size. It can be an integer vector or left undefined.
- `lambda`: We assume the in-control time follows an exponential distribution with mean 1/`lambda`. Default value is `0.05`.
- `delta`: Shift in process mean in standard deviation units when assignable cause occurs (`delta` $= (\mu_1 - \mu_0)/\sigma$), where σ is the standard deviation of observations; μ_0 is the in-control process mean; μ_1 is the out-of-control process mean. Default value is `2`.
- `P0`: Profit per hour earned by the process operating in control.
- `P1`: Profit per hour earned by the process operating out of control (`P0` > `P1`).
- `C0`: Cost per hour due to nonconformities produced while the process is in control.
- `C1`: Cost per hour due to nonconformities produced while the process is out of control (`C1` > `C0`).
- `Cr`: Cost for searching and *r*epairing the assignable cause, including any downtime.
- `Cf`: Cost per *f*alse alarm, including the cost of searching for the cause and the cost of downtime if production ceases during search.
- `T0`: Time to sample and chart one item.
- `Tc`: Expected time to discover the assignable *c*ause.
- `Tf`: Expected search time when *f*alse alarm occurs.
- `Tr`: Expected time to *r*epair the process.
- `a`: Fixed cost per sample.
- `b`: Cost per unit sampled.
- `d1`: Flag for whether production continues during searches (`1` = yes, `0` = no). Default value is `1`.
- `d2`: Flag for whether production continues during repairs (`1` = yes, `0` = no). Default value is `1`.
- `nlevels`: Number of contour levels desired. Default value is `30`. It works only when `contour.plot = TRUE`.
- `sided`: Distinguish between one- and two-sided $\bar{X}$ chart by choosing `"one"` or `"two"` respectively. When `sided = "one"`, `delta > 0` means the control chart for detecting a positive shift, and vice versa. Default is `"two"`.

- `par`: Initial values for the parameters to be optimized over. It can be a vector of length 2 or 3.
- `contour.plot`: A logical value indicating whether a contour plot should be drawn. Default is `FALSE`. Only works when the parameters `h`, `L` and `n` are all specified.
- `call.print`: A logical value indicating whether the 'call' should be printed on the contour plot. Default is `TRUE`.
- `...`: Other arguments to be passed to `optim` function.

It maybe embarrassing to see so many arguments, but in fact they are easy to understand and easy to remember. Usually the parameters only need to be specified once and then take advantage of the `update` function to update the arguments which makes the work quite easy. Examples will be shown later.

It is important to figure out when the function will use the optimization algorithm and when will use the 'grid method' to get the optimum parameters. When parameters `h`, `L`, `n` are all undefined, `ecoXbar` function tries to find the global optimum point to minimize the ECH using the optimization algorithm. When `h` and `L` are undefined but the domain of `n` is given as an integer vector, `ecoXbar` function tries to find the optimum point for each `n` value using the optimization algorithm. When the domains of `h`, `L` and `n` are all given, `ecoXbar` function will use the 'grid method' to calculate the optimum point, that is the ECH for all the combinations of the parameters will be calculated. The 'grid method' is much slower than using the optimization algorithm, but it would be a good choice when optimization algorithm fail to converge.

This function returns an object of class '`edcc`', which is a list of elements `optimum`, `cost.frame`, `FAR` and `ATS`; `optimum` is a vector with the optimum parameters and the corresponding ECH value; `cost.frame` is a dataframe with the optimum parameters and the corresponding ECH values for all given `n` (if `n` is not specified, `cost.frame` will not be returned); `FAR` indicates the *false alarm rate* during the IC time, which is calculated as $\lambda\times$(average number of false alarm); `ATS` indicates the *average time to signal* after the occurrence of an assignable cause, calculated as $h \times ARL2 - \tau$. For function `ecoCusum` and `ecoEwma`, the return values have the same structure with `ecoXbar`.

When you want to calculate the ECH for only one single design point using the economic design of the $\bar{X}$ chart, you can use `echXbar` function ,which shares all the important arguments with `ecoXbar`, but the returned value is simpler, only one single real number is returned.

3.2. Function `ecoCusum`

The control procedure of the CUSUM chart for detecting positive shifts is to give an out-of-control signal if $T_i^+ > H$; for detecting negative shift is to give an out-of-control signal if $T_i^- > H$; for detecting both positive and negative shifts is to give an out-of-control signal if either $T_i^+ > H$ or $T_i^- > H$; where $T_i^+ = \max\{0, T_{i-1}^+ + (\frac{\bar{X}_i-\mu_0}{\sigma/\sqrt{n}} - k)\}$, $T_i^- = \max\{0, T_{i-1}^- - (\frac{\bar{X}_i-\mu_0}{\sigma/\sqrt{n}} + k)\}$ and $k = \frac{\sqrt{n}\delta}{2}$. The function `ecoCusum` calculates the optimum parameters, `n` (sample size), `h` (sampling interval) and `H` (decision interval) for economic design of the CUSUM control chart. It is used as:

```
ecoCusum(h, H, n, delta = 2, lambda = 0.01, P0 = NULL, P1 = NULL,
  C0 = NULL, C1 = NULL, Cr = 20, Cf = 10, T0 = 0, Tc = 0.1,
  Tf = 0.1, Tr = 0.2, a = 0.5, b = 0.1, d1 = 1, d2 = 1, nlevels = 30,
  sided = "one", par = NULL, contour.plot = FALSE, call.print = TRUE,
  ...)
```

Most of the arguments are the same as `ecoXbar`, and the different ones are:

- `H`: Decision interval. It can be a numeric vector or left undefined.
- `sided`: Distinguish between one-, two-sided and Crosier's modified two-sided CUSUM scheme by choosing `"one"`, `"two"`, and `"Crosier"`, respectively. See details in `spc::xcusum.arl`.

Similar to `ecoXbar` function, when parameters `h`, `H`, `n` are all undefined, `ecoCusum` function tries to find the global optimum point to minimize the ECH using the optimization algorithm (`optim` function). When `h` and `H` are undefined but the domain of `n` is given as an integer vector, `ecoCusum` function tries to find the optimum point for each `n` value using the optimization algorithm. When `h`, `H` and `n` are all given, `ecoCusum` function will use the 'grid method' to calculate the optimum point.

One may note that an important parameter of CUSUM control chart, `K` (reference value) is missing in this function. The reason is that there is strong numerical and theoretical evidences that for given ARL2, the value of ARL1 approaches its maximum when `K` is chosen as the mid-point between the acceptance quality level (AQL) and the rejectable quality level (RQL); that is, $K = (\mu_0 + \mu_1)/2$ (see Chiu 1974, for further details). For this reason we fix $K = (\mu_0 + \mu_1)/2$ and optimize the other chart parameters, `n`, `h` and `H`.

When you want to calculate the ECH for only one single design point using the economic design of the CUSUM chart, you can use `echCusum` function, which shares all the important arguments with `ecoCusum`, but the returned value is simpler, only one single real number is returned.

3.3. Function `ecoEwma`

The control procedure of the EWMA chart for detecting positive and negative shifts is to give an out-of-control signal if $|\frac{S_i - \mu_0}{\sigma/\sqrt{n}}| \geq k\sqrt{\frac{w}{2-w}}$, where $S_i = w\bar{X}_i + (1-w)S_{i-1}$, $S_0 = \mu_0$. The function `ecoEwma` calculates the optimum parameters, `n` (sample size), `h` (sampling interval), `w` (weight to the present sample) and `k` (number of standard deviations from control limits to the center line) for the economic design of the EWMA control chart. It is used as:

```
ecoEwma(h = seq(0.7, 1, by = 0.1), w = seq(0.7, 1, by = 0.1),
  k = seq(2, 4, by = 0.1), n = 4:8, delta = 2, lambda = 0.05,
  P0 = NULL, P1 = NULL, C0 = NULL, C1 = NULL, Cr = 25, Cf = 10,
  T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1, d1 = 1,
  d2 = 1, nlevels = 30, sided = "two", par = NULL, contour.plot = FALSE,
  call.print = TRUE, ...)
```

Most of the arguments are the same as `ecoXbar`, and the different ones are:

- `w`: The weight value between 0 and 1 given to the latest sample. It must be specified.
- `k`: Control limit coefficient. It can be a numeric vector or left undefined.
- `sided`: Distinguish between one- and two-sided EWMA control chart by choosing `"one"` and `"two"` respectively. See details in `spc::xewma.arl`.

Different from other arguments, `w` should always be given, because the range of `w` is so restricted that optimization algorithms usually do not converge. When parameters `h`, `k`, `n` are all undefined, `ecoEwma` function tries to find the global optimum point to minimize the ECH using the optimization algorithm (`optim` function). When `h` and `k` are undefined but the domain of `n` is given as an integer vector, `ecoEwma` function tries to find the optimum point for each `n` value using the optimization algorithm. When the domains of `h`, `k` and `n` are all given, `ecoEwma` function will use the 'grid method' to calculate the optimum point.

When you want to calculate the ECH for only one single design point using the economic design of the CUSUM chart, you can use `echEwma` function, which shares all the important arguments with `ecoEwma`, but the returned value is simpler, only one single real number is returned.

3.4. Class 'edcc'

The 'edcc' class objects can be created by calling the `ecoXbar`, `ecoCusum` or `ecoEwma` function. Three S3 methods functions in this package are defined on this class: `contour`, `update` and `print`.

3.5. Function update

The function `update` is a S3 method function for an 'edcc' class object. It will update and (by default) refit a model. It does this by extracting the call stored in the object, updating the call and (by default) evaluating that call. It is used as:

```
update(object, ..., evaluate = TRUE)
```

The arguments of the function are described as follows:

- `object`: A 'edcc' class object returned by calling `ecoXbar`, `ecoCusum` or `ecoEwma` function.
- `...`: Additional arguments to the call, or arguments with changed values.
- `evaluate` If true evaluate the new call else return the call.

There are too many arguments need to be specified using the `ecoXbar`, `ecoCusum` and `ecoEwma` functions. So taking advantage of the `update` function would save you a lot of time during the model building procedure.

3.6. Function contour

The function `contour` is a S3 method function used to create contour plot for an 'edcc' class object. It is used as:

```
contour (x, call.print = TRUE, ...)
```

where `x` should be a ‘`edcc`’ class object ; `call.print` controls whether the R command should be printed on the contour plot; `...` is passed to `graphics::contour` function.

4. Validity confirmation

Some examples of the published papers are used to verify the correctness of the functions in **edcc**.

First, we verify the `ecoXbar` function using the result from Chung (1990). Three examples from Table 2 of the paper are selected. The results are shown in Table 1, where the results of the **edcc** package are exactly the same as Chung's results with negligibly small errors.

Next, we verify the `ecoCusum` function using the result from Chiu (1974). Three examples from Table 3 of the paper are selected. The results are shown in Table 2, where the results of the **edcc** package are exactly the same as Chiu's results with negligibly psmall errors.

Last, we verify the `ecoEwma` function using the result from Ho and Case (1994). Again, three examples from the *Table 1* of the paper are selected. The results are shown in Table 3, where the results of the **edcc** package are exactly the same as Ho and Case's results with negligibly small errors.

Case	**edcc** solutions				Paper solutions			
	h	L	n	ECH	h	L	n	ECH
1	1.41	3.08	5	4.0128	1.41	3.08	5	4.0129
5	0.41	2.95	4	26.9753	0.41	2.95	4	26.9761
21	23.62	2.16	38	0.8308	24.24	2.16	38	0.8310

Table 1: Comparison of results by **edcc** and Chung (1990).

Case	**edcc** solutions				Paper solutions			
	h	H	n	ECH	h	H	n	ECH
1	1.41	0.56	5	2.261	1.41	0.56	5	2.261
2	0.64	0.55	5	6.973	0.65	0.55	5	6.973
14	2.62	0.37	30	4.208	2.62	0.37	30	4.208

Table 2: Comparison of results by **edcc** and Chiu (1974).

Case	**edcc** solutions					Paper solutions				
	h	k	n	w	ECH	h	k	n	w	ECH
1	1.4062	3.0881	5	0.94	4.0113	1.3956	3.1047	5	0.9394	4.0114
5	0.4030	2.9738	4	0.90	26.9539	0.4102	2.9671	4	0.9032	26.9545
21	22.5938	2.1728	36	0.85	0.8282	23.2301	2.1827	37	0.8531	0.8283

Table 3: Comparison of results by **edcc** and Ho and Case (1994).

5. A classical example

First, we present a most classical example, *The Glass Bottles*, introduced in (Montgomery 2009, pp. 469–471) to illustrate the usage of the functions in this package.

> A manufacture produces nonreturnable glass bottles for packaging a carbonated soft-drink beverage. The wall thickness of the bottles is an important quality characteristic. If the wall is too thin, internal pressure generated during the filling will cause the bottle to burst.
>
> Based on an analysis of quality control technicians' salaries and the costs of test equipment, it is estimated that the fixed cost of taking a sample is USD 1. The variable cost of sampling is estimated to be USD 0.01 per bottle, and it takes approximately 1min (0.0167h) to measure and record the wall thickness of a bottle.
>
> The process is subject to several different types of assignable causes. However, on the average, when the process goes out of control, the magnitude of the shift is approximately two standard deviations. Process shifts occur at random with a frequency of about one every 20h of operation. Thus, the exponential distribution with parameter $\lambda = 0.05$ is a reasonable model of the run length in control. The average time required to investigate an out-of-control signal is 1 h. The cost of investigating an action signal that results in the elimination of an assignable cause is USD 25, whereas the cost of investigating a false alarm is USD 50.
>
> The bottles are sold to a soft-drink bottler. If the walls are too thin, an excessive number of bottles will burst when they are filled. When this happens, the bottler's standard practice is to backcharge the manufacturer for the costs of cleanup an lost production. Based on this practice, the manufacturer estimates that the penalty cost of operating in the out-of-control state for one hour is USD 100.

Accordingly we know `a = 1`, `b = 0.1`, `Cr = 25`, `Cf = 50`, `T0 = 0.0167`, `Tc = 1`, `Tf = 0`, `Tr = 0`, `P0 = 110`, `P1 = 10`, `d1 = 1`, `d2 = 1`. Some of the values are approximately used, for example, in the function `ecoXbar`, `Cr` stands for the cost for searching and repairing the assignable cause, but in this material only the searching cost is mentioned, so we just specify `Cr` as the searching cost here; expected search time for false alarm is not mentioned, we use `Tf = 0` instead; expected time to repair the process is not mentioned, we use `Tr = 0` instead; we assume the production continues during both searches and repairs, which indicate `d1 = 1`, `d2 = 1`; the penalty cost of operating in the OC state for one hour is USD 100, which means $\texttt{P0} - \texttt{P1} = 100$, so we set `P0 = 110` and `P1 = 10` here, in fact only if $\texttt{P0} - \texttt{P1}$ keeps the same, the results do not change.

We use the economic design of the $\bar{X}$, CUSUM and EWMA charts to implement this example, and compare the results of the three methods.

5.1. Example: `ecoXbar`

First we use the optimization algorithm, so just leave `h`, `L` and `n` undefined.

```
R> (x <- ecoXbar(lambda = 0.05, delta = 2, P0 = 110, P1 = 10, Cr = 25,
+    Cf = 50, T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1, d1 = 1,
+    d2 = 1))
```

```
$optimum
 Optimum h  Optimum L  Optimum n        ECH
 0.8146052  2.9813756  5.0000000 10.3670006

$FAR
[1] 0.003451395

$ATS
[1] 0.4695136
```

The above results show that the best chart parameters are `h = 0.81`, `L = 2.98`, `n = 5` and the minimum ECH is USD 10.367. The result is slightly better than that of the Montgomery (2009)'s book: `h = 0.76`, `n = 5`, `L = 2.99`, and cost USD 10.38.

The 'grid method' is to specify the arguments `h`, `L` and `n`, thus function `ecoXbar` will calculate ECH values for all combinations of the parameters and select the optimum point, which of course takes more time than using the optimization algorithm. We illustrate this method using `update` function here.

```
R> x1 <- update(x, h = seq(0.7, 0.9, by = 0.01),
+    L = seq(2.8, 3.2, by = 0.01), n = 4:6,
+    contour.plot = TRUE, call.print = FALSE)
R> x1

$optimum
Optimum h Optimum L Optimum n       ECH
  0.81000   2.98000   5.00000  10.36708

$cost.frame
 Optimum h Optimum L Optimum n      ECH
      0.77      2.83         4 10.48951
      0.81      2.98         5 10.36708
      0.85      3.13         6 10.38023

$FAR
[1] 0.003487046

$ATS
[1] 0.4666755
```

As we note, the results of 'grid method' are almost identical with that of the optimization algorithm. We specify `contour.plot = TRUE` to generate a contour plot shown in Figure 2, contour plot is helpful in figuring out the shape of the ECH around the optimum point.

5.2. Example: `ecoCusum`

We use the same steps to find out the optimum point of the CUSUM chart as the previous section. So first we use the optimization algorithm:

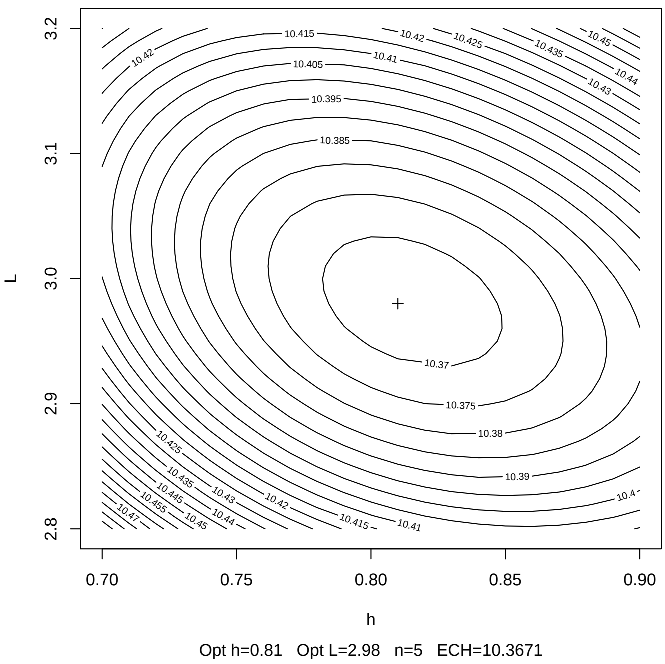

Figure 2: Contour plot of `ecoXbar`.

```
R> (y <- ecoCusum(lambda = 0.05, delta = 2, P0 = 110, P1 = 10, Cr = 25,
+     Cf = 50, T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1, d1 = 1,
+     d2 = 1, sided = "two"))

$optimum
 Optimum h  Optimum H  Optimum n        ECH
 0.8128006  0.7671013  5.0000000 10.3611214

$FAR
[1] 0.003285769

$ATS
[1] 0.4689758
```

We get the best chart parameters are `h = 0.81`, `H = 0.77`, `n = 5` and the minimum ECH is USD 10.361.

Either we can use the 'grid method', although unnecessary for this case, we still list it as:

```
R> y1 <- update(y, h = seq(0.75, 0.9, by = 0.01),
+     H = seq(0.6, 0.9, by = 0.01), n = 4:6,
+     contour.plot = TRUE, call.print = FALSE)
R> y1

$optimum
Optimum h Optimum H Optimum n       ECH
  0.81000   0.77000   5.00000  10.36114
```

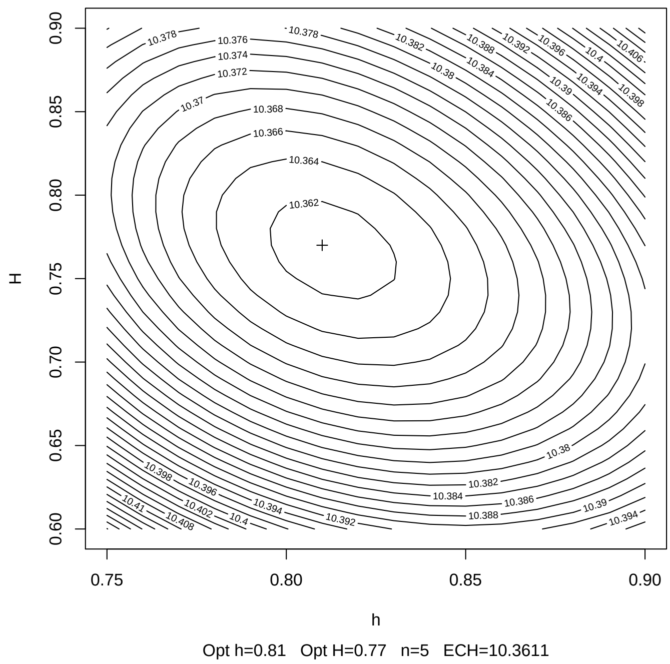

Figure 3: Contour plot of `ecoCusum`.

```
$cost.frame
 Optimum h Optimum H Optimum n       ECH
      0.76      0.88         4 10.46955
      0.81      0.77         5 10.36114
      0.85      0.69         6 10.37844

$FAR
[1] 0.003266555

$ATS
[1] 0.4676873
```

The result is identical with the optimization algorithm. The contour plot is shown in Figure 3.

5.3. Example: `ecoEwma`

The use of function `ecoEwma` is a little more complicated than the functions `ecoXbar` and `ecoCusum`, because there is one more argument, `w`, to be optimized. The following code tries to find the location of the optimum `w` value where the minimum ECH may exist.

```
R> z <- ecoEwma(w = seq(0.1, 1, by = 0.1), lambda = 0.05, delta = 2,
+    P0 = 110, P1 = 10, Cr = 25, Cf= 50, T0 = 0.0167, Tc = 1, Tf = 0,
+    Tr = 0, a = 1, b = 0.1, d1 = 1, d2 = 1)
R> z

$optimum
```

```
 Optimum h  Optimum k  Optimum n  Optimum w        ECH
 0.8282283  3.0311392  5.3426142  1.0000000 10.3603658

$cost.frame
 Optimum h Optimum k Optimum n Optimum w      ECH
 1.0425911  2.017433  9.573487       0.1 11.62842
 0.9416400  2.509054  8.106431       0.2 11.11996
 0.8936145  2.728392  7.085852       0.3 10.83632
 0.8657535  2.850289  6.418526       0.4 10.66056
 0.8483392  2.925022  5.970261       0.5 10.54421
 0.8372298  2.972906  5.667958       0.6 10.46501
 0.8303430  3.003967  5.470132       0.7 10.41171
 0.8267789  3.022460  5.357692       0.8 10.37827
 0.8260608  3.031458  5.315691       0.9 10.36159
 0.8282283  3.031139  5.342614       1.0 10.36037

$FAR
[1] 0.002881128

$ATS
[1] 0.465852
```

Note that the optimum value of `w` is near 1, `n` is around 5, `k` is around 3 and `h` is around 0.8. We would directly use the 'grid method' here since from the results above we can lock a region to search.

```
R> z1 <- update(z, h = seq(0.7, 0.9, by = 0.01), w = seq(0.8, 1, by = 0.01),
+    k = seq(2.9, 3.1, by = 0.01), n = 4:6, contour.plot = TRUE,
+    call.print = FALSE)
R> z1
```

```
  $optimum
Optimum h Optimum k Optimum n Optimum w       ECH
  0.81000   2.99000   5.00000   0.95000  10.36482

$cost.frame
 Optimum h Optimum k n Optimum w      ECH
      0.75      2.90 4      0.91 10.48318
      0.81      2.99 5      0.95 10.36482
      0.86      3.10 6      0.97 10.38031

$FAR
[1] 0.003373639

$ATS
[1] 0.4673824
```

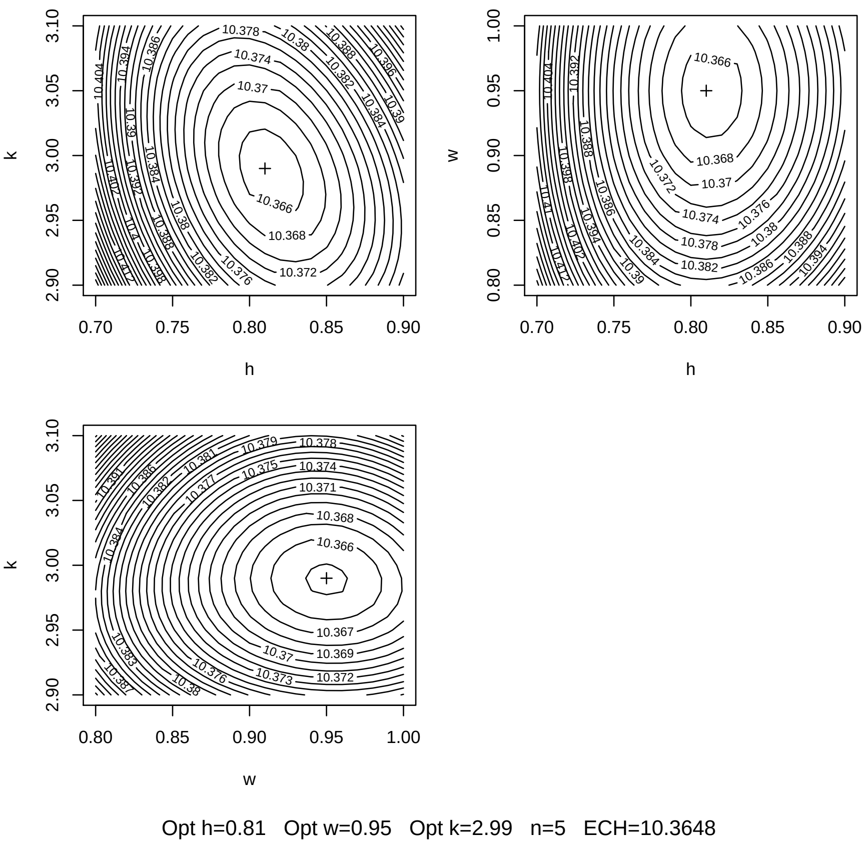

Figure 4: Contour plot of `ecoEwma`.

Finally we get the best chart parameters are `h = 0.81, w = 0.95, k=2.99, n = 5` and the minimum ECH is USD 10.365. The contour plot is shown in Figure 4.

5.4. Comparison of results

We list all of the results showed in previous subsections in Table 4. Among the three methods we used, economic CUSUM design seems to work best since its ECH is the smallest one. We may note that no matter which method is used the optimum `n` and `h` values almost keep the same. We can also note that the weight value `w` in function `ecoEwma` is very close to 1 and `k` value is very close to `L`, so in this case economic design of the EWMA chart is almost identical with economic design of the $\bar{X}$ chart.

`ecoXbar`	`n`	`h`	`L`		ECH
	5	0.81	2.98		10.367
`ecoCusum`	`n`	`h`	`H`		ECH
	5	0.81	0.77		10.361
`ecoEwma`	`n`	`h`	`k`	`w`	ECH
	5	0.81	2.99	0.95	10.365

Table 4: Results comparison.

6. An industrial application: KERASTAR tiles

This section presents a detailed industrial application of economic design of the control chart. *The Tile Quality Monitoring* example first introduced by Nikolaidis, Rigas, and Tagaras (2007) is used to show how to apply the economic desigon of control chart to the real world problems.

6.1. The product and the production process

The type of tile that was selected for this study is KERASTAR $30 \times 30cm^2$, unglazed tiles produced by Philkeram-Johnson s.a. (PJ for brevity) factory in Greek.

The production process of this kind of tile is briefly described as follows (for more details about this process, refer to Nikolaidis *et al.* 2007):

- Pretreatment of raw material: All raw materials are mixed, grinded and dried to 'dry' dust.
- Tile formation stage: Tiles are formed by a press using the appropriate dies. Three tiles are produced in each cycle (strike).
- Drying and firing: Tiles are first dried, and after ensuring an acceptably low moisture level, they are driven into the kiln.

This study concentrates on the tile formation stage. The press operation is controlled mainly by measuring the penetrability of tiles at fixed time intervals. The level of penetrability is connected with the dimensions and shape of tiles coming out from the kiln at a subsequent stage of the production line.

The penetrability of a tile is measured at three points on each side of the tile for a total of eight measurements, as shown in Figure 5. The sum of the three measurements of each side is calculated and then the difference between the largest and the smallest of these four sums is computed and recorded as the value of the controlled variable (quality characteristic) X for that tile. This random variable expresses the dissimilarity of press conditions, according to PJ's know-how and practice.

The target value for the quality characteristic X is obviously 0, as that value typically implies the greatest possible uniformity of compression by the press. If the value of X does not exceed

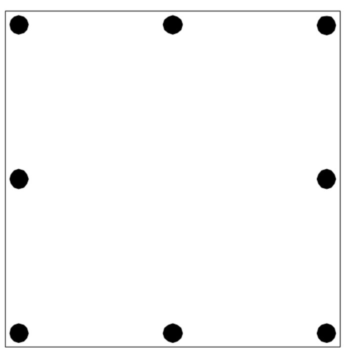

Figure 5: Points where penetrability is measured on a square tile.

0.05 mm, then the tile is considered A class; if the value is between 0.05 and 0.1 mm, the tile is classified as B class; finally, if the value of X exceeds 0.1 mm, then such a tile is characterized as scrap, due to the faulty press conditions that eventually result in unacceptable departure from the required orthogonal shape.

6.2. Quality monitoring procedure in tile formation

The currently used quality monitoring procedure in the tile formation stage is described as follows: For every h hours n $(1 \leq n \leq 3)$ tiles can be collected from each strike and their penetrability is measured. If $\bar{X}$ is lower than the control limit, then the production process continues its operation without intervention or else it is considered that there is an indication of possible disturbance (occurrence of an assignable cause) on the press operation, then the entire press operation is immediately interrupted and the maintenance technician is called. Note that a one-sided control chart for monitoring positive shifts is considered here. If the indication is indeed correct, the technician proceeds with the necessary adjustment of the pressing mechanism and turns on the press, having ensured that the assignable cause has been removed. If the alarm is false, the maintenance technician can easily and quickly identify it as such and he turns on the press immediately.

The quality characteristic X behaves as a normally distributed random variable with mean $\mu = 0.0308$ mm and standard deviation $\sigma = 0.0092$ mm when the process is in control. The normality assumption was verified by means of a standard χ^2 goodness-of-fit test. The values of the parameters μ and σ were estimated using a large number of data from the company's database.

The duration of a production run is typically either 72 hours or 112 hours. For simplicity, we assume the production process operates indefinitely. At the beginning of every production run the press is operating well and the mean value of X is equal to the nominal value $\mu_0 = 0.0308$ mm.

6.3. Parameters specification

In order to find the best monitoring scheme from an economic point of view, we should specify all necessary parameters first.

- According to the records of the factory, the mean time between successive occurrences of the assignable cause is approximately 20 days, which translates to a mean rate of occurrence equal to 0.0021 per hour. The probability distribution of that time is not sufficiently documented. For the purposes of this study, it was assumed that the occurrence time follows an exponential distribution ($\lambda = 0.0021$).
- The press operation is subject to the occurrence of a single assignable cause, which increases the mean value of X without significantly affecting σ , which is therefore considered constant ($\sigma = 0.0092$ mm). According to the analysis of measurements in the out-of-control state, the mean of X increases from $\mu_0 = 0.0308$ mm to $\mu_1 = \mu_0 + \delta\sigma \approx 0.0446$ mm, which implies that the approximate magnitude of the shift in the mean is $\delta = 1.5$ standard deviations.
- The sampling cost per unit consists of two parts: the cost of sampling and measurement (labor and materials) and the cost of the sample tiles, since the measurement is

destructive. The estimate for the total sampling cost per unit is b = EUR 0.56. For simplicity we let the fixed cost per sample is $a = 0$.

- Operation of the process in the out-of-control state incurs a cost due to the production of an increased percentage of B class tiles; the percentage of non-conforming tiles is negligible even when the process is out of control. Taking into consideration the profit from the production and sale of A class tiles and that of B class tiles (15% lower profit than for A class tiles), the average cost of out-of-control operation (profit reduction) is estimated at C_1 = EUR 52.80 per hour. Because when the process is in control the total expected cost related to monitoring the process is almost 0, so C_0 = EUR 0.

- Every time the monitoring scheme erroneously suggests the occurrence of an assignable cause, the time required for the maintenance technician to assert that the signal was indeed false is approximately 10 min, thus $T_f = 1/6$ hour, accordingly we also suppose the expected time to discover the assignable cause is $T_c = 1/6$ hour here. As mentioned in Park (2012), the weakness of economic design of the control charts is mainly due to a possible excessively large number of false signals, but such incidences can be avoided by assigning high false signal costs. For this purpose, set the cost of a false alarm as two times of C_1, that is C_f = EUR 105.6.

- When the control chart correctly indicates the occurrence of an assignable cause, the adjustment of the pressing mechanism takes on average 45 min, thus $T_r = 3/4$ hour. Taking into account the relevant labor costs and the cost of measurements required to ensure that the adjustment was successful, the estimate of the restoration cost is C_r = EUR 16.84.

- Since the time spent in sampling and charting one item is negligible, we set it as $T_0 = 0$ hour.

- As it is mentioned in Section 6.2, when out of control signal is given, the entire press operation is immediately interrupted. So $d_1 = d_2 = 0$.

Thus we have found out all necessary parameters in calculating the optimum design parameters.

6.4. The choice of the chart type

There are three well-known chart types, practitioners may wonder which one to use. The chart type has been chosen traditionally by the statistical efficiency according to the amount of shift to detect: When the amount of shift to detect is substantially large or more ($\delta \geq 1.5\sigma$), $\bar{X}$ is used, otherwise CUSUM or EWMA chart is prefered. Besides, the CUSUM chart performs better when the distribution of the process variable is known and the EWMA chart is more robust than the others.

However, in the economic design, it would be suggested to try all of the three chart types and compare the ECHs directly. One thing to note is that the outcomes are just theory optimum values, you should always combine the results with the reality, such as give more restrictions to the arguments to meet your requests.

6.5. Implementation

Firstly, the economic design of the $\bar{X}$ chart is performed using `ecoXbar` with a restriction `n` ≤ 3 (since three tiles are produced at each strike) as follows:

```
R> ecoXbar(n = 1:3, lambda = 0.0021, delta = 1.5, C0 = 0, C1 = 52.8,
+    T0 = 0, Tf = 1/6, Tc = 1/6, Tr = 3/4, d1 = 0, d2 = 0, Cf = 105.6,
+    Cr = 16.84, a = 0, b = 0.56, sided = "one")

$optimum
Optimum h Optimum L Optimum n       ECH
 4.440508  2.382580  3.000000  1.200600

$cost.frame
 Optimum h Optimum L Optimum n      ECH
  1.571077  2.426821         1 1.771073
  3.063392  2.378137         2 1.366087
  4.440508  2.382580         3 1.200600

$FAR
[1] 0.001926778

$ATS
[1] 5.36979
```

From the above results we know that the $\bar{X}$ chart will be optimized with ECH $= 1.201$ when `h = 4.44`, `L = 2.38` and `n = 3` are used.

Next, the economic design of the CUSUM chart is performed as follows:

```
R> ecoCusum(n = 1:3, lambda = 0.0021, delta = 1.5, C0 = 0, C1 = 52.8,
+    T0 = 0, Tf = 1/6, Tc = 1/6, Tr = 3/4, d1 = 0, d2 = 0, Cf = 105.6,
+    Cr = 16.84, a = 0, b = 0.56, sided = "one")

$optimum
Optimum h Optimum H Optimum n       ECH
 3.967289  1.407082  3.000000  1.137429

$cost.frame
 Optimum h Optimum H Optimum n      ECH
  1.154082  3.407265         1 1.231337
  2.474811  2.022927         2 1.185271
  3.967289  1.407082         3 1.137429

$FAR
[1] 0.001199237

$ATS
[1] 5.068349
```

From the above results, the CUSUM chart is optimized with ECH = 1.137 when `h = 3.97`, `H = 1.41` and `n = 3` are used.

Finally, the economic design of the EWMA chart is performed as follows:

```
R> a <- ecoEwma(n = 1:3, w = seq(0.1, 1, by = 0.1), lambda = 0.0021,
+    delta = 1.5, C0 = 0, C1 = 52.8, T0 = 0, Tf = 1/6, Tc = 1/6, Tr = 3/4,
+    d1 = 0, d2 = 0, Cf = 105.6, Cr = 16.84, a = 0, b = 0.56, sided = "one")
R> a$optimum

Optimum h Optimum k Optimum n Optimum w       ECH
 4.001052  2.593417  3.000000  0.700000  1.176290
```

The code above is used to catch the optimum region of `w`, then we use the `update` function to get more precise results by setting a smaller interval for `w`:

```
R> a1 <- update(a, w = seq(0.6, 0.8, by = 0.01))
R> a1$cost.frame <- NULL
R> a1

$optimum
Optimum h Optimum k Optimum n Optimum w       ECH
 4.058130  2.572139  3.000000  0.740000  1.175667

$FAR
[1] 0.001469887

$ATS
[1] 5.24934
```

From the above results, the EWMA chart is optimized with ECH = 1.176 when `h = 4.06`, `k = 2.57`, `n = 3` and `w = 0.74` are used.

Comparing the three chart types by the ECH, we suggest to use the CUSUM chart with sampling interval `h = 4` hours, decision interval `H = 1.4` and sample size `n = 3` to monior the process. Note that the performances of the three types are not much different from each other.

Additionally, as an extension of this example, we decrease `delta = 0.5` and all other parameters remain the same to see how the results change:

For the $\bar{X}$ control chart:

```
R> ecoXbar(n = 1:3, lambda = 0.0021, delta = 0.5, C0 = 0, C1 = 52.8,
+    T0 = 0, Tf = 1/6, Tc = 1/6, Tr = 3/4, d1 = 0, d2 = 0, Cf = 105.6,
+    Cr = 16.84, a = 0, b = 0.56, sided = "one")

$optimum
Optimum h Optimum L Optimum n       ECH
 3.833360  1.679788  3.000000  3.463123
```

```
$cost.frame
 Optimum h Optimum L Optimum n      ECH
  1.826099  1.867745         1 4.200316
  2.892537  1.747225         2 3.737460
  3.833360  1.679788         3 3.463123

$FAR
[1] 0.02416283

$ATS
[1] 16.05405
```

For the CUSUM control chart:

```
R> ecoCusum(n = 1:3, lambda = 0.0021, delta = 0.5, C0 = 0, C1 = 52.8,
+    T0 = 0, Tf = 1/6, Tc = 1/6, Tr = 3/4, d1 = 0, d2 = 0, Cf = 105.6,
+    Cr = 16.84, a = 0, b = 0.56, sided = "one")

$optimum
Optimum h Optimum H Optimum n       ECH
 1.778829  3.427894  3.000000  2.864148

$cost.frame
 Optimum h Optimum H Optimum n      ECH
 0.5667945  6.940050         1 2.917616
 1.1600311  4.510438         2 2.890728
 1.7788286  3.427894         3 2.864148

$FAR
[1] 0.004367996

$ATS
[1] 13.38979
$ATS
[1] 5.068349
```

For the EWMA control chart:

```
R> b <- ecoEwma(n = 1:3, w = seq(0.1, 1, by = 0.1), lambda = 0.0021,
+    delta = 0.5, C0 = 0, C1 = 52.8, T0 = 0, Tf =1/6, Tc = 1/6, Tr = 3/4,
+    d1 = 0, d2 = 0, Cf = 105.6, Cr = 16.84, a = 0, b = 0.56, sided = "one")
R> b$optimum

Optimum h Optimum k Optimum n Optimum w       ECH
 1.832254  2.288473  3.000000  0.200000  2.909938

R> b1 <- update(b, w = seq(0.1, 0.3, by = 0.01))
R> b1$cost.frame <- NULL
R> b1
```

```
$optimum
Optimum h Optimum k Optimum n Optimum w       ECH
 1.787243  2.283902  3.000000  0.170000  2.906996

$FAR
[1] 0.004610554

$ATS
[1] 13.60751
```

From the above results we can conclude that the CUSUM and the EWMA charts perform better than the $\bar{X}$ chart in detecting small shifts of the process mean, but the choice between the CUSUM and the EWMA depends on the operator with which he/she prefers. Note that the difference in the ECH between $\bar{X}$ chart and the CUSUM (or the EWMA) is significantly large. Thus, the CUSUM or the EWMA should be used for cases where small amount of the mean shift is to be detected.

7. Summary

It has been almost sixty years since Duncan (1956) first introduced the concept of economic design of the control chart, and many theories and methods about this field have been developed successfully during this period of time. However, no software for the economic design of the control chart has been known to authors. Our contribution of the R package **edcc** will make the economic design of the control chart more accessible to applied researchers or plant engineers. We hope that it is easy for users to learn this package both from theory and practice.

For further works, the economic statistical design of the control chart proposed by Saniga (1989) should be implemented to the **edcc** package.

Acknowledgments

We are very grateful to the associate editor and two reviewers for their thoughtful and constructive comments. Weicheng Zhu's research was supported by Chung-Ang University Young Scientist Scholarship (2010). Changsoon Park's research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-003545).

References

Chiu WK (1974). "The Economic Design of Cusum Charts for Controlling Normal Means." *Journal of the Royal Statistical Society C*, **23**(3), 420–433.

Chung K (1990). "A Simplified Procedure for the Economic Design of $\bar{X}$-Chart." *International Journal of Production Research*, **28**(7), 1239–1246.

Duncan AJ (1956). "The Economic Design of Charts Used to Maintain Current Control of a Process." *Journal of the American Statistical Association*, **51**, 228–242.

Ho C, Case KE (1994). "The Economically-Based EWMA Control Chart." *International Journal of Production Research*, **32**, 2179–2186.

Knoth S (2012). ***spc**: Statistical Process Control*. R package version 0.4.2, URL http://CRAN.R-project.org/package=spc.

Lorenzen TJ, Vance LC (1986). "The Economic Design of Control Charts: A Unified Approach." *Technometrics*, **28**(1), 3–10.

Montgomery DC (2009). *Statistical Quality Control – A Modern Introduction*. John Wiley & Sons.

Nikolaidis Y, Rigas G, Tagaras G (2007). "Using Economically Designed Shewhart and Adaptive $\bar{X}$ Charts for Monitoring the Quality of Tiles." *Quality and Reliability Engineering International*, **23**(2), 233–245.

Park C (2012). "Economic Design of $\bar{X}$ Charts When Signals May Be Misclassified and the Bounded Reset Chart." *IIE Transactions*, **45**(4), 436–448.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Saniga EM (1989). "Economic Statistical Control-Chart Designs With an Application to $\bar{X}$ and R Charts." *Technometrics*, **31**(3), 313–320.

Taylor HM (1968). "The Economic Design of Cumulative Sum Control Charts." *Technometrics*, **10**, 479–448.

Torng JCC, Montgomery DC, Cochran JK (1994). "Economic Design of The EWMA Control Chart." *Economic Quality Control*, **9**, 3–23.

Affiliation:

Weicheng Zhu, Changsoon Park
Department of Statistics
Chung-Ang University
156-756 Huksuk-Dong, Dongjak-Gu, Seoul, Korea
E-mail: mingsnu@gmail.com, cspark@cau.ac.kr
URL: http://dreamhunter.me, http://cau.ac.kr/~cspark/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 52, Issue 9 *Submitted:* 2012-04-01
January 2013 *Accepted:* 2012-10-29