Approximation for In–control and Out–of–control Average Run Lengths of Cusums

Douglas M. Hawkins

Technical Report #545

March 1990

A Fast Accurate Approximation for In-control and Out-of-control Average Run Lengths of Cusums

Douglas M Hawkins
School of Statistics
University of Minnesota
St Paul MN

Cumulative sum charts (cusums) are very effective in detecting special causes which persist until remedied. In designing a cusum. it is important to know the run lengths corresponding to various possible choices of cusum mask parameters wshile the process is in control, and also when it goes out of control. This paper provides a relatively simple yet very accurate (typically within 1%) representation of the ARL of an in-control cusum. It is pointed out that this representation also gives the ARL's for out-of-control situations. Some uses of the approach are illustrated - evaluating the ARL of specific parameter values; finding values that give a desired ARL; and evaluating the out-of-control ARL's of location and scale cusums. A computer program embodying the approximation is also given.

Footnote Dr Hawkins is a Professor in the Department of Applied Statistics. He is a member of ASQC.

Introduction

Consider a process giving a measurement X which, while the process is in control, follows a normal distribution with mean μ and standard deviation σ and is independent from one observation time to the next. We will write $X \sim N(\mu,\sigma^2)$ as shorthand for the statement 'X follows a normal distribution with mean μ and standard deviation σ'. Write X_j for the value of X observed at observation time j. The decision interval cusum to check for the constancy of the mean of X is defined as follows. First, a standardized version of X is computed:-

$U_j = \frac{(X_j - \mu)}{\sigma}$, which in control, follows a standard normal distribution N(0,1).

Then define the location cusums as

$$L_0^+ = L_0^- = 0$$
$$L_j^+ = \max(0, L_{j-1}^+ + U_j - k_+)$$
$$L_j^- = \min(0, L_{j-1}^- + U_j - k_-)$$

If L_j^+ exceeds h_+, then the process is diagnosed as out of control, the mean of X having increased from the in-control level μ. Similarly, if L_j^- is less than $-h_-$, the process is diagnosed as out of control, its mean having decreased from the in-control level μ. This cusum is defined by the parameters k (called the 'allowances' or the 'reference values'); and h (the 'decision intervals').

The performance of the cusum is commonly measured by the average run length, or ARL. This depends on the h and k parameters, larger values of both corresponding to longer runs. Though it is theoretically possible to use only one of the decision interval cusums L^+, L^- (thereby screening out any signals of a mean shift in the direction of the other cusum), this is not common. Thus a run of the whole cusum scheme ends when one or the other cusum goes out of control. Writing ARL_+ and ARL_- for the average run lengths of these two component cusums, and ARL for the overall average run length, we have that (see for example Van Dobben de Bruyn 1968)

$$\frac{1}{ARL} = \frac{1}{ARL_+} + \frac{1}{ARL_-}.$$

It is easily seen that the average run length of the L^- cusum with its parameters k_- and h_- say is exactly the same as the ARL of a positive cusum with the same parameters. It is thus sufficient (and conventional) to concentrate on evaluating just ARL_+, from which ARL_- can be inferred.

A second design question relates to the performance of the cusum when the process goes out of control - ie if the process does undergo a shift in mean and/or standard deviation, what ARL ensues before the shift is detected. We will return to this issue after competing discussion of the in-control ARL problem.

The formulation given above is very general; in most applications the parameters k are the same for the + and - cusums as are the parameters h (ie $k_+=k_-=k$ say; and $h_+=h_-=h$ say) and in most of what follows we will assume that there is a single k and a single h parameter for both cusums.

The accepted wisdom is that when deciding on suitable values of h and k, one should start by choosing for k one-half the value of a mean shift one is most interested in detecting. Suppose for example that the in-control measurement is N(30,100), and that it is of interest to detect as quickly as possible a shift in mean from 30 to 35. Expressing the shift of 5 units as a multiple of the in-control standard deviation, shows that this shift from 30 to 35 would be a shift of 0.5 standard deviations. The recommended choice of k is then one-half this value, or 0.25.

Having chosen k, h is then selected by deciding what ARL is desired while the process is in control. Finding the h value which for that k gives the desired ARL then completes the design.

It is not a trivial matter to find the ARL corresponding to a given choice of h and k. One approach (outlined for example by Van Dobben de Bruyn) uses an integral equation arising from renewal theory. Brook and Evans (1972) sketched what is probably the most popular approach today - that of writing the cusum as a Markov process, approximating its transition probabilities, and then using the theory of Markov chains to evaluate the ARL. This remains a cumbersome computation however.

Some tables and graphs of the ARL are given in Lucas (1976). These provide the in-control and out-of-control ARL of cusums with a range of values from k=.25 to 1.5, with the range of h varying with k. While these tables are valuable for the ARL's corresponding to actual entries, they fall short of the ideal for design in several respects. First, they cover a limited range of k and h values. Second, they have quite large spacings between the tabled values, creating problems for evaluating the ARL for parameter values not listed explicitly. While some form of interpolation could presumably be used, it is not clear what interpolation, or what errors this would introduce. Finally, while there is some discussion of the numerical error to which the ARL's are computed, this error is not directly controlled or reported.

The starting point of our work was a more closely-spaced table of ARL values. This covered h values from 0_+ to 8 in steps of 0.5, and k values from -0.75 to 2.00 in steps of 0.125 (the reason for including negative values of k will emerge later). The table was produced by a refinement of Brook and Evans' method, and (except for ARL's in excess of 10^8 where subtractive cancellation started to become an issue) the ARL's were accurate to at least 0.1%

Next, an effort was made to try to reproduce the table with some simple formula. One very attractive approach to modeling a rectangular table with entries Y_{hk} say is Mandel's model (see for example Bradu 1984) -

$$Y_{hk} = \alpha_h + \beta_k + \xi_h \eta_k + \xi_h^* \eta_k^* \qquad (1)$$

The terms α_h, ξ_h and ξ_h^* are specific to the h^{th} row; the terms β_k η_k and η_k^* to the k^{th} column.

The ARL's studied ranged from slightly over 1 to 10^{10}, so a representation like (1) to provide a good fit to the ARL data would clearly require some sort of transformation. Thus the first analysis was an attempt to find some transformation of the ARL's to values that would be more amenable to this simple modeling.

Several transformations were tried, but the most successful was an inverse normal transformation. Define Φ^{-1} to be the inverse normal transformation - ie $\Phi^{-1}(p)$ is the standard normal value that has area p to the left of it. Let ARL_{hk} be the average run length of the cusum with parameters h and k, and define

$$Y_{hk} = -\Phi^{-1}(1/ARL_{hk}).$$

It was found that Y_{hk} was very accurately represented over the entire range by a Mandel model (1). The coefficients α, β, ξ, η, ξ^* and η^* were fitted to the table by least squares, ignoring ARL's in excess of 1 million. The resulting coefficients are given in Table 1.

Using the formula

The basic method of using this model is as follows:-

(i) for the selected h and k, obtain the necessary values of β_k, η_k, η^*_k, α_h, ξ_h and ξ^*_h.

(ii) From these compute

$$Y_{hk} = \alpha_h + \beta_k + \xi_h\eta_k + \xi^*_h\eta^*_k$$

Finally, (iii) obtain the ARL from a table or algorithm for the cumulative normal, as $1/\Phi(-Y_{hk})$.

We will illustrate this first with two hand calculations. Suppose we wish to design a cusum for maximum sensitivity to a shift on 0.75 standard deviations in mean. For this, the choice k=0.375 is indicated. A guess of a suitable h value might be h = 5. The table has entries for this choice of k and h. Looking these up, we get

$$\begin{aligned} Y_{hk} &= \alpha_h + \beta_k + \xi_h \times \eta_k + \xi^*_h \times \eta^*_k \\ &= 2.0134 + 0.8543 + (-0.3293)\times 0.2952 + 0.0986\times(-0.1440) \\ &= 2.756. \end{aligned}$$

From normal tables, $\Phi(-2.76) = 0.00289$, so taking entering the table with just two digits gives ARL = 1/0.00289 = 346. If we are using two-sided cusumming (L^+ and L^-), then the overall ARL will be half this, or 173. The exact value of the ARL is very close - it is 171.

As another manually computed example, suppose one wanted the ARL for k=0.375 but h=5.20, where we have table entries for k, but not for h. For this calculation, make a linear interpolation of the α, ξ and ξ^* values, getting the following figures:-

$$\begin{aligned} Y_{hk} &= \alpha_h + \beta_k + \xi_h \times \eta_k + \xi^*_h \times \eta^*_k \\ &= 2.0792 + 0.8543 + (-0.3886)\times 0.2952 + 0.0861\times(-0.1440) \\ &= 2.806. \end{aligned}$$

From normal tables, $\Phi(-2.81) = 0.00248$, so taking the normal table to two digits gives the ARL as 403, or 201 for two-sided testing. The exact ARL is 200. In both these examples, the major source of error turns out to be, not the approximation inherent in the model, but reading the Y value to only two decimals in the normal table.

Interpolating for k, and interpolating for both h and k proceeds in essentially the same way.

While this hand calculation is acceptable if only a few ARL computations are to be done, for more frequent use, it is helpful to use a computer program. Such a program is given in the Appendix. The program has two modes of operation - in the first, the user supplies the h and k values of interest, and the program calculates the ARL using the Mandel model. In the second mode, the user supplies k and the required ARL of the cusum, and the program computes the h value needed to achieve that ARL.

As an illustration of the use of the second option, suppose as before that we have decided on k=0.375, but want to set the cusum so that the in-control ARL is 300. With a two-sided cusum, this means that ARL_+ must be 600. Given these values as input, the program computes the value h=5.723 as providing this ARL.

Accuracy of the overall approximation

The approximation was tested by taking 500 randomly selected h and k values in the range (0,8) and (-0.75,2) respectively. At each the ARL was calculated exactly, and using the Mandel approximation, and the percentage difference between the exact and approximate ARL's found. Among the ARL's less than 100,000 (the range of greatest practical interest) the error had a standard deviation of 0.7%. This figure increased to 3% for ARL's in the range 10^5 to 10^9, but even this larger figure is more than accurate enough for practical purposes.

Out-of-control ARL's

The second question of interest is the ARL when the data depart from control. In control, the data follow a $N(\mu,\sigma^2)$ distribution; let us suppose that the mean, the standard deviation or perhaps both change - specifically that the distribution of the X_j changes to $N(\mu+\Delta\sigma,\tau^2\sigma^2)$.

When this shift occurs, the distribution of the standardized quantity U_j changes from N(0,1) to

$U_j \sim N(\Delta,\tau^2)$. Another way of saying this is that

$U_j = \Delta+\tau W_j$, where $W_j \sim N(0,1)$

Returning to the definition, the cusums L^+ and L^- were given by

$$L_0^+ = 0; \; L_0^- = 0$$
$$L_j^+ = \max(0, L_{j-1}^+ + U_j - k)$$
$$L_j^- = \min(0, L_{j-1}^- + U_j + k.$$

Rewriting these cusums in terms of the W_j, these latter expressions become

$$L_j^+ = \max(0, L_{j-1}^+ + \{\tau W_j + \Delta\} - k)$$
$$L_j^- = \min(0, L_{j-1}^- + \{\tau W_j + \Delta\} + k, \text{ or regrouping terms.}$$
$$L_j^+ = \max(0, L_{j-1}^+ + \tau W_j - \{k-\Delta\})$$
$$L_j^- = \min(0, L_{j-1}^- + \tau W_j + \{k+\Delta\}).$$

From inspection, it can be seen that $L_j^+ = \tau M_j^+$, and $L_j^- = \tau M_j^-$, where the decision interval cusums M^+ and M^- are defined by

$$M_0^+ = 0; \; M_0^- = 0$$
$$M_j^+ = \max(0, M_{j-1}^+ + W_j - [\{k-\Delta\}/\tau])$$
$$M_j^- = \min(0, M_{j-1}^- + W_j + [\{k+\Delta\}/\tau]),$$

and that $L_j^+ > h$ if and only if $M_j^+ > h/\tau$; $L_j^- < -h$ if and only if $M_j^- < -h/\tau$.

In other words, the out-of-control ARL of the cusum L^+ equals the in-control ARL of the cusum M^+ whose allowance is $(k-\Delta)/\tau$ and decision interval h/τ. The out-of-control ARL of the cusum L^- equals the in-control ARL of a (+) cusum with allowance $(k+\Delta)/\tau$ and decision interval h/τ.

The implication of this identity is that there is no need for separate formulas for in-control and out-of-control ARL's, and that our approximation may therefore also be used to find the ARL for out-of-control situations.

Example: The cusum with h=5.723, k=0.375 and in-control ARL 300 was designed for detection of a shift of 0.75 standard deviations. Let us evaluate this cusum when the process mean shifts by 0.75 standard deviation while the standard deviation remains fixed - ie set $\Delta=0.75$, $\tau=1$. The ARL's of the L^+ and L^- cusums are obtained by taking the in-control ARL's of cusums with decision interval h=5.723, and with allowances k=(0.375-0.75) = -0.375 and k=(0.375+0.75) = 1.125 respectively. These ARL's are given by the program as 14.8 and 1.9×10^6 respectively.

Not surprisingly, the ARL to the signal on L^- is astronomically large, and may be ignored, giving the overall ARL as 14.8.

This picture changes if the standard deviation of the data also shifts. To illustrate this, suppose that the process mean shifts by 0.75 standard deviations and that simultaneously the standard deviation increases by a factor of 2. Then the ARL's of the L^+ and L^- parameters are given by in-control ARL's with decision interval 5.723/2 = 2.862 and allowances -0.375/2 = -0.1875 and 1.125/2 = 0.5625 respectively. This changes the ARL to 10.5 on L^+, and 136 on L^- The overall ARL is 9.7, and it becomes a possibility for the increased mean to be signaled by L^- and not L^+.

As another illustration of the use of the approximation, it was mentioned that conventional wisdom is to tune the cusum by setting k to one half the shift Δ of most concern, though there is no guarantee that this choice will give the best out-of-control performance. With the program, it is easy to explore this recommendation. For example, the following choices of k and h all give an in-control ARL of 300. For each choice the out-of-control ARL when the mean shifts by 0.75 is also listed:-

k	0.350	0.355	0.360	0.365	0.370	0.375	0.380	0.385
h	6.015	5.955	5.895	5.837	5.780	5.723	5.666	5.609.
Non-null ARL	14.91	14.89	14.87	14.86	14.85	14.84	14.85	14.86

Despite the lack of a theoretical guarantee, for this design the shortest ARL does indeed occur with k=0.375, though the profile of ARL's is very flat and any k in the range would give effectively the same performance.

Performance of a scale cusum

We have concentrated on the cusum for location. Hawkins (1981) discussed a cusum for scale, based on the observation that with U ~ N(0,1),

$\sqrt{|U_j|} \sim N(0.822, 0.349^2)$ to a very close approximation, so that

$V_j = \dfrac{[\sqrt{|U_j|} - 0.822]}{0.349}$ is close to N(0,1).

Cusums of the V_j are sensitive to shifts in scale. If the variance of the X_j increases, then a cusum of V_j will show an upward shift in mean, while if the variance decreases, there will be a downward shift. As V_j is close to N(0,1), it is convenient and sensible when designing to use the same k and h parameters for its cusum as for the location cusum of the U_j, and to plot both cusums on the same chart. The out-of-control performance of the scale cusum is then of interest.

To investigate this, suppose the standard deviation of the original X changes by a factor of ρ, while the mean remains fixed. It is then an easy calculation that the distribution of V_j changes to

$$V_j \sim N[\frac{0.822(\rho^{\frac{1}{2}}-1)}{0.349}, \rho] = N[2.355(\rho^{\frac{1}{2}}-1), \rho].$$

The performance of the scale cusum can therefore be checked using the present program, setting $\Delta=2.355(\rho^{\frac{1}{2}}-1)$ and $\tau=\rho^{\frac{1}{2}}$.

Suppose for example that the cusum k=0.375, h=5.723 is used, and that the standard deviation of the data increases by a factor of 1.5. This gives τ=1.22 and Δ=0.529. Thus the out-of-control ARL will be the in-control ARL of a cusum with allowance (0.375-0.529)/1.22 = -0.126, and decision interval 5.723/1.22 = 4.691. The program gives this ARL as 22.1.

Conclusion

Many problems relating to the design and evaluation of cusum schemes are greatly facilitated by having a fast, accurate, easily computed approximation to the average run length of a cusum. The approximation developed in this paper seems to meet the requirements of easy computation coupled with surprisingly high accuracy. Over a wide range of design parameters, the ARL's given by the approximation are generally within 1% of the exact values, and even at the margins of applicability the error remains of the order of 3%.

The utility of the approximation has been illustrated with several design calculations involving the null and non-null behavior of cusums for location and scale.

References

Bradu, D., (1984) 'Response surface model diagnosis in two-way tables', Communications in Statistics, A13, 3059-3103.

Brook D., and and Evans, D. A., (1972), 'An approach to the probability distribution of cusum run length', Biometrika, 59, 539-549.

Hawkins D. M., (1981) 'A cusum for a scale parameter', Journal of Quality Technology, 13, 228-231.

Lucas, J. M., (1976), 'The design and use of V-mask control schemes', Journal of Quality Technology, 8, 1-11.

Van Dobben de Bruyn, C. S., (1968), 'Cumulative Sum Tests: Theory and Practice' Griffin, London.

Table 1. Terms in the Mandel model

k	β_k	η_k	η_k^*
-0.750	-0.6781	0.7320	0.1163
-0.625	-0.5774	0.7459	0.0826
-0.500	-0.4679	0.7541	0.0626
-0.375	-0.3469	0.7541	0.0568
-0.250	-0.2104	0.7417	0.0623
-0.125	-0.0532	0.7104	0.0689
0.000	0.1311	0.6515	0.0561
0.125	0.3469	0.5585	0.0054
0.250	0.5914	0.4352	-0.0734
0.375	0.8543	0.2952	-0.1440
0.500	1.1245	0.1528	-0.1810
0.625	1.3939	0.0161	-0.1829
0.750	1.6580	-0.1117	-0.1594
0.875	1.9148	-0.2299	-0.1212
1.000	2.1628	-0.3384	-0.0721
1.125	2.4018	-0.4376	-0.0183
1.250	2.6330	-0.5306	0.0291
1.375	2.8580	-0.6199	0.0619
1.500	3.0765	-0.7053	0.0820
1.625	3.2891	-0.7877	0.0871
1.750	3.4904	-0.8600	0.0871
1.875	3.6882	-0.9321	0.0688
2.000	3.8731	-0.9943	0.0253

h	α_h	ξ_h	ξ_h^*
0.00	-0.8165	1.0577	-0.1732
0.50	-0.3487	1.0331	-0.1417
1.00	0.0592	0.9532	-0.0879
1.50	0.4140	0.8230	-0.0323
2.00	0.7238	0.6595	0.0284
2.50	0.9963	0.4823	0.0893
3.00	1.2375	0.3077	0.1314
3.50	1.4537	0.1423	0.1399
4.00	1.6527	-0.0185	0.1336
4.50	1.8387	-0.1761	0.1206
5.00	2.0134	-0.3293	0.0986
5.50	2.1782	-0.4776	0.0673
6.00	2.3349	-0.6217	0.0296
6.50	2.4838	-0.7606	-0.0173
7.00	2.6263	-0.8954	-0.0698
7.50	2.7630	-1.0263	-0.1271
8.00	2.8945	-1.1534	-0.1894

Appendix

```
      program qwkarl
c     ******************************************************************
c     Program for cusum average run lengths.
c     Functions called - (i) phi(x) returns the normal integral
c     evaluated over the rqnge -infinity to x
c     (ii) phinv(p) returns the normal quantile - ie the value
c     whose normal integral is p.
c     ******************************************************************
      implicit double precision (a-h,o-z)
      write(*,101)
 101  format(///' QWKARL program for average run lengths of cusums'/
     1 ' Douglas M Hawkins, University of Minnesota, 1990'/
     2 ' You can use the program in two ways:'/' (i) Enter h',
     3 ' k and value 1 to compute ARL for given h and k, or '/
     4 ' Enter ARL, k, and value 2 to compute h giving that ARL'/
     5 ' Enter zero values for all to stop the run')
 1    write(*,*) ' Enter arguments'
      read (*,*) h, ak, idoc
      if (h .eq. 0 .and. ak .eq. 0) stop
      if (idoc .ne. 2) then
        valu = pre(h,ak)
        arl = 1.d0 / phi(-valu)
        write(*,'('' h,k,arl'',3f8.3,g20.10)') h,ak,arl
        go to 1
        else
        arl = h
        targ = -phinv(1.d0/arl)
        guesl = 0
        guesh = 8
        fnl = pre(guesl,ak)
        fnh = pre(guesh,ak)
        if (fnl .gt. targ .or. fnh .lt. targ) then
          write(*,'('' Value '',g15.6,'' out of range'',
     1      2g15.6)') targ,fnl,fnh
          go to 1
          endif
        do 3 loop = 1, 14
        guesm = (guesl + guesh) / 2
        fnm = pre(guesm,ak)
        if (fnm .gt. targ) then
          guesh = guesm
          else
          guesl = guesm
          endif
 3      continue
        write(6,'('' h,k,arl'',3f8.3,g20.10)') guesm,ak,arl
        go to 1
        endif
      end
```

```
      function pre(h,ak)
      implicit double precision (a-h,o-z)
      dimension row(17,3),col(23,3),rowint(3),colint(3)
      data row/-0.8165,-0.3487, 0.0592, 0.4140, 0.7238, 0.9963, 1.2375,
     +  1.4537, 1.6527, 1.8387, 2.0134, 2.1782, 2.3349, 2.4838, 2.6263,
     +  2.7630, 2.8945, 1.0577, 1.0331, 0.9532, 0.8230, 0.6595, 0.4823,
     +  0.3077, 0.1423,-0.0185,-0.1761,-0.3293,-0.4776,-0.6217,-0.7606,
     + -0.8954,-1.0263,-1.1534,-0.1732,-0.1417,-0.0879,-0.0323, 0.0284,
     +  0.0893, 0.1314, 0.1399, 0.1336, 0.1206, 0.0986, 0.0673, 0.0296,
     + -0.0173,-0.0698,-0.1271,-0.1894/,   col/-0.6781,-0.5774,-0.4679,
     + -0.3469,-0.2104,-0.0532, 0.1311, 0.3469, 0.5914, 0.8543, 1.1245,
     +  1.3939, 1.6580, 1.9148, 2.1628, 2.4018, 2.6330, 2.8580, 3.0765,
     +  3.2891, 3.4904, 3.6882, 3.8731, 0.7320, 0.7459, 0.7541, 0.7541,
     +  0.7417, 0.7104, 0.6515, 0.5585, 0.4352, 0.2952, 0.1528, 0.0161,
     + -0.1117,-0.2299,-0.3384,-0.4376,-0.5306,-0.6199,-0.7053,-0.7877,
     + -0.8600,-0.9321,-0.9943, 0.1163, 0.0826, 0.0626, 0.0568, 0.0623,
     +  0.0689, 0.0561, 0.0054,-0.0734,-0.1440,-0.1810,-0.1829,-0.1594,
     + -0.1212,-0.0721,-0.0183, 0.0291, 0.0619, 0.0820, 0.0871, 0.0871,
     +  0.0688, 0.0253/
       if (h .lt. 0 .or. h .gt. 8 .or. ak .lt. -.75 .or. ak .gt. 2)
     1   then
        write(*,*) 'Value out of range (0,8) for h, (-0.75,2) for k',
     1   h, ak
        return
        endif
       h2 = h * 2 + 1
       ih = h2
       ak8 = 8 * (ak + .75) + 1
       k8  = ak8
       oh = h2 - ih
       ohi = 1.d0 - oh
       ok = ak8 - k8
       oki = 1.d0 - ok
       do 3 i = 1, 3
       rowint(i) = row(ih,i) * ohi + row(ih+1,i) * oh
       colint(i) = col(k8,i) * oki + col(k8+1,i) * ok
3      continue
       pre = rowint(1) + colint(1) + rowint(2) * colint(2) +
     1  rowint(3) * colint(3)
       return
       end
```