



品質管理における管理図係数表の真値の計算方法

稲葉, 太一
田中, 逸郎

(Citation)

神戸大学大学院人間発達環境学研究科研究紀要, 6(2):19-23

(Issue Date)

2013-03

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/81005345>

(URL)

<https://hdl.handle.net/20.500.14094/81005345>



品質管理における管理図係数表の真値の計算方法

Method for calculating true values on factors for computing control chart lines at Quality Control

稲葉 太一* 田中 逸郎**
 Taichi INABA* Itsuro TANAKA**

要約：品質管理における計量値データに関する管理図の管理限界線を求めるために不可欠な表として「管理図係数表」がある。これらの計算式や計算方法は確立しているが、従来から複数の数値表において、発行者によって「表示桁」が違ったり、末尾の桁が食い違ったりしており、一部では用いる数値表によって判断結果が変わるといった混乱が起こる可能性がある。この論文では、これらの数値の真値を実際に計算した表を与える。

1. 問題の背景

管理図の係数表は、日本規格協会(2010)[1]には、日本工業規格(JIS)の「シューハート管理図(1998)」として記載がある。また、頻繁に用いられる数値に関しては、日本科学技術連盟が出版している森口(2012)[2]にも記載がある。これらの数値表は、いずれも、ASTM(American Social Testing and Materials)に準拠していると明記されている。しかし、ASTMに記載されている数値が発行年によって異なっており、一部の数値の末尾に不一致が見られる。このことは、工程での異常の有無判定において結論が逆転する可能性があり、起こる頻度は僅かではあるが混乱が生じている。

従来、品質管理学界では、日本規格協会と日本科学技術連盟の2つの組織が独立に機能しており、互いに切磋琢磨して発展してきたが、この数年、品質管理検定(QC検定)の共同実施に代表される融合が進んでいる。このような融合期にあつて、その基本となる数値表に齟齬があるのは、非常に不都合である。

例えば、統計ソフトでは、このように複数の数値表がある場合、自前のプログラムで計算した数値を採用せず、権威があると思われる方の数値表の表示値を、そのまま採用していることがある。この場合には、用いるソフトによって結果が異なる危険性がある。

従来の上記の2つの数値表には、今回の詳細な検討の結果、いずれも若干の問題があることが分かったので、この論文では、正しいと考えられる数値表を作成する。また、2章では計算式を述べ、3節では具体的な数値表を記載し、4章ではその誤差評価の概要を示す。なお、山内(1972)[3]に記載されている数値を利用した計算結果と、その記載されている値の数値積分による確認も行ふ。この確認

プログラムを付録に記載する。

2. 管理図に用いる計算式

「管理図係数表」には、多数の値が記載されているが、大きく分類すると、2つの群に分けられる。1つ目の群の係数は、以下の計算式でガンマ関数から求められる。なお、これらの計算式は、日本規格協会(2010)[1]に記載されている。

$$c_4 = \sqrt{\frac{2}{n-1} \Gamma\left(\frac{n}{2}\right) / \Gamma\left(\frac{n-1}{2}\right)} \quad (1)$$

$$c_5 = \sqrt{1 - c_4^2} \quad (2)$$

$$A_3 = \frac{3}{\sqrt{nc_4}} \quad (3)$$

$$B_6, B_5 = c_4 \pm 3c_5 \quad (4)$$

$$B_4, B_3 = 1 \pm 3 \frac{c_5}{c_4} \quad (5)$$

これらの値は、 $\Gamma(1/2) = \sqrt{\pi}$ と階乗の(多倍長を用いた)正確な計算によって確認できる。確認の結果、すべての数値に問題がなかった。

残りのもう1つの群は、標準正規分布 $N(0,1)$ からの無作為標本 x_1, x_2, \dots, x_n に対する、範囲 R (最大値から最小値を引いた値)の期待値 d_2 、標準偏差 d_3 と、これらに関連する数値で、次の式で求められる。

* 神戸大学大学院人間発達環境学研究所准教授

** 放送大学

(2012年10月1日 受付)
 (2012年12月15日 受理)

$$A_2 = \frac{3}{\sqrt{nd_2}} \quad (6)$$

$$D_2, D_1 = d_2 \pm 3d_3 \quad (7)$$

$$D_4, D_3 = 1 \pm 3 \frac{d_3}{d_2} \quad (8)$$

また、山内(1972)[3]によると、これらの式に用いられている d_2 と d_3 の計算方法のために必要な範囲 R の密度関数、その k 次モーメントが、以下の式で与えられている。

$$f(R, n) = n(n-1) \int_{-\infty}^{\infty} \left[\int_x^{x+R} \phi(t) dt \right]^{n-2} \phi(x) \phi(x+R) dx$$

$$E(R^k) = \int_0^{\infty} R^k f(R, n) dR \quad (9)$$

これらの式を用いて、

$d_2 = E(R)$, $d_3^2 = Var(R) = E(R^2) - \{E(R)\}^2$ (10) のように求めることができる。なお、山内(1972)[3]には、これらの小数点以下7桁までの値も記載されている。3章では、この値を用いて管理図係数表の値を確認する。

ただし、日本規格協会(2010)[1]の数値表における d_2 には問題がなかったが、 $1/d_2$ の数値に問題があった。どのような問題かという、 d_2 の表示桁での値を真値として、その逆数が求められ、 $1/d_2$ として表示されている。例えば、 $n=2$ においては、 $d_2 = 1.128$ より $1/d_2 = 1/1.128 = 0.8865$ のように表示されている。我々は、数値表での表示には、真値を表示桁で四捨五入して表示すべきであると考えている。その根拠は、計算途中で四捨五入などを許すと、結果が安定しないからである。この意味では、 $1/d_2 = 1/1.1283792 = 0.8862$ が正しい値であり、 $1/d_2 = 0.8865$ は広い意味での誤植であると解釈できる。 $1/d_2$ と $1/c_4$ における誤植とは、すべてこの意味での誤植にすぎない。

なお、山内(1972)[3]に掲載されている d_2 , d_3^2 の値をフリーソフト Maxima で確認した所、これらの数値は、表示されている最終桁まで、すべて正確であることが確認された。この確認作業に用いたプログラムの精度評価は、その概要を4章で述べる。

3. 管理図係数表の真値

日本規格協会(2010)[1]の「管理図係数表」に記載されている値のうち、 $1/d_2$ と $1/c_4$ と $D_1 \sim D_4$ には食い違いがあった。そこでこれらの値についての正しい値(真値)を巻末の表1に記す。また参考のため、山内(1972)[3]に記載されている d_2 , d_3^2 の値も表

1に含めることにする。なお、食い違っている数値は $1/d_2$ と $1/c_4$ における食い違いを除いて、すべて末尾が1違っているだけである。参考のため、食い違っている数値について太字で表示する。

>

表1によると、日本工業規格(JIS)のシューハート管理図では、

$1/c_4$ で7カ所、 D_1 で8カ所、 D_2 で6カ所、 D_3 で3カ所、 D_4

で4カ所、 $1/d_2$ で2カ所の、計30カ所の修正が必要である。こ

のうち、 $1/c_4$ と $1/d_2$ の数値に関しては、真値の逆数を表示しているため必ずしも誤植であるとは言い切れないが、本来数値表とは、その真値を表示桁まで四捨五入して表示するべきだと考えて、今回の報告に含めた。

一方、従来の日科技連数値表(2012年第2版第3刷)には、表1によれば、 D_1 で3カ所、 D_2 で2カ所、 D_4 で1カ所、計6カ所の修正が必要であったが、最新の改訂作業の結果、2012年第2版第4刷以降、すなわち、日科技連数値表(2012)[2]には誤植がない。

4. 誤差評価の概要

この節では、正規分布に従う標本からの範囲のモーメントである d_2 , d_3^2 の値を、 R での数値積分によって求める方法と、その精度評価の概要を示す。なお計算には、Maxima というフリーソフトを利用した。このソフトは、計算の有効桁を自動的に十分取ってくれる性質があり、精度評価が容易である。更に、正規分布の分布関数やシンプソンの公式も組み込まれている。今回の計算には、 R と x での2重積分が必要となるので、1回目の x での積分は組み込み関数を用い、2回目の R での積分に関してはプログラムを作成した。そのプログラム例を付録に掲載している。

このプログラムを用いて、すべての $n=2 \sim 25$ で、 d_2 , d_3^2 の表示桁での値に関して、 x での積分における積分区間を $[0,10]$ と $[0,20]$ のいずれにしても、変化がないことを確かめた。また、積分区間 $[0,10]$ において、区間の分割数として $M=500$ と $M=1000$ で値に変化がないことも確かめた。更に、2回目の R での積分における、積分区間は $[0,20]$ と $[0,40]$ で変化がないこと、 $[0,20]$ における分割数 $L=1000$ と $L=2000$ で値に変化がないことも確かめた。

一方、シンプソンの公式には、被積分関数 $f(x)$ を積分区間 $[a,b]$ で積分する場合、関数 $f(x)$ の4回微分の最大値 M_4 を用いて、誤差

の最大値は理論的に $(b-a)^5 / (2880M_4)$ 以下と分かっている。

今回の数値積分計算では、 x での積分の精度はソフトが保証してくれているので、 R での積分において $n=2 \sim 25$ に対して4回微分を評価した。その結果、今回用いたシンプソンの公式の精度は、概ね十分であると解釈できる。結論として、3章で求めた数値表は信用できる。

5. まとめ

この論文では、山内(1972)[3]に記載された値からの計算結果と、Maxima を用いた R での数値積分による結果を比較検討した。その結果、田中(2010)[5]による x での数値積分による結果と併せると、3つの結果がすべて一致している。これらのことから、表1に

示した値が正しいと信ずるに十分な根拠がある。

今後の課題としては、現在一般的に使われている、日本規格協会(2010)[1] に対して速やかな修正が求められると共に、森口(2012)[2] の迅速な対応に謝意を表したい。

なお、末尾に付録としてプログラム例を添付する。このプログラム例は、正規分布に従う標本からの範囲のモーメントである d_2 ,

d_3^2 の値を、範囲 R での数値積分によって求めるものである。プログラム言語としては、誰でも手軽に用いることのできるフリーソフト Maxima を用い、利用できる組み込み関数は利用する方針とした。

6. 参考文献

- [1] 日本規格協会(2010) “JIS ハンドブック, 日本工業規格(JIS Z 9021:1998) 「シューハート管理図」 ”
- [2] 森口繁一(2012) ”日科技連数値表第 2 版第 4 刷”
- [3] 山内二郎(1972) 統計数値表, 日本規格協会.
- [4] ASTM(2010) “American Social Testing and Materials”
- [5] 田中逸郎(2010), $\bar{X} - R$ 管理図係数表における高精度計算法の研究, 放送大学 2010 年度卒業論文.

表1 管理関係数表 (正しい数値)

n	D_1	D_2	D_3	D_4	$1/c_4$	$1/d_2$	d_2	d_3^2
2	0.000	3.686	0.000	3.267	1.2533	0.8862	1.1283792	0.7267605
3	0.000	4.358	0.000	2.575	1.1284	0.5908	1.6925688	0.7891977
4	0.000	4.698	0.000	2.282	1.0854	0.4857	2.0587507	0.7740625
5	0.000	4.918	0.000	2.114	1.0638	0.4299	2.3259289	0.7466376
6	0.000	5.079	0.000	2.004	1.0509	0.3946	2.5344127	0.7191713
7	0.205	5.204	0.076	1.924	1.0424	0.3698	2.7043568	0.6942311
8	0.388	5.307	0.136	1.864	1.0362	0.3512	2.8472006	0.6721237
9	0.547	5.394	0.184	1.816	1.0317	0.3367	2.9700263	0.6525962
10	0.686	5.469	0.223	1.777	1.0281	0.3249	3.0775055	0.6352898
11	0.811	5.535	0.256	1.744	1.0253	0.3152	3.1728727	0.6198643
12	0.923	5.594	0.283	1.717	1.0230	0.3069	3.2584553	0.6060285
13	1.025	5.647	0.307	1.693	1.0210	0.2998	3.3359804	0.5935411
14	1.118	5.696	0.328	1.672	1.0194	0.2935	3.4067631	0.5822042
15	1.203	5.740	0.347	1.653	1.0180	0.2880	3.4718269	0.5718557
16	1.282	5.782	0.363	1.637	1.0168	0.2831	3.5319828	0.5623621
17	1.356	5.820	0.378	1.622	1.0157	0.2787	3.5878840	0.5536131
18	1.424	5.856	0.391	1.609	1.0148	0.2747	3.6400638	0.5455164
19	1.489	5.889	0.404	1.596	1.0140	0.2711	3.6889630	0.5379951
20	1.549	5.921	0.415	1.585	1.0132	0.2677	3.7349501	0.5309838
21	1.606	5.951	0.425	1.575	1.0126	0.2647	3.7783358	0.5244270
22	1.660	5.979	0.435	1.565	1.0120	0.2618	3.8193846	0.5182773
23	1.711	6.006	0.443	1.557	1.0114	0.2592	3.8583234	0.5124938
24	1.759	6.032	0.452	1.548	1.0109	0.2567	3.8953481	0.5070411
25	1.805	6.056	0.459	1.541	1.0105	0.2544	3.9306292	0.5018883

注1 : 太字は、日本規格協会(2010) [1]との不整合のある数値である。

注2 : d_2 , d_3^2 の値は、山内(1972) [3]に掲載されている表示値である。

付録 (プログラム例)

***** (Maxima によるプログラム例 : ここから) *****

```

load(simpson);
load(distrib);
array(a_kotae,2001); array(b_kotae,2001); array(c_kotae,2001);

N:3; M:500; L:1000;
IntR:0; Intx:20; Intx:0; Intx:10;
print("n=",N); print(" ,M(x_Kirikazu)=" ,M); print(" ,L(R_Kirikazu)=" ,L);
print(" ,IntR=" ,IntR); print(" ,Intx=" ,Intx);

for k:0 thru M do(
stn_DXT1(x):=((cdf_normal(x+(Intx*(2*k/M-1)),0,1)
-cdf_normal((Intx*(2*k/M-1)),0,1))^(N-2))
*pdf_normal((Intx*(2*k/M-1)),0,1)*pdf_normal(x+(Intx*(2*k/M-1)),0,1),
a_kotae[k]:float(simpson(stn_DXT1,0,IntR,L)),
stn_DXT2(x):=x*stn_DXT1(x), b_kotae[k]:float(simpson(stn_DXT2,0,IntR,L)),
stn_DXT3(x):=x^2*stn_DXT1(x), c_kotae[k]:float(simpson(stn_DXT3,0,IntR,L)));

suma0:a_kotae[0]+a_kotae[M];
suma1:0; for i:0 thru M do(suma1:suma1+a_kotae[i]);
suma2:0; for i:0 thru M/2 do(suma2:suma2+a_kotae[2*i]);
da:4*Intx/M*((suma0+2*(suma2-suma0)+4*(suma1-suma2))/6);

sumb0:b_kotae[0]+b_kotae[M];
sumb1:0; for i:0 thru M do(sumb1:sumb1+b_kotae[i]);
sumb2:0; for i:0 thru M/2 do(sumb2:sumb2+b_kotae[2*i]);
db:4*Intx/M*((sumb0+2*(sumb2-sumb0)+4*(sumb1-sumb2))/6);

sumc0:c_kotae[0]+c_kotae[M];
sumc1:0; for i:0 thru M do(sumc1:sumc1+c_kotae[i]);
sumc2:0; for i:0 thru M/2 do(sumc2:sumc2+c_kotae[2*i]);
dc:4*Intx/M*((sumc0+2*(sumc2-sumc0)+4*(sumc1-sumc2))/6);

d1:da*N*(N-1); print("d1=",d1);
d2:db*N*(N-1); print(" ,d2=" ,d2);
d2inv:1/d2; print(" ,d2inv=" ,d2inv);
d3:sqrt(float(dc*N*(N-1)-d2^2)); print(" ,d3=" ,d3);

dd1:d2-3*d3; print("D1=" ,dd1);
dd2:d2+3*d3; print(" ,D2=" ,dd2);
dd3:1-3*d3/d2; print(" ,D3=" ,dd3);
dd4:1+3*d3/d2; print(" ,D4=" ,dd4);
***** (Maxima によるプログラム例 ; ここまで) *****

```