PERGAMON

Microelectronics Reliability 43 (2003) 495–502

MICROELECTRONICS RELIABILITY

www.elsevier.com/locate/microrel

An algorithm for calculating the lower confidence bounds of C_{PU} and C_{PL} with application to low-drop-out linear regulators

W.L. Pearn [a,*], Ming-Hung Shu [b]

[a] *Department of Industrial Engineering & Management, National Chiao Tung University, 1001 Ta Hsueh Road, Hsin Chu 30050 Taiwan, ROC*

[b] *Department of Management Science, Chinese Military Academy, Taiwan, ROC*

Received 3 July 2002; received in revised form 13 August 2002

Abstract

In assessing the performance of normal stable manufacturing processes with one-sided specification limits, process capability indices C_{PU} and C_{PL} have been widely used to measure the process capability. The purpose of this paper is to develop an algorithm to compute the lower confidence bounds on C_{PU} and C_{PL} using the UMVUEs of C_{PU} and C_{PL}. The lower confidence bound presents a measure on the minimum capability of the process based on the sample data. We also provide tables for the engineers/practitioners to use in measuring their processes. A real-world example taken from a microelectronics device manufacturing process is investigated to illustrate the applicability of the algorithm. Implementation of the existing statistical theory for capability assessment fills the gap between the theoretical development and the in-plant applications.

1. Introduction

Process capability indices are used to measure the capability of a process to reproduce items within the specified tolerance preset by the product designers or customers. Several capability indices, C_p, C_{PU}, C_{PL}, C_{pk}, and C_{pm}, have been developed for the manufacturing industry (Kane [1], Chan et al. [2], Pearn et al. [3]). Those indices essentially compare the specification tolerance range with the actual production tolerance range, which have been defined as

$$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma},$$

$$C_{PU} = \frac{\text{USL} - \mu}{3\sigma}, \quad C_{PL} = \frac{\mu - \text{LSL}}{3\sigma},$$

$$C_{pk} = \min\left\{\frac{\text{USL} - \mu}{3\sigma}, \frac{\mu - \text{LSL}}{3\sigma}\right\},$$

$$C_{pm} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\sigma^2 + (\mu - T)^2}},$$

where USL is the upper specification limit, LSL is the lower specification limit, μ is the process mean, σ is the process standard deviation (overall process variation), and T is the preset target value.

The indices C_p, C_{pk}, and C_{pm} are appropriate for processes with two-sided specification limits. Some procedures have been developed based on these indices (Cheng [4], Pearn and Chen [5], Pearn and Chen [6]) for the practitioners to use in making decision on whether their processes meet the preset capability requirement. On the other hand, the indices C_{PU} and C_{PL} are designed specifically for processes with one-sided specification limit. Pearn and Chen [7] considered the indices C_{PU} and C_{PL}, and obtained their uniformly minimum-variance

* Corresponding author. Tel.: +886-35-714261; fax: +886-35-722392.

E-mail address: roller@cc.nctu.edu.tw (W.L. Pearn).

0026-2714/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0026-2714(02)00264-0

unbiased estimators (UMVUEs). Lin and Pearn [8] developed some efficient SAS/Maple computer programs for calculating the critical values and the p-values using those UMVUEs in capability testing, particularly for normal processes.

Critical values are used for making decisions in capability testing with designated type-I error α, the risk of misjudging an incapable process ($H_0: C_{PU} \leqslant C$) as a capable one ($H_1: C_{PU} > C$). The p-values are used for making decisions in capability testing, which presents the actual risk of misjudging an incapable process ($H_0: C_{PU} \leqslant C$) as a capable one ($H_1: C_{PU} > C$). Thus, if $p < \alpha$ then we reject the null hypothesis, and conclude that the process is capable with actual type-I error p (rather than α). Both approaches, the critical values and the p-values, do not convey any information regarding the minimal value (lower confidence bound) of the actual process capability. The development of the lower confidence bound on the actual process capability is essential. The lower confidence bound not only gives us a clue on the minimal level of the actual performance of the process which is tightly related to the nonconforming units (product units fallout the specification limit USL or LSL), but is also useful in making decisions for capability testing. Table 1 displays some capability values of C_{PU} (or C_{PL}) and the corresponding nonconforming unit (in ppm).

Montgomery [9] recommended some minimum quality requirements on C_{PU} and C_{PL}, shown in Table 2, for specific process types which must run under some designated capability conditions. Therefore, it would be desirable to determine a bound which practitioners would be expected to find the true value of the process capability no less than the bound value with certain level of confidence.

Table 1
Various C_{PU} values and the corresponding nonconformities

C_{PU}	ppm
0.5	66 807
0.7	17 864
0.9	3467
1.1	484
1.3	48.10
1.5	3.40
1.7	0.1698
1.9	0.0060
2.1	0.0001488
2.3	0.0000026

Table 2
Some minimum capability requirements on C_{PU} and C_{PL}

C_{PU} (or C_{PL})	Process types
1.25	Existing processes
1.45	New processes, or existing processes on safety, strength, or critical parameters
1.6	New processes on safety, strength, or critical parameters

2. Estimations of C_{PU} and C_{PL}

A random sample is taken from a stable process. The conventional estimates of μ and σ are:

$$\overline{X} = \sum_{i=1}^{n} X_i/n,$$

$$S = \left[(n-1)^{-1}\sum_{i=1}^{n}(X_i - \overline{X})^2\right]^{1/2}.$$

Thus, we may consider the natural estimators to estimate the indices C_{PU} and C_{PL} which can be defined as the following:

$$\widehat{C}_{PU} = \frac{\mathrm{USL} - \overline{X}}{3S}$$

$$\widehat{C}_{PL} = \frac{\overline{X} - \mathrm{LSL}}{3S}.$$

Chou and Owen [10] showed that under normality assumption the estimator $\widehat{C}_{PU}$ and $\widehat{C}_{PL}$ are distributed as $(3\sqrt{n})^{-1}T_{n-1}(\delta)$, where $T_{n-1}(\delta)$ is distributed as the noncentral t distribution with $n-1$ degrees of freedom and noncentrality parameter $\delta = 3\sqrt{n}C_{PU}$ and $\delta = 3\sqrt{n}C_{PL}$, respectively. Both estimators are biased (Pearn and Chen [7]). But, Pearn and Chen [7] showed that by adding the well-known correction factor

$$b_{n-1} = \left(\frac{2}{n-1}\right)^{1/2}\Gamma\left(\frac{n-1}{2}\right)\Gamma\left(\frac{n-2}{2}\right)^{-1}$$

to $\widehat{C}_{PU}$ and $\widehat{C}_{PL}$, the unbiased estimators $b_{n-1}\widehat{C}_{PU}$ and $b_{n-1}\widehat{C}_{PL}$ can be obtained which have been denoted as $\widetilde{C}_{PU}$ and $\widetilde{C}_{PL}$. Thus, we have $E(\widetilde{C}_{PU}) = C_{PU}$ and $E(\widetilde{C}_{PL}) = C_{PL}$. Since $b_{n-1} < 1$ for $n > 2$, then $\mathrm{Var}(\widetilde{C}_{PU}) < \mathrm{Var}(\widehat{C}_{PU})$ and $\mathrm{Var}(\widetilde{C}_{PL}) < \mathrm{Var}(\widehat{C}_{PL})$. Pearn and Chen [7] further showed that $\widetilde{C}_{PU}$ and $\widetilde{C}_{PL}$ are the UMVUEs of C_{PU} and C_{PL}, respectively. The probability density function of $\widetilde{C}_{PU}$ and $\widetilde{C}_{PL}$ can be expressed as

$$f(x) = \frac{3\sqrt{n/(n-1)} \times 2^{-n/2}}{b_{n-1}\sqrt{\pi}\Gamma[(n-1)/2]} \times \int_0^{\infty} y^{(n-2)/2} \exp\left\{-\frac{1}{2}\left[y + \left(\frac{3x\sqrt{ny}}{b_{n-1}\sqrt{n-1}} - \delta\right)^2\right]\right\}\mathrm{d}y.$$

3. Lower confidence bound on C_{PU} (or C_{PL})

Chou et al. [11] established the lower confidence bounds on C_{PU} and C_{PL} based on the natural estimator $\widehat{C}_{PU}$ and $\widehat{C}_{PL}$. We note that those two estimators are

Table 3
Lower confidence bounds C_{U} of C_{PU} for $\widetilde{C}_{\mathrm{PU}} = 0.7(0.1)1.8$, $n = 5(5)200$, $\gamma = 0.95$ (Panel A) and $\widetilde{C}_{\mathrm{PU}} = 1.9(0.1)3.0$, $n = 5(5)200$, $\gamma = 0.95$ (Panel B)

n	$\widetilde{C}_{\mathrm{PU}}$											
	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
Panel A												
5	0.304	0.364	0.423	0.481	0.538	0.594	0.650	0.705	0.760	0.815	0.870	0.924
10	0.414	0.486	0.557	0.627	0.697	0.766	0.835	0.943	0.972	1.040	1.108	1.175
15	0.465	0.542	0.618	0.693	0.768	0.843	0.917	0.991	1.065	1.139	1.213	1.286
20	0.495	0.575	0.654	0.733	0.811	0.889	0.966	1.044	1.121	1.198	1.275	1.352
25	0.516	0.598	0.679	0.760	0.840	0.920	1.000	1.080	1.159	1.239	1.318	1.397
30	0.531	0.614	0.697	0.780	0.862	0.944	1.025	1.107	1.188	1.269	1.350	1.431
35	0.543	0.628	0.712	0.795	0.879	0.962	1.045	1.127	1.210	1.293	1.375	1.457
40	0.553	0.638	0.723	0.808	0.892	0.977	1.061	1.144	1.228	1.312	1.395	1.478
45	0.561	0.647	0.733	0.819	0.904	0.989	1.074	1.158	1.243	1.327	1.412	1.496
50	0.568	0.655	0.741	0.828	0.914	0.999	1.085	1.170	1.256	1.341	1.426	1.511
55	0.574	0.661	0.748	0.835	0.922	1.008	1.094	1.181	1.267	1.352	1.438	1.524
60	0.579	0.667	0.755	0.842	0.929	1.016	1.103	1.190	1.276	1.363	1.449	1.535
65	0.583	0.672	0.760	0.848	0.936	1.023	1.110	1.197	1.285	1.371	1.458	1.545
70	0.588	0.677	0.765	0.853	0.941	1.029	1.117	1.205	1.292	1.379	1.467	1.554
75	0.591	0.681	0.770	0.858	0.947	1.035	1.123	1.211	1.299	1.387	1.474	1.562
80	0.595	0.684	0.774	0.862	0.951	1.040	1.128	1.217	1.305	1.393	1.481	1.569
85	0.598	0.688	0.777	0.866	0.956	1.044	1.133	1.223	1.311	1.399	1.488	1.577
90	0.600	0.691	0.781	0.870	0.959	1.049	1.138	1.228	1.316	1.405	1.494	1.583
95	0.603	0.693	0.784	0.873	0.963	1.053	1.142	1.233	1.321	1.411	1.500	1.589
100	0.605	0.696	0.786	0.877	0.966	1.056	1.146	1.238	1.326	1.416	1.505	1.595
105	0.608	0.699	0.789	0.879	0.970	1.060	1.150	1.242	1.330	1.420	1.510	1.601
110	0.610	0.701	0.792	0.882	0.972	1.063	1.153	1.246	1.334	1.424	1.515	1.606
115	0.612	0.703	0.794	0.855	0.975	1.066	1.156	1.249	1.338	1.428	1.519	1.610
120	0.613	0.705	0.796	0.887	0.978	1.068	1.159	1.253	1.341	1.432	1.523	1.614
125	0.615	0.707	0.798	0.889	0.980	1.071	1.162	1.256	1.344	1.435	1.527	1.618
130	0.617	0.709	0.800	0.891	0.982	1.073	1.164	1.259	1.347	1.438	1.530	1.622
135	0.618	0.710	0.802	0.893	0.984	1.075	1.166	1.261	1.350	1.441	1.533	1.625
140	0.620	0.712	0.804	0.895	0.986	1.077	1.169	1.264	1.352	1.444	1.536	1.628
145	0.621	0.713	0.805	0.897	0.988	1.079	1.171	1.266	1.355	1.446	1.539	1.631
150	0.622	0.715	0.807	0.899	0.990	1.081	1.173	1.268	1.357	1.449	1.541	1.634
155	0.624	0.716	0.808	0.900	0.992	1.083	1.175	1.271	1.359	1.451	1.543	1.636
160	0.625	0.717	0.810	0.902	0.994	1.085	1.177	1.273	1.361	1.453	1.546	1.639
165	0.626	0.719	0.811	0.903	0.995	1.087	1.178	1.275	1.363	1.456	1.548	1.641
170	0.627	0.720	0.812	0.905	0.997	1.088	1.180	1.277	1.365	1.458	1.550	1.643
175	0.628	0.721	0.814	0.906	0.998	1.090	1.182	1.279	1.367	1.460	1.552	1.645
180	0.629	0.722	0.815	0.908	1.000	1.092	1.184	1.280	1.369	1.462	1.554	1.648
185	0.630	0.723	0.816	0.909	1.001	1.093	1.185	1.282	1.371	1.464	1.556	1.650
190	0.631	0.724	0.817	0.910	1.002	1.094	1.187	1.284	1.373	1.465	1.558	1.652
195	0.632	0.725	0.818	0.911	1.004	1.096	1.188	1.286	1.374	1.467	1.560	1.653
200	0.632	0.726	0.819	0.912	1.005	1.097	1.189	1.287	1.376	1.469	1.562	1.655
	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0
Panel B												
5	0.978	1.032	1.086	1.140	1.194	1.247	1.301	1.355	1.408	1.462	1.515	1.568
10	1.243	1.310	1.378	1.445	1.512	1.580	1.647	1.714	1.781	1.848	1.915	1.982
15	1.359	1.433	1.510	1.579	1.652	1.725	1.798	1.871	1.944	2.017	2.090	2.163
20	1.429	1.506	1.582	1.659	1.736	1.812	1.889	1.965	2.041	2.118	2.194	2.271
25	1.477	1.556	1.635	1.714	1.793	1.872	1.950	2.029	2.108	2.187	2.266	2.344
30	1.512	1.593	1.673	1.754	1.835	1.916	1.996	2.077	2.157	2.238	2.318	2.399
35	1.539	1.622	1.704	1.786	1.868	1.950	2.032	2.114	2.196	2.278	2.359	2.441
40	1.562	1.645	1.728	1.811	1.895	1.978	2.061	2.144	2.227	2.310	2.393	2.476
45	1.580	1.664	1.749	1.833	1.917	1.980	2.085	2.169	2.252	2.336	2.420	2.504
50	1.596	1.681	1.766	1.851	1.935	1.999	2.105	2.190	2.274	2.359	2.443	2.528

(*continued on next page*)

Table 3 (*continued*)

n	$\widetilde{C}_{PU}$											
	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0
55	1.610	1.695	1.781	1.866	1.952	2.017	2.123	2.208	2.293	2.379	2.464	2.549
60	1.621	1.707	1.794	1.880	1.966	2.031	2.137	2.223	2.309	2.395	2.481	2.567
65	1.632	1.718	1.805	1.891	1.978	2.044	2.150	2.237	2.323	2.410	2.496	2.582
70	1.641	1.728	1.815	1.902	1.988	2.055	2.162	2.249	2.335	2.422	2.509	2.596
75	1.649	1.737	1.824	1.911	1.998	2.065	2.173	2.260	2.347	2.434	2.521	2.608
80	1.657	1.745	1.833	1.921	2.008	2.075	2.183	2.271	2.358	2.446	2.533	2.621
85	1.665	1.753	1.841	1.929	2.017	2.085	2.193	2.281	2.369	2.457	2.545	2.633
90	1.672	1.761	1.849	1.938	2.026	2.094	2.203	2.291	2.379	2.468	2.556	2.644
95	1.679	1.768	1.857	1.946	2.034	2.103	2.212	2.301	2.389	2.478	2.567	2.656
100	1.685	1.774	1.864	1.953	2.042	2.111	2.221	2.310	2.399	2.488	2.577	2.666
105	1.691	1.780	1.870	1.960	2.050	2.119	2.229	2.319	2.408	2.498	2.587	2.677
110	1.696	1.786	1.877	1.967	2.057	2.127	2.237	2.327	2.417	2.507	2.597	2.687
115	1.701	1.792	1.882	1.973	2.064	2.134	2.245	2.335	2.425	2.516	2.606	2.696
120	1.706	1.797	1.888	1.979	2.070	2.140	2.252	2.343	2.433	2.524	2.615	2.706
125	1.710	1.801	1.893	1.984	2.076	2.147	2.258	2.350	2.441	2.532	2.623	2.714
130	1.714	1.805	1.897	1.989	2.081	2.152	2.265	2.357	2.448	2.540	2.631	2.723
135	1.717	1.809	1.902	1.994	2.086	2.158	2.270	2.363	2.455	2.547	2.639	2.731
140	1.721	1.813	1.906	1.998	2.091	2.163	2.276	2.369	2.461	2.554	2.646	2.738
145	1.724	1.816	1.909	2.002	2.095	2.167	2.281	2.374	2.467	2.560	2.653	2.746
150	1.727	1.819	1.912	2.006	2.099	2.172	2.285	2.379	2.472	2.566	2.659	2.752
155	1.729	1.822	1.916	2.009	2.102	2.175	2.290	2.384	2.477	2.571	2.665	2.758
160	1.732	1.825	1.918	2.012	2.106	2.179	2.294	2.388	2.482	2.576	2.670	2.764
165	1.734	1.827	1.921	2.015	2.109	2.182	2.297	2.392	2.486	2.581	2.675	2.770
170	1.737	1.830	1.924	2.018	2.112	2.185	2.300	2.395	2.490	2.585	2.680	2.775
175	1.739	1.832	1.926	2.020	2.114	2.188	2.304	2.399	2.493	2.589	2.684	2.779
180	1.741	1.834	1.928	2.023	2.117	2.191	2.306	2.402	2.497	2.592	2.688	2.783
185	1.743	1.837	1.931	2.025	2.119	2.194	2.309	2.405	2.500	2.596	2.691	2.787
190	1.745	1.839	1.933	2.027	2.122	2.196	2.312	2.407	2.503	2.599	2.695	2.791
195	1.747	1.841	1.935	2.029	2.124	2.198	2.314	2.410	2.505	2.602	2.698	2.794
200	1.749	1.843	1.937	2.031	2.126	2.200	2.316	2.412	2.508	2.604	2.701	2.797

biased. In the following, we use $\widetilde{C}_{PU}$ and $\widetilde{C}_{PL}$, the UMVUEs of C_{PU} and C_{PL}, to obtain the lower confidence bound on C_{PU} and C_{PL}.

Let $USL = \overline{X} + k_1 S$ and $LSL = \overline{X} - k_2 S$, so $k_1 = 3\widetilde{C}_{PU}/b_{n-1} = 3\widehat{C}_{PU}$ and $k_2 = 3\widetilde{C}_{PL}/b_{n-1} = 3\widehat{C}_{PL}$. A $100\gamma\%$ lower confidence bound C_U for C_{PU} satisfies $Pr(C_{PU} \geqslant C_U) = \gamma$. It can be written as:

$$
\begin{aligned}
Pr\left(\frac{USL-\mu}{3\sigma} \geqslant C_U\right)
&= Pr\left(\frac{\overline{X}+k_1 S-\mu}{3\sigma} \geqslant C_U\right) \\
&= Pr\left(\frac{Z-3\sqrt{n}C_U}{S/\sigma} \geqslant -k_1\sqrt{n}\right) \\
&= Pr\left(\frac{Z-3\sqrt{n}C_U}{S/\sigma} \geqslant -\frac{3\widetilde{C}_{PU}}{b_{n-1}}\sqrt{n}\right) \\
&= Pr(T_{n-1}(\delta_1) \geqslant t_1) = \gamma,
\end{aligned}
$$

and $Pr(T_{n-1}(\delta_1) \leqslant t_1) = 1-\gamma$. Similarly, a $100\gamma\%$ lower confidence bound C_L for C_{PL} satisfies $Pr(C_{PL} \geqslant C_L) = \gamma$. It can be shown as $Pr(T_{n-1}(\delta_2) \leqslant t_2) = \gamma$, where Z is distributed as $N(0,1)$, $T_{n-1}(\delta)$ is the noncentral t distribution with $n-1$ degrees of freedom and noncentrality parameter δ, $t_1 = -k_1\sqrt{n}$, $t_2 = k_2\sqrt{n}$, $\delta_1 = -3\sqrt{n}C_U$, and $\delta_2 = 3\sqrt{n}C_L$. Thus, to obtain the lower confidence bound (LCB), we may proceed as follows:

Procedure of obtaining the LCB

(a) Determine the γ level of confidence (normally set to 0.95) for the lower confidence bound.
(b) Calculate the value of estimator, $\widetilde{C}_{PU}$ (or $\widetilde{C}_{PL}$), from the sample data.
(c) Check the appropriate table listed in Table 3(a) and (b) and find the corresponding C_U (or C_L) based on $\widetilde{C}_{PU}$ (or $\widetilde{C}_{PL}$) and n.
(d) Conclude that the true value of the process capability is no less than the C_U (or C_L) with $100\gamma\%$ level of confidence.

4. An algorithm for calculating the lower confidence bound

Based on the procedure described above, a Matlab algorithm, called LCB, is developed for computing the lower confidence bounds, C_U (or C_L), on C_{PU} (or C_{PL}). Four auxiliary functions for evaluating C_U (or C_L), the normal distribution function, the Gamma function, polynomial roots function, and the noncentral t distribution function, are required. In order to accelerate the computation, Levinson [12] provided a good initial guess of δ (below the real δ) for numerical iteration. This equation is used to solve for δ if t_1 is given.

$$t_1 \approx \frac{\delta R_{n-1} + Z_\gamma \sqrt{R_{n-1}^2 + (1 - R_{n-1}^2)(\delta^2 - Z_\gamma^2)}}{R_{n-1}^2 - Z_\gamma^2 (1 - R_{n-1}^2)},$$

where $R_{n-1} = [2/(n-1)]^{1/2}\Gamma(n/2)/\Gamma[(n-1)/2]$ and $Z_\gamma = 100\gamma\%$ of the standard normal distribution.

Algorithm for the LCB

Step 1. Input the sample data $(X_1, X_2, \ldots, X_n)$, USL (or LSL), and γ.
Step 2. Calculate $\overline{X}$, S, b_{n-1}, and $\widetilde{C}_{PU}$ (or $\widetilde{C}_{PL}$).
Step 3. Compute a good initial guess for δ.
Step 4. Find the lower confidence bound C_U (or C_L) on C_{PU} (or C_{PL}), through numerical iterations.
Step 5. Output the conclusive message, "The true value of the process capability C_{PU} (or C_{PL}) is no less than the C_U (or C_L) with $100\gamma\%$ level of confidence".

We implement the algorithm, and develop a *Matlab* program to compute the lower confidence bounds (see Appendix A). Table 3(a) and (b) tabulate the lower confidence bounds C_U (or C_L) for $\widetilde{C}_{PU}$ (or $\widetilde{C}_{PL}$) = 0.7(0.1)3.0, for $n = 5(5)200$, and $\gamma = 0.95$. Table 4 compares the lower confidence bounds obtained by Chou et al. [11] with one obtained by our approach. It is noted that with the same confidence level α, our approach provides better bound (particularly for small n), which is closer to the real $\widetilde{C}_{PU}$ (or $\widetilde{C}_{PL}$), and therefore should be recommended for real-world applications.

5. An application example of low-drop-out linear regulators

The positive linear series pass voltage regulators are tailored for low-drop-out applications where low quiescent power is considered important. These regulators include the reverse voltage sensing that prevents current in the reverse direction. The regulators are fabricated with the BiCMOS technology, which is ideally suited for the low input-to-output differential applications. These products are specifically designed to provide well-regulated supply for low IC applications such as high-speed bus termination, low current logic supply, and VGA cards.

The products investigated here are low-drop-out 3A linear regulators with a low dropout voltage and short-circuit protection. These regulators are in three-lead packages and five-lead packages, as depicted in Fig. 1. The three-lead packages have preset outputs at 3.3 V or 5.0 V. The output voltage is regulated to 1.5% at room temperature. The five-lead packages regulate the output

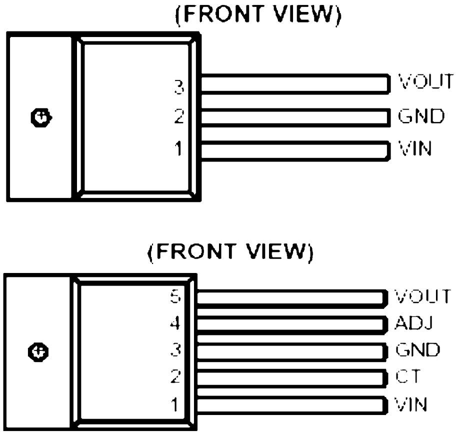

Fig. 1. Low-drop-out 3A linear regulators.

Table 4
Comparisons of the LCB C_U between the Chou's and our approach (New), for $\widetilde{C}_{PU} = 0.7(0.1)1.5$, $n = 10(10)50$, $\gamma = 0.95$

n		$\widetilde{C}_{PU}$								
		0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
10	New	0.41	0.49	0.56	0.63	0.70	0.77	0.84	0.93	0.97
	Chou	0.37	0.44	0.50	0.57	0.63	0.70	0.76	0.82	0.88
20	New	0.50	0.58	0.65	0.73	0.81	0.89	0.97	1.04	1.12
	Chou	0.47	0.55	0.63	0.70	0.78	0.85	0.93	1.00	1.08
30	New	0.53	0.61	0.70	0.78	0.86	0.94	1.03	1.11	1.19
	Chou	0.52	0.60	0.68	0.76	0.84	0.92	1.00	1.08	1.16
40	New	0.55	0.64	0.72	0.81	0.89	0.98	1.06	1.14	1.23
	Chou	0.54	0.63	0.71	0.79	0.87	0.96	1.04	1.12	1.20
50	New	0.57	0.66	0.74	0.83	0.91	1.00	1.09	1.17	1.26
	Chou	0.56	0.64	0.73	0.81	0.90	0.98	1.07	1.15	1.24

Table 5
The 80 sample observations

428.63	408.09	417.62	317.20	519.40	438.75	380.42	457.60	474.87	395.22
344.88	497.17	373.81	337.62	430.78	363.31	475.43	270.65	357.53	454.24
402.19	462.12	403.40	463.31	513.74	433.62	361.49	360.67	400.46	540.87
361.72	303.60	377.56	376.22	379.27	332.91	258.16	357.04	352.24	432.00
432.66	395.20	371.64	315.74	330.47	405.25	324.87	337.89	435.80	399.28
433.73	358.42	422.63	419.25	387.12	277.72	464.39	388.89	384.68	455.70
387.57	337.50	444.43	494.75	472.60	369.92	393.09	492.78	429.29	403.41
274.69	324.70	528.84	443.18	331.91	428.18	423.13	397.66	440.76	332.47

voltage programmed by an external resistor ratio. Short-circuit current is internally limited. The device responds a sustained overcurrent condition by turning off after a t_{ON} time delay. The device then stay off for a period, t_{OFF}, that is 32 times the t_{ON} delay. The device then begins pulsing on and off at the $t_{ON}/(t_{ON}+t_{OFF})$ duty cycle of 3%. This drastically reduces the power dissipation during the short-circuit, which means that the heat sinks need only accommodating the normal operation.

The quiescent current is an essential product characteristic, which has significant impact to product quality. For the quiescent current of a particular model of low-drop-out 3A linear regulators, the upper specification limit, USL, is set to 650 μA. Sample data of 80 observations are collected, which are displayed in Table 5. Fig. 2 displays the histogram of the 80 observations. Fig. 3 displays the normal probability plot of the sample data, and from this figure the sample data appears to be normal. Further, we perform Shapiro–Wilk test to check whether the sample data is normal. The statistic W, is found to be 0.9926, we can conclude that the sample data can be regarded as taken from normal process.

In order to obtain the lower confidence bound on C_{PU}, we execute the Matlab program attached in Appendix A. The program reads the sample data file, and input of $n = 80$, USL $= 650$, and $\gamma = 0.95$, then output

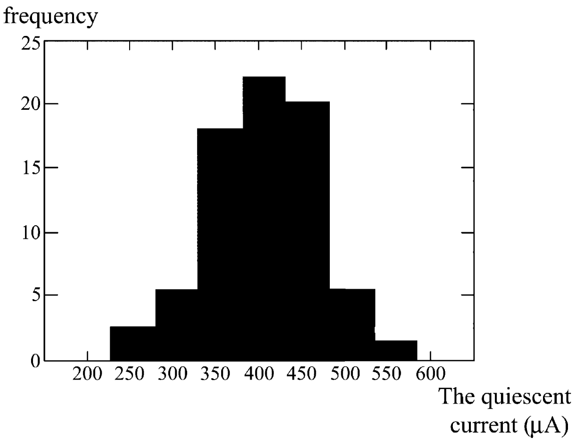

Fig. 2. Histogram of the 80 observations.

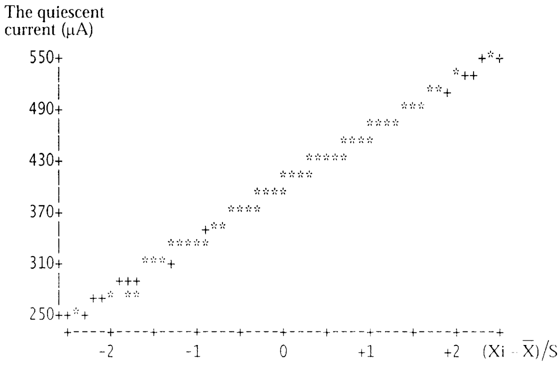

Fig. 3. The normal probability plot.

with the sample mean $\overline{X} = 398.85$, the sample standard deviation $S = 61.65$, the correction factor $b_{n-1} = 0.9905$, the estimator $\widetilde{C}_{PU} = 1.371$, and the lower confidence bound $C_U = 1.191$. The corresponding execution output is included in Appendix B. We therefore conclude that the true value of the process capability C_{PU} is no less than 1.191 with 95% level of confidence.

6. Conclusions

In assessing the performance of normal stable manufacturing processes with one-sided specification limits, process capability indices C_{PU} and C_{PL} have been widely used to measure the process capability. In this paper, we developed an algorithm to compute the lower confidence bounds on C_{PU} and C_{PL} using the UMVUEs of C_{PU} and C_{PL}. The lower confidence bound presents a measure on the minimum capability of the process based on the sample data. We also provided tables for the engineers/practitioners to use in measuring their processes. The lower bounds we provided are closer to the true C_{PU} and C_{PL} index values than the existing ones. A real-world example taken from a microelectronics device manufacturing process is investigated to illustrate the applicability of the algorithm. Implementation of the existing statistical theory for capability assessment fills the gap between the theoretical development and the in-plant applications.

Appendix A

```
%------------------------------------------

% Input the sample data (X1, X2, ..., Xn), USL, and γ
%------------------------------------------

function lcb(n,usl,r)
t = 0;   b = 0;   z = 0;   e = 0;   f = 0;   g = 0;
rt = zeros(2,1);   x = 0;   x1 = 0;   x2 = 0;   y = 0;
y1 = 0; delta = 0; delta1 = 0; x3 = 0;
data = zeros(n,1);
[data(1 : n, 1)] = textread('lingen.dat','%f',n);

%------------------------------------------

% Compute X̄, S, b_{n-1}, and C̃PU.
%------------------------------------------

mdata = mean(data);
stddata = std(data);
b = sqrt(2/(n - 1)) * gamma((n - 1)/2)/
gamma((n - 2)/2));
ecpu = (usl - mdata)/(3 * b * stddata);
fprintf('The sample Mean is %g.\n',mdata)
fprintf('The sample standard Deviation is %g.\n',std-
data)
fprintf('The Correction Factor is %g.\n',b)
fprintf('The UMVUE of Cpu is %g.\n',ecpu)%%

%------------------------------------------

% Compute a good initial value of δ.
%------------------------------------------

t = 3 * (ecoy/b) * sqrt(n);
R = sqrt(2/(n - 1)) * gamma(n/2)/
gamma((n - 1)/2);
z = norminv(r,0,1);
e = R^2 - z^2 * (1 - R^2);
f = -2 * R * (t * R^2 - z^2 * (1 - R^2) * t);
g = (t * R^2 - z^2 * (1 - R^2) * t)^2
- z^2 * R^2 + z^4 * (1 - R^2);
pol = [e f g];
rt = roots(pol);
x = nctinv(r, n - 1, rt(2));
x1 = nctinv(r, n - 1, rt(2) + 0.01);
x2 = (x1 - x)/9.5;
y = abs(x - sqrt(n) * 3 * ecpu/b);
y1 = y/x2;
delta1 = rt(2) + y1 * 0.001;

%------------------------------------------

% Evaluate the lower confidence bound cU (or cL) on
CPU (or CPL), by numerical iterations.
%------------------------------------------
delta = delta1 + 0.001 : 0.001 : delta1 + 0.5;
for i = 1 : 1 : 500
   x3 = nctinv(r, n - 1, delta(i));
   if(abs(x3 - sqrt(n) * 3 * ecpu/b)) <= 0.01
     cu = delta(i)/(3 * sqrt(n));
     fprintf('The Lower Confidence Bound is
     %g.\n',cu)
     break
   end
end
%------------------------------------------

% Output the conclusive message, "The true value of
% the process capability CPU (or CPL) is no less
% than cU (or cL) with 100γ% level of confidence"
%------------------------------------------

fprintf('The true value of the process capability Cpu
is no less than %g', cu)
fprintf('with %g',r)
fprintf('level of confidence')
%------------------------------------------

% The End
%------------------------------------------
```

Appendix B

Input:
>>lcb(80,650,0.95)
Output:
The sample Mean is 398.85.
The Sample Standard Deviation is 61.6503.
The Correction Factor is 0.990471.
The UMVUE of Cpu is 1.37099.
The Lower Confidence Bound is 1.19104.
The true value of the process capability Cpu is no less than 1.19104 with 0.95 level of confidence

References

[1] Kane VE. Process capability indices. J Qual Technol 1986;18(1):41–52.

[2] Chan LK, Cheng SW, Spiring FA. A new measure of process capability: C_{pm}. J Qual Technol 1988;20(3):162–75.

[3] Pearn WL, Kotz S, Johnson NL. Distributional and inferential properties of process capability indices. J Qual Technol 1992;24(4):216–31.

[4] Cheng SW. Practical implementation of the process capability indices. Qual Eng 1994;7(2):239–59.

[5] Pearn WL, Chen KS. A practical implementation of the process capability index C_{PK}. Qual Eng 1997;9(4):721–37.

[6] Pearn WL, Chen KS. Making decisions in assessing process capability index C_{pk}. Qual Reliab Eng Int 1999; 15:321–6.

[7] Pearn WL, Chen KS. One-sided capability indices C_{PU} and C_{PL}: decision making with sample information. Int J Qual Reliab Manag 2002;19(3):221–45.

[8] Lin PC, Pearn WL. Testing process capability for one-sided specification limit with application to the voltage level translator. Microelectron Reliab 2002;42(12):1975–83.

[9] Montgomery DC. Introduction to statistical quality control. 4th ed New York, NY: John Wiley & Sons, Inc; 2001.

[10] Chou YM, Owen DB. On the distributions of the estimated process capability indices. Commun Stat: Theory Meth 1989;18(12):4549–60.

[11] Chou YM, Owen DB, Borrego SA. Lower confidence limits on process capability indices. J Qual Technol 1990;22(3):223–9.

[12] Levinson WA. Exact confidence limits for process capabilities. Qual Eng 1997;9(3):521–8.