ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850

APPLICATION NOTE NO. 415

Nov 1, 2014

DO RANDOM SEQUENCES NEED "IMPROVING"?

Let's suppose you have become disillusioned with methods of generating a random sequence. You may have decided that there are artifacts associated with PRBS (Pseudo-Random Binary Sequences) [1] and/or are wary of software noise generators [2]. Too many unanswered questions. Perhaps you think it is time to look to something like random thermal noise or radioactive decay. Or perhaps just coin flips. It is well to keep in mind that not all applications of random noise generators require the same rigor. If you are using a "white noise" generator to produce snare drum sounds for music synthesis, you can get away with something crude like a reversed-bias transistor junction or indeed, a PRBS. If however you are testing a theory with a Monte Carlo approach, on the other hand, you may well need to look at subtle irregularities and effects.

The output of a PRBS generator is often described as a coin toss sequence. It is not – at least because the PRBS is deterministic, and as discussed [1,3,4] it has structure (and can even be audible). One can get into involved philosophical arguments as to whether a physical coin toss is deterministic. Of course it is – if you have a detailed enough physical description. "John tossed a spinning coin high in the air and it fell to the sidewalk" is not enough. While we generally know exactly what someone means by "coin toss" (implications of a fair coin, a vigorous toss, 50%:50% probability etc.) we may have in the back of our minds the notion that in some cases, the outcome is possibly subject to deliberate influence.

So let's say you do decide that you really want a million coin flips. You start out with great enthusiasm; toss a coin high in the air with lots of rotations. One result – 999,999 to go. After perhaps 20 tosses, you decide to calculate how long this is going to take. Something like a year of 8-hour days! This won't do. If you are lucky at this point it may be the case that you have at your disposal a captive workforce of 300 students in a class and can get each of them to flip coins as a week's homework assignment. The only question would be if they would do it right - or take shortcuts.

What could go wrong? Everything. Different students will throw coins in different ways with different reliability, and possible injection of non-random patterns. Some may suppose they can "make up" random data – a known failing of human beings. Sometimes, students copy each other's homework assignments. Many will write a simulation program instead of toss a real coin. Since you plan to concatenate all the homework-generated sequences into one, any "bad apples" might well poison the result. Is it possible to examine any one student's contribution and vet it as a legitimate tossing effort? Suppose as an initial test you ask each student to submit the results of a length-10 trial. Suppose you see such things as:

(1) H T H T T H H T H T

(2) H H H H H H H H H H

(3a) H H H H H T H H H T where the H is an apparent erasure replaced by:

(3b) H H T H H T H T H T

Of course there is nothing "wrong" with any of the four length-10 sequences.

But, (1) has no runs of three (or more) H's or three (or more) T's. For length 10 we should expect a run of three in about 83% of cases (program on page 15 bottom). So (1) is unlikely, but very far from actually suspect. If we had a large number of students reporting similar to (1) we would be justifiably suspicious. Or if we had a longer length (say 100) of no triples, we would expect someone was just writing things down, and that person erroneously suspects that runs of three should be rare (Figs. 5a and 5b below).

The run of 10 H's in (2) is perfectly okay. We expect that in one of 2^{10}=1024 cases. A run of 10 H's or T's is expected in 512 trials. Here we speculate that we had submissions from 300 students, so this is quite likely. Had we had only 10 submissions (not 300), we might suspect a prank of some sort.

In (3a) and (3b) we have postulated some evidence of erasure, which might indicate a recording error. Or you might suppose the student got HHHHHTHHHT and become alarmed at having 8 of 10 as H, and the long runs (5 and 3) of H's. Fearing a poor grade, the student decided to amend the result for what seemed more random. Both (1) and (3b) are possibly indications of a poor idea many people have of what a random sequence should look like.

How hard is it to do an honest series of flips? Not hard at all. The difficult is in reporting your results. This is the situation where one might be tempted to change a H←→T for a better balance and/or to break up a long run. Or at least, do a new run! Nope – not allowed, although we have sympathy for an author who wants neither to give an untypical impression, nor a long-winded apology for an untypical result. Perhaps several examples can be given at times – perhaps expressly to show the variability.

However, bravely I am going to throw a coin 10 times and report exactly what I get (Pause):

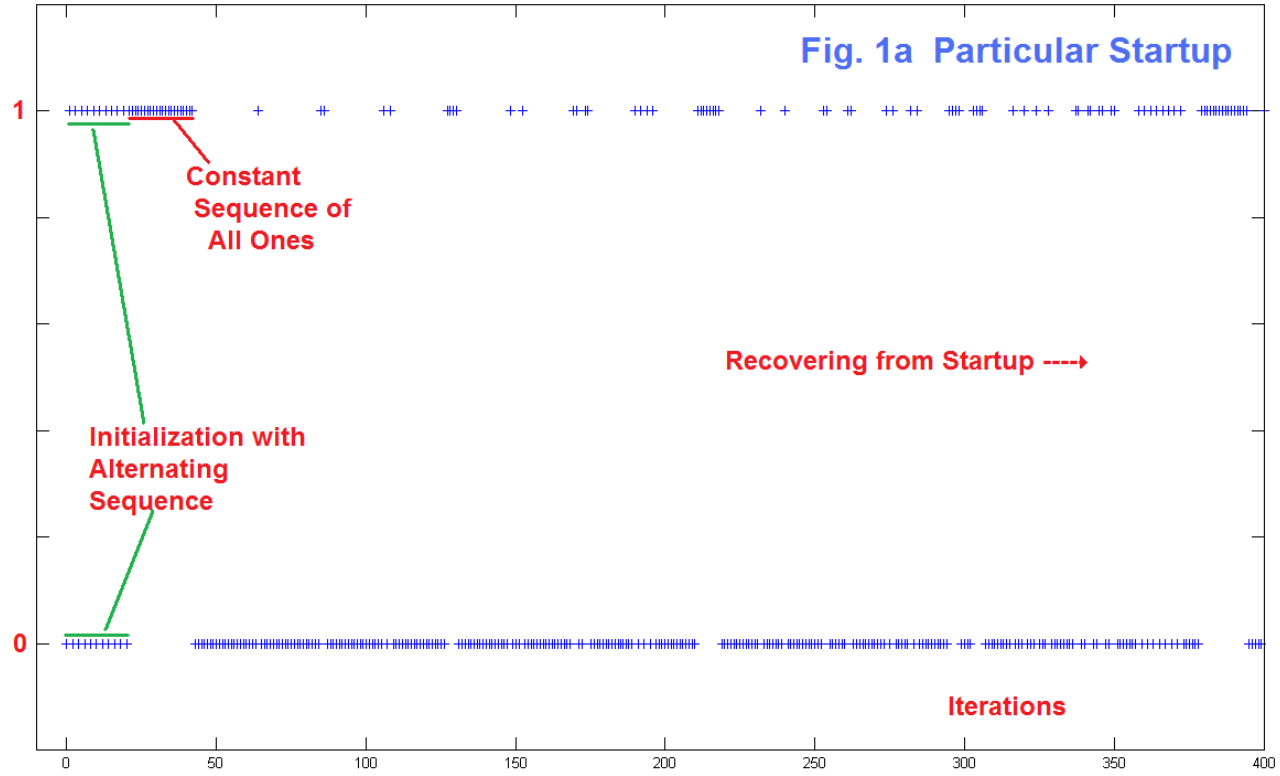
(4) T T H H H H H T H T

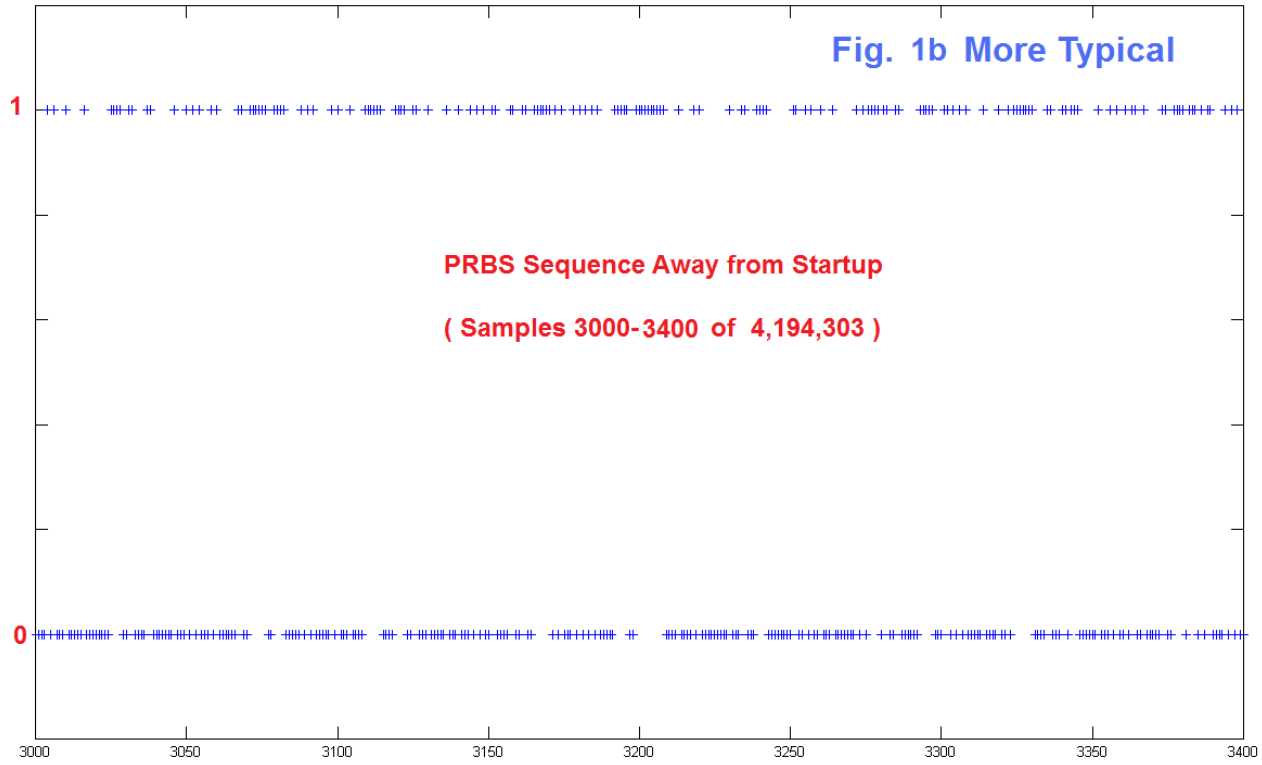
Not bad, although I was a bit worried after that 5th H in a row.

LOOKING AT THE DETAILS – OF THE WHOLE THING?

Just above we are looking at some very short sub-sequences of what we presume must be a much much longer Pseudo-Random Generator (PRG) or perhaps a true random generator based on same physical process (perhaps cosmic rays!). In postulating the involvement of numerous student contributors, the viewpoint was pretty much that of

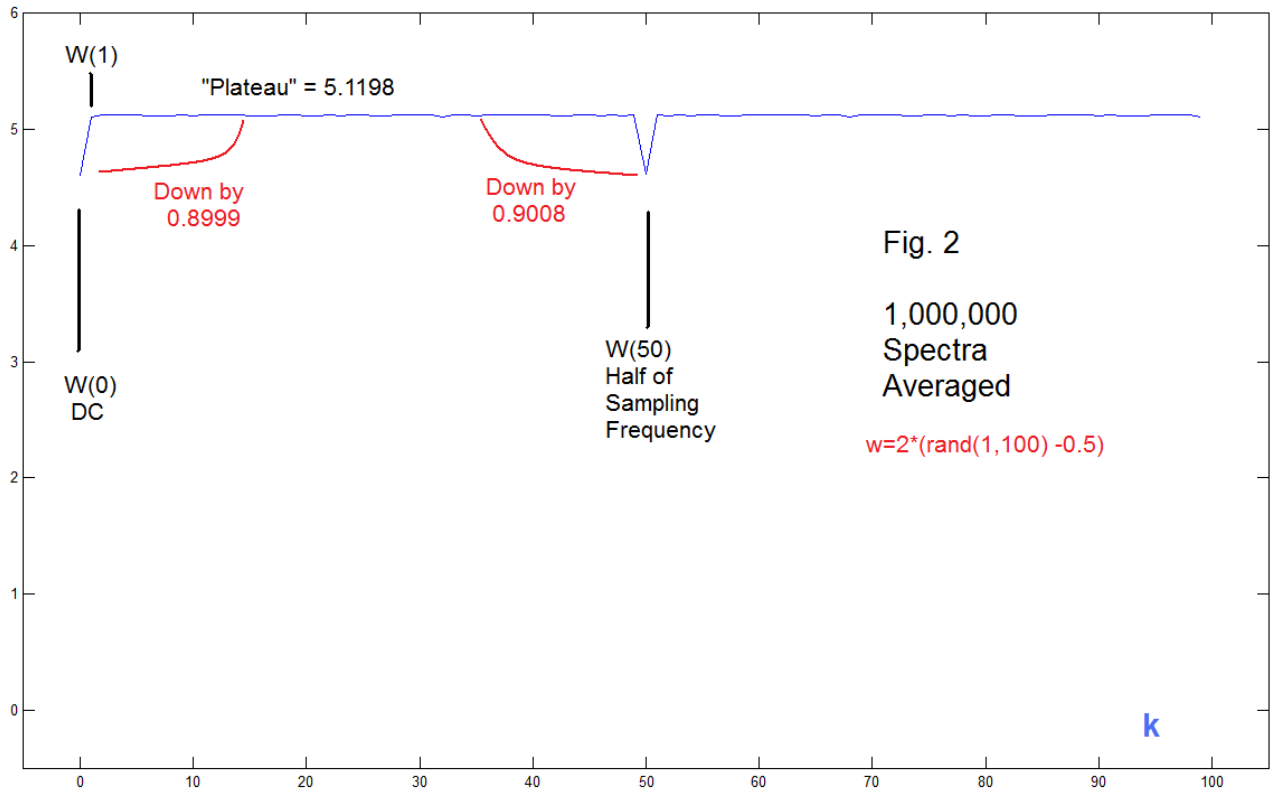
"vetting" the contributions of the individual by looking at their small subsequence contributions. Another view might be to look at sub-sequences of the PRG device to see if they themselves look random, or are consistent with a random origin. PRG's are often tested mainly for desired amplitude distribution [5]. This is straightforward for the most part. Less easy to access are matters of correlation and spectral aspects and artifacts (like [1], Fig. 1a and Fig. 1b; and [2], Fig. 2).





In Fig. 1b (from [1]), we show a rather "typical" portion of a PRBS generator output. This is from a length 22 shift register version. More fundamental to understanding the PRBS generator is Fig. 1a which shows an atypical and potentially disastrous portion of the guaranteed output sequence. While this is supposed to be a coin-flip (say H=1, T=0) it is clear that should you be in the vicinity just beyond about sample 50, we hope you bet on tails! Less you suppose that this is unlikely poor timing, note that it follows rather directly what would be a standard initialization to all 1's. Further, while extreme, the example of Fig. 1a has similar additional (but shorter) artifacts as discussed in [1]. If you must use a PRBS approach, it is not enough the simply run off a lot of samples after initialization. You should probably "chop up" sequences by multiplying (Exclusive ORing) sequences from several (perhaps 3 or more) generators of different lengths.

We also know full well that we can intentionally shape the spectrum of the random noise with a filter, for example as red noise [6]. We want to believe that the spectrum should start out white (uncorrelated) but we found evidence in [2] that when we tested the spectrum with an FFT, there was a dip to about 90% ($2^{3/2}/\pi$) below white at DC, and in the case of an even length FFT, at half the sampling rate, and it is argued in the reference that this is a fundamental finding (Fig. 2). Perhaps it is the case that the FFT should be seen as only an ESTIMATE of the spectrum – not THE spectrum.



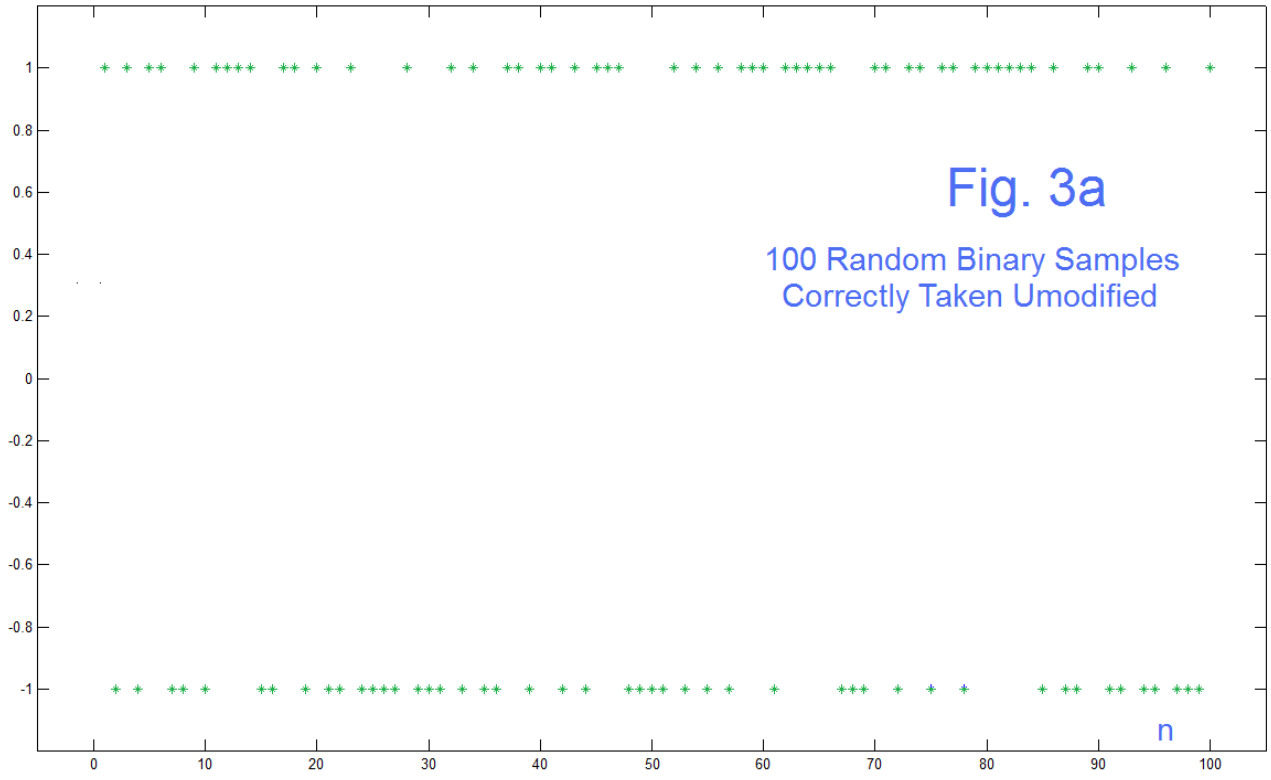
What we have looked at, in these examples of previous findings, is basically analysis of existing noise sequences – finding unanticipated results. Here we will next be supposing that there might well be consequences to any attempts to "improve or correct" individual contributions to random procedures. We will be thinking in terms of smaller pieces constituting a full sequence. This might be in the context of assembling a full sequence from vetted pieces, or it may be the extraction of a sub-sequence for actual use (as would often be the case when we need random numbers).

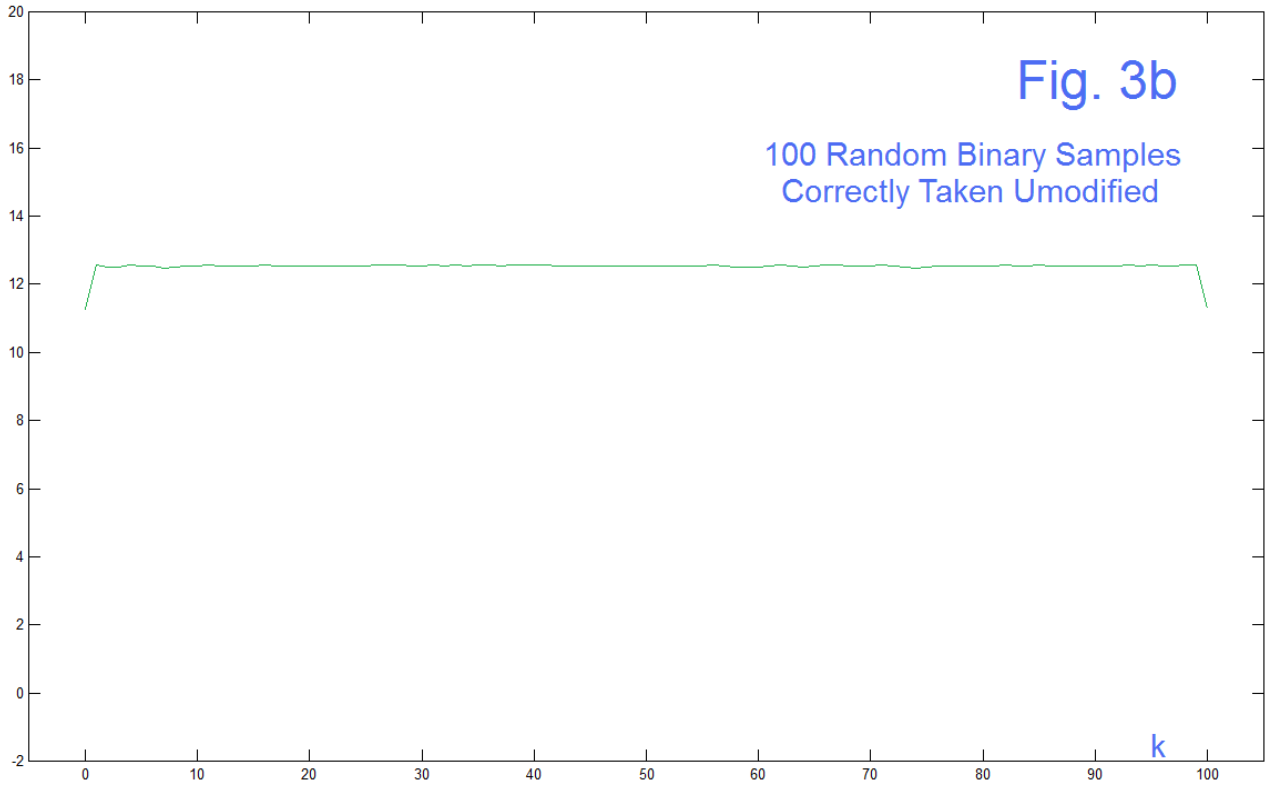
GENERATION AND VETTING OF PIECES

The thought here was inspired by comments on the "Watts Up With That" blog [7] by Robert G. Brown of the Physics Dept of Duke (screen name rgbatduke) which is one of the rare cases where anyone discusses actually testing noise generators. Brown points out that small sequences may well pass a test for randomness while combined into longer sequences the result may fail. This was the origin of the scenario of having students construct a long sequence by individual sub-sequence submissions, and of the thoughts about how the students might try to "improve" their homework submissions.

So the viewpoint is first one of "synthesis" where we want to see if we can put pieces together, particularly as each piece is individually "tested" to see if it is random. For example, a sub-sequence of [1 1 1 1 1 1 1 1 1 1 1 1 1 1] (14 ones) is perfectly fine, and popped out during testing of some ideas here. However, by itself it doesn't look random. Perhaps we should toss it (as one in 16,384). What if we just decide to toss anything more rare than 1/128, for example? The essence of Brown's point is that if you toss some proposed component as being unexpected (rare) for a particular length, for a much larger length it is almost guaranteed to be there somewhere - but if tossed as a component, it would then become conspicuous by its absence, and the full sequence could fail a test of randomness.

Fig. 3a and Fig. 3b show an example that looks very similar to that of Figures 1a, 1b, and 2. Looking at Fig. 3a we see that it is defined on levels +1 and -1 instead of +1 and 0, which is not too important, but here we note that Fig. 3a is NOT the output of a PRBS but rather was generated by Matlab's ***rand*** function [as 2*round(rand) - 1]. What is important about this is that the sequence is NOT subject to the feedback generation process, and this makes a huge difference. Note that Fig. 3a resembles the "scrambled"





PRBS example of Fig. 1b. Fig. 3a in fact shows a repeat of six 1's (probability 1/64 which we expect to see in 100 samples). We generated millions of samples, and happened to observe, by eye (observing plots like Fig. 3a), repeats of 11, 10, 9, and 14, among many shorter ones, like the six actually seen in this Fig. 3a. Looking at versions of Fig. 3a, people tend to suppose that there are too many clusters of repeats, a human failing we have mentioned above. We note that the long clusters are expected, and that we would find clusters as long as 22 (as in Fig. 1a) and longer. The essential difference between the Matlab ***rand*** approach and the PRBS is that <u>when we do</u> find the length-22 repeat using rand, we <u>would NOT expect</u> to have it followed by a sequence of 21 zeros, nor would we expect it to be preceded by an alternating sequence, as in Fig. 1a, etc. Rather we would expect it to be flanked by more typical random-looking portions.

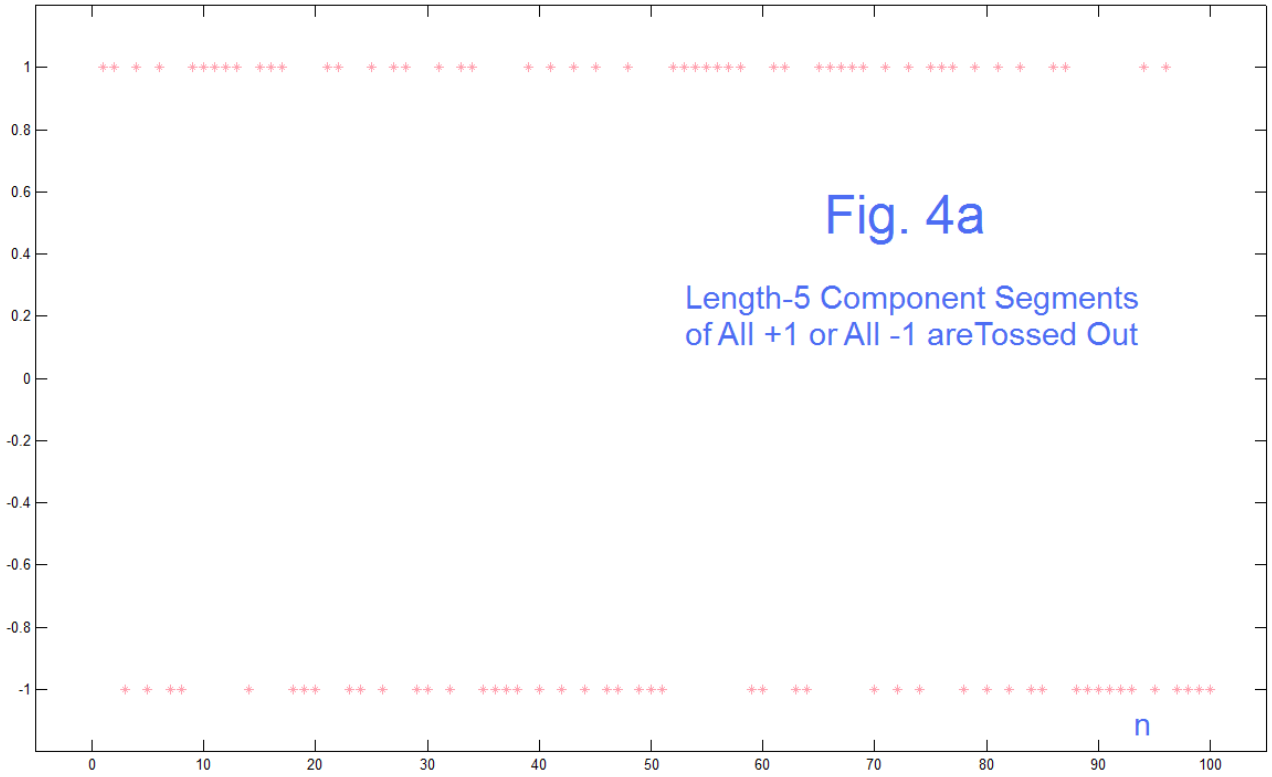
Keep in mind that Fig. 1a and Fig. 1b are PRBS outputs while Fig. 2 (the FFT spectrum) is Matlab ***rand***. (They are from two separate reports [1] and [2].) Fig. 2 was from ***rand***. Fig. 3b is from **2*round(rand)-1** and is thus binary (+1 and -1). Fig 3b is actually the average of 1,000,000 trials of length 200. The important thing is that we get the same 90% dips in the spectrum from the binary case here, Fig. 3b, that we got in the uniform distribution of ***rand*** in Fig. 2. This we would have guessed would have occurred.

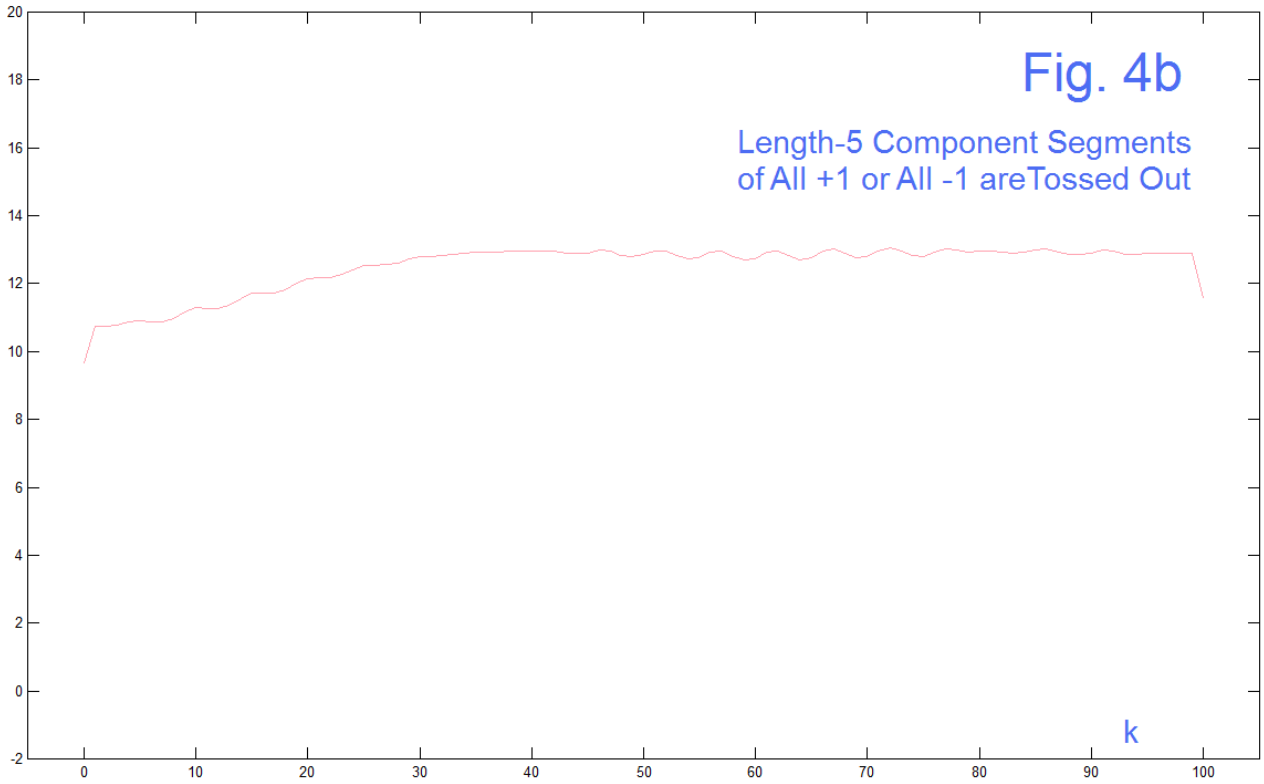
In total, Figs. 3a and 3b are a baseline case where we have not employed any attempts at making the sequences LOOK more random. Below we will make some (erroneous) attempts to "fix" the appearances. <u>Our principal tool for analyzing the results will be averaging of FFT's</u> as in Fig. 2 and Fig. 3b. We will note that the endpoint dips will remain. Direct examination of small portions of sequences in the time domain will also continue to be informative and useful.

Our first “refinement” ploy is related to the classical problem of throwing out outliers. In some cases, these are obvious mistakes – perhaps what might well be a transcription error of someone’s age as 350 instead of 35. (This leaves the problem of whether or not you choose to then discard, or to correct.) Yet an outlier from a normal distribution (Gaussian) could be more of a mathematical prank of nature. If you had numbers like 117, 189, 177, 215, 148, and 350, the number 350 might by an outlier at perhaps five standard deviations relative to the first five numbers. Mathematically, it could happen.

Perhaps we tend to forget that, matters such as the “central limits theorem” notwithstanding, it is hard to guarantee (thus to even claim) that a distribution is normal. A distribution of ages of a person with a popular name might be a reasonable match to normal, but how would one address the fact that the distribution predicted a person with that name of age -7! Distributions are originally just the results of plotting actual data. Mathematical equations of distributions are models, often known not to be correct in fact, or even found contrary to a reasonable theory of a phenomena.

All this being said, we can postulate data segments being analyzed and then possibly tossed out if they have what we choose to regard as outliers. Likely in a binary case we would agree that finding runs of five 1’s or -1’s are not unusual, but here we will choose this exclusion because we want to exaggerate the effect. To generate the data seen in Fig. 4a, we have taken length-5 random segments and if they are [1 1 1 1 1] or [-1 -1 -1 -1 -1] we toss them and try again. Clearly the result in Fig. 4a shows runs of 5, and indeed of 6 and 7 here. Our procedure does not exclude runs of 5 or greater than 5 in the final result, but only in the component sub-sequences. Back-to-back components of [-1 1 1 1 1] and [1 1 1 1 -1] would pass the test and give [-1 1 1 1 1 1 1 1 1 -1], a run of 8 in the output. Indeed, it is a simple matter to find many examples similar to Fig. 4a and





visually scan them; and after perhaps a hundred or more trials, 10 runs of 8 were found, but not a single example of a run of 9, so the coded procedure seems to work. Examining hundreds of graphs of a hundred or so samples is tedious. Can we learn anything from the spectrum (the FFT)? Yes.

Fig. 4b is the average of 1,000,000 runs of length 200 samples which eliminate runs of 5 (either +1's or -1's) and tries again. The spectrum shows three interesting things. First, there are the endpoint dips which continue to appear. Secondly there is a high-pass reduction in low-frequency. This too we might have expected in some general way, since we are eliminating some constant segments. More curious is perhaps the obvious "ripple" which seems to be related to increments of k=5, best seen perhaps from k=45 to k=75. This we have not investigated enough to discuss this except to note that 5 is the length of the segments tossed. The important thing is that the manipulation (some manipulation) becomes quite obvious in the spectrum, so we have a useful tool.

We postulated the "mischief" of Figs. 4a and 4b as being the result of our supposed contributing conscript (students) not wishing to turn in conspicuous homework. What is again actually just another example of an unwarranted but ubiquitous human prejudice against long runs, would be a "forced choice" by which a student figures that a run of two of anything is enough, and that certainly a break must follow (in the extreme, the proverbial "gambler's fallacy"). Fig. 5a shows a sequence that is generated by starting with two random samples. From then on the procedure codes by looking back at the previous two samples, and if they are the same, we must make a forced change. If they are different, we again make a random selection. Then we consider the two that are now the most recent, and so on. The result as in the example of Fig. 5a is remarkably smooth. There is never a case where more than two or anything occurs. Lots of [1 1 -1] and [-1 -1 1], alternating [1 1] and [-1 -1], and some [1 -1] stuff.

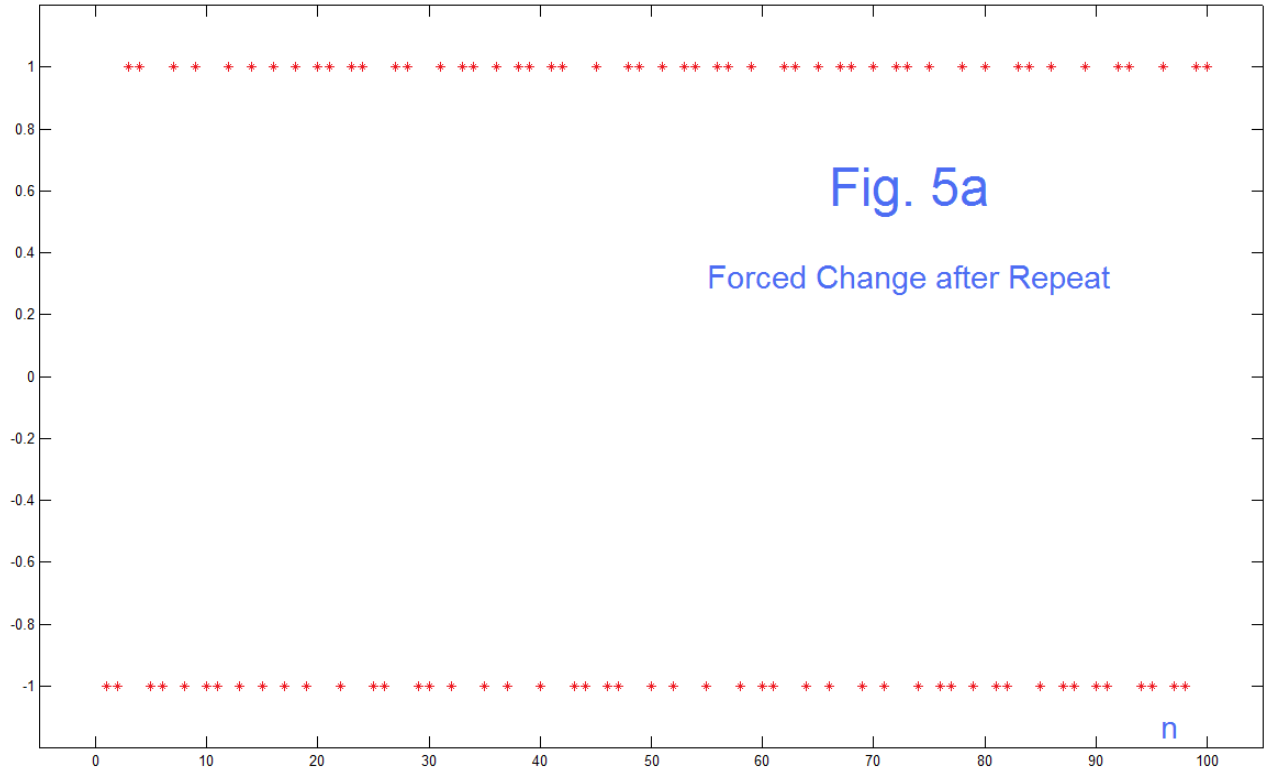


Fig. 5a
Forced Change after Repeat
n
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60
70
80
90
100

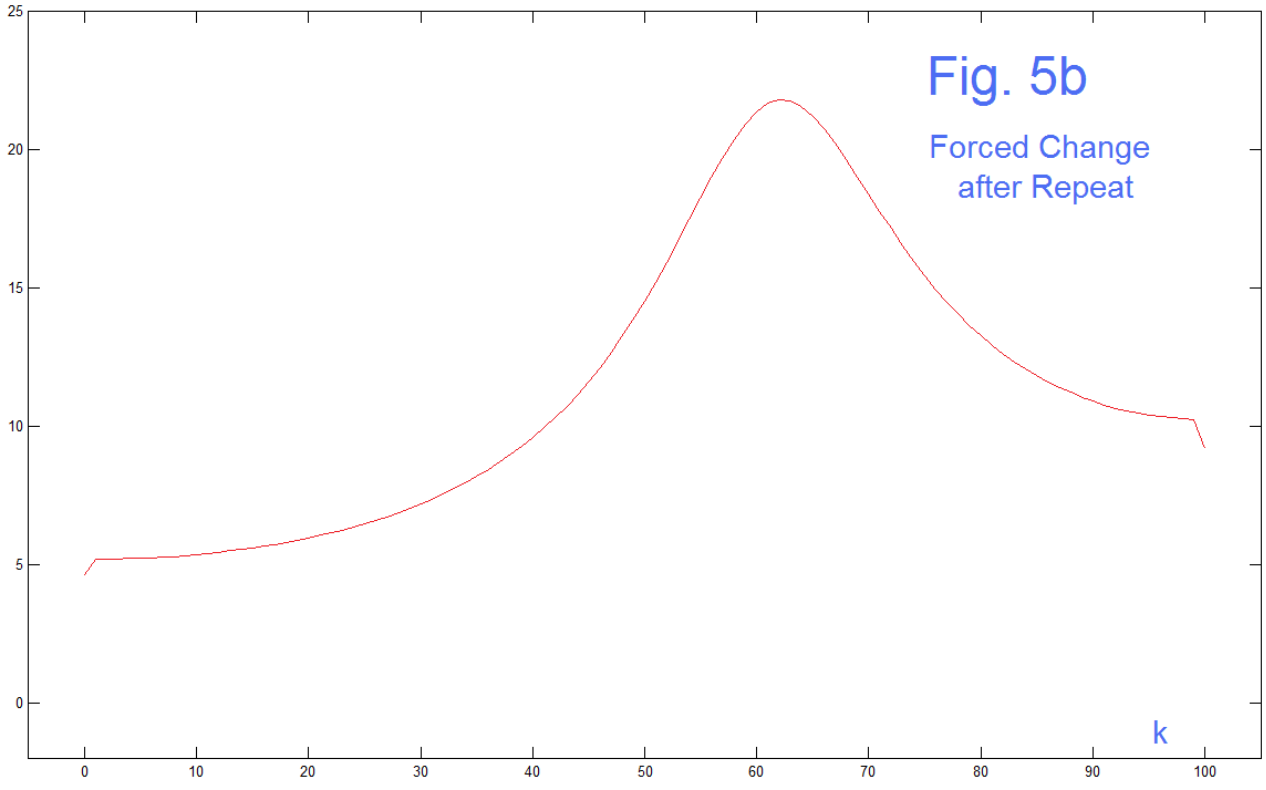
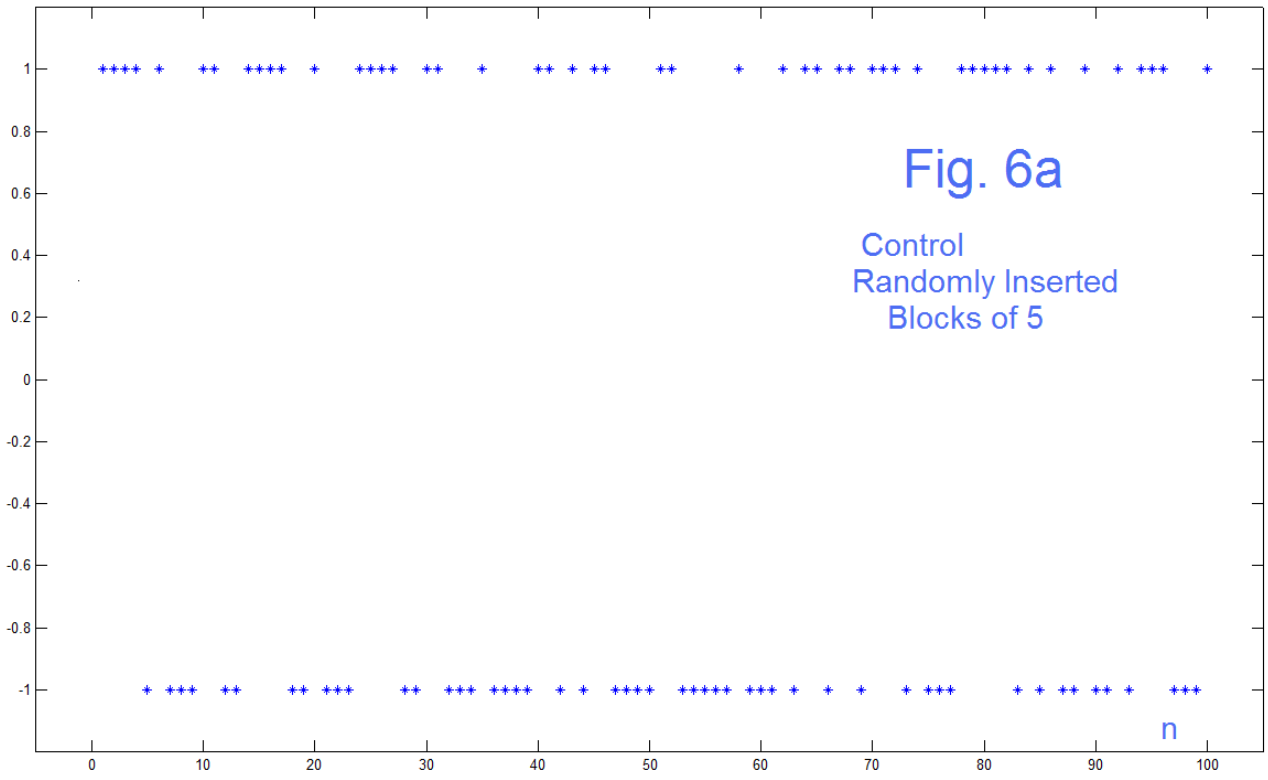


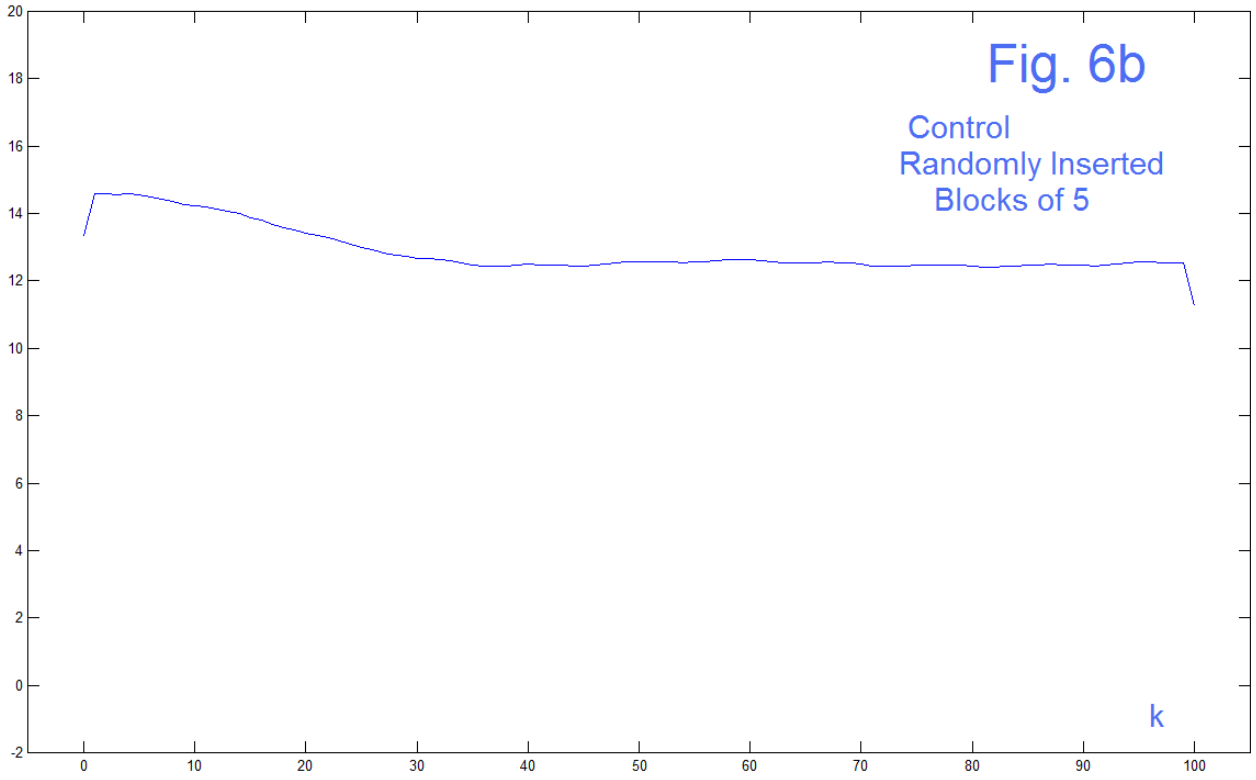
Fig. 5b
Forced Change
after Repeat
k
25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100

As smooth as the sequence (Fig. 5a) appears to the eye, the average spectrum (Fig. 5b) is very peaked. Again we have the endpoint dips, but the striking result is the band-pass like peak just above k=60. Because this is an average of length 200 sequences, the peak at about 65 is a frequency of 1/3 the sampling frequency, and thus of period 3. This we interpret as the dominance of segments like [1 1 -1] and [-1 -1 1] as broadened a bit by the other common elements. Once again, the spectrum makes the anomaly obvious.

Originally thought of as a "control" we have also tried a case where instead of tossing out trial blocks we inserted runs of 5 with a probability the same as the exclusion (1/16). This does not show very well in the time domain, likely because of the low probability, and because we have no way of knowing if any run of 5 is artificial. In fact, Fig. 6a looks very untypical, and repeats of the program do show runs of 6, 7, 8, etc. (All code is attached so you can make your own runs). But we are committed to NOT selecting here, so we stick to the first one that came out. To see what happens, more typically, we will go right to the spectral view as seen in Fig. 6b.

As expected, we see the endpoint dips. We also see the enhancement of low-frequencies, and this is what we would expect if we note that we are adding in long runs to what would otherwise be random. So it more or less looks like the reflection of Fig. 4b – except – what happened to the ripples? In the sense that this was a control or a test we were kind of expecting the length-5 ripples. But as much as we don't really understand the ripples on Fig. 4b anyway, their disappearance should not upset us. The evidence is probably trying to tell us something. If only these tests were not so tedious. The code is not difficult, but proper observation of the time sequence requires detailed examinations. The code to manipulate takes a lot of time (or needs to be written in a more clever manner), not to mention that and runs of spectral averaging require at least a million runs to clearly show ripples. In short – a good project for another time.





REFERENCES

[1] B. Hutchins, "Anomalies in Pseudo-Random Generators," Electronotes Application Note No. 402, Jan 10, 2014
http://electronotes.netfirms.com/AN402.pdf

[2] B. Hutchins, "A White Noise Curiosity," ***Electronotes***, Volume 22, Number 208 January 2012.
http://electronotes.netfirms.com/EN208.pdf

[3] Hutchins, B, "Theory and Application of Noise Generators in Electronic Music," ***Electronotes***, Vol. 8, No. 64, April 1976, pp 3-17. (Online posting in progress – when originals are found! - check webpage for requested items)

[4] Hutchins, B., "On Middle Length Pseudo-Random Sequences," Electronotes Mid-Month Letter #3 (75A) March 20, 1977

[5] John D. Cook, "Testing a Random Number Generator," Chapter 10 of ***Beautiful Testing***
http://www.johndcook.com/Beautiful_Testing_ch10.pdf

[6] B. Hutchins, “Fun with Red Noise,” Electronotes Application Note No. 384, September 1, 2012 http://electronotes.netfirms.com/AN384.pdf
and “More Fun with Red Noise,” Electronotes Application Note No. 412, May 25, 2014 http://electronotes.netfirms.com/AN412.pdf

[7] R. G. Brown, Comment by rgbatduke (Robert Brown, a Lecturer in Physics at Duke) in an extended comment June 13,2014 at 8:25 am
http://wattsupwiththat.com/2014/06/13/a-question-about-proxies-and-calibration-with-the-adjusted-temperature-record/
an earlier similar comment by Brown was made Jan 24, 2014 at 2:47 pm
http://wattsupwiththat.com/2014/01/21/sunspots-and-sea-level/

PROGRAM

Program here is offered for complete documentation as to how figures were produced. Code likely could be improved.

```
% randstudy.m


% Original – No manipulations
% Fig. 3a and Fig. 3b
N=1000000
XT1=zeros(1,200);
x=[];
for k=1:N
  for n=1:200
    x(n)=round(rand);
  end
  x=2*x-1;
  X=abs(fft(x));
  XT1=XT1+X;

  if k==1
    figure(1)
    plot(x(1:100),'*')
    axis([-5 105 -1.2 1.2])
  end
  end

XT1=XT1/N;
figure(2)
plot([0:100],XT1(1:101))
axis([-5 105 -2 20])
```

```
%
% Discard cases all 1's and all 0's and replace with new component of 5 samples
% Fig. 4a and Fig. 4b
N=1000000
XT2=zeros(1,200);
x=[];
for k=1:N
  x=[];
  n=1;
  while n<41
    for nn=1:5
      xx(nn)=round(rand);
    end
    g=1;
    if sum(xx)==0;g=0;end
    if sum(xx)==5;g=0;end
    if g==1
    x=[x xx];
    n=n+1;
    end
  end

  x=2*x-1;
  if k==1
    figure(3)
    plot(x(1:100),'*')
    axis([-5 105 -1.2 1.2])
  end

  X=abs(fft(x));
  XT2=XT2+X;
end
XT2=XT2/N;
figure(4)
plot([0:100],XT2(1:101))
axis([-5 105 -2 20])
```

```
%
% Discard three-peats – forced choice
% Fig. 5a and Fig. 5b
N=1
XT2=zeros(1,200);
x=[];
for k=1:N
  x=[];
  n=1;
  x(1)= 2*round(rand)-1;
  x(2)= 2*round(rand)-1;
  for n=3:200
    x(n)=2*round(rand)-1;
    if x(n-2)*x(n-1)==1
      x(n)=-x(n-1);
    end
  end

  if k==1
    figure(5)
    plot(x(1:100),'*')
    axis([-5 105 -1.2 1.2])
  end

  X=abs(fft(x));
  XT2=XT2+X;
end
XT2=XT2/N;
figure(6)
plot([0:100],XT2(1:101))
axis([-5 105 -2 25])
```

```
%
% Control – insertion of runs of 5
% Fig. 6a and Fig. 6b
N=1000000
XT2=zeros(1,200);
x=[];
for k=1:N
   x=[];
   n=1;
   while n<41
      for nn=1:5
         xx(nn)=round(rand);
      end

      if rand<0.01625; xx =(2*round(rand)-1)*ones(1,5); end
      x=[x xx];
      n=n+1;
  end

   x=2*x-1;
   if k==1
      figure(7)
      plot(x(1:100),'*')
      axis([-5 105 -1.2 1.2])
   end

   X=abs(fft(x));
   XT2=XT2+X;
end
XT2=XT2/N;
figure(8)
plot([0:100],XT2(1:101))
axis([-5 105 -2 20])
```

*******Program for 83% calculated for page (2)*****************

```
% randstudy1.m
M=0
N=1000000
for k=1:N
   for n=1:10
      x(n)=round(rand);
   end
   y=conv(2*x-1,[1 1 1]);
   if max(abs(y))>2
      %     if max(y)>2
      M=M+1;
   end
end
R=M/N
```