USING RANDOM NUMBER GENERATORS ON THE SENSOR SIGNAL PROCESSING SYSTEM

Jonathan A. Mills
Advanced Sensors Directorate
Research, Development, and Engineering Center

JUNE 1988

Approved for public release; distribution is unlimited.
using random number generators on the sensor signal processing system

This report describes the use of three random number generators installed on the SSPS VAX 11/785. There are three random number generators: a uniform (URAND), normal (ANRMN), and exponential (EXPRN).
CONTENTS

I. INTRODUCTION ........................................ 1
II. PROCEDURE ............................................... 1
APPENDIX - RANDOM NUMBER PROGRAM .................. A-1
I. INTRODUCTION

Random number generators are used in support of autonomous acquisition and processors for the sensor system processing system.

Three random number generators URAND (), ANRMRN (), and EXPRN () developed at Oak Ridge National Laboratory, are used for multiplicative noise added to images, computer benchmarking, and other scientific applications. URAND is a uniform number generator, ANRMRN is a normal number generator, and EXPRN is an exponential random number generator.

II. PROCEDURE

The random number generator requires an integer array (a block of memory) of dimension 4 (suggested name iseed) for the seed and a diskfile ranseed.dat in the working directory in order to execute. A seed is an initial value given to a random number generator and each individual seed produces a certain set of random numbers. Ranseed.dat is a diskfile that is opened by the program to hold the current seed and should initially contain a single record in 4010 format (4 octal entries of length ten) of all zeros. RANFIX is the subroutine that initializes the program and must be called for the generators to work. RANFIX has to be called only once for each user program. If a certain seed is desired, the subroutine rango () is called to initialize the new seed, ex. call rango(iseed). If iseed(1), the first entry, in ranseed.dat is less than or equal to zero, then the seed is not initialized. If iseed(1) is greater than zero, the seed in ranseed.dat is used as the initial point for the random number generation. To save the current seed in the random number generator URAND, the subroutine ransav() is called, ex. call ransav(iseed). Ransav() copies the current seed into ranseed.dat and the array iseed. If the current seed is desired, call the subroutine ranfet(), ex. call ranfet(iseed). The Random Number Generator Parameters table contains the high value, low value, and the delta of these values for each random number generator (the default iseed is used). In the table, there are various amounts of random numbers generated ranging from 100-1,000,000 for each random number generator.

NOTE

The normal and exponential generators use the uniform generator (URAND) to achieve their distributions. In other words, URAND is used to get random values for the generators ANRMRN() and EXPRN() to manipulate into their distribution.

The described functions and subroutines are in the library DISK$USERDISK: [SSPSLIB]RANDOM.OLB in the sensor signal processing system's VAX 11/785 of the Advanced Sensors Directorate. Provided, in this report, is an example of how the random number generators are used in a program and a figure showing a plot of the output of this program shown. The Appendix contains the random number program.
Figure. Plot of the program output.
<table>
<thead>
<tr>
<th>RANDOM NUMBER GENERATOR</th>
<th>POINTS</th>
<th>HIGH</th>
<th>LOW</th>
<th>DELTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIFORM</td>
<td>1,000,000</td>
<td>0.99999  == 1</td>
<td>5.52E-7  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td></td>
<td>500,000</td>
<td>0.99999  == 1</td>
<td>3.38E-6  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td></td>
<td>100,000</td>
<td>0.99999  == 1</td>
<td>3.79E-6  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>0.99986  == 1</td>
<td>3.79E-6  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
<td>0.99864  == 1</td>
<td>4.06E-4  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.99342  == 1</td>
<td>0.00143  == 0</td>
<td>APPROX. = 1</td>
</tr>
<tr>
<td>NORMAL</td>
<td>1,000,000</td>
<td>5.252260</td>
<td>-3.66426</td>
<td>10.61869</td>
</tr>
<tr>
<td></td>
<td>500,000</td>
<td>4.526156</td>
<td>-4.321178</td>
<td>8.847335</td>
</tr>
<tr>
<td></td>
<td>100,000</td>
<td>4.325014</td>
<td>-4.551002</td>
<td>8.876016</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>3.684598</td>
<td>-3.702339</td>
<td>7.386937</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
<td>3.854763</td>
<td>-3.518877</td>
<td>7.373640</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3.023248</td>
<td>-2.448892</td>
<td>5.472140</td>
</tr>
<tr>
<td>EXPONENTIAL</td>
<td>1,000,000</td>
<td>14.76969</td>
<td>1.979E-6  == 0</td>
<td>APPROX. = 14.8</td>
</tr>
<tr>
<td></td>
<td>500,000</td>
<td>13.37706</td>
<td>2.533E-7  == 0</td>
<td>APPROX. = 13.4</td>
</tr>
<tr>
<td></td>
<td>100,000</td>
<td>12.88441</td>
<td>1.536E-5  == 0</td>
<td>APPROX. = 12.9</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>10.94290</td>
<td>5.423E-5  == 0</td>
<td>APPROX. = 10.9</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
<td>10.19361</td>
<td>3.790E-6  == 0</td>
<td>APPROX. = 10.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3.612674</td>
<td>0.005789  == 0</td>
<td>APPROX. = 3.6</td>
</tr>
</tbody>
</table>
EXAMPLE:

Create the file RANSEED.DAT in your working directory

A SIMPLE PROGRAM

INTEGER ISEED(4)
REAL UNIFORM(1000),NORMAL(1000),EXP(1000),FRAN,DUMMY,DUMMY2

DO I=1,4
ISEED(I)=0
END DO
OPEN(UNIT=11,NAME='RANSEED.DAT',STATUS='OLD',FORM='FORMATTED')
WRITE(11,10) (ISEED(I),I=1,4)
10 FORMAT(4Ol.0)
CLOSE(UNIT=11)

CALL RANFIX

DO I=1,1000

UNIFORM(I)=URAND(FRAN)
NORMAL(I)=ANMRN(DUMMY)
EXP(I)=EXPRN(DUMMY2)

END DO
END
APPENDIX

RANDOM NUMBER PROGRAM
SUBROUTINE RANFIX
C SUBROUTINE TO INITIALIZE UNIFORM RANDOM NUMBER GENERATOR PACKAGE
C
C SUBROUTINES RANSET, URAND, AMNRN AND EXPRN COMPRISE A PACKAGE OF RANDOM
C NUMBER ROUTINES OBTAINED FROM THE OAK RIDGE NATIONAL LABORATORY (ORNL).
C REFERENCE: E. J. MCGRATH AND D. C. IRVING, ORNL-RSIC-38 (VOL. II),
C TECHNIQUES FOR EFFICIENT MONTE CARLO SIMULATION, VOLUME II,
C RANDOM NUMBER GENERATION FOR SELECTED PROBABILITY DISTRIBUTIONS,
C URAND IS A MACHINE INDEPENDENT UNIFORM RANDOM NUMBER GENERATOR U(0,1).
C AMNRN GENERATES NORMALLY DISTRIBUTED RANDOM NUMBERS N(0,1).
C EXPRN GENERATES EXPONENTIALLY DISTRIBUTED RANDOM NUMBERS WITH UNITY MEAN.
C RANSET IS AN INITIALIZATION ROUTINE FOR URAND.
C
C RANFIX, RANGO, RANSAV AND RANFET ARE SPECIAL ROUTINES ADDED FOR IMPLEMENTATION
C ON THE VAX.
C
C TO MAKE POSSIBLE RESTARTING THE SEQUENCE WITH SPECIFIED SEEDS, ONE PROVIDES
C IN THE CALLING AN INTEGER ARRAY (SUGGESTED NAME ISEED) OF DIMENSION 4.
C
C IN ADDITION, A DISK FILE NAMED RANSEED.DAT IS REQUIRED FOR STORING THE
C CURRENT VALUES OF ISEED. THIS FILE SHOULD INITIALLY CONTAIN A SINGLE RECORD
C IN 4010 FORMAT WITH SUGGESTED ENTRIES ALL ZERO.
C
C TO INITIALIZE THE PACKAGE CALL RANFIX. THIS NEED BE DONE ONLY ONE TIME,
C TO INITIALIZE TO A GIVEN RANDOM NUMBER SEED, ONE THEN CALLS RANGO(ISEED).
C IF ISEED(1), STORED IN RANSEED.DAT, IS LESS THAN OR EQUAL TO ZERO, THEN NO
C INITIALIZATION TAKES PLACE. OTHERWISE, THE ISEED ARRAY STORED IN
C RANSEED.DAT IS USED AS THE STARTING POINT FOR THE RANDOM NUMBER GENERATION.
C
C TO SAVE THE CURRENT SEED IN URAND FOR FURTHER RANDOM GENERATION, CALL
C RANSAV(ISEED). THE CURRENT SEED IS OVERWRITTEN INTO RANSEED.DAT, IN
C ADDITION TO THE CURRENT DATE AND TIME. THIS SEED IS ALSO PLACED IN THE
C ISEED ARRAY.
C
C TO FETCH THE CURRENT SEED IN URAND (BUT NOT TO SAVE IT), CALL RANFET(ISEED).
C
C IN ORDER TO BE ABLE TO REPEAT SIMULATION RUNS WITH THE SAME RANDOM
C SEQUENCE, IT IS SUGGESTED THAT THE INITIAL AND FINAL ISEED ARRAYS BE
C PRINTED.
C
C DATA MAXINT/"17777777777/
C DATA NSTART/2001/
C CALL RANSET(MAXINT,NSTART)
C RETURN
END
SUBROUTINE RANSET(MAXINT,NSTRT)
C SUBROUTINES RANSET, URAND, AMNRN, EXPRN OBTAINED FROM OAK RIDGE
C NATIONAL LABORATORY (ORNL)
C COMMON /MIRNG/RAN(10), GEN(10), NWRD, BASE, MOD, FBASE, FMOD
C INTEGER RAN, GEN, BASE, CARRY, REM
C MAXI=MAXINT/4
C IB=0
C BASE=1
C 99 IF(BASE.GT.MAXI) GO TO 100
C BASE=BASE*4
C IB=IB+1
C GO TO 99
C 100 BASE=2*IB
C FBASE=BASE
C NWRD=47/IB+1
REAL FUNCTION URAND(FRAN)
C SUBROUTINES RANSET,URAND,AMNMRN,EXPRN OBTAINED FROM OAK RIDGE
NATIONAL LABORATORY (ORNL)
COMMON /MIRNG/RAN(10),GEN(10),NWRD,BASE,MOD,FBASE,FMOD
DIMENSION SUM(10)
INTEGER RAN,GEN,BASE,CARRY,SUM,PROD,HPROD
DO 30 IS=1,NWRD
30 SUM(IS)=0
DO 1 IG=1,NWRD
DO 2 IR=1,N2
IS=IR+IG-1
PROD=RAN(IR)*GEN(IG)
HPROD=PROD/BASE
LPROD=PROD-HPROD*BASE
SUM(IS)=SUM(IS)+LPROD
IF(IS.LE.NWRD) SUM(IS+1)=SUM(IS+1)+HPROD
1 CONTINUE
N2=NWRD-1
DO 5 IS=1,N2
CARRY=SUM(IS)/BASE
SUM(IS)=SUM(IS)-CARRY*BASE
SUM(IS+1)=SUM(IS+1)+CARRY
5 CONTINUE
SUM(NWRD)=SUM(NWRD)-MOD*(SUM(NWRD)/MOD)
DO 20 IS=1,NWRD
20 RAN(IS)=SUM(IS)
FRAN=SUM(1)
DO 10 IS=2,NWRD
10 FRAN=FRAN/FBASE+SUM(IS)
FRAN=FRAN/FMOD
URAND=FRAN
RETURN
END
FUNCTION ANMRN(DUMMY)
C SUBROUTINES RANSET, URAND, AMNMRRN, EXPRN OBTAINED FROM OAK RIDGE
C NATIONAL LABORATORY (ORNL)
R=URAND(D)
IF(R.GT.0.8638) GO TO 10
ANMRN=2.*(URAND(D)+URAND(D)+URAND(D)-1.5)
RETURN
10 IF(R.GT.0.9745) GO TO 20
   ANMRN=1.5*(URAND(D)+URAND(D)-1.0)
RETURN
20 IF(R.GT.0.997302039) GO TO 100
25 X=6.*(URAND(D)-3.0)
   Y=0.358*URAND(D)
   XSQ=A*X
   GX=17.49731196*EXP(-XSQ*.5)
   AX=ABS(X)
   IF(AX.GT.1.0) GO TO 30
   IF(Y.GT.(GX-17.44392294*4.73570326*XSQ+2.15787544*AX)) GO TO 25
   ANMRN=X
RETURN
30 AX3=2.36785163*(3.-AX)**2
   IF(AX.GT.1.5) GO TO 40
   IF(Y.GT.(GX-AX3-2.15787544*(1.5-AX))) GO TO 25
   ANMRN=X
RETURN
40 IF(Y.GT.(GX-AX3)) GO TO 25
   ANMRN=X
RETURN
100 X=SQRT(9.+2.*EXPRN(D))
   IF(URAND(D).GT.3./X) GO TO 100
   IF(URAND(D).GT.0.5) X=-X
   ANMRN=X
RETURN
END

FUNCTION EXPRN(DUMMY)
C SUBROUTINES RANSET, URAND, AMNMRRN, EXPRN OBTAINED FROM OAK RIDGE
C NATIONAL LABORATORY (ORNL)
C EXPONENTIAL RANDOM NUMBER BY VON NEUMANN REJECTION TECHNIQUE
I=0
100 R=URAND(D)
   X=R
105 Y=URAND(D)
   IF(X.LT.Y) GO TO 120
110 X=URAND(D)
   IF(X.LT.Y) GO TO 105
115 I=I+1
   GO TO 100
120 AI=I
   EXPRN=R+AI
RETURN
END

SUBROUTINE RANSAV(ISEED)
C SUBROUTINE TO SAVE RANDOM NUMBER SEED ON VAX
COMMON /MIRNG/RAN(10),GEN(10),NWRD,BASE,MOD,FBASE,FMOD
INTEGER RAN,GEN
CHARACTER*9 IDATE
CHARACTER*8 ITIME
DIMENSION ISEED(1)
DATA IRAN/11/ DO 200 I=1,4
C SUBROUTINE TO REINITIALIZE RANDOM NUMBER GENERATOR ON VAX
C IF (ISEED(1).EQ.0) NO INITIALIZATION
C IF (ISEED(1).LT.0) INITIALIZE WITH ABS(ISEED)
COMMON /MIRNG/RAN(10),GEN(10),NWRD,BASE,MOD,FBASE,FMOD
INTEGER RAN,GEN
DIMENSION ISEED(1)
DATA IRAN/il!
ICLOSE=1
IF (ISEED(1).GE.0) GO TO 180
ICLOSE=0
ISEED(1)=-IABS(ISEED(1))
GO TO 190
180 CONTINUE
OPEN(UNIT=IRAN,NAME='RANSEED.DAT',STATUS='OLD',FORM='FORMATTED')
READ(IRAN,101) (ISEED(I),I=1,4)
101 FORMAT(4010)
IF (ISEED(1).LE.0) GO TO 210
190 CONTINUE
DO 200 I=1,4
RAN(I)=ISEED(I)
200 CONTINUE
IF (ICLOSE.LE.0) RETURN
CLOSE(UNIT=IRAN)
RETURN
END
SUBROUTINE RANFET(ISEED)
COMMON /MIRNG/RAN(I0),GEN(I0),NWRD,BASE,CARRY,REM
DIMENSION ISEED(I)
DO 200 I=1,4
ISEED(I)=ISEED(I)
200 CONTINUE
RETURN
END
FUNCTION RAYLGH(R)
C SUBROUTINE TO GENERATE SAMPLES FROM RAYLEIGH DISTRIBUTION WITH SIGMA=1
DATA SQRT2/1.4142136/
R=SQRT2*SQRT(-ALOG(URAND(DUM)))
RAYLGH=R
RETURN
END
DISTRIBUTION

U. S. Army Materiel System
Analysis Activity
ATTN: AMXSYS-MP
Aberdeen Proving Ground, MD 21005
1

IIT Research Institute
ATTN: GACIAC
10 W. 35th Street
Chicago, IL 60616
1

AMSMI-RD, Dr. McCorkle
Dr. Rhoades
- RD-AS-SS, Mr. John L. Hatcher
   Mr. S. Richard F. Sims
   Mr. Jonathan A. Mills
   10
- RD-AS
- RD-CS-R
- RD-CS-T
AMSMI-GC-IP, Mr. Fred Bush

DIST-1/(DIST-2 blank)