# DECUS
## PROGRAM LIBRARY

| | |
|---|---|
| DECUS NO. | 8-300 |
| TITLE | NOISE GENERATOR |
| AUTHOR | H. D. Schenk |
| COMPANY | Deutsche Forschungs - und Versuchsanstalt, Flughaven, Germany |
| DATE | July 31, 1969 |
| SOURCE LANGUAGE | MACRO-8 |

# NOISE GENERATOR

DECUS Program Library Write-up — DECUS No. 8-300

## ABSTRACT

The program, which is written as a subroutine, creates a pseudo-random voltage with a gaussian probability density function. Also the appropriate binary noise is available. The bandwidth can be selected by the programmer.

## REQUIREMENTS

### Storage

The program requires one page of the memory.

### Equipment

Basic PDP8 with 12-bit Digital Output (Device Code No. 30) and one Digital-to-Anlog Converter (Device Code No. 55).

## USAGE

### Loading

The program is written as a subroutine and ends with a PAUSE Statement. It must be assembled with the main program.

### Calling Sequence

Initially call the program with "JMS NOIINI". This part sets the timer and the register REGA is loaded with the content of the switch register. Return is to the next instruction. Normally call the generator with "JMS NOIGEN". The number of callings per second determines the bandwidth of the noise voltage.

### Switch Setting

The content of the switch register is strobed into the accumulator, if the initial subroutine is called. It determines the presetting of register A. In the subroutine NOIGEN the switch register is never more used.

### Start Up

None.

$$T = 8.388.607 * \frac{1}{f_T} .$$

If the subroutine is called once in 20 ms, the bit pattern is repeated not before 46 hours.

The power spectrum of this binary noise is equal to the $\sin^2 x/x^2$ function with the first zero at $f_T$.
The next part of the program uses the binary noise to create analog noise with a gassian amplitude distribution. For this the binary noise is filtered by an ideal rectangular low pass filter with a bandwidth of about 1/15 of $f_T$. The pulse response of such an ideal filter is

$$G_{(t)} \sim \frac{\sin 2\pi(t-t_o)/f_g}{2\pi(t-t_o)/f_g}$$

The response of any inputsignal to this filter can be obtained by using

$$y_{(t)} = \int_0^\infty G_{(u)} * x_{(t-u)}\, du$$

or by replacing the integral by a discret adding

$$y_{(t)} \sim \sum_{n=1}^{\infty} a_n * x_{(t-n\Delta t)}.$$

The part $x(t-n\Delta t)$ means that the input voltage must be delayed by predetermined values. Since in the program $x_{(t)}$ is a binary pattern and comes from a shift register with 36 bits the above equation can be performed on a simple way. The coefficients $a_n$ determine the values of $G_{(t)}$. The limited length of the shift register do not allow to integrate to the infinite but only to n = 36. So the function sinx/x is only used between $x = -3\pi$ and $x = +3\pi$ . The remaining part is seperated into 36 intervalls and the appropriate values are stored in a table. The values are calculated by an appropriate FOCAL Program which uses the equation:

Miscellaneous

The following labels are used in this subroutine

| | |
|---|---|
| ANALOG | NOIDUR |
| BITCTR | NOIGEN |
| BITHIG | NOIINI |
| EBIT23 | RAND |
| EBIT5 | RAOFFS |
| EXOR | REGA |
| EXRES | REGB |
| LOADRA | REGC |
| M12 | TABBEG |
| NOICTR | TABENT |

## RESTRICTIONS

None.

## DESCRIPTION

Discussion

The program makes use of a 36 bit shift register, containing three words named REGA, REGB and REGC. Each time the core, named NOICTR, overflows the full register is shifted one bit to the right. The input of the register is determined by the subroutine EXOR, which performs a logical exclusive-or-gate from bit 5 and 23 of the shiftregister. One part of the shiftregister is transferred to the Digital Output Register No. 30. and gives a binary noise with the two levels zero and one. This voltage has a cycletime which depends on the callings per seconds. The period of a whole cycle is calculated by

$$T = (2^n - 1) * \frac{1}{f_T}$$

with $n$ number of bits of the shiftregister

$f_T$ calling frequency of the noise generator.

Since the exclusive-or-operation is performed between bit 5 and bit 23 the number n becomes 23 and

$$a_n = \frac{\sin (n-19)\pi/6}{(n-19)\pi/6}$$

with $n \in 1, 2, \ldots, 37$

The filter can easily be changed by loading the table with other coefficients. The program masks the Binary pattern of the shiftregister with the table and adds the appropriate values to form the output voltage. To get a mean of zero volt an offset voltage is added before converting the voltage by the 10 bit D/A converter.

The amplitude of the analog noise is 1,15 $V_{RMS}$ and the maximal possible value is 4,87 V. So the crest factor becomes 4,2.

Example

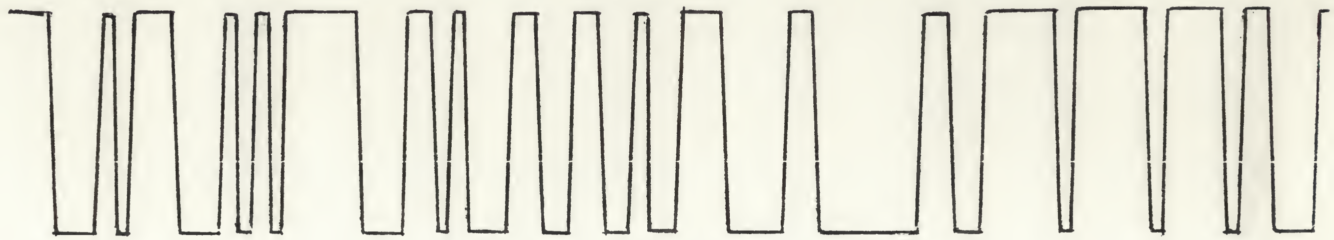
The example in the appendix shows the binary noise and the analog noise. Also the correlation function and the power spectrum of both are calculated. Last not least the probability density function of the analog noise is shown.

## PROGRAM

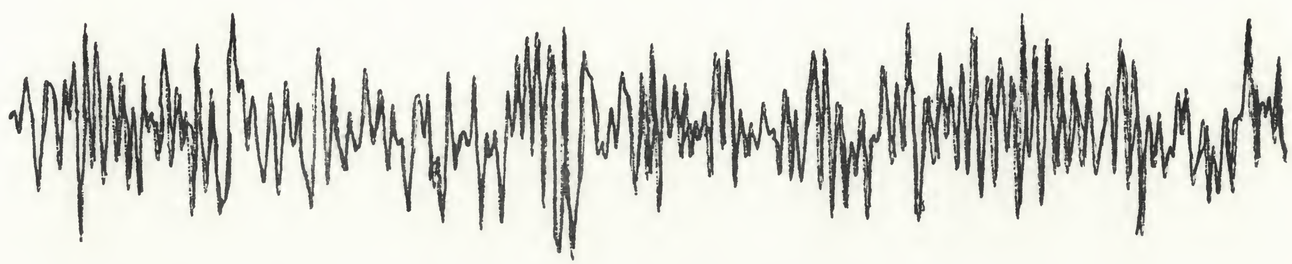
Program listing

A listing of the source is given in the appendix.

EXAMPLE



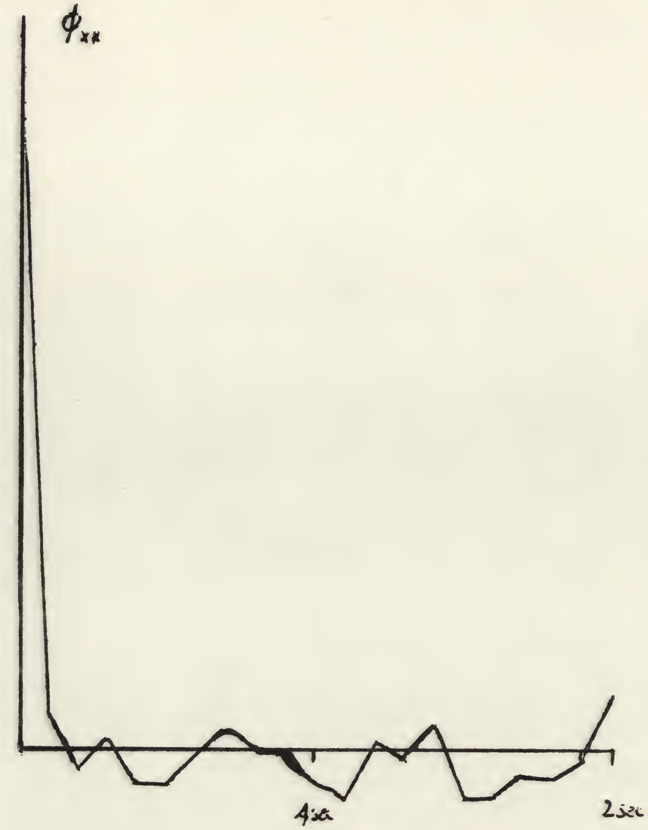
a) Binary Noise (Clockfrequency 10 cps, 2 V/cm, 0,5 sec/cm)



b) Analog Noise (Clockfrequency 50 cps, 2 V/cm, 2 sec/cm)

Output voltages of the Noise Generator

EXAMPLE

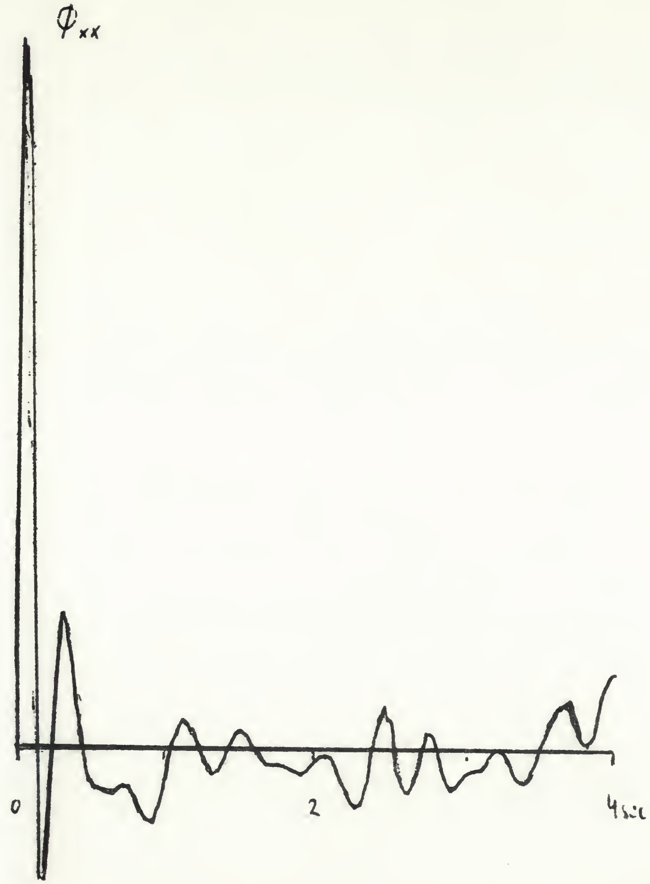


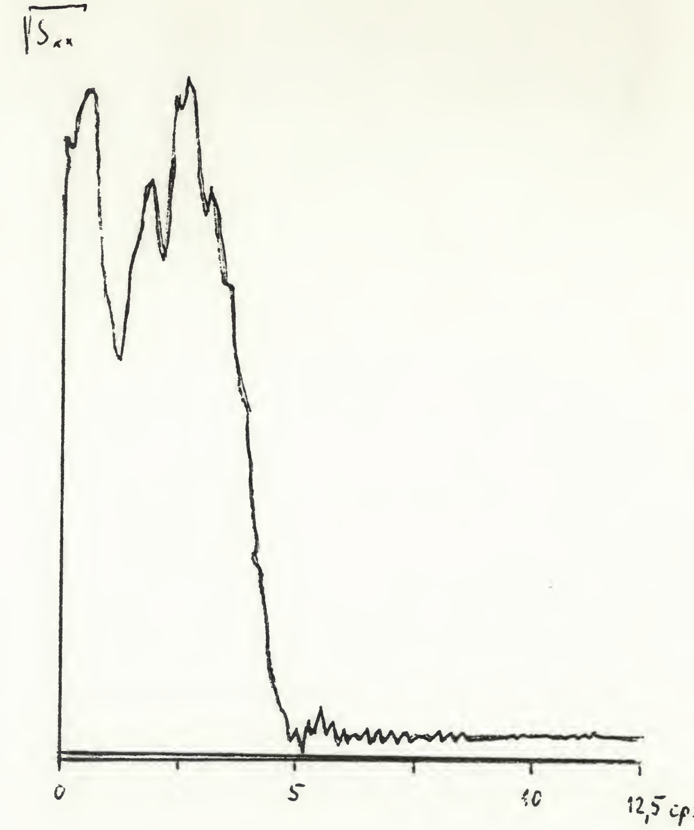

Correlationfunction


Powerspectrum

of the Binary Noise (Clockfrequency 10 cps,
Samplefrequency 50 cps)

EXAMPLE



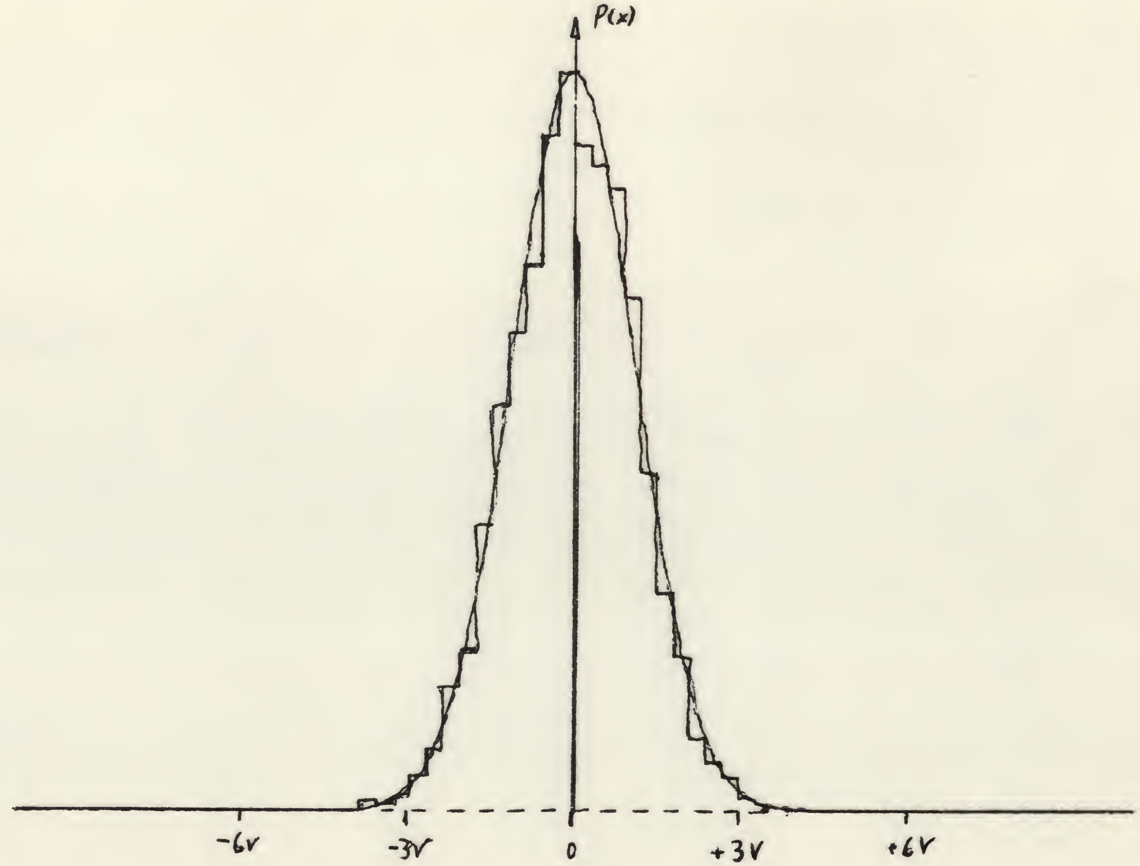




Correlationfunction

Powerspectrum

of the Analog Noise (Clockfrequency 50 cps, Samplefrequency 25 cps)

EXAMPLE




Histogram of the analog noise (Clockfrequency 25 cps, Samplefrequency 25 cps, N = 3000) Class width 0,3 V.

```
                              / DEVLE-BRG.-L-LA
                              / USE 09 - 03
                              / NOISE GENERATOR
                              / 31 JULY 1969
                              / PDP8, BASIC CONFIGURATION
/
/ CALL SUBROUTINE BY JMS NOIGEN.
/ RETURN IS TO FOLLOWING INSTRUCTION.
/ INITIALLY CALL JMS NOIINI.
/ OUTPUT IS VIA REGISTER #30 FOR BINARY NOISE
/ AND VIA D/A-CONVERTER #55 FOR ANALOG NOISE.
/

NOIINI, 0                   / INITIALIZE
        TAD NOIDUR
        DCA NOICTR          / SET COUNTER
        LAS                 / GET CONTENT OF SR
        STL                 / SET LKB
        RAR                 / SHIFT ONE BIT TO THE RIGHT
        DCA REGA            / STORE IN REGISTER A
        DCA REGB            / CLEAR REGISTER B AND C
        DCA REGC
        JMP I NOIINI
/
NOIGEN, 0                   / NOISE GENERATOR
        ISZ NOICTR          / TIME TO CREATE VALUE ?
        JMP I NOIGEN        / NO
        JMS EXOR            / EXL. OF OPERATION OF BIT 5 AND 23
                            / RESULT IS IN LKB
        TAD REGA
        RAL                 / SHIFT REGISTER A
        DCA REGA
        TAD REGB
        RAR                 / SHIFT REGISTER B
        DCA REGB
        TAD REGC
        RAR                 / SHIFT REGISTER C
        6302                / LOAD REGISTER #30 WITH BIN. NOISE
        DCA REGC
        JMS ANALOG          / GET ANALOG NOISE
        TAD NOIDUR
        DCA NOICTR          / RESET COUNTER
        JMP I NOIGEN        / EXIT
/
REGA,   0
REGB,   0
REGC,   0                   / REGISTERS
NOICTR, 0                   / TIMING COUNTER
NOIDUR, -1                  / COUNTER BEGIN
/
EXOR,   0                   / EXCLUSIVE OR OPERATION
                            / OF BIT 5 AND BIT 23
                            / RESULT IN LINKBIT
        DCA EXRES           / CLEAR
        TAD REGA            / GET REGISTER A
        AND EBIT5           / MASK BIT 4
        SZA CLA             / BIT HIGH ?
        ISZ EXRES           / YES, LOAD EXRES
        TAD REGB            / GET REGISTER B
```

```
        AND EBIT23          / MASK BIT 10
        SZA CLA             / BIT HIGH ?
        ISZ EXRES           / YES, EXRES+1
        TAD EXRES           / RESULT OF OPERATION IN BIT 11
        RAR                 / SHIFT IN LKB
        CLA
        JMP I EXOR          / EXIT
/
EXRES,  Ø                   / BIT 11 CONTAINS RESULT
EBIT5,  200                 / MASK FOR BIT 5
EBIT23, 2                   / MASK FOR BIT 23
/
ANALOG, Ø                   / SET ANALOG NOISE VOLTAGE
                            / BY DIGITAL FILTERING
        TAD TABENT
        DCA BITCTR          / SET TABLE ENTRY
        DCA RAND            / CLEAR OUTPUT
        TAD M12
        DCA NOICTR          / SET COUNTER TO FULL WORD
        TAD REGA            / GET REGISTER A
        JMS LOADRA          / LOAD OUTPUT
        TAD M12
        DCA NOICTR          / RESET COUNTER
        TAD REGB            / GET REGISTER B
        JMS LOADRA          / LOAD OUTPUT
        TAD M12
        DCA NOICTR          / RESET COUNTER
        TAD REGC            / GET REGISTER C
        JMS LOADRA          / LOAD OUTPUT
        TAD RAND            / GET OUTPUT
        TAD RAOFFS          / ADD OFFSET VOLTAGE
        6552                / CONVERT TO ANALOG BY D/A #55
        CLA
        JMP I ANALOG        / EXIT
/
LOADRA, Ø                   / LOAD OUTPUT WITH APPROPRIATE
                            / VALUE FROM TABLE ( SIN X / X )
        CLL                 / CLEAR LINKBIT
        RAL                 / SHIFT BITS IN LKB
        SZL                 / CHECK IF BIT IS ZERO
        JMP BITHIG          / FOUND BIT HIGH
        ISZ BITCTR          / INCREMENT TABLEPOINTER
        ISZ NOICTR          / HAVE ALL 12 BITS ?
        JMP LOADRA+1        / NO
        JMP I LOADRA        / YES, EXIT
BITHIG, DCA EXRES           / FOUND BIT HIGH, SAVE AC
        TAD I BITCTR        / GET VALUE FROM TABLE
        TAD RAND            / ADD TO OUTPUT APPROPRIATE VALUE
        DCA RAND            / STORE
        TAD EXRES           / GET SAVED AC
        JMP LOADRA+5
/
M12,    -14                 / BITS PER WORD
RAND,   Ø                   / STORAGE FOR OUTPUT
BITCTR, Ø                   / POINTER TO TABLE
TABENT, TABBEG              / TABLE ENTRY
/
DECIMAL
RAOFFS, -319                / OFFSET
TABBEG, Ø                   / TABLE FOR (FSIN X)/X
        6                   / FOR X=PI/6.*(N-19)
```

```
        10
        13
        12
        7
        0
        -9
        -17
        -21
        -21
        -14
        0
        19
        41
        64
        83
        96
        100
        96
        83
        64
        41
        19
        0
        -14
        -21
        -21
        -17
        -9
        0
        7
        12
        13
        10
        6
/
OCTAL
PAUSE
```