AMRL-TR-65-133
(SUP. II)

SYSTRAN (SYSTEMS ANALYSIS TRANSLATOR): A DIGITAL COMPUTER PROGRAM

SUPPLEMENT TWO

E. B. WEIS, JR., M.D.

D. J. LAJEUNESSE

Foreword

This report is the second in a series of reports to supplement AMRL-TR-65-133 SYSTRAN (SYSTEMS ANALYSIS TRANSLATOR): A DIGITAL COMPUTER PROGRAM, D. J. Lajeunesse, E. B. Weis, Jr., T. J. Hogan, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, July 1965. The work described herein was performed during the period from August 1965 to October 1966. The work was done in support of Project 7231, "Biomechanics of Aerospace Operations," Task 723101, "Effects of Vibration and Impact."

The report describes extensions of the capability of SYSTRAN and program modifications and corrections.

Copies of the Binary or the Symbolic program will be available to authorized requestors from Mr. D. J. Lajeunesse of the Digital Computation Division, Wright-Patterson Air Force Base, Ohio.

This technical report has been reviewed and is approved.

WAYNE H. McCANDLESS
Technical Director
Biomedical Laboratory
Aerospace Medical Research Laboratories

Abstract

SYSTRAN was originally developed around a data acquisition system and was primarily intended for use in analysis of periodic and transient data. It was modified to include some capability for handling random data. In use, the program was found to be cumbersome in that a tremendous number of control cards were needed to accomplish certain types of data analysis. A new system was added to allow program controlled generation of these cards. Also a few additional capabilities were included to extend the usefulness of the system. Specifically, this report describes a control card generator, an option to allow redefinition of Fourier Transforms over parts of the frequency range, an option to allow direct loading from cards of Fourier Transforms, an extension of the digital filter length, a major modification of the existing auxiliary system and other minor corrections and modifications.

Table of Contents

List of Illustrations

SECTION I.
Introduction

The previous reports in this series explain the background, intentions and uses of SYSTRAN. Therefore, this second supplement will be concerned exclusively with explaining and documenting modifications and corrections.

The continued use of SYSTRAN has revealed at least one major problem. Specifically, the number of control cards that must be generated (that is, written in the first place) is absolutely huge when repeated processing of one test or sequential processing of many tests or data channels is desired. This is primarily because SYSTRAN cannot retain the definitions of the processing path while exchanging overlay links and because closed loops in the processing path may not be self-generated until such definitions can be retained. Furthermore, it is simply not feasible to create recursive processing paths under the control card setup. To alleviate the problem somewhat, a facility has been developed in which the user can specify a prototype control card set that lists changing parameters as variables. The system allows the generation on system files of a sequence of card images that are combinations of the cards in the prototype and that contain desirable processing parameters in the variable fields.

Another problem, which has arisen from time to time, is that one desires to generate a Fourier Transform which is described by one analytic function over some part of the frequency range and by other analytic functions over other parts of the frequency range. To accomplish this, the RESET option has been added to the Fouriertran system.

It is occasionally desirable to be able to load Fourier transforms directly from the card input file. To accomplish this, the Transform Load option has been modified.

The Filter System Group was not originally set up to readjust the normal by a field on the control card. This was found to be a disadvantage in certain cases so the option was included. The original filter specifications have been modified to allow somewhat more versatility in generating an arbitrary filter. The width of the filter function and the available time span of the filter convolution have been modified. The plotted output has been modified to obtain better resolution in the generation of filter shapes. An option was added to facilitate obtaining the difference between an input function and its filtered output.

The Auxiliary System Group has been modified in structure to permanently contain the Random Number Generator System and MIMIC. It also allows for one to eight service functions similar in scope to the original Auxiliary System.

The Inverse Transform System has been modified to allow the user to save on binary tape the generated time dependent functions.

Other minor modifications are also included.

SECTION II.
The Prototype System

This capability was included by extending the control card definitions and routines included under the Utility System Group.

THE UTILITY SYSTEM MONITOR (MODIFIED)

Control is transferred to this system monitor when the *BEGIN UTILITY SYSTEM card is encountered by the job monitor. The Utility System Monitor in turn recognizes and transfers control to the system indicated by the following control cards: *MERGE, *RETRIEVE, *WRITE, *REWIND, *ENDFILE, *UNLOAD, and *PROTOTYPE. The entry to and use of the Data Storage System, the Data Retrieval System, and the Physical File Manipulation System are described in the original SYSTRAN report (AMRL-TR-65-133, page 32). (A program block diagram is included as figure 1.)

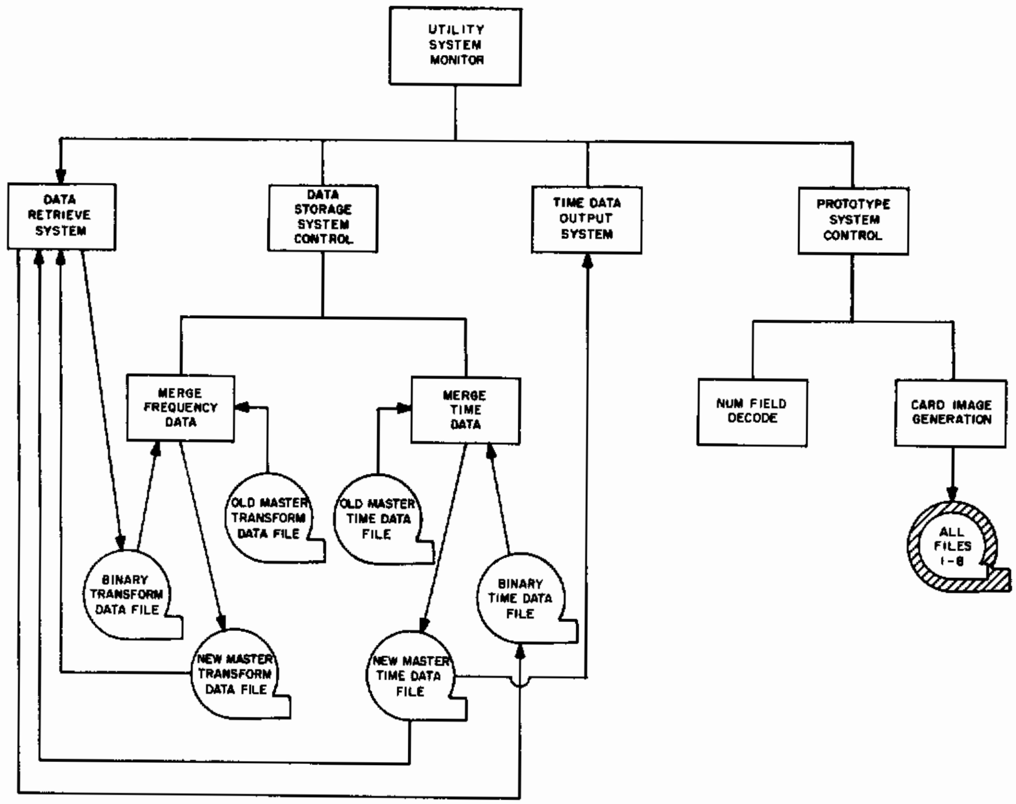

Figure 1. Utility System Monitor (Modified)

PROTOTYPE SYSTEM INPUT-OUTPUT

This system accepts as input, BCD card images (normally SYSTRAN control cards) from the normal BCD input file and produces as output, lists of BCD card images on any of the normal SYSTRAN files. The system operates by changing the SYSTRAN BCD input file designation from the normal card input file to a SYSTRAN file on which has been generated a list of appropriate

BCD card images. This list is produced by repetitions of a prototype card set under specific format defined by the prototype system control cards.

SPECIAL CONSIDERATIONS

Since the list is generated on any of the normal SYSTRAN files chosen by the user it is the responsibility of the user to be sure that this usage of the file does not interfere with the basic normal usage of that file. In other words, if one uses file 2 (Binary Time Data File) to input a certain time series and also uses file 2 to generate a card list, the second use of file 2 will overlay the first use.

This system is versatile enough so that only one type of input control card (including modifications specified herein) cannot be generated by the system. This one exception is encountered in the BCD time data edit feature. This exception is not unrealistic since the BCD time data edit system requires an absolute minimum number of control cards as it stands.

THE PROTOTYPE SYSTEM CONTROL CARDS

The implementation of this system introduced the following control cards to the SYSTRAN system.

*PROTOTYPE, OUTPUT UNIT/[M]

*PROTOINPUT, UNIT/[I]

*PROTOTYPE, SWITCH

*PROTOTYPE, CARD/[N], $ [prototype card set] $

*PROTOTYPE, VAR [J], SET [K] ([L, M]), NUM ([V_1, V_2 . . . , V_m]) [N]

RECYCLE/ ([VAR [J], SET [K]]) [I]

*PROTOTYPE, REWIND/ [N]

*PROTOTYPE, GENERATE

*PROTOTYPE, PRINT

In addition to the above user supplied control cards, the Prototype System itself internally generates the following control cards.

*SWITCH

*PROTOTYPE, RWND

*PROTOTYPE, EOFT

The individual field definitions are as follows:

*PROTOTYPE, OUTPUT UNIT/ [M]

M – The integer designation (refer to the TAPEN table) of the file which is to contain the generated control card set. As many files as are desired can be used. Care must be taken, however, so that a file is not used both as a prototype input file and as a file that passes data from one system to another.

*PROTOINPUT, UNIT/ [I]

This card is recognized by the card input routine CARD and instructs this routine to switch to the generated prototype input on file I (refer to TAPEN table, AMRL-TR-65-133, p. 56).

*PROTOTYPE, SWITCH

This card causes the prototype system to write the card image *SWITCH onto the current prototype output unit.

*SWITCH

This card is recognized by the card input routine (CARD) and causes the routine to switch back to the normal input unit for further processing.

*PROTOTYPE, CARD/ N, $ [prototype card set] $

N – The total number of physical cards that are to be generated using the given [prototype card set]. Generated comment cards are not included in this count.

[Prototype Card Set] – This set contains the cards that are to be generated for future input. The format of this is as follows:

(1) Each physical card which is to be generated must be enclosed in double quotation marks (i.e., "card image").

(2) Each physical card image must be entirely contained on one physical card. More than one such image may appear on each card, however.

(3) If a comment or label field appears in a requested card image, the following character sets are illegal in this field VAR or " (exception: if a portion of a label field is to be changed from set to set the variable parameter indication field VAR [N] can be used.).

(4) Comment cards can be generated by placing the desired comments between the card image fields defined above or before the first such image or after the last. In such comment cards the characters $ and " are illegal. It must be noted that any such comments as are generated either by this method or in label fields in card images will be void or blanks when processed by the system. Therefore, if spacing is desired a spacing character must be used.

(5) A maximum of 55 card images (including comment cards) may be used.

(6) Any card image (enclosed in double quotes) with the exception of the *PROTOTYPE, CARD – and the *PROTOTYPE, VAR – control cards can be used. Thus the system can be used to generate BCD input for the LOAD option of the Fouriertran system.

(7) A single card image can be *repeated in place* in the following manner:

"(card image) N"

Thus in the generated set the *card image* would appear N times in succession. Such a repetition counts N times in the generated card count (N in the CARD/[N] field) but only once in the count of the prototype card set (item 5 above). In like manner a series of card images may also be repeated. "(Card image$_1$" "Card image$_2$" "Card image$_3$"... "Card image$_m$)N". In this case M*N cards are generated in the generated card count while M+ comments are introduced into the prototype card set. Due to this feature no card image contained in a repeated set may appear that terminates with the characters)X" where X is a one to 6 character integer. If such a character set is necessary for terminating a card, it must be written in the form)X", which for the purposes of the SYSTRAN processing features is identical. Due also to this feature no card image may begin with "(. This is not an unrealistic restriction, however, since there are no control cards or continuations of such that may legitimately begin with the character (.

(8) The parameters that are to be given values during the generation of the prototype input file are designated in the card images by the characters VAR [J] where J is the assigned variable number.

*PROTOTYPE, VAR [J], SET [K] ([L, M]), NUM ([V_1, $V_2, \ldots,$ V_m]) [N], RECYCLE/ (VAR [J], SET [K]) [I].

This card contains the specifications for the values of the variable parameters contained in the prototype card set. The fields VAR, SET, and NUM constitute a single definition and as such must appear as a unit in the order given above. The RECYCLE field is completely self-contained and as such may appear between any of the above described *units* in any number or order.

VAR [J]

J – The variable number corresponding to the variable in the prototype card set to which the corresponding values in the NUM field apply.

SET [K] ([L, M])

K – The set number for the corresponding VAR and NUM fields.

L – The repeat in place value for each individual number in the associated NUM field.

M – The total number of V fields in the associated NUM field.

NUM ([V_1, V_2, $V_3, \ldots,$ V_m]) [N]

V_i – The value definition field

V_i may be written in any of the following forms:

1) $v_1, v_2, v_3, \ldots, v_k,$

2) $(v_1, v_2, v_3, \ldots, v_k)$ n

3) $(v_1, v_2, (k))$

The first of these forms $v_1, v_2, \ldots, v_k$ is a listed subset. Using such a subset causes the following action (assume V_1 is used). The first L values used for the associated variable VAR [J] with the associated set SET [K] ([L, M]) would be v_1, the next L values v_2, etc., through v_k. The process would then move to V_2. The second form [$(v_1, v_2, v_3, \ldots, v_k)$ n] is identical in use to the first set except that instead of moving directly to V_2 from V_1, V_1 is itself repeated n times. The third form is the implied DO form. The values actually used would be

$\bar{v}_1, \bar{v}_2, \ldots \bar{v}_k$ where

$$\bar{v}_1 = v_1$$

$$\bar{v}_2 = \bar{v}_1 + v_2$$

.

.

.

$$\bar{v}_k = \bar{v}_{k-1} + v_2.$$

These values are used in the same manner as the v_j in forms 1 and 2 above. The v_j in forms 1 and 2 can be of any type, integer numeric, floating point numeric, or alpha-numeric. Integer numeric are restricted to six characters, floating point numeric to 12 characters, and alpha-numeric to 36 characters. If any v_j contains a left parenthesis, right parenthesis or a comma, the entire v_j must be enclosed by double quotes. The total number of characters allowable in any one NUM field is 102. Each v_i uses $6 + \gamma$ characters where

$$\gamma = 6 \sum_{i=1}^{k} \gamma_i$$

and

γ_i = the lowest integer multiple of 6 such that $6 * \gamma_i$ is greater than or equal to the number of characters in v_i.

N – The number of times the entire NUM field is to be repeated when the SET is used in generation. If N is not given, it is assumed to be 1.

It is often desirable to include a card in a repeated prototype card set or subset that is executed at some times and not executed at others. It is possible to accomplish this in the following manner. Precede the desired card by a variable (VAR). Assign to this variable the value ,, when the card is to be skipped in the execution and a null value when the card is to be executed. The value ,, is obtained by using ",," as a V_i value in the NUM field of the variable definition. A null value is obtained by using "" as the V_i value.

Each variable VAR [J] may have five different sets associated with it. The sets for each variable can be repeated as often as is necessary as long as the total number of set repetitions for any one variable is less than or equal to 100. Upon completion of the user designated set repetitions, the program automatically cycles to the beginning of the user designated set.

RECYCLE/([VAR[J], SET[K]]) [M]

This field allows the user to set up SET repetitions for any VAR, SET combination.

VAR [J], SET [K] – The VAR, SET combination that is to be repeated.

M – The number of repetitions. (Assumed to be 1 if not given.)

The order of the use of the defined sets for each VAR is the order in which they occur on the card.

*PROTOTYPE, REWIND/[N]

This card causes the system to write the two following card images on the prototype output unit [N]

*SWITCH

*PROTOTYPE, RWND

*PROTOTYPE, GENERATE

This card causes the system to generate the current prototype set on the current prototype output unit.

*PROTOTYPE, RWND

When this card is encountered by the card input routine, the currently defined prototype input unit is rewound before processing continues.

*PROTOTYPE, EOFT

This card image is written on *all* prototype output units used in any one entry to the prototype system. It is used by the card input routine to signal end of tape condition. If it is actually read, there is an error. If this error occurs, the input condition is automatically set back to the standard input unit and the remaining portion of the job is bypassed.

*PROTOTYPE, PRINT

This card causes the program to print the card images as they are generated.

SYSTEM OPERATION

The system automatically generates the currently defined prototype set if any of the below conditions are satisfied.

(1) A *PROTOTYPE, OUTPUT UNIT/[N] card is encountered.
(Exception: When this is the first such card encountered.)

(2) A *PROTOTYPE, REWIND/[N] card is encountered.

(3) A *PROTOTYPE, SWITCH card is encountered.

(4) A *PROTOTYPE, GENERATE card is encountered.

At the conclusion of generation both the prototype card set and the VAR, SET, NUM definitions are destroyed.

It is the user's responsibility to insure that:

(1) The proper control of the input unit is maintained. Thus, if control is to be switched back to the primary input unit within a generated prototype card set the user must supply the necessary *SWITCH card.

(2) The physical card images used in the prototype card set meet system requirements.

(3) The units used by the prototype system are not at the same time used for another purpose.

(4) The units that are to be used by the system are in the proper status. Thus, the

*REWIND feature of the Utility system must be used before a unit is generated.
To insure proper designation of the prototype output file to avoid double utilization errors, the following list of system utilization of files is presented.

System or System Monitor	*Files Used*
Time System Group	
Data Check System	2
Data Edit System	2, 3
Binary Tape	2, 3
Card	2

Transform Generator	1, 2
Regression System	2, 5
UTILITY System Group	
Rewind Option	As designated
Unload Option	As designated
End File Option	As designated
Merge System	
Time Data	2, 6, 7
Frequency Data	1, 5, 8
Time Data Output System	7
Prototype System	As designated
Correlation System	1, 2, 4, 5
RELOAD Option	1, 4, 5
CRLOAD Option	1, 2, 5
Frequency System Group	
Matrix System	4, 5
Inverse Transform System	2, 4
Fouriertran System	1, 4
LOAD Option	1
SAVE Option	4
Auxiliary System	
User Written	?
Random Number Generator	7
Filter System	2, 6
Probability Density System	2, 5, 7

SECTION III.
Modifications

THE FOURIERTRAN SYSTEM

The RESET Option

This option allows the user to generate a Fouriertran function composed of parts of other Fouriertran functions or complex constants. This option introduces the following control card into the SYSTRAN control card set.

*RESET/F [N], WITH/F [K], FOR W = [I_1 I_2], USE W = [J_1, J_2], WITH/etc. This card instructs the Fouriertran system to place into the Fouriertran function F[N] for frequencies W_{I1}, through W_{I2} the values of the Fouriertran function F[K] for frequencies W_{J1}, through W_{J2}. The value of J_2 is not actually used in the program. Its value regardless of what is placed on the card is $J_2 = J_1 + I_2 - I_1$. If the USEW field is omitted, $J_1 = I_1$, $J_2 = I_2$ is assumed. The WITH field may take on the alternate form WITH/(r, i) in which case the complex constant (r, i) is used in place of the values of the Fouriertran function F[K]. In such cases the USE field must not be present.

As many WITH, FORW, USEW, field sets as are desired may be used. Any portion of F[N] that is not modified by a set of these fields retains the value that it previously had. Since no internal check is made as to the range of the frequency subscripts (i.e., I_1, I_2, and J_1, J_2), care must be exercised by the user to insure that the particular F[N] with which he is working is not overloaded.

The Fouriertran Function Card Load Option

The Fouriertran function card load option was added to the Fouriertran system to allow the user to load Fouriertran functions directly from cards. This option introduces the following control cards into the SYSTRAN card set.

*LOAD, TYPE/BCD, NOF/[N], MODE/$\begin{bmatrix} \text{MP} \\ \text{RI} \end{bmatrix}$

*FORMAT$[(format)]$

*FUNCTION/[M]

These cards instruct the system in the following manner:

*LOAD, TYPE/BCD, NOF/[N], MODE/$\begin{bmatrix} \text{MP} \\ \text{RI} \end{bmatrix}$

TYPE/BCD

This field instructs the LOAD option that this is to be a direct card load.

NOF/[N]

N is the number of function values on a card.

MODE/$\begin{bmatrix} \text{MP} \\ \text{RI} \end{bmatrix}$

MP — The function values are given as magnitude and phase angle.

RI — The function values are given as Real and Imaginary parts.

*FORMAT$[(format)]$

(format) – The FORTRAN format under which the cards are to be read.

*FUNCTION/[M]

M – The Fouriertran function number into which the load is to be effected (i.e., F[M]).

The direct card load option and the normal load option can both be used in the same entry into the Fouriertran System. If only the direct card load option is used, the *TEST control card must be used to set the test number for output purposes. The *FORMAT card can be used to change the input format between any two direct loads. However if this is desired, this card *must* precede the *FUNCTION control card for that load.

THE FILTER SYSTEM GROUP

The Norms Field

The *FILTER control card of the Filter System has been modified to allow the use of a normalization factor on the input data. The following field has been added.

NORMS/[(N_1, N_2, . . . , N_m)]

This field is somewhat different than the other fields on the card in that the subfields bear a direct one to one relationship to input channels 1 through M (i.e., N_1 is the norm for channel 1, N_2 for channel 2, etc.). If this field is not present the first data point from each channel is used as the norm as was the case before the introduction of this field.

The Filter Options

The *COPY control card has been modified to include the following:

If *COPY/N is used only, the first N channels of the input data are copied and the number of input buffers is increased appropriately. This enables the user to compress his data and thereby obtain a wider time span on the filter convolution. At the same time the filter width has been increased to a maximum of 199 points. In addition to increasing the width of the filter function to 199 points, the user also has the ability to specify the time interval between the computed filter function points in terms of the specified input time increment value. The latter is accomplished by setting the value of N, in the WIDTH/N field of the *OPTION card, to a value less than zero. The action taken in this case is the following:

1. The width of the filter function becomes $|N|$ *199
2. The filter function is computed at every $|N|$
3. The intermediate points of the filter function are obtained by linear interpolation.

The original plotted output method called for using the smallest minimum and largest maximum obtained by the input channel or one of the associated output channels on each buffer. This has been modified to allow the exclusion of the input channel or one of the associated output channels on each buffer. This enables the user to produce the filter shape with maximum resolution obtainable. This is accomplished by setting the appropriate output key to a negative value.

The OPTION field of the *FILTER card has been modified to allow the user to obtain an output channel containing an unfiltered input data channel minus a filtered output data channel. This is illustrated in the following example.

*FILTER, TEST/[(1, 2)], CHIN[(1, 1)], CHOUT/[(3, 4)]

OPTION/[(1,–3)], . . .

The following action takes place:

1. Input channel 1 is filtered and placed on Output channel 3 using filter option 1.

2. Output Channel 3 is subtracted from input channel 1 and the result is placed on Output channel 4.

Step 1 of the action is indicated by the 1 in the OPTION field. Step 2 is accomplished by use of the –3 in the option field.

Restriction: The indicated generation of the necessary output channel (in the above case, channel 3) must precede the subtraction action indicator.

This is accomplished by setting the output channel of the filtered data, which is to be subtracted, before the output channel which will contain the result, in the CHOUT field. The CHIN and OPTION fields must of course correspondingly be compatible. Increasing the filter width to 199 points made it necessary, for reasons of storage, to reduce the number of possible filter functions, on any one pass, to three. Thus at most three unique filter options may be defined.

THE LOAD DECK

Due to the size of SYSTRAN it is impossible to load the entire system in one computer run. Thus it was necessary to include in the SYSTRAN deck various dummy routines than can replace system groups or systems and thus allow the loading of the program. A complete list of these routines and the instructions for their utilization is included in the source program listing that may be obtained on request.

THE JOB MONITOR

The binary file buffer size has been modified to 201 words. This change affects all of the files listed in the TAPEN table.

The card input routine has been modified to recognize the control cards.

*PROTOINPUT, UNIT/[I]

*SWITCH

*PROTOTYPE, RWND

*PROTOTYPE, EOFT

The use of these cards is explained in the PROTOTYPE system section contained in this report.

THE AUXILIARY SYSTEM GROUP

The Auxiliary System Group has been extensively modified in format and use to enable SYSTRAN to load the Random Number Generator System, the MIMIC (SESCA Internal Memo 65-12, A Digital Simulator Program) System, and one to eight user supplied programs on the same computer run. This has been accomplished by providing the Auxiliary System Monitor with ten calls, one to the Random Number System, one to the MIMIC System and one to each of the routines SERV02, SERV03, . . . , SERV10. A program block diagram is included as figure 2. The user supplied routines now have the above eight names instead of the previous one name

AUXIL. The Auxiliary System Monitor is entered as before upon encountering a "BEGIN AUXILIARY SYSTEM control card. This system monitor in turn recognizes the following control cards:

*RANDOM, TEST/etc.

*MIMIC

*SER3

*SER4

.

.

.

*SER9

*INCREMENT, etc.

*SER0

Upon encountering one of these cards, control is transferred to the appropriately named system for execution. The control cards for the Random Number Generator System are those defined in supplement I of AMRL-TR-65-133.

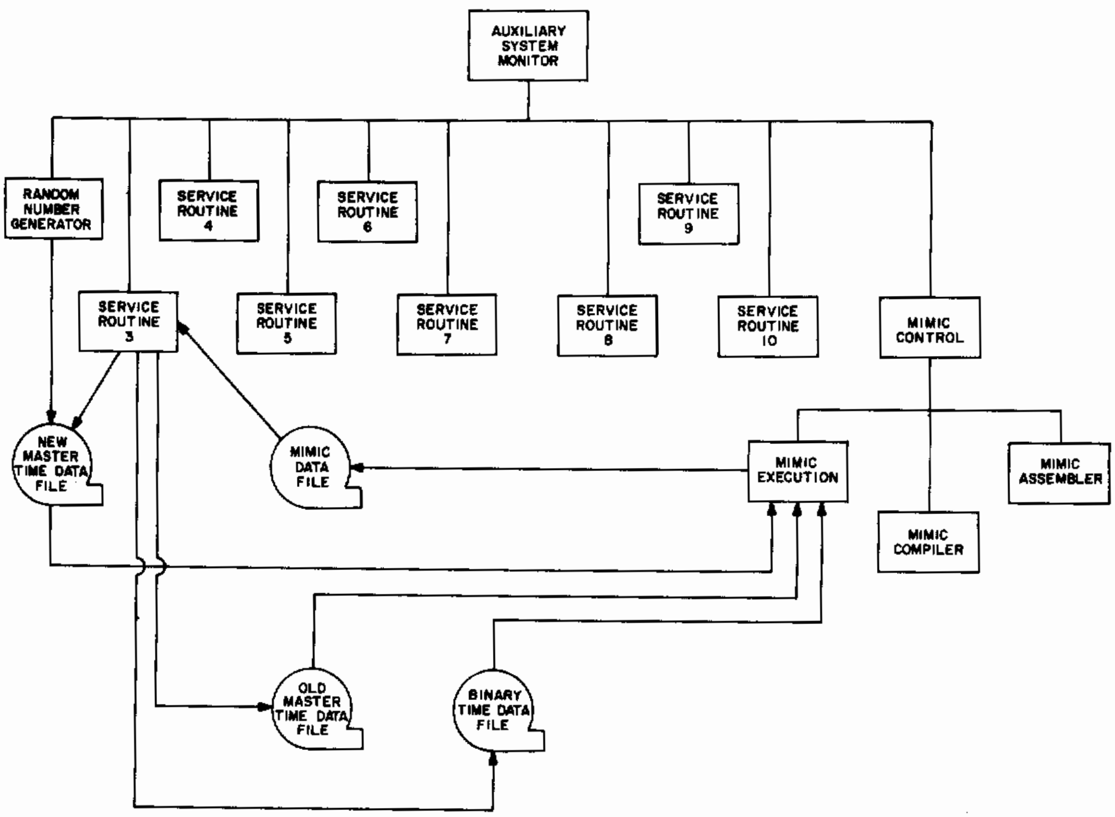

Figure 2. Auxiliary System Monitor (Modified)

MIMIC

The *MIMIC control card is the only SYSTRAN defined control card for this system. All other inputs to this system are as defined for MIMIC. The MIMIC service routines SR1 and SR2 are prewritten so as to allow the following options:

SR1 (FV, C, AKEY, TEST, FILE, RN)—This routine generates one of the Time Data files.

The routine input is the following:

FV – The value of the function (the MIMIC name)

C – The channel for the output

AKEY – Action key

= 1 Initialize file

= 2 Increment time

= 3 Insert f(t) value

= 4 Terminate Test

TEST – Output Test Number

FILE – File indicator

= 1 Binary Time Data File Used

= 2 Old Master Time Data File Used

= 3 New Master Time Data File Used

RN – Parameter to force Sort

SR2 (T, C, AKEY, FILE, TEST, STEP, FN) – This routine furnishes the integration program with a value of a forcing function. The forcing function values are on one of the Time Data files. The Routine input is the following:

T – Time of entry

C – Input channel desired

AKEY – Action key

= 1 Initialize File

= 2 Initialize time Increment

= 3 Obtain value

FILE – Input File indicator same as for SR1

TEST – Input Test Number

STEP – Ratio of dt of input
tape to dt of solution

FN – Returned value

Restriction: STEP must be evenly divisible into 200. The format of calls to these routines is as defined in MIMIC (i.e., SR1 (A, B, C, D, E, F, R))

Routine Actions:

1. If time of solution exceeds total time contained on forcing function, tape will cycle.

2. If time of entry does not correspond to an exact time on the input tape, linear interpolation will occur.

Special Entry SR2:

When AKEY = 1

C = dt of solution rather than the channel subscript

Special Entry SR1:

When AKEY = 1

C = 1 if this is the last pass through the system

C = 0 if more passes will follow

FILE = Maximum file subscript to be used on this pass

The MIMIC System is too large to allow the generation of a SYSTRAN time data file directly due to the size of the necessary buffer. Therefore the routine SERV03 has been written that takes as input the file generated by MIMIC and converts this file to the desired SYSTRAN Time data file(s). To accomplish this, however, the user must provide the entry to the routine by use of the control card *SER3. This card should immediately follow the MIMIC data cards.

The routine SERV03 that is used to generate the SYSTRAN Time data files from MIMIC data recognizes the following control card.

$$\text{*NORM,}\begin{bmatrix}\text{FILE/ } N(V_1, V_2, \ldots, V_m) \\ \text{INITIAL CONDITIONS} \\ \text{NULL}\end{bmatrix}, \ldots$$

The following action takes place

FILE/[$N(V_1, V_2, V_3, \ldots, V_m)$], . . .

N – The file reference number

(same correspondence as in SR1)

V_i – The values for the norms on the channels

V_i is composed as follows:

V(i) where V is the value and i is the channel number.

Using this field on generated file N, the channels for which a V_i is present assume a normal value of v(i). All other channels assume a zero normal value (i.e., $V_i = 0$). If all channels for a given file are to assume a zero value, deletion of the entire FILE field or using the field as FILE/N( ) will accomplish this. If all generated files are to be set up with all zero norms, the third condition (null) given above will accomplish this. If the initial conditions are to be used as the norms for all files, the alternate form of the card is used (i.e., *NORMS, INITIAL CONDITIONS). If a *NORMS card is to be used with any test on a single pass through SERV03, then a *NORMS card must be present for each test generated on that pass (the *NORMS cards must be in order).

SERVICE ROUTINE SERV09

It has been repeatedly requested that SYSTRAN be modified to accept and operate on data that is recorded at unevenly spaced values of the independent parameter. Since such a modification would require a design change in all computational routines in the system it was decided to provide a means of interpolating such data at evenly spaced intervals. This capability has been included as service routine SERV09 in the Auxiliary System. The control cards required for the utilization of this capability are the following:

*BEGIN AUXILIARY SYSTEM

(enter the Auxiliary System)

*SER9

(enter the Service Routine)

*INCREMENT, TEST([I, J]), CHAN([K1, K2, . . . , KL]) TIME ([M]), OPTION([N])

(Service Routine control parameters)

The definition of the control parameters are as follows:

I – Input Test Number

J – Output Test Number

K1, K2, . . . , KL – Channels containing data to be interpolated

M – Channel containing the values of the independent parameter

N – Interpolation Option

N=1 Linear Interpolation

N=2 2nd degree interpolation

The input data is taken from the Binary Time Data File. The output is generated on the New Master Time Data File. Input channel K1 is generated onto channel I of the output file. The interpolation uses the input points as described below:

For Linear Interpolation

To compute $X_o(t_l)$

Use $X_I(t_k)$, $X_I(t_j)$

Such that

$$t_k < t_l \leq t_j$$

where:

$$X_o(t_l) = \frac{[X_I(t_j) - X_I(t_k)]*[t_l - t_k]}{[t_j - t_k]} + X_I(t_k)$$

For 2nd Degree Interpolation

To compute $X_o(t_l)$

Use $X_I(t_k)$, $X_I(t_j)$, $X_I(t_m)$

Such that

$$t_k < t_l \leq t_j$$

and

$$m = k-1 \text{ if } t_l - t_{k-1} \leq t_{j+1} - t_l$$

or

$$m = j+1 \text{ if } t_l - t_{k-1} > t_{j+1} - t_l$$

where

$$X_o(t_l) = a_1 t_l^2 + a_2 t_l + a_3$$

such that for a_1, a_2, and a_3

$$X_I(t_m) = a_1 t_m^2 + a_2 t_m + a_3$$

$$X_I(t_k) = a_1 t_k^2 + a_2 t_k + a_3$$

$$X_I(t_j) = a_1 t_j^2 + a_2 t_j + a_3$$

The values of t_l are

$$t_{lo} = 0$$

$$t_{l1} = \Delta t$$

$$t_{l2} = t_{l1} + \Delta t$$

.

.

.

$$t_{ln} = t_{ln-1} + \Delta t$$

The last value of t_l equals the last value of t on the input channel.

Δt is the basic time increment set by the *DELTAT control card in the Time System.

THE INVERSE TRANSFORM SYSTEM

The Inverse Transform System originally produced only printed and plotted output of the generated time function. This has been modified to allow the user to obtain a binary file, which contains this data. This binary file is produced on the New Master Time Data File in the format of said file to allow its use as input time data on the same computer run. It must be noted, however, that on other than a Probability System run the generated data must first be retrieved, by use of the Utility System, onto the Binary Time Data File. To accomplish the generation of the file from the Inverse Transform system, one needs only to set the value of K on the *INVERSE/K control card less than zero. Under this modification the system always uses $|K|$ as the key for input from the Save file.

SECTION IV.
Corrections

1. The outputing of the correlation functions onto the Binary Transform Data File for use in the Frequency System has been corrected so that only those that have an associated nonzero output field are written. This was not the case previously.

2. The diagram of the Probability System contained in Sup. I is incorrect. A correct diagram is included herein as figure 3.

3. The unnumbered diagram at top of page 40 in AMRL-TR-65-133, SYSTRAN (Systems Analysis Translator): A Digital Computer Program, is incorrect and a corrected version is included as figure 4.

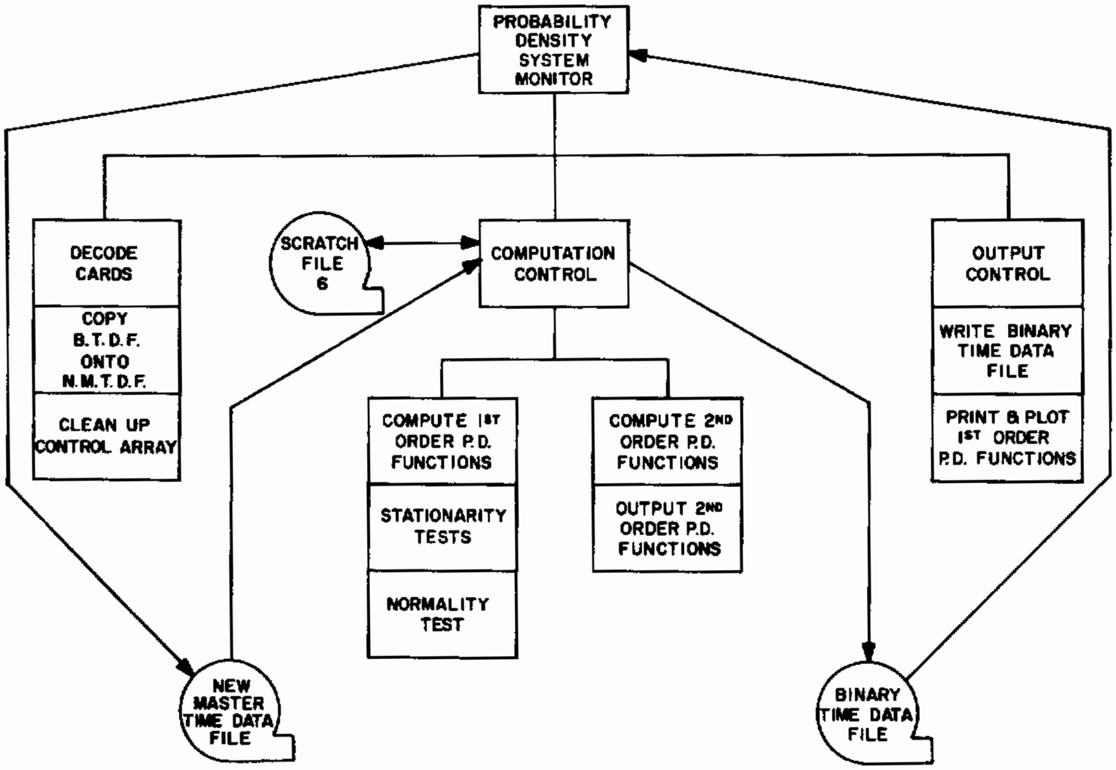

Figure 3. Probability Density Monitor (Corrected)

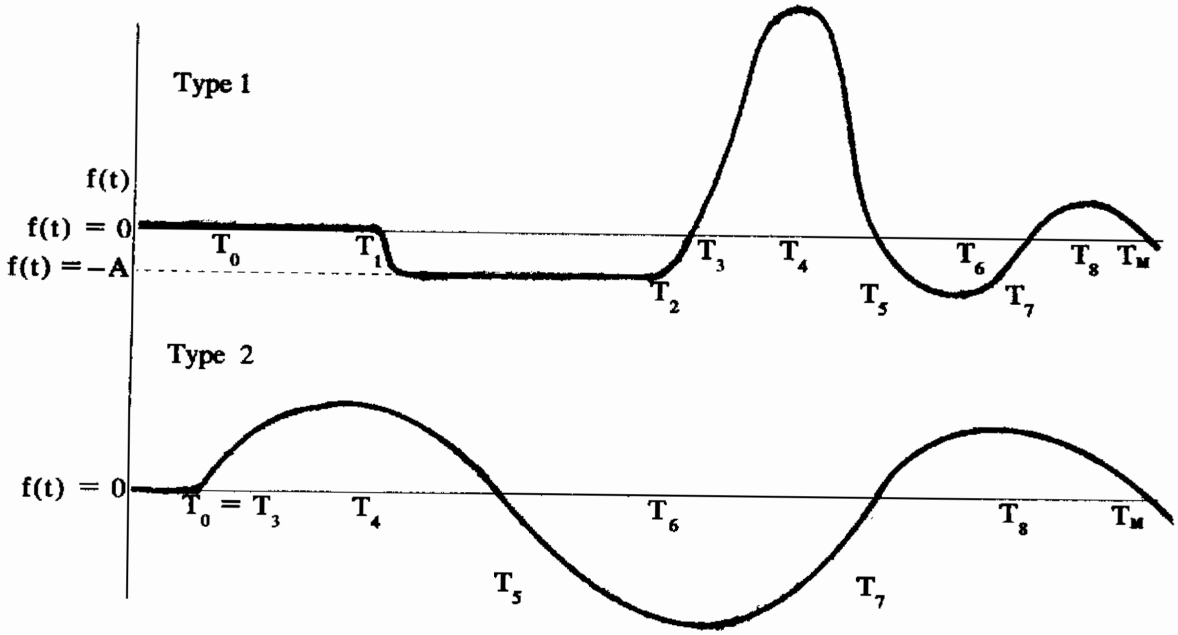

Figure 4. Regression System Function Types

References

1. Lajeunesse, D. J., E. B. Weis, Jr., and T. J. Hogan, Jr., *SYSTRAN (Systems Analysis Translator): A Digital Computer Program,* AMRL-TR-65-133, AD 624 468, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, July 1965.

2. Weis, E. B., Jr., and D. J. Lajeunesse, *SYSTRAN (Systems Analysis Translator): A Digital Computer Program,* Supplement One, AMRL-TR-65-133 (SUP. I), AD 638 168, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, March 1966.

3. Sanson, F. J., and H. E. Petersen, *MIMIC Programming Manual,* SEG-TR-67-31, Systems Engineering Group, Wright-Patterson Air Force Base, Ohio, July 1967.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Aerospace Medical Research Laboratories Aerospace Medical Div., Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

SYSTRAN (SYSTEMS ANALYSIS TRANSLATOR): A DIGITAL COMPUTER PROGRAM SUPPLEMENT TWO

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report, August 1965 - October 1966

5. AUTHOR(S) *(First name, middle initial, last name)*

E. B. Weis, Jr., MD
D. J. Lajeunesse*

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1967	19	3

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 7231	
c. Task No. 723101	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	AMRL-TR-65-133 (SUP. II)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
*Digital Computation Division Research and Technology Division Wright-Patterson AFB, OH 45433	Aerospace Medical Research Laboratories Aerospace Medical Div., Air Force Systems Command, Wright-Patterson AFB, OH 45433

13. ABSTRACT

SYSTRAN was originally developed around a data acquisition system and was primarily intended for use in analysis of periodic and transient data. It was modified to include some capability for handling random data. In use, the program was found to be cumbersome in that a tremendous number of control cards were needed to accomplish certain types of data analysis. A new system was added to allow program controlled generation of these cards. Also a few additional capabilities were included to extend the usefulness of the system. Specifically, this report describes a control card generator, an option to allow redefinition of Fourier Transforms over parts of the frequency range, an option to allow direct loading from cards of Fourier Transforms, an extension of the digital filter length, a major modification of the existing auxiliary system and other minor corrections and modifications.

DD FORM 1 NOV 65 1473

Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Digital Computer Program Frequency Analysis Correlation Analysis Systems Analysis Translator						

Security Classification