TRW SYSTEMS

ONE SPACE PARK • REDONDO BEACH, CALIFORNIA

8976-6027-RU-000

TRACKING ACCURACY MATRIX PROCESSOR (TAPP MOD III)

by W. W. Lemmon

11 December 1963

TAPP 3

Prepared for

JET PROPULSION LABORATORY
California Institute of Technology
Contract No. 950045

Approved [signature] for
P. A. Penzo
Project Engineer

Approved [signature] for
R. M. Page, Manager
Systems Analysis Department

TRW/SPACE TECHNOLOGY LABORATORIES
One Space Park
Redondo Beach, California

CONTENTS

1. INTRODUCTION

The Tracking Accuracy Matrix Processor Program is the third program in the Tracking Accuracy Prediction Program series* and is therefore denoted as Mod III or TAPP III. The purpose of the Tracking Accuracy Matrix Processor (TAPP III) is to extend the range of applicability and the versatility of the TAPP-series and to eliminate the duplication of TAPP Mod I runs for cases involving the same trajectory and tracking noise models. To accomplish this, a simplified version of TAPP I is used to produce a tape, T_{13}, which drives TAPP III. Depending upon manual inputs, TAPP III produces either a statistical printout, 0_3, or a TAPP II Tape, T_{32}, or both. It is possible, therefore, to use this system to replace two previous versions of TAPP I.

In the T_{13} tape input to TAPP III, as generated by TAPP I, the trajectory, tracking stations, tracking intervals, and noise models are all specified. In operating TAPP III, one has the option of selecting or deleting the various stations or tracking intervals and thereby studying different tracking patterns without rerunning TAPP I. One of the principal features of TAPP III is that the assumptions used to fit the data to determine orbits, the so called "fit-world", may be entirely different from the "true" or "real-world". For example, in a trajectory determination one might neglect the uncertainty in the GM of the earth, μ_e, which is the same as setting $\sigma_{\mu_e} = 0$, a priori. Using TAPP III, one can permit this assumption to be made in the fit world; make $\sigma_{\mu_e} > 0$ in the real world, and then determine the deleterious effect on the orbit determination. This is a very simple, almost trivial, example of the types of problems which can be analyzed. A more sophisticated problem is to determine the deleterious effects of making erroneous assumptions in the fit world regarding the relationships between parameters.

To summarize, the new TAPP I, II, III system has two major advantages over previous versions of TAPP I:

---

*Documented in References 1, 2 and 5.

a) The desired TAPP III outputs are specified by coded inputs. This is not accomplished by option flags but by a matrix language which permits one to choose a virtually arbitrary number of output entities. Each entity is determined by a set of symbolic indices which permit a choice of approximately 50,000 substantially different types of output entities.

b) The assumptions used to fit the data (the fit-world) may be completely different from the true statistical model (the real-world). These assumptions include possible constraints among the constant parameters.

The printout of TAPP III, 0_3, is designed to answer complex statistical questions regarding erroneous assumptions in the fit world, as well as to present the usual tracking simulation accuracy results. In addition, by making several TAPP III runs using the same T_{13} tape, one can obtain results based on different assumptions without rerunning TAPP I.

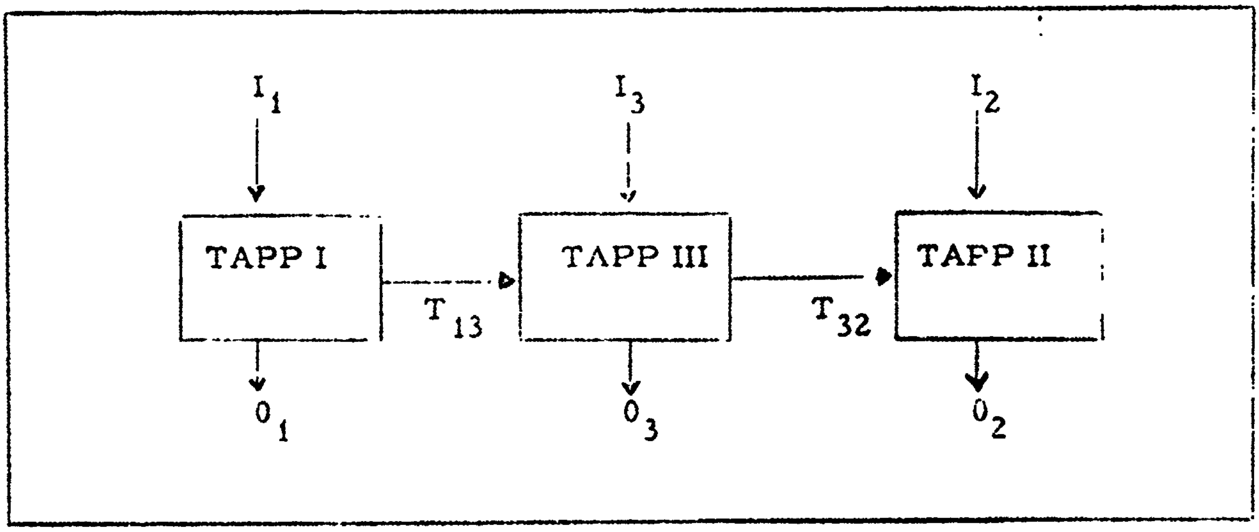


Figure 1-1. General Information Flow for the TAPP Program

The second function of TAPP III is to act as a buffer program between TAPP I and TAPP II, see Figure 1-1. Thus, a set of inputs (I_1) to TAPP I, consisting of output times and trajectory initial conditions together with fit- and real-world tracking noise models, is used by TAPP I to produce a tape (T_{13}) of partials and normal matrices going with the specified time intervals. TAPP III accepts T_{13} and produces a tape (T_{32}) for use by TAPP II. Either simultaneously or separately, TAPP III can produce a statistical printout (0_3). The manual inputs (I_3) to TAPP III include the following information:

a) Print output times
b) Print output format
c) Tape output times
d) Tape output format
e) Flags stating whether or not a station is operating
f) Constraints assumed for the purposes of fitting the tracking data
g) "True" constraints
h) A priori normal matrix used for the fit
i) "True" a priori covariance matrix

Thus a single T_{13} tape may be processed in a large number of different ways.

An important feature of TAPP III is that it extends the applicability of TAPP II to cases in which the constraints satisfied by the constant parameters are different in the fit and real worlds. This is accomplished by a built-in tape output format which always provides TAPP II with partials and normal matrices under valid TAPP II assumptions. Thus the fit-world constraints are always subtracted out of the "consider" parameters.

TAPP III may also be driven by tapes produced by tracking programs other than TAPP I. For example, a tracking program using exact integration techniques could be modified to produce a "T_{13}" tape of appropriate format. The modifications to the tracking program would be relatively minor since TAPP III imposes no requirements on the nature of the physical constants and biases under regression.

2. REGRESSION TO A CONSTRAINED PARAMETER SET

One of the most important subroutines of TAPP III is to accept the normal matrices "$A^T WA$" and "$A^T WMWA$" accumulated from the TAPP I output tape, together with the assumed and true a priori statistics, to produce the assumed and true a posteriori statistics. A method is presented for accomplishing this for the most general type of linear minimum-variance regression--namely, one in which the assumptions made, for the purpose of the regression, about the a priori statistics of the parameters, including the nature of their constraints, are completely different from the "true" a priori statistics.

This philosophy of treating constraints as part of the a priori statistics of the parameters being fitted not only permits a generalization of the purpose of TAPP III but considerably simplifies its programming. Furthermore, the a priori and a posteriori statistics are stored in the TAPP I representation thus greatly reducing the updating problems. Both the input (a priori) and output (a posteriori) statistics are stored in the same format (a normal matrix for the fit world and a covariance matrix for the real world), which permits a relatively simple sequencing and further simplifies updating.

2.1 NOTATION FOR A REGRESSION

Let z^r represent a random vector with a sample vector space V^r of dimension n_r* which is being used as a set of measurements for regressing to a random vector x^o which had a previously estimated value

---

*In this section all column vector spaces are designated "V^a" where "a" is a letter or number. The dimension of the vector space is designated by "n_a" and sample member of V^a possess a superscript "a". Matrices taking members of V^a into V^β are designated "M^β_a", where "M" may be replaced by any capital letter. This notation is a specialization of the covariant--contravariant notation of tensor analysis to matrix algebra. Understanding its motivation is not essential to the development.

μx^o. The random vector x^o may be interpreted as the parameter set of TAPP I. The error in x^o previous to the regression is given by

$$\delta x^o = x^o - \mu x^o \tag{2-1}$$

Let μz^r be the value that z^r would have for $x^o = \mu x^o$ if no measurement errors were present. The residual of the measurement, defined by

$$\delta z^r = z^r - \mu z^r \tag{2-2}$$

is assumed to be related to x^o by the regression equation

$$\delta z^r = A^r_o \delta x^o + \delta m^r \tag{2-3}$$

where A^r_o is a $n_r x n_o$ matrix and where δm^r is the measurement error. After the regression, a new estimate $\mu' x^o$ is formed by the relation

$$\mu' x^o = \mu x^o + \mu' \delta x^o \tag{2-4}$$

where $\mu' \delta x^o$ is the new estimate of the error in x^o (obviously the old estimate of the error is zero). The new error in x^o is given by

$$\delta' x^o = x^o - \mu' x^o = \delta x^o - \mu' \delta x^o \tag{2-5}$$

2.2 THE FIT WORLD

In this section, $\mu' \delta x^o$ is a linear, minimum-variance estimator of δx^o based on a set of second-moment statistical assumptions about x^o and δm^r. These statistical assumptions are called the fit world (FW), and are described below.

In the fit world, δx^o has zero mean and x^o is constrained with probability one to a submanifold of V^o. Thus the statistics of x^o must be specified for a parametization of this submanifold defined by the n_2 independent constraints

$$f^2(x^o) = c^2 (\text{FWP} = 1) \tag{2-6}$$

where c^2 is a nonstatistical (known) n_2-dimensional column vector, and where "FWP" means "fit world probability". To find a parametric representation of the submanifold defined by the constraints (2-6), a $n_1 \times n_o$ matrix C_o^1 (where $n_1 = n_o - n_z$) is chosen such that the n_o equations

$$x^1 = D_o^1 x^o \tag{2-7a}$$

$$x^2 = f^2(x^o) \tag{2-7b}$$

may be solved uniquely for x^o in a sufficiently large region about μx^o. Designating this solution

$$x^o = e^o(x^1, x^2) \tag{2-8}$$

the parametric representation of (2-6) takes the form

$$x^o = e^o(x^1, c^2) \qquad (\text{FWP} = 1) \tag{2-9}$$

Thus according to the fit world, the n_o-dimensional vector x^o may be expressed as a function of the n_1 dimensional vection x^1. In other words, x^1 is the parameter set which is being "fitted on" and hence the components of x^1 are the <u>fit parameters</u> of the previous terminology. The n_2-dimensional vector x^2 is assumed in the fit world to have a perfectly known value. Defining

$$\mu x^1 = D_o^1 \mu x^o, \ \mu x^2 = f^2(\mu x^o) \tag{2-10}$$

$$\delta x^1 = x^1 - \mu x^1, \ \delta x^2 = x^2 - \mu x^2$$

It follows from taking variations of (2-7) and (2-8) about μx^o that for small values of δx^o

$$\delta x^1 = D_o^1 \delta x^o \tag{2-11a}$$

$$\delta x^2 = D_o^2 \delta x^o \tag{2-11b}$$

$$\delta x^o = E_1^o \delta x^1 + E_2^o \delta x^2 \tag{2-12}$$

where

$$D_o^2 = \frac{\partial f^2}{\partial x^o} (\mu x^o) \tag{2-13}$$

$$E_a^o = \frac{\partial c^o}{\partial x^a} (\mu x^1, \mu x^2) \quad a = 1, 2 \tag{2-14}$$

The matrices in (2-11) and (2-12) are called the FW transformation matrices. They are seen to satisfy the identities

$$D_o^1 E_1^o = I_1^1, \qquad D_o^1 E_2^o = 0_2^1 \tag{2-15a, b}$$

$$D_o^2 E_1^o = 0_1^2, \qquad D_o^2 E_2^o = I_2^2 \tag{2-15c, d}$$

$$E_1^o D_o^1 + E_2^o D_o^2 = I_o^o \tag{2-15e}$$

where I_a^a represents the $n_a \times n_a$ identity matrix, and where 0_β^a represents the $n_a \times n_\beta$ zero matrix.

Since δx^o has zero mean, so have δx^1 and δx^2. Because of (2-6),

$$\delta x^2 = D_o^2 \delta x^o = 0 \qquad (\text{FWP} = 1) \tag{2-16}$$

hence from (2-12)

$$\delta x^o = E_1^o \delta x^1 \qquad (\text{FWP} = 1) \tag{2-17}$$

The subspace defined by (2-17) is called the FW space.

In the fit world, the normal matrix of δx^1 is assumed to exist and is designated S_{11}. Thus the matrix pair (E_1^o, S_{11}) completely defines the FW statistics of δx^o. For some purposes it is desirable to represent these statistics by a matrix pair (E_1^o, S_{oo}), where S_{oo} is a normal matrix in V^o. This may be accomplished artificially by letting S_{oo} be any matrix for which

$$S_{11} = E_1^{o\,T} S_{oo} E_1^o \qquad (2\text{-}18)^*$$

Such a matrix is said to be a valid extension of (E_1^o, S_{11}). By virtue of (2-15a) it is seen that one such valid extension, called a canonical extension of (E_1^o, S_{11}), is given by

$$S_{oo}^c = D_o^{1\,T} S_{11} D_o^1 \qquad (2\text{-}19)$$

In the fit world, δm^r is uncorrelated with δx^1 and is a zero-mean random variable with normal matrix J_{rr} ("W" in the old terminology). This information is available from TAPP I as the $n_o \times n_o$ matrix

$$J_{oo} = A_o^{r\,T} J_{rr} A_o^r \quad (= \text{"}A^T WA\text{"}) \qquad (2\text{-}20)$$

For future reference it is noted that on defining

$$A_\alpha^r = A_o^r E_\alpha^o \qquad \alpha = 1,\ 2 \qquad (2\text{-}21)$$

and

$$J_{\alpha\beta} = A_\alpha^{r\,T} J_{rr} A_\beta^r \quad \alpha = 1, 2;\ \beta = 1, 2 \qquad (2\text{-}22)$$

---

*The superscript "T" represents the transpose of the matrix it follows.

then

$$J_{\alpha\beta} = E_{\alpha}^{o\,T} J_{oo} E_{\beta}^{o} \qquad (2\text{-}23)$$

Thus J_{oo} is a valid extension of (E_1^o, J_{11}). It is assumed that the data is not poorly conditioned for regressing to δx^o—thus $(S_{11} + J_{11})^{-1}$ exists.

2.3 THE ESTIMATOR AND THE ERROR AFTER A REGRESSION

Since the estimate $\mu' x^o$ must possess the same constraints as those possessed by x^o in the fit world, it follows that $\mu'\delta x^o$ must possess the same constraints as those of δx^o in the fit world. In other words, $\mu'\delta x^o$ must be in the FW space. Hence

$$\mu'\delta x^o = E_1^o \mu'\delta x^1 \qquad (2\text{-}24)$$

where $\mu'\delta x^1$ is the linear, minimum-variance estimator of δx^1. This estimator may be rapidly found by substituting the constraint (2-17) into the regression equation (2-3). Thus

$$\delta z^r = A_1^r \delta x^1 + \delta m^r \quad (\text{FWP} = 1) \qquad (2\text{-}25)$$

where, by the definition (2-21), $A_1^r = A_o^r E_1^o$. Thus

$$\mu'\delta x^1 = (S_{11} + A_1^{r\,T} J_{rr} A_1^r)^{-1} A_1^{r\,T} J_{rr}\, \delta z^r \qquad (2\text{-}26)$$

Recalling the definition (2-22), $J_{11} = A_1^{r\,T} J_{rr} A_1^r$, and defining

$$'S_{11} = S_{11} + J_{11} \qquad (2\text{-}27)$$

$$'\Lambda^{11} = ('S_{11})^{-1} \qquad (2\text{-}28)$$

it follows that (2-26) may be written in the form

$$\mu'\delta x^1 = {}'\Lambda^{11} A_1^{r\,T} J_{rr}\,\delta z^r \tag{2-29}$$

In order to find $\mu'\delta x^o$, (2-29) is substituted into (2-24). Thus

$$\mu'\delta x^o = E_1^o\,{}'\Lambda^{11}(A_o^r E_1^o)^T J_{rr}\delta z^r$$

or

$$\mu'\delta x^o = {}'\Lambda^{oo} A_o^{r\,T} J_{rr}\,\delta z^r \tag{2-30}$$

where

$${}'\Lambda^{oo} = E_1^o\,{}'\Lambda^{11} E_1^{o\,T} \tag{2-31}$$

The matrices $'S_{11}$, $'\Lambda^{11}$, and $'\Lambda^{oo}$ have important FW statistical interpretations. From the theory of minimum-variance estimators, it follows that $'S_{11}$ is the FW normal matrix of the error

$$\delta' x^1 = x^1 - \mu' x^1 = \delta x^1 - \mu'\delta x^1 \tag{2-32}$$

in x^1 after the regression. Hence $'\Lambda^{11}$ is the FW covariance matrix of $\delta' x^1$. By (2-5), (2-17), and (2-24) it follows that the error $\delta' x^o$ in x^o after the regression is given by

$$\delta' x^o = E_1^o(\delta x^1 - \mu'\delta x^1) = E_1^o\,\delta' x^1 \tag{2-33}$$

Hence $'\Lambda^{oo}$ is the FW covariance matrix of $\delta' x^o$. Because of the FW constraints, the FW normal matrix of $\delta' x^o$ does not exist. However, on defining

$$'S_{oo} = S_{oo} + J_{oo} \tag{2-34}$$

where S_{oo} is any matrix satisfying (2-18), then it follows from (2-23) and (2-27) that

$$'S_{11} = E_1^{o\,T} S_{oo} E_1^o \qquad (2\text{-}35)$$

In other words, if S_{oo} is a valid extension of (E_1^o, S_{11}), then $'S_{oo}$ is a valid extension of $(E_1^o, 'S_{11})$. The use of (2-34) and (2-35) instead of (2-27) in the present TAPP III program offers a number of advantages. Since J_{oo} is available from TAPP I, it does not have to be "collapsed" by (2-22). Furthermore, if S_{oo} must be updated to the time of J_{oo}, then this may be accomplished by the propagation matrix available from TAPP I whereas an updating of S_{11} would require a transformation of the updating matrix.

The new error $\delta' x^o$ may be expressed as a linear combination of the old error δx^o and the measurement error δm^r. Substituting (2-3) into (2-30) yields

$$\mu' \delta x^o = '\Lambda^{oo} (A_o^{r\,T} J_{rr} A_o^r \delta x^o + A_o^{r\,T} J_{rr} \delta m^r) \qquad (2\text{-}36)$$

Substituting (2-36) into (2-5) and defining

$$\ell_o^o = I_o^o - '\Lambda^{oo} J_{oo} \qquad (2\text{-}37)$$

and

$$\delta m_o{}^T = A_o^{r\,T} J_{rr} \delta m^r \qquad (2\text{-}38)$$

yields

$$\delta' x^o = \ell_o^o \delta x^o - '\Lambda^{oo} \delta m_o{}^T \qquad (2\text{-}39)$$

2.4 THE REAL WORLD

It is of interest to note that so far no assumptions have been made concerning the "actual" statistics of δx^o and δm^r. These statistical assumptions are referred to as the real world (RW) and are described below. Once the real world is defined, the statistics of the error $\delta' x^o$ after the regression is determined by (2-39).

In the real world it is assumed that δx^o is a zero-mean random variable with covariance matrix $\bar{\Lambda}^{oo}$. Thus the possibility that δx^o is constrained to a subspace of V^o is unimportant, since it merely means that $\bar{\Lambda}^{oo}$ is singular. Initially, however, the statistics of δx^o may be given in the form of a covariance matrix of some parametrization of the space to which δx^o is constrained in the real world. Thus by analogy with the fit world, it is assumed that there is available a set of RW transformation matrices D^3_o, D^4_o, E^o_3, E^o_4 such that the equations

$$\delta x^3 = D^3_o \delta x^o \qquad (2\text{-}40a)$$

$$\delta x^4 = D^4_o \delta x^o \qquad (2\text{-}40b)$$

have the inverse

$$\delta x^o = E^o_3 \delta x^3 + E^o_4 \delta x^4 \qquad (2\text{-}41)$$

where, letting "RWP" mean "real world probability",

$$\delta x^4 = D^4_o \delta x^o = 0 \quad (\text{RWP} = 1) \qquad (2\text{-}42)$$

or

$$\delta x^o = E^o_3 \delta x^3 \quad (\text{RWP} = 1) \qquad (2\text{-}43)$$

Thus

$$\Lambda^{oo} = E_3^o \Lambda^{33} E_3^{o\,T} \tag{2-44}$$

where Λ^{33} is the RW covariance matrix of δx^3.

In the real world, δm^r is uncorrelated with δx^3 and is a zero-mean random variable with covariance matrix Σ^{rr} (called "M" in the old terminology). This information is available from TAPP I as the $n_o \times n_o$ matrix

$$\tilde{J}_{oo} = \Lambda_o^{r\,T} J_{rr} \Sigma^{rr} J_{rr} \Lambda_o^r \quad (= \text{"}A^T WMWA\text{"}) \tag{2-45}$$

Note that this "normal-type" matrix is merely the RW covariance matrix of the random variable δm_o^T defined by (2-38). The FW covariance matrix of δm_o^T is J_{oo}. In this sense, therefore, $\tilde{J}_{oo}$ is the RW analog of J_{oo}. In both the real and the fit world δm_o^T is correlated with δx^o.

From the RW statistics of δx^o and δm_o^T, it is easy to determine the RW statistics of $\delta' x^o$ from (2-39). Thus, since δx^o and δm_o^T are uncorrelated and have RW covariance matrices Λ^{oo} and $\tilde{J}_{oo}$, respectively, it follows that the RW covariance matrix of $\delta' x^o$ is given by

$$'\Lambda^{oo} = \ell_o^o \Lambda^{oo} \ell_o^{o\,T} + {'\Lambda^{oo}} \tilde{J}_{oo} {'\Lambda^{oo}} \tag{2-46}$$

2.5 THE REGRESSION ALGORITHM

The results of this section that are applicable to the programming of TAPP III are presented in the form of a set of computations—called The Regression Algorithm—which may be considered as a basic subroutine of TAPP III.

2.5.1 Inputs to the Algorithm

Fit World (FW) Inputs

a) E_1^o ($n_o x n_1$ matrix): The FW Constraint Matrix*

This matrix represents the partials of the TAPP I parameters with respect to the fit parameters (taking into account the constraints assumed by the fit world).

b) S_{oo}($n_o x n_o$ matrix): An Extended FW A Priori Normal Matrix

If S_{11} represents the a priori normal matrix of the fit parameters, then S_{oo} may be any matrix such that $S_{11} = E_1^{oT} S_{oo} E_1^o$.

c) J_{oo}($n_o x n_o$ matrix): The "$A^T WA$" Matrix

Real World (RW) Inputs

d) $\bar{\Lambda}_{oo}$($n_o x n_o$ matrix): The RW A Priori Covariance Matrix

This matrix represents the "true" a priori covariance matrix of the parameters of TAPP I.

e) $\tilde{J}_{oo}$($n_o x n_o$ matrix): The "$A^T WMWA$" Matrix

2.5.2 Outputs of the Algorithm

a) An extended FW a posteriori normal matrix:

$$'S_{oo} = S_{oo} + J_{oo}$$

b) The FW a posteriori normal matrix of the fit parameters:

$$'S_{11} = E_1^{o\,T}\,'S_{oo} E_1^o$$

c) The FW a posteriori covariance matrix of the fit parameters:

$$'\Lambda^{11} = ('S_{11})^{-1}$$

*This matrix, or more accurately, its transpose, is set up in a set of temporary locations from coded lists of permutations and constants.

d) The FW a posteriori covariance matrix of the TAPP I parameters:

$$ {}'\Lambda^{oo} = E_1^o \, {}'\Lambda^{11} E_1^{o\,T} $$

e) Partials of a posteriori errors in TAPP I parameters with respect to a priori errors in TAPP I parameters:

$$ L_o^o = I_o^o - {}'\Lambda^{oo} J_{oo} $$

f) RW a posteriori covariance matrix of TAPP I parameters:

$$ {}'\bar{\Lambda}^{oo} = L_o^o \, \bar{\Lambda}^{oo} L_o^{o\,T} + {}'\Lambda^{oo} \tilde{J}_{oo} \, {}'\Lambda^{oo} $$

3. THE MATRIX TRANSFORMATION TECHNIQUES OF TAPP III

In TAPP III, a wide variety of matrix manipulations is required. TAPP III must be able to permute, constrain, transform to different epochs, and partition the rows and columns of a number of different matrices. Furthermore, to fulfill its assigned tasks, these manipulations cannot be fixed, or determined by a few option flags. Indeed, it was found necessary to design a matrix language that would allow the user of the program to command, in any desired order, a number of different matrices, the rows and columns of which may relate to a large number of different parameter sets.

The possible output options are determined by the Coding Control Tables, which are "built into" the program as constants. The program is so general that the output options and the methods of coding them may be changed by appropriately modifying these tables. However, it would not be advisable to do this until an intimate knowledge of the program is attained.

3.1 THE MATRIX FORMAT TABLE (T3)

In TAPP III it is necessary to deal with many different types of matrices. Because of storage limitations, some matrices must "share" locations with a symbolic designation σ of each matrix a set of values that completely defines the representation of the matrix in the computer. This is accomplished by the table lookup operation,

$$(F, \zeta_t; \Delta_1, \Delta_2) = T3(\sigma),$$

which looks up the symbolic contents of σ in the first column of T3 and stores the contents of the associated entries of the table in $F, \zeta_t, \Delta_1, \Delta_2$. The significance of the contents of these locations is as follows:

F: The storage format of the matrix

ζ_t: The location of the matrix

Δ_1: The number of rows of the matrix

Δ_2: The number of columns of the matrix

The matrix defined by this table lookup operation is designated $[\sigma]$.

The storage format, F, of a matrix determines the method by which each entry of the matrix is stored in the program. There are five types of matrix storage formats used in TAPP III:

F = NS: The Matrix is Not Stored.

F = RR: Rectangular by Rows. Let $[\sigma]$ represent the matrix labled by the symbolic contents of σ. The value of the a-th row, b-th column of σ is given by

$$[\sigma; a, b] = \text{Con}[\ell_t - 1 + \Delta_2(a-1) + b] \begin{cases} 0 < a \leq \Delta_1 \\ 0 < b \leq \Delta_2 \end{cases} \quad (3\text{-}1)$$

where "Con" means "contents of."

F = TR: Triangular by Rows. This format is used to store symmetric matrices (hence $\Delta_1 = \Delta_2$). For this format:

$$[\sigma; a, b] = \text{Con}\left[\ell_t - 1 + \frac{(2\Delta_2 + 2 - a)(a-1)}{2} + b\right], \quad (3\text{-}2)$$

$$0 < a \leq b \leq \Delta_1$$

F = IP: Indirect Permutation Matrix. This format is used to represent a $n_\sigma \times n_\sigma$ permutation matrix by storing only the coordinates of its non-diagonal, non-zero entries. Since the entries are one, their values need not be stored. In physical format, an IP is merely a list of words of the form (a, b), indicating that there is a "1" in the a-th row, b-th column of $[\sigma]$. Let u and v be two n_σ-dimensional column vectors such that

$$u = [\sigma] v, \quad (3\text{-}3)$$

then the presence of a word (a, b) in the list represents the statement

$$"u(a) = v(b)" \tag{3-4}$$

If any component u(a') is not mentioned in the class of all such statements implied by the list, then u(a') = v(a').

F = IK: Indirect Constraint Matrix. This format is used to represent a $n_\sigma \times n_\sigma$ constraint matrix by storing only the coordinates and values of its non-diagonal, non-zero entries. In physical format, an IK is a list of word pairs of the form (a,b), e, where a and b are integers and e is a floating point number. The classes a and b of all a- and b-values, respectively, must have no values in common. If $[\sigma]$ is operating on a n_σ-dimensional column vector v, then $[\sigma]$ is represented as a set of mutually commuting transformations. Each word pair (a, b), e induces the transformation

$$\begin{aligned} v(i) &\Leftarrow v(i) \quad \text{for} \quad i \neq a \\ v(a) &\Leftarrow v(a) - e\, v(b) \end{aligned} \tag{3-5}$$

In other words, e times the b-th component of v is subtracted from the a-th component of v. Thus, interpreting a word pair as representing the statement

$$"\frac{\partial v(a)}{\partial v(b)} = e" \tag{3-6}$$

an IK matrix subtracts all non-zero constraints implied by the list of word pairs from the matrix v.

3.2 THE REPRESENTATION OPTIONS

A representation is a set of n_σ parameters that completely describes the physical system assumed by TAPP I. The representation options available* are

---

*Stating that a representation is available does not mean that the actual parameters or their nominal values are present in the program. TAPP III manipulates only the partials and statistical matrices describing the properties of these random parameters about fixed nominal values determined by TAPP I.

indexed by the contents of the two locations τ, χ. Each available representation $x^{\tau,\chi}$ may be written in the form

$$x^{\tau,\chi} = \begin{bmatrix} p^{\tau} \\ q^{\chi} \end{bmatrix} \qquad (3\text{-}7)$$

where p^{τ} is an orbit set option (n_p-dimensional) and q^{χ} is a system set option ($n_{\sigma} - n_p = n_q$ dimensional). For its internal computations, TAPP III represents its statistical matrices in the standard representation x (sometimes called the program representation), which is also a representation option defined by the option indices $\tau = 0$, $\chi = 1$. Notations applicable to x are:

$$x = x^{0,1} = \begin{bmatrix} p^{o} \\ q^{1} \end{bmatrix} \equiv \begin{bmatrix} p \\ q \end{bmatrix} \qquad (3\text{-}8)$$

TAPP III must be able to express normal and covariance matrices in arbitrary representations. It must also be able to find the partials matrix connecting any pair of representations. To accomplish this, the program computes a transformation matrix $\langle \mu, \tau, \chi |$ defined by

$$\langle \mu, \tau, \chi | = \begin{cases} \partial x^{\tau,\chi} / \partial x & \text{if} \quad \mu = 0 \\ \left[\partial x / \partial x^{\tau,\chi} \right]^{T} & \text{if} \quad \mu = 1 \end{cases} \qquad (3\text{-}9)$$

and stored in the utility matrix X_{σ} called the transformation accumulator. Thus the index μ, called the parity flag of the transformation matrix, determines whether or not partials are to be inverse-transposed (i.e., adjointed). In order to compute the transformation matrix, the partials $\partial x^{\tau,\chi}/\partial x$ must be expressed as a product (or chain) of primative matrices (called links), such that each link may be adjointed* without the use of a general inversion routine.

*The adjoint is chosen instead of the inverse because the adjoint preserves the order of matrix multiplication, thereby simplifying the logical sequencing of the program.

The first step in such a decomposition is to note that

$$\frac{\partial x^{\tau,\chi}}{\partial x} = \frac{\partial x^{\tau,\chi}}{\partial x^{\tau,1}} \cdot \frac{\partial x^{\tau,1}}{\partial x} \tag{3-10}$$

The function of the Orbit Set Options Table (see Table B2) is to locate for the program the matrix that determines $\partial x^{\tau,1}/\partial x$. The function of the System Set Options Table (see Table B3) is to locate the links of $\partial x^{\tau,\chi}/\partial x^{\tau,1}$. To see how this is done, it is necessary to describe the functional relationships that are assumed to exist between the representations $x^{\tau,\chi}$.

3.2.1 The Orbit Link

It is assumed that for each τ, p^{τ} is a function of the standard representation, i.e.,

$$p^{\tau} = p^{\tau}(\chi) = p^{\tau}(p, q) \tag{3-11}$$

and may be broken up in the form

$$p^{\tau} = \begin{bmatrix} r^{\tau} \\ v^{\tau} \end{bmatrix} \tag{3-12}$$

where r^{τ} and v^{τ} are $n_p/2$ dimensional vectors representing generalized position and momentum variables of classical mechanics. It follows from the well-known Poisson-Bracket relations between p^{τ} and p that

$$\left(\frac{\partial p^{\tau}}{\partial p}\right)^{*} = \Gamma \frac{\partial p^{\tau}}{\partial p} \Gamma^{T}, \quad \Gamma = \begin{bmatrix} O & I \\ -I & O \end{bmatrix} \tag{3-13}$$

where (*) represents the adjoint operation and where O and I are $n_p/2 \times n_p/2$ zero and identity matrices (in this formula only). From equations (3-7), (3-8), and (3-11) it follows that $\partial x^{\tau,1}/\partial x$, called the orbit link, is given by

$$\frac{\partial x^{\tau,1}}{\partial x} = \begin{bmatrix} \partial p^{\tau}/\partial x \\ \partial q/\partial x \end{bmatrix} = \left[\begin{array}{c|c} \partial p^{\tau}/\partial p & \partial p^{\tau}/\partial q \\ O_{qp} & I_{qq} \end{array}\right] \tag{3-14}$$

where O_{qp} is the $(n_{\sigma} - n_p) \times n_p$ zero matrix and I_{qq} in the $(n_{\sigma} - n_p) \times (n_{\sigma} - n_p)$ identity. From (3-13) the adjoint of the orbit link is given by

$$\frac{\partial x^{\tau,1}}{\partial x}^{*} = \left[\begin{array}{c|c} \left(\frac{\partial p^{\tau}}{\partial p}\right)^{*} & O_{pq} \\ \hline -\left(\frac{\partial p^{\tau}}{\partial q}\right)^{T}\left(\frac{\partial p^{\tau}}{\partial p}\right)^{*} & I_{qq} \end{array}\right] \tag{3-15}$$

3.2.2 <u>The Orbit Set Options Table</u> (TO)

From equations (3-14) and (3-15) it is seen that the matrices $\partial p^{\tau}/\partial x$ uniquely determine the orbit links and their adjoints. These $n_p \times n_{\sigma}$ matrices are stored in the computer in certain locations, which are, for some orbit set options (namely $\tau = 4$ and $\tau = 5$) row-partitionings of larger matrices of format RR. The function of the Orbit Set Options Table (see Table B2) is to associate with each orbit set index τ a set of values that determines the location of the associated orbit set partials $\partial p^{\tau}/\partial x$. This is accomplished by the table lookup operation

$$"(\lambda_t, \xi) = TO(\tau)"$$

from which the location of $\partial p^{\tau}/\partial x$ is determined by

$$Loc(\partial p^{\tau}/\partial x) = \lambda_t + \xi n_p n_{\sigma} \tag{3-16}$$

From Table B2 it is seen that there are two types of orbit set options. Those representing orbit parameters at various times ($\tau = 0, 1, 2, 3$) will always be valid and cannot be altered or added-to without modifying the overall sequencing. However, those options representing the second and third set of six miss parameters (τ = 4, 5) are valid only for the present set of miss parameters assumed by TAPP I. The first set of six miss parameters is not an orbit set option since its partials do not satisfy (3-13).

3.2.3 The System Links

The system sets are supposed to represent physical constants and biases, hence they must be functionally independent of the standard orbit set p. Thus for all

$$q^{\chi} = q^{\chi}(x) = q^{\chi}(q) \tag{3-17}$$

On recalling (3-7) and (3-8), it follows that

$$\frac{\partial x^{\tau,\chi}}{\partial x^{\tau,1}} = \begin{bmatrix} I_{pp} & O_{pq} \\ O_{qp} & \frac{\partial q^{\chi}}{\partial q} \end{bmatrix} \tag{3-18}$$

This matrix, called the system chain of χ, is independent of τ, but must be factored into a product of analytically adjointable links before it can be produced by the program. Accordingly, it is assumed that to each system index χ, except $\chi = 1$, there is assigned a unique system index χ' (called the predecessor of χ), such that $\chi = \chi'$ and

$$q^{\chi} = f^{\chi}(q^{\chi'}) \tag{3-19}$$

where f^{χ} is either a permutation function (meaning that q^{χ} is a reordering of the components of $q^{\chi'}$) or a constraint function (meaning that f^{χ} subtracts functions of one subset of the components of $q^{\chi'}$ from another disjoint subset of its components). From (3-19)

$$\frac{\partial x^{\tau,\chi}}{\partial x^{\tau,\chi'}} = \begin{bmatrix} I_{pp} & O_{pq} \\ O_{qp} & \frac{\partial q^{\chi}}{\partial q^{\chi'}} \end{bmatrix} \tag{3-20}$$

Such matrices, called system links, are independent of τ and are either permutation or constraint matrices - both of which are very easy to adjoint, as is shown below.

Let P be an arbitrary permutation matrix. Then for arbitrary $(n_{\bar{o}} \times 1)$ matrices u, v

$$(Pu)^T(Pv) = u^T v$$

since both sides of this equation represent sums of the same product in a different order. Hence

$$u^T(P^T P)v = u^T v$$

for all u, v. It follows that $P^T P = I$. In other words

$$P^* = (P^{-1})^T = P \qquad (3\text{-}21)$$

Therefore, permutation matrices are self-adjoint.

By definition, an arbitrary constraint matrix K may be written in the form

$$K = I - E \qquad (3\text{-}22)$$

where E is a matrix having the property that the class of all integers labeling the rows of E that possess non-zero entries has no member in common with the class of all integers labeling the columns of E that possess non-zero integers. Therefore E is nilpotent ($E^2 = 0$). Thus

$$(I + E)(I - E) = I - E^2 = I$$

whence it follows from (3-22) that

$$K^* = I + E \qquad (3\text{-}23)$$

3.2.4 The System-Set Options Table (T2)

The system chain of χ is formed from system links by chain-ruling down to $\chi = 1$ in the fashion:

$$\frac{\partial x^{\tau,\chi}}{\partial x^{\tau,1}} = \frac{\partial x^{\tau,\chi}}{\partial x^{\tau,\chi'}} \cdot \frac{\partial x^{\tau,\chi'}}{\partial x^{\tau,\chi''}} \cdots \underbrace{\frac{\partial x^{\tau,\chi^{(n-1)}}}{\partial x^{\tau,\chi^{(n)}}}}_{\text{the n-th -link}} \cdots \frac{\partial x^{\tau,\chi^{(\chi-1)}}}{\partial x^{\tau,1}} \tag{3-24}$$

where n_χ is called the order of χ. Thus n_χ represents the number of links, including the orbit link in the chain of the transformation $\langle\mu, \tau, \chi|$. The function of the System-Set Options Table (see Table B3) is to assign to each meaningful ordered pair (χ, n) the symbolic designation σ of the n-th χ-link by means of the table lookup operation

$$\text{"}\sigma = T2(\chi, n)\text{"}$$

It then follows from Section 3.1 that the operation

$$\text{"}(F, \ell_t ; \Delta_1, \Delta_2) = T3(\sigma)\text{"}$$

completely defines the matrix σ. In other words

$$\left[T2(\chi, n)\right] = \frac{\partial x^{\tau,\chi^{(n-1)}}}{\partial x^{\tau,\chi^{(n)}}} \tag{3-25}$$

Hence

$$\frac{\partial x^{\tau,\chi}}{\partial x^{\tau,1}} = \left[T1(\chi, 1)\right] \cdots \left[T1(\chi, n)\right] \cdots \left[T1(\chi, n_\chi - 1)\right] \tag{3-26}$$

From Table B1 and B3 it is seen that each of the links in this chain are either indirect permutations (F = IP) or indirect constraint (F = IK); hence they are not stored as full-sized ($n_\sigma \times n_\sigma$) matrices. Thus the multiplications implied by (3-26) are carried out indirectly and from right to left. These multiplications are represented in the program as operations performed on the columns of the transformation accumulator X_σ. Furthermore, whether the links are used or the adjoints of the links are used is automatically determined by the value of the parity index μ. This is made possible by the order preserving property of the adjointing operation, together with the simple adjointing algorithms implied by equations (3-21) and (3-23).

3.3 THE SET-PARTITION OPTIONS TABLE (T1)

Thus far, a technique for obtaining a transformation matrix $\langle \mu, \tau, x|$ defined by equation (3-9) has been described. By use of such matrices in conjunction with the matrix options (see Table B3), it would be possible to find covariance matrices and normal matrices relative to any representation x^{τ}. It would also be possible to find partials relating any pair of representations. As powerful as this tool is, it is inadequate from several viewpoints. Printing out a ($n_\sigma \times n_\sigma$) matrix where n_σ may be as large as 50 would be rather awkward, if one is interested in only a small partition of this matrix. Also TAPP II requires partitionings of its matrices on the T_{32} tape. Finally, the transformation accumulator is inherently incapable of describing the partials of arbitrary miss parameters with respect to the standard representation (and this must be done if covariance matrices of such parameters are to be found).

To alleviate these difficulties the partials matrix $(\mu, \tau, \rho|$ is introduced. As its notation implies, this matrix, which is stored in the partials accumulator X, is a function of the contents of the three locations μ, τ, ρ. As before (see Section 3.2.2):

μ is the parity flag

τ is the orbit-set option

The index ρ, called the set-partition option, serves as an input to the table lookup operation

$$"(\chi, n; t, h) = TH(\rho)" \tag{3-27}$$

on the <u>Set-Partition Options Table</u> (see Table B4). The outputs of this operation are used to determine $(\mu, \tau, \rho|$ in a manner described below.

To each pair of integers (t, h) satisfying the inequalities

$$0 \leq t \leq h \leq n_{\varpi} \tag{3-28}$$

there is associated a $(h - t \times n_{\varpi})$ matrix ϕ_h^t called a <u>partition of tail t and head h</u> that assigns to each n_{ϖ}-dimensional column vector v, a vector $(\phi_h^t v)$ with components

$$(\phi_h^t v)(i) = v(i + t), \quad 0 < i \leq h - t \tag{3-29}$$

Thus, a partition matrix may be considered as a row-partitioning of the [illegible] identity matrix I. The definition of the partials matrix is now written in [illegible] cise form:

$$(\mu, \tau, \rho| = \begin{cases} \phi_h^t \langle\mu, \tau, \kappa| & \text{for} \quad \chi \neq 0 \\ \phi_h^t \langle n| & \text{for} \quad \chi = 0 \end{cases} \tag{3-30}$$

where χ, [illegible] are the functions of [illegible] defined by (3-27), and where $\langle n|$ [illegible] cates an $n_{\varpi} \times n_{\varpi}$ matrix* whose first location is given by the contents of [illegible]

---

*Actually $\langle n|$ is meant to refer to an $\alpha \times n_{\varpi}$ matrix, where α is not necessarily equal to n_{ϖ}. However, this makes no difference to the partitioned matrix $\phi_h^t\langle n|$, provided $h \leq \alpha$.

If $\chi \neq 0$, ρ is called a system-partition option. Such an option designates both the desired system q^{χ} and a partitioning of the representation

$$x^{\tau,\chi} = \begin{bmatrix} p^{\tau} \\ q^{\chi} \end{bmatrix}$$

It is of interest to note that for $\mu = 0$:

$$(0, \tau, \rho| = \frac{\partial(\phi_h^t x^{\tau,\chi})}{\partial x} \qquad (3\text{-}31)$$

However, "$(1, \tau, \rho| = \left[\partial x / \partial(\phi_h^t x^{\tau,\chi})\right]^T$" is not true unless $t = 0$, $h = n_{\sigma}$. This is because the parameter set $\phi_h^t x^{\tau,\chi}$ does not determine the representation $x^{\tau,\chi}$. If $\chi \neq 0$, n is not needed to define the partials matrix; and therefore communicates to the program the number of links in the chain of partials defining the transformation matrix.

If $\chi = 0$, ρ is called an isolated set option. An isolated set may be interpreted as a partitioning of an available set - that is, parameter sets whose partials with respect to the standard representation are stored in TAPP III. Notice that the partials matrix for isolated set options is independent of μ and τ. However, since no machinery is available for adjointing the particles matrix of an available set, it is appropriate to set $\mu = 0$ for this case. Let α_n designate the available set whose partials are stored at the location n. Then from (3-30) it follows that

$$(0, -, \rho| = \frac{\partial(\phi_h^t \alpha_n)}{\partial x} \qquad (3\text{-}32)$$

where the blank is used to indicate that the orbit set index is not used. Comparing equations (3-31) and (3-32) shows that system-partition and isolated set options may be treated on the same footing for $\mu = 0$ - the only difference being that system-partition options require an additional index τ to define them.

3.4 THE OUTPUT ENTRIES

TAPP[illegible] has three [illegible] reserved for the storage of command lists. The [illegible] locat[illegible]

UPC: The Update Commands

POC: The [illegible] Commands

[illegible]: The [illegible] Output Commands

The Update Commands (UPC) [illegible] start [illegible] program [illegible] standard [illegible] The Request Output Commands (RQC) [illegible] [illegible] these [illegible] the [illegible] (PC) [illegible] to the program [illegible] previous.

Each command [illegible] of five words storing the seven [illegible] $(\sigma_0, \mu_1, \tau_1, \rho_1, \mu_2, \tau_2, \rho_2)$ [illegible]

← word 1 →	← word 2 →	← word 3 →	← word 4 →	← word 5 →
σ_0	μ_1, τ_1	ρ_1	μ_2, τ_2	ρ_2

These seven [illegible] τ_2 stored [illegible] and deter[illegible]

$$[\mu_1, \tau_1, \rho_1] [\sigma_{0}] [\mu_2, \tau_2, \rho_2]^T$$

[illegible] $[\sigma_0]$ represents the matrix operator designated by σ_0 [illegible] Table 3P, and the partial matrix $[\mu, \tau, \rho]$ is that matrix [illegible] μ, τ, ρ defined in the previous section. [illegible] abbreviation

$$(---|^T = |---) \qquad (3\text{-}34)$$

and omitting the brackets about σ_o in (3-33), an output entity may be designated more economically in the form

$$Z = (\mu_1, \tau_1, \rho_1 \big| \sigma_o \big| \mu_2, \tau_2, \rho_2) \qquad (3\text{-}35)$$

The parity flags μ_1 and μ_2 cannot be chosen arbitrarily but depend upon the matrix option σ_o if Z is to have physical meaning. If (3-35) is valid, then it is said that σ_o has parity (μ_1, μ_2).

3.4.1 The Coding of Covariance Matrices

A covariance matrix has parity (0, 0). Thus any values of τ_1, τ_2 in the Orbit-Set Options Table and any values of ρ_1, ρ_2 in the Set-Partition Options Table may be chosen. It is well to keep in mind, however, that isolated set options require less machine time than do system-partition options (since the former merely sets the format of the Partials Accumulator X, which is a dummy matrix). For example, although the partials

(0, 1, p |
(0, 1, cp |
(0, 1, fp |
(0, 1, σp |
(0, –, pk |

are all identical, the last one should be chosen.

3.4.2 The Coding of Normal Matrices

A normal matrix has parity (1, 1), hence only system-partition options may be chosen for them. Usually in the case of normal matrices, one chooses $\tau_1 = \tau_2$ and $x_1 = x_2$ (where, of course, x_1 and x_2 refer to the values obtained for x, when ρ_1 and ρ_2, respectively, are substituted into (3-37)). For this choice the output entity takes the form

$$Z = \phi_{h_1}^{t_1} \left\langle 1, \tau, x \middle| \sigma_o \middle| 1, \tau, x \right\rangle \left(\phi_{h_2}^{t_2}\right)^T \tag{3-36}$$

and thus represents a partitioning of the rows and columns of the normal matrix referred to the representation $x^{\tau, x}$. Excellent examples of output entities of this type are provided by RŌC for $\omega = 4, 5,$ and 6. One subtility must be mentioned here: the extended normal matrices S and S' (see Section 2) are valid only for $\tau = fp$ and $\tau = fl$. The reason for this is that they were originally only defined for the parameters being fitted and were artificially extended into the standard representation

3.4.3 The Coding of Partials

The matrix option $\sigma_o \equiv I$ has two valid parities, namely (0, 1) and (1, 0). It is easy to show, however, that

$$(0, \tau_1, \rho_1 | I | 1, \tau_2, \rho_2) = (1, \tau_2, \rho_2 | I | 0, \tau_1, \rho_1)^T \tag{3-37}$$

The use of the identity matrix option inables one to find partials of the form

$$\left[\frac{\partial\left(\phi_{h_1}^{t_1} \, x^{\tau_1, x_1}\right)}{\partial x^{\tau_2, x_2}}\right]\left(\phi_{h_2}^{t_2}\right)^T \tag{3-38}$$

and

$$\left[\frac{\partial\left(\phi_{h_1}^{t_1} \, \alpha_n\right)}{\partial x^{\tau_2, x_2}}\right]\left(\phi_{h_2}^{t_2}\right)^T \tag{3-39}$$

and, by the use of (3-37), their transposes.

4. A GENERAL DESCRIPTION OF TAPP III

The purpose of this section is to outline, in as concise a form as possible, the operation of the subroutines and the overall logic flow of the program. A complete functional description of the program from the engineering viewpoint is given in the tables and flow diagrams of the appendixes. Therefore, this section, in conjunction with the matrix-manipulation techniques and theory discussed in section 2, serves as a guide to understanding the program defined in the appendixes. In order for this section to serve as a reference to the program, it is made as independent as possible from the rest of the report.

4.1 THE INPUT TAPE FORMAT

In order for TAPP I to produce the input tape T_{13}, TAPP I is given a list of input times, $\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n$ (in ascending order), and a trajectory. From the trajectory, TAPP I computes a set of epoch times, $\hat{t}_o, \hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n$ (in ascending order), which are interpolated into the input times to produce a set of times, $t_o, t_1, t_2, \ldots, t_{nIT}$ (in ascending order), called input-tape times (or T_{13} times), which include all input and all epoch times. Thus, each T_{13} time is either an input time or an epoch time. (Of course $t_o = \hat{t}_o$ and is the injection time or epoch for the each phase.) The information on the T_{13} tape is stored in the following format:

Initial Information

IDEN	Identification of tape
B ($n_m \times n_o$) matrix	The partials of the miss parameters with respect to the TAPP I parameters at injection

The k-th Record (The information associated with the time interval (t_{k-1}, t_k), $k = 1, 2, \ldots, n_{IT}$ is stored in the following format:)

j	Epoch time index(By definition j either remains the same for each successive time interval or, if it changes it changes to k-1.)
k	The current time index

INFO	This data block is reserved to supply any desired information associated with time t_k (for example, t_k itself, orbit parameters etc.)
U (n_p x n_o matrix)	The partials of the time-varying parameters at t_k with respect to the TAPP I parameters at t_j

(For the s-th Station:)

s	The station number*
ΔJ	The A^TWA matrix for the s-th station during the time interval (t_{k-1}, t_k) referred to the epoch t_j
$\Delta\tilde{J}$	The A^TWMWA matrix for the s-th station during the time interval (t_{k-1}, t_k) referred to the epoch t_j

---

* The end of this block is terminated by a zero value for s. The stations need not be in numerical order.

4.2 THE SUBROUTINES OF TAPP III

TAPP III keeps track of the following quantities:

Sequencing Controls

i^*	Number of next print output block
ℓ'	Index of next print output time
i'	Number of next recycle time
ℓ'	Index of next recycle time
j_T	Epoch index of the next T_{13} record (i.e., k + 1st record)

Output Block Identification

j	Program epoch
ℓ	Index of last recycle time[1]
k	Index of current time (i.e., record number)
INFO	Identification of the k-th record

Output Entity Identification

ω	Entity number
COM	Entity designation (coded sequence of commands for producing the desired output)

Stored Partials (The following matrices are partials with respect to the standard representation[2] x of the indicated parameter sets:)

I	$(\partial p^j/\partial x)$ orbit parameters at program epoch (upper $n_p \times n_o \times n_o$ identity matrix)
U	$(\partial p^k/\partial x)$ orbit parameters at current time
V	$(\partial p^\ell/\partial x)$ orbit parameters at last recycle time[2]
W	$(\partial p^i/\partial x)$ orbit parameters at injection
B	$(\partial m/\partial x)$ miss parameters

---

[1] The last recycle time is the time at the beginning of the i'-th tracking interval of T_{32}. In this context, recycle times are also midcourse correction times.

[2] $x = \begin{bmatrix} p^j \\ \hline q \end{bmatrix}$ where p^j represents the orbit parameters at the program epoch and q represents the TAPP I system of constant parameters.

Tracking Accuracy (The following matrices are referred to the program epoch:)

J	The A^TWA matrix from ℓ to k
$\tilde{J}$	The A^TWMWA matrix from ℓ to k

A Priori Statistics (Also referred to program epoch)

S	The fit world normal matrix
$\bar{\Lambda}$	The real world covariance matrix

Before discussing the overall sequencing of TAPP III, it is necessary to briefly describe the functioning of the following subroutines:

4.2.1 The Partials Accumulator Subroutine

This subroutine was discussed in detail in section 3. However, the concepts developed in that section are outlined below for easy reference.

In TAPP III there are many types of parameter sets that must be considered. Since all pertinent statistical matrices are stored relative to the standard representation x, it is possible to transform them to any representation α (or partition thereof), provided a subroutine is available that can generate $\partial\alpha/\partial x$ or $(\partial x/\partial\alpha)^T$ and appropriately partition its rows. This is the task assigned to the partials accumulator subroutine. Basically, this subroutine accepts as inputs the symbols μ, τ, ρ to produce an output, designated by $(\mu, \tau, \rho|$, at the variable location X (called the partials accumulator). For certain values of ρ (referred to as the isolated set options) μ and τ are ignored, and X is set equal to certain partitioning of the stored partials. Other values of ρ (called the system-partition options) represent both a choice of constant parameters q^{χ} and partitioning of representation

$$x^{\tau,\chi} = \begin{pmatrix} p^{\tau} \\ q^{\chi} \end{pmatrix}$$

where p^{τ} is the orbit parameter option labeled by the index τ. In the latter case a fixed set of locations X_o (called the transformation accumulator) is filled in accordance with the relation

$$X_o = \begin{cases} \partial x^{\tau,\chi}/\partial x \text{ for } \mu = 0 \\ (\partial x/\partial x^{\tau,\chi})^T \text{ for } \mu = 1 \end{cases}$$

and X is set equal to the required partitioning of X_o.

4.2.2 The Executive Program (See Figure C2)

The Executive Program uses the partials accumulator to produce an output option (called an entity) at the set of locations Z from the coded command COM. This command consists of five words, which contain values of the seven variables σ_o; μ_1, τ_1, ρ_1; u_2, τ_2, ρ_2 stored in the format

Word 1	Word 2	Word 3	Word 4	Word 5
σ_o	(μ_1, τ_1)	ρ_1	(μ_2, τ_2)	ρ_2

The Executive Program accepts COM as an input to produce output

$$Z = (\mu_1, \tau_1, \rho_1| \begin{bmatrix}\sigma_o\end{bmatrix} |\mu_2, \tau_2, \rho_2)$$

where $|. \ . \ .) = (. \ . \ .|^T$ and where $\begin{bmatrix}\sigma_o\end{bmatrix}$ is a matrix option labeled by the index σ_o.

4.2.3 The Statistical Input Program (See Figure C3)

This program accepts the inputs

S_o $(n_{fl} \times n_{fl})$	Fit world normal matrix of the fitted parameters at injection
Λ_o $(n_{rl} \times n_{rl})$	Real world covariance matrix at injection of those parameters that are nominally unconstrained in the real world

These inputs are introduced by the read-in program as the a priori inputs to TAPP III and produce the outputs

$$S' = |0, 0, fl) S_o (0, 0, fl|$$

$$\Lambda'' = |1, 0, rl) \Lambda_o (1, 0, rl|$$

which are transformations of the a priori statistics to the standard representation.

4.2.4 The Statistical Output Program (See Figure C3)

This program processes the input matrices J, $\tilde{J}$, S, $\bar{\Lambda}$ and produces, with the aid of the Partials Accumulation Subroutine, the outputs

S'	The fit world normal matrix at current time
Λ'	The fit world covariance matrix at current time
$\bar{\Lambda}'$	The real world covariance matrix at current time

in the following manner:

1) $S' \leftarrow S + J$

2) $Z \leftarrow (1,\ 0,\ fl|\ S'\ |1,\ 0,\ fl)$

$Z \leftarrow Z^{-1}$

$\Lambda' \leftarrow |1,\ 0,\ fl)\ Z\ (1,\ 0,\ fl|$

3) $X_o \leftarrow I - \Lambda' J$

4) $Z \leftarrow X_o\ \bar{\Lambda}'\ X_o^T$

$\bar{\Lambda}' \leftarrow Z$

5) $X_o \leftarrow \Lambda'$

$Z \leftarrow X_o\ J\ X_o^T$

$\bar{\Lambda}' \leftarrow \bar{\Lambda}' + Z$

4.3 THE SEQUENCING OF OPERATIONS

TAPP III is initiated by the Read-In Program, which places the manual inputs in their allotted locations. The sequencing is initialized as follows:

$i^* = 0$

$\ell^* = 0$

$i' = 0$

$\ell' = 0$ $\quad W \leftarrow I$

$j_T = 0$

$j = -1$

$\ell = 0$

$k = 0$

The initial information IDEN and B of T_{13} is read into the locations INFO and B of the program, after which TAPP III enters the k-th cycle for k = 0. During the k-th cycle, the epoch for the k + 1st record of T_{13} is read-in, whereupon the program executes the following four phases (indicated by the phase flag PH) in the indicated order:

PH = 1. The Update Phase

PH = 2. The Print Output Phase

PH = 3. The Recycle Phase

PH = 4. The Tape Read-in Phase

The events that occur during the first three phases are controlled by the following mode flags, which are introduced as manual inputs:

δ_{PM} If this flag is zero, there is no print output, and no statistical computations occur.

δ_P If $\delta_P = 1$, each recycle output is printed.

δ_T If $\delta_T = 1$, each recycle output is written on the T_{32} tape.

4.3.1 PH = 1 The Update Phase

The purpose of the update phase is to update the stored quantities, provided $j_T \neq j$. If $j_T = j$, PH = 1 is skipped. Thus when the k + 1st record of T_{13} is referred to a new epoch, the program epoch will be the same at the end of PH = 1, thereby enabling the read-in of the k + 1 st record to give valid results when the program enters the read-in phase. If an update occurs (and $k \neq 0$), the program executes the Update Phase Commands (UPC) which cause the matrices V, W, B, J, $\mathfrak{J}$, S, $\bar{\Lambda}$ to be updated as follows:

Operation	Formal Command
$V \leftarrow V\ \partial x/\partial x^k = \partial p\ell/x^k$	(I; 0, -, $p\ell$; 1, 1, 0)
$W \leftarrow W\ \partial x/\partial x^k = \partial p^i/\partial x^k$	(I; 0, -, pi; 1, 1, 0)
$B \leftarrow B\ \partial x/\partial x^k = \partial m/\partial x^k$	(I; 0, -, m; 1, 1, 0)
$J \leftarrow (\partial x/\partial x^k)^T\ J(\partial x/\partial x^k)$	(J; 1, 1, 0; 1, 1, 0)
$\tilde{J} \leftarrow (\partial x/\partial x^k)^T\ J(\partial x/\partial x^k)$	($\tilde{J}$; 1, 1, 0, 1, 1, 0)
$S \leftarrow (\partial x/\partial x^k)^T\ S(\partial x \pi \partial x^k)$	(S; 1, 1, 0; 1, 1, 0)
$\bar{\Lambda} \leftarrow (\partial x^k/\partial x)\ (\partial x^k/\partial x)^T$	($\bar{\Lambda}$; 0, 1, 0; 0, 1, 0)

The update of S and $\bar{\Lambda}$ is omitted if $\delta_{PM} = 0$. After executing the above updates, the program perform. the following operations for all k (including k = 0).

$$U \leftarrow I$$

$$j = k$$

In effect, therefore, when the epoch of the k + 1 st record changes, it is changed to the representation associated with the current (k-th)time; that is,

$$x \leftarrow x^k = \begin{bmatrix} p^k \\ q \end{bmatrix}$$

4.3.2 PH = 2. The Print Output Phase

If $k = \ell^*$ and $\delta_{PM} = 1$, a print output is executed, otherwise PH = 2 is bypassed. During a print output the following events occur in the indicated order:

1) If k = 0, S' and $\bar{\Lambda}'$ are computed by use of the Statistical Input Program. Otherwise S', Λ', and $\bar{\Lambda}'$ are computed by use of the Statistical Output Program.

2) The heading of the i*-th output to 0_3 is printed. This heading takes the format

i*, j, ℓ, k, INFO

3) For $\omega = 1, 2, \ldots, n_{PO}$ the Print Output Commands (POC), which are manual inputs, are executed by the Executive Program and are printed out to 0_3 in the format

$$\omega; \sigma_0; \mu_1, \tau_1, \rho_1, \mu_2, \tau_2, \rho_2$$

[Contents of Z]

4) The index of the next print output time is selected from the Print Output Times (POT), which are manned inputs, by the following operations

$$i^* = i^* + 1$$

$$\ell^* = \text{POT}(i^*)$$

4.3.3 PH = 3. The Recycle Phase

If $k = \ell'$ the program recycles, otherwise PH = 3 is bypassed. During a recycle the following events occur in the indicated order:

1) The matrices S', Λ', and $\overline{\Lambda}'$ are computed from the Statistical Output Program. If these matrices were previously computed in PH = 2, or if $\delta_{PM} = 0$, this step is skipped.

2a) If $\delta_T = 1$, the T_{32} heading is written.

2b) If $\delta_P = 1$, the 0_3 heading for a recycle output is printed in the format

$$i', j, \ell, \kappa, \text{INFO}$$

3) If $k \neq 0$, and if δ_T and δ_P are not both zero, then for $\omega = 1, 2, \ldots n_{RO}$ the ω-th command of the Recycle Output Commands (ROC), which are manual inputs, is executed by the Executive Program. Each output is communicated to T_{32} or 0_3 or both, in accordance with the following logic:

a) If $\delta_T = 1$, the contents of Z are written on T_{32} in a tape format compatible with TAPP II.

b) If $\delta_P = 1$, the contents of Z are printed out to 0_3 in the same format as that of PH = 2.

4) The last recycle time is set equal to the current time and the next recycle time index is selected from the Recycle Times (RET), a manual input, by the following operations:

$$\ell \longleftarrow k$$

$$V \longleftarrow U$$

$$S \longleftarrow S'$$

$$\bar{\Lambda} \longleftarrow \bar{\Lambda}'$$

$$J \longleftarrow 0$$

$$\tilde{J} \longleftarrow 0$$

$$i' \longleftarrow i' + 1$$

$$\ell \longleftarrow RET(i')$$

4.5.4 PH = 4. The Tape Read-In Phase

If $k = n_{IT}$, the program stops. If $k < n_{IT}$, the program executes a read-in of the k + 1 st record of T_{13} (except for the epoch time index of this record, which was read in at the beginning of the k-th cycle) in the following manner:

1) k, INFO, and U are obtained from T_{13}.

2) For s = 1, 2, . . . n_{TS} the J and $\tilde{J}$ matrices are modified in the following manner: If the Time-Station Flag $\Delta_{TS}(k, s) = 1$, then

$$J \longleftarrow J + \Delta J$$

$$\tilde{J} \longleftarrow \tilde{J} + \Delta\tilde{J}$$

Otherwise ΔJ and $\Delta\tilde{J}$ are skipped. Since the Time-Station Flags are manual inputs (which are normally equal to one), it is therefore possible to omit any station at any time by setting the corresponding flag equal to zero.

REFERENCES

1. "Monte Carlo Simulation of Midcourse Guidance (TAPP MOD II)," by W. H. Pace, Jr., STL No. 8976-6012-RU-000, dated 31 October 1962.

2. "Capability Guide -- Tracking Accuracy Prediction Program (TAPP MOD I) (Revised)," by A. S. Liu, STL No. 8976-6013-RU-000, 1 November 1962.

3. "TAPP III, Part 1. Regression to a Constrained Parameter Set," STL No. 9861.5-182, by W. W. Lemmon, dated 6 March 1963.

4. "TAPP III, Part 2. Specification of the Format of the Tape Input to TAPP III," STL No. 9861.5-194, by W. W. Lemmon, dated 29 March 1963.

5. "Partial Derivatives of Radar Observational Parameters (for TAPP I)," STL No. 9851-76, by K. Heuhn, dated 27 March 1963.

6. "Computer Program Guide, Tracking Accuracy Prediction Program (TAPP MOD I-III)," by M. Kemp.

7. "Computer Program Guide, Tracking Accuracy Prediction Program (TAPP MOD I-III)," by S. Senda.

APPENDIX A. TAPP III MASTER SYMBOL LIST

Following is a complete list of definitions of the symbols used in the program. If a symbol represents a matrix with α rows and β columns, its format is indicated by $(n;\alpha x\beta)$, where n is its storage format. If n = RR, the storage is rectangular by rows; if n = TR, the storage is triangular by rows. Triangular format is only used for symmetric matrices. If each entry of the matrix consists of ν consecutive words (where $\nu \neq 1$), this fact is indicated by the notation $(n,\nu;\alpha x\beta)$. The following additional notation is used to indicate special types of symbols:

> $(\equiv\theta)$ indicates that the first location represented by the symbol is the same as that of the symbol θ.
>
> NS indicates that the matrix represented by the symbol is not stored but is generated internally by the program when needed
>
> DM indicates a dummy matrix: that is, a matrix whose first location is a variable determined by the program.

A. 1 PROGRAM INPUTS

Tape Description

n_{IT} : Number of tape input times

n_{TS} : Number of tracking stations

n_o : Number of parameters in the original representation

n_p : Number of parameters in the orbit set (must = 6 or = 12)

n_m : Number of parameters in the miss set

Print Controls

δPM : Print Mode Flag. If δPM = 0, there is no print output, and all statistical computations are omitted

n^* : Number of print outputs

POT : (RR; $n^* \times 1$) Print Output Time List. This list contains the indices, in ascending order, of the input tape times at which a print output is desired.

n_{pO} : Number of print output commands

POC : (RR, 5; n_{po} x 1) Print Output Command List

Recycle Controls

n' : Number of recycle times

RET : (RR, n' x 1) Recycle Time List. This list contains the indices, in ascending order, of the input tape times at which a recycle is desired.

δ_P : Print Recycle Flag. If $\delta_P = 1$, a print occurs at the recycle time.

δ_T : Tape Recycle Flag. If $\delta_P = 1$, a tape is written at the recycle time.

n_{RO} : Number of recycle outputs

RO : (RR, 5; n_{Ro} x 1) Recycle Output Command List

Common World

Δ_{TS} : (RR, 1/12, $n_t \times n_s$) Time-Station Flags. If $\Delta_{TS}(k, s)$ is set equal to zero, the s-th station is ignored at the k-th time. Normally all the locations of Δ_{TS} have unit value, indicating that all stations are operating at all times.

n_{c1} : Number of common parameters ($n_p \leq n_{c1} \leq n_o$)

n_{Pc} : Number of permutations in P_c ($0 \leq n_{Pc} \leq n_o - n_p$)

P_c : (RR; n_{Pc} x 1) Common Permutation List

n_{Kc} : Number of common constraints

K_c : (RR; n_{Kc} x 2) Common Constraint Partials List

Fit World (FW)

n_{f1} : Number of fitted parameters ($n_p \leq n_{f1} \leq n_{c1}$)

n_{Kf} : Number of FW constraints

K_f : (RR; n_{kf} x 2) FW Constraint Partials List

$S_o (\equiv S)$: (TR; $n_{f1} \times n_{f1}$) FW A Priori Normal Matrix of Fitted Parameters

Real World (RW)

n_{r1} : Number of unconstrained parameters

n_{Pr} : Number of permutations in P_r

P_r : (RR; n_{pr} x 1) RW Permutation List

n_{Kr} : Number of RW constraints

K_r : (RR; n_{kr} x 2) RW Constraint Partials List

$\Lambda_o (\equiv \Lambda)$: (TR; $n_{r1} \times n_{r1}$) RW A Priori Covariance Matrix of Unconstrained Parameters

Optional Miss Partitions

tm1	= 0	nominally	:	tail of first optional miss partition
hm1	= 6	nominally	:	head of first optional miss partition
tm2	= 0	nominally	:	tail of second optional miss partition
hm2	= 3	nominally	:	head of second optional miss partition
tm3	= 3	nominally	:	tail of third optional miss partition
hm3	= 6	nominally	:	head of third optional miss partition
tm4	= 6	nominally	:	tail of fourth optional miss partition
hm4	= 12	nominally	:	head of fourth optional miss partition
tm5	= 12	nominally	:	tail of fifth optional miss partition
hm5	= 18	nominally	:	head of fifth optional miss partition

tm6 - tail of sixth optional miss partition

hm6 - head of sixth optional miss partition

tm7 - tail of seventh optional miss partition

hm7 - head of seventh optional miss partition

tm8 - tail of eighth optional miss partition

hm8 - head of eighth optional miss partition

tm9 - tail of ninth optional miss partition

hm9 - head of ninth optional miss partition

A.2 PROGRAM CONSTANTS

Important Fixed Matrix Locations

$\mathcal{L}_{txo}$: Contains the location of X_o (Transformation Accumulator)

$\mathcal{L}_{ty}$: Contains the location of Y (Intermediate Storage)

$\mathcal{L}_{tz}$: Contains the location of Z (Input-Output Area)

Symbolic Addresses of Major Subroutines

GET X : The Partials Accumulator Subroutine

EXEC : The Executive Program

SIP : The Statistical Input Program

SOP : The Statistical Output Program

Output Coding Control Tables

T3 : (RR, 3; 26 x 1) The Matrix Format Table

T0 : (RR, 1; 6 x 1) Orbit-Set Option List

T2 : (RR, 2; 13 x 1) System-Set Option Table

T1 : (RR, 3; 33 x 1) Set-Partion Option Table

Update Controls

n_{UP} : The number of update commands

UPC : (RR, 5; n_{UP} x 1) Update Command List

Constant Matrices

Γ : (RR, $n_o \times n_o$) Orbit Set Partials Adjoining Matrix

$\dagger$: (RR, $n_p \times n_o$) Orbit Set Rows of the ($n_o \times n_o$) Identity Matrix

I : (NS) The ($n_o \times n_o$) Identity Matrix

A. 3 PROGRAM OUTPUTS

Sequence Controls

PH : Program Phase Flag. This flag indicates the computational phase of the program in accordance with the following coding:

Value of PH	Name of Phase
1	Update (Epoch Shift)
2	Print Output
3	Recycle
4	Tape Read-In

ι^* : Number of next Print Output Phase (Not valid if $\partial PM = 0$)

ℓ^* : Time index of next Print Output Phase, $\ell^* = PoT(\iota^*)$ (not valid if $\partial PM = 0$)

ι' : Number of next Recycle Phase

ℓ' : Time index of next Recycle Phase ($\ell^* = RET(\iota')$)

jT : Epoch index of next tape input record (k+1 st record)

Output Block Identification

j : Index of program epoch

ℓ : Index of last recycle time (time from the start of the tracking run)

k : Index of current time

INFO : Information associated with the k-th record

Output Entity Identification

ω : Output entity number

COM : Command defining the ω-th output entity. This set of locations is split up in the program into the following 7 quantities

σ_o : Matrix Option Index

Representation of the Columns of the Matrix Option $\left[\sigma_o\right]$:

μ_1 : Column Parity of $\left[\sigma_o\right]$

τ_1 : Orbit Set Option

ρ_1 : Set-Partition Option

Representation of the Rows of the Matrix Option $\left[\sigma_o\right]$:

μ_2 : Row Parity of $\left[\sigma_o\right]$

τ_2 : Orbit Set Option

ρ_2 : Set-Partition Option

Stored Partials. In the following locations are stored the partials of the indicated parameter sets with respect to the program representation x(= xj).

I : (RR, $n_p \times n_o$). $\partial p^j/\partial x$. Orbit set at program epoch

U : (RR, $n_p \times n_o$). $\partial p^k/\partial x$. Orbit set at current time

V : (RR, $n_p \times n_o$). $\partial p^\ell/\partial x$. Orbit set at last recycle time

W : (RR, $n_p \times n_o$). $\partial p^\iota/\partial x$. Orbit set at injection

B : (RR, $n_m \times n_o$). $\partial m/\partial x$. Miss set

Tracking Accuracy (from ℓ to k, referred to x)

J : (TR, $n_o \times n_o$). The "$A^T WA$" Normal Matrix

$\tilde{J}$: (TR, $n_o \times n_o$). The "$A^T WMWA$" Normal Matrix

A Priori Statistics (at ℓ, referred to x. Not valid if $\delta_{PM} = 0$)

S : (TR, $n_o \times n_o$). Fit World Normal Matrix (Extended)

$\bar{\Lambda}$: (TR, $n_o \times n_o$). Real World Covariance Matrix

Current Statistics (at k referred to x). The following symbols possess the indicated significance only if $\partial_{PM} = 1$.

δcs : Current Statistics Validity Flag. If $\delta cs = 1$, which occurs only if $k = \ell^*$ or $k = \ell'$, the following matrices contain valid information unless otherwise indicated.

S' : (TR; $n_o \times n_o$) Fit World Normal Matrix (Extended)

Λ' : (TR; $n_o \times n_o$) Fit World Covariance Matrix (not valid if $k = 0$)

$\bar{\Lambda}'$: (TR, $n_o \times n_o$) Real World Covariance Matrix

$\Sigma(Z)$: (RR, $n_o \times n_o$). Real World Tracking Noise Covariance Matrix (valid only if $\omega = 1$, PH = 2)

A.4 UTILITY LOCATIONS

Tracking Accuracy Tape Read-In (from t_{k-1} to t_k, referred to x)

s : Station Number

$\Delta J(\equiv S')$: (TR; $n_o \times n_o$) The "A^TWA" matrix for the s-th station

$\Delta\tilde{J}(\equiv\Lambda')$: (TR; $n_o \times n_o$) The "A^TWMWA" matrix for the s-th station

Partials Accumulator ($x = (\mu, \tau, \rho|)$)

μ : Parity of x

τ : Orbit Set Option of x

ρ : Set-Partition Option of x

$\mathcal{L}_{tx}$: The location of x

n_{1x} : The number of rows of x

X : (DM) (RR; $n_{1x} \times n_o$): The Partials Accumulator

X^T : (RR, $n_o \times n_{1x}$): Fixed locations for storing the transpose of x

Transformation Accumulator (X_o = <μ, τ, χ|

δ_x : Transformation Accumulator Validity Flag. If $\delta_x = 1$, X_o contains a valid transformation matrix defined by the indices μx, τx, χx

μ_x : Parity of X_o

τ_x . Orbit Set Option of X_o

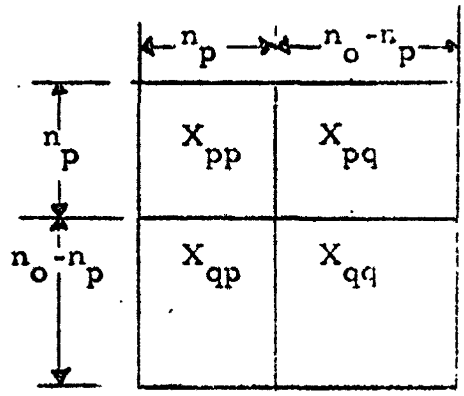
χ_x System Option of X_o

X_o . (RR, $n_o \times n_o$) The Transformation Accumulator

Natural Partitioning of X_o

X_{pp}
X_{pq}
X_{qp}
X_{qq}

These symbols represent the indicated partitioning of the Transformation Accumulator X_o:



X_{po} (X_{pq}, X_{qq})

The Input-Output Area (z)

δ_{ZI} · The Identity Flag. If $\partial_{zI} = 1$ the z locations are created by the Executive Program as though z contained the $(n_o \times n_o)$ identity I.

n_{1z} : The number of rows of z

n_{2z} · The number of columns of z

Z · (RR, $n_o \times n_o$) The Input-Output Area

Intermediate Storage Area (Y)

n_{1y} · The number of rows of Y

n_{2y} : The number of columns of Y

Y : (RR, $n_o \times n_o$) The Intermediate Storage Area

The Row-Transformed Matrix ($\overline{Y}$)

$\mathcal{L}_{t\overline{y}}$: The location of $\overline{Y}$

$n_{1\overline{y}}$: The number of rows of $\overline{Y}$

$\overline{Y}$: (DM) (RR, $n_{1\overline{y}} \times n_o$). The Row Transformed Matrix

Miscellaneous. Most of the following symbols represent locations used by the table lookup routines (see Appendix B)

σ : The symbolic input to T3

F : The format of a matrix

$\mathcal{L}_t$: The first (or in some cases one minus the first) location of a matrix

Δ_1 : The number of rows of a matrix

Δ_2 : The number of columns of a matrix

χ : The System Index ($\chi = 0$ indicates an isolated set)

n : For $\chi \neq 0$: the number of links required to chain-rule from $x^{T,\chi}$ to the standard representation x

t : One minus the first location of a parameter set partition (the tail of a partition)

h : The last location of a parameter set partition

ξ : The Orbit Set Partition Type

$$\xi = 0 \longleftrightarrow t = 0,\ h = n_p$$

$$\xi = 1 \longleftrightarrow t = n_p,\ h = 2n_p$$

$$\xi = 2 \longleftrightarrow t = 2n_p,\ h = 3n_p$$

a : The row index of a matrix entry

b : The column index of a matrix entry

APPENDIX B. OUTPUT OPTION CONTROLS

Table B-1. Matrix Format (T3)

T3

INPUT	T3(σ)			
σ	*F	*t	$^*\Delta_1$	$^*\Delta_2$
BCI	PZE		PZE	
X_o	RR	$\mathcal{L}_{txo}$	n_o	n_o
Y	RR	$\mathcal{L}_{t_y}$	n_{1y}	n_{2y}
Z	RR	$\mathcal{L}_{tz}$	n_{1z}	n_{2z}
X	RR	$\mathcal{L}_{tx}$	n_{1x}	n_o
$\overline{Y}$	RR	$\mathcal{L}_{t\overline{y}}$	$n_{1\overline{y}}$	n_o
P_c	IP	$P_c{}^{-1}$	n_{Pc}	$\underline{1}$
K_c	IK	$K_c{}^{-1}$	n_{Kc}	$\underline{2}$
K_f	IK	K_f 1	n_{Kf}	$\underline{2}$
P_r	IP	$P_r{}^{-1}$	n_{Pr}	$\underline{1}$
K_r	IK	$K_r{}^{-1}$	n_{Kr}	$\underline{2}$
S_o	TR	S •	n_{f1}	n_{f1}

Note: (*) indicates indirect storage
$\underline{a}$ indicates an address where a is stored.

UTILITY MATRICES (Fixed $\mathcal{L}_t$)

The Transformation Accumulator

Intermediate Storage

Input-Output Storage

DUMMY MATRICES (Variable $\mathcal{L}_t$)

The Partials Accumulator

SYSTEM LINKS (These matrices represent, indirectly, $n_o \times n_o$ transformations)

Common Permutations	$\partial x^{\tau,2}/\partial x^{\tau,1}$
Common Constraints	$\partial x^{\tau,3}/\partial x^{\tau,2}$
Fit World Permutations	$\partial x^{\tau,4}/\partial x^{\tau,3}$
Real World Permutations	$\partial x^{\tau,5}/\partial x^{\tau,3}$
Real World Constraints	$\partial x^{\tau,6}/\partial x^{\tau,5}$

A PRIORI STATISTICS

FW Normal Matrix of x^{f1}

Table B-1. Matrix Format (T3) - Continued

T3

INPUT	T3(σ)				
σ	*F	*t	*Δ_1	*Δ_2	
BCI	PZE		PZE		
					A PRIORI STATISTICS (Continued)
$\bar{\Lambda}_o$	TR	$\bar{\Lambda}$	n_{r1}	n_{r1}	RW Covariant Matrix of x^{r1}
					AVAILABLE PARTIALS
I	RR	I	n_p	n_o	Orbit Set at Epoch $\partial p^j/\partial x$
U	RR	U	n_p	n_p	Orbit Set at Current Time $\partial p^k/\partial x$
V	RR	V	n_p	n_o	Orbit Set at Last Recycle Time $\partial p^\ell/\partial x$
W	RR	W	n_p	n_o	Orbit Set at Injection Time $\partial p^\iota/\partial x$
B	RR	B	n_m	n_o	Miss Set
					MATRIX OPTIONS
I	NS		n_o	n_o	Identity Matrix
					Tracking Accuracy (from ℓ to k)
J	TR	J	n_o	n_o	"$A^T WA$"
$\tilde{J}$	TR	$\tilde{J}$	n_o	n_o	"$A^T WMWA$"

Note: (*) indicates indirect storage
$\underline{a}$ indicates an address where a is stored.

Table B-1. Matrix Format (T3) - Continued

T3

INPUT	T3(σ)			
σ	*F	*t	$*\Delta_1$	$*\Delta_2$
BCI	PZE		PZE	
S	TR	S	n_o	n_o
$\bar{\Lambda}$	TR	$\bar{\Lambda}$	n_o	n_o
S'	TR	S'	n_o	n_o
Λ'	TR	Λ'	n_o	n_o
$\bar{\Lambda}'$	TR	$\bar{\Lambda}'$	n_o	n_o
Σ	TR	$\mathscr{L}_{tz}$	n_o	n_o

Note: (*) indicates indirect storage
<u>α</u> indicates an address where α is stored.

<u>Statistics at *l*</u>

FW Normal Matrix

RW Covariant Matrix

<u>Statistics at k (Current Statistics)</u>

FW Normal Matrix

FW Covariant Matrix

RW Covariant Matrix

RW Tracking Noise Covariant Matrix

Table B-2. Orbit-Set Options (T0)

T0	τ	T0(τ): $\mathcal{L}_t$	T0(τ): ξ	
		PZE		
				ORBIT PARAMETERS AT:
	0	I	0	Epoch (t_j) (This entry is not needed in the program)
	1	U	0	Current Time (t_k)
	2	V	0	Last Recycle Time (t_ℓ)
	3	W	0	Injection Time (t_i)
				TARGET PARAMETERS (for present TAPP I miss)
	4	B	1	2nd Set of n_p Miss Parameters (UDC)
	5	B	2	3rd Set of n_p Miss Parameters (DCA)

Table B-3. System-Set Options

	INPUTS		$\leftarrow T2(\chi, n) \rightarrow$	
T2	χ	n	σ	
	PZE		BCI	
				INTERMEDIATE COMMON SYSTEM (not in T1)
	2	1	P_c	$\partial x^{T,2} / \partial x^{T,1}$
				COMMON SYSTEM
	3	2	P_c	$\partial x^{T,2} / \partial x^{T,1}$
	3	1	K_c	$\partial x^{T,3} / \partial x^{T,2}$
				FIT SYSTEM
	4	3	P_c	$\partial x^{T,2} / \partial x^{T,1}$
	4	2	K_c	$\partial x^{T,3} / \partial x^{T,2}$
	4	1	K_f	$\partial x^{T,4} / \partial x^{T,3}$
				INTERMEDIATE REAL SYSTEM (not in T1)
	5	3	P_c	$\partial x^{T,2} / \partial x^{T,1}$
	5	2	K_c	$\partial x^{T,3} / \partial x^{T,2}$
	5	1	P_r	$\partial x^{T,5} / \partial x^{T,3}$
				REAL SYSTEM
	6	4	P_c	$\partial x^{T,2} / \partial x^{T,1}$
	6	3	K_c	$\partial x^{T,3} / \partial x^{T,2}$
	6	2	P_r	$\partial x^{T,5} / \partial x^{T,3}$
	6	1	K_r	$\partial x^{T,6} / \partial x^{T,5}$

Table B-4. Set Partition Options (T1)

T1	INPUT	T1(ρ)				
	ρ	χ	n	*t	*h	
	BCI	PZE		PZE		Note: (*) indicates indirect storage $\underline{a}$ indicates an address where a is stored
						SYSTEM PARTITIONS
						<u>Original System</u> (TAPP I)
	o	1	1	$\underline{0}$	n_o	Complete Set
	p	1	1	$\underline{0}$	n_p	Orbit · · ameters
	q	1	1	n_p	n_o	Constants
						<u>Command System</u>
	c	3	3	$\underline{0}$	n_{c1}	Complete Set (cp: cq: c2)
	c1	3	3	$\underline{0}$	n_{c1}	Common Parameters (cp: cq)
	cp	3	3	$\underline{0}$	n_p	Orbit Parameters
	cq	3	3	n_p	n_{c1}	Common Constants
	c2	3	3	n_{c1}	n_o	Eliminated Constants
						FIT SYSTEM
	f	4	4	$\underline{0}$	n_o	Complete Ste
	f0	4	4	$\underline{0}$	n_{c1}	Fit-Consider System (fp: fq: fz)
	f1	4	4	$\underline{0}$	n_{f1}	Fitted Parameters (fp: fq)

Table B-4. Set Partition Options (T1) - Continued

T1	INPUT	T1(ρ)				
	ρ	χ	n	*t	*h	Note: (*) indicates indirect storage $\underline{\alpha}$ indicates an address where α is stored
	BCI	PZE		PZE		
						FIT SYSTEM (Continued)
	fp	4	4	$\underline{0}$	n_p	Orbit Parameters (TAPP 2 "x")
	fq	4	4	n_p	n_{f1}	Fitted Constants (TAPP 2 "y")
	f2	4	4	n_{f1}	n_{c1}	Considered Parameters (TAPP 2 "z")
						Real System
	r	6	5	$\underline{0}$	n_o	Complete Set
	r0	6	5	$\underline{0}$	n_{c1}	Uncertain-Certain System (rp: rq: r2)
	r1	6	5	$\underline{0}$	n_{r1}	Uncertain Parameters
	rp	6	5	$\underline{0}$	n_p	Orbit Parameters
	rq	6	5	n_p	n_{r1}	Uncertain Constants
	r2	6	5	n_{r1}	n_{c1}	Certain Constants
						ISOLATED SETS
						Orbit Sets
	pj	0	1	$\underline{0}$	n_p	At Epoch
	pk	0	U	$\underline{0}$	n_p	At Current Time

Table B-4. Set Partition Options (T1) - Continued

	INPUT	T1(ρ)				
T1	ρ	χ	n	*t	*h	
	BCI	PZE		PZE		Note: (*) indicates indirect storage $\underline{a}$ indicates an address where a is stored
	p_ℓ	0	V	$\underline{0}$	n_p	At Last Recycle Time
	p_ι	0	W	$\underline{0}$	n_p	At Injection
						Miss Set Partitions
	m	0	B	$\underline{0}$	n_m	Complete Set
	m1	0	B	t_{m1}	h_{m1}	1st Optional Partition
	m9	0	B	t_{m9}	h_{m9}	9th Optional Partition

Table B-5. Permanently Stored Commands

ω	σ_o BCI	μ_1 PZE	τ_1	ρ_1 BCI	μ_2 PZE	τ_2	ρ_2 BCI
ω	UPC(ω)						
1	I	0	-	pℓ	1	1	o
2	I	0	-	pℓ	1	1	o
3	I	0	-	m	1	1	o
4	J	1	1	o	1	1	o
5	$\tilde{J}$	1	1	o	1	1	o
6	S	1	1	o	1	1	o
$n_{UP} = 7$	$\overline{\Lambda}$	0	1	o	0	1	o
ω	ROC(ω)						
1	I	0	-	pk	1	2	fp
2	I	0	-	pk	1	2	fq
3	I	0	-	pk	1	2	f2
4	J	1	2	f1	1	2	f1
5	$\tilde{J}$	1	2	f1	1	2	f1
6	J	1	2	f1	1	2	f2
7	I	0	-	m1	1	1	fp
8	I	0	-	m1	1	1	fq
$n_{Ro} = 9$	I	0	-	m1	1	1	f2

APPENDIX C. THE FLOW DIAGRAMS

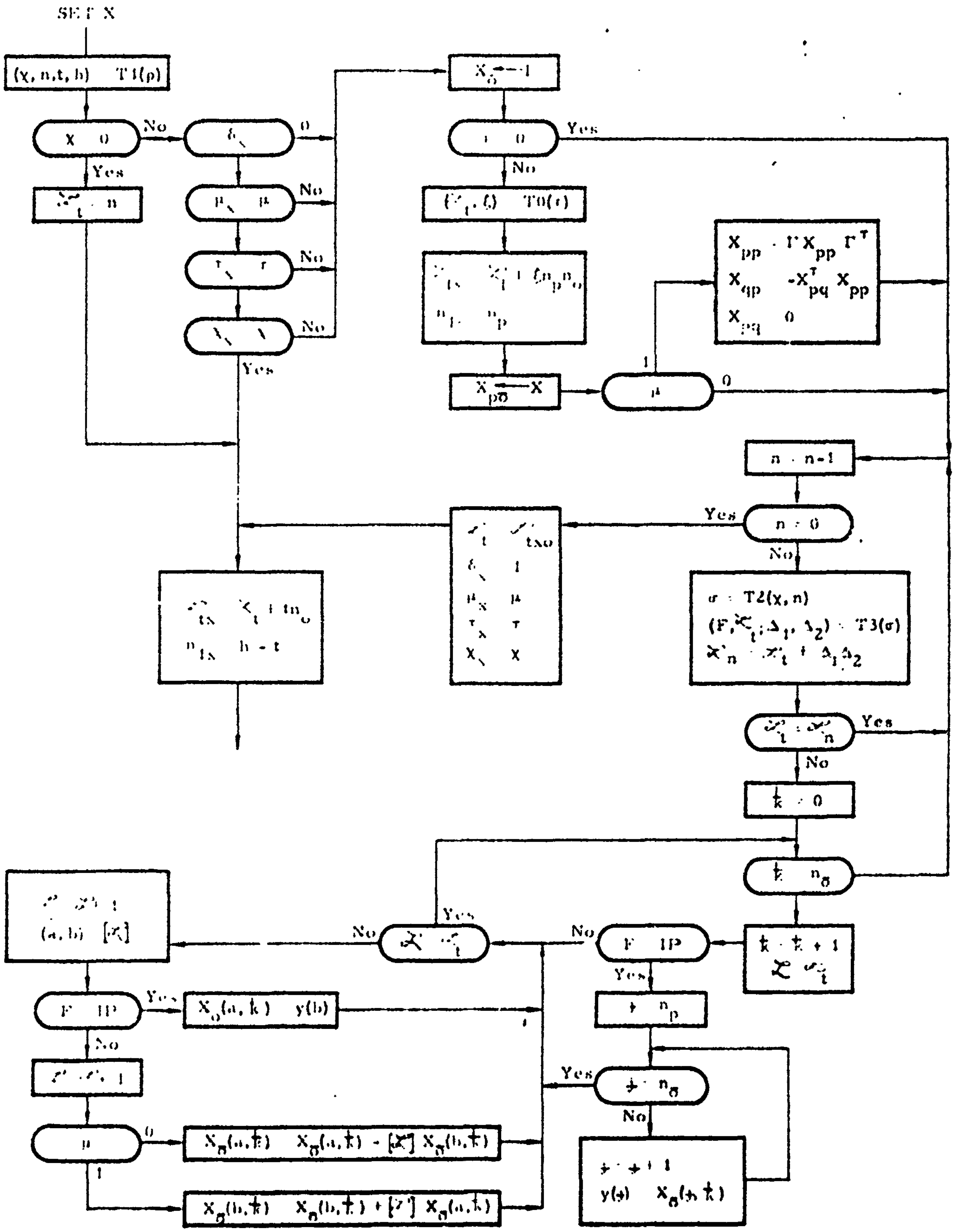


Figure C-1. Partials Accumulator Subroutine

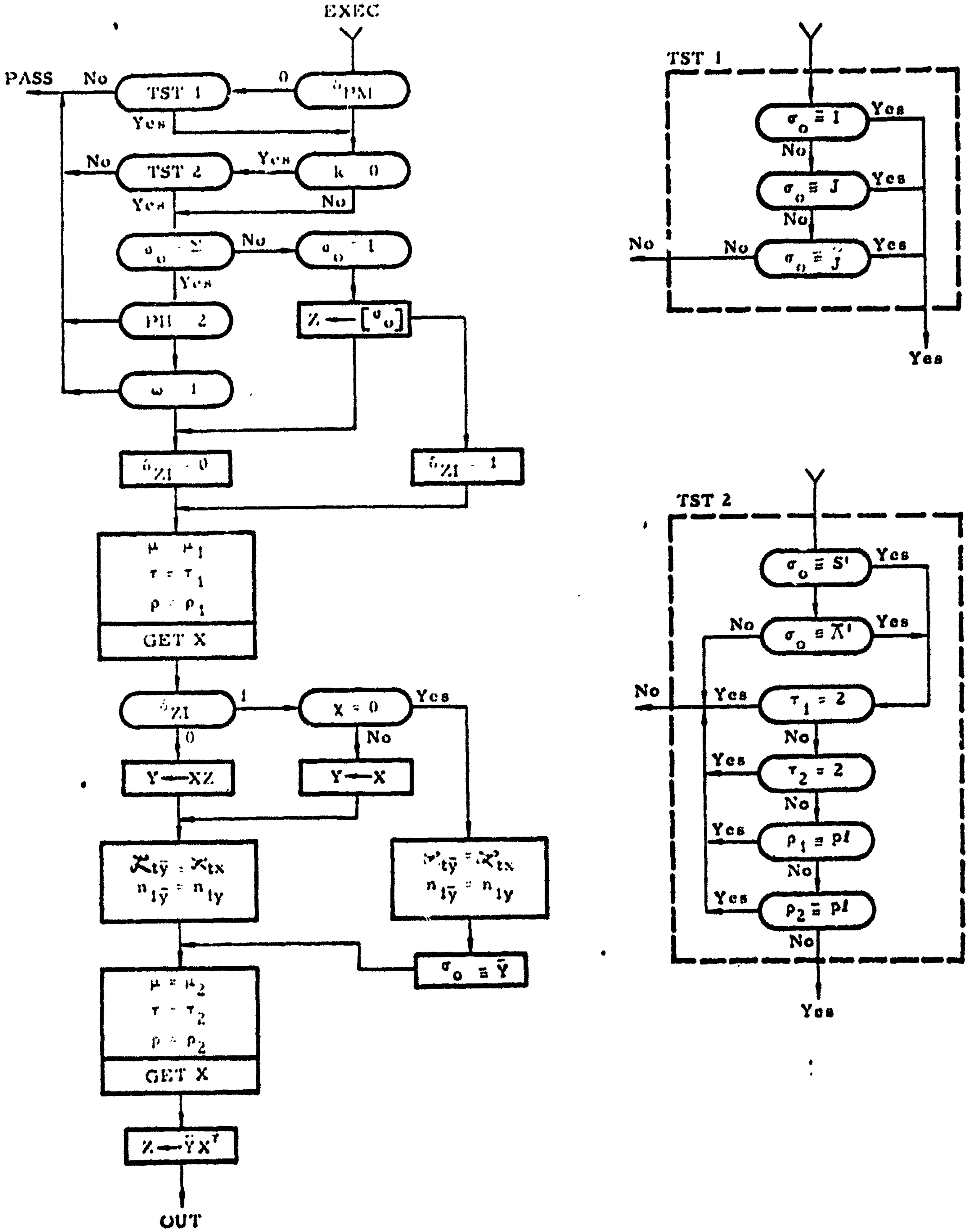


Figure C-2. The Executive Program

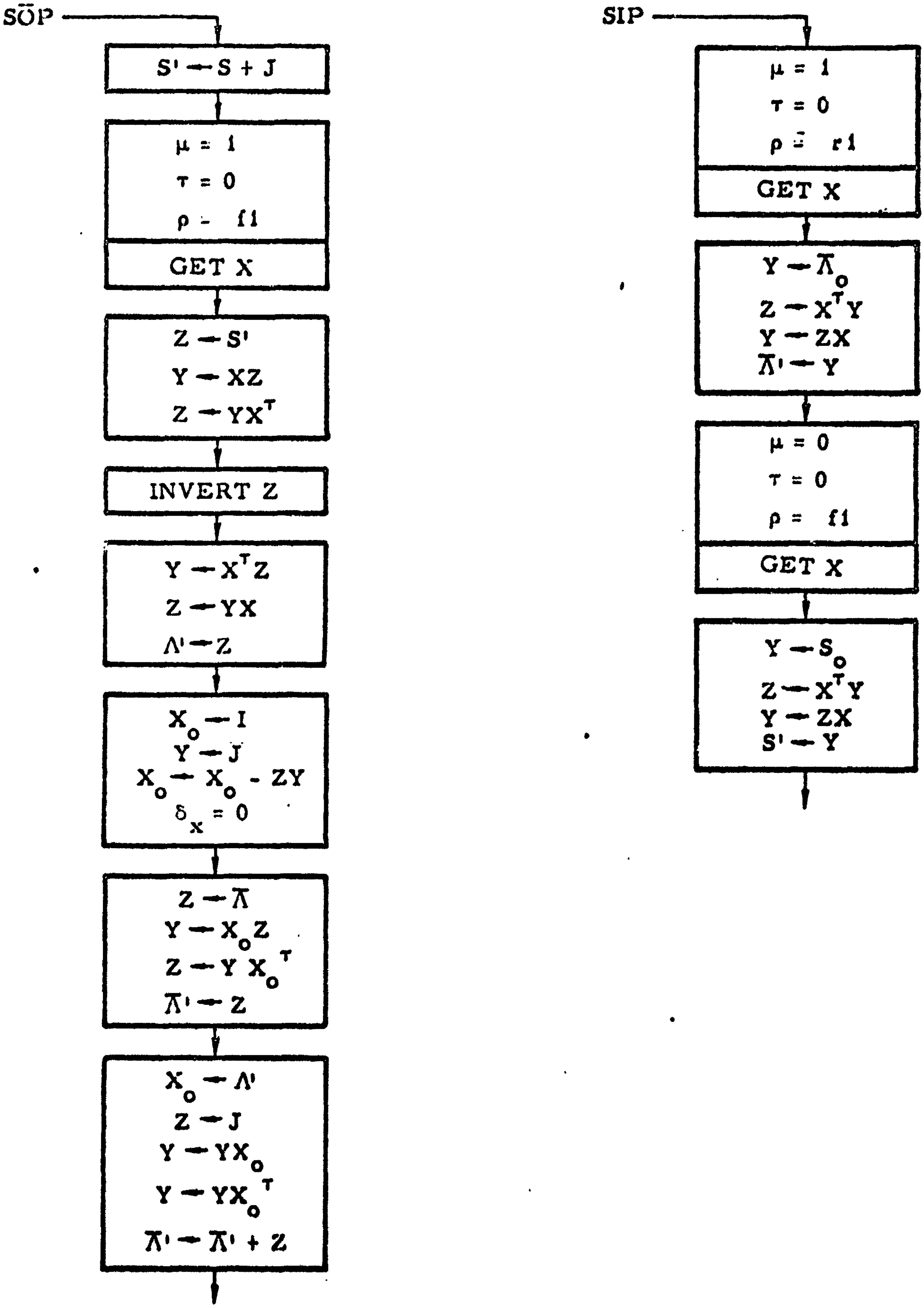


Figure C-3. The Statistical Output Program

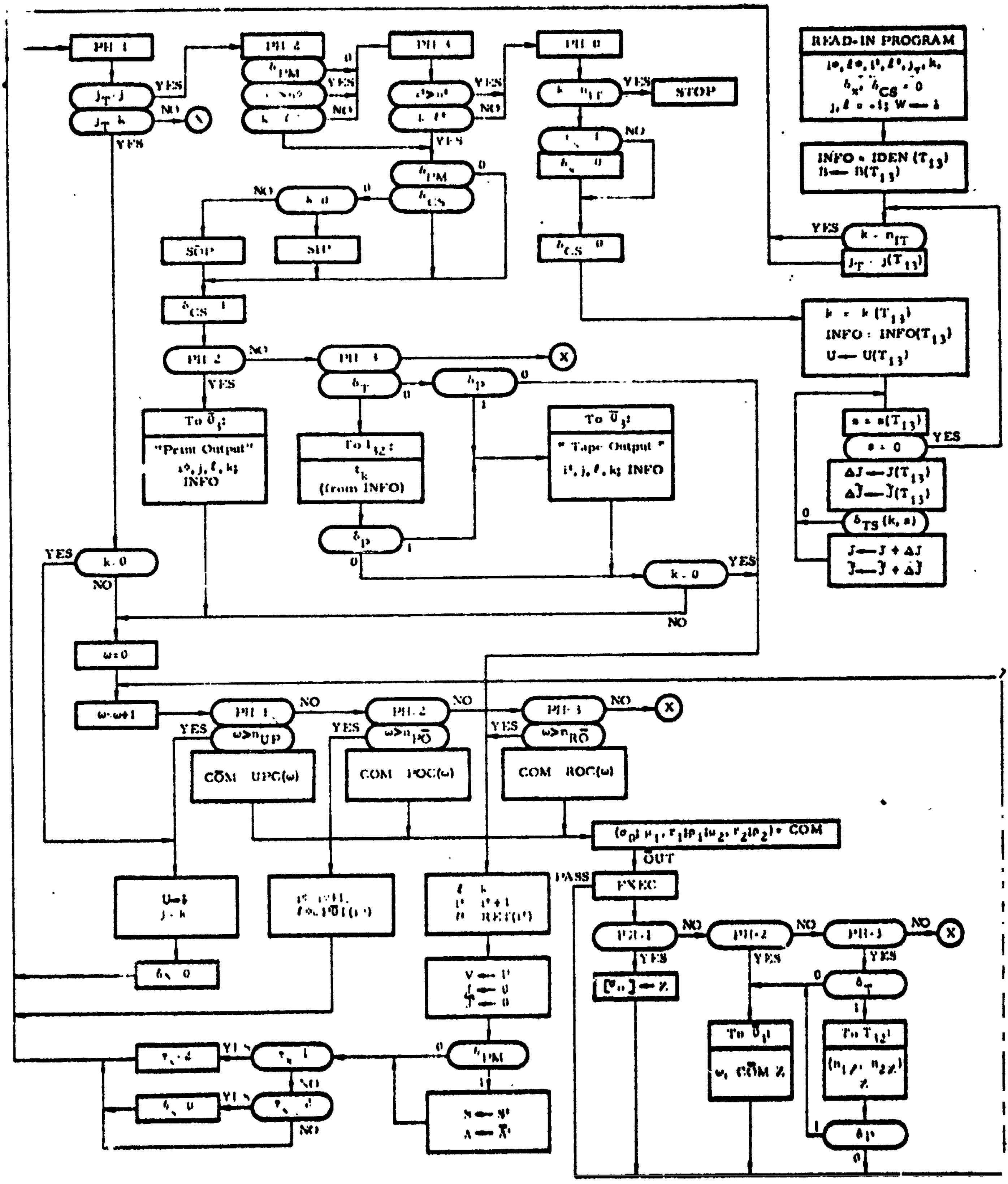
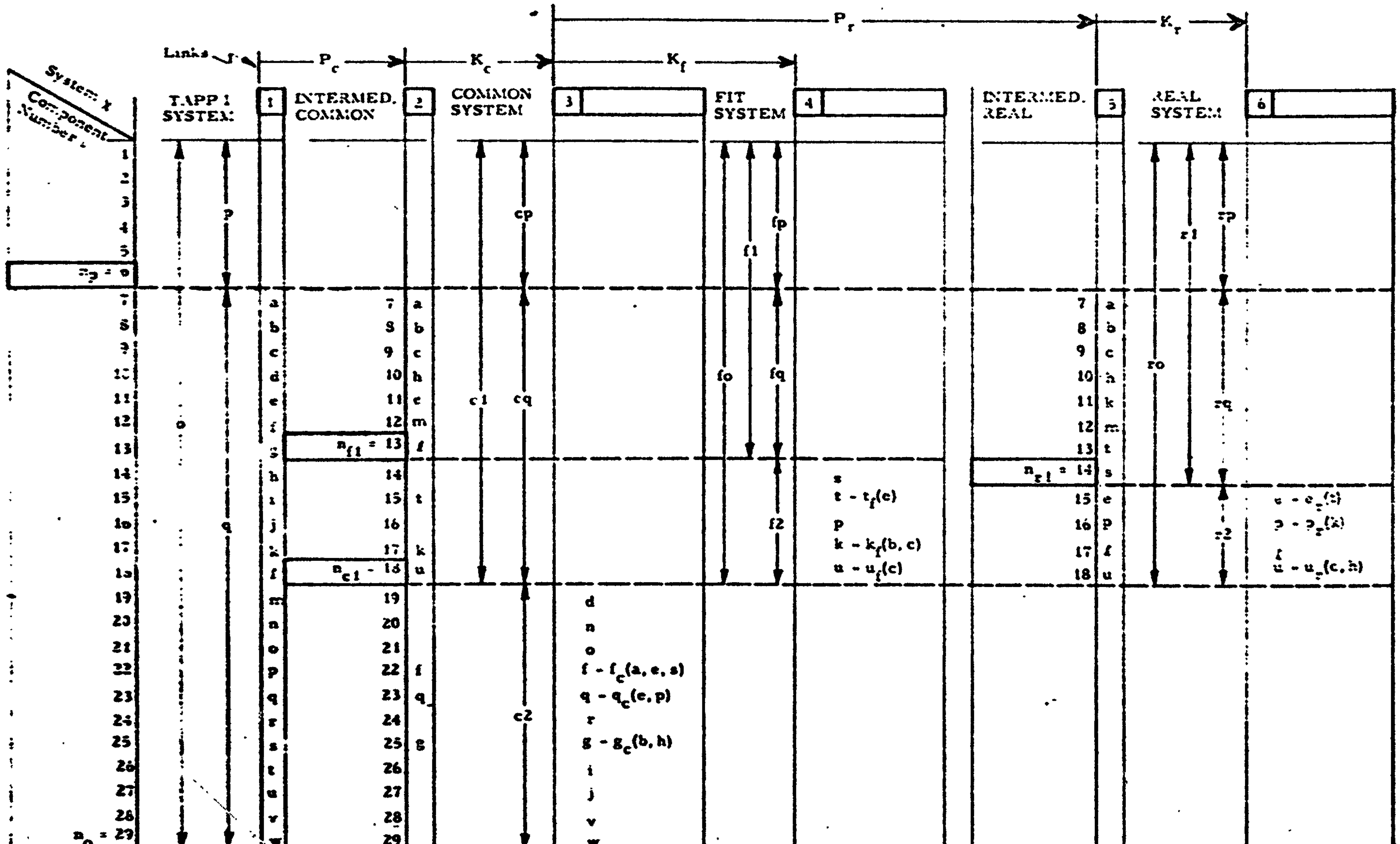


Figure C-4. TAPP III Master Sequencing

APPENDIX D. THE CODING OF INPUT SYSTEMS AND PARTITIONS



Note: The permissable set-partitions are shown for built-in Options Table T1.

Figure D-1. Example of an Ordering and Constraining of Parameters in TAPP III

Table D-1. Coding of the Information in Figure D-1

Tape Description

n_o = 29

n_p = 6

Common Assumptions

n_{c1} = 18

n_{Pc} = 12

P_c = 10, 14
12, 19
13, 18
14, 25
15, 26
16, 22
18, 19
19, 10
22, 12
25, 13
26, 15
27, 16

n_{kc} = 7

K_c = 22, 7
∂fc/∂a
22, 11
∂fc/∂e
22, 14
∂fc/∂s
23, 11
∂qc/∂e
23, 16
∂qc/∂p
25, 8
∂gc/∂b
25, 10
∂gc/∂b

Fit World

n_{f1} = 13

n_{Kf} = 4

Kf = 15, 11
∂tf/∂e
17, 8
∂kf/∂b
17, 9
∂kf/∂c
18, 9
∂uf/∂c

Real World

n_{r1} = 14

n_{Pr} = 4

P_r = 11, 17
13, 15
15, 11
17, 13

n_{kr} = 4

K_r = 15, 13
∂er/∂t
16, 11
∂pr/∂k
18, 9
∂ur/∂c
18, 10
∂ur/∂h