TECHNICAL NOTE NO. 1
Project No. A-588

INTRODUCTION TO NONSTATIONARY STOCHASTIC
PROCESSES FOR ANALOG MONTE CARLO STUDIES

By R. S. Johnson
and R. D. Loftin

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473
(Development of New Methods and
Applications of Analog Computation)

12 November

Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
INTRODUCTION TO NONSTATIONARY STOCHASTIC PROCESSES FOR ANALOG MONTE CARLO STUDIES

By
R. S. Johnson
and
R. D. Loftin

TECHNICAL NOTE NO. 1
on
Contract No. NAS8-2473
(Development of New Methods and Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama

Approved:

F. Dixon
Project Director A-588, and
Head, Special Problems Branch
Georgia Tech Research Project No. A-588 was established under Contract NAS8-2473 on 12 September 1961 to assist the Flight Simulation Branch of the Computation Division at Marshall Space Flight Center in the investigation and development of new methods and applications of analog computation within certain specified areas of interest. Efforts during the past year have been devoted to (I) design and construction of an experimental electronic generalized integrator suitable for use with existing analog computer equipment, and (II) investigation of techniques for generating nonstationary noise voltages to be used in analog Monte Carlo studies. The present report completes the work performed under the latter task. Mr. Robert S. Johnson (formerly Project Director A-588 and Supervisor of the Georgia Tech Analog Computer Laboratory) is now a staff member at the General Electric Company's Huntsville Computer Center. Mr. Ralph D. Loftin, upon completing his M.S. degree in E.E., accepted employment with the Minneapolis-Honeywell Company in Clearwater, Florida.

F. D.
This Technical Note is intended as a brief introduction to nonstationary stochastic processes from the point of view of the analog computer engineer. It is assumed that the reader is familiar with stationary processes and the manner in which they are generated and used in analog computation.

Chapter I presents the principal terms, definitions, and conventions adopted throughout the report and discusses certain important functionals of nonstationary processes. Chapter II is devoted to a particular nonstationary functional--the power spectral density function; in this chapter, the various definitions of power spectra are compared and the usefulness of the spectral concept is discussed. Chapter III deals with a particularly useful class of processes and the way in which they are generated. The Technical Note is concluded in Chapter IV with an example of process synthesis using analog computer components.
Chapter I

TERMS, SYMBOLS, AND CONVENTIONS

A Glossary of the principal terms and symbols employed in this report is presented as Appendix A (page 11 below). For the most part the Glossary is self-explanatory, although some comment is in order concerning conflicting definitions by various authors.

Apart from the power spectral density function, to which Chapter II is devoted, the covariance and correlation functions seem to have the widest range of definitions throughout the literature. As defined in this report, the covariance function of a process \( X(t) \) is

\[
C_X(t,T) = E\{[X(t) - \mu_X(t)][X(t+T) - \mu_X(t+T)]\}
\]

where \( \mu_X(t) = E\{X(t)\} \) is the process mean, or "expected value" of \( X \). We define a related functional, the normalized covariance function, as

\[
\rho_X(t,T) = \frac{C_X(t,T)}{\sigma_X(t) \sigma_X(t+T)},
\]

where

\[
\sigma_X^2(t) = E\{[X(t) - \mu_X(t)]^2\} = C_X(t,0)
\]

is the process variance at the instant \( t \). The time-average statistic roughly corresponding to (1.1) is the correlation function:

\[
R_X(T) = \lim_{a \to \infty} \frac{1}{2a} \int_{-a}^{a} X(t)X(t+T) \, dt
\]

The functionals defined in Equations (1.1), (1.2), and (1.4) are given a wide variety of conflicting names by various authorities. Blackman and Tukey\(^1\), for example, use the term "autocovariance function" to describe (1.1) and "autocorrelation function" to describe (1.2). Similar terms are used by Mood\(^2\). Schwartz\(^3\) refers to (1.4) as the "time autocorrelation" and calls \( E\{X(t)X(t+T)\} \) the "autocorrelation." Bendat\(^4\) uses "autocovariance function" for (1.1) and concurs with Schwartz in the use of "autocorrelation." Goldman\(^5\), Bell\(^6\), Middleton\(^7\), and Stewart\(^8\) adopt the convention of
this report in using correlation function to mean (1.4) and covariance function to denote (1.1), but Middleton calls (1.2) the "correlation coefficient." This latter term (sometimes "coefficient of correlation") is also used by Wadsworth and Bryan, Laning and Battin, and Bartlett. Davenport and Root likewise adopt this term, but mention that "normalized covariance function" is preferred by many authors.

References, Chapter I

Chapter II
THE NONSTATIONARY POWER SPECTRUM

The power spectral density function (sometimes called the power spectrum) is a useful functional when the stochastic process in question is "wide-sense stationary"—i.e., when the covariance function is invariant with the instant of observation (t) and depends only on the time shift (T). The utility of the concept is due mainly to the simple relationship which exists between input and output spectra of linear networks and to the intuitive interpretation of "power in narrow bands." In addition, ergodicity is usually assumed, so that the Wiener-Khintchine relations permit easy computation of the power spectrum from the covariance function. Thus, for stationary processes,

\[ \phi_X(\omega) = \int_{-\infty}^{\infty} C_X(T) e^{-j\omega T} dT. \]

In the case of nonstationary processes, most of the conveniences of the power spectral concept are lost. Application of the usual defining equation,

\[ \phi_X(\omega) = \lim_{T \to \infty} \left[ \frac{1}{2T} \int_T^{T+2T} X(t) e^{-j\omega t} dt \right]^2, \]

now yields a random variate; moreover, all reference to the inherent time dependence of a nonstationary process is lost. If the Wiener-Khintchine relation (2.1) is used to define the nonstationary spectrum, the resulting expression contains an imaginary component which is very difficult to interpret. Lastly, the relationship between input and output power spectra for linear networks driven by nonstationary noise involves a convolution integral which is very unwieldy and impossible to evaluate in closed form for many cases of interest. If X is the input and Y the output, the expression is

\[ \phi_Y(\omega, t) = H(-j\omega) \int_0^\infty h(a) \phi_X(\omega, t-a) e^{j\omega a} da, \]

where \( H(j\omega) \) is the network system function and \( h(t) \) the impulse response.\(^1\,2\)

Several authorities—notably Page\(^3\), Lampard\(^4\), Turner\(^5\), Kharkevich\(^6\), and Silverman\(^7\)—have attempted to develop the concept of an instantaneous power spectrum. We will herein discuss Page's basic paper and briefly mention the extensions thereof by Lampard, Turner, and Kharkevich. Silverman's approach is somewhat different and his paper will be considered separately.
Page develops an expression for the instantaneous power spectrum by considering the time rate of change of an instantaneous energy spectrum. The energy spectrum is obtained by taking the squared magnitude of a "running" transform, defined as

\begin{equation}
G_t(\omega) = \int_{-\infty}^{t} G(\alpha) e^{-j\omega\alpha} d\alpha,
\end{equation}

where \( G(t) \) is the random process under consideration. The functional \( |g_t(\omega)|^2 \) is considered to be a measure of the energy density of \( G(t) \) at the frequency \( \omega \) up to the present instant, \( t \). The instantaneous power spectrum, \( \psi \), is obtained by taking the partial derivative with respect to time of \( |g_t(\omega)|^2 \). This operation yields

\begin{equation}
\psi(t,\omega) = \frac{\partial}{\partial t} |g_t(\omega)|^2,
\end{equation}

which can be written

\begin{equation}
\psi(t,\omega) = 2 \int_{0}^{t} G(t) G(t-T) \cos \omega T \, dT.
\end{equation}

Note that the spectrum defined by (2.6) is valid only for signals which contain finite energy because of the infinite integration limit in (2.4). Further, as Page implies, the only stochastic processes to which the instantaneous spectrum applies are those which can be considered to be switched on at \( t = 0 \). To obtain the instantaneous spectrum for this type of process, an ensemble average of (2.6) is taken. This yields

\begin{equation}
E\{\psi(t,\omega)\} = 2 \int_{0}^{t} \mathbb{E}\{G(t)G(t-T)\} \cos \omega T \, dT,
\end{equation}

where the upper limit is now \( t \) since the integrand is identically zero for \( T > t \). This expression may also be written

\begin{equation}
E\{\psi(t,\omega)\} = 2 \int_{0}^{t} C_G(T) \cos \omega T \, dT,
\end{equation}

where \( C_G(T) \) is the covariance function for the process \( G(t) \). It is now clear that Page's instantaneous spectrum does not apply to nonstationary processes in general, but only to ergodic processes which can be considered to be switched on at \( t = 0 \). Thus, Page's spectrum given in (2.8) actually illustrates the development in time of the spectrum of an ergodic process.
The work by Lampard is a straightforward extension of the work by Page. Lampard begins with the spectrum given in (2.8) and develops the inversion integral. The results are essentially as follows:

\begin{align}
(2.9a) \quad \phi(t,\omega) &= 2 \int_{0}^{\infty} C(t,T) \cos \omega T \, dT , \\
(2.9b) \quad C(t,T) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(t,\omega) \cos \omega T \, d\omega ,
\end{align}

where the symbol \( C(t,T) \) is used in lieu of the finite limit of integration.

Turner demonstrates that the spectrum obtained from (2.8) is not unique for a given time signal. He argues on the basis that two observers of the same time signal may not begin their measurements at the same instant. Thus, Turner shows that the difference in the spectra recorded by these two observers takes the form of a complementary spectral function with the property that the integral of this function over all frequency is identically zero.

Kharkevich, while working along lines parallel to those of Page, does not restrict himself to functions which are switched on at a particular time. He develops the concept of a mean spectrum and a mean correlation (covariance) function. The resulting Fourier Cosine Transform pair are cited here for reference:

\begin{align}
(2.10a) \quad A\{\phi(t,\omega)\} &= \frac{2}{\pi} \int_{0}^{\infty} A\{C_X(t,T)\} \cos \omega T \, dT , \\
(2.10b) \quad A\{C_X(t,T)\} &= \int_{0}^{\infty} A\{\phi(t,\omega)\} \cos \omega T \, d\omega ,
\end{align}

where

\[ C_X(t,T) = E\{X(t)X(t+T)\} . \]

The primary objection to the spectrum defined above is the fact that the time dependence is lost in the time-averaging process. It would seem reasonable to expect the spectrum of a nonstationary process to retain the time dependence inherent in the nonstationarity.

Silverman has developed the concept of power spectra as applied to a particular class of nonstationary processes--the so-called "locally

*Lampard's equations appear in a slightly different but equivalent form due to his conventions regarding negative frequency.*
stationary" processes. These are processes whose covariance function can be written in the form

\[ C_X(t,t') = E\{X(t)X(t')\} = C_1(\frac{t+t'}{2}) C_2(t-t') , \]

where \( C_1() \) is a non-negative function and \( C_2(t-t') \) is a stationary covariance function--i.e., is independent of the time origin. Processes having a covariance function of the form (2.11) are said to be locally stationary in the wide sense, or have a locally stationary covariance. If \( X(t) \) is a locally stationary process, then by the generalized spectral density is meant the double Fourier Transform of the locally stationary covariance--viz.,

\[ \phi(\omega,\omega') = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C_X(t,t') e^{-j(\omega t - \omega' t')} dt \, dt' . \]

Silverman proves that if \( X(t) \) is a locally stationary process having spectral density given by (2.12), then the spectral density function itself can be written as a locally stationary covariance--that is,

\[ \phi(\omega,\omega') = \phi_1(\frac{\omega+\omega'}{2}) \phi_2(\omega-\omega') , \]

where

\[ \phi_1(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C_2(\lambda) e^{-j\omega\lambda} d\lambda , \]

and

\[ \phi_2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C_1(\lambda) e^{-j\omega\lambda} d\lambda . \]

These last two expressions, along with the obvious inversions, are called the generalized Wiener-Khintchine relations. It is readily seen that the covariance functions and spectra defined above reduce to the usual forms when the process is stationary.

It should be noted that any spectrum which results from an operation on the covariance function will suffer from two major disadvantages. First, unless artificial precautions are taken, the spectrum is likely to have imaginary components. In almost every case the imaginary component reduces to zero when averaged in time or when integrated over all frequency, but
this does not alleviate the difficulty in interpretation. Second, as previously seen, no convenient relationship exists between the spectra of the input and output of linear networks. This would seem to present almost insurmountable problems of analysis and synthesis. For these and other reasons, one might consider using the covariance function in lieu of the spectral density function as a process specification in Monte Carlo studies. This would bypass many problems of interpretation while not sacrificing any information, since all the information contained in the spectrum is also contained in the covariance function.

References, Chapter II


Chapter III
STATIONOID PROCESSES AND THE SEPARATED-NETWORK TECHNIQUE

The problem of generating an arbitrary nonstationary process is closely tied to the problem of synthesizing general time-varying networks. This latter task has been discussed by many authorities\(^1\)\(^-\)\(^3\) and the difficulties are recognized as being almost insurmountable. However, a wide class of useful processes may be generated by means of "separated networks" comprising two cascaded sections, one of which is invariant and the other of which is nonreactive or "zero-memory." The class of processes to which this type of network is appropriate is developed below.

With reference to Figure 3-1, the input process, \(X\), is considered to be zero-mean Gaussian white noise, such as that produced by the standard noise generator, with a flat power spectral density of height \(\varphi_X\) extending well beyond the cutoff frequency of the network \(N\). This input is stationary and \(N\) is time invariant; hence \(Y\) is zero-mean, Gaussian, and stationary, with a well-defined power spectrum given by

\[
\varphi_Y(\omega) = |N(j\omega)|^2 \varphi_X
\]

and covariance function

\[
C_Y(T) = \frac{1}{2\pi} \int_0^\infty \varphi_Y(\omega) \cos \omega T \, d\omega = \frac{1}{2\pi} \varphi_X \int_0^\infty |N(j\omega)|^2 \cos \omega T \, d\omega.
\]

Network \(K\), which might be a standard function multiplier, produces from \(Y\) the output process \(Z\):

\[
Z(t) = K(t) Y(t).
\]

Figure 3-1: "Separated-Network" Diagram
The covariance function of $Z$ is then

$$C_Z(t,T) = \mathbb{E}\{K(t) Y(t) K(t+T) Y(t+T)\} = K(t) K(t+T) C_Y(T),$$

and the normalized covariance is given by

$$\rho_Z(t,T) = \frac{C_Z(t,T)}{\sigma_Z(t) \sigma_Z(t+T)} = \frac{K(t) K(t+T) C_Y(T)}{K(t) K(t+T) \sigma_Y^2},$$

or

$$\rho_Z(T) = \frac{C_Y(T)}{\sigma_Y^2} = \rho_Y(T),$$

which is invariant with $t$. To the process characterized by an invariant (or stationary) normalized covariance function we have attached the descriptor "wide-sense stationoid." It is this class of stationoid processes which can be generated by the separated network technique.

The synthesis problem now takes the following form. Given the spectral height of the input $X$, and given the covariance function of $Z$ to be produced, determine $N(j\omega)$ and $K(t)$. With no loss of generality, we may choose $N(j\omega)$ so that $C_Y(0) = 1$. Then from (3.4), $K(t)$ can be expressed in terms of the (given) variance of $Z$:

$$K(t) = \sqrt{C_Z(t,0)} = \sigma_Z(t).$$

Now, since $G_Y(\omega)$ is the Fourier transform of $C_Y(T)$, we may express the requirements on $N$ by taking the transform of (3.2). This yields

$$|N(j\omega)|^2 = \frac{1}{G_X} \int_{-\infty}^{\infty} G_Y(T) \cos \omega T dT$$

$$= \frac{1}{G_X} \int_{-\infty}^{\infty} \rho_Z(T) \cos \omega T dT,$$

the latter expression following from (3.6) and the assumption that $\sigma_Y^2 = C_Y(0) = 1$. 

-9-
In summary, the separated network technique can be used in conjunction with a standard noise generator to generate wide-sense stationoid processes having a specified normalized covariance function and a specified instantaneous variance. Equations (3.7) and (3.8) constitute the design formulas for such separated networks.

References, Chapter III

Chapter IV
SYNTHESIS OF NONSTATIONARY PROCESSES FOR MONTE CARLO STUDIES

As an example of how a nonstationary process might be synthesized for use in an analog computer Monte Carlo study, we consider the following hypothetical problem. A rocket is to be fired in a vertical attitude and assumed to rise at a constant rate. While moving through the atmosphere it will be subject to buffeting winds, which are regarded as random in nature. We wish to simulate the rocket system and observe its response to the simulated random wind. A number of computational runs will be performed to gather statistical data on the rocket performance.

Conceptually at least, the required data on the statistical behavior of the wind could be obtained by placing sensing elements at various altitudes and recording sample functions of wind velocity. If we assume that the speed and direction of the wind are independent scalar quantities, then these may be simulated separately and combined in the proper relationship by a resolver. To simplify the present treatment we consider only the scalar wind speed.

Viewed as a whole, the wind speed may be thought of as a stochastic process of two parameters—altitude, \( z \), and time, \( t \). Let this process be denoted by \( X(z,t) \) and designate the mean and covariance by \( \mu_X(z,t) \) and \( C_X(z,z,t,T) \), respectively, where \( Z \) signifies the altitude interval corresponding to a time interval \( T \).

If it is understood that the rocket will be launched only under good weather conditions and at a particular time of day, then it may be assumed that the wind at any given altitude is describable as a stationary stochastic process. (It is the fact that the rocket will be constantly changing altitude which leads to a nonstationary process). Under these conditions, the mean and covariance are invariant in \( t \) and may therefore be written as \( \mu_X(z) \) and \( C_X(z,z,T) \).

Since the rocket is to rise at a constant rate, \( v \), we have for the altitude parameters
(4.1) \[ z = \nu t , \]

(4.2) \[ Z = \nu T . \]

The process may now be written in terms of the single variable \( t \), as \( X(\nu t, t) \). This is expressed more concisely as a new process \( Y(t) \) having mean and covariance given by

(4.3) \[ \mu_Y(t) = E\{Y(t)\} = \mu_X(\nu t) = E\{X(\nu t, t)\} , \]

(4.4) \[ C_Y(t, T) = E\{[Y(t) - \mu_Y(t)][Y(t+T) - \mu_Y(t+T)]\} = C_X(\nu t, \nu T, T) . \]

To provide a concrete example, suppose that data taken from the wind sensing devices indicate for the wind process a mean and covariance of the following form:

(4.5) \[ \mu_X(z) = Ae^{-mz} , \]

(4.6) \[ C_X(z, z, T) = Be^{-\alpha T} e^{-(z+|z|)} . \]

(It is understood that \( z, Z, t, \) and \( T \) will normally be positive quantities in the problem under consideration; however, absolute magnitudes are used in order to give the covariance function the necessary characteristic of being an even function of \( Z \) and \( T \).) The foregoing expressions lead to a mean, variance, and covariance for the process \( Y(t) \) as follows:

(4.7) \[ \mu_Y(t) = \mu_X(\nu t) = Ae^{-m\nu t} , \]

(4.8) \[ \sigma_Y^2(t) = Be^{-\nu t} , \]

(4.9) \[ C_Y(t, T) = C_X(\nu t, \nu T, T) = Be^{-\alpha |T|} e^{-(\nu t+\nu |T|)} . \]

Dividing (4.9) by \( \sigma_Y(t) \sigma_Y(t+T) \) from (4.8) we obtain the normalized covariance:

(4.10) \[ \rho_Y(T) = e^{-(a+i\nu |T|)} . \]
This functional form is characteristic of the wide-sense stationoid process discussed in Chapter III, and is therefore amenable to treatment through the separated-network technique.

By applying Equations (3.7) and (3.8), the multiplying function $K$ is found to be

\begin{equation}
K(t) = \alpha_Y(t) = \sqrt{B} \, e^{-\frac{1}{2} \nu t},
\end{equation}

while the invariant-network power transfer function is

\begin{equation}
|N(j\omega)|^2 = \frac{1}{\varphi_W} \int_{-\infty}^{\infty} \rho_Y(T) \cos \omega T \, dT
= \frac{1}{\varphi_W} \int_{-\infty}^{\infty} e^{-(\mu + \frac{1}{2} \nu) |T|} \cos \omega T \, dT
= \frac{1}{\varphi_W} \cdot \frac{1}{\nu + 2a} \cdot \frac{1}{1 + \left(\frac{2}{\nu + 2a}\right)^2 \omega^2},
\end{equation}

where $\varphi_W$ is the (constant) spectral height of the noise generator. This last expression will be recognized as the power transfer function of a low-pass filter having transfer function

\begin{equation}
N(s) = \frac{2}{\sqrt{\varphi_W(v + 2a)}} \cdot \frac{1}{\left(\frac{2}{v + 2a}\right)s + 1}.
\end{equation}

The required nonstationary stochastic process, $Y(t)$, may then be generated as shown in the figure below.
### Appendix A

**GLOSSARY OF TERMS**

<table>
<thead>
<tr>
<th>TERM</th>
<th>SYMBOL</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$t$</td>
<td>(independent variable)</td>
</tr>
<tr>
<td>Time Interval</td>
<td>$T$</td>
<td>(parameter)</td>
</tr>
<tr>
<td>Process Variable</td>
<td>$X(t)$</td>
<td>(a function of time)</td>
</tr>
<tr>
<td>Time Average</td>
<td>$A{X(t)}$</td>
<td>$\lim_{T \to \infty} \frac{1}{T} \int_{-T}^{T} X(t) , dt$</td>
</tr>
<tr>
<td>Ensemble Average</td>
<td>$E{X(t)}$</td>
<td>$\int_{-\infty}^{\infty} X(t) , dP_X[X(t)]$</td>
</tr>
<tr>
<td>Variance</td>
<td>$\sigma_X^2$</td>
<td>$E{[X(t) - \mu_X]^2}$</td>
</tr>
<tr>
<td>Covariance Fcn.</td>
<td>$C_X(t,T)$</td>
<td>$E{[X(t) - \mu_X(t)][X(t+T) - \mu_X(t+T)]}$</td>
</tr>
<tr>
<td>Normalized Covariance Fcn.</td>
<td>$\rho_X(t,T)$</td>
<td>$\frac{C_X(t,T)}{\sigma_X(t) \sigma_X(t+T)}$</td>
</tr>
<tr>
<td>Correlation Fcn.</td>
<td>$R_X(T)$</td>
<td>$\lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} X(t)X(t+T) , dt$</td>
</tr>
<tr>
<td>Power Spectral Density Fcn.</td>
<td>$\Phi_X(\omega)$</td>
<td>$\lim_{T \to \infty} \frac{1}{2T} \left</td>
</tr>
<tr>
<td>Probability Distribution Fcn.</td>
<td>$P_X[a]$</td>
<td>Probability that $X \leq a$</td>
</tr>
<tr>
<td>Probability Density Fcn.</td>
<td>$p_X[a]$</td>
<td>$\frac{dP_X[a]}{da}$, for continuous $P_X$</td>
</tr>
</tbody>
</table>
# Appendix B

## LIST OF AUTHOR REFERENCES

<table>
<thead>
<tr>
<th>Authors Cited</th>
<th>Chapter References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bartlett MS</td>
<td>I-11</td>
</tr>
<tr>
<td>Bell DA</td>
<td>I-6</td>
</tr>
<tr>
<td>Bendat JS</td>
<td>I-4</td>
</tr>
<tr>
<td>Brodin J</td>
<td>III-1</td>
</tr>
<tr>
<td>Darlington S</td>
<td>III-2, -3</td>
</tr>
<tr>
<td>Davenport WB Jr, Root WL</td>
<td>I-12</td>
</tr>
<tr>
<td>Gerlach AA</td>
<td>III-4</td>
</tr>
<tr>
<td>Goldman S</td>
<td>I-5</td>
</tr>
<tr>
<td>Kharkevich AA</td>
<td>II-6</td>
</tr>
<tr>
<td>Lampard DG</td>
<td>II-4</td>
</tr>
<tr>
<td>Laning JH, Battin RH</td>
<td>I-10, III-5</td>
</tr>
<tr>
<td>MacDonald JR, Edmondson DE</td>
<td>III-6</td>
</tr>
<tr>
<td>Middleton D</td>
<td>I-7</td>
</tr>
<tr>
<td>Miller KS</td>
<td>III-7</td>
</tr>
<tr>
<td>Mood AM</td>
<td>I-2</td>
</tr>
<tr>
<td>Page CH</td>
<td>II-3</td>
</tr>
<tr>
<td>Pipes LA</td>
<td>III-8, -9</td>
</tr>
<tr>
<td>Schwartz M</td>
<td>I-3</td>
</tr>
<tr>
<td>Silverman RA</td>
<td>II-7</td>
</tr>
<tr>
<td>Stewart JL</td>
<td>I-8</td>
</tr>
<tr>
<td>Turner CHM</td>
<td>II-5</td>
</tr>
<tr>
<td>Wadsworth GP, Bryan JG</td>
<td>I-9</td>
</tr>
<tr>
<td>Zadeh LA</td>
<td>II-1, -2, III-10, -11, -12, -13</td>
</tr>
</tbody>
</table>

-15-
TECHNICAL NOTE NO. 2
Project No. A-588

OUTLINE OF NETWORK ANALYZER APPLICATIONS

By R. E. Bryan

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473
(Development of New Methods and Applications of Analog Computation)

15 August
Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
OUTLINE OF NETWORK ANALYZER APPLICATIONS

By
R. E. Bryan

TECHNICAL NOTE NO. 2

on
Contract No. NAS8-2473
(Development of New Methods and
Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama

Approved:

F. Dixon
Project Director A-588, and
Head, Special Problems Branch
# CONTENTS

1. **INTRODUCTION**
   
   1-1. Background Information .................................. 1  
   1-2. Reference Sources ........................................ 1  
   1-3. General Description of Network Analyzers ............. 2

2. **MATHEMATICAL APPLICATIONS**
   
   2-1. Ordinary Differential Equations .......................... 4  
      a. Linear and Nonlinear Equations  
      b. Systems of Equations  
   2-2. Partial Differential Equations ............................ 4  
      a. Laplace and Poisson Equations  
      b. Diffusion Equation  
      c. Wave Equation  
      d. Modified Wave Equation  
      e. Biharmonic Equation and Mixed Derivatives  
      f. Maxwell's Equations  
   2-3. Linear Algebraic Equations ................................ 5  
   2-4. Secular Equations ........................................ 5

3. **ENGINEERING AND SCIENTIFIC APPLICATIONS**
   
   3-1. Electrical Power Distribution ............................. 6  
   3-2. Mechanical Structures ...................................... 6  
      a. Beam Simulation  
      b. Large Elastic Structures  
      c. Plane Stress Analysis  
   3-3. Heat Flow .................................................. 7  
      a. One-Dimensional  
      b. Two-Dimensional  
      c. Three-Dimensional  
   3-4. Fluid Flow .................................................. 7  
      a. Incompressible  
      b. Compressible  
   3-5. Particle Diffusion .......................................... 8  
   3-6. Nuclear Reactor Kinetics ................................... 8  
   3-7. Quantum Mechanics .......................................... 8  
      a. Polyatomic Molecule Vibration  
      b. Schrodinger Equation  
   3-8. Biomedical Studies ......................................... 9  
   3-9. Waveguides and Cavity Resonators ........................ 9
Chapter I
INTRODUCTION

1-1. Background Information

This Technical Note presents results of a brief survey of the literature on network analyzers, which was performed under Contract NAS8-2473 ("New Methods and Applications of Analog Computation") as part of a special task assignment entitled "Comparison of Passive Element Computers with Other Computers." Additional information to be gathered under this task should provide a basis for determining what size and type of network analyzer, if any, the Computation Division at Marshall Space Flight Center might acquire in order to supplement existing analog and digital computer facilities most effectively. The present report is intended to serve as a check-list or guide to the classes of problems for which the network analyzer might be considered.

1-2. Reference Sources

The bulk of the information on types of problems which have been solved with network analyzers was obtained from the following textbooks:


For convenience, these textbooks are referred to throughout this report by the simple designations FIFER, KARPLUS, KORN, and SOROKA, respectively. Specific sections are identified by page numbers in parentheses following the author designation. In the case of FIFER, the volume number is noted as well.

The textbook citations have been supplemented by references to journal articles for several of the engineering and scientific applications covered in Chapter III. These references are given in complete form under the appropriate topic headings.
1-3. General Description of Network Analyzers

The network analyzer (otherwise known as "direct analogy computer" or "passive element computer") utilizes a one-to-one relationship between physical laws describing the behavior of a dynamic system and the voltage and current laws describing the behavior of a lumped-constant linear electrical network. In general, each element of the analogue network (resistor, inductor, capacitor) corresponds to a parameter of the physical system.

Network analyzers may be categorized on the basis of which combination of circuit elements they contain. A DC network analyzer contains only resistive elements, while an AC network analyzer may be of the resistive-capacitive, inductive-capacitive, or resistive-inductive-capacitive type. The inductive-capacitive type is essentially a resistive computer extended to handle negative resistances. However, the other AC network analyzers may be further differentiated according to their use as either transient or steady-state analyzers. A general-purpose network analyzer will have both the input-output devices and the necessary network components to permit operation in any of the above modes.

Amplifiers and transformers are used in the analogue network when the system equations are not otherwise physically realizable.

The input-output equipment associated with both DC and AC steady-state network analyzers is essentially the same: variable voltage and current sources for inputs; and voltmeters, ammeters, and wattmeters for reading the outputs. Also, a digital meter may be used in conjunction with a printer to permanently record output data.

AC transient analyzers, which work on a repetitive operation basis, require more specialized equipment. The input is applied periodically; a switching network, synchronized with it, restores initial conditions to the capacitors during the "off" period. An oscilloscope is required to observe the output, and a camera is used to permanently record the data.

Network analyzers are particularly adapted to solving partial differential equations with either the space variables or space and time variables discretized through finite-difference approximations. If both space and time variables are discretized, the resulting system of algebraic equations may be
programmed on the DC network analyzer. With the space variables discretized, the AC resistive-capacitive analyzer will handle systems whose equations have the general form of either the Laplace, Poisson, or diffusion equations; more sophisticated systems require a resistive-inductive-capacitive analyzer. For analysis of large elastic structures, another area in which the network analyzer excels, one needs not only the full complement of basic elements but also transformers.

Since there is no isolation between components, the network analyzer calls for more engineering knowledge and ingenuity in programming than does the electronic analog computer. On the other hand, FIFER contends that experienced personnel can mechanize very complicated systems on the network analyzer without the intermediate step of writing equations. This type of direct programming lends itself to intuitive model building when precise equations for a system cannot be easily derived.

Accuracies normally obtained with network analyzers fall in the range of 1 to 5 percent. The better accuracies are realized on resistive and resistive-capacitive computers, since highly accurate resistors and capacitors are commercially available at reasonable costs. Grosser errors are usually associated with inductors, transformers, and output equipment. Accurate readout equipment (servo recorders) may be used if the mechanization is scaled sufficiently slow, but this in turn greatly increases the cost of capacitors, inductors, and transformers. Another type of refinement which improves accuracy is the addition of feedback amplifiers in the network. Although this constitutes a rather costly addition, FIFER notes a recent trend towards a hybrid type of analyzer which includes not only the feedback amplifier but also electronic multipliers and function generators.
2. MATHEMATICAL APPLICATIONS

This chapter provides an outline of classes of mathematical problems to which network analyzers have been applied. Each of the listed applications is followed by one or more references to the textbooks cited in Section 1-2 above, which indicate the particular methods or techniques used.

2-1. Ordinary Differential Equations

a. Linear and Nonlinear Equations

SOROKA (289-295): Finite difference methods.

b. Systems of Equations

SOROKA (295-296): Finite difference methods.

2-2. Partial Differential Equations

a. Laplace and Poisson Equations \((\nabla^2 \phi = 0, \nabla^2 \phi = k)\)

FIPER (III, 781-784): Laplace equation with space variables discretized solved on DC network analyzer. Also, comment on same solution using inductive-capacitive AC network analyzer.

KARPLUS (172-186): DC network analyzer solution to Laplace equation in cartesian, cylindrical, and spherical coordinates. Boundaries, net spacing, diagonal nets, terminations, nonuniform fields, and distributed internal excitation also discussed.

SOROKA (286-289): DC network analyzer solution to Laplace equation in curvilinear coordinates—cylindrical, polar, and axial symmetry.


SOROKA (296-299): Generalization of methods used on Laplace and Poisson equations.

b. Diffusion Equation \((\nabla^2 \phi = k \cdot \partial \phi / \partial t)\)

KARPLUS (188-190): DC network analyzer solution to diffusion equation; both space and time variables discretized.
KARPLUS (206-209): Diffusion equation solution using R-C net; only space variables discretized.

c. Wave Equation \( \nabla^2 \phi = k \frac{\partial^2 \phi}{\partial t^2} \)

FIFER (781-787): Wave equation with space variables discretized; special techniques and iterative procedure necessary to realize required negative resistance.

KARPLUS (190-191): Wave equation solution on DC network analyzer; both space and time variables discretized.

KARPLUS (218-220): Wave equation solution using L-C net, with only space variables discretized.

d. Modified Wave Equation \( \nabla^2 \phi = k_1 \frac{\partial^2 \phi}{\partial t^2} + k_2 \frac{\partial \phi}{\partial t} + k_3 \phi \)

KARPLUS (222-223): Modified wave equation solution by R-L-C net, with only space variables discretized.

e. Biharmonic Equation and Mixed Derivatives

KARPLUS (191-192): DC network analyzer solution of the biharmonic equation, \( \nabla^4 \phi = \delta \frac{\partial^4 \phi}{\partial x^4} + 2 \delta \frac{\partial^2 \phi}{\partial x^2 \partial y^2} + \delta \frac{\partial^4 \phi}{\partial y^4} = 0 \); also, comments on the related equation \( \nabla^4 \phi = k \).

SOROKA (299-301): Solution of partial differential equations involving mixed derivatives (no time variable present) using L-C net, with space variables discretized.

f. Maxwell's Equations

FIFER (III, 781-784): Solution of certain forms of Maxwell's equations using AC network analyzer, with space variables discretized.

2-3. Linear Algebraic Equations


2-4. Secular Equations

SOROKA (125-126): Solution of typical secular equation (fourth-order) using L-C net.
3. ENGINEERING AND SCIENTIFIC APPLICATIONS

This chapter provides an outline of the major areas of applied science and engineering in which network analyzers have been employed. Reference is again made to pertinent material from the textbooks cited in Section 1-2; also listed are various journal articles dealing with specialized problems of interest.

3-1. Electrical Power Distribution

FIFER (I, 4): Comment on mechanization without mathematical formulation for power system analysis.


3-2. Mechanical Structures

a. Beam Simulation

SOROKA (255-261): Beam simulation using the "influence coefficient" concept.

SOROKA (270-276): Dynamic beam simulation developed through finite difference approach (as opposed to influence numbers).

KARPLUS (390-397): General introduction to beam simulation, with example using R-L-C net.

FIFER (III, 796-807): Beam vibration—static bending, dynamic bending, coupled transient bending, uncoupled transient torsion, plus discussion on boundary conditions.

b. Large Elastic Structures

KORN (147): Comment on simulating large elastic structures (such as an airplane wing) by means of a network computer.

FIFER (IV, 1206-1215): Flutter analysis using section-by-section analysis of wing to determine flutter speed and frequency; development of circuits for the forcing functions (aerodynamic equations).
c. Plane Stress Analysis

KARPLUS (399-403): Plane stress problems using resistance net.
SOROKA (313-336): Lattice analogy for plane stress systems (i.e., continuous elastic body replaced by framework of pinned elastic members) in cartesian and cylindrical coordinates.

3-3. Heat Flow

a. One-Dimensional

KARPLUS (337-339): Heat conduction with space variables discretized, resulting in R-C net.
KARPLUS (339-341): Heat conduction with space and time variables discretized, resulting in resistance net.
FIFER (III, 788-790): Heat equation with both space and time variables discretized, resulting in L-C net.

b. Two-Dimensional

FIFER (787): Heat equation with space variables discretized, resulting in R-C net.

c. Three-Dimensional

KARPLUS (333-337): R-C net for heat conduction in one, two, and three dimensions, with emphasis on time scaling.

3-4. Fluid Flow

a. Incompressible

KARPLUS (306-309): Viscous fluid flow patterns in the vicinity of oil wells.

b. Compressible


3-5. Particle Diffusion

KARPLUS (323-327): Analysis to determine steady-state distribution of particles emitted from point, line, and area sources in the presence of wind and atmospheric turbulence; space variables discretized, resulting in R-C net.

3-6. Nuclear Reactor Studies

KARPLUS (327-329): Comments on determination of neutron flux distribution in nuclear reactor cores and reflectors.


3-7. Quantum Mechanics

a. Polyatomic Molecule Vibration

FIFER (III, 937): Comment on General Electric Company's use of their network analyzer to solve the eigenvalue problem associated with the vibration of polyatomic molecules.


b. Schrödinger Equation

KARPLUS. (376-377): Comments on both the resistance method and the L-C method for analyzing the Schrödinger equation.


3-8. Biomedical Studies


3-9. Waveguides and Cavity Resonators


TECHNICAL NOTE NO. 3
Project No. A-588

GENERATION AND MEASUREMENT OF NONSTATIONARY RANDOM PROCESSES

By Roger P. Webb
Joseph L. Hammond, Jr.
Richard E. Bryan
Thomas M. White, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation)

22 July, 1964

Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
GENERATION AND MEASUREMENT OF NONSTATIONARY RANDOM PROCESSES

By
Roger P. Webb
Joseph L. Hammond, Jr.
Richard E. Bryan
Thomas M. White, Jr.

TECHNICAL NOTE NO. 3
on
Contract No. NAS8-2473
(Development of New Methods and Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
## CONTENTS

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Illustrations</td>
<td>iii</td>
</tr>
<tr>
<td>I. INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>II. SYNTHESIS OF TIME VARYING NETWORKS FOR THE GENERATION OF NONSTATIONARY STOCHASTIC PROCESSES</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>2.2 Representation of a Stochastic Process</td>
<td>3</td>
</tr>
<tr>
<td>2.3 The Covariance Function</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Selected Topics from the Theory of Ordinary Differential Equations</td>
<td>5</td>
</tr>
<tr>
<td>2.5 The Synthesis Procedure</td>
<td>12</td>
</tr>
<tr>
<td>2.6 Summary and Example</td>
<td>15</td>
</tr>
<tr>
<td>III. MEASUREMENTS ON NONSTATIONARY STOCHASTIC PROCESSES</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Averages</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Computer Instrumentation</td>
<td>21</td>
</tr>
<tr>
<td>3.4 Sampling Errors</td>
<td>25</td>
</tr>
<tr>
<td>3.5 Summary of Results on Sampling Errors</td>
<td>33</td>
</tr>
<tr>
<td>3.6 Computations Performed by the Digital Computer</td>
<td>34</td>
</tr>
<tr>
<td>IV. EXAMPLES</td>
<td>37</td>
</tr>
<tr>
<td>Example 1</td>
<td>37</td>
</tr>
<tr>
<td>Example 2</td>
<td>40</td>
</tr>
<tr>
<td>Example 3</td>
<td>50</td>
</tr>
<tr>
<td>V. CONCLUSIONS</td>
<td>55</td>
</tr>
<tr>
<td>References</td>
<td>56</td>
</tr>
</tbody>
</table>


## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A Generalized Computer Implementation of Equation (2.11)</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Computer Implementation of Example</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Generation of Sample Functions for a Nonstationary Random Process</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Measuring System</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>80% Confidence Limits on Fractional Errors</td>
<td>31</td>
</tr>
<tr>
<td>3.4</td>
<td>Format for Tabular Presentation of Covariance Estimates</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Mean and Standard Deviation as a Function of Time for Example 1. n = 180</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>Measured and Theoretical Covariance Functions with t=1 for Example 1. n = 180</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Mean and Standard Deviation as a Function of time for Example 1. n = 18</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Measured and Theoretical Covariance Functions with t=1 for Example 1. n = 18</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>Implementation of $\dot{x} + \frac{X}{t} = \frac{Y}{t}$</td>
<td>46</td>
</tr>
<tr>
<td>4.6</td>
<td>Improved Implementation of $\dot{x} + \frac{X}{t} = \frac{Y}{t}$</td>
<td>46</td>
</tr>
<tr>
<td>4.7</td>
<td>Normalized Mean and Standard Deviation as a Function of Time for Example 2. n = 150</td>
<td>48</td>
</tr>
<tr>
<td>4.8</td>
<td>Measured and Theoretical Covariance Functions with t=1 for Example 2. n = 150</td>
<td>49</td>
</tr>
<tr>
<td>4.9</td>
<td>Implementation of $\dot{x} + x \left(\frac{1}{t+1}\right) = y \left(\frac{1}{t+1}\right)$</td>
<td>51</td>
</tr>
<tr>
<td>4.10</td>
<td>Normalized Mean and Standard Deviation as a Function of Time for Example 3. n = 150</td>
<td>52</td>
</tr>
<tr>
<td>4.11</td>
<td>Measured and Theoretical Covariance Functions with t=1 and t=3 for Example 3. n = 150</td>
<td>53</td>
</tr>
</tbody>
</table>
I. INTRODUCTION AND SUMMARY

This Technical Note is concerned with certain aspects of the generation and use of nonstationary stochastic processes in Monte Carlo studies using an analog computer. It represents a continuation and extension of Technical Note 1 of the current project.

Certain aspects of the study are essentially complete. Section II of this Note outlines in detail a general method for the synthesis of analog computer circuits which when excited by stationary Gaussian white noise produce nonstationary random outputs with prescribed first and second statistical moments.

Section III of the Note describes a system for the measurement and processing of the nonstationary random processes. Also included in Section III is an error analysis which gives a confidence level criterion for use with the measured random data.

Section IV presents several examples for which computer circuits are constructed to realize given covariance functions, and for which the moments of the experimentally generated processes are measured. These examples illustrate the use of the synthesis procedure and the analog implementation, and also give an indication of the accuracy which can be achieved in this type of work.

Work is continuing on certain aspects of the analytical representation of random processes and on possible simplifications and extensions of the current results. Plans for the continuation are discussed in Section V of this Note.
II. SYNTHESIS OF TIME VARYING NETWORKS
FOR THE GENERATION OF NONSTATIONARY STOCHASTIC PROCESSES

2.1 Introduction

Frequently in analog simulation of physical systems it is necessary
to generate stochastic processes. In some cases the generation of stochas-
tic processes is conveniently accomplished by shaping the output of a sta-
tionary Gaussian white noise source with an appropriate computer network.
The structure of this network is related to the statistical moments of the
required process. In case the required process has stationary statistics,
the appropriate network transfer function can be obtained by factoring the
spectral density function of the process. However, for nonstationary pro-
cesses the appropriate network must be obtained in a different manner. In
the sequel, a technique is presented which will realize any random process,
insofar as its first two statistical moments are concerned, as the output
of a linear time varying network excited by stationary white noise. The
technique involves expanding the covariance function of the given process
into a finite series. The series expansion is then used to determine the
coefficients of a linear differential equation whose analog realization is
the required network. The order of the differential equation, and hence the
complexity of the required analog network is directly related to the number of
terms in the series expansion. Without loss in generality the technique
presented herein is concerned only with the realization of processes with
mean zero. Processes with nonzero mean can be realized as the sum of the
random process generated by this technique and a deterministic function equal
to the required mean.
2.2 Representation of a Stochastic Process

Due to a theorem by Doob [1], it is known that for any random process with given mean and covariance, there exists a Gaussian random process with identical mean and covariance. Consequently if only the first two statistical moments of a stochastic process are of interest, the generation of an appropriate Gaussian process will suffice.

It is also known ¹ that any Gaussian random variable \( x(t) \) with mean zero can be represented formally by the equation

\[
x(t) = \int_0^t G(t,s)y(s)ds
\]  

(2.1)

For processes with nonzero mean a variable \( x^*(t) \) can be defined to obtain

\[
x^*(t) = x(t) - m(t) = \int_0^t G(t,s)y(s)ds
\]

which is equivalent to Equation (2.1).

In Equation (2.1) \( y(t) \) is a stationary Gaussian white noise process and \( G(t,s) \) is a Green's function for an appropriate linear differential equation. A procedure for synthesizing a network for producing \( x(t) \) from \( y(t) \) can be based on Equation (2.1) by noting that this equation represents the solution to an \( n^{th} \) order linear differential equation of the form

\[
x^{(n)} + p_{n-1}(t)x^{(n-1)} + \cdots + a_1(t)\dot{x} + a_0(t)x = q_m(t)y^{(m)} + \cdots + q_1(t)\dot{y} + q_0(t)y
\]

¹ Levy [2], Webb [3].
or in shorter notation, \[ L_t x = N_t y \] \hspace{1cm} (2.2)

This differential equation and hence the realization of the required stochastic process can be obtained by an analog computer realization of Equation (2.2) with a white noise input source. The synthesis of the required network is thus completed by the specification of the coefficients \( p_i(t), q_i(t) \). It is convenient for computational purposes to realize Equation (2.2) as \( n \) first order differential equations rather than as an \( n^{th} \) order equation. This alternate representation and its equivalence to the form of Equation (2.2) will be given in a subsequent section.

2.3 The Covariance Function

The synthesis procedure for the required analog network is developed using certain properties of the covariance function for the required process. It is assumed that this function is known in some form.

The covariance function for the random variable \( x(t) \) with mean zero is given by

\[ r(t', t) = E[x(t_1) x(t_2)] \] \hspace{1cm} (2.3)

where \( E \) is the expectation operation and

\[ t' = \text{larger of } (t_1 \text{ and } t_2) \]

\[ t = \text{smaller of } (t_1 \text{ and } t_2) \] .

In many cases with physical significance the covariance function can be
expressed as

\[ r(t',t) = \sum_{i=1}^{n} \bar{\phi}_i(t')\gamma_i(t) \quad , \]  

(2.4)

where \( \bar{\phi}_i(t) \), \( \gamma_i(t) \) are known functions of time. If Equation (2.4) does not apply exactly, Mercer’s theorem guarantees that any bounded covariance function can be approximated to arbitrary accuracy by such a sum. Since the form of Equation (2.4) is particularly useful in developing the required networks, this representation of the covariance function will be used throughout.

As indicated above, the output of an analog computer network with a white noise input can be expressed by Equation (2.1). A straightforward computation shows that the covariance function of the output of such a network can be expressed as

\[ r(t',t) = \int_0^t G(t',s)G(t,s)ds \quad . \]  

(2.5)

The procedure for synthesizing the computer network is developed from Equations (2.4) and (2.5), in which \( r(t',t) \) is known, by determining the parameters in the differential equation, (of which \( G(t,s) \) is the Green's function), by use of the expansion of Equation (2.4).

2.4 Selected Topics from the Theory of Ordinary Differential Equations

This Section summarizes certain topics from differential equation theory and provides further background for the synthesis procedure which will be developed in Section 2.5.

---

2Davenport and Root [4]
Consider a linear $n^{th}$ order differential equation of the form of Equation (2.2). Associated with this equation is the homogeneous differential equation

$$L_t x = 0$$

(2.6)

This equation has $n$ linearly independent solutions which are denoted by $\phi_1(t), \phi_2(t), \ldots, \phi_n(t)$. The linear differential operator $L_t$ can be specified in terms of the $\phi_i$'s by the relation

$$L_t x = W(x, \phi_1, \phi_2, \ldots, \phi_n)(t) = 0$$

(2.7)

where the Wronskian $W$ is given by

$$W(x, \phi_1, \phi_2, \ldots, \phi_n)(t) = \det \begin{vmatrix} x & \phi_1 & \ldots & \phi_n \\ x & \phi_1 & \ldots & \phi_n \\ \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots \\ x(n) & \phi_1(n) & \ldots & \phi_n(n) \end{vmatrix}$$

In addition, there exists for Equation (2.6) a function $H(t,s)$ which satisfies

$$L_t H(t,s) = 0$$
and is defined by

\[
H(t, s) = 0, \quad t < s
\]

and

\[
H(t, s) = \begin{vmatrix}
\phi_1(s) & \phi_2(s) & \cdots & \phi_n(s) \\
\phi_1(s) & \phi_2(s) & \cdots & \phi_n(s) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_1(n-2)(s) & \phi_2(n-2)(s) & \cdots & \phi_n(n-2)(s) \\
\phi_1(n-1)(s) & \phi_2(n-1)(s) & \cdots & \phi_n(n-1)(s)
\end{vmatrix}, \quad t > s
\]  

(2.8)

where

\[
W(\phi_1, \phi_2, \ldots, \phi_n)(s) = \det
\]

\[
\begin{vmatrix}
\phi_1(s) & \cdots & \phi_n(s) \\
\phi_1(s) & \cdots & \phi_n(s) \\
\vdots & \ddots & \vdots \\
\phi_1(n-1)(s) & \cdots & \phi_n(n-1)(s)
\end{vmatrix}
\]

The nonhomogeneous differential Equation (2.2) with zero initial conditions then has the unique solution

\[
x(t) = \frac{1}{t_0 t} \int_{t_0}^{t} H(t, s) N_s y(s) ds
\]  

(2.9)

where \( N_s \) is the operator defined in Equation (2.2).
Then, using Green's formula\(^3\), Equation (2.9) can be rewritten as

\[
x(t) = \int_0^t G(t,s)y(s)\,ds,
\]

(2.10)

where \(G(t,s)\) is the Green's function for an equation of the form of Equation (2.2). In network terminology, \(G(t,0)\) is the impulse response of a network.

As noted previously, it is convenient for computational purposes to convert the \(n\)th order Equation (2.2) into a set of \(n\) first order differential equations. It is also convenient to accomplish this in such a manner that no derivatives of the input process are required. To make this conversion the following identifications are made. It is assumed here without loss of generality that \(m = n-1\).

\[
x(t) = x_1(t)
\]

\[
\dot{x}_1 = x_2 - a_{n-1}(t)x_1 + b_m(t)y
\]

\[
\dot{x}_2 = x_3 - a_{n-1}(t)x_1 + b_{m-1}(t)y
\]

\[
\ldots
\]

\[
\dot{x}_{n-1} = x_n - a_1(t)x_1 + b_1(t)y
\]

\[
\dot{x}_n = -a_0(t)x_1 + b_0(t)y
\]

\(^3\)Coddington and Levinson [5] p. 86.
In matrix notation this set can be written more concisely as

\[
\dot{x} = A(t) x + B(t)y
\]

\[
x = H x
\]

where \( x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad B = \begin{bmatrix} b_m(t) \\ b_{m-1}(t) \\ \vdots \\ b_1(t) \\ b_0(t) \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}
\]

\[
A(t) = \begin{bmatrix} -a_{n-1}(t) & 1 & 0 & \cdots & \cdots & 0 \\ -a_{n-2}(t) & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots \\ \vdots & \vdots & & \vdots & \ddots & \ddots \\ -a_1(t) & 0 & \cdots & \cdots & \cdots & 1 \\ -a_0(t) & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}
\]

It can be verified that the elements \( a_k, b_k \), in Equation (2.11) are related to the coefficients \( p_k, q_k \), in Equation (2.2) by
If the $p_k$, $q_k$ are known, Equations (2.12) and (2.13) can be solved sequentially for the $a_k$, $b_k$.

Note that Equation (2.11) which is the alternate representation of Equation (2.2) results in the particularly simple analog realization shown in Figure 2.1.

Associated with the vector differential equation (2.11) is the homogeneous equation

$$\dot{\mathbf{x}} = A(t) \mathbf{x}$$

(2.14)

If $\phi_1(t)$, $\phi_2(t)$, ..., $\phi_n(t)$ are again the $n$ linearly independent solutions to Equation (2.6), then there exists for Equation (2.14) a fundamental matrix solution $\Phi(t)$ satisfying

$$\frac{d}{dt} \Phi(t) = A(t)\Phi(t).$$

The matrix $\Phi(t)$ is defined by

$$\Phi(t) = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \cdots & \phi_{nn} \end{bmatrix}$$

(2.15)
Figure 2.1. A Generalized Computer Implementation of Equation 2.11.
where \[ \phi_{i,j} = \frac{d}{dt} \phi_{i,j-1} + a_{n-j+1} \phi_{i,l} \]

and \[ \phi_{k,1} = \phi_{k,l} \quad (k = 1, \ldots, n) \].

The nonhomogeneous differential Equation (2.11) with zero initial conditions then has the unique solution

\[ x(t) = \int_0^t \phi(t)\phi^{-1}(s)B(s)y(s)\,ds \quad (2.16) \]

where \( \phi^{-1} \) is the matrix inverse of \( \phi \).

Again using the properties of white noise a covariance matrix for the vector \( x(t) \) can be written as

\[ R(t',t) = E[x(t_1)x^T(t_2)] = \int_0^t \phi(t')\phi^{-1}(s)B(s)B^T(s)[\phi^{-1}(s)]^T\phi(t)\,ds \quad (2.17) \]

where \( (\cdot)^T \) denotes matrix transpose.

2.5 The Synthesis Procedure

With the preceding background it is now possible to outline the synthesis procedure for the required analog network. The procedure requires a specification of the elements \( a_k \) and \( b_k \) of the representation of Equation (2.11).

Combining Equations (2.4) and (2.5) results in

\[ r(t',t) = \int_0^t G(t',s)G(t,s)\,ds = \sum_{i=1}^n \phi_1(t')\gamma_i(t) \quad . \quad (2.18) \]

Then using the fact that \( G(t',s) \) satisfies \( L_t G(t',s) = 0 \) it follows from
applying the operator $L_t$ to both sides of Equation (2.18) that 

$$L_t \sum_{i=1}^{n} \Phi_i(t') \gamma_i(t) = 0. \tag{2.19}$$

Consequently, the $n \Phi_i$ in Equation (2.4) are solutions to the homogeneous Equation (2.6). Since $n$ linearly independent solutions to Equation (2.6) form a unique basis for all solutions, the $\Phi_i$ can be taken as the fundamental set of solutions $\phi_i(t)$ defined in Section 2.4. Consequently, the $- \gamma$ will be dropped hereafter.

The operator $L_t$, and hence the coefficients $p_k$ can be obtained using Equation (2.7). The elements $a_k$ of matrix $A$ can then be obtained directly using Equation (2.12). Once the $a_k$ are determined, the matrix $\phi(t)$ can be written using Equation (2.15). There remains only to determine the elements $b_k$ of $B$.

Upon identifying a matrix $D(t)$ by

$$D(t) = \int_0^t \phi^{-1}(s)B(s)B^T(s)[\phi^{-1}(s)]^T ds \tag{2.20}$$

the covariance matrix Equation (2.17) can be written as

$$R(t^0, t) = \phi(t^0)D(t)\phi^T(t). \tag{2.21}$$

The element in the first row and first column of this matrix is just the scalar covariance $r(t^0, t)$. Consequently, using Equations (2.4) and (2.9)

---

4Note that this assumes that the $\phi_i$ in Equation (2.4) are linearly independent. However, if one of the $\phi_i$ is not linearly independent, it may be expressed as a combination of the remainder and the index $n$ can be reduced by one.
\[ \sum_{i=1}^{n} \phi_i(t') \gamma_i(t) = \sum_{i=1}^{n} \phi_i(t') \sum_{j=1}^{n} d_{ij}(t) \phi_i(t) \]  

(2.22)

where \( d_{ij} \) are elements of matrix \( D \). Equation (2.22) is satisfied by

\[ d_{ij}(t) = \frac{\gamma_i(t)}{\phi_i(t)} , \quad i=j \]

and

\[ d_{ij}(t) = 0 , \quad i \neq j \]  

(2.23)

Equation (2.23) determines the matrix \( D(t) \) and hence the covariance matrix \( R(t',t) \) is known. The elements of \( B(t) \) can be determined from the properties of this matrix.

Define by \( R^*(t',t) \) the extension of \( R(t',t) \) when the sign of the difference \( t_1 - t_2 \) changes. Then

\[ R^*(t',t) = \phi(t)D(t')\phi^T(t') = R^T(t,t'). \]  

(2.24)

Let \( \Delta(t',t) \) denote the difference

\[ \Delta(t',t) = R - R^* = - \int_t^{t'} \phi(t_1)^{-1}(s)B(s)B^T(s)\phi^{-1}(s)^T(t_2)ds. \]  

(2.25)

Upon taking the partial derivative of Equation (2.25) with respect to \( t' \) and evaluating at \( t' = t \) there results

\[ \frac{\partial \Delta(t',t)}{\partial t'} \bigg|_{t'=t} = - B(t)B^T(t) \]  

(2.26)
Then since the diagonal elements of the matrix $BB^T$ are just the $b_k^2$, the elements $b_k$ can be evaluated from

$$b_{m-i+1} = \sqrt{-\delta_{ii}}$$  \hspace{1cm} (2.27)

where $\delta_{ii}$ are the diagonal elements of

$$\frac{\partial \Delta(t', t)}{\partial t'} \bigg|_{t'=t}$$

The matrix $B$ is then specified completing the synthesis procedure.

2.6 Summary and Example

The synthesis procedure is summarized by the following step-by-step procedure:

1. Express the covariance for the required process in the form of Equation (2.4).

2. Using the $\phi_i$ in Equation (2.4) determine:
   a) the operator $L_t$ and coefficients $p_k(t)$ by using Equation (2.7).
   b) the elements $a_k(t)$ from Equation (2.12).
   c) the matrix $\Phi(t)$ from Equation (2.15).

3. Determine the elements of matrix $D(t)$ from Equation (2.23).

4. Use Equations (2.21), (2.24), and (2.25) to determine the matrices $R$, $R^*$, and $\Delta$.

5. Perform the operation of Equation (2.26) and determine the elements of $B(t)$ by means of Equation (2.27).

6. Realize the resulting differential equation by the circuit shown in Figure 1.
An example will serve to clarify the procedure. Consider the generation of a process with covariance

\[ r(t^0, t) = 4e^{-t^0}e^{t} + 2e^{-3t^0}e^{3t} \]

A stationary process has been purposely chosen for clarity. The analytical results of this example can be verified by spectral factorization.

Step 1. The given covariance is \( r(t^0, t) = 4e^{-t^0}e^{t} + 2e^{-3t^0}e^{3t} \). By inspection the \( \phi_i \) are \( \phi_1 = 2e^{-t} \) and \( \phi_2 = \sqrt{2}e^{-3t} \).

Step 2. The operator \( L_t x \) is given by

\[
L_t x = \begin{vmatrix}
   x & 2e^{-t} & \sqrt{2}e^{-3t} \\
   \dot{x} & -2e^{-t} & -3\sqrt{2}e^{-3t} \\
   \ddot{x} & 2e^{-t} & 9\sqrt{2}e^{-3t}
\end{vmatrix} = 0
\]

or

\[
\ddot{x} + 4\dot{x} + 3x = 0.
\]

Thus

\[ p_1 = 4, \ p_0 = 3. \]

Then using Equation (2.12)

\[ p_1 = a_1 = 4 \]
\[ p_0 = a_0 = 3. \]

Using Equation (2.15)

\[ \phi_{11} = 2e^{-t} \quad \phi_{21} = \sqrt{2}e^{-3t} \]
\[ \phi_{12} = -2e^{-t} + 4\cdot 2e^{-t} \quad \phi_{22} = -3\sqrt{2}e^{-3t} + 4\sqrt{2}e^{-3t}, \]
so that

\[
\phi(t) = \begin{vmatrix}
2e^{-t} & \sqrt{2}e^{-3t} \\
6e^{-t} & \sqrt{2}e^{-3t}
\end{vmatrix}.
\]

Step 3.

\[
d_{12} = d_{21} = 0, \quad d_{11} = \frac{2e^t}{2e^{-t}} = e^{2t}, \quad d_{22} = \frac{\sqrt{2}e^{3t}}{\sqrt{2}e^{-3t}} = e^{6t}.
\]

Step 4.

\[
R(t', t) = \begin{vmatrix}
2e^{-t'} & \sqrt{2}e^{-3t'} & e^{2t} & 0 \\
6e^{-t'} & \sqrt{2}e^{-3t'} & 0 & e^{6t}
\end{vmatrix} = \begin{vmatrix}
2e^{-t} & 6e^{-t} \\
\sqrt{2}e^{-3t} & \sqrt{2}e^{-3t}
\end{vmatrix} = \begin{vmatrix}
4e^{-(t'-t)} + 2e^{-3(t'-t)} & 12e^{-(t'-t)} + 2e^{-3(t'-t)} \\
12e^{-(t'-t)} + 2e^{-3(t'-t)} & 36e^{-(t'-t)} + 2e^{-3(t'-t)}
\end{vmatrix}
\]

and

\[
R^* = \begin{vmatrix}
4e^{-(t-t')} + 2e^{-3(t-t')} & 12e^{-(t-t')} + 2e^{-3(t-t')} \\
12e^{-(t-t')} + 2e^{-3(t-t')} & 36e^{-(t-t')} + 2e^{-3(t-t')}
\end{vmatrix}
\]

and \(\Delta(t', t) = R - R^*\).
Step 5.

\[-b_1^2(t) = \frac{\partial}{\partial t^r} \left[ 36e^{-(t'-t)} + 2e^{-3(t'-t)} - 36e(t'-t) - 2e^3(t'-t) \right] \bigg|_{t'=t} = -20\]

\[-b_0^2(t) = \frac{\partial}{\partial t^r} \left[ 36e^{-(t-n'-t)} + 2e^{-3(t'-t)} - 36e(t'-t') - 2e^3(t'-t') \right] \bigg|_{t'=t} = -84\]

Step 6.

\[
A(t) = \begin{bmatrix}
-4 & 1 \\
-3 & 0
\end{bmatrix}, \quad B(t) = \begin{bmatrix}
20 \\
84
\end{bmatrix}
\]

The analog computer circuit obtained from Equation (2.11) is shown in Figure 2.2.
Figure 2.2. Computer Implementation of Example.
III. MEASUREMENTS ON NONSTATIONARY STOCHASTIC PROCESSES

3.1 Introduction

Section II of this Note describes a synthesis procedure for constructing an analog computer circuit which produces from a stationary white noise input a nonstationary output with prescribed first and second moments. This Section will be devoted to a discussion of measurement techniques for estimating the first two moments of nonstationary random processes. The procedures to be described are used in verifying the results of the synthesis procedure, and can also be used in determining the moments of the outputs of simulated systems which are excited by nonstationary random processes.

3.2 Averages

A moment of a random process can be defined as an average over time for a fixed sample function, or as an average over an ensemble of possible sample functions at a fixed time. For processes which are stationary and ergodic, these two methods of averaging yield equivalent results. For nonstationary random processes the two methods of averaging do not in general yield the same results. The question of under what conditions time and ensemble averages are equal is difficult, and this question has received a considerable amount of attention from mathematicians in various ergodic theorems\(^5\). In the present work with nonstationary processes, time variations in such quantities as the variance and covariance are of central interest. Thus, time averages which obscure these variations cannot be used and ensemble averages are employed.

\(^5\) See for example Doob [1].
An ensemble average, for example \( E x^n(t) \) which defines the \( n \)th moment of \( x(t) \), will in general be a function of the time \( t \) at which the average is computed. In the physical generation of a nonstationary random process it is desirable to generate a number of different sample functions with a common time reference, rather than a single sample function as might be generated for a stationary process. Values of the various sample functions at fixed values of time with respect to the reference can then be averaged to produce an estimate of the ensemble average.

The considerations involved in the physical generation of many sample functions from a nonstationary process are the following. A filter produced by the synthesis procedure of Section II is in general time variable, and the statistical properties of its output will change in synchronism with its time variable elements. A single sample function, \( x_1(t) \), of duration \( T \) is thus generated when the time variable elements start at some reference time \( t = t_0 \) and progress through their prescribed variation to \( t = t_0 + T \). A number of sample functions can be generated by cycling the time variable elements so that they are first reset to \( t = t_0 \), and then allowed to progress through their prescribed variations to \( t = t_0 + T \), at which time they are reset and the process repeated. A block diagram for the equipment required to accomplish this cyclic operation using an analog computer is given in Figure 3-1.

3.3 Computer Instrumentation

The instrumentation chosen to measure the moments of \( x(t) \) is shown in Figure 3-2. The Process Source is the analog computer equipment, including the noise generator, required to generate \( x(t) \). The Time Reference provides the signals necessary for cyclic operation of the Process Source as described
Figure 3-1. Generation of Sample Functions for a Nonstationary Random Process.
Figure 3-2. Measuring System.
in Section II, and also triggers a Sample Pulse Generator which produces a signal \( s(t) \) which synchronizes the Digital Voltmeter and the Serializer. The Digital Voltmeter serves as an analog-to-digital converter, and the Serializer converts the parallel binary output code of the voltmeter to a serial code. The output of the Serializer is a digital signal \( x^*(t) \) which is punched on paper tape.

The instrumentation of Figure 3-2 accomplishes the following. Sample functions of a nonstationary random process \( x(t) \) are produced by the Process Source so that each is started and ended at a fixed time with respect to the time variations of the elements in the Process Source. As each sample function is generated, it is sampled at fixed times by the analog-to-digital converter with a sampling interval \( \alpha \). The analog samples are thus converted to digital form, and the composite of these samples make up the digital signal \( x^*(t) \) which is recorded on paper tape. The final step in the measurement is to process the paper tape with a general purpose digital computer.

There are three parameters of the measurement system which must be assigned values for each measurement. These are (1) the sampling interval \( \alpha \), (2) the length, \( T \), of each sample function, and (3) the number, \( n \), of sample functions. The sampling interval is chosen by considering the highest frequency component of the signal \( x(t) \). In most cases, the smallest possible sampling interval is used and the Process Source is scale factored so that the highest frequency components of \( x(t) \) are consistent with this interval. The length \( T \) is chosen so that \( r(t, t') \) can be estimated over the ranges of \( t \) and \( t' \) of interest in each particular case. In this regard, it should be noted that both \( t \) and \( t' \) must lie in the interval \( t_o \) to \( t_o + T \). The third
parameter, \( n \), is chosen to be sufficiently large to give the required statistical accuracy in estimating the moments of \( x(t) \). The next paragraph presents a detailed discussion of the factors involved in choosing \( n \).

3.4 Sampling Errors

As discussed in the last paragraph, the final step in the measurement procedure is to process the paper tape containing the data on the output process, \( x(t) \), with a general purpose digital computer. The computer is programmed to compute estimates of the first and second moments of the \( x(t) \) process as an average over a finite number of data points. This use of an average over a finite number of samples to approximate an ensemble average leads to a sampling error which depends on the number of samples and the distribution function of the true quantity being estimated. It is the purpose of this paragraph to obtain expressions for the magnitude of the sampling error.

For generality let \( z(t) \) denote any function of \( x(t) \) whose ensemble average is to be estimated. To estimate the first moment of \( x(t) \), \( z(t) \) would be chosen equal to \( x(t) \); to estimate the second moment it would be chosen equal to \( x(t)^2 \), etc. The paper tape produced by the instrumentation system contains a finite number \( n \) of sample functions from the \( x(t) \) process and thus a quantity \( M_n(t) \) defined as

\[
M_n(t) = \frac{1}{n} \sum_{i=1}^{n} z_i(t) \tag{3.1}
\]

can be computed from the data on the tape and used as an estimate of the ensemble average \( z(t) \). The statistical properties of \( M_n(t) \) are determined by the statistical properties of \( z(t) \) and the number \( n \).
A quantity of interest is the error, $\varepsilon_M$, between $M_n(t)$ and $E z(t)$ expressed as

$$\varepsilon_M = M_n(t) - E z(t). \tag{3.2}$$

This error is a random variable and in some cases it is possible to determine its distribution function. In other cases it is not possible to determine the distribution function, and in such cases the first two moments of $\varepsilon_M$, defined as

$$E\varepsilon_M = EM_n(t) - E z(t) \tag{3.3}$$

and

$$E\varepsilon_M^2 = \varepsilon_M^2 = E \{ M_n(t) - E z(t) \}^2, \tag{3.4}$$

are adequate to give bounds on the sampling error. The first moment of $\varepsilon_M$ is frequently referred to as the bias error, and since

$$EM_n(t) = \frac{1}{n} \sum_{i=1}^{n} E z_i(t) = E z(t), \tag{3.5}$$

this error is zero. The second moment of $\varepsilon_M$ is called the mean square error. This error is frequently normalized to obtain the relative error, $B^2$, defined as

---


Most experiments result in a single evaluation of \( M_n(t) \), and thus a single realization of the random variable \( \varepsilon_M \). The most positive statement that can be made concerning a single sample of a random variable which can take on any value in a given range is the following: "With probability \( p \), \( \varepsilon_M \) lies in the interval \((a,b)\)". The probability \( p \) is often referred to as a confidence probability and typical choices of \( p \) lie in the range 0.8 to 0.99.

If the distribution function of \( \varepsilon_M \) is known, the probability

\[
P \{ a \leq \varepsilon_M \leq b \} = p(a,b)
\]  

(3.7)

This is equivalent to a statement of the form: "With confidence probability greater than \( 1 - 1/C^2 \), a single determination of \( \varepsilon_M \) yields a value whose magnitude is less than \( C[\varepsilon_M^{-2}]^{1/2} \)."

---

8 See for example Hammond [8].
The mean square error $\varepsilon_M^2$ can be expressed in terms of $n$ and the variance of $z(t)$ for any distribution function of $z(t)$ as follows. From the definition of $M_n(t)$,

$$\varepsilon_M^2 = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} E[z_i(t)z_j(t)] - [E z(t)]^2.$$  \hfill (3.9)

In most cases of importance the values of $z_i(t)$ and $z_j(t)$ from the $i^{th}$ and $j^{th}$ sample functions of $z(t)$ will be statistically independent so that

$$E z_i(t)z_j(t) = E z_i(t) E z_j(t) = [E z(t)]^2, \ i \neq j.$$  \hfill (3.10)

Thus

$$\varepsilon_M^2 = \frac{E z^2(t)}{n} - \frac{[E z(t)]^2}{n}.$$  \hfill (3.11)

But

$$E z^2(t) - [E z(t)]^2$$ is the variance, $\sigma_z^2$, of $z(t)$. Thus,

$$\varepsilon_M^2 = \sigma_z^2/n.$$  \hfill (3.12)

and $B^2$ is given by

$$B^2 = \frac{\sigma_z^2}{n[E z(t)]^2}.$$  \hfill (3.13)
Much more specific results can be obtained for the case of primary interest in this Note; namely, the case where \( x(t) \) is a Gaussian random variable. Estimates of the mean, the variance, and the covariance of \( x(t) \) will now be considered in turn with the three errors denoted respectively by \( \varepsilon_m \), \( \varepsilon_\sigma \), and \( \varepsilon_r \).

**Mean of \( x(t) \):** In this case \( z(t) \) is chosen to be equal to \( x(t) \) so that \( M_n(t) \) is given by

\[
M_n(t) = \frac{1}{n} \sum_{i=1}^{n} x_i(t) = \mu(t),
\]

and \( \varepsilon_m \) by

\[
\varepsilon_m = \frac{1}{n} \sum_{i=1}^{n} x_i(t) - E x_i(t).
\]

The moments of \( \varepsilon_m \) are thus given by

\[
E\varepsilon_m = 0 \tag{3.15}
\]

and

\[
E\varepsilon_m^2 = \frac{\sigma_x^2}{n} \tag{3.16}
\]

Since \( x(t) \) is a Gaussian random variable, it follows that \( x_i(t) - E x_i(t) \) and \( \varepsilon_m \) are also Gaussian random variables. Specifically \( \varepsilon_m \) is Gaussian with mean zero and variance \( \sigma_x^2/n \). From tabulated values of the Gaussian distribution function, bounds on \( \varepsilon_m \) with a given confidence probability can be obtained.
For example, with probability 0.8, $\epsilon_m$ lies between $\pm 1.3\sigma_x / \sqrt{n}$. The 80% confidence limits on $\epsilon_m / \sigma_x$ are plotted versus $n$ in Figure 3.3.

**Variance of $x(t)$:** To estimate the variance and covariance, $z(t)$ is chosen as $[x_i(t) - m(t)] [x_i(t') - m(t')]$ so that

$$M_n(t', t) = \frac{1}{n} \sum_{i=1}^{n} [x_i(t) - m(t)] [x_i(t') - m(t')]$$

(3.17)

The expected value of $M_n(t', t)$ is then $r(t', t)$ so that $M_n(t', t)$ is an unbiased estimate of the covariance. The error, $\epsilon_r$, is defined as

$$\epsilon_r = M_n(t', t) - r(t', t)$$

(3.18)

and this quantity can be expressed as

$$\epsilon_r = \frac{1}{n} \sum_{i=1}^{n} [x_i(t) - m(t')] [x_i(t) - m(t)] - r(t', t)$$

(3.19)

Tractable results for the statistical properties of the sampling error can be obtained when $M_n(t)$ is an estimate of the variance of $x(t)$. In this case the error is denoted $\epsilon_o$, and it can be expressed as

$$\epsilon_o = \frac{1}{n} \sum_{i=1}^{n} [x_i(t) - m(t)]^2 - \sigma_x^2$$

(3.20)

The random variable $x_i(t) - m(t)$ is Gaussian with mean zero and variance $\sigma_x^2$. The variable $\xi$ defined as
Figure 3-3. 80% Confidence Limits on Fractional Errors.
\[ \xi = \frac{n(e_{\sigma} + \sigma_x^2)}{\sigma_x^2} = \frac{n}{\sum_{i=1}^{n} \left( \frac{x_i(t) - m(t)}{\sigma_x} \right)^2} \quad , \quad (3.21) \]

is then distributed according to a chi-squared distribution\(^9\) with \(n\) degrees of freedom.

From the known properties of the chi-squared distribution

\[ E\xi = n. \quad (3.22) \]

and

\[ E(\xi - n)^2 = 2n. \quad (3.23) \]

Thus

\[ E \left( \frac{n(e_{\sigma} + \sigma_x^2)}{\sigma_x^2} - n \right)^2 = 2n, \quad (3.24) \]

from which it follows that

\[ E e_{\sigma}^2 = \frac{2\sigma_x^4}{n}, \quad (3.25) \]

and

\[ E\sigma^2 = \frac{2\sigma_x^4}{n\sigma_x^4} = \frac{2}{n} \quad . \quad (3.26) \]

\(^9\)See for example Cramér [6] p. 234.\]
The expected value of $\varepsilon_0$ is known to be zero from previous calculations and this can be verified from Equations (3.21) and (3.22).

The chi-squared distribution is tabulated so that $p(a,b)$ in Equation (3.7) can be computed as a function of $a$ and $b$. One form the results can take is given in Figure 3-3 which gives a plot of a positive and a negative bound on $\varepsilon_0/\sigma_x^2$ versus $n$ for a confidence probability of 0.8.

3.5 Summary of Results on Sampling Errors

As indicated in the development above, the curves of Figure 3-3 give bounds which apply with 80% confidence to the errors in estimating the mean or the variance of a Gaussian random variable. Specifically the dashed curve of Figure 3-3 gives a bound on the magnitude of the variable $\varepsilon_m/\sigma_x$ as a function of $n$, the number of samples used to estimate the ensemble average. As an example of the use of this curve, consider determining bounds on the sampling error in using 100 samples to approximate the ensemble mean of a Gaussian variable with a standard deviation of 2.0. Use of the curve gives a bound of 0.15 on $|\varepsilon_m/\sigma_x|$ for $n = 100$. Thus $\varepsilon_m$ will lie in the interval $(-0.3, +0.3)$ with probability 0.8. Said in another way, with 80% confidence probability the measured mean differs in magnitude from the true mean by no more than 0.3.

The solid and dotted curves of Figure 3-3 give respectively the positive and negative bounds on the variable $\varepsilon_0/\sigma_x^2$ as a function of $n$. As an example of the use of these curves, consider the estimation of the variance of a Gaussian random variable by the use of either 10 samples or 400 samples. For 10 samples the curves show that the error $\varepsilon_0/\sigma_x^2$ lies in the interval $(-0.55, 0.6)$ with confidence probability 0.8. The range for $n = 400$ is $(-0.09, 0.09)$. Stated in
another way, with probability 0.8 the measured value of the variance lies between 0.45 and 1.6 times the true value. The corresponding range for \( n = 400 \) is 0.91 to 1.09 times the true value.

The standard deviation in the two cases would lie between \( \sqrt{0.45} \) and \( \sqrt{1.6} \) times the true value for \( n = 10 \), and between \( \sqrt{0.91} \) and \( \sqrt{1.09} \) times the true value for \( n = 400 \).

### 3.6 Computations Performed by the Digital Computer

As discussed in Section III and indicated in Figure 3-2, the final step in any given measurement is to process the signal \( x^*(t) \) recorded on paper tape with a general purpose digital computer. The signal \( x^*(t) \) consists of samples taken from various sample functions of the \( x(t) \) process at various times determined by the sampling interval. Thus, assuming no errors in the analog-to-digital conversion, \( x^*(t) \) consists of the numbers

\[
x_1(t_0), x_1(t_1), x_1(t_2), \ldots, x_1(T) \\
\vdots \\
x_i(t_0), x_i(t_1), x_i(t_2), \ldots, x_i(T) \\
\vdots \\
x_n(t_0), x_n(t_1), x_n(t_2), \ldots, x_n(T)
\]

The computer is programmed to compute estimates of the mean, the variance, and the covariance of \( x(t) \) defined respectively as follows

\[
\hat{m}_x(t_v) = \frac{1}{n} \sum_{i=1}^{n} x_i(t_v),
\]

\[
\hat{\sigma}_x^2(t_v) = \frac{1}{n} \sum_{i=1}^{n} x_i^2(t_v) - \hat{m}_x^2(t_v),
\]
and

\[ f(t_u, t_v) = \frac{1}{n} \sum_{i=1}^{n} x_i(t_u) x_i(t_v) \]  \hspace{1cm} (3.29)

Note that \( f(t_u, t_v) \) is an estimate of the true covariance plus \( \hat{m}_x(t_u)\hat{m}_x(t_v) \).

In all cases discussed in this Note, the mean of \( x(t) \) was adjusted to be as nearly zero as possible. Thus, the experimental results in all cases show that \( \hat{m}_x(t_u)\hat{m}_x(t_v) \) is small with respect to statistical fluctuations, and \( f(t_u, t_v) \) is used directly as an estimate of the covariance function unless otherwise indicated. Figure 3-4 shows the format used for tabular presentation of the covariance estimates.
**Figure 3-4. Format for Tabular Presentation of Covariance Estimates.**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \hat{\gamma}(t_0, t_0) )</td>
<td>( \hat{\gamma}(t_0, t_1) )</td>
<td>( \hat{\gamma}(t_0, t_2) )</td>
<td>( \hat{\gamma}(t_0, T) )</td>
</tr>
<tr>
<td>1</td>
<td>( \hat{\gamma}(t_1, t_1) )</td>
<td>( \hat{\gamma}(t_1, t_2) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>( \hat{\gamma}(t_2, t_2) )</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>( \hat{\gamma}(T, T) )</td>
</tr>
</tbody>
</table>
IV. EXAMPLES

Example 1.

This example illustrates the use of the measuring system in experimentally determining the mean, standard deviation, and covariance function of a stationary process generated to have a prescribed covariance by passing white noise through a filter designed by the method described in Section II. The covariance chosen for this example is given by

$$r(t^0, t) = K_1 [1 + 2 (t^0 - t) e^{-2(t^0 - t)}].$$  \hspace{1cm} (4.1)

A similar filter is designed in detail as an example in paragraph 2.6 of Section II so design details are omitted here. Application of the six step procedure of paragraph 2.6, Section II leads to a differential equation describing the analog network, or what is more convenient here to a transfer function of the form

$$H(j\omega) = K_2 \left( \frac{1}{j\omega + 2} \right)^2.$$  \hspace{1cm} (4.2)

Note that since this process is stationary spectral factorization can be used to obtain the same result. The differential equation or the transfer function of Equation (4.2) can be implemented on the computer in a straightforward way.

The measured values of the mean and the standard deviation of the output process are shown in Figure 4.1. The mean and the standard deviation fit reasonably within the 80% confidence level error boundaries (shown in dashed lines) computed in Section III. Figure 4.2 shows the theoretical
Figure 4-1. Mean and Standard Deviation as a Function of Time for Example 1. n = 180.
Figure 4-2. Measured and Theoretical Covariance Functions with \( t = 1 \) for Example 1. \( n = 180 \).
covariance curve of Equation (4.1) along with several curves calculated from the measured data for different values of time. Note that the measured covariance curves differ from the theoretical covariance curve by approximately the same amount that the measured and theoretical standard deviation curves differ. Measured data, after being processed by the digital computer, is presented in the tabular format described by Figure 3.4 in Section III. Data for each calculated curve is obtained by fixing \( t \) at some value, say \( t = t_0 \), and plotting the discrete points representing \( \hat{r}(t, t') \) as a function of \( t_0 - t \) for the fixed value of \( t \). Since the process is stationary, \( \hat{r}(t', t) \) is a function of \( t_0 - t \) only and the points representing \( r(t', t) \) should fall on the theoretical curve of Equation (4.1) independent of the particular value of \( t \) chosen.

The error to be expected from the measuring system depends on the parameter values \( a \) and \( n \) defined in paragraph 3.3 of Section III. The results of Figures 4.1 and 4.2 were obtained with \( a = 1 \) second and \( n = 180 \). A measurement with \( a = 1 \) second and \( n = 18 \) leads to the results given in Figures 4.3 and 4.4. These results, as expected, have a much larger statistical fluctuation than the curves for \( n = 180 \); however, the 80% confidence limit error boundaries are still indicative of the errors encountered.

Example 2

This example illustrates the synthesis procedure for deriving a time-varying filter which with a white noise input delivers an output with a prescribed nonstationary second moment, and also demonstrates the use of the measuring system in ascertaining that the output has the desired second moment.
Figure 4-3. Mean and Standard Deviation as a Function of Time for Example 1. \( n = 18 \).
Figure 4-4. Measured and Theoretical Covariance Functions with $t=1$ for Example 1. $n = 18$. 

---

**Covariance Function**

- THEORETICAL CURVE
- MEASURED POINTS, $t=0$
- MEASURED POINTS, $t=1$
- MEASURED POINTS, $t=2$

**Figure 4-4.** Measured and Theoretical Covariance Functions with $t=1$ for Example 1. $n = 18$. 

---

42
The covariance function to be generated in this example is given by

\[ r(t^0, t) = \frac{1}{t} \]

(4.3)

The filter design is carried out by following the six step procedure of paragraph 2.6 of Section II.

1. A comparison of Equation (4.3) with Equation (2.4) yields the result

\[ \phi_1 = \frac{1}{t}, \gamma_1 = 1, \]

\[ \phi_i = \gamma_i = 0, i \neq 1. \]

(4.4)

2. (a) Equation (2.7) is used to give

\[ L_t x = \begin{bmatrix} x & \frac{1}{t} \\ \dot{x} - \frac{1}{t^2} & 0 \end{bmatrix} = 0 = \left( \frac{\ddot{x}}{t} + \frac{x}{t^2} \right) \]

which simplifies to

\[ \ddot{x} + \frac{1}{t} x = 0. \]

(4.5)

(b) \( a_o(t) = p_o(t) \) from Equation (2.12), and

(c) \( \phi(t) = \phi_1(t) = \phi_{11}(t) \) from Equation (2.15).
3. Equation (2.23) gives the result

\[ D = d_{11} = \frac{1}{t} = t. \]  

(4.6)

4. Equations (2.21), (2.24), and (2.25) determine

\[ R = \phi(t')D(t)\phi^T(t) = \frac{1}{t} \quad \frac{1}{t} \quad \frac{1}{t}, \]  

(4.7)

\[ R^* = \frac{1}{t} \]  

(4.8)

and

\[ \Delta = \frac{1}{t} - \frac{1}{t}. \]  

(4.9)

5. Use of Equations (2.26) and (2.27) allows the calculation of \( b_0 \) as

\[ \frac{\partial \Delta}{\partial t'} \bigg|_{t'=t} = -\frac{1}{t^2} = -b_o^2 \]

or

\[ b_o = \frac{1}{t}. \]  

(4.10)

6. The resultant differential equation is given by

\[ \ddot{x} + \frac{1}{t} \dot{x} = \frac{1}{t} y. \]  

(4.11)
where y is the white noise input and x is the desired nonstationary output process with covariance \( r(t^0, t) = \frac{1}{t} \).

The synthesis procedure is complete at this point and the remaining discussion concerns implementing the filter on the analog computer and measuring the moments of the generated process.

A straightforward implementation of Equation (4.11) is shown in Figure 4.5. Since \( \frac{1}{t} \) is unbounded at \( t=0 \), the solution must start at some \( t \) greater than zero, and \( t_0 = 1 \) was chosen in this case. In the circuit of Figure 4.5, \( x(t=1) \) is the required integrator initial condition. Since \( x(t=1) \) is a random variable proportional to \( \int_0^1 y dt \), it is difficult to apply this initial condition. To circumvent this difficulty it was convenient to implement the computer circuit as shown in Figure 4.6 rather than in the form of Figure 4.5.

The computer diagram of Figure 4.6 is derived by making the substitution \( x = uv \) in

\[
\dot{x} + \frac{x}{t} = \frac{y}{t} \quad (4.11)
\]

which gives

\[
u \left( \frac{dv}{dt} + \frac{v}{t} \right) + (v \frac{du}{dt} - \frac{v}{t}) = 0. \quad (4.12)
\]

Each part of Equation (4.12) is solved separately to give

\[
v = \frac{1}{t} \quad (4.13)
\]

and

\[
u = \int_0^t y dt + c. \quad (4.14)
\]
Figure 4.5. Implementation of $\dot{x} + \frac{x}{t} = \frac{y}{t}$.

Figure 4.6. Improved Implementation of $\dot{x} + \frac{x}{t} = \frac{y}{t}$.

$v(t) = \frac{100}{t}$
In the circuit of Figure 4.6 \( v \) is held constant at 100V over the first second while the integrator starts at \( t=0 \) with an initial condition of 0. This sets \( c=0 \) and produces \( u = \int_0^1 y dt \) as the integrator output voltage at \( t=1 \). The final output, \( x = uv \), is thus correct for the interval \( 1 \leq t \leq 10 \).

The measuring system described in Section III, paragraph 3.3 when used to measure the output process \( x(t) \) simulated on the analog computer in the manner shown in Figure 4.6, gives the results shown in Figures 4.7 and 4.8. Figure 4.7 shows the mean and the process standard deviation about the mean. The theoretical value of the mean is zero and the theoretical curve for the standard deviation is obtained from

\[
\sigma(t) = \frac{1}{t} \int_0^t \phi(t', t) dt,
\]

where \( \phi \) is the power spectral density of the white noise. The 80% confidence level is indicated about the theoretical curve by the dashed lines. Note that the measured points check very well with respect to the expected error limits. Figure 4.8 shows the covariance function, \( r(t', t) \), as a function of \( t'-t \) with \( t=1 \). The theoretical curve is calculated from

\[
r(t', t) = 1 - \frac{1}{t'}.
\]
Figure 4-7. Normalized Mean and Standard Deviation as a Function of Time for Example 2. n = 150.
Figure 4-8. Measured and Theoretical Covariance Functions with $t=1$ for Example 2. $n = 150$. 

THEORETICAL CURVE

MEASURED POINTS

NORMALIZED COVARIANCE FUNCTION

$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$

$t' - t$ (seconds)
Note that the points produced by the measuring system follow the theoretical curve with an error about the same as that observed in the standard deviation measurement.

**Example 3**

The third example realizes a more complicated covariance function of the form

\[ r(t, t') = \frac{t}{(t^2 + 1)(t + 1)} \]  \hspace{1cm} (4.16)  

The theoretical equation for the standard deviation is given by

\[ \sigma(t) = \frac{\sqrt{t}}{t+1} \]  \hspace{1cm} (4.17)  

Application of the synthesis procedure leads to

\[ \dot{x} + x \left( \frac{1}{t+1} \right) = y \left( \frac{1}{t+1} \right) \]  \hspace{1cm} (4.18)  

as the defining equation of the desired filter. This equation is implemented in a straightforward manner by the computer diagram of Figure 4.9.

The measuring system, when used to measure the output process \( x(t) \) generated as shown in Figure 4.9, gives the results shown by the dots in Figures 4.10 and 4.11. Figure 4.10 shows the mean, whose theoretical value is zero, and the standard deviation about the mean for which the theoretical curve is given by Equation (4.17). The 80% confidence level error is indicated about the theoretical curves by the dashed lines. Figure 4.11 shows the
Figure 4.9. Implementation of \( x + x \left( \frac{1}{t+1} \right) = y \left( \frac{1}{t+1} \right) \).
Figure 4.10. Normalized Mean and Standard Deviation as a Function of Time for Example 3. n = 150.
Figure 4.11. Measured and Theoretical Covariance Functions
With $t=1$ and $t=3$ for Example 3. $n = 150$. 

53
covariance function \( r(t', t) \) as a function of \( t' \) for two choices of \( t \) with the theoretical curves of Equation (4.16) plotted as solid curves. Note again that the measured points follow the theoretical curve with an error of the same order of magnitude as that observed in the standard deviation measurement.
V. CONCLUSIONS

Presented in this Technical Note are methods for the generation and measurement of nonstationary random variables. The desired random variables are generated using analog computer circuitry and a Gaussian white noise source. A technique for synthesizing the computer circuit from the specified first two statistical moments of the required process has been developed and its correctness and utility have been demonstrated.

A specified Gaussian process, which is determined completely by its first two statistical moments, is realized exactly by this procedure. If the required process is non Gaussian, the procedure generates an approximation to the required process having the same first two moments as the required process.

A method for measuring and processing the generated random variables is also presented. An error analysis which yields confidence estimates for the measurement and processing operations is included. Several examples which demonstrate the utility of the generation and measuring techniques have been completed and in each case the random processes generated show an excellent agreement between prescribed and measured parameters when the results are subjected to the computed error bounds.

Further work in connection with this study will be devoted to the analytical representation of nonstationary processes. In particular, methods for constructing appropriate analytical representations of the first two statistical moments from measured data on nonstationary processes will be investigated. Also, additional effort will be devoted to investigating possible extensions and simplifications of the results presented in this Note.
References


TECHNICAL NOTE NO. 4
Project No. A-588

BIBLIOGRAPHY AND SUMMARY OF METHODS
RELATED TO THE ERROR ANALYSIS OF HYBRID COMPUTERS

By Thomas M. White, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation)

14 October 1964

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
BIBLIOGRAPHY AND SUMMARY OF METHODS RELATED TO THE ERROR ANALYSIS OF HYBRID COMPUTERS

By Thomas M. White, Jr.

TECHNICAL NOTE NO. 4 on Contract No. NAS8-2473 (Development of New Methods and Applications of Analog Computation) for GEORGE C. MARSHALL SPACE FLIGHT CENTER Huntsville, Alabama
BIBLIOGRAPHY AND SUMMARY OF METHODS RELATED TO THE ERROR ANALYSIS OF HYBRID COMPUTERS

Introduction

A general approach to error analysis for any kind of computing (or problem solving) equipment is very difficult, if not impossible, to achieve.

Most straightforward approaches result in equations for the error which are an order of magnitude more complicated than the original equations. Much work, however, has been done toward specialized error analysis for analog computers and sampled-data systems. Since the field of hybrid computation has developed rapidly only in the last several years, less work has been done in this area. It is the purpose of this note to summarize, or at least list, the methods (some which have been used solely with analog computers, some which have been used solely with sampled-data systems and some which have been used with hybrid computer systems) most likely to lead to useful results in error analyses of hybrid computers.

In addition to the errors caused by non-ideal passive elements, finite gain amplifiers, drift in amplifiers, offset, phase shift, and improper setting of coefficients associated with analog computers the hybrid computer has errors caused by the digital section as well. Among the errors contributed by the digital section are errors due to quantization, truncation, sampling interval, sampling time, and time delay.

An ideal error analysis would be short, tractable, and accurate. It must, furthermore, answer many questions about the system being analyzed. Some typical questions are:

1. What effect does the deviation in the value of one or more elements have on the static and dynamic error of the system?
2. What is the effect of the discrete data on the static and dynamic error of the system?
3. How should analog and digital sections be chosen for use together so that the static and dynamic error of the hybrid system satisfies certain error requirements?
4. What effect does noise have on the static and dynamic error of the system?

The following sections give current methods of attacking the error problems present in analog computers, sampled-data systems, and hybrid computers.
Error Analysis of Analog Computers

A good survey of methods developed for error analysis of analog computers is given by chapter 3 of Tomovic and Karplus*. A similar chapter appears in Eterman**. A brief description of some of the most widely used methods is presented in this note. Details are available from the references cited.

A general approach to error analysis of analog computation has been developed by Miller and Murray7. Their method of analysis applies when a computer is solving a system of linear or nonlinear differential equations of the form

\[ F_i(\dot{x}_1, \ldots, \dot{x}_n, x_1, \ldots, x_n, t) = 0, \quad i = 1, \ldots, n. \]

(1)

The system (1) is realized on the machine in the form

\[ G_i(\dot{x}_1, \ldots, \dot{x}_n, x_1, \ldots, x_n, t, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_p) = 0, \]

\[ i = 1, \ldots, n. \]

(2)

The \( \alpha \), \( \beta \), and \( \gamma \) values account for perturbation errors in the system components. The \( \alpha \) and \( \beta \) values do not change the order of the system but the \( \gamma \) values do.

The solution of system (2) has the general form

\[ x_i = x_i(t, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_p) \]

(3)

Since the solution (3) depends analytically on the \( \alpha \), \( \beta \), and \( \gamma \) values, the solution can be expanded in a power series where partial derivatives of \( x_i \) with respect to \( \alpha \), \( \beta \), and \( \gamma \) appear in the expansion. These partial derivatives are termed sensitivity coefficients and calculation or measurement of these sensitivity coefficients allows the errors in the solution to be expressed by means of Taylor-series expansions. The partial derivatives may be calculated by solving a linear system of differential equations whose homogeneous part is derived from the original equations. The partial derivatives may be measured by means of an experimental approach. They can also be obtained from the computer solution of the system of differential equations derived from the original equations.


** Superscripts refer to Bibliography reference numbers.
When the system is restricted to be linear the error analysis becomes more straightforward and easier to handle. Many authors have used a Laplace transform approach which transfers the analysis of the linear system from the time domain to the complex-frequency domain. The characteristic equation of an $n^{th}$ order differential equation can be expressed as

$$\sum_{i=0}^{n} a_i S^i = 0.$$  \hspace{1cm} (4)

If the coefficients of equation (4) are perturbed a small amount, a new equation of the same form is obtained. The roots of the new equation are shifted with respect to the roots of the original equation. The root shifts can be determined by methods proposed by Macneee, Marsocci, Miura and Nagata, and others. Probably the most sophisticated analysis of this type is that due to Miura and Nagata.

Work by Nelson is also of interest. The approach refers primarily to Miller and Murray; sensitivities are defined; and root shifts are considered.

Error Analysis of Sampled-Data Systems

Extensive literature is available on the analysis of sampled-data systems but only a limited amount is available on the error analysis problem. It is impossible to summarize and perhaps even to identify all the work which might be of future use in the analysis of hybrid systems. Summaries of some of the various approaches follow.

Kuo proposes the use of state transition flow graphs for analysis. The state transition flow graph of a continuous-data system is defined as the analog computer simulation flow graph of the system with initial conditions applied at corresponding nodes of the state variables as input signals. The state transition flow graph of a discrete-data system is similar but it contains arithmetic operations portraying digital programming. The two state transition flow graphs are combined for a system with discrete and continuous data. In the actual mathematical analysis $Z$ transform theory is used.

Errors introduced by sampling and quantization are discussed by Nelson, Katzenelson, Widrow, Turtle, and others. The paper by Nelson develops the mechanics of error in a digital differential analyzer and he applies the result of this development to rectangular and trapezoidal integration. The $W$ transform is proposed for use in general analysis and several examples are included showing how the errors cause a shift in the roots of the characteristic equation.
The work by Katzenelson\textsuperscript{3} evaluates the mean-square error, caused by sampling and quantization at the output of a linear network which contains a single quantizer.

Widrow\textsuperscript{12} presents a statistical description of the quantization process. Several examples of closed loop systems containing quantizers in various locations are shown and an error analysis is included.

A statistical analysis of round-off error and a detailed discussion of truncation error is presented in the work by Turtle\textsuperscript{11}. Analytical simplicity is obtained by representing an integrator by a transform so that block diagram algebra applies. Generation of an exponential is used as an example. The error bounds determined through analysis of the increasing exponential serve as error bounds for a large class of linear problems.

Synthesis procedures for sampled-data systems and the use of $Z$ transforms are subjects covered by numerous authors, many of whom are included in the bibliography.

**Error Analysis of Hybrid Computers**

Since hybrid computers combine analog and digital systems, the errors of the overall system come from the errors inherent in each individual system but complications arise because of interaction between the systems. The methods of analysis applicable to analog and digital systems individually do not generally apply without modification to the hybrid computer although there is reason to expect modified forms of the methods could apply.

Very few papers in the literature deal directly with the effects of digital computation when it is used as part of a closed loop hybrid system. A promising article in this area is "Effects of Digital Execution Time in a Hybrid Computer" by Miura and Iwata\textsuperscript{9}. This analysis is based on approximating the transfer function of the digital computer section by a function of the form

$$D(s) = Ae^{-(\tau + \frac{T}{2})s} \quad (5)$$

where $\tau$ represents a time delay due to digital computing time and $T$ is the sampling period. This transfer function, when expanded in a power series, is used in conjunction with the characteristic equation of an ideal analog computer representation to derive a new characteristic equation which includes the effect of the digital system. The root shift of the hybrid system with respect to the ideal analog system is a measure of the error due to the digital section.
In summary, the survey of the literature indicates that the best approaches to error analysis of hybrid computers are probably:

1) The use of sensitivity coefficients based on a modification of the method of Miller and Murray; and

2) The use of root shifting techniques in linear problems or problems which can be linearized.

In both approaches approximations for transfer functions of different components will be useful and the use of a variety of mathematical tools such as Z transforms will be helpful.
Bibliography

The following references are separated into three categories:

I. References most directly concerned with error analysis of hybrid computers:


II


TECHNICAL NOTE NO. 5
Project No. A-588

NOTES ON A CLASS OF STOCHASTIC PROCESSES
FOR USE IN ANALOG SIMULATION STUDIES

By David L. Finn

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

6 December

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
NOTES ON A CLASS OF STOCHASTIC PROCESSES
FOR USE IN ANALOG SIMULATION STUDIES

By
David L. Finn

TECHNICAL NOTE NO. 5
on
Contract No. NAS8-2473
(Development of New Methods and Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Page</td>
<td>1</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>iii</td>
</tr>
<tr>
<td>I. INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>II. BACKGROUND OF THE RESEARCH</td>
<td>2</td>
</tr>
<tr>
<td>III. PARTIALLY-STATIONARY STOCHASTIC PROCESSES</td>
<td>4</td>
</tr>
<tr>
<td>IV. FIRST-ORDER DISTRIBUTIONS FOR PARTIALLY-STATIONARY STOCHASTIC PROCESSES</td>
<td>12</td>
</tr>
<tr>
<td>V. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH</td>
<td>18</td>
</tr>
<tr>
<td>References</td>
<td>21</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.1</td>
<td>Probability Density Function for Phase Angle</td>
</tr>
<tr>
<td>3.2</td>
<td>Power Spectral-Density for Example 3.1</td>
</tr>
<tr>
<td>3.3</td>
<td>Power Spectral-Density for Example 3.2</td>
</tr>
<tr>
<td>3.4</td>
<td>Power Spectral-Density for Sample Functions of Example 3.3</td>
</tr>
<tr>
<td>3.5</td>
<td>Power Spectral-Density for Example 3.4</td>
</tr>
<tr>
<td>4.1</td>
<td>Altitude and Random Disurbance as Functions of Time for k Rocket Flights</td>
</tr>
<tr>
<td>4.2</td>
<td>Generation of $E[x_s(t)]$</td>
</tr>
<tr>
<td>5.1</td>
<td>Mechanization of a Partially-Stationary Stochastic Process</td>
</tr>
</tbody>
</table>
I. INTRODUCTION AND SUMMARY

This technical note is concerned with one aspect of the problem of generating and using nonstationary stochastic processes in Monte Carlo studies with an analog computer. The work covered by this note is a continuation of that reported in Technical Note 1 and Technical Note 2 of the current project.¹ ²

A central consideration in any use of stochastic process theory is a careful definition of a sample space and appropriate random variables. This technical note considers the physical problem of random wind perturbations of a missile trajectory, and defines a general mathematical model which describes the random wind forces. The class of stochastic processes defined by the model are termed partially-stationary stochastic processes. A single partially-stationary stochastic process which will generate the random disturbance encountered in a wide variety of individual missile flights is developed.

This technical note is subdivided into five sections. In Section II the relationship between the material presented in this note and that in previous notes is discussed.

Section III is devoted to the definition of a partially-stationary stochastic process and to the consideration of an example selected to provide physical insights into implications contained in the definition.

Some properties of first-order probability distributions for partially-stationary stochastic processes are investigated in Section IV. This is done by considering the physical problem of simulating a random disturbance affecting a rocket in flight.

The final section, Section V, is utilized for the presentation of conclusions and recommendations for further research.
II. BACKGROUND OF THE RESEARCH

Technical Note 3 for this project is concerned with the problem of generating a nonstationary stochastic time process having prescribed first and second moments. This is accomplished by means of the decomposition:

\[ X(t) = X^*(t) + m(t) \]  

(2.1)

where:

- \( X(t) \): stochastic process having prescribed first and second statistical moments.
- \( m(t) = E[X(t)] \): first statistical moment of \( X(t) \).
- \( X^*(t) \): stochastic process having \( E[X^*(t)] = 0 \).

The function \( m(t) \) is generated as a fixed, prescribed function of time and added to \( X^*(t) \). In turn, \( X^*(t) \) is generated by passing a stationary Gaussian stochastic process, having mean value of zero and a white-noise spectrum, through an appropriate time-varying filter. The characteristics of the time-varying filter are dependent on the second statistical moment (covariance) of \( X(t) \).

The process \( X(t) \) generated by the procedure developed in Technical Note 3 is a Gaussian stochastic process. In Technical Note 3, it is pointed out that for any stochastic process having given mean and covariance, there exists a Gaussian process with identical mean and covariance. Thus, if only the first two statistical moments of a stochastic process are of interest in a simulation study, the generation of an appropriate Gaussian process will suffice.

Typically, the stochastic process \( X(t) \) might be used in the analog simulation of a rocket flight. The process would perhaps represent a random disturbance encountered by the rocket, such as a force due to buffeting winds. If this should be the case, it is apparent that the statistical characteristics of the stochastic
process \( X(t) \) would depend on a great many factors. These factors conceivably include such variables as position of the rocket (particularly altitude), velocity, and acceleration. They may include variables characterizing the atmosphere such as temperature, general level of turbulence, or even season of the year.

It seems unlikely that a general theoretical formulation, tractable enough to be of practical use, could be obtained for a stochastic process dependent on all variables of the type just mentioned. On the other hand, in order to simulate a given rocket flight it is obviously undesirable to be compelled to first perform the actual flight in order to experimentally determine the statistical properties of the random disturbances encountered. A reasonable goal seems to be the development of a theoretical model that adequately represents the random disturbances encountered on a rocket flight that occurs under a restricted set of idealized conditions. This model could then be used to predict the random disturbances that would occur on any flight where actual flight conditions are judged to be reasonably approximated by the idealized conditions.

The purpose of this technical note is the discussion of a class of stochastic processes, called partially-stationary processes, that appears to be appropriate for use in analog simulation studies similar to the one just mentioned.
III. PARTIALLY-STATIONARY STOCHASTIC PROCESSES

The definitions of random variable and stochastic process to be used in this technical note are:

**Definition 3.1 - Random Variable**

A random variable \( n_s \) is a finite, real-valued function defined on an abstract probability space \( S \) of individual elements \( s \). This function has the property that, for every fixed real number \( a \), the set of points \( s \) for which \( n_s \leq a \) is one of the sets in \( S \) for which a probability is defined. This probability is denoted as \( P[n_s \leq a] \).

**Definition 3.2 - Stochastic Process**

A stochastic process \( n_s(w) \) is a collection of random variables indexed by some arbitrary parameter set \( W \) of individual elements \( w \).

A class of stochastic processes useful in many physical problems is the class of strictly-stationary stochastic processes.

**Definition 3.3 - Strictly-Stationary Stochastic Process**

Let \( W \) be the set of all real numbers or some subset closed under the operation of addition. The stochastic process \( n_s(w) \) is strictly-stationary if the random variables \( n_s(w_1), n_s(w_2), \ldots, n_s(w_m) \) have the same joint probability distribution as the random variables

\[
  n_s(w_1 + h), n_s(w_2 + h), \ldots, n_s(w_m + h)
\]

for any positive integer \( m \), any \( w_1, w_2, \ldots, w_m \) and all \( h \) in \( W \).

A strictly-stationary stochastic process can be described as a process for which the probability structure is unaffected by change of the \( w \)-origin. The structure is invariant under parameter translation.

The purpose of this technical note is the discussion of a class of processes that will be designated here as partially-stationary stochastic processes.
Let \( w = (u_1, u_2, \ldots, u_k; v_1, v_2, \ldots, v_e) \) be a multidimensional parameter. Each \( u_i \) is a member of the set \( U_i \) which is either the real-number system or a subset closed under addition. Each \( v_j \) is a member of the set \( V_j \) which is either the real number system or an arbitrary subset. Let each \( w_q \), with \( q \) a positive integer, be a particular selection of the multi-dimensional parameter \( w \). Thus,

\[
w_q = (u_{1q}, u_{2q}, \ldots, u_{kq}; v_{1q}, v_{2q}, \ldots, v_{eq})
\]

where each \( u_{iq} \) is an arbitrary number in \( U_i \) and each \( v_{jq} \) is an arbitrary number in \( V_j \). Using this notation, there results:

**Definition 3.4 - Partially-Stationary Stochastic Process**

The stochastic process \( n_s(w) \) is partially stationary if the random variables

\[
\begin{align*}
    n_s(w_1) &= n_s(u_{11}, \ldots, u_{i1}, \ldots, u_{ik}; v_{11}, \ldots, v_{1e}) \\
    n_s(w_2) &= n_s(u_{12}, \ldots, u_{i2}, \ldots, u_{ik}; v_{12}, \ldots, v_{2e}) \\
    &\vdots \\
    n_s(w_m) &= n_s(u_{1m}, \ldots, u_{im}, \ldots, u_{ik}; v_{1m}, \ldots, v_{em})
\end{align*}
\]

have the same joint probability distribution as the random variables

\[
\begin{align*}
    n_s(u_{11}, \ldots, u_{i1} + h, \ldots, u_{ik}; v_{11}, \ldots, v_{1e}) \\
    n_s(u_{12}, \ldots, u_{i2} + h, \ldots, u_{ik}; v_{12}, \ldots, v_{2e}) \\
    &\vdots \\
    n_s(u_{1m}, \ldots, u_{im} + h, \ldots, u_{ik}; v_{1m}, \ldots, v_{em})
\end{align*}
\]

for any positive integer \( m \), for any positive integer \( i = 1, 2, \ldots, k \), for any defined \( w_1, w_2, \ldots, w_m \), and for all \( h \) in \( U_i \).

A partially-stationary stochastic process can be described as a process for which the probability structure is unaffected by change of the origin for any number of \( u_i \) components. However, the structure may be affected by change in the origin for any \( v_j \) component.
It is seen that the process generated by a partially-stationary \( n_s(w) \) when all component parameters are held fixed except a single \( u_i \) is strictly-stationary in that \( u_i \). Also it is seen that the process generated when all component parameters are held fixed except a single \( v_j \) has a probability structure that is identical for all possible selections of the fixed \( u_i \) components.

In a physical problem, say the simulation of a random disturbance encountered during the flight of a rocket, the component parameters \( u_i \) and \( v_j \) may represent quantities such as time, position, velocity, and acceleration. These quantities, in turn, are all functions of time. Thus, when used in a problem of this type, a partially-stationary process \( n_s(w) \) generates a stochastic process in time \( X_s(t) = n_s(u_1(t), \ldots, u_k(t); v_1(t), \ldots, v_e(t)) \). The process \( X_s(t) \) is, in general, non-stationary in time. The process \( X_s(t) \), rather than the process \( n_s(w) \), is ordinarily an essential component in the overall simulation problem. The process \( n_s(w) \) is of use only to generate the process \( X_s(t) \).

Insight into implications of Definition 3.4 may be gained by consideration of several examples. These examples were selected because of their mathematical simplicity and are not intended to represent actual physical situations.

Suppose that the electric field intensity (magnitude) throughout space can be described by the formula

\[
\sin(x) \cos(y) \cos(t + z + s)
\]  

(3.1)

Here:

\[
x, y, z \text{ are position coordinates of a rectangular coordinate system.}
\]

(3.2)

\[t = \text{time variable}\]

\[s = \text{element of a probability space.}\]

The variables \( x, y, z, t, s \) are allowed to be arbitrary real numbers.

The presence of the variable \( s \) in the formula (3.1) accomplishes the purpose of describing the electric field intensity as a stochastic process. At a fixed
point in space, the field intensity waveform for this stochastic process has a random phase angle in time. The range of all real numbers for $s$ describes all possible phase angles. A particular realization (sample function) for the stochastic process is obtained by selection of a specific value for $s$. The particular realization represents a single electric field out of the infinite number of possible electric fields.

Let:

$$ w = (u_1, u_2; v_1, v_2) = (t, z; x, y) $$

where

$$ u_1 = t $$
$$ u_2 = z $$
$$ v_1 = x $$
$$ v_2 = y $$

It can be shown that the stochastic process $n_s(w) = n_s(t, z; x, y)$ is partially-stationary provided that the probability density function $p(s)$ shown in Figure 3.1 is assigned to $s$.

$$ n_s(w) = n_s(t, z; x, y) = \sin(x)\cos(y)\cos(t + z + s) $$

is partially-stationary provided that the probability density function $p(s)$ shown in Figure 3.1 is assigned to $s$.

![Figure 3.1. Probability Density Function for Phase Angle.](image-url)
Now let is be supposed that a small probe, measuring the electric field intensity, is moved through space. The position of the probe is described by three functions of time $x(t)$, $y(t)$, and $z(t)$.

It is seen that the electric field intensity $E_s(t)$ measured by the probe may be described as a stochastic process in time generated by $n_s(w)$.

$$E_s(t) = n_s(t, z(t); x(t), y(t))$$

$$= \sin x(t) \cos y(t) \cos (t + z(t) + s). \quad (3.4)$$

The process $E_s(t)$, as will be shown by several cases for the motion of the probe, may be either stationary or nonstationary.

**Example 3.1**

$x(t) = \frac{\pi}{2}, \quad y(t) = 0, \quad z(t) = 0$

$$E_s(t) = \cos (t + s) \quad (3.5)$$

This process is strictly-stationary and ergodic. Each sample function is a sinusoid. The ensemble power spectral-density for the process is shown in Figure 3.2.

![Power spectral density for Example 3.1.](image)

**Example 3.2**

$x(t) = \frac{\pi}{2}, \quad y(t) = 0, \quad z(t) = 2t$

$$E_s(t) = \cos (3t + s) \quad (3.6)$$
This process is strictly-stationary and ergodic. Each sample function is a sinusoid. The power spectral-density for the process is shown in Figure 3.3.

![Power spectral-density](image)

**Figure 3.3.** Power spectral-density for Example 3.2.

**Example 3.3**

\[ x(t) = \frac{\pi}{2}, \quad y(t) = 2t, \quad z(t) = 0 \]

\[ E_x(t) = \cos(2t) \cos(t + s) \]
\[ = 0.5 \cos(3t + s) + 0.5 \cos(t - s) \quad (3.7) \]

Each sample function is the sum of two sinusoids. This process is not strictly-stationary as can be seen by an evaluation of the covariance function \( C(t, T) \) for two selections \( t_1, t_2 \) of the variable \( t \).

Let:

\[ t_1 = -\frac{\pi}{8} \]
\[ t_2 = 0 \]
\[ T = \frac{\pi}{4} \quad (3.8) \]

Thus,

\[ C(-\frac{\pi}{8}, \frac{\pi}{4}) \]
\[ = \int_{-\infty}^{\infty} E_s(-\frac{\pi}{8}) E_s \left(+\frac{\pi}{8}\right) p(s) \, ds \quad (3.9) \]
\[ = \frac{1}{4\sqrt{2}} \]
The covariance function is not invariant with the observation time $t_0$. Thus, the process is not strictly-stationary, and further the ensemble power spectral-density of the process is not defined.

It is of interest to note that, because each sample function is a sum of sinusoids, the power spectral density (using a definition involving the time correlation function $1$) is defined for every sample function. Inspection of (3.7) shows that it is identical for every sample function. This power spectral-density for each sample function is shown in Figure 3.4.

![Figure 3.4. Power spectral-density for sample functions of Example 3.3.](image)

**Example 3.4**

$x(t) = \frac{\pi}{2}$, $y(t) = 0$, $z(t) = -t$

$E_s(t) = \cos(s) \quad (3.11)$

This process is strictly-stationary in time. However, it is not ergodic. The covariance function of the process is
The ensemble power spectral-density is shown in Figure 3.5.

\[ C(t, T) = \int_{-\infty}^{\infty} E_s(t) E_s(t + T) p(s) \, ds \]
\[ = \frac{1}{2\pi} \int_{0}^{2\pi} \cos^2(s) \, ds = \frac{1}{2} \]  

(3.12)

Each sample function is a constant in time. This is because the motion of the probe follows exactly the wave motion of the electric field intensity. The magnitude of the field intensity of each sample function varies with the cosine of the random phase angle \( s \). Thus, time averages vary from one sample function to another.

The four examples used here to describe the motion of the probe show that the processes \( E_s(t) \) generated from \( n_s(\omega) \) have statistical properties that differ greatly for different probe movements. Further study shows that the processes \( E_s(t) \) are nonstationary in time for most of the possible paths that can be assumed for probe motion.
In this section some of the properties of first-order distributions for partially-stationary processes will be discussed. To help give physical insight, this will be done in terms of a specific physical example.

Suppose that the partially-stationary process

\[ n_s(w) = n_s(t, x, y; z) \] (4.1)

represents a random disturbance encountered by a rocket in flight.

Here:

\[ w = (t, x, y; z) \]
\[ t = \text{time} \] (4.2)
\[ x, y, z = \text{position of rocket at time } t \text{ described in terms of a rectangular coordinate system. The variable } z \text{ is altitude.} \]

It is assumed that all the sample functions of the process are real-valued. The variables \( t, x, \) and \( y \) are allowed to be any real number. The variable \( z \) is allowed to be any non-negative real number.

For a particular selection of \( s \), say \( s = s_1 \), the sample function \( n_{s_1}(t, x, y; z) \) represents the disturbance encountered on the flight of a particular rocket \( s_1 \).

The set of all \( s \) is the set of all conceivable rockets having disturbance in flight adequately described, in a statistical sense, by the specific stochastic process \( n_s(w) \).

It will be assumed that the first-order probability distributions for \( n_s(w) \) are specified by probability density functions:

\[ p[n_s(w)] = p[n_s(t, x, y; z)] \] (4.3)

Because the process is partially-stationary this may be written as:
\[ p [n_s(t, x, y; z)] = p [n_s(t + a, x, y; z)] = p [n_s(t + a, x + b, y + c; z)] \]

where \( a, b, \) and \( c \) are arbitrary real numbers. For any fixed selection \( t_o, x_o, y_o, z_o \) these real numbers may be taken as

\[ a = -t_o, \quad b = -x_o, \quad c = -y_o. \]

There results:

\[ p [n_s(t_o, x_o, y_o; z_o)] = p [n_s(0, 0, 0; z_o)] \]

It is thus seen that the probability density function \( p [n_s(w)] \) depends only on altitude. It is not dependent on \( t, x, y. \) This means that all statistical parameters whose calculation involves only the first-order distributions depend only on altitude. For example, the mean value and variance for the stochastic process

\[ X_s(t) = n_s(t, x(t), y(t); z(t)) \]

generated by a particular rocket flight depend only on the altitude time-function \( z(t). \)

\[ E[X_s(t)] = m_x[z(t)] \]
\[ E[(X_s(t) - m_x[z(t)])^2] = \sigma_x^2[z(t)] \]

For different selections of \( s \) the sample functions of the stochastic process

\[ X_s(t) = n_s(t, x(t), y(t); z(t)) \]

generated by particular flights of the rockets \( s \) afford only partial realizations (or sample functions) for the stochastic process \( n_s(w). \) A sample function \( n_s(w) \) for a fixed rocket \( s_1 \) is completely specified only when the numerical value of the random disturbance is specified for all real values of \( t, x, y, \) and all non-negative real values of \( z. \) The sample function \( n_{s_1}(t, x(t), y(t); z(t)) \) specifies
only a portion of these numerical values that depends on the trajectory of the flight taken by the rocket $s_1$. In spite of this, the fact that first-order distributions depend only on altitude makes it possible to estimate all statistical parameters depending only on the first-order distributions from data taken on individual rocket flights. The procedure for making estimates will now be explained by considering an estimate for $E\left[n_s(t, x, y; z)\right] = m_n(z)$.

A single realization (sample function) $n_{s_1}(t, x, y; z)$ describes the random disturbance encountered by a particular rocket $s_1$ in flight in terms of a fixed coordinate system for time and position. Conceptually, another realization is obtained by considering the random disturbance encountered by a different rocket with respect to exactly the same coordinate system. Because the process is partially-stationary, the selection of the origins for the variables $t, x, y$ is arbitrary in the sense that it does not affect any statistical parameters. This fact makes it plausible to assume that the origins for the variables $t, x, y$ may be different, and arbitrary, for the determination of different sample functions of $n_s(w)$. Even further, it seems plausible to assume that two distinct partial realizations of the process $n_s(w)$ are obtained by two different flights of the same rocket $s_1$. Here, it would of course be necessary to require that the same general atmospheric conditions were present for both flights.

To estimate $m_n(z)$, the mean value of $n_s(w)$, suppose that the disturbance

$$X_{s_1}(t) = n_{s_1}(t, x_1(t), y_1(t); z_1(t))$$

and the altitude $z_1(t)$ for rocket flights $i = 1, 2, \ldots, k$ are recorded as functions of time as shown in Figure 4-1.
Figure 4.1. Altitude and random disturbance as functions of time for k rocket flights.
The estimate for $E[n_s(t_o, x_o, y_o; z_o)] = m_n(z_o)$ for the altitude $z = z_o$ is taken to be the sample mean:

$$m_n = \frac{1}{k} (n_1 + n_2 + \ldots + n_k) \quad (4.8)$$

It is again noted that only the altitude and the random disturbance for each flight are recorded as functions of time. No information is needed concerning $x$ and $y$. The time origin for each flight is shifted so that altitude $z = z_o$ always occurs at time $t = t_o$.

The same data may be used to estimate the variance $\sigma^2_n(z_o)$ of $n_s(t_o, x_o, y_o, z_o)$ by means of the sample variance $\hat{\sigma}^2_n$:

$$\hat{\sigma}^2_n = \frac{1}{k-1} \left[ \frac{1}{k} \sum_{i=1}^{k} n_i^2 - \frac{1}{k} \left( \frac{1}{k} \sum_{i=1}^{k} n_i \right)^2 \right]$$

The procedure just discussed can be used to determine the function $m_n(z)$, the mean value of the process $n_s(w)$. In case this function can suitably be approximated by a function generator, then the mean value for the nonstationary stochastic process $X_s(t) = n_s(t, x(t), y(t); z(t))$ can be mechanized as a function of time for any particular rocket flight as shown in Figure 4.2.

![Figure 4.2. Generation of $E[X_s(t)]$.](image-url)
This provides a method of generating the time function \( m(t) \) in equation (2.1).

It is recalled that the function \( m(t) \) in equation (2.1) is utilized in the synthesis procedure described in Technical Note No. 3. This synthesis procedure provides a nonstationary stochastic process having prescribed first and second statistical moments.

It was pointed out earlier in this technical note that the stochastic process \( X_s(t) \) generated by the synthesis procedure described in Technical Note No. 3 is a Gaussian process. For this reason, at any time \( t = t_0 \) the random variable \( X_s(t_0) \) is a Gaussian random variable. In turn, if \( X_s(t) \) is generated by the partially-stationary process \( n_s(w) \) in the way explained earlier, then \( n_s(t_0, x_0, y_0, z_0) \) is a Gaussian random variable where \( x_0 = x(t_0), y_0 = y(t_0), z_0 = z(t_0) \). This is required to be true for all possible functions of time \( x(t), y(t), z(t) \). Thus, the random variable \( n_s(w) = n_s(t, x, y; z) \) is required to be Gaussian for any fixed selection of \( t, x, y, z \). By use of (4.5) the probability density function for this random variable can be written in the standard Gaussian form:

\[
p[n_s(w)] = \frac{1}{\sqrt{2\pi} \sigma_n(z)} e^{-\frac{[n_s(w) - m_n(z)]^2}{2 \sigma_n^2(z)}}
\]

(4.10)

The mean value \( m_n(z) \) and variance \( \sigma_n^2(z) \) appearing in this formula can be determined, or confirmed, experimentally by the method described earlier in this section.

The probability density function of (4.10) may be used to calculate any statistical parameter for the process \( X_s(t) = n_s(t, x(t), y(t); z(t)) \) that depends only on first-order distributions.

The results in this section have been presented in terms of a special partially-stationary stochastic process \( n_s(t, x, y; z) \). However, all of the results of Section IV are applicable to an arbitrary partially-stationary stochastic process of the form

\[
n_s(u_1, u_2, \ldots, u_k; v_1)
\]

(4.11)

where \( k \) is an arbitrary positive integer.
V. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

A class of stochastic processes, called partially-stationary processes, has been discussed in this technical note. The class is of potential usefulness in analog simulation studies. In a physical problem, such as the simulation of a random disturbance $X_s(t)$ affecting a rocket flight, a partially-stationary process $n_s(t, x, y; z)$ may be used to generate the nonstationary process $X_s(t) = n_s(t, x(t), y(t); z(t))$ that represents the actual random disturbance encountered in flight. Here, a single partially-stationary process generates the random disturbances encountered for an unlimited variety of trajectories of individual rocket flights.

In Section IV there is discussed the probability density function for a partially-stationary process that generates a Gaussian stochastic process in time having a prescribed first statistical moment. A procedure is suggested for experimentally determining, or confirming, the two parameters that appear in this analytical representation.

It is apparent that a similar analytical representation can be obtained for the joint probability density function of a partially-stationary process $n_s(\omega)$ that generates a Gaussian process $X_s(t)$. In this case $X_s(t_a)$ and $X_s(t_b)$ are jointly Gaussian random variables for any selection of time instants $t_a, t_b$.

If the process $X_s(t)$ is generated by the partially-stationary process $n_x(t, x, y; z)$ then it is seen that the two random variables $n_s(t_a, x_a, y_a; z_a)$ and $n_s(t_b, x_b, y_b; z_b)$ are jointly Gaussian where $x_a = x(t_a)$, $y_a = y(t_a)$, $z_a = z(t_a)$, $x_b = x(t_b)$, $y_b = y(t_b)$, $z_b = z(t_b)$. If $X_s(t_a)$ and $X_s(t_b)$ are required to be jointly Gaussian for arbitrary time functions $x(t)$, $y(t)$, $z(t)$ then it is seen that the two random variables $n_s(t_a, x_a, y_a; z_a)$ and $n_s(t_b, x_b, y_b; z_b)$ are required to be jointly Gaussian for any allowed selection of $t_a, x_a, y_a, z_a, t_b, x_b, y_b, z_b$. This being the case, the joint probability density function for these two random variables
can be written in the explicit Gaussian form.

The properties of the resulting Gaussian joint probability density function have not yet been studied. It is apparent that if the process is to be of use in a simulation study it will be necessary as a further research project to find experimental or theoretical methods of specifying the parameters that appear in this density function.

Once determined numerically, the joint probability density function for the random variables \( n_s(t_a, x_a, y_a; z_a) \) and \( n_s(t_b, x_b, y_b; z_b) \) can be used to calculate second-order statistical parameters for any time process \( X_s(t) \) that is generated by \( n_s(w) \). In particular, the covariance function can be calculated for any time process \( X_s(t) \) generated by \( n_s(w) \). If \( X_s(t) \) is to represent, say, a random disturbance encountered on a rocket flight, then the covariance function for \( X_s(t) \) can be calculated for an unlimited variety of flight trajectories by use of the single joint probability density function associated with \( n_s(w) \).

Use of the procedures discussed thus far would allow the calculation of the first two statistical moments for an arbitrary time process \( X_s(t) \) generated by a single partially-stationary process \( n_s(w) \). In turn, once these two calculations are made the synthesis procedure presented in Technical Note No. 3 can be used to mechanize a Gaussian process having the prescribed first two statistical moments.

The utilization of the synthesis procedure just discussed would require the construction of a different physical system for each distinct time process \( X_s(t) \). A research goal that is obviously of greater interest, and also of greater difficulty, is the direct mechanization of a partially-stationary process \( n_s(w) \). With reference to the example discussed in Section IV, the simulation of a random disturbance encountered on a rocket flight, the mechanization of \( n_s(w) \) would be utilized as is shown in Figure 5.1.
The input to the mechanization system would be some easily generated time process, such as white noise, along with the variables appearing in the multi-dimensional parameter \( w \) of \( n_s(w) \). In the system of Figure 5.1 these parameters are the time functions \( x(t), y(t), z(t) \) that describe the position of the rocket in flight. These parameters ordinarily would be directly available as terminal voltages in the analog computer utilized in the overall flight simulation problem. It is the availability of these parameters as time functions that provides the possibility of using the single physical system mechanizing \( n_s(w) \) to generate the random disturbance \( X_s(t) \) encountered on a wide variety of individual rocket flight trajectories.
REFERENCES


INVESTIGATION OF ELECTRONIC SWITCHES
FOR ANALOG AND HYBRID COMPUTATION

By John W. Robertson

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473
(Development of New Methods and
Applications of Analog Computation)

31 March 1965

Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
INVESTIGATION OF ELECTRONIC SWITCHES
FOR ANALOG AND HYBRID COMPUTATION

By
John W. Robertson

Project No. A-588
Technical Note No. 6

Contract No. NASA-2473
(Development of New Methods and Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
ABSTRACT

On the basis of a preliminary literature survey, selected analog switches were investigated to determine the relative performance characteristics of types considered most promising in analog or hybrid computer applications. Measured performance parameters are tabulated and evaluations given for examples of the modified Guennou chopper, the nonsaturating two- and four-transistor switches, the complementary transistor bridge, the series-shunt unifet switch, and the six-diode bridge gate, all constructed from available standard semiconductor components. From these results, it is recommended that further work be done with the six-diode bridge gate and with the series-shunt field-effect-transistor switch to optimize their performance for specific analog and hybrid computer functions.

FOREWORD

Contract NAS8-2473 was initiated in September 1961 and has covered a variety of assignments relating to work at the Flight Simulation Branch of Marshall Space Flight Center's Computation Laboratory. The MSFC Project Technical Officer has been Dr. W. K. Polstorff (R-COMP-RS). A large portion of the effort on this program is now being provided by personnel of the Georgia Tech School of Electrical Engineering in the general areas of (a) nonstationary noise studies, and (b) error analysis for hybrid computation. However, the laboratory investigations described herein were performed by Mr. John W. Robertson, with technical guidance from Mr. Frank R. Williamson, Jr., in the Special Problems Branch of the Engineering Experiment Station's Physical Sciences Division, under which the project has thus far been administered. It is planned that these investigations will be continued on a more limited basis during the remainder of the current contract period, terminating in December 1965.

F. Dixon
Project Director A-588
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I - INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II - DESCRIPTION OF SWITCH TYPES STUDIED</td>
<td>3</td>
</tr>
<tr>
<td>2-1. Modified Guennou Chopper</td>
<td>3</td>
</tr>
<tr>
<td>2-2. Nonsaturating Transistor Switches</td>
<td>4</td>
</tr>
<tr>
<td>2-3. Complementary Transistor Bridge</td>
<td>5</td>
</tr>
<tr>
<td>2-4. Series-Shunt Unifet Switch</td>
<td>6</td>
</tr>
<tr>
<td>2-5. Six-Diode Bridge Gate</td>
<td>7</td>
</tr>
<tr>
<td>III - SWITCH MODEL DEFINITIONS AND PERFORMANCE CRITERIA</td>
<td>8</td>
</tr>
<tr>
<td>3-1. Ideal and Nonideal Switch Models</td>
<td>8</td>
</tr>
<tr>
<td>3-2. Switch Performance Parameters</td>
<td>9</td>
</tr>
<tr>
<td>3-3. Model Performance Parameters</td>
<td>10</td>
</tr>
<tr>
<td>3-4. Parameter Calculations for Specific Circuits</td>
<td>11</td>
</tr>
<tr>
<td>IV - EXPERIMENTAL RESULTS AND DISCUSSION</td>
<td>13</td>
</tr>
<tr>
<td>4-1. Performance Data Summary</td>
<td>13</td>
</tr>
<tr>
<td>4-2. The Modified Guennou Chopper</td>
<td>15</td>
</tr>
<tr>
<td>4-3. The Nonsaturating Transistor Switch</td>
<td>16</td>
</tr>
<tr>
<td>4-4. The Complementary Transistor Bridge</td>
<td>17</td>
</tr>
<tr>
<td>4-5. The Series-Shunt UNIFET Switch</td>
<td>18</td>
</tr>
<tr>
<td>4-6. The Six-Diode Bridge Gate</td>
<td>19</td>
</tr>
<tr>
<td>V - CONCLUSIONS AND RECOMMENDATIONS</td>
<td>21</td>
</tr>
<tr>
<td>5-1. Summary Evaluation</td>
<td>21</td>
</tr>
<tr>
<td>5-2. Future Work</td>
<td>22</td>
</tr>
<tr>
<td>Appendix - BIBLIOGRAPHY</td>
<td>23</td>
</tr>
<tr>
<td>A. Transistor Gate References</td>
<td>24</td>
</tr>
<tr>
<td>B. Diode Gate References</td>
<td>25</td>
</tr>
<tr>
<td>C. FET Gate References</td>
<td>26</td>
</tr>
<tr>
<td>D. Miscellaneous Gate References</td>
<td>26</td>
</tr>
<tr>
<td>E. Supplementary References</td>
<td>27</td>
</tr>
<tr>
<td>iii</td>
<td></td>
</tr>
</tbody>
</table>
The general aim of the work on which this report is based is to obtain a design for an analog switch of improved performance.

Analog switching devices, both electromechanical and electronic, are used in many fields of electronics. In communications, they are variously referred to as commutators, multiplexers, modulators or demodulators, and gates. In analog computation, switching devices may be called choppers, modulators, samplers, sampling switches, input selectors, or simply switches. Other fields also have a variety of technical names for analog switches, each with its own specialized meaning. However, the basic function of all of the devices is really the same, and the different terminologies tend to indicate not so much the nature of a device as its mode of operation or application. Thus, some applications may allow use of devices that are restricted to a particular frequency of operation, while others may require that the device be capable of more or less random operation. But in either case the fundamental objective is to provide, alternately, either lossless transmission or complete attenuation of an input signal, when and as desired.

Typical examples of analog and hybrid computer applications of switches are (1) sampling switches for sample-and-hold circuits, (2) reset switches in mode control logic, and (3) reset pulse generator switches for incremental analog-to-digital conversion devices.\(^*\) Such applications require switches capable of switching at intervals ranging from infinity (continuous operation) to possibly microseconds, with nonsymmetrical on and off times. In addition, successive on and off times need to be independent.

For a switch to meet the above requirements, it must be capable of continuous operation. This in itself rules out many of the possibilities in the literature, including the majority of switches with ac-coupled drives. Analog switches for computer application must also have dc-coupled signal channels. Several types of switches, such as the diode ring-bridge modulator and the diode capacitance modulator are inherently incapable of continuous operation. Other types, such as the transistorized switch of Williams, have ac-coupled signal channels.

\(^*\) As in the "AID Converter" developed by F. R. Williamson, Jr. under Contract DA-01-009 ORD-853 and subsequently incorporated in a generalized analog integrator ("EGI") under the present NASA contract.
Several switches having ac-coupled drive signals, such as the Bright circuit, may be modified to provide continuous operation through the use of a silicon controlled switch (SCS) and a bias source, and would be expected to give the same performance as the unmodified versions.

Other desirable qualities in an analog switch are economy, reliability, high switching speed, and high frequency capabilities. Besides yielding already known performance, the modified units mentioned above are more complex and limited in switching speed. Photoconductive gates also fall into the slow switching speed category with presently available drives (laser diodes excluded).

The switching circuits chosen for further investigation on the bases indicated above are identified and described in the next chapter.
II - DESCRIPTION OF SWITCH TYPES STUDIED

2-1. Modified Guennou Chopper (see References A13,A18)

The Guennou chopper is a simple two-transistor switching circuit, designed by S. Guennou in France and subsequently described by H. Kemhadjian in the April 1960 issue of Mullard Technical Communications (Reference A13). The modified version shown below was arrived at by removing the coupling capacitor of the standard chopper and direct-coupling the load.

Current drive is supplied to the base of each transistor at the value resulting in lowest inverse-mode saturation voltage $V_{CE(sat.)_i}$, achieved by adjusting the base resistors $R_{b1}$ and $R_{b2}$. The load resistance $R_L$ should be large enough so as not to load the balance potentiometer $R$, which in turn is chosen so as to make the effects of $R_{in}$ in series with the pot negligible. The transistors are operated in the inverse mode in order to provide a low offset voltage, and they should be matched against one another for inverse-mode current gain $\beta_i$, collector-to-emitter leakage current $I_{CEO_i}$, and saturation voltage $V_{CE(sat.)_i}$.

Figure 1. Modified Guennou Chopper.
The nonsaturating transistor switch is a type of switched complementary emitter-follower circuit. Shown below is the "four-transistor" design suggested by Brubaker in Reference A5. His "two-transistor" version is similar but omits the first emitter-follower stage (Q3, Q4).

Voltage drive is supplied to points a and b, thus alternately shorting to ground and releasing the bases of the first complementary emitter-follower (c, d). Transistors for each emitter-follower should be a complementary pair, matched for normal-mode current gain ($\beta_n$) and having a high current gain at the temperatures of operation.

Figure 2. Four-Transistor Nonsaturating Switch.
2-3. Complementary Transistor Bridge (see Reference A12)

Depicted below is the transistor bridge circuit described by Kalfaian in Reference A12 for low-level switching applications. This has been previously referred to on this project as the "complementary microvolt transistor bridge."

Complementary drive signals are applied at points a and b to turn all transistors off or on at the same time. This circuit requires a matched set of four transistors, two npn and two pnp types, matched for normal-mode current gain $\beta_n$, collector-to-emitter leakage current $I_{CEO}$, and saturation resistance $R_{CE\text{(sat.)}}$ over the temperature range of interest.

Figure 3. Complementary Transistor Bridge.
2-4. Series-Shunt Unifet Switch (see Reference C3)

The simple switching circuit shown below utilizes a pair of unipolar field-effect transistors (FET's). Push-pull drive signals applied at points a and b serve to turn on one FET and simultaneously turn off the other. There are no matching requirements for this circuit.

![Series-Shunt Unifet Switch Diagram]

Figure 4. Series-Shunt Unifet Switch.
2-5. Six-Diode Bridge Gate (see Reference B7)

In an excellent 1955 paper (Reference B7), Millman and Puckett analyzed the performance of low-level gating circuits incorporating two, four, or six diode units in a bridge-type configuration. Their results indicated that the six-diode version would be most suitable for further investigation under the present project.

In the circuit as shown below, push-pull control voltages are applied at points a and b and cause the four bridge diodes simultaneously to either conduct or block. For optimal performance, the fixed supply voltages (± V) need to be quite accurately balanced, the four bridge diodes should have a high ratio of forward conductance to reverse leakage, and the two output diodes should be closely matched as regards their leakage currents.

Figure 5. Six-Diode Bridge Gate.
3-1. Ideal and Nonideal Switch Models

Ideally, a switching device should exhibit zero on-resistance, infinite off-resistance, total absence of either offset voltage or drive leakage, and infinite operating speed (zero actuation time). Of course, no real switch can provide perfectly ideal performance. Electronic and electromechanical switches alike are limited in operating speed, have some offset voltage, have less than infinite off-resistance, and more than zero on-resistance. In addition, they are not completely isolated from ground, and electrically actuated types will possess some drive leakage. These nonideal properties, except for the drive leakage, have been represented in the switch equivalent-circuit diagrams below. Alternative models are of course possible (e.g., see Reference A17).

Figure 6. SPST Switch Models: (a) actual, (b) ideal.

Figure 7. SPDT Switch Models: (a) actual, (b) ideal.
3-2. Switch Performance Parameters

After some consideration, it was decided that the electronic switches to be studied should be treated as amplifiers, with gain and offset measurements made on them under both "on" and "off" conditions. Switching frequency limits were also to be determined, and additional data would be taken if deemed necessary or desirable. Definitions of the primary performance parameters measured are shown in equation and diagrammatic form below.

\[
A_v(\text{on}) = \frac{\Delta E_{\text{out}}}{\Delta E_{\text{in}}} \quad \text{switch on} \quad R_L = 1\Omega
\]

\[
V_o = E_{\text{out}} \quad \text{Switch on} \quad E_{\text{in}} = 0\,\text{v}
\]

\[
A_v(\text{off}) = \frac{\Delta E_{\text{out}}}{\Delta E_{\text{in}}} \quad \text{switch off} \quad R_L = 1\Omega
\]

\[
I_D = E_{\text{out}} \quad \text{Switch off} \quad E_{\text{in}} = 0\,\text{v}
\]

\[-E_{\text{out}} \quad \text{Switch on} \quad E_{\text{in}} = 0\,\text{v}
\]

Figure 8. Switch Parameter Measurement Circuits.
3-3. Model Performance Parameters

As a matter of general interest, the actual switch performance parameters defined above may be expressed in terms of the theoretical switch models presented under Section 3-1. The following approximate relationships are obtained after eliminating terms of negligible magnitude in accordance with the assumptions noted.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SPST Model</th>
<th>SPDT Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{V}(on)$</td>
<td>$\frac{R_L}{R_L + R_2}$</td>
<td>$\frac{R_2 R_1}{(R_L + R_2 + R_S) R_S}$</td>
</tr>
<tr>
<td>$V_o$</td>
<td>$V_1 \left{ \frac{R_L R_1}{R_L R_1 + R_2 (R_L + R_L)} \right}$</td>
<td>$V_1 \left{ \frac{R_L}{R_L + R_1} \right}$</td>
</tr>
<tr>
<td>$A_{V}(off)$</td>
<td>$\frac{R_L R_3}{R_L (R_3 + R_1) + R_1 R_3}$</td>
<td>$\frac{R_3}{R_S + R_1}$</td>
</tr>
<tr>
<td>$L_D$</td>
<td>$L_1 \left{ \frac{R_L R_1}{R_3 (R_L + R_1) + R_1 R_1} \right}$</td>
<td>$L_1 \left{ \frac{R_2 R_1}{R_3 (R_2 + R_1) + R_2 R_1} \right}$</td>
</tr>
</tbody>
</table>

Conditions:
- $R_2 << R_L << R_1, R_3$
- $R_S << R_1, R_4$
- $R_3 << R_1$
- $e_{in} >> L_1$
- $e_{in} >> L_1 (R_1/R_3)$
3-4. Performance Calculations for Specific Switches

Although the foregoing mathematical expressions and the models from which they are derived can provide useful insights into switch behavior, they are not of much help in attempting to predict the performance of an actual switch. One may, however, obtain quantitative estimates of desired performance parameters in any given case by fairly straightforward analysis of the switch circuitry. Presented hereunder are results derived for the two most important switch types considered.

Series-Shunt Unifet Switch

Given \( I_{GSS} \) = gate-to-source leakage current of FET,
\( r_{ds} \) = drain-to-source resistance with zero gate-to-drain voltage,
\( R_{off} \) = (drain-to-source voltage)/(pinch-off drain current),

then for the circuit shown in Section 2-4, provided \( R_L \) is small compared to \( r_{ds}(Q_2) \) and \( R_{off}(Q_2) \), the switch performance parameters may be calculated as follows:

\[
A_v(\text{on}) = \frac{R_L}{r_{ds}(Q_1)} \quad A_v(\text{off}) = \frac{r_{ds}(Q_2)}{r_{ds}(Q_2) + R_{off}(Q_1)}
\]

\[
V_o = I_{GSS}(Q_2) \cdot r_{ds}(Q_1) \quad I_D = I_{GSS}(Q_1) \cdot r_{ds}(Q_2)
\]

Six-Diode Bridge Gate

Analysis of the switch given in Section 2-5 is facilitated by use of the diode equivalent circuit shown below (containing an assumed ideal diode).

Figure 9. Equivalent Circuit for Diode.

Any offset voltage \( V_o \) in the six-diode gate can normally be balanced out by adjusting potentiometer \( R \). This adjustment will typically provide compensation for values of \( R_f \) in the bridge output diodes in the ratio of 3:1 (with \( R = 50 \Omega, R_f = 25 \Omega \)).
The drive leakage $L_D$ of the six-diode gate can be assumed to be produced solely by the difference in leakage currents of the two output diodes. This assumption will be valid under the conditions that the bridge output diodes have back resistance ($R_b$) large compared to the load resistance ($R_L$), that the two drive diodes have small forward resistance ($R_f$), and that the impedance of the drive source is small.

Values for on voltage gain and off voltage gain of the six-diode gate may be calculated quite readily with formulas derived from an ac model of the circuit—or, with considerably more calculations, from a dc analysis (see Reference B7). The approximate expressions for predicting switch performance are thus as follows.

$$A_v(\text{on}) = \frac{R_c/2}{R_c/2 + R_f/2 + R/L} \cdot \frac{R_L}{R_f/2 + R_L}, \quad V_o = 0 \text{ (adjusted)};$$

$$A_v(\text{off}) = \frac{R_c/2}{R_c/2 + R_b/2 + R/L} \cdot \frac{R_L}{R_b/2 + R_L}, \quad I_D = (I_{b3} - I_{b2}) R_L.$$
IV - EXPERIMENTAL RESULTS AND DISCUSSION

4-1. Performance Data Summary

Performance data are tabulated hereunder for the various electronic switches that have been constructed and tested on this project. The switches are identified in the table as follows:

- MGC - modified Guennou chopper
- 4TNS - four-transistor nonsaturating switch
- 2TNS - two-transistor nonsaturating switch
- CTB - complementary transistor bridge
- SSUS - series-shunt unifet switch
- 6DG/F - six-diode bridge gate using Fairchild 1N3595's
- 6DG/G - six-diode bridge gate using G.E. 1N441.1.3's

The table includes values for the main performance parameters defined in Section 3-2, namely: on voltage gain, $A_v$ (on); offset voltage, $V_o$; off voltage gain, $A_v$ (off); and drive leakage, $L_D$. In the case of the series-shunt unifet switch and the two six-diode gates, these parameters were determined not only at the normal room temperature of 25°C but also at an oven-controlled temperature of 50°C. The two sets of data then yielded values for the nominal temperature sensitivities $A_{v_0}/AT$ and $A_{L_D}/AT$, as shown in the table.

The column designated $f_{\text{max}}$ gives the maximum frequency of operation determined by taking the reciprocal of the summed rise and fall times (i.e., times required for the output voltage to reach its final value, rising and falling) of the switching waveform as observed on an oscilloscope. Obviously, this method suffers somewhat in accuracy due to the necessity of determining a point at which the waveform may be said to have reached its final value. A different, more accurate, procedure might be desirable for some applications.

The entries marked $E_{\text{in}}$ represent maximum signal input levels permitted for the particular circuit components used. These are not listed as ultimate limits but do indicate what might be considered appropriate levels of operation for each type of switch tested.

As a final point of interest, it should be mentioned that the first four switch models listed were built on standard phenolic assembly boards, with no special precautions for shielding. The last three switch models, on the other hand, were enclosed in small aluminum boxes and were connected to the drive circuitry by shielded cables. The method of construction for these models may be seen from the photographs below, one of which also shows a typical arrangement of the instrumentation involved in a switch performance test.
### Table 1. Experimental Values of Switch Performance Parameters.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MOG</td>
<td>0.69</td>
<td>0.2</td>
<td>$6\times10^{-1}$</td>
<td>0.1</td>
<td></td>
<td></td>
<td>60</td>
<td>± 1</td>
</tr>
<tr>
<td>LTNS</td>
<td>0.81</td>
<td>0.2</td>
<td>$8\times10^{-3}$</td>
<td>0.2</td>
<td></td>
<td></td>
<td>50</td>
<td>± 5</td>
</tr>
<tr>
<td>2TNS</td>
<td>0.72</td>
<td>0.1</td>
<td>$1\times10^{-3}$</td>
<td>17</td>
<td></td>
<td></td>
<td>50</td>
<td>± 5</td>
</tr>
<tr>
<td>CTB</td>
<td>0.9</td>
<td>3.0</td>
<td>$5\times10^{-3}$</td>
<td>0.1</td>
<td></td>
<td></td>
<td>500</td>
<td>± 30</td>
</tr>
<tr>
<td>SSUS-25°</td>
<td>0.95</td>
<td>0.4</td>
<td>$5\times10^{-4}$</td>
<td>0.040</td>
<td></td>
<td></td>
<td>350</td>
<td>± 2</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>0.4</td>
<td>$5\times10^{-4}$</td>
<td>0.040</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6DG/F-25°</td>
<td>0.989</td>
<td>0*</td>
<td>$1\times10^{-7}$</td>
<td>0.120</td>
<td>24</td>
<td>27.2</td>
<td>2.5</td>
<td>±20</td>
</tr>
<tr>
<td></td>
<td>0.990</td>
<td>0*</td>
<td>$2.5\times10^{-7}$</td>
<td>0.800</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6DG/G-25°</td>
<td>0.989</td>
<td>0*</td>
<td>$1\times10^{-7}$</td>
<td>0.001</td>
<td>18</td>
<td>0.75</td>
<td>2.5</td>
<td>±20</td>
</tr>
<tr>
<td></td>
<td>0.988</td>
<td>0*</td>
<td>$1\times10^{-7}$</td>
<td>0.020</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* $V_o$ given as zero, indicating that circuit permits adjusting to zero for any particular temperature and pair of supply voltages (balanced within 1%)

---

Figure 10. Shielded Construction of Six-Diode Gate.

Figure 11. Arrangement of Instrumentation for Switch Performance Tests.
1-2. The Modified Guennou Chopper

The Guennou chopper, in its modified form, offers little benefit over a simple single-transistor shunt chopper. The small improvement afforded lies in making the offset voltage and drive leakage more nearly equal, and thus easing the compensation problem slightly.

A number of different transistor types were tried in the Guennou circuit, but none provided any really outstanding advantage. The types tested included 2N220, 2N404A, 2N697, 2N706, 2N1132, 2N1179, 2N1605A, 2N2102, and 2N2613. Reference A18 points up the need for transistors with inverse current gains ($\beta_i$) on the order of 10. Special switching transistors, both silicon and germanium, are available with high inverse current gains. However, from the basic capabilities of the Guennou circuit, it is doubtful whether the construction of further switches of this type could be justified. Even if the transistors were replaced by ideal switches, the ratio of the on voltage gain to the off voltage gain (a suggested figure of merit) would not be greatly enhanced.
4-3. The Nonsaturating Transistor Switches

The two- and four-transistor nonsaturating switches suggested by Brubaker (Reference A5) gave much better results than the Guennou chopper. Construction of this switch employed pairs of complementary transistors (2N697, 2N1132) matched to within 10% for $\beta_n$, $R_{CE\text{sat}}$, and $I_{CBO}$ on a Tektronix 575 transistor curve tracer. This was the best match possible with the limited number of transistors available. Performance of the switch could undoubtedly be improved by the use of matched pairs of transistors (or matched quadruples) purchased from the manufacturer. Dual transistors, consisting of one npn and one pnp each drawn from a matched pair and mounted in the same can, should provide even better performance by reducing temperature differences between transistors in the same stage of the switch.

It is doubtful, however, whether the off voltage gain can be reduced greatly with the bases of the switch transistors being switched to ground as at present. An arrangement whereby the bases of all of these transistors could be switched to some bias voltage, so as to provide reverse bias to the switch transistors, would probably reduce the off voltage gain by at least one order of magnitude.

The magnitude of the input voltage is limited by the available supply voltage and the limited heat dissipation of the transistors. The supply voltage may be increased up to the limit of the transistor ratings, provided steps are taken to insure that the transistor dissipation remains within limits.

This switch suffers generally from the requirements for matched, high-gain, low-leakage transistors, and balanced power-supply voltages. Unbalance of the supply voltages will be reflected as an offset in the output voltage, as will any mismatch of the switching transistors. Adjustment of the output voltage to remove the offset is accomplished by adjusting the bias current of the switching transistors. This offset adjustment is usable only for long-term variations in supply voltage and hence the short-term stability requirements can become rather acute.
Complementary Transistor Bridge

Construction of the complementary transistor bridge utilized a matched quadruple of transistors, two npn (2N697) and two pnp (2N1132). Matching of the units was for $\beta_n$, $R_{CE\text{ (sat.)}}$, and $I_{CEO}$, and was accomplished to within 10% by means of a Tektronix 575 transistor curve tracer. This was the best match possible with the available transistors. Better matching and low leakage transistors would probably result in lower offset voltage, lower off voltage gain, and lower drive leakage. Temperature effects could be minimized by using a matched quadruple of transistors mounted in a single TO-5 enclosure.

The maximum input voltage to this switch (in our case, ± 30 volts) was set by transistor breakdown and can be increased by using higher-voltage transistors.* Another factor limiting the input signal maximum is the insulation of the bias supply for the bridge and the supplies for the drive circuitry.

The lack of any requirement for balanced voltages by the bridge itself causes the unbalance sensitivity to be zero. Variations in the bias supply have relatively little effect. However, a major disadvantage of this switch is the number of power supplies required.

* Note that it is easier to match transistors of the same type, and that higher voltage ratings are available with silicon transistors of the npn variety than with the pnp type. Thus, a bridge might be constructed using all npn transistors, but at the cost of a higher degree of complexity in the drive circuitry (Reference A12).
4-5. Series-Shunt Unifet Switch

The series-shunt unipolar field-effect transistor switch reported on was constructed from a Siliconix telemetry designer's kit. The FET units involved (types 2N3380 and 2N3386), although giving good results, can be improved upon. Desirable properties include low drain cutoff current $I_{D\text{off}}$, low gate-to-source leakage current $I_{GSS}$, low drain-to-source resistance $r_{ds}$ (at zero gate-to-drain voltage), and low gate-to-channel capacitance $C_{GSS}$.

Note that some means must be provided to discharge the gate capacitance of the FETs during the on time for each. This may be accomplished by utilizing the leakages of the control diodes $CR_1$ and $CR_2$, the gate-channel diode leakages of the FETs $Q_1$ and $Q_2$ themselves, or other means.

No matching of components is necessary in this switch, and no supply voltage is required. The field effect transistor presents a pure resistance to the signal with no intrinsic offset voltage. Input voltage is limited by the drive signal levels and gate-to-channel breakdown.

In experimenting with this switch, shielding was found necessary for the first time—at least partly because of the long lines required to allow operation inside the oven for high temperature measurements. Frequency measurements were thus taken with approximately six feet of RG-62/U between the output of the gate and the load. A Tektronix 545A with 47 pf shunt capacitance was used to observe the output.

The absence of a power supply requirement in the FET switch is a major advantage. This feature materially reduces the amount of auxiliary equipment needed for the switch, and at the same time the output contains no components due to power-supply variations or unbalance. Temperature variations have little effect on the gate offset voltage and drive leakage. The inherent simplicity of the circuit should provide greatly increased reliability and lower cost.

Linearity of the series-shunt unifet switch was roughly checked in the course of the laboratory tests by applying a sine-wave input to the switch and observing its output while driven at a high switching rate. No distortion was visible in the chopped output sine wave. A more accurate measurement of linearity would of course be desirable in any subsequent evaluation of improved FET switches.
The six-diode gate gave the lowest value of off voltage gain of any circuit tested. Switch units were constructed using Fairchild type 1N3595 diodes and General Electric 1N4443 diodes. The initial unit using 1N3595 diodes gave very poor results. When the diodes were removed from the bridge and tested, all were found to be out of tolerance in leakage current. As a result of this discovery, the remaining available diodes were tested to ascertain their adherence to the published parameter limits. Only four 1N3595 diodes were found to be within tolerance. All of the 1N4443's were within tolerance for leakage current and only two slightly out of tolerance on forward drop at one value of current. It was decided to utilize the four good 1N3595's as the bridge diodes in one circuit, and 1N4443's as the drive diodes for both circuits.

It was also decided to use the leakage current values measured at the voltages specified in the data in an attempt to provide matched diode pairs for the input and output legs of the bridge. This, in effect, neglected the various phenomena such as surface leakage which cause the leakage currents of diodes to depart from the values predicted by the diode equation: $I = I_o [e^{-\frac{eV}{KT}} - 1]$. Subsequent measurements indicated that, in this case, the other effects predominate.

Matching of the reverse leakage, of the output diodes particularly, at the actual voltage to be used is necessary for best results. Low-leakage diodes with high forward conductance are desirable in the bridge. The drive diodes' leakage contributes far less to the over-all gate offset voltage. In the particular circuit used, the drive diodes were calculated to produce about 1 microvolt of offset voltage for each 40 nanoamperes difference in leakage current between the two.

Contributions to the gate offset voltage by unbalance of the supply voltages in the on state require extremely stringent regulation of the supply voltages to ensure that the unbalance remains small enough to introduce no significant error. In order to limit this output effect to a 1 microvolt contribution, the necessary regulation of the supplies was calculated to be .00015%. With the actual gates constructed, sensitivity of the output voltage to supply unbalance was observed to be approximately 5 millivolts for each volt of unbalance.

Formulas presented in Section 3-4 enable $I_D$, $A_v(on)$, and $A_v(off)$ for the six-diode gate to be predicted from measured characteristics of the individual diodes. In general, such theoretical calculations agreed well with experimental
results for the actual gates constructed, which were designed to operate at a 20-volt signal level. The formulas were also used to calculate the performance for a gate designed to operate at a 1-volt signal level. The predicted performance was found to be considerably poorer in this case. Indications are that the largest ratios of maximum input signal to drive leakage, and likewise the smallest values of off voltage gain, will be obtained by using the highest practicable supply and drive voltages.

Switching speed could be considerably improved by connecting a second six-diode gate so as to switch to ground the output of the first gate during the time that it is nonconducting.

Although no attempt was made to balance out gate drive leakage during the investigations described herein, it is probable that such compensation could be provided for in the case of the six-diode bridge circuit.
5-1. Summary Evaluation

The Guennou chopper, in its modified form, offers little improvement over a simple single-transistor shunt chopper. The advantages afforded lie in the drive leakage, which is lower than for the shunt chopper, and the lack of a supply voltage, variations of which could affect the output. The possibility of improving the performance greatly appears quite small.

Advantages of the four-transistor nonsaturating switch are few, except that it will give better performance than the Guennou chopper or a simple shunt transistor chopper. A further reduction in $A_v(\text{off})$ by a factor of $\beta$ is probable with provision for back-biasing the transistor bases which are now switched to ground. Dual transistors in one enclosure, matched over a temperature range of operation, also should improve performance.

The complementary transistor bridge offers improved performance over the four-transistor switch. Major advantages of this circuit are the increase in maximum signal voltage which can be applied without damaging the components, and the relative immunity to power supply variations. Improvements should be possible utilizing ultra-low-leakage transistors, matched over a temperature range. Dual- and quad-transistors mounted in a single can might also offer some improvement.

The circuit offering the most advantages of any tried thus far is the series-shunt FET switch. No matched components are necessary, no supply voltage variations can affect the output, no adjustments are necessary to the output voltage. Finally, this circuit gave lower $A_v(\text{off})$ than any other except the six-diode bridge. Improvements to this circuit may be possible through the use of FET's with lower $r_{ds}$ and MOS FET's with much lower gate leakage.

The six-diode bridge had the highest ratio of $A_v(\text{on})$ to $A_v(\text{off})$ of any circuit tested, due to the extremely low $A_v(\text{off})$. Few other advantages are offered by this circuit, however. The probability of significant improvements in this circuit's performance is very small at the present time. Diodes having the highest ratio of forward to reverse current known to us were used, and several years will probably be required for production diodes with significantly better ratios of forward to reverse current to reach the market.
5-2. Future Work

It is recommended that further investigative effort be applied to the FET switches. It is also recommended that a bridge circuit incorporating silicon controlled switches (SCS's) in the approximate configuration of the complementary transistor bridge be constructed and tested. The construction of the FET switches would demonstrate the improvement possible with the best available unipolar and MOS type FET's. The SCS bridge would permit use of very narrow drive pulses for on and off periods approaching infinity, thus allowing ac coupling of the drive signal without limiting the lower frequency of operation of the switch.
Appendix - BIBLIOGRAPHY

Initial efforts on this program were applied to a study of the pertinent literature on electronic switching techniques and applications. The literature survey has included a review of The Engineering Index since 1961 and continuing coverage of the DDC Technical Abstract Bulletin from the January 1963 issue on. In addition, several of the major electronic engineering journals are regularly received and scanned, as are new-product announcements and data sheets from a number of manufacturers in the semiconductor components field.

The bibliography presented herein contains 72 entries divided into the following subject categories—(the number of documents cited in each category is shown in parentheses):

A. Transistor Gate References (25)
B. Diode Gate References (9)
C. FET Gate References (11)
D. Miscellaneous Gate References (18)
   Bidirectional Transistor Gates (1)
   Hall Generator Gates (1)
   Photo Gates (6)
   Varicap Gates (1)
   Integrated Tube Gates (2)
   Vacuum Tube Gates (2)
   Nickle Oxide Gate (1)
   Thyristor Gates (3)
E. Supplementary References (9)

The relative importance of various references to the investigations reported herein is indicated by asterisks in the margin. A double asterisk denotes a primary reference—one which provided essential or valuable information for a particular switch under study. A single asterisk denotes a secondary reference—one which served to supplement a primary reference or which provided information permitting the elimination of certain switches from further study.
Bibliography

A. Transistor Gate References


-24-


B. Diode Gate References


C. FET Gate References


D. Miscellaneous Gate References

Bidirectional Transistor Gates


Hall Generator Gates


Photo Gates


Varicap Gates


Integrated Circuit Gates


Vacuum Tube Gates


Nickle Oxide Gate


Thyristor Gates


E. Supplementary References


TECHNICAL NOTE NO. 7
Project A-588

PRELIMINARY ERROR ANALYSIS OF A
SPECTRUM MEASUREMENT SYSTEM

By David L. Finn

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

23 April 1965

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
PRELIMINARY ERROR ANALYSIS OF A SPECTRUM MEASUREMENT SYSTEM

By

David L. Finn

TECHNICAL NOTE NO. 7

on

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation)

For

GEORGE C. MARSHALL SPACE FLIGHT CENTER

Huntsville, Alabama
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>II OBJECTIVES OF INVESTIGATION</td>
<td>2</td>
</tr>
<tr>
<td>III QUALITATIVE DESCRIPTION OF SYSTEM OPERATION</td>
<td>3</td>
</tr>
<tr>
<td>IV SYSTEM ANALYSIS FOR DETERMINISTIC INPUTS</td>
<td>6</td>
</tr>
<tr>
<td>V DISCUSSION OF ERROR ANALYSIS</td>
<td>12</td>
</tr>
<tr>
<td>VI BIBLIOGRAPHY</td>
<td>13</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Figure 1</td>
<td>Spectrum Measurement System</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Waveforms in the Spectrum Measurement System of Figure 1</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Idealized Model to Describe the Operation of the System of Figure 1</td>
</tr>
</tbody>
</table>
I SUMMARY

The problem of analyzing a spectrum measurement system that utilizes a time-compression device is considered. A qualitative description of the system operation is given, and an idealized model is proposed to describe the system operation when the input is a deterministic time function. Input-output relationships are presented for each operation in the idealized model. Using the model proposed, it is estimated that approximately six additional man-weeks would be required for developing a procedure for determining the error in the measured spectrum of a deterministic function of time.

The problem of error analysis in measuring the spectrum of a more general class of functions including random processes is indicated to be much more difficult. A minimum of perhaps three times the above estimate might be required to obtain significant results.
II OBJECTIVE OF INVESTIGATION

The objective of the investigation outlined in this report is to appraise the problem of making an error analysis for a spectrum-measurement system that utilizes a time-compression device. An idealized model is proposed to describe the system operation when the input is a deterministic time function, and an estimate is made for the amount of time required to perform an error analysis by use of the idealized model.
The system to be analyzed is shown in block diagram form in Figure 1. The operation of this system may be described as follows:

An input function of time $v(t)$ (as shown in Figure 2) whose spectrum is to be measured is applied to the time-compression device. The first function of the time-compression device is to obtain from $v(t)$ a discrete number $N$ of samples, with equal time spacing $d$ between samples, as is indicated in Figure 2. Allowing the time interval $d$ for each sample, the total time interval required for $N$ samples is $T = dN$. All the sample values in this interval are stored within the time-compression device. The output $c(t)$ of the time-compression device is composed of a periodic replication of the original train of $N$ samples. However, a hold operation is performed on each sample, and the time interval between successive samples is compressed by a factor $K$. The time duration of individual sample pulses in the waveform $c(t)$ is thus $d/K$ as is shown in Figure 2. Here, a zero-order hold is depicted for each sample of $c(t)$.

Provided the sampling interval $d$ is sufficiently small, the output $c(t)$ of the time-compression device is to a good approximation a periodic repetition of the portion of the input waveform $v(t)$ within the time interval $T$. The time interval $T$ for the input $v(t)$ is compressed to the time interval $T/K$ within the waveform $c(t)$. This provides a compression in time for $v(t)$ by a factor $K$ and an expansion in frequency for the spectrum of $v(t)$ by the same factor $K$.

The output $c(t)$ of the time-compression device is combined in the mixer with the output $f(t)$ of a fixed-frequency oscillator. This operation translates the frequency spectrum of $c(t)$ by the amount of the oscillator frequency $\omega_f$. The frequency $\omega_f$ is selected so that the spectrum-component to be measured is positioned in frequency at the mid-frequency in the pass band of the band-pass filter. Essentially, this spectrum-component is passed through the filter, and other components are rejected. In responding to the waveform $b(t)$ the voltage measurement device thus gives an indication which approximates the magnitude of the selected spectrum-component for the original input function $v(t)$. 

Figure 1. Spectrum Measurement System.
Figure 2. Waveforms in the Spectrum Measurement System of Figure 1.
IV SYSTEM ANALYSIS FOR DETERMINISTIC INPUTS

The idealized model shown in block diagram form in Figure 3 is proposed as a model to describe the operation of the system of Figure 1 when the input is a deterministic function of time. The individual operations depicted in the model are designated as:
1. Truncation
2. Repetition
3. Sampling
4. Holding
5. Frequency Expansion
6. Mixing
7. Narrow-band Filtering

Notation for the waveform in the model of Figure 3 has been selected to conform to that of the system shown in Figure 1. The voltage measurement device of Figure 1 has not been included in the model because it is not expected to make a substantial contribution to the system error.

The logical order in which the operations in the model occur is in part arbitrary and has been selected in what appears to be an advantageous way to facilitate determination of operation outputs and errors. The idealized operation of the system may be analyzed as follows:

1. Truncation

An input function of time \( v(t) \) is assumed to possess a Fourier transform

\[
V(\omega) = \int_{-\infty}^{\infty} v(t) e^{-j\omega t} dt
\]

This function is expected to have appreciable frequency components up to perhaps 5,000 cycles per second. It is desired to measure the spectrum amplitude \( |V(\omega)| \) at a selected frequency \( \omega \).

The output \( 1(t) \) of the truncation operation is

\[
1(t) = v(t)[u(t + T/2) - u(t - T/2)]
\]

where \( u(t) \) is the unit step-function at the time origin \( t = 0 \). The transform of \( u(t + T/2) - u(t - T/2) \) may be taken to be
Figure 3. Idealized Model to Describe the Operation of the System of Figure 1.
\[
\frac{e^{j\omega T/2} - e^{-j\omega T/2}}{j\omega} = \frac{2}{\omega} \sin \left(\frac{\omega T}{2}\right)
\]

Since \( l(t) \) is the product of two time functions as shown by (2), the Fourier transform \( L(\omega) \) of \( l(t) \) may be found by frequency convolution.* The result is

\[
L(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} V(\gamma) \frac{2}{(\omega - \gamma)} \sin \left(\frac{\omega - \gamma T}{2}\right) d\gamma
\]

The function \( l(t) \) is not band limited in the frequency domain even if \( v(t) \) is assumed to be band-limited. However, if \( T \) is sufficiently large then \( L(\omega) \) closely approximates \( V(\omega) \). This can be seen by considering an alternate form for the transform \( L(\omega) \).

\[
L(\omega) = \int_{-T/2}^{T/2} v(t) e^{-j\omega t} dt.
\]

It is seen by use of (1) and (5) that

\[
|V(\omega) - L(\omega)| = \left| \int_{-T/2}^{T/2} v(t) e^{-j\omega t} dt + \int_{-\infty}^{\infty} v(t) e^{-j\omega t} dt \right| \leq \int_{-T/2}^{T/2} |v(t)| dt + \int_{T/2}^{\infty} |v(t)| dt.
\]

The spectrum error obtained in processing \( l(t) \) with the truncation operation can be determined or bounded for a specific time function by use of (6).

2. Repetition

The truncated time function \( l(t) \) is duplicated every \( T \) seconds by the repetition operation forming a periodic function of time \( r(t) \) having a period of \( T \) seconds. Each period of \( r(t) \) is a repetition of the waveform \( v(t) \) within the interval \(-T/2, T/2\) seconds. The periodic waveform \( r(t) \) can be described by the Fourier series**

\[
r(t) = \sum_{n=-\infty}^{\infty} R e^{jnw_0 t}
\]

where the fundamental frequency is \( w_0 = 2\pi/T \) and the Fourier coefficients are

---

*Reference 12, page 27, ** page 43.
The transform of \( r(t) \) may be represented in terms of impulses as

\[
R(\omega) = 2\pi \sum_{n=-\infty}^{\infty} R \delta(\omega - n\omega_o)
\]

where \( \delta(\omega) \) is the unit impulse at the frequency origin \( \omega = 0 \). It is seen from (9) that \( l(t) \) is represented by a discrete spectrum having components separated by frequency intervals of \( \omega_o = 2\pi/T \).

By use of (1) and (8) it is seen that if \( T \) is sufficiently large then \( TR_n \) is a good approximation to \( V(\omega) \) at the frequencies \( \omega = n\omega_o = 2\pi n/T, n = 1, 2, \ldots \)

The error in this approximation is

\[
V(n\omega_o) - TR_n = \int_{-T/2}^{T/2} v(t)e^{-jn\omega_o t}dt + \int_{-\infty}^{\infty} v(t)e^{-jn\omega_o t}dt.
\]

This is precisely the same error that is found by (6) in considering \( L(\omega) \) as an approximation to \( V(\omega) \) at frequencies \( \omega = n\omega_o \).

3. Sampling

In the model of Figure 3 the periodic function of time \( r(t) \) is sampled at all time instants \( t_q = qd, q = \ldots -2, -1, 0, 1, 2, \ldots \)

Here, it is assumed that an even number \( N \) of samples is taken in the observation time \( T \) for the original system of Figure 1. A slight modification would be necessary to provide a more exact model if \( N \) is assumed to be an odd integer. The transform of the resulting infinite train of impulses \( s(t) \) may be represented as

\[
S(\omega) = \frac{1}{d} \sum_{q=-\infty}^{\infty} R(\omega - q\omega_s)
\]

with \( \omega_s = 2\pi/d \).

In conformity with the operations in the model of Figure 3 the transform \( R(\omega) \), shown in (9), is not a band-limited frequency function. Thus, there is theoretically no sampling rate above which "aliasing" errors (errors due to

---

*Reference 12, page 43, **Reference 18, page 96
overlap in the spectral compositions shown in (11)) can be eliminated. However, such errors are expected to be small provided the sampling rate is substantially more than twice the highest frequency present in \( v(t) \). A calculation of aliasing errors can be made by making appropriate use of (9), (10), and (11).

The fundamental frequency \( \omega_0 \) for the waveform \( r(t) \) and the sampling rate \( \omega_s \) are related by the equation \( \omega_s = N \omega_0 = T \omega_0 / d \).

4. Holding

The train of impulses \( s(t) \) is subjected to a holding operation. For a fixed system, the holding procedure can be represented as a linear operation on the function \( s(t) \). The Fourier transform of the output \( h(t) \) of the holding operation is

\[
H(\omega) = G_h(\omega)S(\omega)
\]

where \( G_h(\omega) \) is the transfer function for the holding operation and \( S(\omega) \) is the transform shown in (11) for the impulse train \( s(t) \).

If a zero-order hold is utilized, then

\[
G_h(\omega) = \frac{2 \sin(\pi \omega / \omega_s)}{\omega \omega_s e^{-j \pi \omega / \omega_s}} \\
\text{with } \omega_s = 2\pi / d,
\]

If a first-order hold is utilized, then

\[
G_h(\omega) = \frac{1 + j2\pi \omega / \omega_s}{2\pi / \omega_s} \frac{4(\sin \pi \omega / \omega_s)^2}{\omega_s e^{j2\pi \omega / \omega_s}} \\
\text{with } \omega_s = 2\pi / d,
\]

In both the above cases the transfer function \( G_h(\omega) \) provides a filtering action that attenuates high frequencies.

5. Frequency Expansion

The Fourier transform \( H(\omega) \) of the output \( h(t) \) of the holding operation is given by (12). The Fourier transform \( C(\omega) \) of the output \( c(t) \) of the frequency expansion operation is

\[
C(\omega) = H(K\omega).
\]

* Reference 18, page 134, ** page 138
The discrete spectral components in $H(\omega)$, the transform of $h(t)$, are spaced at frequency intervals of $\omega_o = 2\pi/T = 2\pi/\text{Nd} = \omega_s/N$. The discrete spectral components in $C(\omega)$, the transform of $c(t)$, are spaced at frequency intervals of $\omega_o = \omega_s/N$.

6. Mixing

It is assumed that the waveform $c(t)$ is mixed with a fixed frequency sinusoid $f(t) = E \cos \omega_f t$.

The resulting output is

$$m(t) = E c(t) \cos \omega_f t.$$  

The Fourier transform of $m(t)$ is

$$M(\omega) = \frac{E}{2} [C(\omega + \omega_f) + C(\omega - \omega_f)].$$  

where $C(\omega)$ is the transform of $c(t)$ given in (15). Essentially, the mixing operation amounts to a translation of the spectrum of $C(\omega)$ in frequency by an amount $\omega_f$.

7. Narrow-Band Filtering

The Fourier transform of the input $m(t)$ to the narrow-band filtering operation is $M(\omega)$ as given by (17). The Fourier transform of the output $b(t)$ of the narrow-band filtering operation is

$$B(\omega) = C_b(\omega)M(\omega).$$

Here, $C_b(\omega)$ is the filter transfer function. This expression may be used to determine the error arising because of imperfect filtering action. The transfer function may be assumed to be that of an ideal band-pass filter, or it may be selected to approximate more closely the characteristics of a physically realizable filter.

The discrete spectrum of the function $b(t)$ can be used to calculate the spectrum measurement for $|V(\omega)|$ that is given by the voltage measurement device in the system of Figure 1.

*Reference 12, page 127, **page 120"
V DISCUSSION OF ERROR ANALYSIS

The error in using the system of Figure 1 for the measurement of the spectrum amplitude of a deterministic function v(t) can be determined by use of the input-output relationships that have been presented for the operations in the model of Figure 3. This model has been selected to describe the errors that are inherent in the operation of the system of Figure 1.

It is seen that for many of the operations in the model the error in the measured spectrum at a single frequency depends on the entire spectrum of the input signal. This point might be overlooked in a cursory examination of the problem since it is tempting to assume that errors in the measured spectrum at a given frequency depend only on the input spectrum at that frequency. This suggests that it may be appropriate to investigate error bounds for classes of input waveforms rather than attempt to find a method of determining an exact error for every possible input condition.

It is apparent that determination of the total error due to all the operations of the model of Figure 3 is very lengthy and complicated. Even so, it is possible that a useful way may be found to characterize the total error in terms of reasonably simple system parameters. This possibility should be investigated. Also it appears desirable to undertake the more straightforward task of performing an independent error analysis for each of the component operations within the model. The results of these analyses could collectively be used to provide estimates of the total system error.

All the individual operations included in the model of Figure 3 are used extensively in the electrical engineering field. Analyses of all these operations have appeared in textbooks and the periodical literature. A representative collection of these articles is listed in the bibliography. During this investigation no analysis has been found in the literature for a system similar to that of Figure 1.

It is estimated that an error analysis of the type just described for the system of Figure 1 would require about six additional man-weeks of effort.

The problem of analysis for more general inputs, including random inputs, appears considerably more difficult than the one just described. An idealized model similar to that shown in Figure 3 may still be used; however, the input-output relationships would be considerably more complicated than the ones presented in this report. It is estimated that a minimum of eighteen weeks would be required to obtain significant results for an error analysis applying to the measurement of the spectrum of a random input.
VI BIBLIOGRAPHY


A STUDY OF ERROR APPROXIMATION
FOR HYBRID COMPUTERS

By Joseph L. Hammond, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473
(Development of New Methods and Applications of Analog Computation)

7 December 1965

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
SUMMARY

This report presents the theoretical portion of a study of error analysis for the solution of a class of problems on a hybrid computer. The class of problems considered are those dynamic systems which can be described by an equation of the form

\[ \dot{x} = f(x, y, t) \]

where \( y \) is a known vector function of time, \( x \) is an unknown state vector, \( f \) is a vector valued function, and \( t \) is time. The hybrid computer is assumed to be used in a mode for which both the analog and digital components process the dependent problem variables, with the analog components performing primarily integration and the digital components performing multiplication and nonlinear operations.

In the study a general linear equation for error is derived. This equation can be time varying and involves a vector forcing function which contains as components the errors resulting in the digital portion of each computational loop. This equation is not felt to be practical from a computational standpoint, but it can account for most dynamic errors and some types of static error.

Several approximate methods for solving the linear error equation are analyzed in the report. The most attractive of the methods investigated involves adjoining an approximation to the general error equation to the original equation and solving both equations on the hybrid computer. To make the results tractable, only dynamic errors are considered under conditions for which errors due to digital execution time and non-zero sampling interval predominate. These two sources of error are shown to have the same effect to a first approximation and thus the results involve only the sampling interval as a parameter.

Plans for evaluating the method by solving representative problems have been made, and definite conclusions will be drawn after the next phase of the study.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Summary</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **Introduction:** This report presents the results of a part of the first phase of a study of hybrid computation. In this first phase, an attempt is made to formulate a tractable error analysis for a nontrivial class of hybrid computer problems, and the report documents a completed study of the theoretical aspects of such an analysis. An experimental evaluation of the theoretical results is presently underway and will be reported when the work is completed.

At the present state of the art there exists a number of different types of hybrid computers and a number of different modes of operation for such computers. Clymer in a recent paper [1] classifies computers into no less than sixteen categories, eleven of which could be called hybrid computers. A recent study of error analysis for hybrid computers made on the present contract [2] indicates that although there is much basic work pertaining to hybrid computer errors there is at the present time no exact or approximate method for error analysis which applies in general for a nontrivial mode of hybrid operation.

The mode of hybrid operation chosen for this analysis is that for which both the analog and digital computer elements process the dependent problem variables. The chosen allocation of functions between the analog and digital section of the computer assigns integration as the primary function of the analog equipment with most other operations being performed digitally. The relatively simple operations of addition and multiplication by a constant are assigned to either type of equipment. This allocation of functions seems reasonable since analog integrators are accurate and offer a time advantage in performing integrations in parallel; and digital equipment can perform nonlinear operations such as function generation and multiplication with more accuracy, and sometimes with more facility, than the corresponding analog equipment.

---

* Original plans called for presenting in this report the results of applying the theoretical error analysis in at least one simple example. The desirability of having this report in print as soon as possible, however, has caused a change to the plans indicated.
The choice of this mode of operation was dictated partially by the fact that, with a further restriction on the class of problems to be solved, a tractable error analysis can be made. However, a study of possible modes of hybrid computer operation* indicates that the mode chosen represents an important use of such computers. In fact, a coarse classification of present day modes of operation for hybrid computers would probably include: (1) the mode for which both analog and digital elements process the dependent variables, (2) the mode for which the digital elements control the analog elements and only the analog elements process the dependent variables, and (3) combinations of these two modes. The error analysis to be presented applies to a large class of problems solved using operation as in category (1) above.

In addition to choosing a particular mode of hybrid operation, it is necessary to choose a particular class of problems in order to develop an error analysis procedure. The broadest class of equations pertaining to dynamic systems seems to be the class defined by

\[ \dot{x} = f(x, y; t) \]  

(1.1)

where \( x \) is a vector of unknowns, \( y \) is a vector of known functions, \( t \) is time and \( f(\cdot) \) is a vector valued function of the indicated arguments. All dynamic systems cannot, of course, be modeled by (1.1), but a large number of useful dynamic systems, both linear, nonlinear, stationary, and time variable can be so represented.

In developing an error analysis procedure for a hybrid computer solution of equations of the form (1.1), it is also necessary to specify the types of component error which will be considered. A partial listing of the more important types of component error includes: (1) errors due to limited frequency response of the analog integrators, (2) errors due to analog passive component inaccuracies, (3) errors due to the finite resolution of potentiometers and initial condition sources, (4) round-off error due to finite word lengths in the digital components, (5) errors due to digital execution time, and (6) errors due to a non-zero sampling interval for the A/D converters.

* See for example Korn [3] section 11-12 or Clancy [4].
Most of these errors can be classified as primarily static or primarily dynamic, although there is some overlapping of the effects of particular errors. For example, round-off error is primarily a static error, but increasing the word length to decrease this error could cause an increase in errors due to digital execution time. In spite of possible overlapping effects of particular errors, it is usually possible to consider static and dynamic errors separately, and this phase of the study is concerned with dynamic errors.

Dynamic errors are all related to the time scale factor chosen for a particular problem. For a fixed scale factor, dynamic errors are caused primarily by limited analog integrator frequency response, digital execution time and the sampling rate of the A/D converters. In most practical hybrid equipment digital execution time and sampling rate cause effective bandwidth limitations an order of magnitude more severe than that caused by analog equipment, and thus analog bandwidth limitations will be neglected.

A consideration of the remaining sources of dynamic errors -- digital execution time and A/D sampling rate -- shows that to a first approximation these two sources of error have the same overall effect. Among other justifications that can be made for this statement is a result of Miura [5] who derives an approximate expression for the transfer function of the digital portion of a hybrid computer. His result can be expressed as

\[ T(s) = Ae^{-\frac{(\tau + \delta/2)s}{2}} \]

where \( T(s) \) is the transfer function in question, \( A \) is a constant, \( \tau \) is the digital execution time, and \( \delta \) is the sampling period. In (1,2) note that an effective sampling period can be defined to account for both the actual sampling period and the digital execution time. Thus, in much of the analysis below a non-zero sampling period \( \delta \) is assumed to be the only source of error.

Section 2 of the report derives a linear equation for error subject to the type of operation just described. The equation obtained is not tractable in its most general form, and later sections of the report consider various approximations which lead to a computationally feasible result.
2. Derivation of a Linear Equation for Error: As discussed above, the class of problems to be considered in the present phase of this study involve solving equations of the form

\[ \dot{x} = f(x, y; t) \quad (2.1) \]

where \( x \) is an \( n \) vector of unknowns, \( y \) is an \( m \) vector of input functions, \( t \) is time and \( f(\cdot) \) is a vector valued function of the indicated arguments. In a less compact notation, (2.1) is a set of \( n \) first order equations of the form

\[ \begin{align*}
\dot{x}_1 &= f_1 (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m; t) = u_1(t) \\
\vdots & \quad \vdots \\
\dot{x}_n &= f_n (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m; t) = u_n(t).
\end{align*} \quad (2.2) \]

An ideal computer program for solving this set of equations is shown in Figure 2.1(a). Figure 2.1(b) gives a more compact notation for the same program.

For the particular use of the hybrid computer being considered here, the program shown in Figure 2.2(a) is used. In this figure the solution vector is denoted \( \bar{x} \) since it will not in general be the true solution \( x \) produced by the ideal circuit of Figure 2.1. Note that the integrator inputs, produced ideally by \( f \) operating on \( x \), are produced in the hybrid computer by first converting the analog \( \bar{x} \) to digital form, then generating the functions \( f_c \) with the digital portion of the computer, and finally converting the digital signals to analog signals. It is convenient to define a composite function \( F \) to account for all three of these operations; and using \( F \), the hybrid circuit can be represented by the diagram of Figure 2.2(b).

The objective of the present study is to obtain a measure of the "error" between \( x \) and \( \bar{x} \). To be of practical value, the error should serve as a guide in scale factoring and other considerations in setting up problems for the

\( ^{\dagger} \) A part of the procedure for error analysis developed in this study requires that the functions \( f_1 \) be analytic.
Figure 2.1. An Ideal Computer Program to Solve Equation (2.1).
Figure 2.2. A Hybrid Computer Program to Solve Equation (2.1).
hybrid computer. At the same time it must be possible to obtain an expression for the error with much less computational difficulty than solving the original problem.

The approach to be used in setting up an error equation can be motivated as follows. Consider the hybrid program of Figure 2.2(a) and compare it to the ideal program of Figure 2.1(b). It is plausible to represent the errors caused by the A/D and D/A conversions and the digital generation of the function \( f \) as noise added to the ideal circuit of Figure 2.1(b). Further, if the errors are small, a linear equation which specifies the error to a good approximation can be obtained. There is much precedent in the literature for this general type of approach [6], [7], [8]. The details of obtaining a linearized error equation, however, can be carried out in many different ways, and the method developed in this study is felt to offer several advantages in this particular application.

Consider the ideal equations (2.2) and the equations

\[
\begin{align*}
\dot{x}_1 &= F_1 (x_1^*, \ldots, x_n^*, y_1, \ldots, y_m; t) = U_1(t) \\
\vdots \\
\dot{x}_n &= F_n (x_1^*, \ldots, x_n^*, y_1, \ldots, y_m; t) = U_n(t).
\end{align*}
\]  

(2.3)

which apply for the hybrid program of Figure 2.2(b). Define a vector error \( \chi \) by the equation

\[
\chi(t) = x(t) - x^*(t) \tag{2.4}
\]

which is equivalent to the set of equations

\[
\begin{align*}
\gamma_1(t) &= x_1(t) - x_1^*(t) \\
\vdots \\
\gamma_n(t) &= x_n(t) - x_n^*(t).
\end{align*}
\]  

(2.5)
Now by use of (2.1), (2.3), and (2.4) \( \dot{y} \) can be expressed as

\[
\dot{y} = f(x^* + y, y; t) - F(x^*, y; t)
\] (2.6)

which is equivalent to

\[
\begin{align*}
\dot{y}_1(t) &= f_1(x^* + y, y; t) - F_1(x^*, y; t) \\
&\vdots \\
\dot{y}_n(t) &= f_n(x^* + y, y; t) - F_n(x^*, y; t).
\end{align*}
\] (2.7)

Note that the right hand side of the \( i \)th equation of (2.7) is the difference between \( F_i \) evaluated for an argument of \( (x^*, y; t) \) and \( f_i \) evaluated for an argument which differs from that of \( F_i \) in that \( y \) is added to \( x^* \). It is convenient to place in evidence on the right hand side of (2.7) two terms, namely: one which accounts for differences in \( f_i \) and \( F_i \) for the same argument, and one which accounts for differences in \( f_i \) evaluated for differences of \( y \) in the \( x \) argument. This can be accomplished by the artifice of adding and subtracting \( f(x^*(t), y(t); t) \) from the right hand side of (2.6) to obtain

\[
\dot{y} = f(x^* + y, y; t) - f(x^*, y; t) + f(x^*, y; t) - F(x^*, y; t). \quad (2.8)
\]

In (2.8), note that \( f(x^* + y, y; t) - f(x^*, y; t) \) is the difference in the functional values of \( f \) for differences of \( y \) in its \( x \) argument, while the term \( f(x^*, y; t) - F(x^*, y; t) \) gives the difference in the functional values of \( f \) and \( F \) for the same argument.

The case of small \( y \) is of primary interest, and if each \( f_i(x^* + y, y; t) \) is analytic each can be expanded in a power series which can be truncated to provide an approximation to \( f_i(x^* + y; y; t) - f(x^*, y; t) \). The function \( f_i(x^* + y; y; t) \) can be expressed as
where it should be noted that each partial derivative term is in general dependent on both time and the particular value of the input vector \( \mathbf{x} \). The symbol \( O(|\gamma|^2) \) denotes a term \( T \) which is at most the order of \( |\gamma|^2 \), i.e.,

\[
\lim_{|\gamma| \to 0} \frac{T}{|\gamma|^2} = c < \infty.
\]

Neglecting the term which is \( O(|\gamma|^2) \), (2.9) can be used with (2.8) to obtain an approximate equation for \( \mathbf{x} \) which when written out becomes

\[
\gamma_1 = \gamma_1 \frac{\partial f_1}{\partial x_1} + \gamma_2 \frac{\partial f_1}{\partial x_2} + \ldots + \gamma_n \frac{\partial f_1}{\partial x_n} + f_1(\mathbf{x}^*, \mathbf{y}; t) - F_1(\mathbf{x}^*, \mathbf{y}; t)
\]

\[ \vdots \]

\[
\gamma_n = \gamma_1 \frac{\partial f_n}{\partial x_1} + \gamma_2 \frac{\partial f_n}{\partial x_2} + \ldots + \gamma_n \frac{\partial f_n}{\partial x_n} + f_n(\mathbf{x}^*, \mathbf{y}; t) - F_n(\mathbf{x}^*, \mathbf{y}; t)
\]

For notation convenience define \( a_{ij}(t) \) as

\[
a_{ij}(t) = \frac{\partial f_i}{\partial x_j},
\]

so that (2.10) becomes

\[
\gamma_1 = a_{11}(t) \gamma_1 + a_{12}(t) \gamma_2 + \ldots + a_{1n}(t) \gamma_n + f_1(\mathbf{x}^*, \mathbf{y}; t) - F_1(\mathbf{x}^*, \mathbf{y}; t)
\]

\[ \vdots \]

\[
\gamma_n = a_{n1}(t) \gamma_1 + a_{n2}(t) \gamma_2 + \ldots + a_{nn}(t) \gamma_n + f_n(\mathbf{x}^*, \mathbf{y}; t) - F_n(\mathbf{x}^*, \mathbf{y}; t),
\]

or in vector notation

\[
\dot{\mathbf{x}} = A(t)\mathbf{x} + f(\mathbf{x}^*, \mathbf{y}; t) - F(\mathbf{x}^*, \mathbf{y}; t).
\]
For any given problem to be solved on the hybrid computer, the inputs $\mathbf{y}$ and the functions $f$ and $F$ are fixed: $f$ is fixed by the problem to be solved, and $F$ is fixed by the particular hybrid configuration chosen for use. Note that (2.11) or equivalently (2.12) specifies the error by a set of linear differential equations with time varying coefficients and forcing functions prescribed by the $f(x^*, y; t) - F(x^*, y; t)$. The solution to this set of equations gives an approximation to $\mathbf{y}$ within an error which is at most the order of

$$|\mathbf{y}|^2 = y_1^2 + y_2^2 + \ldots + y_n^2.$$ 

3. The Forcing Functions $f(x^*, y; t) - F(x^*, y; t)$: Equation (2.12) expresses the vector error $\mathbf{y}$ as the solution of a set of linear first order equations which have $f(x^*, y; t) - F(x^*, y; t)$ as the forcing functions. This section considers these forcing functions and derives approximate mathematical expressions for them.

Any element of the vector $f(x^*, y; t)$, for example $f_i(x^*, y; t)$, is a real valued function of time which is denoted $u_i(t)$. If computation begins at $t = 0$, then $u_i(t)$ is defined for $t > 0$. $u_i(t)$ can be visualized as the output of an ideal function generator (generating $f_i$) in response to an input of $(x^*, y; t)$. The functional form of $f_i$ is known as a part of the problem specification; but $u_i(t)$, which depends on $x^*$, is not generally known precisely before solving the problem.

The element of the vector $F(x^*, y; t)$ which corresponds to $u_i(t)$ is denoted $U_i(t)$. This quantity is the input to the $i$th integrator in the actual program for solving the given problem. From another point of view, $U_i(t)$ is the output of a digital to analog converter which is the last element in the group of elements which digitally generates $F_i$ as an approximation to $f_i$. 

10
The most important difference between $U_i(t)$ and $u_i(t)$ lies in the fact that $U_i(t)$ is an analog signal constructed by using some fixed rule to extrapolate between the sample points of the digital computer output.

In further discussion of $U_i(t)$ in this section it will be assumed that the digital computer generates $f_i(x^*, y; t)$ without error, and that the analog signal $U_i(t)$ is obtained by passing the digital signal through a first order sample and hold device. These assumptions serve to make the analysis more tractable while preserving the essential features of the hybrid device.

Using the stated assumptions, $U_i(t)$ becomes a staircase function with step values $u_i(k\delta)$, where $\delta$ is the interval between samples and $k = 0, 1, 2, \ldots$. A typical $u_i(t)$ along with the corresponding $U_i(t)$ is sketched in Figure 3.1(a).

The function of primary interest in this study is $\epsilon_i(t) = u_i(t) - U_i(t)$, and this function for the illustrative $u_i(t)$ of Figure 3.1(a) is given in Figure 3.1(b). In order to study $\epsilon_i(t)$ it is useful to express $u_i(t)$ by power series expansions about the sample points $k\delta$. An expansion of this sort for any value of $k$ is given by

$$u_i(t) = u_i(k\delta) + (t - k\delta) \dot{u}_i(k\delta) + \frac{(t - k\delta)^2}{2!} \ddot{u}_i(k\delta) + \cdots, \quad (3.1)$$

for any $k = 0, 1, 2, \ldots$.

To account for the sample and hold operation, use can be made of a short rectangular pulse $I_k(t)$, which can be constructed as the product of two step functions. The equation is

$$I_k(t) = \mu_{-1}(t-k\delta) u_{-1}((k+1)\delta-t) \quad (3.2)$$

where $\mu_{-1}(t)$ is the step function. Several members of the class $I_k(t)$ are sketched in Figure 3.2.

Using the functions just discussed, $u_i(t)$, $U_i(t)$ and $\epsilon_i(t)$ can be
Figure 3.1. A Typical $U_i(t)$ With Corresponding $u_i(t) - U_i(t)$. 
Figure 3.2. Representative Members of The Class of $I_k(t)$ Functions.
expressed as

\[ u_1(t) = \sum_{k=0}^{\infty} \left[ u_1(k\delta) + (t-k\delta) \frac{d}{dt} u_1(k\delta) + \frac{(t-k\delta)^2}{2!} \frac{d^2}{dt^2} u_1(k\delta) + \ldots \right] I_k(t) \]  

(3.3)

\[ U_1(t) = \sum_{k=0}^{\infty} u_1(k\delta) I_k(t) \]  

(3.4)

\[ \epsilon_1(t) = \sum_{k=0}^{\infty} \left[ (t-k\delta) \frac{d}{dt} u_1(k\delta) + \frac{(t-k\delta)^2}{2!} \frac{d^2}{dt^2} u_1(k\delta) + \ldots \right] I_k(t) \]  

(3.5)

Note that \( \epsilon_1(t) \) is the sum of an infinite collection of periodic functions each of whose amplitudes is determined on each basic interval by the value of the appropriate derivative of \( u_1(t) \) evaluated at \( k\delta \). It is convenient to define a collection of basic periodic functions by the equations

\[ b_1(t) = \sum_{k=0}^{\infty} (t-k\delta) I_k(t) \]

\[ b_2(t) = \sum_{k=0}^{\infty} \frac{(t-k\delta)^2}{2!} I_k(t) \]  

(3.6)

\[ b_n(t) = \sum_{k=0}^{\infty} \frac{(t-k\delta)^n}{n!} I_k(t) \]

The first four of these basic periodic functions are sketched in Figure 3.

Using the notation of (3.6), (3.5) becomes

\[ \epsilon_1(t) = \sum_{s=1}^{\infty} b_s(t) u_1^{(s)}(k\delta) \]  

(3.7)

Frequently the series of (3.7) can be truncated to approximate \( \epsilon_1(t) \).

---

1 Rules for extrapolating between the digital sample points to construct \( U_1(t) \) are frequently obtained by truncating the series for \( u_1(t) \) expressed by (3.3). A first order sample and hold, which is being assumed here, is obtained by using only the first term in the series. A second or higher order sample and hold could be obtained by using more terms in the series of (3.3) to produce \( U_1(t) \). Note that the series for \( \epsilon_1(t) \) consists of the terms in the series for \( u_1(t) \) which are neglected in defining \( U_1(t) \). Thus the approach being used here could be generalized in a straightforward fashion to account for more complicated extrapolation rules.
Figure 3.3. The Functions $b_1(t)$, $b_2(t)$, $b_3(t)$ and $b_4(t)$ Plotted for a Typical Period.
In some cases it is useful to obtain the magnitude of the amplitude spectrum of $\varepsilon_i(t)$ by taking the magnitude of the Fourier transform of both sides of (3.7). To expedite this calculation it is convenient to replace $\varphi_i^{(s)}(k\delta)$ by $\varphi_i^{(s)}(t)$ and assume that this approximation has little effect on the resulting spectrum. Thus

$$E_i(\omega) = |\Gamma[\varepsilon_i(t)]| = \sum_{s=1}^{\infty} |\Gamma[b_s(t)](\varphi_i^{(s)}(t))| = \sum_{s=1}^{\infty} |\Gamma[b_s(t)]| * |\Gamma[\varphi_i^{(s)}(t)]| \quad (3.8)$$

where $E_i(\omega)$ is the magnitude spectrum of $\varepsilon_i(t)$, $\Gamma[f(t)]$ is the Fourier transform of $f(t)$, and $X_1(\omega) * X_2(\omega)$ denotes the convolution of $X_1$ and $X_2$. Using the definitions

$$|\Gamma[b_s(t)]| = B(s;\omega); \quad |\Gamma[\varphi_i^{(s)}(t)]| = V_i(s;\omega),$$

(3.8) becomes

$$E_i(\omega) = \sum_{s=1}^{\infty} B(s;\omega) * V_i(s;\omega). \quad (3.9)$$

Each $b_s(t)$ can be considered periodic for $t>>0$, and in such a case each $b_s(t)$ has a discrete spectrum. The integral of $E_i(\omega)$ can thus be expressed by the Stieltjes integral

$$\hat{E}_i(\omega) = \sum_{s=1}^{\infty} \int \hat{V}_i(s;\omega-z) \, dB(s;z)$$

where

$$\hat{E}_i(\omega) = \int E_i(u) \, du; \quad \hat{V}_i(\omega) = \int V_i(u) \, du; \quad \hat{B}(\omega) = \int B_i(u) \, du. \quad (3.10)$$

The fact that $\hat{B}(s;z)$ has jumps of magnitude $B_s(\omega_r)$ at the frequencies $\omega_r$ and is constant elsewhere makes it possible to evaluate the integral of (3.10) to obtain

$$\hat{E}_i(\omega) = \sum_{s=1}^{\infty} \sum_{r=0}^{\infty} B_s(\omega_r) \hat{V}_i(s;\omega-r). \quad (3.11)$$
Differentiating both sides of (3.11) with respect to \( \omega \) then yields

\[
E_i(\omega) = \sum_{s=1}^{\infty} \sum_{r=0}^{\infty} B_s(\omega_r) V_i(s; \omega-\omega_r) .
\] (3.12)

The property of the Fourier transforms of derivatives gives

\[
V_i(s; \omega-\omega_r) = (\omega-\omega_r)^s V_i(0; \omega-\omega_r)
\] (3.13)

so that (3.12) can be expressed as

\[
E_i(\omega) = \sum_{r=0}^{\infty} V_i(0; \omega-\omega_r) \{ \sum_{s=1}^{\infty} (\omega-\omega_r)^s B_s(\omega_r) \}. 
\] (3.14)

The \( B_s(\omega_r) \) are fixed numbers which can be obtained from the Fourier series expansions of the \( b_s(t) \) defined by (3.6). In each expansion \( \omega_r \) is given by

\[
\omega_r = \frac{2\pi r}{\delta}, \quad r = 0, 1, 2, \ldots
\] (3.15)

The first three \( B_s(\omega_r) \) are given by

\[
B_1(0) = \frac{\delta}{2}; \quad B_1(\omega_r) = \frac{\delta}{\pi r}, \quad r = 1, 2, \ldots
\]

\[
B_2(0) = \frac{\delta^2}{6}; \quad B_2(\omega_r) = \frac{\delta^2}{2\pi r} \sqrt{\frac{4/(2\pi r)^2 + 1}{r = 1, 2, \ldots}}
\] (3.16)

\[
B_3(0) = \frac{\delta^3}{24}; \quad B_3(\omega_r) = \frac{\delta^3}{2\pi r} \sqrt{\frac{1/(2\pi r)^2 + [2/(2\pi r)^2 - 1/3]^2}{r = 1, 2, \ldots}}
\]

With reference to (3.14) the sum in braces, \( \{ \} \), can be considered as a filter having a magnitude transfer function, \( T(\omega) \), with "windows" at \( \omega = 0, 2\pi/\delta, 4\pi/\delta, \ldots \), such as sketched in Figure 3.4(a). The signal \( u_i(t) \) can then be visualized as being applied to this filter in such a way that its frequency spectrum, \( V_i(0, \omega) \), is applied directly to the first window, and then is shifted so that the spectrum is centered at each of the other
Figure 3.4. $T(\omega)$ Versus $\omega$ and a Representative Input Spectrum.
windows as shown in Figure 3.4(b). An enlarged sketch of the shape of the windows at zero and the shape assumed by the windows for large \( r \) is given in Figure 3.5.

For accurate computation the highest sensible frequency components of \( V_i(0, \omega) \) must lie in the region of large attenuation from \( T(\omega) \), say \( \delta \omega < 0.1 \), (reference Figure 3.5). Thus \( \omega_{\text{max}} \) for \( V_i(0, \omega) \) must satisfy an inequality such as

\[
\omega_{\text{max}} < \frac{0.1}{\delta}
\]  

(3.17)

For most problems, the time scaling necessary to insure that the inequality (3.17) holds also causes the error equation to be unresponsive to frequencies as high as the second and subsequent windows. Thus for accurate computation (3.14) can be approximated by

\[
E_i(\omega) = V_i(0, \omega) T(\omega)
\]  

(3.18)

and \( T(\omega) \) in turn can be approximated by

\[
T(\omega) \approx \frac{\omega \delta}{2}
\]  

(3.19)

Equations (3.18) and (3.19) then give the result that, where they apply, \( \epsilon_i(t) \) has the same amplitude spectrum as a signal given by \( \delta/2 \dot{u}_i(t) \).

The result that \( \epsilon_i(t) \) can under some conditions be approximated by \( \delta/2 \dot{u}_i(t) \) can be obtained qualitatively in an alternative fashion by considering (3.7) along with (3.6). From (3.6) as \( \delta \) approaches zero it seems reasonable to neglect all terms but \( b_i(t) \) and thus approximate \( \epsilon_i(t) \) by the equation

\[
\epsilon_i(t) = \sum_{k=0}^{\infty} (t-k\delta) I_k(t) \dot{u}_i(k\delta).
\]  

(3.20)
Figure 3.5. Typical Shapes for "Windows" in $T(\omega)$. 

$\pi r T_r(\omega)$ or $T_o(\omega)$
In deriving (3.18) it was assumed that for small computational error the problem is scaled so that the error equation is unresponsive to frequencies as high as say $2\pi/\delta$. An equivalent assumption is that the error equation acts as an averaging filter on $\epsilon_1(t)$, and that $\epsilon_1(t)$ can thus be approximated by the equation

$$\epsilon_1(t) = \text{ave} \left\{ \sum_{k=0}^{\infty} (t-k\delta)I_k(t) \right\} \hat{u}_1(k\delta)$$

(3.21)

$$\epsilon_1(t) = \delta/2 \hat{u}_1(t).$$

This is the same result as that obtained above.

4. Discussion of the Error Equation: Using the notation of section 3 expressing $f(\mathbf{x}, \mathbf{y}; t) - F(\mathbf{x}, \mathbf{y}; t)$ as $\epsilon(t)$, (2.12) becomes

$$\dot{\mathbf{y}} = A(t) \mathbf{y} + \epsilon(t).$$

(4.1)

This is the state vector form of the error equation. The solution to this equation for the type of $\epsilon(t)$ discussed in section 3 gives the desired errors as functions of time.

Unfortunately, it is only practical in exceptional cases to obtain an exact solution to (4.1); and further, an analytical approach even with approximate methods would seem prohibitively tedious. One reason that an analytical approach is not practical lies in the fact that the forcing functions for (4.1) depend on the machine solutions of the given equations.

Section 5 is devoted to the details of a machine approach to estimating error. The present section is intended to discuss several aspects of an analytical solution to (4.1) even though an entirely analytical approach is not considered practical. Equation (4.1) will be discussed under the headings of stability and general solutions.
Before giving attention to these topics, it is useful to express (4.1) as an equivalent \( n \)th order equation for \( y_1 \), the first component of the error vector. This equation is most useful in the case that \( A(t) \) is independent of time, and such will be assumed to be the case here. In this case, (4.1) can be converted into an \( n \)th order transformed equation for \( y_1 \) by taking the Laplace transform of the set of equations (4.1) and then solving for \( \Gamma_1(s) \), the Laplace transform of \( y_1 \). The result can be expressed as

\[
|sI-A| \Gamma_1(s) = E_1(s) |sI-A|_{11} + E_2(s) |sI-A|_{21} + \ldots + E_n(s) |sI-A|_{ni}
\]

(4.2)

where \( E_i(s) \) is the Laplace transform of \( \varepsilon_i(t) \), \( |sI-A| \) is the determinant of the matrix \( sI-A \), \( |sI-A|_{ij} \) is the cofactor of the element of \( sI-A \) in the \( i \)th row and the 1st column, \( I \) is the identity matrix and \( s \) is the complex frequency variable. For convenient reference the determinant of \( sI-A \) is given explicitly by

\[
|sI-A| = \begin{vmatrix}
  s-a_{11} & -a_{12} & \cdots & -a_{1n} \\
  -a_{21} & s-a_{22} & \cdots & \vdots \\
  \vdots & \vdots & \ddots & \vdots \\
  -a_{n1} & \cdots & \cdots & s-a_{nn}
\end{vmatrix}
\]

(4.3)

In the special cases of \( n = 2 \) and \( n = 3 \), (4.2) becomes

\[
\begin{align*}
\text{n=2} & \quad \left[ s^2 - (a_{11} + a_{22})s + a_{11}a_{22}a_{12}a_{21} \right] \Gamma_1(s) = (s-a_{22})E_1(s) + a_{12}E_2(s) \quad (4.4) \\
\text{n=3} & \quad \left[ s^3 - (a_{11} + a_{22} + a_{33})s + [a_{11}a_{22} + a_{11}a_{33} - a_{12}a_{21} - a_{13}a_{31} + a_{22}a_{33} - a_{23}a_{32}]s \\
& \quad + [a_{11}a_{22}a_{33} - a_{11}a_{22}a_{13}a_{31} - a_{12}a_{21}a_{33} + a_{13}a_{31}a_{22} - a_{13}a_{21}a_{32} - a_{23}a_{32}] \right] \Gamma_1(s) = \\
& \quad \left[ s^2 - (a_{22} + a_{33})s + a_{22}a_{33}a_{23}a_{32} \right] E_1(s) + (a_{21}s-a_{12}a_{33}-a_{13}a_{32}) E_2(s) \\
& \quad + (a_{13}s-a_{12}a_{23}-a_{13}a_{22}) E_3(s).
\end{align*}
\]

(4.5)
Stability: Equation (4.1) for the error is always linear in the present development, but in general it has time variable coefficients and a forcing function $e(t)$ which is always present. The stability of the solution of equations of the form (4.1) is discussed in the literature by a number of authors, for example [9], [10], and [11].

Of the several types of stability defined and discussed in these references, the type referred to by Zadeh* as stability in the sense that any bounded input produces a bounded output, (i.e., b.i.b.o.) seems to have the most direct application. In general a system of equations of the form (4.1) is stable b.i.b.o. if for all $t_0$, for all initial states $\gamma(t_0)$, and for all bounded inputs $e(t)$ on $[t_0, \infty]$ the solution $\gamma(t)$ is bounded on $[t_0, \infty]$. In the specific case of the error equation (4.1), $\gamma(t_0)$ is always zero. The input $e(t)$ has the typical component

$$e_1(t) = f_1(x^*, y; t) - F_1(x^*, y; t),$$

and for each component $F_1(x^*, y; t)$ is always bounded since this quantity is a physical machine variable. The quantity $f_1(x^*, y; t)$ is a mathematical variable which could be unbounded. However each function $f_i(\ )$ must be specified mathematically for all values of the argument and thus if there are values of the argument for which any $f_i(\ )$ is unbounded this fact will be known a priori.

Thus the error equation is stable b.i.b.o. if, given that all $f_i(\ )$ are bounded for all values of the arguments, the solution $\gamma(t)$ is bounded on

*Reference [9], page 385.

**This assumption is stronger than necessary since the $f_i(\ )$ need only be bounded for the argument values actually occurring in the problem solution. However, refinements along this line would seem to have little application in the present study.
[t_0^{\infty}]. Use of this concept of stability can be made in the special cases for which stability b.i.b.o. can be established by a relatively simple test applied to (4.1). Then knowing that (4.1) is stable, if the f_i( ) are bounded for all values of the arguments, it follows that \( \psi(t) \) is bounded on \([t_0^{\infty}]\).

Unfortunately simple tests for stability in the general case do not exist so that stability for the general case is not pursued further as a practical part of this error analysis study. *

In the special case that (4.1) is time invariant, tests for stability can give practical results. Equation (4.1) is time invariant if the elements of \( A(t) \) are constants and in this (4.1) is stable b.i.b.o. if and only if all the eigenvalues of \( A \) have nonpositive real parts and those eigenvalues of \( A \) that lie on the imaginary axis are simple zeros of the minimal polynomial of \( A \). **

An equivalent statement of stability can be determined for (4.2), the single \( n \) th order differential equation for \( y_1(t) \). Tests for the stability of linear constant coefficient differential equations are well known and amply documented in the literature. *** A useful sufficient condition is that the zeros of the polynomial \( sI - A \), in (4.2), lie in the left half of the complex \( s \) plane, and many convenient methods exist for testing to see if this is the case.

* Lyapunov stability theory offers the only approach known to the author to finding a general test for stability. This approach is discussed in the references [9], [10], and [11] cited above.

** This statement of the stability theorem is given by Zadeh, [9], page 375. A minimal polynomial of \( A \) is the polynomial \( \psi(\lambda) \) of least degree such that \( \psi(A) = 0 \) and the coefficient of the highest power of \( \lambda \) is unity.

*** See for example Truxal [12].
The application of stability analysis to the error equation would have the objective of a gross evaluation of a particular program. Computation with a program for which the error equation is unstable would obviously be undesirable if it could be avoided. However, it should be pointed out that instability b.i.b.o. as discussed above only implies that the error is unbounded somewhere on \([t_{0}, \infty]\). Thus it is not impossible that useful results could be obtained on some subinterval of \([t_{0}, \infty]\) even if the error equation is unstable. Conversely if the error equation is known to be stable this guarantees only that the error is bounded and not that it is small.

**General Solutions**: Equation (4.1) for the error is linear with potentially time varying coefficients. The forcing functions \(e(t)\), which are discussed in section 3, are determined by the vector valued functions \(f(\ )\) and \(F(\ )\) evaluated at the argument \((x^{*}, y; t)\). Since \(x^{*}\) is the computed solution vector for the equation being solved, \(e(t)\) is not known precisely prior to the computation although its general structure is usually known.

The solution of linear equations with time varying coefficients is a classic topic in differential equation theory and is discussed by many authors including Coddington and Levinson [13], and Ince [14]. For \(e(t)\) completely specified, as would be the case after \(x^{*}\) is computed, the theoretical approaches given in the above references will apply. However, these analytical approaches are not felt to offer a practical means of evaluating error in the general case.

Zadeh [15], [16], [17], among others, has developed approximate but more tractable procedures for analytically determining the solutions to linear time varying \(n^{th}\) order differential equations which include the type of (4.1) written as an \(n^{th}\) order equation for \(\gamma(t)\). The procedure given by Zadeh in [17] is perhaps the most tractable. It assumes a slow variation of the parameters in the differential equation and approximates the impulse response \(\omega(t, \xi)\) by a
truncated series. Each term of the series can be computed in a straightforward fashion using Laplace transforms, and once $\omega(t, \xi)$ is determined, $\gamma_1(t)$ can be computed from the expression

$$\gamma_1(t) = \int_{-\infty}^{\infty} \omega(t, \xi) g[\varepsilon_1(\xi), \varepsilon_2(\xi), \ldots, \varepsilon_n(\xi)] \, d\xi. \quad (4.6)$$

where $g(\cdot)$ is a known function of $\varepsilon_1, \ldots, \varepsilon_n$ which results when (4.1) is reduced to an $n^{th}$ order equation for $\gamma_1(t)$. This procedure, although well defined computationally, would seem to require implementation on a computer in any but trivial cases and thus affords no advantages over the procedure to be given in section 5.

The only case for which an analytical procedure could yield either appreciable insight or computational results would seem to be the case for which (4.1) either has time independent coefficients or can be reduced to an approximate equation which does. Assuming this to be the case the Laplace transform method applies, and for known $z(t)$ an analytical solution to either (4.1) or (4.2) can be obtained in a straightforward manner. The computational labor involved depends on the nature of the $z(t)$ and the order of the equation.

Consider (4.2) which, when solved for $\gamma_1(s)$ becomes

$$\gamma_1(s) = \frac{|sI - A|}{|sI - A|} E_1(s) + \frac{|sI - A|}{|sI - A|} E_2(s) + \ldots + \frac{|sI - A|}{|sI - A|} E_n(s). \quad (4.7)$$

Note the order of the polynomials $|sI - A|$ and the $|sI - A|_{ii}$, $i = 1, 2, \ldots, n$. Examination of (4.3) shows that $|sI - A|_{11}$, which is the determinant consisting of the elements of $sI - A$ less those in the first row and first column, is a polynomial of order $n-1$. All other cofactors, $|sI - A|_{ii}$ for $i = 2, 3, \ldots, n$, are polynomials of order $n-2$, and $|sI - A|$ is a polynomial of order $n$. Thus by considering the initial value theorem for Laplace transforms or a Bode plot,
it is clear that rapid changes in $e_1(t)$ can potentially have a greater effect on $y_1(t)$ than rapid changes in any other $e_i(t)$. This indicates that feedback through a digital element around the integrator whose output is $x_1^*(t)$ produces more error than similar feedback around other integrators. Considerations of this sort, visualizing (4.7) as a filter with inputs $e_1(t)$, $e_2(t)$... $e_n(t)$ and an output $y_1(t)$, would seem to be useful in many cases to determine qualitatively the behavior of $y_1(t)$. In such considerations the curves and discussion of section 3 concerning the spectrum of $e_1(t)$ should also be useful.

The discussion of section 3 indicated that each $e_i(t)$ could be visualized as the sum of a number of basic periodic functions each with an amplitude modulated by the value of some derivative of $u_i(t)$, (see (3.7) and Figure 3.3). A method due to Truxal* could find some application in expressing the $y_1(t)$ caused by using a single basic periodic function to approximate one of the $e_i(t)$. Truxal's method greatly facilitates Laplace transform calculations when the driving function is periodic. His method consists of: (a) computing a "transient" solution for all $t$, (b) computing the total output during the first period, and (c) computing the "steady state", (defined as the total output minus the transient output), during the first period. With this information the output during any period can be determined as the sum of the transient output during the period in question and the steady state output.

5. Machine Methods of Solving the Error Equation: The error equation (4.1) expresses the error in a hybrid computer solution of the class of equations given by (2.1). Although (4.1) is linear, an examination of possible general and approximate solutions to this equation, as discussed in section 4, indicates that in most cases an analytical solution is not practical. In this section a procedure is developed for obtaining an error equation which can be adjoined

* Ref. [12], pages 48-51.
to the given equation and solved with it on the hybrid computer.

As discussed in section 4, one serious problem in solving (4.1) lies in the fact that $e(t)$ is a collection of functions which depend on $x^*$, the computed solution vector for (2.1). A machine solution of (4.1) which proceeds concurrently with the machine solution of (2.1) has the very definite advantage of having available the functions

$$F_1(x^*, y; t) = U_1(t)$$
$$\vdots$$
$$F_n(x^*, y; t) = U_n(t)$$

as well as $x^*(t)$ and $y(t)$. A procedure is now outlined for machine computation of an approximation to the $y(t)$ which uses these functions, but does not require further knowledge of the true solution to (2.1).

The procedure makes use of (3.21) as an approximation to each $e_i(t)$ so that

$$e_1(t) = \delta/2 \dot{u}_1(t)$$
$$\vdots$$
$$e_n(t) = \delta/2 \dot{u}_n(t).$$

The conditions which make (3.21) or (5.1) a reasonable approximation are discussed in section 3 and will be summarized and restated below.

Using (5.1) in (4.1) yields

$$\ddot{\psi}(t) = A(t)\dot{\psi}(t) + \delta/2 \dot{u}(t); \quad \dot{\psi}(t=0) = 0$$

where $\dot{\psi}(t)$ denotes the approximation to $\chi(t)$ and $u(t) = \{u_1(t), \ldots, u_n(t)\}$. Equation (5.2) can be rearranged to obtain
\[
\frac{d}{dt} [\psi(t) - (\delta/2)u(t)] = A(t)\psi(t),
\]

and a vector \( z(t) \) can be defined by the equation

\[
z(t) = \psi(t) - (\delta/2)u(t).
\]

Using (5.4) in (5.3) then yields

\[
\dot{z}(t) = A(t)z(t) + (\delta/2)A(t)u(t); \quad z(t=0) = -(\delta/2)u(t=0)
\]

The components of \( u(t) \) are the values of the ideal functions \( f_i(x, y; t) \) which are not available for use in computation. Thus a further approximation is made by replacing \( u(t) \) in (5.5) by \( U(t) \) to obtain

\[
\dot{z}(t) = A(t)z(t) + (\delta/2)A(t)U(t); \quad z(t=0) = -(\delta/2)U(t=0)
\]

which when written out becomes

\[
\begin{align*}
\dot{z}_1 &= a_{11}z_1 + \ldots + a_{1n}z_n + \frac{\delta}{2}(a_{11}U_1 + \ldots + a_{1n}U_n); \quad z_1(t=0) = -(\delta/2)U_1(t=0) \\
\vdots & \quad \vdots \\
\dot{z}_n &= a_{n1}z_1 + \ldots + a_{nn}z_n + \frac{\delta}{2}(a_{n1}U_1 + \ldots + a_{nn}U_n); \quad z_n(t=0) = -(\delta/2)U_n(t=0)
\end{align*}
\]

where \( a_{ij}(t) = \partial f_i(x, y; t)/\partial x_j \). Approximations to any or all of the \( y \) can be obtained using (5.4) and (5.6). The appropriate expressions for the components of \( y \) are

\[
\begin{align*}
\gamma_1 &= z_1 + (\delta/2)U_1 \\
\vdots & \quad \vdots \\
\gamma_n &= z_n + (\delta/2)U_n
\end{align*}
\]

Note that (5.6) involves only the \( a_{ij}(t) \), which can be computed from the known \( f_i \), and the \( U_i \), which are available from the program for the equation being solved.
Equation (5.6) is very attractive computationally, but it necessarily involves several approximations which are reasonable only under restricted conditions. Possibly the most stringent condition is imposed by use of (5.1) to approximate the $c_i(t)$. For this assumption to be reasonable the problem being solved must be scaled so that highest sensible frequency components of all of the $U_i(t)$ are smaller than $0.1/\delta$. Furthermore, the error equation must be unresponsive at frequencies on the order of $2\pi/\delta$ and higher. In addition to these conditions, it should be recalled that the sample and hold operation of the hybrid computer is the only nonideal characteristic accounted for in the analysis. Finally, for the method of setting up the original error equation (2.10) to be valid, each component of $f(x, y; t)$ must have derivatives of all orders, and this of course restricts the class of problems for which the work applies. With respect to this latter restriction, however, it should sometimes be possible to approximate a nonanalytic $f_i$ by an analytic function in setting up the error equation. A complete procedure for machine analysis of error is now outlined.

A Procedure for Machine Analysis of Hybrid Computer Error

Given: 1. An equation of the form

$$\dot{X} = f(X, Y; t')$$

where $X$ is an $n$ vector of unknowns, $Y$ is an $m$ vector of knowns and $f$ is a vector valued function with each component analytic in $X$.

2. The equation is to be solved by a hybrid computer programmed as in Figure 2.1 (assuming scale factors to be applied.)

3. The predominant error is caused by the sample and hold operation of the digital part of the computer.

Error Computation Procedure: 1. Estimate the highest sensible frequency components of the $U_j(t)$ signals in Figure 2.1 and choose a time scale factor so that these frequencies are less than $0.1/\delta$, where $\delta$ is the interval between samples of the A to D converters. This yields a scaled equation of the form
2. Evaluate

\[ a_{ij}(t) = \frac{\partial f_i(x, y; t)}{\partial x_j} \]  

(5.9)

for all \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, n \). (If some \( f_i \) is not analytic, it may be possible to approximate it by an analytic function.)

3. Consider the equation

\[ \dot{x} = A(t) x + \varepsilon(t). \]

Perform any analytical calculations that can be done with reasonable labor. Specifically estimate whether or not the components of \( y \) of interest are affected appreciably by frequency components in the \( \varepsilon(t) \) higher than \( 2\pi/6 \). (The discussion of sections 3 and 4 should be helpful in these considerations.) Rescaling is necessary if frequency components of \( \varepsilon(t) \) higher than \( 2\pi/6 \) affect the error components of interest.

4. Set up the equation

\[ \dot{x}(t) = A(t) x(t) + \frac{\delta}{2} A(t) u(t); x(t=0) = -\frac{\delta}{2} u(t=0) \]  

(5.10)

and adjoin (5.8) and (5.10) to obtain the computational equation

\[ \ddot{\tilde{\tilde{n}}} = G(n, y; t) \]

(5.11a)

where

\[
\tilde{n} = \begin{bmatrix} \dot{x} \\ \vdots \\ z \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} F(x^*, y; t) \\ \vdots \\ \frac{\delta}{2} A(t)F(x^*, y; t) \end{bmatrix}.
\]

Written out these equations become
\[ \dot{x}_1^* = F_1(x_1^*, \ldots, x_n^*, y_1, \ldots, y_m; t) \]
\[ \vdots \]
\[ \dot{x}_n^* = F_1(x_1^*, \ldots, x_n^*, y_1, \ldots, y_m; t) \]

(5.11b)

\[ \ddot{z}_1 = a_{11}z_1 + \ldots + a_{1n}z_n + \delta/2 a_{11}F_1(t) + \ldots + \delta/2 a_{1n}F_n(t) \]
\[ \vdots \]
\[ \ddot{z}_n = a_{n1}z_1 + \ldots + a_{nn}z_n + \delta/2 a_{n1}F_1(t) + \ldots + \delta/2 a_{nn}F_n(t), \]

with appropriate initial conditions.

Equation (5.11) is solved on the hybrid computer and appropriate members of the set

\[ \gamma_1 = z_1 + \delta/2 F_1(t) \]
\[ \vdots \]
\[ \gamma_n = z_n + \delta/2 F_n(t) \]

(5.12)

are then used to evaluate the error components of interest. Note that the terms on the right hand side of the equations for \( \ddot{z}_1 \ldots \ddot{z}_n \) in (5.11b) can be grouped as follows

\[ \ddot{z}_1 = a_{11}[z_1 + \delta/2 F_1(t)] + \ldots + a_{1n}[z_n + \delta/2 F_n(t)] \]
\[ \vdots \]
\[ \ddot{z}_n = a_{n1}[z_1 + \delta/2 F_1(t)] + \ldots + a_{nn}[z_n + \delta/2 F_n(t)]. \]

Each term in brackets is one of the \( \gamma_i \), and thus these quantities already appear in the program for (5.11) and need not be computed separately.
6. Bibliography:


Technical Note No. 9. Not issued.
THE GENERATION OF A GAUSSIAN RANDOM PROCESS IN A POSITION PARAMETER

By
David L. Finn
W. A. Yates

TECHNICAL NOTE NO. 10
on
Contract No. NASA-2473
(Development of New Methods and Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
ABSTRACT

A method is presented for approximating, by use of analog computer components, any prescribed stationary Gaussian random process depending only on a position parameter $x$. The output of the analog mechanization system simulates the effect of the prescribed random process on some physical device or sensing element whose variable location is specified by the position parameter $x$. The two inputs to the mechanization system are Gaussian white noise and the first derivative of the function of time describing the position of the sensing element. Best approximation of the prescribed random process is obtained when the second derivative of the function of time describing the position of the sensing element is restricted to small values.
The research work reported in this technical note was done by Dr. D. L. Finn and Mr. W. A. Yates in association with Project No. A-588 of the Engineering Experiment Station at the Georgia Institute of Technology. This is in support of the program of the Simulation Branch of the Computation Laboratory at the Marshall Space Flight Center. The work done by Mr. Yates, a doctoral fellow in the School of Electrical Engineering, has been carried out as a portion of his doctoral thesis research.
CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Page</td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>Preface</td>
<td>ii</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>iv</td>
</tr>
<tr>
<td>I. INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>II. RANDOM PROCESSES DEPENDENT ON A POSITION PARAMETER</td>
<td>3</td>
</tr>
<tr>
<td>2.1 General Description</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Transformation into Random Processes Dependent on a Time Parameter</td>
<td>3</td>
</tr>
<tr>
<td>2.3 Mathematical Characterization</td>
<td>5</td>
</tr>
<tr>
<td>III. THE MECHANIZATION PROCEDURE</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Derivation of an Equation Characterizing the Mechanization System</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Adjustment of Level of White Noise Generator</td>
<td>13</td>
</tr>
<tr>
<td>3.3 A Mechanization System</td>
<td>13</td>
</tr>
<tr>
<td>IV. EXAMPLE, EXPONENTIAL AUTOCORRELATION FUNCTION</td>
<td>17</td>
</tr>
<tr>
<td>V. CONCLUSIONS</td>
<td>24</td>
</tr>
<tr>
<td>Bibliography</td>
<td>25</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.1</td>
<td>Sample Functions of a Random Process Depending on a Position Parameter</td>
</tr>
<tr>
<td>2.2</td>
<td>Transformation of a Position Parameter Process $g(x)$ into a Time Parameter Process $g(x(t))$ by Specification of Position as a Function of Time $x(t)$</td>
</tr>
<tr>
<td>3.1</td>
<td>Mechanization of the Composite Process $g(x(t))$ with $x(t) = t$</td>
</tr>
<tr>
<td>3.2</td>
<td>Mechanization of the Composite Process $g(x(t))$ with $x(t) = vt + K$</td>
</tr>
<tr>
<td>3.3</td>
<td>Approximation of the Composite Process $g(x(t))$</td>
</tr>
<tr>
<td>3.4</td>
<td>A Mechanization System for the Composite Process $g(x(t))$</td>
</tr>
<tr>
<td>4.1</td>
<td>Mechanization System for Exponential Autocorrelation Function</td>
</tr>
<tr>
<td>4.2</td>
<td>Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 1</td>
</tr>
<tr>
<td>4.3</td>
<td>Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 2</td>
</tr>
<tr>
<td>4.4</td>
<td>Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 3</td>
</tr>
<tr>
<td>4.5</td>
<td>Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 4</td>
</tr>
</tbody>
</table>
I. INTRODUCTION AND SUMMARY

In Technical Note No. 5 for this project a special class of stochastic processes was defined. These processes were designated as partially stationary stochastic processes. This class has potential usefulness in the development of methods of generating nonstationary stochastic processes for simulating the random wind disturbances that affect a rocket in flight.

A stochastic process representing random wind disturbances necessarily depends on one or more position parameters as well as time. For example, the wind disturbance affecting a rocket in flight might be represented by a random process $g(x, t)$. Here, $x$ is taken to represent the altitude of the rocket at the time instant $t$. For a particular rocket flight the altitude is described by a function of time $x(t)$. In this case, the wind disturbance affecting the rocket is represented by the composite process $g(x(t))$. The composite process $g(x(t))$ depends only on a time parameter.

In Technical Note No. 3 for this project a synthesis method was given for the generation of nonstationary time-parameter random processes having a prescribed first and second moment. This synthesis method can be used in the simulation of the random disturbance affecting a rocket in flight following a prescribed trajectory. The present research effort has been directed toward the development of simulation methods that do not require a prior specification of the flight trajectory.

It appears that the most difficult aspect of the mechanization of a process of the type $g(x, t)$ mentioned above is obtaining the required dependence on the position parameter $x$. For this reason the preliminary work that is reported in this technical note has been directed toward the generation of a random process that depends only on a position parameter. A study of possible applications listed in Chapter II indicates that the mechanization procedure presented in this technical note has substantial potential usefulness in its own right. More important, the present study has provided techniques that appear to be applicable in the more general problem of generating a random process depending on both a position and time parameter.

Chapter II of this technical note is devoted to a general discussion of random processes that depend on a position parameter. Chapter III presents a derivation of the transfer function that is used as a basis for the approximation of a stationary Gaussian random process depending on a single position parameter $x$. In general for good approximation the position parameter, which
is taken to describe the position of some physical element, is constrained to have a small second derivative with respect to time. This means that the physical element must have a small acceleration.

Section 3.3 of this technical note presents one mechanization system that generates a random process of the type described in Section 3.1. Chapter IV contains the results of a theoretical and experimental study of a mechanization system that generates a stationary Gaussian process having an exponential autocorrelation function. It is shown that this system theoretically provides an exact realization of the desired random process.
II. RANDOM PROCESSES DEPENDENT ON A POSITION PARAMETER

A general description of random processes depending on a position parameter is given in this chapter. Several applications of possible interest in analog simulation studies are mentioned for random processes depending on a position parameter. The transformation of a given position-parameter process into a composite process depending on time is discussed. The composite process evaluates the effect of the position-parameter process on some physical element whose position varies with time.

2.1 General Description

A random process (or stochastic process) \( g(x) \), \(-\infty < x < \infty\) is an indexed collection of random variables. The process may be characterized as an ensemble of sample functions associated with a probability measure. In this technical note the sample functions of the process \( g(x) \) are considered to depend on a position parameter \( x \).

Sample functions of a random process of the type under consideration might typically have an appearance as shown in Figure 2.1. Many different physical processes conceptually might be represented as random processes depending on a position parameter. The sample functions depicted in Figure 2.1 might represent the random surface variations of an airport runway or a highway. They might represent surface irregularities in a channel or tube that is guiding the movement of a liquid or gaseous substance. They possibly could depict small random variations in the density or hardness of a solid material that is being processed by a saw, lathe, or drill press. The sample functions might simulate variations in the width or thickness of a long ribbon of material subjected to a rolling operation in an industrial plant. In all cases, the sample functions are taken to represent some type of random irregularity whose value is dependent only upon its position with respect to a spatial coordinate system.

2.2 Transformation into Random Processes Dependent on a Time Parameter

For the physical processes mentioned in the previous section a physical device or sensing element of some kind whose location in space may be described by a position coordinate \( x \) is affected by a random process \( g(x) \) depending only on the position coordinate. As the sensing element changes position a function of time \( x(t) \) is generated that describes the instantaneous location of the element. At the position \( x \) the effect of the random process on the sensing element is described by the value of \( g(x) \). When the position is a function of
Figure 2.1. Sample Functions of a Random Process Depending on a Position Parameter.
time \( x(t) \) the effect of the random process on the sensing element at the instant of time \( t \) is described by the value of the composite process \( g(x(t)) \). A typical sample function for a process \( g(x) \) is shown in Figure 2.2. Also, the corresponding sample function for the composite time dependent process \( g(x(t)) \) is shown for an assumed time variation of position \( x(t) \).

The fact that the composite process \( g(x(t)) \) is dependent upon a time parameter provides the potentiality that the effect of the process \( g(x) \) on a sensing element having position \( x(t) \) may be simulated by an analog computer system operating in the time domain.

### 2.3 Mathematical Characterization

Gaussian random processes are utilized in the mechanization system whose synthesis is presented in this technical note. A Gaussian process is uniquely characterized by specification of its mean and its autocorrelation function.

The mean \( m_g \) of a random process \( g(x) \) is defined as the expected value of the random variable generated by the process for a fixed position \( x \).

\[
  m_g = E(g(x)). \tag{2.1}
\]

The autocorrelation function \( R_g(x_1, x_2) \) is defined as

\[
  R_g(x_1, x_2) = E(g(x_1)g(x_2)). \tag{2.2}
\]

A random process is said to be stationary in the wide-sense if its mean and autocorrelation function are independent of the origin for position \( x \). For a wide-sense stationary random process there results:

\[
  m_g = C. \tag{2.3}
\]

\[
  R_g(x_1, x_2) = E(g(x_2 + \tau)g(x_2)) = R_g(\tau), \quad \tau = x_1 - x_2.
\]

Here, the mean is a constant and the autocorrelation function depends only on the position difference \( \tau = x_1 - x_2 \).

---

* Reference 2, page 145.
Figure 2.2. Transformation of a Position Parameter Process $g(x)$ into a Time Parameter Process $g(x(t))$ by Specification of Position as a Function of Time $x(t)$.
The power spectral density $S_g(\omega)$ for a wide-sense stationary random process is defined as

$$S_g(\omega) = \int_{-\infty}^{\infty} R_g(\tau)e^{-j\omega\tau}d\tau, \quad (2.4)$$

The autocorrelation function may be expressed in terms of the power spectral density as

$$R_g(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_g(\omega)e^{j\omega\tau}d\omega, \quad (2.5)$$

---

III. THE MECHANIZATION PROCEDURE

In this chapter a synthesis procedure is presented for a mechanization system to approximate composite random processes of the type described in Chapter II. Such a composite process represents a random effect, depending only on a space coordinate $x$, that is experienced by a physical device or sensing element having a variable position specified by the coordinate $x$. This variable position is described by a function of time $x(t)$.

The procedure is restricted to the generation of a Gaussian random process $g(x)$ that is stationary in a position parameter $x$. The associated composite process $g(x(t))$ is in general nonstationary in time. In general the function of time $x(t)$ should be constrained to have a small second derivative with respect to time in order for the output of the mechanization system to provide a close approximation to the composite process $g(x(t))$. This means that the physical device or sensing element should have a small acceleration.

It is known that for any random process having given mean and autocorrelation function, there exists a Gaussian process with identical mean and autocorrelation function.* Thus, if only the first two statistical moments of an arbitrary wide-sense stationary random process are of interest in a simulation study, the generation of an appropriate Gaussian process will suffice.

3.1 Derivation of an Equation Characterizing the Mechanization System

Without loss of generality the synthesis technique presented in this technical note is concerned with the realization of processes $g(x)$ having mean equal to zero. Processes having nonzero mean can be realized as the sum of the random process generated by this technique and the output of a source having constant output numerically equal to $m_g$.

It is assumed that the stationary Gaussian process $g(x)$ is such that if $x$ is replaced by $t$ then the time dependent process $g(t)$ can be realized by the application of Gaussian white noise to a linear filter. Further, the power spectral density is assumed to be expressed as a ratio of two polynomials in $\omega$. Thus restricted, the power spectral density may be expressed as

$$S_g(\omega) = \left|H_g(p)\right|^2 e^{-j\omega}$$

(3.1)

* Reference 8, page 3.
** Reference 2, page 227.
Here, $H(p)$ is the transfer function of a time invariant linear filter. This transfer function may be expressed as a ratio of two polynomials in complex frequency $p$.

$$H(p) = \frac{a_0 + a_1 p + a_2 p^2 + \ldots + a_m p^m}{b_0 + b_1 p + b_2 p^2 + \ldots + b_n p^n}.$$ (3.2)

A technique for determining the $a_k$ and $b_k$ in the transfer function of equation (3.2) is discussed in Reference 2, page 233. These are evaluated for a simple example in Chapter 4 of this technical note.

The transfer function $H(p)$ of (3.2) may be realized in a variety of ways by the use of analog computer elements. Specifically, this transfer function $H(p)$ may be realized by an appropriate mechanization of the differential equation shown in equation (3.3).

$$b_0 \frac{de}{dt} + b_1 \frac{d^2 e}{dt^2} + b_2 \frac{d^3 e}{dt^3} + \ldots + b_n \frac{d^n e}{dt^n} = a_0 w + a_1 \frac{dw}{dt} + a_2 \frac{d^2 w}{dt^2} + \ldots + a_m \frac{d^m w}{dt^m}.$$ (3.3)

If a Gaussian white noise random process $w(t)$ having a power spectral density of unity is applied to a linear filter having the transfer function $H(p)$ of (3.2) the output of the filter is a Gaussian random process $e(t)$ having power spectral density equal to $S_e(\omega)$ of (3.1) and corresponding auto-correlation function $R_e(t)$. Thus, the filter output $e(t)$ is a realization of the random process $g(x)$ where $x$ has been replaced by $t$. It is clear that the output $e(t)$ thus represents a realization of the composite random process $g(x(t))$ where the position of the sensing element is described by the function $x(t) = t$. This mechanization is depicted in Figure 3.1.

The mechanization of the composite process $g(x(t))$ when the position of the sensing element is described by the equation $x(t) = vt + K$ may be accomplished in a straightforward manner. Here, $v$ and $K$ are assumed to be arbitrary constants. It is noted that $v$ is the derivative of $x(t)$ with respect to time.

The mechanization just described is again accomplished by applying Gaussian white noise $w(t)$ to a linear filter having output $e(t)$. The output of the filter

*Reference 4, Reference 6.*
Figure 3.1. Mechanization of the Composite Process $g(x(t))$ with $x(t) = t$.

Figure 3.2. Mechanization of the Composite Process $g(x(t))$ with $x(t) = vt + K$.

Figure 3.3. Approximation of the Composite Process $g(x(t))$. 
is to be

\[ e(t) = g(x(t)) = g(vt + x). \]

(3.4)

The autocorrelation function of the output is

\[ R_v(t_2 + \tau, t_2) = E(g(vt_2 + vt + K)g(vt_2 + K)) = R_g(vt). \]

(3.5)

Thus, mechanization may be accomplished by providing a filter whose output \( e(t) \) has autocorrelation function:

\[ R_v(\tau) = R_g(vt). \]

(3.6)

The power spectral density for \( e(t) \) is

\[ S_v(\omega) = \int_{-\infty}^{\infty} R_v(\tau) e^{-j\omega\tau} d\tau \]

\[ = \int_{-\infty}^{\infty} R_g(v\tau) e^{-j\omega\tau} d\tau \]

\[ = \frac{1}{|v|} \int_{-\infty}^{\infty} R_g(\alpha) e^{-j\frac{\omega}{\alpha}} d\alpha \]

\[ = \frac{1}{|v|} S_g(\frac{\omega}{v}). \]

(3.7)

The mechanization may be accomplished by applying a Gaussian white noise random process \( w(t) \) having \( S_w(\omega) = 1 \) to a linear filter having transfer function \( H_v(p) \) such that

\[ \left| H_v(p) \right|^2 \bigg|_{p=j\omega} = S_v(\omega) = \frac{1}{|v|} S_g(\frac{\omega}{v}) = \frac{1}{|v|} \left| H_g(p) \right|^2 \bigg|_{p=j\omega}. \]

(3.8)

Thus, a satisfactory transfer function to accomplish the mechanization is

\[ H_v(p) = \frac{1}{\sqrt{|v|}} H_g\left( \frac{p}{v} \right) = \frac{1}{\sqrt{|v|}} \frac{a_0 + a_1 \frac{p}{v} + \ldots + a_m \frac{p^m}{v^m}}{b_0 + b_1 \frac{p}{v} + \ldots + b_n \frac{p^n}{v^n}}. \]

(3.9)
This transfer function may, again, be realized by a variety of methods. In particular, \( H_v(p) \) may be realized by a mechanization of the following differential equation.

\[
\sqrt{v(t)} \left( b_0 v^n e + b_1 v^{n-1} \frac{de}{dt} + \ldots + \frac{d^n e}{dt^n} \right) = a_0 v^n w + a_1 v^{n-1} \frac{dw}{dt} + \ldots + a_{n-1} v^{1} \frac{d^{n-1} w}{dt^{n-1}}.
\]

Here, it has been assumed, without loss of generality, that \( m = n-1 \) and that \( b_n = 1 \). This mechanization is depicted in Figure 3.2. It is noted that the transfer function \( H_v(p) \) may be realized by the use of a fixed parameter linear filter. However, a different filter must be utilized for each selection of \( v \).

Consideration of the differential equation of (3.10) shows a possible means for synthesizing a system that generates a random process that approximates the composite process \( g(x(t)) \) when \( x(t) \) is an arbitrary function of time having a second derivative restricted to small magnitudes. If the \( v \) in (3.10) is re-defined as

\[
v(t) = \frac{dx(t)}{dt},
\]

the differential equation of (3.10) becomes

\[
\sqrt{v(t)} \left( b_0 v^n e + b_1 v^{n-1} \frac{de}{dt} + \ldots + \frac{d^n e}{dt^n} \right) = a_0 v^n w + a_1 v^{n-1} \frac{dw}{dt} + \ldots + a_{n-1} v^{1} \frac{d^{n-1} w}{dt^{n-1}}.
\]

The differential equation of (3.12) may be mechanized by the use of standard analog computer elements such as multipliers and operational amplifiers. Such a mechanization is depicted in Figure 3.3.

The previous discussion has shown that the mechanization system depicted in Figure 3.3 generates exactly the composite process \( g(x(t)) \) for the case that \( x(t) = vt + K \) regardless of the fixed value assigned to \( v \). A change in the value of \( v \) is achieved merely by changing the value of the input \( v(t) \).

Because the system of Figure 3.3 provides an exact realization of \( g(x(t)) \) for any constant value of \( v(t) \) it is expected to provide a close approximation to a correct realization for slowly varying functions of time \( v(t) \) defined by (3.11). Thus, in general, the second derivative of \( x(t) \) is to be kept to a small value for best approximation.

It is to be emphasized that in general any analog computer realization of the transfer function of (3.9) provides a possibility of adaptation for approximating the composite process \( g(x(t)) \). To accomplish this adaptation, multipliers and other components are added to the system to allow \( v(t) \) to be applied as a system input. One method of accomplishing this is presented in the next section.

In general the errors obtained in approximating \( g(x(t)) \) are expected to depend on the procedure used to implement the transfer function \( H_v(p) \). A general analysis has not been made to evaluate errors in approximation. However, for the example presented in Chapter 4 it is shown that an exact realization of the composite process is obtained for arbitrary non-negative functions \( v(t) \).

### 3.2 Adjustment of Level of White Noise Generator

The random process \( g(x) \) is assumed to be stationary. Thus, the statistical mean squared value \( E(g(x(t))^2) \) of the composite process \( g(x(t)) \) is independent of the waveform \( x(t) \). This fact provides a convenient method of adjusting the white noise input of the system of Figure 3.3 to provide the desired mean squared value of \( g(x(t)) \). To accomplish this, the input \( v(t) \) is adjusted to any convenient fixed value. The output of the system is then a stationary Gaussian process in the time parameter \( t \). The timewise average of the squared output is then equal to the statistical average of the squared output. Thus an rms meter may be used to determine \( E(g(x(t))^2) \). The levels of the white noise input may be varied until the rms meter at the output indicates the correct mean squared value for the composite process.

### 3.3 A Mechanization System

It was noted in the previous section that any procedure that realizes the transfer function \( H_v(p) \) of (3.9) provides a possibility of adaptation for the approximation of the composite process \( g(x(t)) \). To accomplish this adaptation
it is necessary to modify the mechanization system so that \( v \), the time derivative of the position \( x \), can be applied as a time-function input to the system.

One method of realizing the transfer function \( H_v(p) \) is to define a new set of equations equivalent to (3.10) for fixed positive values of \( v \).

\[
e_1 = e
\]
\[
\frac{\text{d}e_1}{\text{d}t} = \sqrt{|v|} \left( e_2 + a_{n-1} w - b_{n-1} \sqrt{|v|} e_1 \right)
\]
\[
\frac{\text{d}e_2}{\text{d}t} = v(e_3 + a_{n-2} w - b_{n-2} \sqrt{|v|} e_1)
\]
\[
\frac{\text{d}e_3}{\text{d}t} = v(e_4 + a_{n-3} w - b_{n-3} \sqrt{|v|} e_1)
\]
\[
\frac{\text{d}e_4}{\text{d}t} = v(e_5 + a_{n-4} w - b_{n-4} \sqrt{|v|} e_1)
\]
\[
\vdots
\]
\[
\frac{\text{d}e_{n-1}}{\text{d}t} = v(e_n + a_1 w - b_1 \sqrt{|v|} e_1)
\]
\[
\frac{\text{d}e_n}{\text{d}t} = v(a_0 w - b_0 \sqrt{|v|} e_1).
\]

A mechanization for this set of equations is shown in Figure 3.4. The output of the system of Figure 3.4 provides an exact realization of the composite process \( g(x(t)) \) when the position parameter varies according to the equation \( x(t) = vt + K \). This realization is exact for any fixed value of \( v \). In accordance with the discussion of the preceding section the output of the system of Figure 3.4 is expected to provide a close approximation to the composite process \( g(x(t)) \) whenever the second derivative of \( x(t) \) is small.

To accomplish the mechanization the input \( v \) to the system is set equal to the derivative of \( x(t) \).

\[
v = \frac{\text{d}x(t)}{\text{d}t}.
\]  

*Reference 8, page 8.*
Figure 3.4. A Mechanization System for the Composite Process $g(x(t))$. 

$$w(t) \quad \text{white noise}$$

$$v = \frac{d x(t)}{dt}$$

$$-\sqrt{|v|} e_1$$

summers

$$b_{n-1} \quad a_{n-1}$$

$$b_{n-2} \quad a_{n-2}$$

$$b_{n-3} \quad a_{n-3}$$

$$+e_1$$

multipliers

$$+e_2$$

integrators

$$+e_3$$

$$(a_0 w - b_0 \sqrt{|v|} e_1)$$

$$-de_n/dt$$

$$+e_n$$
The $a_k$ and $b_k$ in the equations of (3.13) are found by use of the power spectral density $S_g(\omega)$ of the random process described in the discussion of equation (3.1).

$$S_g(\omega) = |H_g(p)|^2_{p=j\omega}$$  \hspace{1cm} (3.15)

$$H_g(p) = \frac{a_0 + a_1 p + a_2 p^2 + \cdots + a_{n-1} p^{n-1}}{b_0 + b_1 p + b_2 p^2 + \cdots + b_{n-1} p^{n-1} + p^n}.$$

Here, the $b_n$ and $m$ in equation (3.2) have been assumed to be $b_n = 1$ and $m = n-1$. 

16
IV. EXAMPLE, EXPONENTIAL AUTOCORRELATION FUNCTION

In this chapter a mechanization system is synthesized for the generation of the composite process \( g(x(t)) \) associated with a stationary Gaussian process \( g(x) \) having an exponential autocorrelation function. Curves are presented showing the measured autocorrelation functions for the output of the mechanization system compared with the autocorrelation function for \( g(x(t)) \). The measurements were processed by the procedure discussed in Technical Note No. 3. Ensembles of 200 sample functions each were used in computation of the measured value of the autocorrelation function.

The autocorrelation function for the stationary Gaussian process \( g(x) \) is assumed to be

\[
R_g(\tau) = \frac{\alpha^2}{2\beta} e^{-\beta|\tau|} \quad (4.1)
\]

The mean of \( g(x) \) is assumed to be zero. The power spectral density for \( g(x) \) is found by use of (2.4) to be

\[
S_g(\omega) = \frac{\alpha^2}{\omega^2 + \beta^2} \quad (4.2)
\]

By use of (3.1) the transfer function \( H_g(p) \) is

\[
H_g(p) = \frac{\alpha}{\beta + p} \quad (4.3)
\]

The transfer function \( H_y(p) \) defined by (3.9) is for the present example

\[
H_y(p) = \frac{1}{\sqrt{v}} \frac{\alpha}{\beta + \frac{p}{v}} \quad (4.4)
\]

The transfer function \( H_y(p) \) may be realized by mechanization of the following differential equation.

\[
\beta v e + \frac{de}{dt} = \alpha \sqrt{v} \quad (4.5)
\]

A simplified diagram of an analog computer network that mechanizes (4.5) is shown in Figure 4.1. The system of Figure 4.1 provides an exact realization.
Figure 4.1. Mechanization System for Exponential Autocorrelation Function.
of g(x) for a position variable x(t) = vt + K. In accordance with the discussion in Chapter 3 the system is expected to approximate the composite process g(x(t)) when the position parameter is a function of time x(t) having a small second derivative.

The quality of the approximation of g(x(t)) may be found by obtaining an analytical solution of the differential equation of (4.5). If v(t) is restricted to be a non-negative function of time, the general solution e(t) for the differential equation of (4.5) is

\[
e(t) = e^0 \int_0^t v(\alpha) d\alpha + \int_0^t \int_0^\alpha v(\gamma) w(\gamma) d\gamma + C e^0 \int_0^t v(\alpha) d\alpha.
\]

Assuming w(t) to be a stationary Gaussian white noise random process having power spectral density of unity, the autocorrelation function for e(t) is found to be

\[
E(e(t_1), e(t_2)) = \frac{2}{\alpha} e^{-\beta |x(t_1) - x(t_2)|} t_1 > t_2.
\]

This is exactly the autocorrelation function for g(x(t)). Thus, the mechanization of Figure 4.1 provides an exact realization of the composite process g(x(t)) when the first derivative of x(t) is restricted to be non-negative. Thus for the mechanization considered in this example, the second derivative of the position function x(t) need not be restricted to small values.

A comparison of theoretical values of the autocorrelation function of g(x(t)) with values determined experimentally from an implementation of the system of Figure 4.1 is shown in Figures 4.2 through 4.5. For this example the parameters of the power spectral density of (4.2) used were \( \alpha = 0.016 \) and \( \beta = 0.005 \).

*Reference 2, page 43.
Figure 4.2. Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 1.
Figure 4.3. Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 2.
Figure 4.4. Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 3.
Figure 4.5. Normalized Autocorrelation Function $R(t_1, t_2)$ for Position Function No. 4.
V. CONCLUSIONS

In this report a mechanization procedure has been presented for the approximation of a Gaussian random process in a position parameter \( x \). The mechanization system simulates the effect of the random process on a sensing element whose variable position is specified by the parameter \( x \). The variation in position described by the function \( x(t) \) is constrained to have a small second derivative.

A synthesis procedure for obtaining a particular mechanization system for approximating the desired random process is presented. Experimental results obtained from measurements on a physical mechanization system are given to indicate the quality of approximation to the desired random process.

The work reported in this technical note is part of a research effort directed toward the generation of a random process depending on both a time and position parameter. Such a random process has utility for the simulation of the random wind disturbances affecting a rocket in flight. It is felt that progress has been made during this investigation on the general problem of approximating a Gaussian random process depending on both a time and position parameter. Also, progress has been made toward the synthesis of a mechanization system to generate a random process depending on a single position parameter without applying the constraint that the second derivative of the position parameter must be small.
BIBLIOGRAPHY


TECHNICAL NOTE NO. 11
Research Project A-588

GENERATION OF NONSTATIONARY RANDOM PROCESSES
DEPENDENT ON TIME AND POSITION PARAMETERS

By David L. Finn

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation)

15 August 1966

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
GENERATION OF NONSTATIONARY RANDOM PROCESSES
DEPENDENT ON TIME AND POSITION PARAMETERS

By

David L. Finn

TECHNICAL NOTE NO. 11

on

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation

For

GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
ABSTRACT

Two methods are presented for the synthesis of analog computer networks that approximate a random process depending on both time and a position parameter. The networks are intended for use in the simulation of random wind disturbances that affect a rocket or other aerospace vehicle in flight. The output of the analog computer network simulates the effect of the prescribed random process on the vehicle as its position varies arbitrarily with time. The two inputs to the analog computer network are a Gaussian white-noise random process and a function of time characterizing the variable position of the vehicle.
TABLE OF CONTENTS

Page

TITLE PAGE ........................................................................................................... 1

ABSTRACT ........................................................................................................... ii

LIST OF ILLUSTRATIONS .................................................................................. iv

I. INTRODUCTION ............................................................................................... 1

II. THE COVARIANCE-EXPANSION SYNTHESIS PROCEDURE ...................... 3

   2-1. Introduction .......................................................................................... 3
   2-2. The Synthesis Procedure .................................................................... 3
   2-3. An Example ......................................................................................... 10

III. THE SPECTRAL-DENSITY SYNTHESIS PROCEDURE ............................. 16

   3-1. Introduction .......................................................................................... 16
   3-2. The Synthesis Procedure .................................................................... 16
   3-3. An Example ......................................................................................... 20

IV. CONCLUSIONS .............................................................................................. 24

BIBLIOGRAPHY .................................................................................................. 26

This report contains 26 pages.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Mechanization System Derived by the Covariance-Expansion Synthesis Procedure</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Mechanization System for the Realization of the Random Process of the Example in Section 2-3</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>Mechanization System Derived by the Spectral-Density Synthesis Procedure</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Mechanization System for the Realization of the Random Process of the Example in Section 3-3</td>
<td>22</td>
</tr>
</tbody>
</table>
In this technical note two methods are presented for the synthesis of analog computer networks that approximate a random process depending on both time and a position parameter. The networks are intended for use in the simulation of random wind disturbances that affect a rocket or other aerospace vehicle in flight. Each network produced by either of the synthesis procedures has two inputs. One input is a Gaussian white-noise process and the second input is a time function characterizing the position of the moving vehicle. The parameter characterizing position may be taken to be altitude in the wind disturbance application. The output of each network is a time-parameter Gaussian random process that approximates the instantaneous random disturbance affecting a rocket or other sensing element as it undergoes an arbitrary position variation in time. It is not necessary that the position variation of the sensing element be specified a priori. Thus, in the wind simulation application it is necessary to synthesize only a single network to represent the disturbances affecting vehicles having a wide variety of flight trajectories.

It is assumed that the first statistical moment of the time/position-parameter process being approximated is identically zero and that the second statistical moment is known. Processes having a nonzero first statistical moment may be generated by the addition of a deterministic function to the random process generated by one of the procedures described in this technical note. The output of the analog computer network derived by each of the procedures is a time-parameter Gaussian random process having first and second statistical moments approximating the moments of the original time/position-parameter random process.

The two synthesis procedures will be designated in this technical note as the "covariance-expansion method" and as the "spectral-density method." The covariance-expansion method is a direct adaptation of the synthesis method for the realization of time-parameter random processes that was presented in Technical Note No. 3 on this project (Reference 8) and further developed in the thesis of Reference 9. The spectral-density method is a direct adaptation of the synthesis method for the realization of position-parameter random processes that was presented in Technical Note No. 10 (Reference 3).

One method of obtaining experimental wind data utilizes a tracking
radar to record the effects of wind disturbance on the motion of an ascending Jimsphere balloon (Reference 7). This data may be used for the calculation of a covariance function that depends on altitude alone. Data in this form is best suited for use with the covariance-expansion synthesis procedure discussed in Chapter II of this technical note. The mechanization system derived by use of data in this form is best suited to simulate the wind disturbances that affect a vehicle in nearly vertical flight. However, the synthesis procedure of Chapter II is not at all restricted for use in vertical flight patterns. It is of greater generality than the method of Chapter III. At the same time, unfortunately, it is of greater mathematical complexity, and in general requires a greater amount of statistical data for implementation.

A second method of obtaining experimental wind data involves the recording of wind effects on a sensor maintained at fixed altitude (Reference 6). This data may be used to calculate a power spectral density function if it is assumed that the wind disturbance at a fixed altitude may be represented as a stationary random process. Data expressed in this form, for a succession of altitudes, is best suited for use with the spectral-density synthesis procedure discussed in Chapter III of this technical note. The mechanization system derived by use of data in this form is best suited to simulate the wind disturbances that affect a vehicle in nearly horizontal flight. It is to be noted that even though the wind disturbance is assumed to be a stationary random process depending on time for each fixed altitude, the process may be nonstationary in the altitude parameter.

Unlike the procedure of Chapter III, the covariance-expansion synthesis method of Chapter II makes no assumption of stationarity. The random process to be approximated may be nonstationary in both time and position parameters.

Chapter II of this technical note is devoted to a presentation of the covariance-expansion synthesis procedure. A simple example is worked to clarify the procedure. Chapter III is devoted to a presentation of the spectral-density synthesis procedure. An example is also worked in that chapter to demonstrate application of the procedure. Chapter IV contains a discussion of conclusions derived on the basis of the research work done to date and contains some comments concerning future plans for the research.
II. THE COVARIANCE-EXPANSION SYNTHESIS PROCEDURE

2-1. Introduction

This chapter is devoted to a presentation of the covariance-expansion synthesis procedure. The output of the analog computer network derived by this procedure approximates a random process \( g(z,t) \) depending on position \( z \) and time \( t \). Specifically, the output of the network approximates the composite random process \( x(t) = g(z(t),t) \). The process \( x(t) \) represents the effect of the original time/position random process on a sensing element having the position \( z \) at the time instant \( t \). The representation is exact provided the random process is Gaussian, has a covariance expansion of the form of (2.2), and provided the position variable \( z \) is a monotone function of time. Otherwise, the output of the analog computer network is an approximation to the original random process.

The notation used in this technical note has been chosen to conform to that used in Technical Note No. 3 (Reference 8). The synthesis procedure presented in this chapter is a direct adaptation of the procedure in Technical Note No. 3. In order to save space, those details of proof of validity of the procedure that appear in Technical Note No. 3 are not repeated here. Only those steps are included that are necessary to demonstrate implementation of the synthesis method.

2-2. The Synthesis Procedure

The time/position-parameter random process to be approximated is denoted as \( g(z,t) \). Here, \( z \) is a variable denoting the instantaneous position at time \( t \) of a sensing element that is affected by the random process.

The covariance function for the random process \( g(z,t) \) is denoted as

\[
r_g(z',t',z,t) = E[g(z_1,t_1)g(z_2,t_2)]
\]  

(2.1)

where \( E \) is the expectation operation and

\[
z' = \text{larger of } (z_1 \text{ and } z_2)
\]

\[
z = \text{smaller of } (z_1 \text{ and } z_2)
\]
Here it is assumed that the expected value of \( g(z,t) \) is identically zero. Processes with nonzero mean may be generated by the addition of \( g(z,t) \) to a deterministic function.

It is assumed that the covariance function of the random process \( g(z,t) \) may be expressed as a finite series expansion in the form

\[
 r_{g}(z', t', z, t) = \sum_{i=1}^{n} \eta_{i}(z', t') \theta_{i}(z, t) \tag{2.2}
\]

where \( \eta_{i}(z', t') \) and \( \theta_{i}(z, t) \) are known functions of time and position. If equation (2.2) does not apply exactly, the covariance function \( r_{g}(z', t', z, t) \) must be approximated by an expansion of this form.

It is recalled that the position variable \( z \) denotes the instantaneous position at time \( t \) of a sensing element that is affected by the random process \( g(z, t) \). Throughout this technical note it will be assumed that the position of the sensing element is described by a nondecreasing function of time \( z(t) \). However, the synthesis procedure developed is equally valid if the position is described by a nonincreasing function of time.

An analog computer network is to be synthesized having two inputs--a Gaussian white-noise waveform, and the function \( z(t) \) representing the instantaneous position of a sensing element. The output of the network is to be a composite random process

\[
x(t) = g(z(t), t) \tag{2.3}
\]

The composite time-parameter random process \( x(t) \) represents the effect of the time/position-parameter random process \( g(z,t) \) on the sensing element having the instantaneous position \( z(t) \).

The covariance function of the random process \( x(t) \) is denoted as \( r(t', t) \). By use of (2.3) and (2.1) this may be expressed as
\[
    r(t', t) = E [x(t_1) x(t_2)] = r_g(z(t'), t', z(t), t)
\]  

(2.4)

By use of (2.2) the covariance function \( r(t', t) \) may be represented as a finite series expansion

\[
    r(t', t) = \sum_{i=1}^{n} \phi_i(t') \gamma_i(t)
\]

(2.5)

where

\[
    \phi_i(t') = \eta_i(z(t'), t')
\]

\[
    \gamma_i(t) = \theta_i(z(t), t)
\]

The analog computer network to be synthesized is characterized by the nth order differential equation

\[
    x^{(n)} + p_{n-1}(t) x^{(n-1)} + \ldots + p_1(t) x^{(1)} + p_0(t) x = \eta_{n-1}(t) y^{(n-1)} + \ldots + \eta_1(t) y^{(1)} + \eta_0(t) y
\]

(2.6)

Here \( x^{(k)} \) denotes the \( k \)th derivative of the function \( x \) with respect to time. In shorter notation, (2.6) may be written as

\[
    L_t x = N_t y
\]

(2.7)

The function \( y \) represents the Gaussian white-noise input to the analog computer network.

In order to avoid differentiation of the noise input \( y \), the equation of (2.7) may be converted into a set of \( r \) first-order differential equations. To make this conversion the following identifications are utilized:

\[
    x(t) = x_1(t)
\]

(2.8)

\[
    x_{1}^{(1)} = x_2 = a_{n-1}(t) x_1 + b_{n-1}(t) y
\]
\[
x_2^{(1)} = x_3 - a_{n-2}(t) x_1 + b_{n-2}(t) y \\
x_{n-1}^{(1)} = x_n - a_1(t) x_1 + b_1(t) y \\
x_n^{(1)} = -a_0(t) x_1 + b_0(t) y
\]

This set can be written more concisely in matrix notation as

\[
\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{bmatrix} = A(t) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + B(t) y
\]

where

\[
X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \quad B(t) = \begin{bmatrix} b_{n-1}(t) \\ b_{n-2}(t) \\ \vdots \\ b_1(t) \\ b_0(t) \end{bmatrix}
\]

\[
H = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -a_{n-1}(t) & 1 & 0 & \cdots & 0 \\ -a_{n-2}(t) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1(t) & 0 & 0 & \cdots & 1 \\ -a_0(t) & 0 & 0 & \cdots & 0 \end{bmatrix}
\]

\[
A(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}
\]
The elements \( a_k, b_k \) in (2.8) are related to the coefficients \( p_k, q_k \) in (2.6) by

\[
p_k = \sum_{j=0}^{n-1-k} \frac{(n-1-j)!}{k! (n-1-j-k)!} a(n-1-j-k) \quad (2.10)
\]

\[
q_k = \sum_{j=0}^{n-1-k} \frac{(n-1-j)!}{k! (n-1-j-k)!} b(n-1-j-k) \quad (2.11)
\]

If the \( p_k, q_k \) are shown, then (2.10) and (2.11) can be solved sequentially for the \( a_k, b_k \).

Associated with the vector differential equation of (2.9) is the homogeneous equation

\[
x^{(1)} = A(t) x \quad (2.12)
\]

The \( \phi_1(t), \phi_2(t), \ldots, \phi_n(t) \) of (2.5), which are taken to be linearly independent, may be used in the construction of a fundamental matrix solution \( \Phi(t) \) satisfying

\[
\frac{d}{dt} \Phi(t) = A(t) \Phi(t) \quad (2.13)
\]

The matrix \( \Phi(t) \) is defined as

\[
\Phi(t) = \begin{bmatrix}
\phi_{11} & \phi_{21} & \cdots & \phi_{n1} \\
\phi_{12} & \phi_{22} & \cdots & \phi_{n2} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{1n} & \phi_{2n} & \cdots & \phi_{nn}
\end{bmatrix}
\quad (2.14)
\]
where
\[
\phi_{ij} = \frac{d}{dt} \phi_{ij-1} + a_{n-j+1} \phi_{i1} \tag{2.15}
\]
and
\[
\phi_k = \phi_k(t) \quad \{k = 1, 2, \ldots, n\}
\]
\[
= \alpha_k(z(t), t)
\]

It is noted that the chain rule is to be used in the differentiation of \(\phi_k(t)\).

\[
\frac{d}{dt} \phi_k(t) = \frac{d}{dt} \alpha_k(z(t), t)
\]
\[
= \frac{\partial \alpha_k}{\partial t} + \frac{\partial \alpha_k}{\partial z} \frac{dz}{dt}
\tag{2.16}
\]

The nonhomogeneous differential equation of (2.9) with zero initial conditions has the unique solution
\[
\phi(t) = \int_0^t \phi(t') \phi^{-1}(s) B(s) y(s) ds
\tag{2.17}
\]
where \(\phi^{-1}(s)\) is the matrix inverse of \(\phi(t)\).

A covariance matrix for the vector \(\phi(t)\) can be written as
\[
R(t', t) = \mathbb{E} \left[ \phi(t_1) \phi^T(t_2) \right]
\tag{2.18}
\]
\[
= \int_0^t \phi(t') \phi^{-1}(s) B(s) B^T(s) \left[ \phi^{-1}(s) \right]^T \phi^T(t) ds
\]
where superscript \(T\) denotes matrix transpose.

The coefficients \(a_k, b_k\) of (2.8) will now be determined. Associated with the differential equation of (2.7) is the homogeneous differential equation
\[ L_t(x) = 0 \quad (2.19) \]

The linear differential operator \( L_t \) can be specified in terms of the \( \phi_k \) of (2.5) by the relation

\[ L_t x = W(x, \phi_1, \phi_2, \ldots, \phi_n) = 0 \quad (2.20) \]

where the Wronskian \( W \) is given by

\[
W(x, \phi_1, \phi_2, \ldots, \phi_n) = \begin{vmatrix}
  x & \phi_1 & \cdots & \phi_n \\
  x(1) & \phi_1(1) & \cdots & \phi_n(1) \\
  \vdots & \cdots & \cdots & \cdots \\
  x(n) & \phi_1(n) & \cdots & \phi_n(n)
\end{vmatrix} \quad (2.21)
\]

The coefficients \( p_k \) can be obtained using equation (2.20). The elements \( a_k \) that appear in (2.8) and in the matrix \( A \) of (2.9) can be obtained directly by using (2.10).

The elements \( b_k \) of (2.8) and (2.9) now must be determined. Once the \( a_k \) are determined by the procedure described above, the matrix \( \Phi(t) \) can be written using equation (2.15). The matrix covariance expression of (2.18) can be written in the form

\[ R(t', t) = \Phi(t') D(t) \Phi^T(t) \quad (2.22) \]

where the elements \( d_{ij}(t) \) of the matrix \( D(t) \) may be expressed in terms of the \( \phi_i(t) \) and \( \gamma_i(t) \) of (2.5) in the form

\[ d_{ij}(t) = \frac{\gamma_i(t)}{\phi_i(t)} \quad i = j \]

\[ = 0 \quad i \neq j \quad (2.23) \]

The determination of the elements of the matrices \( \Phi(t) \) and \( D(t) \) allows the determination of the elements of \( R(t', t) \) by use of (2.22).
A matrix $R^*(t', t)$ is now defined as

$$ R^*(t', t) = R(t', t) $$

$$ = \Phi(t) D(t') \Phi^T(t') $$

Let $\Delta(t', t)$ be defined as

$$ \Delta(t', t) = R(t', t) - R^*(t', t) $$

Finally, let $\delta_{ii}$ be the diagonal elements of the matrix

$$ \Delta(t', t) \bigg|_{t' = t} $$

It can be shown that

$$ b_{n-1} = \sqrt{-\delta_{ii}} \quad i = 1, 2, \ldots, n $$

The matrix $B(t)$ is now specified, thus completing the synthesis procedure.

The mechanization system that generates the random process $x(t)$ is characterized by the set of differential equations in (2.8). A block diagram of the mechanization system is shown in Figure 2.1.

It is important to achieve the proper integrator initial conditions at the beginning of a computation cycle ($t = 0$) if the output of the mechanization system is to realize the correct covariance function. Different initial conditions applied to the same mechanization system with the same position-function excitation may give rise to widely divergent covariance functions. In general, the initial conditions may be altered by manipulation of the white-noise input and the position-parameter $z(t)$ input during resetting of the analog computer just prior to $t = 0$.

2-3. An Example

A numerical example is presented in this section to help clarify the covariance-expansion synthesis procedure.

The covariance function for the random process $g(z, t)$ is assumed to be
Figure 2.1. Mechanization System Derived by the Covariance-Expansion Synthesis Procedure.
The coefficients in (2.2) are seen from the above to be

\[ \eta_1(z', t') = c e^{-\alpha z' - \beta t'} \]

(2.29)

In a similar manner, the coefficients in (2.5) are seen to be

\[ \phi_1(t') = c e^{-\alpha z(t') - \beta t'} \]

(2.30)

\[ \gamma_1(t) = c e^{\beta t} \]

The \( p_k \) coefficients of (2.6) are determined by use of the relation in (2.20):

\[ L_t(x) = \begin{vmatrix} x & \phi_1(t) \\ x^{(1)} & \phi_1^{(1)}(t) \end{vmatrix} = 0 \]

(2.31)

Expanding this determinant with the use of (2.30), there results

\[ x^{(1)} + \left( \beta + \alpha z(t) \right) x = 0 \]

(2.32)

A comparison of (2.32) and (2.6) shows that

\[ p_0(t) = \beta + \alpha z^{(1)}(t) \]

(2.33)

Utilization of (2.10) provides the single coefficient of the \( A(t) \) matrix of (2.9):
\[ a_0(t) = p_0(t) \]
\[ = \beta + \alpha z(t) \]  
(2.34)

The single coefficient of the $A(t)$ matrix of (2.14) is found by use of (2.15) and (2.30):
\[ \phi_{11} = \phi_1(t) = c e \]  
(2.35)

The single coefficient of the $D(t)$ matrix of (2.22) is found by use of (2.23):
\[ d_{11} = \frac{\gamma_1(t)}{\phi_1(t)} = e \]  
(2.36)

The single element matrix $R(t', t)$ of (2.22) is found by use of (2.35) and (2.36):
\[ R(t', t) = \Phi(t') D(t) \Phi(t)^T \]
\[ = \begin{bmatrix} c^2 e & -\alpha z(t') & -\beta t' & -\beta t \\ e & e & e & e \end{bmatrix} \]  
(2.37)

The matrix $R^*(t', t)$ of (2.24) is
\[ R^*(t', t) = R(t', t)^T \]
\[ = \begin{bmatrix} c^2 e & -\alpha z(t) & -\beta t & -\beta t' \\ e & e & e & e \end{bmatrix} \]  
(2.38)

The difference matrix of (2.25) is
\[ \Delta(t', t) = R(t', t) - R^*(t', t) \]
\[ = \begin{bmatrix} c^2 e & -\alpha z(t') & -\beta t & -\beta t' \\ e & e & e & e \end{bmatrix} \]  
(2.39)
The matrix of (2.26) is found by use of (2.39):

$$\frac{\partial}{\partial t} \Delta(t', t) \bigg|_{t'=t} = [\delta_{11}] \quad (2.40)$$

$$= \begin{bmatrix}
 - c^2 e & - \alpha z(t) \\
 2 \beta + \alpha z(t)
\end{bmatrix}^{(1)}$$

Finally, the single coefficient in the \( B(t) \) matrix of (2.9) is found by use of (2.27) and (2.40):

$$b_0 = \sqrt{-\delta_{11}}$$

$$= c e^{-\alpha z(t)/2} \sqrt{2 \beta + \alpha z(t)}$$

Specification of this coefficient completes the synthesis procedure.

The differential equation characterizing the mechanization system for the realization of the random process \( x(t) = g(z(t), t) \) is obtained by substitution of the coefficients of (2.34) and (2.41) in the system of (2.8):

$$\frac{dx(t)}{dt} = - a_0(t) x(t) + b_0(t) y(t) \quad (2.42)$$

$$= - \left( \beta + \alpha \frac{dz(t)}{dt} \right) x(t) + c e^{-\alpha z(t)/2} \sqrt{2 \beta + \alpha \frac{dz(t)}{dt}} y(t)$$

A block diagram of the mechanization system that implements the differential equation (2.42) is shown in Figure 2.2.
Figure 2.2. Mechanization System for the Realization of the Random Process of the Example in Section 2-3.
III. THE SPECTRAL-DENSITY SYNTHESIS PROCEDURE

3-1. Introduction

This chapter is devoted to a presentation of the spectral-density synthesis procedure. The output of the analog computer network derived by this procedure approximates a random process \( g(z, t) \) depending on position \( z \) and time \( t \). Specifically, the output of the network approximates the composite random process \( x(t) = g(z(t), t) \). The process \( x(t) \) represents the effect of the original time/position-parameter random process on a sensing element having the position \( z \) at the instant \( t \). The representation is exact for each fixed position \( z \) provided the original random process is stationary in time for that position and has power spectral density in the form of equation (3.1) below. Otherwise, the output of the analog computer network is an approximation to the original process. The approximation is best when \( z \) is a slowly varying function of time.

3-2. The Synthesis Procedure

The time/position-parameter random process to be approximated is denoted as \( g(z, t) \). Here, \( z \) is a variable denoting the instantaneous position at time \( t \) of a sensing element that is affected by the random process.

It is assumed that the expected value of \( g(z, t) \) is identically zero. Processes with nonzero mean may be generated by the addition of \( g(z, t) \) to a deterministic function.

At any fixed position \( z \), it is assumed that the random process \( g(z, t) \) is wide-sense stationary in the time parameter \( t \). The power spectral density at the position \( z \) is denoted as \( S_z(w) \). It is assumed that this power spectral density may be expressed as a ratio of polynomials in \( w \).

\[
S_z(w) = \frac{c_{2n-2}(z) w^{2n-2} + \cdots + c_2(z) w^2 + c_0(z)}{d_{2n}(z) w^{2n} + \cdots + d_2(z) w^2 + d_0(z)}
\]  
(3.1)

The \( c \) and \( d \) coefficients of \( w^k \) are functions of position \( z \). If equation (3.1) does not apply exactly, the power spectral density must be approximated by an expression of this form.

The power spectral density of (3.1) may be expressed as follows (see
Here, \( H_z(s) \) is the transfer function of a time-invariant linear filter. Physically, for any fixed position \( z \) the application of a white-noise random process to the input of a filter having transfer function \( H_z(s) \) produces at the output a random process \( x(t) \) having power spectral density \( S_z(w) \).

The transfer function \( H_z(s) \) may be expressed as a ratio of two polynomials in complex frequency \( s \).

\[
H_z(s) = \frac{b_n(z)s^{n-1} + \cdots + b_1(z)s + b_0(z)}{s^n + \cdots + a_1(z)s + a_0(z)}
\]  

(3.3)

The \( a_k \) and \( b_k \) coefficients of \( s^k \) are functions of position \( z \). A technique for determining these coefficients if \( S_z(w) \) is known, is given on page 233 of Reference 1.

The transfer function \( H_z(s) \) of (3.3) may be realized in a variety of ways by the use of analog computer components (see References 4 and 5).

In particular, this transfer function may be realized for any fixed \( z \) by a mechanization of the differential equation

\[
\frac{d^n}{dt^n}x + \cdots + a_1(z) \frac{dx}{dt} + a_0(z) x = \frac{b_n(z)}{dt^{n-1}}y + \cdots + b_1(z) \frac{dy}{dt} + b_0(z) y
\]

Here, \( y \) represents the white-noise random process that is applied as input to the mechanization system. The output of the mechanization system is the random process \( x(t) = g(z, t) \) for the particular fixed value of position \( z \) previously selected.

A method of mechanizing the transfer function of (3.3) for fixed position \( z \) that avoids differentiation of the white-noise input \( y \) may be obtained.
by converting the differential equation of (3.4) to a set of first-order differential equations equivalent to (3.4) for any fixed value of z (see Reference 8, page 8):

\[ x_1 = x \quad (3.5) \]

\[ \frac{dx_1}{dt} = x_2 - a_{n-1}(z) x_1 + b_{n-1}(z) y \]

\[ \frac{dx_2}{dt} = x_3 - a_{n-2}(z) x_1 + b_{n-2}(z) y \]

\[ \ldots \]

\[ \frac{dx_{n-1}}{dt} = x_n - a_1(z) x_1 + b_1(z) y \]

\[ \frac{dx_n}{dt} = - a_0(z) x_1 + b_0(z) y \]

As was discussed previously, a mechanization of the set of equations in (3.5) provides an exact realization of the random process \( x(t) = g(z, t) \) for any fixed value of position \( z \). It might therefore be expected that the mechanization of this set of equations for variable \( z \) provides an approximate realization of the composite random process \( x(t) = g(z(t), t) \) when the parameter \( z \) is allowed to become a slowly-varying function of time \( z(t) \). An analog computer network that mechanizes the set of equations in (3.5) when the position parameter is described by a function of time \( z(t) \) is shown in Figure 3.1.

It is again important to achieve the proper integrator initial conditions at the beginning of a computation cycle \( t = 0 \) if the output of the mechanization system is to realize the correct random process. This is relatively straightforward to accomplish with this mechanization system. The white-noise input \( y(t) \) is applied to the network at all times. During the reset operation of the analog computer just prior to \( t = 0 \), the position-parameter input \( z(t) \) is maintained constant at its desired initial value long enough for the output random process \( x(t) \) to become stationary in time.
Figure 3.1. Mechanization System Derived by the Spectral-Density Synthesis Procedure.
3-3. An Example

A numerical example is presented in this section to help clarify the spectral-density synthesis procedure.

It is assumed that the random process \( g(z, t) \) is wide-sense stationary in the time parameter \( t \) at each fixed position \( z \). It is assumed that the power spectral density of the equivalent time-parameter process \( x(t) \) for each fixed \( z \) is given by the expression

\[
S_z(\omega) = \frac{28c^2 e^{-\alpha z}}{\omega^2 + \beta^2} \quad (3.6)
\]

It is noted that not enough information is contained in this characterization to specify the covariance function for the random process \( g(z, t) \) uniquely. One possible covariance function consistent with (3.6) is

\[
r_g(z', t', z, t) = c^2 e^{-\alpha z'} - \beta (t' - t) \quad (3.7)
\]

This is identical to the covariance function of (2.28) used in the numerical example of Section 3-2. Another of the infinite number of covariance functions consistent with (3.6) is

\[
r_g(z', t', z, t) = c^2 e^{-2\alpha z'} + \alpha z - \beta (t' - t) \quad (3.8)
\]

By use of (3.2) and (3.6) the transfer function \( H_z(s) \) of (3.3) is determined to be

\[
H_z(s) = \frac{c \sqrt{2\beta} e^{-\alpha z/2}}{s + \beta} \quad (3.9)
\]

Finally, a comparison of the expression in (3.6) termwise with the expression in (3.3) shows that
Specification of these coefficients completes the synthesis procedure. The differential equation characterizing the mechanization system for the example of this section is obtained by substitution of the coefficients of (3.10) in the system of (3.5)

\[
\frac{dx(t)}{dt} = -\beta x(t) + c \sqrt{2\theta} e^{-\alpha z(t)/2} y(t)
\]  

(3.11)

A mechanization system for generation of the random process \( x(t) \) is shown in Figure 3.2. This system provides an exact realization of the random process characterized by the power spectral density of (3.6) for every fixed value of the position parameter \( z \). The system is expected to provide a good approximation to the original random process for slowly varying functions of position \( z(t) \).

As has been indicated previously, specification of the power spectral density function is not sufficient to characterize the original time/position-parameter random process uniquely. Thus, further statistical information must be known in order to determine constraints that must be present on the position variable \( z(t) \) to provide good approximation. If it is assumed that the covariance function to be approximated is given by (3.7), then a comparison of the coefficients of the differential equations in (2.42) and in (3.11) shows that the following constraint must be imposed on \( z(t) \) for good approximation:

\[
\left| \frac{dz(t)}{dt} \right| \ll \frac{\beta}{\alpha}
\]  

(3.12)
Figure 3.2. Mechanization System for the Realization of the Random Process of the Example in Section 3-3.
The solution to the differential equation of (2.42) provides an exact realization of the given covariance function. If the constraint of (3.12) on $z(t)$ is imposed, then the coefficient (and thus the solution) of the equation of (3.11) characterizing the approximation are closely equal to the coefficients (and hence the solution) of the exact equation of (2.42).
IV. CONCLUSIONS

In this technical note two methods have been presented for the synthesis of an analog computer network whose output approximates a random process depending on both time and a position parameter. The covariance-expansion synthesis method is the more general of the two methods. It involves no restriction on the rate of change of the position parameter \( z(t) \) and no assumption of stationarity in time for a fixed position, whereas both of these restrictions are present with the spectral-density synthesis procedure.

On the other hand, the covariance-expansion method is mathematically more complex than the spectral-density method. It requires a larger number of computer elements for implementation than the spectral-density method. Further, it requires a more extensive specification of the statistical properties of the random process being simulated than does the spectral-density method. Unlike the spectral-density method, the covariance-expansion procedure requires the generation of derivatives of various orders for the position variable \( z(t) \). These derivatives may or may not be readily available as a portion of the analog computer solution of the overall simulation problem being studied.

A Gaussian white-noise random process is one of the inputs for the network derived by either synthesis procedure. In neither case is a differentiation of the white-noise waveform required.

The great advantage of the synthesis techniques presented in this technical note over those previously investigated is that only a single network need be synthesized for use with a wide range of vehicle trajectories. This is a factor of importance because of the considerable mathematical and physical complexity encountered in synthesizing a network to generate a prescribed nonstationary random process. In some simulation problems, such as a trajectory optimization problem, the trajectory varies from one computer run to another. In a problem of this type it would ordinarily not be feasible to construct a different random process generator for each different trajectory.

In general, it is felt that the covariance-expansion synthesis procedure is to be preferred over the spectral-density synthesis procedure. The resulting network can always be simplified by eliminating appropriate terms in the differential equations characterizing the network. Use of the spectral-density method seems to be indicated primarily if wind data is presented in terms of
the power spectral density function at a succession of altitudes.

The two examples presented in this technical note were selected to illustrate the basic procedures involved in each synthesis method. These examples are not intended to be of practical importance. In the near future, experimental wind data will be utilized in an application of one of the procedures to the synthesis of an analog computer network for the simulation of random wind disturbances.


TECHNICAL NOTE NO. 12

Research Project A-588

APPROXIMATIONS FOR HYBRID COMPUTER
ERROR DUE TO SAMPLING INTERVAL
AND EXECUTION TIME

By Joseph L. Hammond, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NASA-2473
(Development of New Methods and
Applications of Analog Computation)

30 August 1966

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
APPROXIMATIONS FOR HYBRID COMPUTER
ERROR DUE TO SAMPLING INTERVAL
AND EXECUTION TIME

By
Joseph L. Hammond, Jr.

TECHNICAL NOTE NO. 12
on
Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

For
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
ABSTRACT

This report develops equations for the error caused by the nonzero sampling interval and execution time of the digital section of a hybrid computer in the solution of certain types of problems. The work applies to the solution of equations of the form \( \dot{x} = f(x,y;t) \), where \( x \) is a dependent vector, \( y \) is a known vector of inputs, \( t \) is time, and \( f \) is a vector function of the indicated arguments. Integration is performed by the analog equipment, and the components of \( f \) are generated digitally.

Three variations of the error equation appropriate to different applications are derived. The three types of equations are as follows:

1. An equation for error with sampling rate and execution time as parameters, and with the hybrid solutions to the problem equation as input variables;
2. An equation for error with sampling rate and execution time as parameters, and with the true solutions to the problem equation as input variables;
3. An equation for error as a function of time and equivalent sampling rate, both as independent variables. The three equations are shown to be valid with minor modifications for the case that some, but not all, of the components of \( f \) are generated digitally and the others are generated by analog methods.

The report suggests the possibility that the error equations can be implemented on the hybrid computer as a means of compensating against errors caused by sampling and execution time. Advantages of such a procedure over more conventional ones are cited.
## List of Figures

2.1 - A typical part of the hybrid program for solving equation (1.1).

2.2 - Typical $f_i$, $F_i$, and $e_i$ as functions of time.
The work reported herein is a part of a general study of error analysis for hybrid computation being carried out as task 2 of the current contract program. Specifically, this technical note documents the results of an extension of the study reported in previous Technical Note No. 8. The latter gives a general equation for hybrid computer error and develops a specific equation for errors due to nonzero sampling interval as a special case of the general equation. The present report covers errors due to execution time as well as nonzero sampling interval, and investigates in detail a number of techniques which may be useful in the analysis and reduction of such errors.

The material presented includes:

(a) an extension of the basic equation to cover execution time of the digital equipment;

(b) several representations for the forcing function in the computational error equation;

(c) a method for approximating hybrid errors of the class being considered without the necessity of having available the hybrid solution of the problem equation;

(d) a technique for determining an approximate solution for error as a function of sampling interval;

(e) an extension of the basic equation to cover the case of partial digital feedback, as opposed to digital feedback around every integrator; and,

(f) a proposal for using the error equation to compensate for errors due to execution time and nonzero sampling interval.

The work applies to errors in solving a problem equation of the general form

\[ \dot{x} = f(x,y;t) , \quad x(0) = B \]  

by means of a hybrid computer programmed to use analog integrators and to generate the \( f(x,y;t) \) with digital equipment. In this equation, \( x \) is a

vector of $n$ problem variables, $\mathbf{y}$ is an $m$ vector of forcing functions, $t$ is time and $f$ is a vector of functions of the indicated arguments. A general error equation which applies for solving (1.1) in the manner specified has been previously derived as equation (2.12) in Technical Note No. 8 and may be written as follows:

$$\dot{\mathbf{y}} = A(\mathbf{x}^*;t)\mathbf{y} + f(\mathbf{x}^*,\mathbf{y};t) - F(\mathbf{x}^*,\mathbf{y};t), \quad y(0) = 0 . \quad (1.2)$$

The indicated error vector $\mathbf{y}$ is given by

$$\mathbf{y}(t) = \mathbf{x}(t) - \mathbf{x}^*(t) , \quad (1.3)$$

and the other symbols have the following meanings:

$x(t)$ -- ideal solution vector of (1.1), with components $x_j$;
$x^*(t)$ -- vector of hybrid computer solutions of (1.1), with components $x^*_j$;
$f(\ )$ -- vector of ideal functions in (1.1), with components $f_i$;
$F(\ )$ -- vector of functions as generated in the hybrid solution of (1.1), with components $F_i$;
$A(\mathbf{x}^*;t)$ -- matrix of $a_{ij}(t)$, where

$$a_{ij}(t) = \frac{\partial f_i(\mathbf{x}^*,\mathbf{y};t)}{\partial x^*_j} . \quad (1.4)$$

In this report, the basic equation (1.2) will be examined in detail and several special cases and computational variations will be developed.
Section 2
ERROR EQUATIONS BASED ON HYBRID SOLUTION VARIABLES

The difference expression \( [f(x^*,y^*;t) - F(x^*,y^*;t)] \) in the right-hand side of (1,2) represents a vector of forcing functions for the error equation. This expression accounts for errors in generating the functions \( f \) using A/D (analog-to-digital) converters, a digital computer, and a D/A converter as shown in Figure 2.1. The forcing functions involved are examined in detail in Technical Note No. 8, where an approximation is obtained for the case that the digital computer is ideal and the D/A converter contains a zero-order hold element.

Minor modifications of the discussion contained in Section 3 of Technical Note No. 8 make it possible to approximate the effect of execution time, \( e \), for the digital computer. Consider the curve of Figure 2.2 which shows a part of the time variation of the \( i \)th component of \( f \) and its approximation as \( F_i(x^*,y^*;t) \). The function \( F_i(x^*,y^*;t) \) is assumed to be generated as follows:

1. From \( t = 0 \) to \( t = \delta + e \), \( F_i = f_i(x^*,y^*;0) \).
2. The variables \( x^*(t) \) are sampled at intervals of \( \delta \) in time, to produce \( x^*(k\delta), \ k = 0,1,2,\ldots \), as inputs to the digital computer.
3. The digital computer generates \( F_i[x^*(k\delta),y(k\delta);k\delta] \) which appears at the input to the D/A converter at time \( k\delta + e \).
4. The D/A converter holds the value \( F_i[x^*(k\delta),y(k\delta);k\delta] \) from the time it is received at \( k\delta + e \) until a new value is received at time \( (k+1)\delta + e \).

The quantity \( f_i(x^*,y^*;t) - F_i(x^*,y^*;t) \), denoted as \( \epsilon_i(t) \), can be expressed as

\[
\epsilon_i(t) = f_i(x^*,y^*;t) - F_i[x^*(k\delta),y(k\delta);k\delta], \ \delta + e < t < (k+1)\delta + e. \quad (2.1)
\]

A typical \( \epsilon_i(t) \) is plotted versus time in Figure 2.2.

For small \( \delta \), \( f_i(x^*,y^*;t) \) can be approximated as

\[
f_i(x^*,y^*;t) \approx f_i[x^*(k\delta),y(k\delta);k\delta] + (t-k\delta)[df_i[x^*(t),y(t);t]/dt]_{t=k\delta}
\]

so that \( \epsilon_i(t) \) is given to good accuracy by
Figure 2.1 - A typical part of the hybrid program for solving equation (1.1).

Figure 2.2 - Typical $f_i$, $F_i$, and $e_i$ as functions of time.
\[ e_i(t) \approx (t-k\delta) \left\{ \frac{df_i[x^*(t),y(t);t]}{dt} \right\}_{t=k\delta} \]  

(2.2)

for \( k\delta+\epsilon \leq t < (k+1)\delta+\epsilon \), \( k = 0,1,2,\ldots \). It will be assumed that (2.2) with \( k = 0 \) also holds for \( t \) in the interval \((0,\epsilon)\) because digital execution time has no effect at \( t = 0 \) and \( F_i(x^*,y;0) = f_i(x^*,y;0) \).

The functions \( f \) are specified as explicit functions of \( x^*,y \), and \( t \) so that the derivatives of the \( f_i \) with respect to \( t \) can be computed using the chain rule for derivatives. The general result is given by

\[ \frac{df_i}{dx} \cdot \frac{dx}{dt} + \frac{df_i}{dy} \cdot \frac{dy}{dt} + \frac{df_i}{dt} \]  

(2.3)

Reference to (1.4) shows that the partial derivatives in the first sum of (2.3) are the \( a_{ij}(k\delta) \). The \( dx_j/dt \) can be identified as the \( F_j(x^*,y;t) \) in the actual machine equation for (1.1), namely

\[ x = F(x^*,y;t) \]  

(2.4)

The second term on the right-hand side of (2.3) involves the explicit dependence of the \( f_i \) on \( t \) with \( x^* \) and \( y \) held constant; it can be determined analytically and the result will not introduce new functions of the dependent variables. The third, and last, term on the right-hand side of (2.3) can be determined analytically provided the \( y_j(t) \) are given as explicit analytical functions of time, as will usually be the case.

Using the facts noted above in (2.3) and then substituting (2.3) in (2.2) yields an expression for the \( e_i(t) \). Using this expression and the definition of the \( e_i(t) \) in (1.2) then yields the following for the error equation:

\[ \dot{y}(t) = A(x^*;t) y(t) + (t-k\delta) A(x^*;k\delta) F(x^*,y;k\delta) + (t-k\delta) \left\{ \frac{df}{dt} + \sum_{j=1}^{m} \frac{df}{dy_j} \frac{dy_j}{dt} \right\}_{t=k\delta} \]  

(2.5a)

for \( k\delta+\epsilon \leq t < (k+1)\delta+\epsilon \); \( k = 0,1,2,\ldots \). For \( t \) on \((0,\epsilon)\) the equation is
\[ \dot{\gamma}(t) = A(\bar{x}^*; t) \gamma(t) + tA(\bar{x}^*; 0) f(\bar{x}^*, \gamma; 0) + t \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \frac{dy_j}{dt} \right\} \bigg|_{t=0}; \]

\[ \gamma(0) = 0. \quad (2.5b) \]

Note that most of the terms on the right-hand side of (2.5) depend on the approximate solution vector, \( \bar{x}^*(t) \), generated by the hybrid program for (1.1). Note further that several variables are digital in nature and have a constant value on the time intervals \( k+\varepsilon \leq t < (k+1)\delta + \varepsilon \), \( k = 0,1,2, \ldots \).

If (2.5) is solved at the same time as (1.1), the digital variables \( F(\bar{x}^*, \gamma; k\delta) \) are available from the program for (1.1) with the proper execution times. The terms in braces \( \{ \} \) in (2.5) must be generated digitally with the sampling interval \( \delta \), and delayed for the execution time \( \varepsilon \).

Assuming the availability of the input terms as noted above, (2.5) can be solved by either all-digital, hybrid or analog methods. An analog or all-digital method would not introduce errors due to execution time and nonzero sampling interval, while a hybrid solution would introduce such errors. In solving (2.5) digitally or by the hybrid method used in solving (1.1), the quantity \( t-k\delta \) must be approximated. If a digital step size on the order of \( \delta \) is used, the average of \( t-k\delta \) over one sampling interval, namely \( (\delta+2\varepsilon)/2 \), gives accuracy consistent with previous approximations. Often it will be convenient to use this same approximation in other work, and it is interesting to note that execution time and \( \delta \) are equivalent in this term, with execution time having twice the effect of the sampling interval.

Equation (2.5) is convenient computationally if the \( f_i \) do not depend explicitly on \( t \) or the \( y_j \), since in such a case the terms in braces are zero. If the \( f_i \) do depend explicitly on these variables, a more tractable version of the error equation can be derived, based on somewhat more stringent assumptions. To obtain this variation in the error equation, let \( t-k\delta \) in (2.1) be approximated by \( (\delta+2\varepsilon)/2 \) and assume that the quantity in braces is a continuous function of time, rather than a variable taking on discrete values. Then for all \( t \geq \delta+\varepsilon \),

\[ e_i(t) = \frac{\delta + 2\varepsilon}{2} \frac{df_i(\bar{x}^*, \gamma; t-\varepsilon)}{dt}, \quad (2.6) \]
where the continuous argument $t-e$ has been used to replace the $k\delta$ of (2.1).

Introducing the vector form of (2.6) into (1.2) and rearranging terms gives

\[ \frac{d}{dt} \left[ y(t) - \frac{\delta + 2e}{2} f(x^*,\tilde{y};t-e) \right] = A(x^*;t) y, \; t \geq \delta+e \]  

(2.7)

Definition of a variable $z(t)$ as

\[ z(t) = y(t) - \frac{\delta + 2e}{2} f(x^*,\tilde{y};t-e), \; t \geq \delta+e \]  

(2.8)

then reduces (2.7) to

\[ \dot{z}(t) = A(x^*;t) \left[ z(t) + \frac{\delta + 2e}{2} f(x^*,\tilde{y};t-e) \right], \; t \geq \delta+e \]  

(2.9)

Note that the $f_i(x^*,\tilde{y};t-e)$ are adequately approximated by the $F_i(x^*,\tilde{y};k\delta)$ from the hybrid program for (1.1); also, as in (2.5), $A(x^*;t)$ is computed using the approximate solution vector $x^*(t)$.

Strictly speaking, the time interval $(0,\delta+e)$ requires special consideration. However, in view of the approximations used elsewhere it does not seem reasonable to complicate the computer program to account for the first interval. Thus (2.6) and hence (2.9) will be assumed to hold for all time with the digitally generated functions used for the $f(x^*,\tilde{y};t-e)$. A choice of $z(0)$ given by

\[ z(0) = -\frac{\delta + 2e}{2} f(x^*,\tilde{y};0) \]  

(2.10)

along with the requirement that

\[ z(t) = y(t) - \frac{\delta + 2e}{2} f(x^*,\tilde{y};0), \; 0 \leq t < \delta+e \]  

(2.11)

seems to be reasonable.

If the $f_i$ depend explicitly on time and the forcing functions $\gamma_j$, solving (2.9) and then (2.8) for the error functions would seem more tractable than solving (2.5). These equations, however, involve different, and possibly more stringent assumptions.
Note that for $e = 0$, equation (2.9) above becomes identical to equation (5.5) of Technical Note No. 8, which was derived using slightly less detailed considerations.
Section 3
ERROR EQUATIONS BASED ON TRUE SOLUTIONS

The error equations of Technical Note No. 8, as well as (2.5) and (2.9) above, must be solved on a hybrid computer so that \( \dot{x}^*(t) \) and \( F(x^*, y; t) \) will be available. This is not always convenient, and in this section we will derive error equations based on true solutions of (1.1)--"true" in the sense that there are no errors due to execution time or nonzero sampling interval.

The steps involved in obtaining equations (2.6) through (2.12) of Technical Note No. 8 can be retraced retaining the argument \( x \) instead of \( x^* \) in the terms that become \( A(t) y \) in equation (2.12) of the referenced report. Carrying through this calculation shows that to a first-order approximation--i.e., taking only the first two terms of the series given in equation (2.9)--equation (2.12) is unchanged by using \( x \) rather than \( x^* \) to compute the \( A(t) \). For present purposes the modified error equation can then be written as

\[
\dot{y}(t) = A(x, t) y(t) + f(x^*, y; t) - F(x^*, y; t) , \quad y(0) = 0 .
\]  (3.1)

The term \( f(x^*, y; t) - F(x^*, y; t) = \epsilon(t) \) has been approximated as given by (2.2) above. Following the method used in deriving (2.6), a further approximation can be made by replacing \( \frac{df_i}{dt} \bigg|_{t=k\delta} \) in (2.2) by \( \frac{df_i}{dt} \bigg|_{t=(t-e)} \) to obtain

\[
\epsilon_i(t) \approx (t-k\delta) \frac{df_i}{dt} \bigg|_{t=(t-e)} .
\]  (3.2)

Now consider \( \dot{f}_i(x-y, y; t-e) \) in (3.2), where \( x^* \) has been replaced by \( x-y \). This term can be expanded by a two-dimensional Taylor series in the arguments \( x-y \) and \( t-e \) to obtain

\[
\dot{f}_i(x-y, y; t-e) = \dot{f}_i(x, y; t) - \sum_{j=1}^{n} y_j \frac{\partial \dot{f}_i(x, y; t)}{\partial x_j} \]

\[
- e \frac{\partial \dot{f}_i(x, y; t)}{dt} + \text{[higher order terms]}. \]  (3.3)

The first term on the right in (3.3) can be expressed as
\[
\frac{df_i}{dt} = \sum_{j=1}^{n} a_{i,j} f_j(t) + \sum_{j=1}^{m} \frac{\partial f_i}{\partial y_j} \cdot \dot{y}_j + \frac{\partial f_i}{\partial t} \quad (3.4)
\]

by the same reasoning used in obtaining (2.3); and, if the \( y_j \) are given as explicit functions of time, (3.4) can be evaluated. To simplify (3.3), it is reasonable to define a matrix \( B(x; t) \) with elements \( b_{ij} \) given by

\[
b_{ij}(t) = \frac{\partial f_i}{\partial x_j} \quad (3.5)
\]
in analogy to the definition of \( A(x; t) \).

The error equation (3.1) can now be reduced to a tractable form using (3.2) through (3.5). The result is

\[
\dot{y}(t) = A(x; t) \left[ y(t) + (t-k\delta) f(x, y; t) \right] + (t-k\delta) \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \cdot \dot{y}_j - B(x; t) \cdot y(0) - f(x, y; t) \right\}, \quad y(0) = 0. \quad (3.6)
\]

Equation (3.6), using the true solution vector \( x(t) \), should be compared to (2.5) which uses the approximate solution vector \( x^*(t) \). The comparison shows that (3.6) is identical in form to (2.5) with the exception that (3.6) contains the additional expression \([ - B(x; t)y(t) - e f(x, y; t) \]) which accounts for use of \( x \) rather than \( x^* \). In solving (3.6) it will usually be inconvenient to generate \( t-k\delta \) and this term can be replaced by its average value \((\delta + 2e)/2\) as noted above. Making this substitution gives as the final equation:

\[
\dot{y}(t) = A(x; t) \left[ y(t) + \frac{\delta + 2e}{2} f(x, y; t) \right] + \frac{\delta + 2e}{2} \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \cdot \dot{y}_j - B(x; t) \cdot y(0) - e f(x, y; t) \right\}, \quad y(0) = 0. \quad (3.7)
\]
Equations (2.5), (2.9), and (3.7) as developed above, give three alternative methods for computing error as a function of time with sampling rate and execution time as parameters. Since the effect of these parameters is of central interest, a fourth equation for error— one involving a function of sampling rate and execution time as an independent variable— will be developed in this section. To facilitate the work it is convenient to define the following composite parameters:

\[ a = \frac{2e}{\delta} \]  \hspace{1cm} (4.1)

\[ \Delta = (a+1) \delta \]  \hspace{1cm} (4.2)

With these definitions (3.7) becomes

\[
\dot{y}(t) = g(y, \Delta; t) = A(x; t) y(t) + \frac{\Delta}{2} \left\{ A(x; t) f(x, y; t) + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \dot{y}_j - B(x; t) y(t) - \frac{a\Delta}{2(a+1)} f^2 \right\}.
\]  \hspace{1cm} (4.3)

Note that \( g(y, \Delta; t) \) will be used to denote the right-hand side of (4.3).

The basic procedure to be employed is due to Murray and Miller.* It involves regarding the solution of an error equation as a function of both \( \Delta \) and \( t \), as can be indicated explicitly by the notation \( y(\Delta, t) \). The solution for any one variable, say \( y_1 \), can then be expanded in a Taylor series about \( \Delta_0 \) to give.

\[
y_1(\Delta, t) = y_1(\Delta_0, t) + (\Delta - \Delta_0) \left[ \frac{\partial y_1}{\partial \Delta} \right]_{\Delta_0} + \frac{1}{2}(\Delta - \Delta_0)^2 \left[ \frac{\partial^2 y_1}{\partial \Delta^2} \right]_{\Delta_0} + \ldots \]  \hspace{1cm} (4.4)

It is expedient to introduce the notation

\[
\begin{align*}
\frac{\partial \gamma_1}{\partial \Delta} \bigg|_{\Delta_0} &= \gamma_{11} \quad \text{and} \quad \frac{\partial^2 \gamma_1}{\partial \Delta^2} \bigg|_{\Delta_0} = \gamma_{12};
\end{align*}
\]

and, to be consistent with this notation, \( \gamma_1 \) itself will be denoted as \( \gamma_{10} \) in this section. The vectors

\[
\begin{bmatrix}
\gamma_{11} \\
\vdots \\
\gamma_{n1}
\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}
\gamma_{12} \\
\vdots \\
\gamma_{n2}
\end{bmatrix}
\]

will be denoted respectively as \( \gamma_1 \) and \( \gamma_2 \).

Although the general method could be applied to any one of the error equations, we will use (3.7)--as given in a different notation by (4.3)-- as the starting point here to illustrate the method. This choice has been made with the feeling that a result based on true, rather than hybrid, solutions can be of more direct use. In the present application \( \Delta_0 \) is chosen as zero so that (4.4) can be used to approximate \( \gamma_{10} \) as

\[
\gamma_{10}(\Delta, t) \approx \Delta \gamma_{11}(\Delta, t) + \frac{1}{2} \Delta^2 \gamma_{12}(\Delta, t). \tag{4.5}
\]

In considering this approximation note that \( \gamma_{10}(0, t) = 0 \) and that the first three terms of the series are used in obtaining (4.5).

The method of Murray and Miller involves obtaining a differential equation for \( \gamma_{11}(\Delta, t) \) and \( \gamma_{12}(\Delta, t) \). The general result for an equation in the form of (4.3) is given by

\[
\dot{\gamma}_1(t) = \sum_{j=1}^{n} \left( \frac{\partial \xi}{\partial \gamma_{j0}} \right)_{\Delta=0} \gamma_{j1} + \left( \frac{\partial \xi}{\partial \Delta} \right)_{\Delta=0} \tag{4.6}
\]
\[
\dot{\gamma}_2(t) = \sum_{j=1}^{n} \left( \frac{\partial g}{\partial \gamma_{j0}} \right) \Delta=0 \cdot \gamma_{j2} + \sum_{j=1}^{n} \sum_{k=1}^{n} \left( \frac{\partial^2 g}{\partial \gamma_{j0} \partial \gamma_{k0}} \right) \Delta=0 \cdot \gamma_{j1} \gamma_{k1}
\]

\[
+ \sum_{j=1}^{n} \left( \frac{\partial^2 g}{\partial \Delta \partial \gamma_{j0}} \right) \Delta=0 \cdot \gamma_{j1} + \left( \frac{\partial^2 g}{\partial \Delta^2} \right) \Delta=0.
\]  

(4.7)

In these equations the partial derivatives with respect to \( \Delta \) account for the explicit dependence of \( g \) or \( \frac{\partial g}{\partial \gamma_{j0}} \) on \( \Delta \). Equations (4.6) and (4.7) are derived in the Appendix and are expressed there in an alternative vector form.

Evaluating (4.6) and (4.7) for \( g(\gamma_0, \Delta; t) \) as given by (4.3) yields:

\[
\dot{\gamma}_1(t) = A(x; t) \gamma_1(t) + \frac{1}{2} \left\{ A(x; t) + (x, \gamma; t) + \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial \gamma_{j}} \gamma_{j} \right\}
\]

(4.8)

\[
\dot{\gamma}_2(t) = A(x, t) \gamma_2(t) - \frac{1}{2} B(x; t) \gamma_1(t) - \frac{a}{2(a+1)} \gamma_2(t) \left( \gamma_1(t)^2 + \frac{\gamma_2(t)^2}{(a+1)} \right),
\]

(4.9)

where \( \gamma_1(0) = 0 \) and \( \gamma_2(0) = 0 \).
In this section modifications of the error equations are made to make them applicable to the case that (1.1) is solved on a hybrid computer programmed to use analog integration and to generate some, but not all, of the \( f_i \) digitally.

Equation (1.2) will apply in the case being considered if the \( F_i(x^*, y, t) \) which are not generated digitally are set equal to \( f_i(x^*, y; t) \). Note that it takes only one digitally generated \( F_i \) to cause \( y \) to be nonzero and \( x^* \) to be different from \( x \). Note further that \( A(x^*; t) \) is defined by (1.4) in terms of the true function \( f \), and therefore does not depend on which functions are generated digitally. Thus, all of the previous work can be made to apply with proper interpretation of the \( F \) functions.

The necessary changes will now be noted for the results of each of the three previous sections.

Consider the quantity \( e_i(t) \) in (2.1). For partial digital feedback \( e_i \) will be zero in correspondence with those \( F_i \) not generated digitally. This result carries through the approximations to \( e_i \), so that in (2.5a) and (2.5b) the terms multiplied by \( t - k \delta \) or \( t \) are zero in the scalar equations for which the \( F_i \) are not generated digitally. The result can be expressed concisely by defining a vector \( S \) having components \( s_i \) such that:

\[
s_i = \begin{cases} 0, & \text{if } F_i = f_i; \\ (t - k \delta) A(x^*; k \delta) F(x^*; y; k \delta) + (t - k \delta) \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \dot{y}_j \right\}_{t=k \delta}, & \text{otherwise.} \end{cases}
\]

Using \( S \), (2.5a) becomes

\[
\dot{y}(t) = A(x^*; t) y(t) + S, \quad k+e \leq t < (k+1) \delta + e
\]

(5.1)

with a similar expression for (2.5b).

A similar expedient will generalize (2.9). In this case a function \( f^* \) can be defined by

\[
\dot{y}(t) = A(x^*; t) y(t) + f^*(t), \quad k+e \leq t < (k+1) \delta + e
\]
\[ \hat{f}_i = 0, \quad \text{if } \hat{F}_i = \hat{f}_i; \]
\[ \hat{f}_i = f_i, \quad \text{otherwise.} \]

Equations (2.8) and (2.9) then retain the same form with \( f \) replaced by \( \hat{f} \).

Examination of the steps leading to (3.6) and (3.7) indicates that these equations remain valid for partial digital feedback if the explicitly appearing \( f \) functions are replaced by \( \hat{f} \) and if \( B \) is computed using \( \hat{f} \). The same changes make (4.8) and (4.9) remain valid. Note that in all cases the net result of using partial digital feedback is to remove the forcing functions from some of the scalar equations. The complexity of the equation is thus reduced to a certain extent.
USE OF ERROR EQUATIONS TO COMPENSATE FOR DELAYS IN DIGITAL FUNCTION GENERATION

The various equations for hybrid error due to sampling rate and execution time, as derived above, give reasonable approximations to actual error. As will be shown in a subsequent technical note, the errors of approximation have in fact been quite small in the examples considered. Thus, since the error equations can be set up in a routine manner, it seems reasonable to consider the use of the error equations as a means for compensating for sampling and execution time errors. To do so would require programming the error equations on the hybrid computer in addition to the problem equations.

The fact that the error equations have a complexity on the order of that of the problem equations is to the detriment of the method. It does, however, have advantages such as the fact that a fairly precise measure of uncompensated error is readily available. Furthermore, if this method is compared to the more common use of phase-advance or prediction networks incorporated in the analog program as an alternative means for compensation*, it will be noted that the error equation method avoids the use of differentiating networks, which can be quite troublesome.

This report is concerned with equations for the error caused by non-zero sampling interval and execution time in a hybrid computer solution of a vector equation of the form

\[ \dot{x} = f(x, y; t) . \]

Integration is performed by the analog equipment and some, but not all, of the \( f \) are generated digitally, with the attendant errors caused by sampling and execution time.

Several variations of the error equation, all designed for computer implementation, are derived. A summary of these equations follows. (See Glossary below for definitions of symbols.)

(a) Error equations using hybrid solution variables

For \( k + e < t < (k+1)e + e \), \( k = 0,1,2,..., \)

\[ \dot{y}(t) = A(x^*;t) y(t) + (t-k\delta) A(x^*;k\delta) F(x^*, y; k\delta) + \]

\[ (t-k\delta) \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \dot{y}_j \right\} _{t=k\delta} . \]

For \( 0 \leq t < e, \)

\[ \dot{y}(t) = A(x^*;t) y(t) + t A(x^*;0) f(x^*, y;0) + t \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \dot{y}_j \right\} _{t=0} , \ y(0) = 0. \]

An alternative equation based on more stringent assumptions is the following:

\[ \dot{z}(t) = A(x^*;t) \left[ z(t) + \frac{\delta + 2e}{2} F(x^*, y; k\delta) \right] , \ k = 0,1,2,..., \]

where

\[ z(t) = y(t) - \frac{\delta + 2e}{2} F(x^*, y; k\delta) , \ k = 0,1,2,..., \]
and
\[ z(0) = -\frac{\delta + 2e}{2} f(x^*, y; 0). \]

(b) Error equation using true solution variables
\[ \dot{\gamma} = A(x; t) \left[ \gamma(t) + \frac{\delta + 2e}{2} f(x, y; t) \right] + \frac{\delta + 2e}{2} \left\{ \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \gamma_j - B(x; t) \gamma(t) - e \right\}, \quad \gamma(0) = 0. \]

(c) Error equation as a function of \( \Delta \)
\[ \gamma_{10}(\Delta, t) = \Delta \gamma_{11}(\Delta, t) + \frac{1}{2} \Delta^2 \gamma_{12}(\Delta, t), \]
where
\[ \dot{\gamma}_{11}(t) = A(x; t) \gamma_{11}(t) + \frac{1}{2} \left\{ A(x; t) f(x, y; t) + \frac{\partial f}{\partial t} + \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \gamma_j \right\}, \quad \gamma_{11}(0) = 0, \]
\[ \dot{\gamma}_{12}(t) = A(x; t) \gamma_{12}(t) - \frac{1}{2} B(x; t) \gamma_{11}(t) - \frac{a}{2(a + 1)} \gamma_{12}, \quad \gamma_{12}(0) = 0. \]

(d) Error equation for partial digital feedback
For any \( f \) which is not generated digitally, \( F_1 = f \). The substitution of \( \hat{f} \), whose components are defined as
\[ \hat{f}_1 = 0, \text{ if } F_1 = f, \]
\[ \hat{f}_1 = f, \text{ otherwise,} \]
for the functions \( f \) appearing explicitly and used in calculating \( B(x, t) \) makes all but the first equation in (a) above valid for partial digital feedback.
Glossary of Symbols

\( x \) - true problem state vector

\( x_j \) - an element of \( x \) (similar notation for components of other vectors)

\( \hat{x} \) - approximate problem state vector obtained from hybrid solution

\( y \) - inputs to problem equation

\( \dot{y} \) - total derivative of \( y \) with respect to time (similar notation for other variables)

\( f \) - vector of true functions

\( F \) - vector of digitally generated functions

\( t \) - time

\( \gamma \) - error vector given by \( \gamma = x - \hat{x} \)

\( \delta \) - sampling interval of digital equipment

\( e \) - execution time of digital equipment

\[
\gamma_1 = \frac{\partial \gamma}{\partial \Delta}
\]

\[
\gamma_2 = \frac{\partial \gamma_1}{\partial \Delta}
\]

\( \Delta \equiv \delta + 2e \)

\( a \equiv \frac{2e}{\delta} \)
\[ A(x^*, t) = [a_{i,j}(t)] \quad ; \quad a_{i,j}(t) = \left. \frac{\partial f_{i,j}(x, y; t)}{\partial x_j} \right|_{x=x^*} \]

\[ B(x; t) = [b_{i,j}(t)] \quad ; \quad b_{i,j}(t) = \frac{\partial f_{i,j}(x, y; t)}{\partial x_j} \]
The purpose of this appendix is to show the derivation of equations (4.6) and (4.7) which are differential equations for $y_1(t)$ and $y_2(t)$. The calculation begins from (4.3) which can be expressed as

$$\dot{y}_0 = g(y_0, \Delta; t)$$  \hspace{1cm} (A1)

where $\Delta$ and $t$ are independent variables and $y_0$ is a function of $\Delta$ and $t$.

To derive the required result with a minimum of algebraic manipulation, two vector manipulations not previously employed will be defined and used—namely:

(1) For $b$ a vector and $x$ a vector,

$$\frac{db}{dx} = \begin{bmatrix} \frac{db_1}{dx} & \frac{db_1}{dx} & \cdots & \frac{db_1}{dx} \\ \frac{db_2}{dx} & \frac{db_2}{dx} & \cdots & \frac{db_2}{dx} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{db_n}{dx} & \frac{db_n}{dx} & \cdots & \frac{db_n}{dx} \end{bmatrix}$$

(2) For $A$ and $B$ matrixes and $z$ a scalar,

$$\frac{d}{dz} AB = A \frac{dB}{dz} + \frac{dA}{dz} B$$

As the first step in the calculation, (A1) can be differentiated partially with respect to $\Delta$ to obtain

$$\frac{\partial}{\partial \Delta} \dot{y}_0 = \frac{\partial}{\partial \Delta} g(y_0, \Delta; t)$$  \hspace{1cm} (A2)

Under rather general conditions on $y_0$ the order of partial differentiation with respect to $\Delta$ and total differentiation with respect to time can be interchanged. Further, $\frac{\partial g}{\partial \Delta}$ can be expanded by the chain rule to obtain finally
\[ \dot{\gamma}_1 = \left( \frac{\partial g}{\partial \gamma_0} \right)_{\Delta=0} \gamma_1 + \left( \frac{\partial^2 g}{\partial \gamma_0 \partial \gamma_1} \right)_{\Delta=0} \left( \frac{\partial^2 g}{\partial \gamma_0^2} \right)_{\Delta=0} \left( \frac{\partial^2 g}{\partial \gamma_1^2} \right)_{\Delta=0} \]  \hfill (A3)

Note that \( \frac{\partial g}{\partial \Delta} \) accounts for the explicit dependence of \( g \) on \( \Delta \) with \( \gamma_0 \) and \( t \) held constant.

Equation (A3) is equivalent to (4.6) after the matrix multiplication in the first term on the right-hand side of (A3) has been carried out.

To obtain (4.7), (A3) is again differentiated partially with respect to \( \Delta \). This yields

\[ \frac{\partial}{\partial \Delta} \dot{\gamma}_1 = \frac{\partial}{\partial \Delta} \left( \frac{\partial g}{\partial \gamma_0} \right) \gamma_1 + \frac{\partial}{\partial \Delta} \left( \frac{\partial g}{\partial \gamma_0} \right) \left( \frac{\partial g}{\partial \gamma_1} \right) \]  \hfill (A4)

Using the property of the derivative of the product of two vectors with respect to a scalar, the interchange of differentiation with respect to \( t \) and \( \Delta \), and the definition of \( \gamma_2 \), equation (A4) reduces to

\[ \dot{\gamma}_2 = \left( \frac{\partial g}{\partial \gamma_0} \right)_{\Delta=0} \gamma_0 + \left( \frac{\partial}{\partial \Delta} \left( \frac{\partial g}{\partial \gamma_0} \right) \right) \gamma_1 + \frac{\partial^2 g}{\partial \gamma_0^2} \left( \frac{\partial^2 g}{\partial \gamma_0 \partial \gamma_1} \right)_{\Delta=0} \]  \hfill (A5)

The final equation is obtained by expanding \( \frac{\partial}{\partial \Delta} \frac{\partial g}{\partial \gamma_0} \) using the chain rule. This yields

\[ \dot{\gamma}_2 = \left( \frac{\partial g}{\partial \gamma_0} \right)_{\Delta=0} \gamma_0 + \left( \frac{\partial}{\partial \Delta} \left( \frac{\partial g}{\partial \gamma_0} \right) \right) \gamma_1 + \frac{\partial^2 g}{\partial \gamma_0^2} \left( \frac{\partial g}{\partial \gamma_0} \right)_{\Delta=0} \]  \hfill (A6)

where the partials with respect to \( \Delta \) account for the explicit dependence of \( g \) and \( \frac{\partial g}{\partial \gamma_0} \) on \( \Delta \) with \( \gamma_0 \) and \( t \) held constant. When the indicated matrix multiplications are carried out, (A6) becomes equivalent to (4.7).

It is of interest to apply (A3) and (A6) to \( g(\gamma_0; t) \) in vector form. Using (4.3) and the definitions given above \( g \) can be expressed as

- 22 -
\[ g = \frac{\partial f}{\partial x} \gamma_0 + \frac{\Delta}{2} \left( \frac{\partial f}{\partial x} f + \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} \frac{dy}{dt} - \frac{\partial f}{\partial x} \gamma_0 - \frac{a}{2(a+1)} \Delta f \right) \]. \quad (A7)

Carrying out the required operations and setting \( \Delta = 0 \) yields

\[ \dot{\gamma}_1 = \frac{\partial f}{\partial x} \gamma_1 + \frac{1}{2} \left[ \frac{\partial f}{\partial x} f + \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} \frac{dy}{dt} \right] \] \quad (A8)

\[ \dot{\gamma}_2 = \frac{\partial f}{\partial x} \gamma_2 - \frac{1}{2} \frac{\partial f}{\partial x} \gamma_1 - \frac{a}{2(a+1)} \frac{\partial f}{\partial t} \] \quad (A9)

Equations (A8) and (A9) are of course alternative forms of (4.8) and (4.9).
TECHNICAL NOTE NO. 13
Research Project A-588

IMPLEMENTATION AND EVALUATION OF
EQUATIONS FOR HYBRID COMPUTER ERROR

By Joseph L. Hammond, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

15 January 1967

School of Electrical Engineering
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia
IMPLEMENTATION AND EVALUATION OF EQUATIONS FOR HYBRID COMPUTER ERROR

by

Joseph L. Hammond, Jr.

TECHNICAL NOTE NO. 13

on

Contract No. NAS8-2473

(Development of New Methods and Applications of Analog Computation)

For

GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama
This technical note presents recent new results obtained in a study of hybrid computer error due to nonzero sampling interval and execution time. The work is a continuation of that documented earlier in Technical Notes Nos. 8 and 12 under the subject contract. The present report is concerned primarily with the following points: (1) a computer implementation of previously derived error equations, (2) verification of each form of the error equations with specific test examples, and (3) the study of typical test examples to establish the general properties of the various equations.

A basic digital computer program for implementing the error equations is described. Among other things, this program can work from a general statement of the error equations to compute results for a particular problem without the necessity of deriving a specific error equation analytically. Several different types of results can be printed out, including a table of maximum and minimum error versus sampling interval.

Data obtained from a number of numerical solutions supports the following conclusions:

(a) With one minor exception requiring further investigation, all forms of the error equation are analytically accurate.

(b) The basic approximation on which the equations are based is sound, and for reasonable conditions the error as determined from the equations is a good approximation of true error.

(c) Little additional approximation error is introduced in the form of the error equations which yields error versus sampling interval. This form of the error equations is recommended for general studies.

(d) The forms of the error equations based on true solutions and on hybrid solutions of the problem equation give comparable accuracy.

(e) Use of the error equations to compensate a hybrid system seems to have much potential. For the test problem used, such compensation reduced error by a factor of ten.
CONTENTS

Abstract ......................................................... iii
Glossary of Symbols ........................................... vi

I. INTRODUCTION .................................................. 1

II. DIGITAL COMPUTER IMPLEMENTATION ....................... 3
   2-1. Basic Computer Program ................................. 3
   2-2. Approximation of Hybrid Solutions .................... 4
   2-3. Basic Implementation of the Error Equations ........... 4
   2-4. Generalized Implementation of the Error Equations ... 7
   2-5. Implementation of the Calculation of Maximum and Minimum Error Versus Sampling Interval .......... 9

III. EVALUATION OF ERROR EQUATIONS ......................... 12
   3-1. Introduction ............................................. 12
   3-2. Error Equations Using Hybrid Solution Variables ...... 13
   3-3. Error Equations Using True Solution Variables ....... 16
   3-4. Error Equations: Versus Sampling Interval ............ 19
   3-5. Error Equations for Partial Digital Feedback ........ 24
   3-6. Error Equations Used to Compensate Hybrid Systems ... 24
   3-7. Execution Time and Problem Equation Inputs .......... 31

IV. CONCLUSIONS RELATIVE TO ERROR EQUATIONS ............ 36
   4-1. Theoretical Validity ...................................... 36
   4-2. Degree of Approximation ................................ 36
   4-3. General Recommendations ............................... 38

Appendix: DETAILED INFORMATION ON BASIC COMPUTER PROGRAM .... 40

LIST OF FIGURES

2-2A. Actual Hybrid Error (MSFC) and Digitally Simulated Hybrid Error (Assuming Execution Time e = 0) in Solution of Nonlinear Pendulum Problem Using Sampling Interval of \( \delta = 0.1 \) ........................................ 5

2-2B. Actual Hybrid Error (MSFC) and Digitally Simulated Hybrid Error (Assuming Execution Time e = 0 or e = 0.03) in Solution of Nonlinear Pendulum Problem Using Various Sampling Intervals (\( \delta \)) .................................. 6

2-4. Error in Hybrid Solution of Nonlinear Pendulum Problem as Approximated by Direct Computation from General Error Equation (2-4.1) and from Analytically Derived Specific Error Equation (3-3.2) ........ 10

3-1A. True Solutions for Typical Cases of the Nonlinear Pendulum Problem ........................................ 14

3-1B. True Solutions for Typical Cases of the Duffin's Equation Problem ............................................. 15
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-2A</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>by Equation (3-2.1) Using Hybrid Solution Variables for Nonlinear</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-2B</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>by Equation (3-2.2) Using Hybrid Solution Variables for Nonlinear</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-3A</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>by Equation (3-3.1) Using True Solution Variables for Nonlinear</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-3B</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>by Equation (3-3.2) Using True Solution Variables for Duffin's</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Equation Problem</td>
<td></td>
</tr>
<tr>
<td>3-4A</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Through a Taylor-Series Expansion by Equations (3-4.1) and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3-4.2) for the Nonlinear Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-4B</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Through a Taylor-Series Expansion by Equations (3-4.1) and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3-4.3) for the Duffin's Equation Problem</td>
<td></td>
</tr>
<tr>
<td>3-4C</td>
<td>Effect of Sampling Interval (6) on Maximum and Minimum Values of</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Hybrid Error as Computed in 10-Second Runs from Equations (3-4.1) and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3-4.2) for Nonlinear Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-5</td>
<td>Illustrating Effect of Partial Digital Feedback on True Simulated-</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Hybrid Error and Error as Approximated by Equation (3-5.1) Using</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hybrid Solution Variables for Nonlinear Pendulum Problem</td>
<td></td>
</tr>
<tr>
<td>3-6A</td>
<td>True and Simulated-Hybrid Solutions of a Control System Problem,</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>with Reduction in True Error Achieved Through Use of Approximate Error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Equation to Compensate Hybrid Solution</td>
<td></td>
</tr>
<tr>
<td>3-6B</td>
<td>Illustrating Reduction in Response Error of a Hybrid Control System</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Through Use of Approximate Error Equation to Compensate Plant Input</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>3-7A</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>by Modified Version of Equation (3-3.2) to Account for Effect of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Forcing Function in Duffin's Equation Problem</td>
<td></td>
</tr>
<tr>
<td>3-7B</td>
<td>Comparison of True Simulated-Hybrid Error and Error as Approximated</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>by Modified Version of Equation (3-3.1) to Account for Effect of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Execution Time in Nonlinear Pendulum Problem</td>
<td></td>
</tr>
</tbody>
</table>
GLOSSARY OF SYMBOLS

\(x\)  - state-variable vector representing true solution of problem equations

\(x_i\)  - an element of \(x\) (similar notation for components of other vectors)

\(\hat{x}\)  - state-variable vector representing hybrid solution of problem equations

\(y\)  - forcing-function or input vector for problem equations

\(f\)  - vector of functions identified with time derivatives of problem state variables

\(\dot{f}\)  - total derivative of \(f\) with respect to time (similar notation for other variables)

\(t\)  - time

\(\gamma\)  - error vector approximating the true error \((x - \hat{x})\) between hybrid and true solutions of a problem equation

\(\delta\)  - analog-signal sampling interval for digital elements in hybrid computer

\(e\)  - execution time for digital elements in hybrid computer

\(\Delta = \delta + 2e\)

\(\gamma_1 = \partial y / \partial \Delta\) ; \(\gamma_2 = \partial^2 y / \partial \Delta^2 = \partial \gamma_1 / \partial \Delta\)

\(A(x,t) = [a_{ij}(t)]\) ; \(a_{ij}(t) = \frac{\partial f_i(x,y,t)}{\partial x_j}\)

\(B(x,t) = [b_{ij}(t)]\) ; \(b_{ij}(t) = \frac{\partial^i f_i(x,y,t)}{\partial x_j}\)
I. INTRODUCTION

The work reported herein is a part of a general study of error analysis for hybrid computation being carried out as one of three tasks under the current contract program. Earlier work, reported in Technical Notes Nos. 8 and 12*, has covered the theoretical development of a class of equations whose solutions approximate hybrid computer error caused by nonzero sampling interval and execution time. Specifically, in Technical Note No. 12 three general forms of the error equation are developed—namely, (1) equations based on hybrid solutions of the problem equation, (2) equations based on true solutions of the problem equation, and (3) equations for direct computation of error as a function of sampling interval and execution time. In addition, the same technical note indicates how to use the error equations if some, but not all, of the computational loops contain both analog and digital operations. Technical Note No. 12 also discusses the use of the error equations to compensate for sampling interval and execution time errors in hybrid computation or in physical sampled-data systems.

The purpose of the present technical note is to discuss the computer implementation of the error equations of Technical Note No. 12, and to present numerical results obtained from the application of the equations to several simple problems.

Computer implementation of the error equations is necessary in almost any nontrivial application, since the error equations have the same order as the problem equation. The fact that the error equations are linear in the error variables is useful, but it does not remove the necessity for machine computation. The particular implementations of the error equations discussed in this note are based on the use of an all-digital computer. Such a computer offers some advantages, but analog or hybrid computers could be used to perform essentially the same computations.

A general digital computer program has been developed to implement all of the error equations. This program, which will be discussed in Chapter 2 of this note, has been designed to handle the error equations in general terms and requires only a limited number of specific inputs to treat any particular case.

Since a part of the present study involves evaluating the error equations, the general computer program has also been given the facility for approximating accurately both the true solution to the problem equation and a hybrid computer solution to the equation.

Chapter 3 of this note presents the results of applying the various error equations in work with several second-order problems. Each of the error equations is evaluated in at least one application by computing an accurate approximation to: (1) the true solution to the problem equation, (2) the hybrid solution*, (3) their difference, or true error, and (4) error as given by the approximate error equation. The objective of this part of the study is to investigate each equation in at least one example to demonstrate that it is theoretically sound, and to give some indication of its accuracy. Later technical notes will deal with studies of sampling and execution time errors in the hybrid computer solution of typical problems. It is expected that the second-order equations used here to test the error equations will also be used in a portion of the later study.

A glossary of symbols employed in this report (similar to those in Technical Note No. 12) is given on page vi above. There will be frequent occasions in the material which follows to reference the general equations of Technical Note No. 12, and to avoid confusion such equations are designated with TN12 preceding the equation number—e.g., equation (TN12:1.2).

*In one example, the simulated hybrid solution for several parameter values is shown to compare favorably with actual hybrid computer solutions furnished by Marshall Space Flight Center.
II. DIGITAL COMPUTER IMPLEMENTATION

2-1. Basic Computer Program

A basic digital computer program has been developed to have the following characteristics:

1. Equations in the standardized, state-variable form

\[ \dot{x} = f(x, y; t) \]  

(2-1.1)

are implemented by reading in on punched cards the \( f \) functions, \( f_1(x, y; t) \), \( \ldots \), \( f_n(x, y; t) \); the forcing function, \( y(t) \); and the required constants and initial conditions.

2. Desired outputs are indicated by specifying them all on a single card.

3. Equations in the standard form can be solved by a Runge-Kutta procedure to approximate the true solution.

4. Equations in the standard form can be solved by a procedure which samples all or part of any or all of the \( f \) functions to simulate a hybrid computer solution.

5. By reading in an appropriate single number the following can be specified: step size for the Runge-Kutta procedure, sampling rate and execution time for the simulated hybrid computer, total solution time, and the times at which data is printed out.

6. Any two variables can be plotted versus time by the digital printer. (This plot is not suitable for precise work.)

7. Several solutions for different values of the constants can be made on one computer run.

A flow diagram and printout of the general basic program is given in Appendix A. The program was used to obtain all of the curves presented

---

*Most of the development of the digital computer program was done by Mr. F. F. Schlae, who was employed part-time on the contract. Some useful modifications and additions to the program were made by Mr. T. H. Tidrick as partial credit for a special problems course in Electrical Engineering. (Mr. Tidrick is a Senior student in EE at Georgia Tech, not employed under the contract.)
in this note with the exception of the true hybrid computer solutions provided by MSFC.

2-2. Approximation of Hybrid Solutions

An evaluation of the capability of the digital program to simulate the hybrid computer was made using the equations for the nonlinear pendulum problem (as written in the standardized, state-variable form):

\[ x_1 = x_2 \]
\[ x_2 = - \sin x_1 - x_2 \]  

This set of equations was solved on an actual hybrid computer at the MSFC Computation Center for a range of parameter values. The results were used with an accurate determination of the true solution for the same parameter values to compute actual hybrid error versus time. Typical results are given in Figures 2-2A and 2-2B for initial conditions of \( x_2 = 0 \) and \( x_1 = 0.8 \) or \( 2 \), and for sampling intervals (\( \delta \)) of 0.05, 0.10, and 0.16 seconds. In the same figures, comparable curves of simulated hybrid error, as computed by the digital program, are also given. Two of the digital simulation curves have zero execution time (\( e \)) and the same sampling intervals as used by the actual hybrid computer. The third digital simulation curve has a sampling interval of 0.1 second and an execution time of 0.03 second; this gives a \( \Delta \) value of 0.16 second (\( \Delta = \delta + 2e \)) to be compared to the actual hybrid computer curve with \( \delta = 0.16 \) second and unknown execution time.

These curves make it reasonable to conclude that the digital simulation provides an accurate approximation of the actual hybrid computer, and that for the chosen test problem the execution time of the MSFC computer was small. The question of exactly what execution time is required to match precisely the actual hybrid computer results is considered unimportant for present purposes. Thus, in subsequent work a digital simulation with \( e = 0 \) was utilized in computing simulated-hybrid problem solutions to permit comparative analysis of the various error equations.

2-3. Basic Implementation of the Error Equations

As noted in Section 2-1, the general digital computer program has the facility for obtaining accurate approximations to true solutions of
Problem Equations
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\sin x_1 - x_2 \\
x_1(0) &= 0.8, \ x_2(0) &= 0
\end{align*}
\]

Legend
- - - Actual Hybrid Error (MSFC)
- - - - Simulated Hybrid Error
\(\delta = 0.1, (e = 0)\)

Figure 2-2A. Actual Hybrid Error (MSFC) and Digitally Simulated Hybrid Error (Assuming Execution Time \(e = 0\)) in Solution of Nonlinear Pendulum Problem Using Sampling Interval of \(\delta = 0.1\).
Figure 2-2B. Actual Hybrid Error (MSFC) and Digitally Simulated Hybrid Error (Assuming Execution Time $e = 0$ or $e = 0.03$) in Solution of Nonlinear Pendulum Problem Using Various Choices of Sampling Interval ($\delta$).
equations in the standard form using the Runge-Kutta procedure. A step size of 0.01 is used in all cases unless otherwise indicated, and except in one case provides an accurate solution. The ability of the digital program to accurately approximate hybrid computer solutions to equations in the standard form has been discussed and demonstrated in the previous section. A "true error" curve is obtained in the computer program by differencing the "hybrid" problem solution and the "true" problem solution.

The error equations themselves can be implemented as follows. For a particular example the error equations are formulated analytically using such general equations as (TN12:2.5) or (TN12:3.7). The result of this formal process is a set of equations requiring as inputs either the true or the hybrid solutions of the problem equation. Since both types of solutions are generated by the general program, either can be used as required by the error equations, and a true or hybrid solution of the error equations can be generated.

In most of the error equation evaluation studies documented herein the following data have been printed out:

1. approximate error as computed from a particular error equation;
2. true error as computed from the difference between true problem solution and hybrid problem solution;
3. other variables such as true problem solution, true error minus approximate error, etc.

2-4. Generalized Implementation of the Error Equations

In problems of practical magnitude the analytical derivation of the error equations for a specific example can become quite tedious. Attention has therefore been given to implementing general forms of the error equation on the computer. An implementation has been developed which makes it possible to read in only the f functions of the problem equations, the true or hybrid solutions of the problem equations if these are not to be computed, the forcing functions, and the constants and initial conditions. The computer then generates the error solutions from the general error equation, eliminating the need for analytical derivation of specific error equations.

The generalized implementation of error equations is accomplished starting with an equation in vector form. For example, the vector equation
equivalent to (TN12:3.7) is

\[
\dot{y}(t) = Ay + \frac{\delta + 2e}{2} [\ddot{f} - Bye - \dddot{e}]
\] (2-4.1)

where the matrixes A and B have the elements \( \partial f_i/\partial x_j \) and \( \partial^2 f_i/\partial x_j \) respectively. For computer work it is convenient to treat the vector time derivatives \( \dot{f} \) and \( \ddot{f} \) as two additional matrixes (called "DF" and "DDF") with elements \( df_i/dt \) and \( d^2f_i/dt^2 \) respectively. Equation (2-4.1) thus represents a set of error equations in the standard form, with the \( f \) functions entering through four matrix quantities (A, B, DF, and DDF). The number of error equations is of course equal to the number of problem equations in standard form.

The general form of the error equations can be implemented by making the following additions to the basic computer program:

1. a general statement of equation (2-4.1);
2. a procedure for computing the elements of A from the f functions of the problem equation and increments in the problem variable x;
3. a procedure for computing the elements of \( \dot{f} \) and \( \ddot{f} \) from the f functions of the problem equations and increments in time t;
4. a procedure for computing the elements of B from the elements of \( \dot{f} \) and increments of x.

The result of making these additions is a general error program for any class of error equations (i.e., error equations based on true problem solutions, or on hybrid solutions, etc.). As noted above, the required input information is the f functions of the problem equations, the number of problem equations, the forcing functions, and all required constants and initial conditions. If the problem equations are not solved by the general program, true or approximate problem solutions can be provided as input data.

A provision is made in the program for the case of partial digital feedback. The net effect on the error equations of having digital elements in some, but not all, of the computational loops is that the total f functions from the problem equations are used in computing some of the matrixes noted above, and only that portion of the f functions generated digitally with the true hybrid computer are used in computing the other
matrixes. The program has a provision for reading in both the total $f$ functions and the digitally generated portion of the $f$ functions so that the error equations for partial digital feedback can be implemented.

Note that the general implementation requires that the computer evaluate first and second time derivatives and partial derivatives with respect to certain variables, while the analytical procedure does not require computer differentiation. Thus, in some problems a difference in accuracy between the two methods could result; or, at best, detailed attention would have to be given to choosing appropriate increments in computing the derivatives.

The general program has been evaluated in only one example, and the results for this case, as shown in Figure 2-4, are quite good.

### 2-5. Implementation of the Calculation of Maximum and Minimum Error Versus Sampling Interval

Technical Note No. 12 presents a method for approximating error directly as a function of sampling interval and execution time. The computation requires equations (TN12:4.5), (TN12:4.8), and (TN12:4.9). To obtain the error as a function of sampling interval and execution time, the two sets of equations (TN12:4.8) and (TN12:4.9) are first solved for $v_{11}(t)$ and $v_{12}(t)$. The error is then computed from equation (TN12:4.5), which reads*

\[ Y_{10}(\Delta, t) = \Delta v_{11}(t) + \frac{1}{2} \Delta^2 v_{12}(t) \]

(2-5.1)

where $\Delta = (\delta + 2e)$.

The quantities $v_{11}(t)$ and $v_{12}(t)$ can be computed in a straightforward fashion using the basic computer program. The error approximation $v_{10}(\Delta, t)$ can then be computed for a given $\Delta$ and $t$, or determined for fixed values of $\Delta$ as a function of time.

A single measure of error for given $\Delta$ but independent of time is

---

* As can be noted from equations (TN12:4.8) and (TN12:4.9), the quantities $v_{11}$ and $v_{12}$ depend only on $t$ and not on $\Delta$. The arguments of $v_{11}$ and $v_{12}$ were erroneously shown to include $\Delta$ in equation (TN12:4.5).
Problem Equations
\[ x_1 = x_2 \\
\frac{dx_1}{dt} = -\sin x_1 + x_2 \\
x_1(0) = 0.8, x_2(0) = 0 \]

Legend
- Error Computed from Analytically Derived Specific Error Equation
- Error Computed from General Error Equation

\[ \delta = 0.1, \epsilon = 0 \]

Figure 2-4. Error in Hybrid Solution of Nonlinear Pendulum Problem as Approximated by Direct Computation from General Error Equation (2-4.1) and from Analytically Derived Specific Error Equation (3-3.2).
desirable in many studies. Two complementary measures of this sort are $\max_{t} \gamma_{10}(\Delta, t)$ and $\min_{t} \gamma_{10}(\Delta, t)$. These quantities can be evaluated as a function of $\Delta$ by the computer program as follows. At each time increment in the computational procedure, $\gamma_{10}(\Delta, t)$ is computed for a selected set of $\Delta$ values. The new results of the computation for fixed values of $\Delta$ are then compared to previously computed "largest" and "smallest" values stored in an appropriate array and the new value is substituted whenever it is larger (or smaller) than the appropriate stored value. At the end of the computation both the maximum and the minimum values of $\gamma_{10}$ for appropriate $\Delta$ values can be made available. It is straightforward to indicate also the time at which the maximum or minimum occurs and this has been done in the program being used.
III. EVALUATION OF ERROR EQUATIONS

3-1. Introduction

In this chapter each of the general error equations of Technical Note No. 12 is applied in at least one test example. The approach is to use the general computer program discussed hereinafore to solve each specific error equation for selected parameter values. The computer is programmed to provide a true error solution in addition to the solution of the approximate error equation. The true and approximate error curves are shown versus time in appropriate plots. The study to date has provided typical results which are considered to be sufficiently extensive to evaluate the error equations. It should be emphasized, however, that the tests have not been exhaustive, and all equations have not been evaluated under all conditions.

Two test problems have been used to study the error equations. The first of these is the nonlinear pendulum problem described by

\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = -a_1 \sin x_1 - a_2 x_2 \]  \hspace{1cm} (3-1.1)

The second problem involves Duffin's equation with a forcing function:

\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = -a_1 x_1 - a_2 x_1^3 - a_3 x_2 + a_4 \cos a_5 t \]  \hspace{1cm} (3-1.2)

An equation describing an optimum servo system is used to evaluate the error equations as a means for compensating hybrid systems.

In each of the following sections of this chapter, the test equation and the specific error equation being evaluated are stated, and numerical results for certain parameter values are given. The specific parameter values are noted on each figure rather than in the text. Execution time and the parameter \(a_4\) in equation (3-1.2) are assumed to be zero in all cases except those given in Section 3-7. Thus, all of the
error equations except those of Section 3-7 do not include the effect of either execution time or a forcing function.

In several cases, notes as to procedures or results are included in the material of this chapter; conclusions based on the test results are presented in Chapter IV. Figures 3-1A and 3-1B give the true solutions to equations (3-1.1) and (3-1.2) respectively, for several sets of parameter values. These true problem solution curves are not repeated below, since primary emphasis is on the error equations.

3-2. Error Equations Using Hybrid Solution Variables

Equations (TN12:2.5) and (TN12:2.9) express approximate error as a function of the hybrid solution variables. These equations differ in two respects. The variable \((t - k\delta)\) appearing in equation (TN12:2.5a) is approximated by its average value, \((\delta + 2e)/2 = A/2\), in equation (TN12:2.9). Further, the derivatives of \(f\) appearing in (TN12:2.5) are avoided in equation (TN12:2.9) by an expedient change of variables.

The first-mentioned change makes equation (TN12:2.9) more approximate than equation (TN12:2.5). However, in most cases the difficulty in obtaining an accurate numerical solution to equation (TN12:2.5) makes the former more attractive computationally. The principal effect of the change of variables is to avoid the necessity of numerically differentiating the forcing function. If there is no forcing function and if \(a(t)/at\) can be computed analytically, then the two equations can be shown to be equivalent when \((t - k\delta)\) is replaced by \(A/2\).

The two equations for error in terms of hybrid solution variables are evaluated primarily with test problem (3-1.1), although one form of equation (TN12:2.9) is used in the compensation method discussed in Section 3-6. Equation (3-1.1) does not involve a forcing function, and execution time is assumed to be zero. Thus, these aspects of the equations are not investigated.

The specific error equations corresponding to (3-1.1) as computed using equation (TN12:2.5) are:

\[
\begin{align*}
\dot{y}_1 &= y_2 - (t - k\delta) \left[ a_1 \sin x_1^* + a_2 x_2^* \right] \\
\dot{y}_2 &= -a_1 y_1 \cos x_1^* - a_2 y_2 - (t - k\delta) \left[ a_1 x_2^* \cos x_1^* - a_2 \left(a_1 \sin x_1^* + a_2 x_2^* \right) \right].
\end{align*}
\]
Figure 3-1A. True Solutions for Typical Cases of the Nonlinear Pendulum Problem.
Problem Equations

\[ \begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -a_1 x_1 - a_2 x_1^3 - a_3 x_2 \\
x_1(0) &= 4, \quad x_2(0) = 0
\end{align*} \]

Legend

I: \( a_1 = 1, a_2 = 0, a_3 = 2 \)
II: \( a_1 = 1, a_2 = 0.3, a_3 = 0 \)

Figure 3-1B. True Solutions for Typical Cases of the Duffin's Equation Problem.
Equation (TN12:2.9) applied to equations (3-1.1) yields

\[\begin{align*}
\gamma_1 & = \gamma_2 - \Delta/2 \left[ a_1 \sin x_1^* + a_2 x_2^* \right] \\
\gamma_2 & = -a_1 \gamma_1 \cos x_1^* - a_2 \gamma_2 - \Delta/2 \left[ a_1 x_2^* \cos x_1^* - a_2 (a_1 \sin x_1^* + a_2 x_2^*) \right]
\end{align*}\]  

(3-2.2)

Note that there is no forcing function for equation (3-1.1) and that the derivatives of the f functions of this example are computed analytically. Thus, equation (3-2.2) results after using the change of variable

\[\chi = z + \Delta f(x^*, y; t)\]

in equation (TN12:2.9).

Comparison of equations (3-2.1) and (3-2.2) shows that they are identical except for the \((t - k\delta)\) and \(\Delta/2\) terms.

Figure 3-2A compares a typical error curve computed with equation (3-2.1) to the true error. Approximate error is computed using two different step sizes in the basic Runge-Kutta numerical procedure for solving the error equation. Note the improvement in accuracy in going from a step size of 0.01 to one of 0.001. For the reason stated in the general discussion above, even the 0.001 step size does not provide accuracy comparable to other equations. Step sizes smaller than 0.001 require excessive computer time and were therefore not investigated.

Results using equation (3-2.2) in a typical case are shown in Figure 3-2B. Note the close agreement between the true and approximate error curves.

3-3. Error Equations Using True Solution Variables

Error equations using true solution variables are investigated for both test examples using equation (TN12:3.7). The specific error equations so derived for problem (3-1.1), with zero execution time, are:
Problem Equations
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\sin x_1 - x_2 \\
x_1(0) &= 0.8, \quad x_2(0) = 0
\end{align*}
\]

Legend
- True Error
- Approximate Error, Using Runge-Kutta with Step Size 0.01
- Approximate Error, Using Runge-Kutta with Step Size 0.01

Figure 3-2A. Comparison of True Simulated-Hybrid Error and Error as Approximated by Equation (3-2.1) Using Hybrid Solution Variables for Nonlinear Pendulum Problem.
Figure 3-2B. Comparison of True Simulated-Hybrid Error and Error as Approximated by Equation (3-2.2) Using Hybrid Solution Variables for Nonlinear Pendulum Problem.
\[ \gamma_1 = \gamma_2 - \frac{\Delta}{2} \left[ a_1 \sin x_1 + a_2 x_2 - (a_1 \gamma_1 \cos x_1 + a_2 \gamma_2) \right] \]  
\[ \gamma_2 = -a_1 \gamma_1 \cos x_1 - a_2 \gamma_2 - \frac{\Delta}{2} \left[ a_1 x_2 \cos x_1 - a_2 (a_1 \sin x_1 + a_2 x_2) \right. \]
\[ \left. + (a_1 a_2 \cos x_1 + a_1 x_2 \sin x_1) \gamma_1 - \left( a_2^2 - a_1 \cos x_1 \right) \gamma_2 \right] . \]

For the second test example, (3-1.2) with execution time and \( a_i \) both zero, the error equations become

\[ \gamma_1 = \gamma_2 + \frac{\Delta}{2} \left[ -a_1 x_1 - a_2 x_1^3 - a_3 x_2 + (3 a_2 x_1^2 + a_1) \gamma_1 + a_3 \gamma_2 \right] \]  
\[ \gamma_2 = -(a_1 + 3 a_2 x_1^2) \gamma_1 - a_3 \gamma_2 - \frac{\Delta}{2} \left[ (a_1 + 3 a_2 x_1^2) x_2 - a_3 (a_1 x_1 + a_2 x_1^3 + a_3 x_2) \right. \]
\[ \left. + a_2 x_1^3 + a_3 x_2 \right) + \left( -6 a_2 x_1 x_2 + a_1 a_3 + 3 a_2 a_3 x_1^2 \right) \gamma_1 \]
\[ + \left( a_3^2 - a_1 - 3 a_2 x_1^2 \right) \gamma_2 \right] . \]

Figure 3-3A gives the solution of equations (3-3.1) for two sets of parameter values. Figure 3-3B gives the solution of equations (3-3.2) for three sets of parameter values. The results shown in Figure 3-3B, involving Duffin's equation, provide a stringent test of the error equations since in this case the problem equation is nonlinear (for curves I and III) and the parameter values are chosen so that true error is relatively large. Note that the curves designated II use a sampling interval of 0.2 seconds.

Agreement between the true and approximate error curves is very good for Figure 3-3A, and is considered adequate in the case of Figure 3-3B.

3-4. Error Equations Versus Sampling Interval

Equation (TN12:4.5) expresses error as

\[ \gamma_{10}(\Delta, t) \approx \Delta \gamma_{11}(t) + \frac{1}{2} \Delta^2 \gamma_{12}(t) \]  
(3-4.1)
Figure 3-3A. Comparison of True Simulated-Hybrid Error and Error as Approximated by Equation (3-3.1) Using True Solution Variables for Nonlinear Pendulum Problem.
Figure 3-3B. Comparison of True Simulated-Hybrid Error and Error as Approximated by Equation (3-3.2) Using True Solution Variables for Duffin's Equation Problem.

Problem Equations

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -a_1 x_1 - a_2 x_1^3 - a_3 x_2
\end{align*}
\]

\[x_1(0) = 4, \quad x_2(0) = 0\]

Legend

- - - - True Error

- - - - Approximate Error

I: \(a_1 = 1, a_2 = 0.3, a_3 = 0; \delta = 0.1\)

II: \(a_1 = 1, a_2 = 0, a_3 = 2; \delta = 0.2\)

III: \(a_1 = 1, a_2 = 0.06, a_3 = 2; \delta = 0.1\)
with \( y_{11} \) and \( y_{12} \) given as the solution to equation (TN12:4.8) and (TN12:4.9). The specific forms of the latter equations as applied to the nonlinear pendulum problem (3-1.1), with zero execution time, are

\[
\dot{y}_{11} = y_{21} - \frac{1}{2} a_2 x_2 - \frac{1}{2} a_1 \sin x_1
\]

(3-4.2a)

\[
\dot{y}_{21} = - a_1 y_{11} \cos x_1 - a_2 y_{21} - \frac{1}{2} a_1 x_2 \cos x_1 + \frac{1}{2} a_2 (a_2 x_2 + a_1 \sin x_1);
\]

(3-4.2b)

\[
\dot{y}_{12} = y_{22} + \frac{1}{2} a_1 y_{11} \cos x_1 + \frac{1}{2} a_2 y_{21}
\]

The corresponding equations for the second test problem, (3-1.2) with execution time and \( a_4 \) both zero, are

\[
\dot{y}_{11} = y_{21} - \frac{1}{2} \left( a_1 x_1 + a_2 x_1^3 + a_3 x_2 \right)
\]

(3-4.3a)

\[
\dot{y}_{21} = - (3 a_2 x_1^2 + a_1) y_{11} - a_3 y_{21} + \frac{1}{2} \left[ - (3 a_2 x_1^2 + a_1) x_2 + a_3 (a_2 x_1^3 + a_1 x_1 + a_3 x_2) \right];
\]

\[
\dot{y}_{12} = y_{22} - \frac{1}{2} \left[ - (3 a_2 x_1^2 + a_1) y_{11} - a_3 y_{21} \right]
\]

(3-4.3b)

\[
\dot{y}_{22} = - (3 a_2 x_1^2 + a_1) y_{12} - a_3 y_{22} + \frac{1}{2} \left[ (6 a_2 x_1 x_2 - 3 a_2 a_3 x_1^2 - a_3 a_3) y_{11} + (3 a_2 x_1^2 + a_1 - a_3^2) y_{21} \right].
\]

In order to evaluate these equations, the error \( y_{10} \) in (3-4.1) is calculated versus time and compared to true error versus time. The results using equations (3-4.2) are shown in Figure 3-4A and those using equations
Figure 3-4A. Comparison of True Simulated-Hybrid Error and Error as Approximated Through a Taylor-Series Expansion by Equations (3-4.1) and (3-4.2) for the Nonlinear Pendulum Problem.
(3-4.3) are shown in Figure 3-4B. In each case, two sets of parameters have been tried and the agreement is quite good.

The primary utility of the equations being evaluated in this section lies in the convenience with which some time-independent function of error can be displayed as a function of sampling interval. Typical results using maximum and minimum error and the computer program discussed in Section 2-5 are shown in Figure 3-4C.

3-5. Error Equations for Partial Digital Feedback

All of the error equations of Technical Note No. 12 apply for partial digital feedback if the nonhomogeneous part of each equation is appropriately modified to account for the fact that only a portion of each \( f \) function is generated digitally. The required modification to any particular equation can be determined by tracing its development to equation (TN12:1.2). In this equation the difference \( f - F \) will be nonzero only for the digitally generated portion of the \( f \) functions.

An equation for partial digital feedback is evaluated for problem (3-1.1) assuming that only the term \(- a_1 \sin x_1\) is generated digitally and that the other terms are generated without sampling error. The error equation using hybrid solution variables is used and the specific equation which corresponds to equation (3-2.2) is

\[
\begin{align*}
\dot{y}_1 &= y_2 \\
\dot{y}_2 &= -a_1 y_1 \cos x_1 - a_2 y_2 - \frac{1}{2} \Delta a_1 x_2^* \cos x_1^*.
\end{align*}
\]

A solution to this approximate error equation is given for one set of parameter values in Figure 3-5, along with the true error curve for partial digital feedback and the true error curve for total digital feedback.

The compensation problem treated in the next section also involves partial digital feedback. In this case, as well as in the case shown in Figure 3-5, agreement between true and approximate error is good.

3-6. Error Equations Used to Compensate Hybrid Systems

An optimum control system is postulated to evaluate the use of the
Figure 3-4B. Comparison of True Simulated-Hybrid Error and Error as Approximated Through a Taylor-Series Expansion by Equations (3-4.1) and (3-4.3) for the Duffin's Equation Problem.
Figure 3-4C. Effect of Sampling Interval (δ) on Maximum and Minimum Values of Hybrid Error as Computed in 10-Second Runs from Equations (3-4.1) and (3-4.2) for Nonlinear Pendulum Problem.
Figure 3-5. Illustrating Effect of Partial Digital Feedback on True Simulated-Hybrid Error and Error as Approximated by Equation (3-5.1) Using Hybrid Solution Variables for Nonlinear Pendulum Problem.
error equations as a means for compensating against sampling errors in hybrid systems.

The control problem can be stated concisely as follows. Consider a plant satisfying the equation

\[
\begin{align*}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -x_2(t) + y(t) \\
X_1(t) &= 1 - x_1(t)
\end{align*}
\] (3-6.1)

where \(X_1(t)\) is the output and \(y(t)\) is the plant input. The variable \(y(t)\) is generated from a system input (in this case assumed to be a unit step) and the current plant state, in such a way that the plant output is optimum with respect to a certain performance measure. The performance measure chosen is

\[
J(t) = \int_t^T \left[ (X_1(t) - 1)^2 + \frac{1}{T} y^2 \right] dt \quad (3-6.2)
\]

where \(t\) is current time and \(T\) is the length of the interval over which the control system is used.

Using standard control theory, it can be shown that the optimum plant input is given by

\[
y(t) = -16 (\psi_{12} x_1 + \psi_{22} x_2) \quad (3-6.3)
\]

where \(\psi_{12}\) and \(\psi_{22}\) are time functions satisfying the equations

\[
\begin{align*}
\dot{\psi}_{11} &= 16 \psi_{12}^2 - 1 \\
\dot{\psi}_{22} &= 16 \psi_{22}^2 - 2 \psi_{12} + 2 \psi_{22} \\
\dot{\psi}_{12} &= 16 \psi_{12} \psi_{22} - \psi_{11} + \psi_{12}
\end{align*}
\] (3-6.4)
with initial conditions dependent on $T$, and equal to $Y_{11}(0) = 0.7498$, $Y_{22}(0) = 0.12491$, $Y_{12}(0) = 0.24987$ for an illustrative value of $T = 3$ seconds.

As a computational problem, equations (3-6.1), (3-6.3), and (3-6.4) must be solved for $X_1$ to obtain the response of the system. From an examination of the equations, it is reasonable to solve equation (3-6.1) with analog equipment, but generate $y(t)$ digitally through the solution of equations (3-6.3) and (3-6.4). Such a computational procedure is subject to sampling errors for which the error equations of Technical Note No. 12 apply.

Using equation (TN12:2.9), as modified to account for partial digital feedback, the following specific error equations can be derived:

$$
\dot{Y}_1 = Y_2
$$

$$
\dot{Y}_2 = -16 \psi_{12} Y_1 - (1 + 16 \psi_{22}) Y_2 + 8\Delta \left[ (16 \psi_{12} \psi_{22} - \psi_{12}) X_1^* + (\psi_{12} - \psi_{22} - 16 \psi_{22}^2 + \psi_{22}) X_2^* \right]
$$

Note that error equations in terms of the hybrid solutions to equation (3-6.1) were chosen for illustration, but other error formulas could also have been used.

Figure 3-6A displays the results of a numerical solution of the control system equations using a sampling period of 0.1 seconds. The upper part of the figure shows a true solution, $X_1$, and an uncompensated hybrid solution, $X_1^*$. The lower part of the figure shows the error in the uncompensated solution, $X_1 - X_1^*$, and the error in a compensated solution. The compensated solution is determined as $X_1^* + Y_1$ and the corresponding error, (plotted in the figure), as $X_1 - (X_1^* + Y_1)$. Note the significant reduction in the error of the compensated solution.

A problem analogous to the computational problem arises in implementing an actual control system. For the case being considered, equation (3-6.1) is the description of a physical device with input $y(t)$. In many systems it is desirable to sample the values of the physical variables.
Figure 3-6A. True and Simulated-Hybrid Solutions of a Control System Problem, with Reduction in True Error Achieved Through Use of Approximate Error Equation to Compensate Hybrid Solution.
x_1(t) and x_2(t) and supply this information to a digital computer which computes y(t). In such a system, sampling causes an error of the same sort as is present for hybrid computation, and the error equations can also be used to compensate such systems. However, in the case of a physical control system the response cannot be corrected by the simple addition of y_1(t) to x_1. Instead, a term to correct the error must be added to y(t). An elementary analysis shows that in this case a term \( \gamma_2 + \gamma_2 \) must be added to y(t) to compensate the system.

Figure 3-6B displays the error in the response of a control system which computes digitally y(t), the solution to the error equation, and y(t) + \( \gamma_2 + \gamma_2 \). The digitally computed quantity is then supplied as input to the control system plant. The figure gives the compensated system error, defined as the difference between actual and ideal system response, and uncompensated system error—both for sampling periods of 0.1 and 0.2 seconds. A significant improvement is evident in both cases.

3-7. Execution Time and Problem Equation Inputs

The work reported in Sections 3-1 through 3-6 has assumed no input to the problem equation and no execution time. Although the effects in question are not investigated in detail, one test example involving each of these terms has been solved to verify the general equations.

Equation (3-1.2) with \( a_4 \) not zero is used to test the effect of a problem equation input on an error equation. The error equation based on true solutions is used and thus equation (3-3.2) can be modified to include the effect of the forcing function. Use of equation (TN12:3.7) shows that the term

\[
\frac{1}{2} a_4 \cos a_5 t
\]

must be added to the right-hand side of the equation for \( \gamma_1 \) and the term

\[
-\frac{1}{2} \Delta (a_3 a_4 \cos a_5 t + a_4 a_5 \sin a_5 t)
\]

must be added to the right-hand side of the equation for \( \gamma_2 \). The solution to the modified error equation and the true error for one set of parameter values are shown in Figure 3-7A. The agreement is good.
Figure 3-6B. Illustrating Reduction in Response Error of a Hybrid Control System Through Use of Approximate Error Equation to Compensate Plant Input Function.
Figure 3-7A. Comparison of True Simulated-Hybrid Error and Error as Approximated by Modified Version of Equation (3-3.2) to Account for Effect of Forcing Function in Duffin's Equation Problem.
The effect of an execution time is tested using equation (3-1.1) and the error equation based on true solution variables. Thus, equation (3-3.1) is modified to include the effect of execution time by adding

\[- \frac{\Delta}{2} e(a_1 a_2 \sin x_1 - a_1 x_2 \cos x_1 + a_2^2 x_2)\]

to the right-hand side of the equation for \( \dot{y}_1 \) and

\[- \frac{\Delta}{2} e [(a_1 \cos x_1 + x_2^2 - a_2^2) a_1 \sin x_1 + 2 a_1 a_2 x_2 \cos x_1 \]
\[- a_2^3 x_2] \]

to the right-hand side of the equation for \( \dot{y}_2 \). The term \( \Delta/2 \) in the original equation is, of course, computed as \((\delta + 2e)/2\). A typical solution of the modified error equation and the true error are shown in Figure 3-7B. The agreement is considered to be acceptable.
Figure 3-7B. Comparison of True Simulated-Hybrid Error and Error as Approximated by Modified Version of Equation (3-3.1) to Account for Effect of Execution Time in Nonlinear Pendulum Problem.
Chapter II of this note presents techniques for implementing the error equations developed earlier in this study. Chapter III presents a representative selection of data collected in the course of evaluating (a) techniques for implementing the error equations, (b) the validity of the equations themselves, and (c) the degree with which they can be expected to approximate true error. Conclusions derived from the foregoing study of the error equations will be given in this chapter under three headings—namely, Theoretical Validity, Degree of Approximation, and General Recommendations.

4-1. Theoretical Validity

Since all of the forms of the error equation are reasonably complicated, it is desirable to check the theoretical derivations by solving appropriate test cases. The numerical solutions given in Chapter III make it possible to compare error computed with each error equation to true error. In all but one case, the agreement between true and approximate error viewed in the light of the form of the equations, would seem to rule out possible analytical errors. The one possible exception to this statement has to do with the effect of execution time as examined in Section 3-7. The terms noted on page 34 as depending explicitly on $e$, amount to a second-order effect and their contribution is small compared to other terms in the error equation. Thus, although there is no specific reason to feel that the theoretical equation is incorrect, the test problem and conditions used do not provide a completely adequate test of the validity of that part of the theoretical equation which yields the terms given on page 34.

With the exception of the one case noted, all of the theoretical error equations have been satisfactorily checked out and no further work of this sort is planned.

4-2. Degree of Approximation

A question of some interest for later work is the accuracy of approximation provided by the error equations. Unfortunately, this is a question which cannot be answered precisely in a general way. Lacking a
general approach to the problem of accuracy, some insight can be gained by noting the discrepancy between true and approximate error in the test cases. To provide as much variety as possible, two different test equations and several sets of parameter values were used in the study. Although each error equation is distinctive in some respects, all of the error equations have a common origin and hence all of the results of Chapter III can be used in commenting on the basic approximation error.

From the nature of the theoretical derivation it would be expected that approximation error would in general increase with the following:

1. sampling interval, $\Delta$, 
2. magnitude of true error, and
3. degree of nonlinearity of the problem equation. Such tendencies can be noted in the test cases.

Relative to increase of approximation error with $\Delta$, note Figure 3-3A which shows that the points representing approximate error for $\delta = 0.05$ lie closer to the true error curve than the points representing approximate error for $\delta = 0.1$. Figure 3-3B shows the same tendency when the curves labeled III, for $\delta = 0.1$, are compared to the curves labeled II, for $\delta = 0.2$. Although in both figures cited there is some increase in approximation error with $\delta$, in neither case is the increase large.

In most of the work with the pendulum equation, the true error is small—on the order of a few percent of the initial value of the problem variable. On the other hand, results using Duffin's equation were chosen with large true error—on the order of twenty-five percent and more of the initial value of the problem solution.

Note that most of the results for the pendulum equation show a close agreement between true and approximate error. On the other hand, note the relatively poorer approximation shown in Figure 3-3B for Duffin's equation when the true error is large. However, in this same figure it is surprising to see how small the difference between true and approximate error curves actually is when the true error is large, and in one case when it tends to become unbounded.

Nonlinear behavior of the problem solution is shown for most of the curves using Duffin's equation and for the curves using the nonlinear pendulum equation with $x_1(0) = 2.0$. Examination of these basically nonlinear cases shows that the error equations give a satisfactory
approximation even when the problem equations are nonlinear. Comparison, for example, of Figures 3-2B, and 3-3A, which apply for linear and nonlinear behavior of the pendulum equation, would seem to show a slight increase in approximation error as the problem becomes nonlinear.

In addition to the general degree of approximation of all of the error equations, each particular equation has its own characteristics. The following conclusions relative to each specific equation are supported by the data:

1. Equation (TN12:2.5) is very poor for computational purposes. (Reference Figure 3-2A above.)

2. Equation (TN12:2.9) based on hybrid solution variables and equations based on true solution variables give comparable accuracy, although the equations based on hybrid solutions may be slightly better. (Reference Figure 3-2B and Figure 3-3A, curve I.)

3. The additional approximation involved in expressing error versus \( \Delta \) introduces little additional error. (Reference Figure 3-4A and Figure 3-4B.)

4. The error equations apply with comparable accuracy to the case of partial digital feedback. (Reference Figure 3-5.)

5. For the one example investigated, the error equations, when used to compensate for sampling error, provide a reduction in error on the order of ten-to-one. (Reference Figure 3-6A and Figure 3-6B.)

6. Execution time or an input function do not appreciably degrade the accuracy of approximation. (Reference Figure 3-7A and Figure 3-7B.)

4-3. General Recommendations

Of the general forms of the error equation, the form given in equations (TN12:4.5), (TN12:4.8), and (TN12:4.9) which yields error as a function of \( \Delta \), would seem to be the most useful for general studies of the effect of \( \Delta \). This equation makes it possible to determine, say, the maximum and minimum error as a function of \( \Delta \) as a single solution to a set of equations. To approximate error with two terms in a power series, as is done in equation (TN12:4.5), twice as many error equations as problem equations are required. A less accurate approximation could be obtained using a single term in the power series and this would require
only half as many error equations.

Error equations in the form being discussed could be expressed in terms of either true or hybrid solution variables, depending on which are the more easily obtained. (Equations (TN12:4.8) and (TN12:4.9) are in terms of true solution variables.)

This study has provided two essentially equivalent forms in which the error equation can be expressed for use in compensating hybrid systems. The form chosen will depend on which variables, true or hybrid, are most easily obtained. The results of the test example indicate that the compensation of hybrid systems may prove to be a fruitful area in which to use the equations, since considerable improvement in accuracy is possible with the compensated system.

The method of implementing the solution of the error equations on a digital computer has proved to be very practical for small-scale problems. For example, the general form of the error equations yielding error versus $\Delta$ can be implemented so that to consider any particular problem the $f$ functions of the problem equation, the forcing functions, the initial conditions, and constant parameters are read in, and a table of, say, maximum and minimum error versus $\Delta$ can be printed out.

The computer implementation has not yet been completely evaluated for large scale systems; but, as might be expected, certain problems such as stability of the equations and computational capacity can arise. Some of these problems will be investigated and reported in later technical notes.
Appendix:
DETAILED INFORMATION ON BASIC COMPUTER PROGRAM

Presented below are a simplified flowchart and computer printout of the general basic program developed for the Burroughs B-5500 digital computer at Georgia Tech to generate the various kinds of data used in this report, as discussed under Chapter II.

This material should be largely self-explanatory, but the following points may be of some assistance in interpreting the program:

(a) The procedure "ANA EQ" (extending between lines 0024 and 0049 in the printout) serves to specify the f functions for a given problem--each component f_i being denoted by the computer as F[i]. The symbols E[ ] denote those parts of the f functions that are assumed to be generated by the digital portion of the hybrid computer for which error estimates are desired.

(b) The procedure "DIG EQ" (lines 0049 to 0060) specifies the above-noted digitally generated parts of the f functions--the symbols E[ ] being synonymous with E[ ].

(c) The procedure "AUX EQ" (lines 0061 to 0099) specifies a set of auxiliary functions designed to simplify the writing of other equations.

(d) The procedure "OUT EQ" specifies which computer variables will be printed out.
SIMPLIFIED FLOWCHART OF BASIC COMPUTER PROGRAM

1. Declarations
2. Read and Store Input Data
3. Write Headings and Initial Condition Values
4. Evaluate "Digital" f Functions
5. Execution Time
6. Evaluate "Analog" f Functions
7. Runge-Kutta Procedure
8. Write Outputs
   - If Time less than Sample Time
   - If Time less than Run Time
9. Finish
GA TECH
8.5500 DC ALGOL COMPILER 12/12/66,
LDMRENT - HYD-ER-AK - P-RODRAMT -HTBAID—DIFTERENTIALEUUATTONS By RUNGE
AND KUTTA
ALGIN

1605.
0000
0000
UUUU

START
CARD" -KAROL . T2iI0),
LINE 6(2,15)1
TITLE(.RUNGE SUTTA

FILE IN
FILE OUT
FORMAT

SOLUTION

OF

OF

0002

SEGMENT
0000
0003
0007

Y.FIX,TI.//
START

OF

SEGMENT

0003

P.4,. "H5".
I , 1
HI
X8,"H6",X8,.H7",X8,.H8.,X8,"H9.,X8,"H10"))

0003

FORMAT

FMTIN(I2,12,1

F ORMAT

FMTI/(10(

FORMAT

PI2,R10.0,R10.0,R10.0PRIO.0,R10.0))

810.0)))

FMIOUT(12(R10.5))1

FORMAT

MARC

LOUA(IUN UNDER AND TTWE -FTRAFILTIRS /

XI.N . X- 1"M .X1 . 0 . Xl.R.X1.0ELTA
FORMAT

INCONC.EQUATION

INITIAL

T.X3.SAMPLE

T.X2"TOTAL

T"))

CONDITIONS"/

xa xe xa TS XR T4 X6

X5 YI
X8..10.);

XM Ts XM

If

X8 T8 Xb TY

NEXT SEG
0007
2
START OF SEGMENT ..... *****
0004 IS 0012 LONG, NEXT SEG
000
START OF SEGMENT
0005 IS 0006 LONG, NEXT SEG
0007
START OF SEGMENT
0006 IS 0006 LONG, NEXT SEG
0007
START OF SEGMENT
0007
0007 IS 0027 LONG, NEXT SEG
0007
START OF SEGMENT
UUU,
0008

FORMAT

CONST(.EQUATION

IS

IS

0032

0029

LONG.

LONG,

CONSTANTS./
START

OF

IS

0027

LONG,

TIME"/
START OF

0010

REAL
PROLLDURE
BEGIN

--

pa-urn:mat

STARTrBOYoMELVAPBEG,FINISH,STAR,NOPRT.NOXJ
K,L,N,MAI,J,Q,R01/
01014.0110)A.001107,FC0110],G(0110],HC01101,SSS(01250),
Er0110), 8C01107, AC0110),ET[Ot10],
DT,TT,ICT,RIPP,SToNT,PT)
ANALA)

FC11 4 0(2)J
FC21..Y/21+ECIII
FC3)+G[1])
F[4].G(21)
j;
FLSJ.
FC6I.YE7II
FC7]..16xY151.Y[6].(1*16xY(41).Y(7)+E[31)
FCC/1+1197J
F[9]..Y/97.16X(Y(51XY[81*YC4).Y[91)1
END ANAEQJ

IS

NEXT
0007

0027

LONG,

0007
0007
NEXT
0007
0007

0 00 01 87
0024
0024

00TO
41
UU%!

!!! 4

O' 0
C;:

g

(0) 4

-1-6E-97

BEGIN
BII/v.16


aRs2432,1401400BI- 3- 1.-GTIO 1
END DIGEHI
POC-E-OUR-E AUXED
BEGIN
GE11+.1.16.Y/51.2)
GC21+.2.Y(57*2xYC4I*16.Y(41.21
GI31.16xYI5).Y[4].YT31+ -NISIT
G[7].YE11.0(611
0181.1. 121 ■ 0/711
G191.16x(Y151xUlf]..I4IxG181).1
GC101. .8 x ST. (Y(51 x G[)7[4] x (GO F. 16 x(Y/51
G(7) + 0C41
GC71 x GC3I + 6[8] x GC21)1 4
x GC8)))
END AUXEQI
PROCEDURE OUTEQJ
BEGIN
H111.F(677
H(21+Y(8)..C17/
H(31‘Y(8).1.07
H(41.Y(11.1.0)
HC57+Y(7)/
HI6I+119).X[2])
Y(
C
END OUTEDI
TIPUTIN,NX0pRiOT,ST,TTAPT/7
LIST
LIST
1JPUT(FOR K.I STEP 1 UNTIL N DO YEK)))
-IKPUT(FOR X+1 - STEP 1 UNTIL R DU ACK)))
LISTLIST
ILPUT( FOR K.1 STEP 1 UNTIL 8 DO ETIK1);
000PUT(RI .EU HI STEP 1 UNTIL M DU HCK))1
LIST
(NO])
FRITE(LINE
I %PRINT FILE.ID GOES IN COLS. 7.13. PUT AFTER DECL.

0006
0002
0007
0002
0008

0002

0002
0010

SEGMENT

X8.E10.)J
LABEL
INTEGER
REAL ARRAY

0005
0002

0007
0009

EXCT(.EXCUTION

00-04
0002

0009

SEGMENT

Am A u 7,
-FURMAT

NEXT SEG
0007

0002

0049
.
0049

7E219:
0060
00:
0061
0061
0064
0069
0079
0081
gg 8
87
0093
0099
0101

101

0101
0102
0104
0106
0108
0110
0112
0114
0116
0130
0139
0148
0157
0167

UUU/

