AN ANALYTICAL MODULARIZED TREATMENT OF AUTOPILOTS FOR GUIDED PROJECTILE SIMULATIONS

EUGENE M. FRIEDMAN
MICHAEL J. AMORUSO

AUGUST 1985

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER
LARGE CALIBER WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.
The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

Destroy this report when no longer needed. Do not return to the originator.
An innovative technique was developed by which piecewise analytic solutions to guidance and control transfer functions were obtained for use in larger but lower frequency computer simulations of guided munitions. Numerical integration, which is typically used to treat such transfer functions, significantly reduces the integration time step, and increases computer execution time for the overall simulation. The modularized analytical approach accommodates integration time (cont)
steps associated with the lower frequencies of airframe dynamics which results in faster simulation execution and considerable savings in cost and execution time.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Discussion</td>
<td>1</td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
</tr>
<tr>
<td>First-Order Lag</td>
<td>2</td>
</tr>
<tr>
<td>First-Order Lag with Differentiation</td>
<td>4</td>
</tr>
<tr>
<td>First-Order Lead/Lag</td>
<td>5</td>
</tr>
<tr>
<td>First-Order Lag with Integrator</td>
<td>6</td>
</tr>
<tr>
<td>Second-Order Lag/Oscillator</td>
<td>8</td>
</tr>
<tr>
<td>Results</td>
<td>13</td>
</tr>
<tr>
<td>Conclusions</td>
<td>14</td>
</tr>
<tr>
<td>Appendixes</td>
<td></td>
</tr>
<tr>
<td>A Program Listing and Sample Execution</td>
<td>15</td>
</tr>
<tr>
<td>B Program Usage</td>
<td>31</td>
</tr>
<tr>
<td>Symbols</td>
<td>35</td>
</tr>
<tr>
<td>Distribution List</td>
<td>39</td>
</tr>
</tbody>
</table>
INTRODUCTION

A substantial need has been demonstrated within Armament Research and Development Center (ARDC) for a convenient-to-use and computationally efficient flight simulation for smart munitions. The simulation must be easily adaptable to varying projectile designs and especially to changes in autopilots which undergo rapid evolution. During the design cycle of the projectile, a large number of individual flight simulations must be performed. As an example, during the development of the Copperhead projectile, the ARDC scientific computers were overwhelmed by the workload and a substantial amount of the available time on the BRL CDC 7600 supercomputer was dedicated to the computations. In spite of the continuous development and improvement in computer systems, flight simulations will continue to provide a substantial workload which must be accomplished efficiently.

To meet this need, ARDC is formulating methods and writing modular software for rapidly developing autopilot simulations for guided projectiles and munitions. An innovative technique was developed by which exact analytical solutions to the required transfer functions are applied in a piecewise manner within a larger but lower frequency problem which must be solved numerically.

Since the time constants associated with the autopilot components are often small compared with their driving terms, the integration time step is driven to very small values to obtain stable numerical integrations which results in very long computer run times. The use of piecewise analytical solutions to the transfer functions guarantees valid integration of the autopilot transfer functions regardless of the integration time step, provided that the driving terms vary inappreciably over the time step. This is a less stringent requirement than is needed for stable numerical integration since the driving term rates are commensurate with the airframe time constants, which are typically large compared to that of the autopilot.

Five transfer functions were solved in closed form to provide a fast executing computer code. These particular transfer functions were selected since an autopilot can generally be represented by a concatenation of these functions to rate sensors, switches, limiters, and dead zones.

DISCUSSION

Apart from switches, limiters, and dead zones, autopilot transfer functions are Laplace transform representations of differential equations. Incorporated into a digital simulation, transfer functions are converted into differential equations and are usually solved numerically. Sometimes this causes considerably longer run times than simulation of unguided projectiles.

The time step required to perform the integration is driven by two constraints: (1) The driving term or input must not vary appreciably in the time step. (2) The time step must be sufficiently small to insure a stable
integration. If the time constants associated with the autopilot are small compared to those of the airframe (which is typical), the integration time step will be driven to very small values and run times will be very long. Since the driving term is a product of the airframe motion, an inherently slower process, stable integration is a stronger driver to fine integration than the driving term's remaining essentially constant during the integration time step. By analytically solving these transfer functions stepwise with constant driving terms, the second constraint is eliminated. The size of the integration time step is limited only by the first constraint. Therefore, the use of closed form, analytic, stepwise solutions subject only to the first constraint can lead to larger integration time steps and shorter run times.1

Only five transfer functions are needed to handle the typical autopilot: the first-order lag, the first-order lag with differentiation, the first-order lead/lag, the first-order lag with integrator, and the second-order lag/oscillator. (Guidance and control systems often contain rate sensors. The treatment of rate sensors will be published in a separate report.) These transfer functions are described and solved analytically below; the computer code implementing the solutions is shown in appendix A.2

Throughout this development, the driving term is assumed not to vary appreciably during a time step. Care must be taken to adjust the time step downward as necessary to satisfy this requirement. To take advantage of this technique for reducing run time, the time step should be increased whenever the driving terms are varying slowly.

ANALYSIS

First-Order Lag

A first-order lag is represented by the Laplace operator \( s \) as

\[
\frac{1}{(Ts + 1)}
\]

In differential equation form

\[
T \frac{dy}{dt} + y = D
\]

(1)


where

\[ y = \text{output or dependent variable} \]
\[ t = \text{time or independent variable} \]
\[ T = \text{time constant} \]
\[ D = \text{driving term} \]

The general solution to equation 1 is the sum of a particular solution and a general solution to the homogeneous form. The homogeneous equation was obtained by setting D to zero and can be written as

\[ -T \frac{dy}{y} = dt \]

which can be verified to yield the solution

\[ y = e^{-\frac{(t - A)}{T}} \]  

where \( A \) is the integration constant determined from the initial conditions \( y = y_o \) at \( t = t_o \). Using the initial conditions yields the general solution to the homogeneous equation 2.

\[ y = y_o e^{-\frac{(t - t_o)}{T}} \]  

(3)

For a particular solution to equation 1, the expression is verified by substitution

\[ y = \left[ 1 - e^{-\frac{(t - t_o)}{T}} \right] D \]  

(4)

Adding equations 3 and 4 gives the complete general solution

\[ y(t) = y_o e^{-\frac{(t - t_o)}{T}} + \left[ 1 - e^{-\frac{(t - t_o)}{T}} \right] D \]  

(5)

Note that equation 5 has the correct limit of D as T goes to zero.
First-Order Lag with Differentiation

This transfer function has the Laplace operator representation

\[ \frac{s}{Ts + 1} \]

or the time domain form

\[ T \frac{dy}{dt} + y = \frac{dD}{dt} \] (6)

This representation is not desirable since the driving term appears as a time derivative that would generally have to be evaluated numerically. However, equation 6 can be recast as a pair of coupled differential equations that do not contain any derivatives of the driving term \( D \). An auxiliary variable \( z \) is introduced and the equation is replaced by the expressions

\[ \frac{dz}{dt} = y \] (7)

and

\[ y = \frac{(D - z)}{T} \] (8)

Differentiating equation 8 and using equation 7 to eliminate \( \frac{dz}{dt} \) verifies that these two equations are equivalent to equation 6. Note that the derivative of \( D \) does not appear. By combining equations 7 and 8, the expression for \( z(t) \) is obtained,

\[ \frac{dz}{dt} + \frac{z}{T} = \frac{D}{T} \] (9)

and, after integration, the results for \( z(t) \) substituted into equation 8. Equation 9 can be integrated from equation 5 by noting the identity of equations 1 and 9. Therefore,

\[ y(t) = e^{-\frac{(t - t_o)}{T}} \left( \frac{D - z_o}{T} \right) \] (10)

where \( z_o = z \) at \( t = t_o \). The auxiliary variable \( z_o \) can be eliminated by using equation 8.
\[ y(t) = \left\{ \begin{array}{l} e^{-(t - t_0)/T} \\ \frac{-(t - t_0)/T}{T} \end{array} \right\} (D - D_0) + y_0 e^{-(t - t_0)/T} \] (11)

Note that the exponential over T goes to \(1/(t - t_0)\) by L'Hopital's rule so that equation 11 has the correct limit \(dD/dt\) as \(T\) goes to zero.

**First-Order Lead/Lag**

This transfer function is represented by

\[ \frac{T_2 s + 1}{T_1 s + 1} \quad T_1 \neq 0 \]

or

\[ T_1 \frac{d y}{d t} + y = T_2 \frac{d D}{d t} + D \] (12)

The derivative of the driving term is usually not available in analytic form. It is eliminated by introducing an auxiliary variable, \(z\), and a pair of coupled equations

\[ \frac{d z}{d t} = \frac{D - y}{T_1} \] (13)

and

\[ y = \left( \frac{T_2}{T_1} \right) D + z \] (14)

The equivalence with equation 14 can be verified by differentiating it and then eliminating \(d z/dt\) with equation 13. To obtain an expression for \(y(t)\), integrate equation 14 as follows:

\[ T_1 \frac{d z}{d t} + z = \left( 1 - \frac{T_2}{T_1} \right) D \] (15)
This is obtained by substituting equation 14 into equation 13, and then substituting the result for z(t) into equation 14 to obtain y(t). The explicit solution will not be written. Instead, an algorithm will be provided based on the observation that equation 15 is equivalent to equation 1 if the following substitutions are made:

\[ T_1 + T \quad z \rightarrow y \quad (1 - T_2/T_1)D + D \quad (16) \]

Algorithm:

1. Substitute equation 16 into equation 5 to obtain z(t).
2. Save z(t) for use as \( z_o \) in the next integration step.
3. Substitute z(t) into equation 14 to obtain y(t).

Note that the lag time constant cannot be zero in equation 16, but a vanishing lead time constant yields the correct limiting case of a simple first-order lag.

**First-Order Lag with Integrator**

The representation of this transfer function is

\[ \frac{1}{s} (Ts + 1) \]

or

\[ T \frac{d^2y}{dt^2} + \frac{dy}{dt} = D \quad (17) \]

Substitution verifies that the following expression is a particular solution to the nonhomogeneous differential equation 17 since, by assumption, D is constant during the integration time step:

\[ y = Dt \quad (18) \]

The homogeneous form of equation 17 can be written as the coupled equations

\[ T \frac{dz}{dt} + z = 0 \quad (19) \]
and

\[ z = \frac{dy}{dt} \]  \hspace{1cm} (20)

Equation 19 is similar to equation 1 with the substitution of \( z \) for \( y \) and zero for \( D \); therefore, the solution can be obtained from equations 2 and 20

\[ \frac{dy}{dt} = z(t) = e^{-\frac{(t - A)}{T}} \]

or

\[ y(t) = -Te^{-\frac{(t - A)}{T}} + B \]  \hspace{1cm} (21)

The complete solution to equation 17 is the sum of the general homogeneous solution (eq 21) and the particular nonhomogeneous solution (eq 18)

\[ y(t) = -Te^{-\frac{(t - A)}{T}} + Dt + B \]  \hspace{1cm} (22)

\[ \frac{dy}{dt} = e^{-\frac{(t - A)}{T}} + D \]  \hspace{1cm} (23)

The integration constants \( A \) and \( B \) can be determined by invoking the initial conditions \( y = y_o \) and \( \frac{dy}{dt} = y' \) at \( t = t_o \)

\[ y_o = -Te^{-\frac{(t_o - A)}{T}} + D t_o + B \]  \hspace{1cm} (24)

\[ y' = e^{-\frac{(t_o - A)}{T}} + D \]  \hspace{1cm} (25)

The integration constant \( A \) can be eliminated from equation 24 by using equation 25 to eliminate the exponential term and solving for \( B \)
\[ B = y_o + T (y' - D) - D t_o \]  \hspace{1cm} (26)

Taking the natural logarithm of equation 25 yields

\[ A = T \ln (y' - D) + t_o \]  \hspace{1cm} (27)

The explicit solution is therefore

\[ y(t) = y_o + T (y' - D) \left[ 1 - e^{-\frac{(t - t_o)}{T}} \right] + D (t - t_o) \]  \hspace{1cm} (28)

and

\[ \frac{dy}{dt} = (y' - D) e^{-\frac{(t - t_o)}{T}} + D \]  \hspace{1cm} (29)

Note that equations 28 and 29 go to the correct limits if T goes to zero.

**Second-Order Lag/Oscillator**

This transfer function can be represented in the form

\[ \frac{1}{Is^2 + Ds + K} \] \hspace{1cm} \[ K \neq 0, I \neq 0 \]

or

\[ I \frac{d^2 y}{dt^2} + \frac{dy}{dt} + Ky = T \]  \hspace{1cm} (30)

The homogeneous solution to equation 30 can be verified to be

\[ y(t) = e^{Lt} \]  \hspace{1cm} (31)

by substituting into the homogeneous form of equation 30 to obtain the characteristic equation

\[ IL^2 + DL + K = 0 \]  \hspace{1cm} (32)
which has the following roots:

\[ M = \left[ -D + (D^2 - 4IK)^{1/2} \right]/2I \]  \hspace{1cm} (33)

\[ N = \left[ -D - (D^2 - 4IK)^{1/2} \right]/2I \]  \hspace{1cm} (34)

The following three cases are treated separately according to whether the radicand is positive, negative, or zero.

**Case 1. Positive radicand, damped solution**

The homogeneous and nonhomogeneous particular solutions can be verified to be

\[ y_h(t) = Ae^{Mt} + Be^{Nt} \]  \hspace{1cm} (35)

\[ y_p(t) = T/K \]  \hspace{1cm} (36)

The complete solution is

\[ y(t) = Ae^{Mt} + Be^{Nt} + T/K \]  \hspace{1cm} (37)

\[ dy/dt = MAe^{Mt} + NBe^{Nt} \]  \hspace{1cm} (38)

The initial conditions \( y_0 = y \) at \( t_0 \) and \( y' = dy/dt \) at \( t_0 \) determine the constants \( A \) and \( B \).

\[ y_0 = Ae^{M_0} + Be^{N_0} + T/K \]  \hspace{1cm} (39)
\[ y' = MAe^{-t} + NB e^{-t} \]

Solving simultaneously gives

\[ B = (y_0 - y'/M - T/K)e^{-Nt} / (1 - N/M) \]  \hspace{1cm} (41)

\[ A = (y_0 - y'/N - T/K)e^{-Mt} / (1 - M/N) \]  \hspace{1cm} (42)

Case 2. Zero radicand, critically damped

The characteristic equation has only one distinct root

\[ L = -D/2I \]  \hspace{1cm} (43)

The homogeneous general solution and nonhomogeneous particular solution can be verified to be

\[ y_H(t) = Ae^{Lt} + Bte^{Lt} \]  \hspace{1cm} (44)

\[ y_H(t) = T/K \]  \hspace{1cm} (45)

The complete solution is

\[ y(t) = (A + Bt)e^{Lt} + T/K \]  \hspace{1cm} (46)

\[ dy/dt = (LA + B + LBT)e^{Lt} \]  \hspace{1cm} (47)
Using the initial conditions \( y_0 = y \) at \( t_0 \) and \( y' = \frac{dy}{dt} \) at \( t_0 \), the integration constant can be determined as follows:

\[
y_0 = (A + Bt_0)e^{Lt_0} + T/K
\]  
(48)

\[
y' = (LA + B + LBt_0)e^{Lt_0}
\]
(49)

\[
B = e^{-Lt_0} \left[ y' + L \left( \frac{T}{K} - y_0 \right) \right]
\]  
(50)

\[
A = \left[ (y_0 - \frac{t}{K}) \left( 1 + Lt_0 \right) - t_0y' \right] e^{-Lt_0}
\]  
(51)

Case 3. Negative radicand, oscillation

The roots of the characteristic equation become complex numbers

\[
\omega = \left[ \left( \frac{K}{I} \right) - \left( \frac{D}{2I} \right)^2 \right]^{1/2}
\]  
(52)

The homogeneous and nonhomogeneous particular solutions are

\[
y_H(t) = Ae^{Lt} \sin (\omega t + \phi) + \frac{T}{K}
\]  
(53)

\[
y_H(t) = \frac{T}{K}
\]  
(54)
where $A$ and $\phi$ are integration constants. The complete solution is

$$y = Ae^{Lt} \sin (\omega t + \phi) + \frac{T}{K}$$  \hspace{1cm} (55)

$$\frac{dy}{dt} = LAe^{Lt} \sin (\omega t + \phi) + \omega Ae^{Lt} \cos (\omega t + \phi)$$  \hspace{1cm} (56)

Using the initial conditions $y_o$ and $y'$ and solving for $A$ in the expression for $y_o$ gives

$$A = (y_o - \frac{T}{K})e^{-Lt_o/sin(\omega t_o + \phi)}$$  \hspace{1cm} (57)

where

$$\sin(\omega t_o + \phi) \neq 0$$  \hspace{1cm} (58)

If equation 58 fails, $A$ can be obtained from the expression for $y'$

$$A = y'e^{-Lt_o/[L \sin(\omega t_o + \phi) + \omega \cos(\omega t_o + \phi)]}$$  \hspace{1cm} (59)

Since $\omega$ in equation 52 is always positive definite and the sine and cosine are orthogonal, either equation 57 or 59 is always defined. Note that the expressions still contain the phase angle $\phi$. Solving for $\phi$ yields the same result whether equation 57 is substituted into the initial condition on $dy/dt$ or equation 59 into the initial condition on $y$.

$$\phi = \tan^{-1} \left[ \frac{\omega(y_o - \frac{T}{K})}{[y' - L(y_o - \frac{T}{K})]} \right] - \omega t_o$$  \hspace{1cm} (60)

The phase $\phi$ can be eliminated in equation 57 or 59 by substituting equation 60. Equation 60 shows that equation 58 is equivalent to

$$y_o - \frac{T}{K} = 0$$  \hspace{1cm} (61)
The imposition in equation 30 of the requirement that parameters K and I not
vanish was necessary for the validity of the solutions. For example, equations
36, 45, and 54 become singular when K vanishes. Similarly, the roots of the
characteristic equation are singular if I is zero (eqs 33, 45, and 52). This is
not a limitation since equation 30 reduces to a first-order lag with integrator
when K vanishes and to a simple first-order lag when I vanishes.

However, the parameter D in equation 30 may be zero. If D vanishes, the
characteristic equation 32 degenerates to

\[ \frac{1}{2} + K = 0 \]

\[ M = \pm (-K/L)^{1/2} \pm i \omega \]

Therefore, the roots are imaginary corresponding to a pure oscillation with no
damping. Only case 3 should be considered. Nowhere does D appear in the denomi-
nator of any expression; therefore, the solution remains valid if D is zero.

RESULTS

The exact saving resulting from this piecewise analytical technique depends
on the speed of execution of the competing numerical method. A sample run is
included in appendix B so that comparisons can be made with other preferred
numerical integration techniques. For comparison, the last case (KTFID=13) was
run analytically with time steps of 0.01 and 0.005 second and run numerically
using Advanced Continuous Simulation Language (ACSL).\(^3\) Whereas the analytical
method gave the same results for time steps 0.001 and 0.005 second, ACSL could
not operate with a time step of 0.001 second. ACSL was able to obtain nearly
converged results with a time step of 0.00001 second but failed after the third
second. These results are compared as follows:

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Analytical DT=0.01 sec</th>
<th>Analytical DT=0.005 sec</th>
<th>Numerical (ACSL) DT=0.00001 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.00000</td>
<td>10.00000</td>
<td>10.00000</td>
</tr>
<tr>
<td>1</td>
<td>1.54640</td>
<td>1.54640</td>
<td>1.54381</td>
</tr>
<tr>
<td>3</td>
<td>-4.49206</td>
<td>-4.49206</td>
<td>-4.48494</td>
</tr>
</tbody>
</table>

\(^3\) Advanced Continuous Simulation Language, User Guide/Reference Manual, Mitchell
In this particular case, the analytical technique resulted in at least two orders of magnitude improvement in integration step size.

CONCLUSIONS

The modular, piecewise analytical technique presented in this report can result in considerable savings in analysis and associated computer time.
APPENDIX A

PROGRAM LISTING AND SAMPLE EXECUTION
PROGRAM LISTING

PROGRAM TFMAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C       MAIN PROGRAM TO DRIVE TRANSFER FUNCTION SUBROUTINE.
C       FOR TEST PURPOSES ONLY.
COMMON/TRNSF/TAU1(50),TAU2(50),TAU3(50),TFTIME(50),TFAUX(50),
        + TFAUX2(50),KTFTYP(50),GAIN(50)
OPEN(8)
WRITE(8,11)
C       READ INPUT DATA AND INITIALIZE.
CALL TRNF IO
WRITE(8,11)
C 200 DEFINE INTEGRATION TIME STEP DT.
   DT = .005
   DT2= DT/2.
   WRITE(8,13) DT
   DO 2 J=1,13
   WRITE (8,12)
   C 100 DEFINE INITIAL TIME T AND FINAL TIME TMAX.
      T = 0.0
      TMAX = 5.0
      KMAX = IFIX(TMAX/DT)+1
      DO 1 K=0,KMAX
         T = DT*FLOAT(K)
   C 300 DEFINE DRIVING TERM "DRIVE"
      DRIVE = COS(2.*T) - 1.0
      CALL TRNF (DRIVE,XOUT,T,J)
      IF(ABS(IFIX(T)-T).LT.DT2) WRITE(8,10) J,T,DRIVE,XOUT
      1 CONTINUE
      2 CONTINUE
      STOP
10 FORMAT(1X,I5,3F20.5)
11 FORMAT('1')
12 FORMAT(//,' LTFID',16X,'TIME-',14X,'DRIVE',19X,'X')
13 FORMAT(' ' THE INTEGRATION TIME STEP DT = ',F9.6,' ')
   CND

SUBROUTINE TRNF IO
C
   LTFTYP  1 -> GAIN/(TAU1*S+1)
   C
   2 -> GAIN*S/(TAU1*S+1)
   C
   3 -> GAIN/(S*(TAU1*S+1))
   C
   4 -> GAIN*(TAU2*S+1)/(TAU1*S+1)
   C
   5 -> GAIN/(TAU1*S**2 + TAU2*S +TAU3)
   C
   LTFID  ID NO. OF CHANNEL (VARIABLE).
C   TU1    FIRST TIME CONSTANT
C   TU2    SECOND TIME CONSTANT (LTFTYP=4 OR 5)
C   TU3    THIRD TIME CONSTANT (LTFTYP=5)
C X0     INITIAL CONDITION ON VARIABLE
C DX0    INITIAL CONDITION ON DERIVATIVE (LTFTYP=5)
C AA     ALPHANERIC LABEL FOR PRINTOUT
COMMON/TRNSF/TAU1(50),TAU2(50),TAU3(50),TFTIME(50),TFAUX(50),
+ TFAUX2(50),KTFTYP(50),GAIN(50)
CHARACTER AA*30,LABEL(5)*30
DATA LABEL/'GAIN/(TAU1*S+1)
+ 'GAIN*S/(TAU1*S+1)
+ 'GAIN/(S*(TAU1*S+1))
+ 'GAIN*(TAU2*S+1)/(TAU1*S+1)
+ 'GAIN/(TAU1*S**2+TAU2*S+TAU3) '/
WRITE (8,1050)
DO 1  I=1,50
  TAU1(I)    = 0.
  TAU2(I)    = 0.
  TAU3(I)    = 0.
  TFTIME(I)  = 0.
  TFAUX(I)   = 0.
  TFAUX2(I)  = 0.
  KTFTYP(I)  = 0
  GAIN(I)    = 0.
1 CONTINUE
IFLAG = 0
DO 2  I=1,50
READ (5,1051,END=4) LTFID,LTFTYP,TU1,TU2,TU3,GA,X0,DX0,AA
IF (LTFTYP.LT.1 .OR. LTFTYP.GT.5) THEN
  WRITE(8,1052)LTFID,LTFTYP,TU1,TU2,TU3,GA,X0,DX0,AA
  IFLAG = 1
  WRITE(8,1004)
  GO TO 2
ELSE
  WRITE(8,1052)LTFID,LTFTYP,TU1,TU2,TU3,GA,X0,DX0,AA,LABEL(LTFTYP)
END IF
IF (LTFID.LT.1 .OR. LTFID.GT.50) THEN
  IFLAG = 1
  WRITE(8,1001)
  GO TO 2
END IF
IF(TAU1(LTFID).NE.0 .OR. TAU2(LTFID).NE.0 .OR. TAU3(LTFID).NE.0.
+ .OR.KTFTYP(LTFID).NE.0) THEN
  IFLAG =1
  WRITE(8,1002)
  GO TO 2
END IF
IF(TU1.GT.0 .AND. TU2.EQ.0 .AND. TU3.GT.0 .AND. LTFTYP.EQ.5)
+ .AND.LTFTYP.EQ.5) THEN
  WRITE(8,1003)
  IF (TU1.EQ.0 .AND. TU3.GT.0 .AND.TU2.GT.0.) THEN
    WRITE(8,1005)
    LTFTYP = 1
GA = GA/TU3
TU1 = TU2/TU3
TU2 = 0.
TU3 = 0.
WRITE (8, 1000)
WRITE (8, 1052) LTFID, LTFTYP, TU1, TU2, TU3, GA, X0, DX0, AA
+ , LABEL (LTFTYP)
GOTO 3
END IF
IF (TU3.EQ.0. .AND. TU1.GT.0. .AND. TU2.GT.0.) THEN
WRITE (8, 1006)
LTFTYP = 3
GA = GA/TU2
TU1 = TU1/TU2
TU2 = 0.
TU3 = 0.
WRITE (8, 1000)
WRITE (8, 1052) LTFID, LTFTYP, TU1, TU2, TU3, GA, X0, DX0, AA
+ , LABEL (LTFTYP)
GOTO 3
END IF
IF (IFLAG = 1)
GOTO 2
END IF
3 CONTINUE
IF (TU1.LE.0. .AND. LTFTYP.LE.4) THEN
IFLAG = 1
WRITE (8, 1003)
GO TO 2
END IF
IF (TU2.GT.0. .AND. LTFTYP.LE.3) THEN
TU2 = 0.
WRITE (8, 1003)
END IF
IF (LTFTYP.EQ.4 .AND. TU2.EQ.0. .AND. TU1.GT.0.) THEN
LTFTYP = 1
WRITE (8, 1003)
WRITE (8, 1005)
WRITE (8, 1000)
WRITE (8, 1052) LTFID, LTFTYP, TU1, TU2, TU3, GA, X0, DX0, AA
+ , LABEL (LTFTYP)
END IF
IF (TU1.EQ.TU2 .AND. LTFTYP.EQ.4) THEN
WRITE (8, 1007)
END IF
IF (TU3.GT.0. .AND. LTFTYP.LT.5) THEN
TU3 = 0.0
WRITE (8, 1003)
END IF
IF (GA.LE.0.) WRITE (8, 1008)
TAV = 1(LTFID) = TU1
SUBROUTINE TRNF(XIN,XOUT,T,KTFID)

C     THIS ROUTINE EVALUATES THE FOLLOWING TRANSFER FUNCTIONS
C     1ST ORDER LAG;
C     1ST ORDER LAG WITH DIFFERENTIATION;
C     1ST ORDER LAG WITH INTEGRATION;
C     COMBINED LEAD/LAG;
C     2ND ORDER LAG - HARMONIC OSCILLATOR.
C     XIN     DRIVING TERM.
C     XOUT    OUTPUT OF TRANSFER FUNCTION.
C     T       TIME.
C     KTFTYP  1 -> 1/(TAU1*S+1)
C             2 -> S/(TAU1*S+1)
C             3 -> 1/(S*(TAU1*S+1))
C             4 -> (TAU2*S+1)/(TAU1*S+1)
C             5 -> 1/(TAU1*S**2 + TAU2*S +TAU3)
C     KTFID   ID NO. OF CHANNEL (VARIABLE).
C     COMMON/TRNSF/TAU1(50),TAU2(50),TAU3(50),TFTIME(50),TFAUX(50),
C             + TFAUX2(50),KTFTYP(50),GAIN(50)
C     REAL KK,II
C     IF (KTFID.LT.1 .OR.KTFID.GT.50) GOTO 102
C     KTF = KTFTYP(KTFID)
C     IF (KTF.LT.1 .OR. KTF.GT.5) GOTO 103

TAU2(LTFID) = TU2
TAU3(LTFID) = TU3
TFAUX(LTFID) = X0
TFAUX2(LTFID) = DX0
KTFTYP(LTFID) = LTFTYP
GAIN(LTFID) = GA

2 WRITE(8,1053)
4 WRITE(8,1053)
   IF (IFLAG.NE.0) STOP
RETURN

1000 FORMAT( ' ***THIS TRANSFER FUNCTION WILL BE TREATED AS: ' )
1001 FORMAT( ' ***ERROR IN TRNF IO*** KTFTID OUT OF RANGE.' )
1002 FORMAT( ' ***ERROR IN TRNF IO*** THIS TRANSFER FUNCTION '
 + ', ALREADY DEFINED.' )
1003 FORMAT( ' ***ERROR IN TRNF IO*** TAU OUT OF RANGE.' )
1004 FORMAT( ' ***ERROR IN TRNF IO*** KTFTYP OUT OF RANGE.' )
1005 FORMAT( ' SUBSTITUTING 1ST ORDER LAG. KTFTYP -> 1.' )
1006 FORMAT( ' SUBSTITUTING 1ST ORDER LAG WITH INTEGRATOR. ',
 + ' KTFTYP -> 3.' )
1007 FORMAT( ' ***ERROR IN TRNF IO*** TAU1 = TAU2. TRANSFER FUNCTION',
 + ' DEGENRATES TO UNITY.' )
1008 FORMAT( ' ***ERROR IN TRNF IO*** GAIN OUT OF RANGE.' )
1050 FORMAT( ///,'1TRANSFER FUNCTIONS DEFINITIONS: ',', KTFID',
 + ' KTFTYP',6X,'TAU1',6X,'TAU2',6X,'TAU3',6X,'GAIN',9X,'X',
 + 8X,'DX',/)
1051 FORMAT(212,5X,6F6.0,5X,A30)
1052 FORMAT(217,6F10.5,/,' DESCRIPTION: ',A30,2X,A30)
1053 FORMAT(/)

END
XOUT = 0.
XTMP = 0.
XTMP2 = 0.

IF (TFTIME(KTFID).GT.T) GOTO 101
TDELTA = T - TFTIME(KTFID)
IF (TDELTA.EQ.0.) THEN
  TFEXP = 1.
  TFEXOT = 1./TAUL(KTFID)
  GOTO 1
END IF
TFEXP = EXP(-TDELTA/TAUL(KTFID))

C
FOLLOWING CODE ASSURES CORRECT LIMITING VALUES AS TAU -> 0.
IF (ABS(THAU1(KTFID)/TDELTA) .LT. 1.E-4) THEN
  TFEXOT = 1.0/TDELTA
ELSE
  TFEXOT = TFEXP/TAUL(KTFID)
END IF

1 DRIVE = XIN
OMEGA = DRIVE
KTF = KTFIDP(KTFID)
IF (KTF.EQ.4) OMEGA = (1.-TAU2(KTFID)/TAUL(KTFID))*
   + DRIVE
IF(KTF.EQ.1 .OR. KTF.EQ.4) XTMP =
   + TFAUX(KTFID)*TFEXP+(1.-TFEXP)*OMEGA
IF (KTF.EQ.1) XOUT = XTMP
IF (KTF.EQ.2) THEN
  XTMP = TFEXOT*(DRIVE-TFAUX2(KTFID)) + TFAUX(KTFID)*TFEXP
  XOUT = XTMP
  XTMP2 = DRIVE
END IF
IF(KTF.EQ.3) THEN
  TEMP = TFAUX2(KTFID)-DRIVE
  XTMP = TFAUX(KTFID)+TAU1(KTFID)*TEMP*(1.-TFEXP)+DRIVE*TDELTA
  XOUT = XTMP
  XTMP2 = TEMP*TFEXP+DRIVE
END IF
IF (KTF.EQ.4) XOUT = DRIVE*(TAU2(KTFID)/TAUL(KTFID))
   + + XTMP
IF (KTF.EQ.5) THEN
  KK = TAU3(KTFID)
  II = TAU1(KTFID)
  BETA = TAU2(KTFID)
  FORCE = DRIVE
  DELZ = TFAUX(KTFID)
  DELDZ = TFAUX2(KTFID)
  TLAST = TFTIME(KTFID)
  CALL SOLAG(II,BETA, KK, DELZ, DELDZ, FORCE, T, TLAST, DELT, DELTD)
  XTMP = DELT
  XTMP2 = DELTD
  XOUT = XTMP
END IF

21
C ABOVE SAVES XD FOR 2ND ORDER LAG/Oscillator AND DX/DT FOR C FIRST ORDER LAG WITH DIFFERENTIATOR. FOR USE AS INITIAL C CONDITIONS ON NEXT INTEGRATION STEP.

RETURN

WRITE(8,200) T,TFTIME(KTFID),KTFID
STOP

WRITE(8,201) KTFID
STOP

WRITE(8,202) KTF
STOP

200 FORMAT(' ***ERROR*** IN TRNF. TIME=',F10.5,' IS GREATER THAN ', + 'LAST TIME=',F10.5,' FOR CHANNEL',I3,/)  
201 FORMAT(' ***ERROR*** IN TRNF. CHANNEL NO. KTFID=',I4, + ' IS OUT OF RANGE.'/)  
202 FORMAT(' ***ERROR*** IN TRNF. TYPE CODE NUMBER KTFID=',I4, + ' IS OUT OF RANGE.'/)  
END

SUBROUTINE SOLAG(II,BETA,KK,DELZ,DELDZ,FORCE,T,TLAST,DELT, + DELTD)

C
C DIFFERENTIAL EQUATION
C II*YDDOT + BETA*YDOT + KK*Y = FORCE
C
C NOTE: THE CALLING ROUTINE MUST SAVE TLAST,DELZ AND DELDZ
C FOR THE NEXT CALL TO SOLAG

REAL NUM ,LAMB1, LAMB2, LAMB, II, KK

IF (BETA.LT.0. .OR. KK.LE.0.) THEN
  WRITE(8,1000) BETA,KK
  STOP
ENDIF

IF (II.LT.0.) THEN
  WRITE (8,1001) II
  STOP
ENDIF

BOV2I = BETA/(2.*II)
RDCND = BOV2I**2 -KK/II
FOVK  = FORCE/KK

IF(RDCND .GT. 0.)GO TO 100
IF(RDCND .EQ. 0.)GO TO 120

C OSCILLATORY SOLUTION
LAMB =-BOV2I
OMEGA = SQRT((-RDCND))
NUM  = OMEGA*(DELZ-FOVK)
DENOM = DELDZ-LAMB*(DELZ-FOVK)
LAMB = -BOV2I

IF(NUM .EQ. 0. .AND. DENOM .EQ. 0.) GO TO 20
PHI = ATAN2(NUM,DENOM) - OMEGA*TLAST
GO TO 30

20 CONTINUE
PHI = 0.
30 CONTINUE
  ETEMP = EXP(-LAMB*TLAST)
  IF(DELZ - FOVK .EQ. 0.) GO TO 40
  A = (DELZ - FOVK) * ETEMP / SIN(OMEGA*TLAST + PHI)
  GO TO 50
40 CONTINUE
  A = DELDZ * ETEMP / (LAMB * SIN(OMEGA*TLAST + PHI)
  + OMEGA * COS(OMEGA*TLAST + PHI))
50 CONTINUE
  FTEMP = EXP(LAMB*T)
  DELT = A * FTEMP * SIN(OMEGA*T + PHI) + FOVK
  DELTD = LAMB*A*FTEMP*SIN(OMEGA*T + PHI)
  + OMEGA*A*FTEMP*COS(OMEGA*T + PHI)
  GO TO 150
100 CONTINUE
C EXPONENTIAL SOLUTION
  LAMB1 = -B0V2I + SQRT(RDCND)
  LAMB2 = -B0V2I - SQRT(RDCND)
  A = (DELZ - DELDZ/LAMB2 - FOVK) * EXP(-LAMB1*TLAST)
  + (1. - LAMB1/LAMB2)
  B = (DELZ - DELDZ/LAMB1 - FOVK) * EXP(-LAMB2*TLAST)
  + (1. - LAMB2/LAMB1)
  DELT = A*EXP(LAMB1*T) + B*EXP(LAMB2*T) + FOVK
  DELTD = LAMB1*A*EXP(LAMB1*T) + LAMB2*B*EXP(LAMB2*T)
  GO TO 150
120 CONTINUE
C CRITICALLY DAMPED SOLUTION
  LAMB = -B0V2I
  A = ((DELZ - FOVK) * (1 + TLAST*LAMB) - TLAST*DELDZ) * EXP(-LAMB*TLAST)
  B = (DELDZ + LAMB * (FOVK - DELZ)) * EXP(-LAMB*TLAST)
  DELT = (A + B*T) * EXP(LAMB*T) + FOVK
  DELTD = (LAMB*A + B + LAMB*B*T) * EXP(LAMB*T)
150 CONTINUE
  TLAST = T
  DELZ = DELT
  DELDZ = DELTD
RETURN
1000 FORMAT(/,' ***-ERR IN SOLAG-*** BETA=',E12.4,',KK=',E12.4,/, "+ ' ONLY POSITIVE DEFINITE VALUES ALLOWED.','.1)
1001 FORMAT(/,' ***-ERR IN SOLAG-*** II=',E12.4,/, "+ ' NEGATIVE VALUES NOT ALLOWED.','.1)
END
SAMPLE DATA INPUT CARDS.

<table>
<thead>
<tr>
<th>K</th>
<th>K</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>G</th>
<th>X</th>
<th>D</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>X</td>
<td>X</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>I</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>I</td>
<td>T</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>N</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>Y</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>P</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>.02</td>
<td>0.</td>
<td>0.</td>
<td>4.0</td>
<td>0.0</td>
<td>0.</td>
<td>PITCH ATTITUDE HOLD</td>
<td></td>
</tr>
<tr>
<td>0202</td>
<td>.2</td>
<td>0.</td>
<td>0.</td>
<td>2.0</td>
<td>0.0</td>
<td>0.</td>
<td>PITCH SYNTHETIC DAMPING</td>
<td></td>
</tr>
<tr>
<td>0303</td>
<td>.02</td>
<td>0.</td>
<td>0.</td>
<td>2.0</td>
<td>0.0</td>
<td>0.</td>
<td>YAW ATTITUDE HOLD</td>
<td></td>
</tr>
<tr>
<td>0404</td>
<td>.02</td>
<td>.01</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.</td>
<td>YAW LOS RATE FILTER</td>
<td></td>
</tr>
<tr>
<td>0504</td>
<td>.02</td>
<td>0.</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.</td>
<td>YAW LOS RATE FILTER</td>
<td></td>
</tr>
<tr>
<td>0605</td>
<td>1.</td>
<td>3.</td>
<td>.25</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td>PURE DAMPING</td>
<td></td>
</tr>
<tr>
<td>0705</td>
<td>1.</td>
<td>1.</td>
<td>.25</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td>CRITICALLY DAMPED</td>
<td></td>
</tr>
<tr>
<td>0805</td>
<td>1.</td>
<td>.5</td>
<td>.25</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td>DAMPED OSCILLATION</td>
<td></td>
</tr>
<tr>
<td>0905</td>
<td>1.</td>
<td>.0</td>
<td>.25</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td>PURE OSCILLATION</td>
<td></td>
</tr>
<tr>
<td>1005</td>
<td>0.</td>
<td>.5</td>
<td>.25</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1105</td>
<td>1.</td>
<td>.5</td>
<td>.0</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1205</td>
<td>.001</td>
<td>.0002</td>
<td>20.</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1305</td>
<td>.0002</td>
<td>0.0</td>
<td>5000.</td>
<td>1.0</td>
<td>10.0</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

24
SAMPLE OUTPUT

TRANSFER FUNCTIONS DEFINITIONS:

<table>
<thead>
<tr>
<th>KTFID</th>
<th>KTFTYP</th>
<th>TAU1</th>
<th>TAU2</th>
<th>TAU3</th>
<th>GAIN</th>
<th>X</th>
<th>DX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.02000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>4.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: PITCH ATTITUDE HOLD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.20000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>2.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: PITCH SYNTHETIC DAMPING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.02000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>2.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: YAW ATTITUDE HOLD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.02000</td>
<td>0.01000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: YAW LOS RATE FILTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.02000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: YAW LOS RATE FILTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em><strong>ERROR IN TRNF 10</strong></em> TAU OUT OF RANGE. SUBSTITUTING 1ST ORDER LAG. KTFTYP -&gt; 1. ***THIS TRANSFER FUNCTION WILL BE TREATED AS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.02000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: YAW LOS RATE FILTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1.00000</td>
<td>3.00000</td>
<td>0.25000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: PURE DAMPING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.25000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: CRITICALLY DAMPED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1.00000</td>
<td>0.50000</td>
<td>0.25000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: DAMPED OSCILLATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESCRIPTION: PURE OSCILLATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0000</td>
<td>0.5000</td>
<td>0.2500</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td><em><strong>ERROR IN TRNF 10</strong></em> TAU OUT OF RANGE.</td>
<td><strong>SUBSTITUTING 1ST ORDER LAG.</strong> KTFTYP → 1.</td>
<td>***THIS TRANSFER FUNCTION WILL BE TREATED AS:</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GAIN/(TAU1<em>S**2+TAU2</em>S+TAU3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>1.0000</td>
<td>0.5000</td>
<td>0.0000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td><em><strong>ERROR IN TRNF 10</strong></em> TAU OUT OF RANGE.</td>
<td><strong>SUBSTITUTING 1ST ORDER LAG WITH INTEGRATOR.</strong> KTFTYP → 3.</td>
<td>***THIS TRANSFER FUNCTION WILL BE TREATED AS:</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GAIN/(S*(TAU1*S+1))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>0.0010</td>
<td>0.0020</td>
<td>20.0000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GAIN/(TAU1<em>S**2+TAU2</em>S+TAU3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>0.0002</td>
<td>0.00005000</td>
<td>0.0000</td>
<td>1.00000</td>
<td>10.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GAIN/(TAU1<em>S**2+TAU2</em>S+TAU3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**THE INTEGRATION TIME STEP DT = 0.005000.**

<table>
<thead>
<tr>
<th>LTFID</th>
<th>TIME</th>
<th>DRIVE</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>1</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-5.53438</td>
</tr>
<tr>
<td>1</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>-6.71729</td>
</tr>
<tr>
<td>1</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-0.20403</td>
</tr>
<tr>
<td>1</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>-4.44207</td>
</tr>
<tr>
<td>1</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>-7.42804</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LTFID</th>
<th>TIME</th>
<th>DRIVE</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>2</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-3.67251</td>
</tr>
<tr>
<td>2</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>1.68669</td>
</tr>
<tr>
<td>2</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>2.25946</td>
</tr>
<tr>
<td>2</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>-3.56729</td>
</tr>
<tr>
<td>2</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>0.70957</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LTFID</th>
<th>TIME</th>
<th>DRIVE</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>3</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-1.04245</td>
</tr>
<tr>
<td>3</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>-4.69789</td>
</tr>
<tr>
<td>3</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-6.27757</td>
</tr>
<tr>
<td>3</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>-6.97196</td>
</tr>
<tr>
<td>3</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>-10.47893</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LTFID</th>
<th>TIME</th>
<th>DRIVE</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>4</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-1.39987</td>
</tr>
<tr>
<td>4</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>-1.66648</td>
</tr>
<tr>
<td>4</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-0.4542</td>
</tr>
<tr>
<td>4</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>-1.12801</td>
</tr>
<tr>
<td>4</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>-1.84804</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LTFID</th>
<th>TIME</th>
<th>DRIVE</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>5</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-1.38360</td>
</tr>
<tr>
<td>5</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>-1.67932</td>
</tr>
<tr>
<td>5</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-0.05101</td>
</tr>
<tr>
<td>5</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>-1.11052</td>
</tr>
<tr>
<td>5</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>-1.85701</td>
</tr>
<tr>
<td>LTFID</td>
<td>TIME</td>
<td>DRIVE</td>
<td>X</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>0.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>6</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>9.35272</td>
</tr>
<tr>
<td>6</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>8.09988</td>
</tr>
<tr>
<td>6</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>7.04001</td>
</tr>
<tr>
<td>6</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>6.35853</td>
</tr>
<tr>
<td>6</td>
<td>5.0000</td>
<td>-1.83907</td>
<td>5.38481</td>
</tr>
<tr>
<td>7</td>
<td>0.0000</td>
<td>0.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>7</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>8.97774</td>
</tr>
<tr>
<td>7</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>6.33926</td>
</tr>
<tr>
<td>7</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>3.54739</td>
</tr>
<tr>
<td>7</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>1.75784</td>
</tr>
<tr>
<td>7</td>
<td>5.0000</td>
<td>-1.83907</td>
<td>1.0284</td>
</tr>
<tr>
<td>8</td>
<td>0.0000</td>
<td>0.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>8</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>8.82387</td>
</tr>
<tr>
<td>8</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>5.36682</td>
</tr>
<tr>
<td>8</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>1.15018</td>
</tr>
<tr>
<td>8</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>-1.86486</td>
</tr>
<tr>
<td>8</td>
<td>5.0000</td>
<td>-1.83907</td>
<td>-4.23518</td>
</tr>
<tr>
<td>9</td>
<td>0.0000</td>
<td>0.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>9</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>8.62980</td>
</tr>
<tr>
<td>9</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>3.87712</td>
</tr>
<tr>
<td>9</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>-3.25255</td>
</tr>
<tr>
<td>9</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>-9.90176</td>
</tr>
<tr>
<td>9</td>
<td>5.0000</td>
<td>-1.83907</td>
<td>-15.20981</td>
</tr>
<tr>
<td>10</td>
<td>0.0000</td>
<td>0.0000</td>
<td>40.0000</td>
</tr>
<tr>
<td>10</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>23.29664</td>
</tr>
<tr>
<td>10</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>11.23015</td>
</tr>
<tr>
<td>10</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>5.73197</td>
</tr>
<tr>
<td>10</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>2.81734</td>
</tr>
<tr>
<td>10</td>
<td>5.0000</td>
<td>-1.83907</td>
<td>-1.12071</td>
</tr>
<tr>
<td>11</td>
<td>0.0000</td>
<td>0.0000</td>
<td>20.0000</td>
</tr>
<tr>
<td>11</td>
<td>1.0000</td>
<td>-1.41615</td>
<td>19.86679</td>
</tr>
<tr>
<td>11</td>
<td>2.0000</td>
<td>-1.65364</td>
<td>18.71997</td>
</tr>
<tr>
<td>11</td>
<td>3.0000</td>
<td>-0.03983</td>
<td>16.91363</td>
</tr>
<tr>
<td>11</td>
<td>4.0000</td>
<td>-1.14550</td>
<td>15.57969</td>
</tr>
<tr>
<td>LTFID</td>
<td>TIME</td>
<td>DRIVE</td>
<td>X</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>12</td>
<td>0.00000</td>
<td>0.00000</td>
<td>10.00000</td>
</tr>
<tr>
<td>12</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>-9.10857</td>
</tr>
<tr>
<td>12</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>8.06504</td>
</tr>
<tr>
<td>12</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-7.32883</td>
</tr>
<tr>
<td>12</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>6.51484</td>
</tr>
<tr>
<td>12</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>-5.97223</td>
</tr>
<tr>
<td>13</td>
<td>0.00000</td>
<td>0.00000</td>
<td>10.00000</td>
</tr>
<tr>
<td>13</td>
<td>1.00000</td>
<td>-1.41615</td>
<td>1.54640</td>
</tr>
<tr>
<td>13</td>
<td>2.00000</td>
<td>-1.65364</td>
<td>-9.52188</td>
</tr>
<tr>
<td>13</td>
<td>3.00000</td>
<td>-0.03983</td>
<td>-4.49206</td>
</tr>
<tr>
<td>13</td>
<td>4.00000</td>
<td>-1.14550</td>
<td>8.13177</td>
</tr>
<tr>
<td>13</td>
<td>5.00000</td>
<td>-1.83907</td>
<td>7.00721</td>
</tr>
</tbody>
</table>
APPENDIX B

PROGRAM USAGE
The program consists of three subroutines (TRNF 10, TRNF, and SOLAG) and a main program, TFMAIN.* TRNF 10 is the initialization routine. The data records that define the transfer functions are read by this routine. (See the explanation for input data format given below.) TRNF performs the analytical transfer function simulations. It accesses subroutine SOLAG for treatment of the second order lag/harmonic oscillator. The driving main program TFMAIN is supplied for demonstration and testing only. Use it as a model for interfacing TRNF 10, TRNF, and SOLAG with your computer program. The initial time is at line 100, and the integration time step is at line 200. The driving term is defined in the lines following 300.

The call to TRNF contains four arguments:

- **DRIVE**: Driving function
- **XOUT**: Output of the transfer function channel
- **T**: Time
- **J**: Channel I.D. number LTFID

One input data record is required for each transfer function variable to be integrated. These records have the following form:

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I2</td>
<td>LTFID</td>
<td>I.D number of channel; values are from 1 to 50</td>
</tr>
<tr>
<td>3</td>
<td>I2</td>
<td>LTFTYP</td>
<td>Code for type of transfer function; values are 1 to 5</td>
</tr>
<tr>
<td>10</td>
<td>F9.0</td>
<td>TAU1</td>
<td>First constant, for all LTFTYP</td>
</tr>
<tr>
<td>19</td>
<td>F9.0</td>
<td>TAU2</td>
<td>Second constant, for LTFTYP = 4 or 5</td>
</tr>
<tr>
<td>28</td>
<td>F9.0</td>
<td>TAU3</td>
<td>Third constant, for LTFTYP=5</td>
</tr>
<tr>
<td>37</td>
<td>F9.0</td>
<td>XO</td>
<td>Initial condition for variable</td>
</tr>
<tr>
<td>46</td>
<td>F9.0</td>
<td>DXO</td>
<td>Initial condition for derivative of variable</td>
</tr>
<tr>
<td>55</td>
<td>A26</td>
<td>AA</td>
<td>Alphanumeric label</td>
</tr>
</tbody>
</table>

LTFID is an integer variable from 1 to 50 that is associated with each transfer function channel (output variable) to be simulated. For example, a lag

---

* A listing of a program implementing these techniques and a sample run are in appendix A.
might be applied to both a fin yaw deflection command and to a fin pitch deflection command. Each of these two channels is defined as a separate data input record and is assigned an identifying integer LTFID. If the user chooses an integer that was previously assigned on another input data record, execution terminates with an explanatory message.

LTFTYP is an integer variable in the range 1 to 5 that defines the type of transfer function:

<table>
<thead>
<tr>
<th>LTFTYP</th>
<th>Description and FORTRAN code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First-order lag (1/(\tau_1 s + 1))</td>
</tr>
<tr>
<td>2</td>
<td>First-order lag with differentiator (s/(\tau_1 s + 1))</td>
</tr>
<tr>
<td>3</td>
<td>First-order lag with integrator (1/[(s\tau_1 + 1)])</td>
</tr>
<tr>
<td>4</td>
<td>Combined lead/lag ((\tau_2 s + 1)/ (\tau_1 s + 1))</td>
</tr>
<tr>
<td>5</td>
<td>Second-order lag/harmonic oscillator (1/((\tau_1 s^2 + \tau_2 s + \tau_3))</td>
</tr>
</tbody>
</table>

\(\tau_1\) is the first constant (see LTFTYP above). All five transfer function types (LTFTYP = 1 to 5) require this quantity to be positive.

\(\tau_2\) is the second constant. Lead/lag and second-order lag/oscillator transfer functions (LTFTYP = 4, 5) require a positive value.

\(\tau_3\) is the third constant. The second-order lag/oscillator (LTFTYP = 5) requires a positive value.

\(x_0\) is the initial condition on the output variable, i.e., \(y_0 = y\) at \(t = t_0\).

\(dx_0\) is the initial condition on the derivative of the output variable, \(y' = dy/dt\) at \(t = t_0\).

AA is an alphanumeric label of 26 characters that identifies the nature of the transfer function when an echo of the input data is printed out by TRNF IO. Examples might be "PITCH SYNTHETIC DAMPING" or "YAW ATTITUDE HOLD."

If an input variable is out of range, the program prints an error message and reformulates the transfer function into an equivalent form. For example, if an attempt is made to run an oscillator (LTFTYP = 5) with \(\tau_1 = 0\), subroutine TRNF IO will substitute the equivalent simple lag (LTFTYP = 1), as can be seen in the sample output for KTFID = 10. Two other examples appear in the sample output in the sample execution, i.e., for KTFID = 5 and KTFID = 11.
SYMBOLS
Algebraic Expressions

A  Constant of integration
B  Constant of integration
D  Coefficient of the Laplace operator $s$ in the transfer function of the second-order lag/oscillator
I  Coefficient of the square of the Laplace operator $s$ in the transfer function of the second-order lag/oscillator
K  Constant term in the transfer function of the second-order lag/oscillator
L  Damping coefficient $-D/2I$ for the second-order lag/oscillator
s  Laplace transform operator
t  Time, independent variable
T  Time constant
D  Driving term
y  Output of transfer function, dependent variable
$y_0$  Initial condition on $y$ at $t = t_0$
$y'$  Initial condition on $dy/dt$ at $t = t_0$
z  Auxiliary dependent variable for first-order lag with differentiation or for first-order lag with integration
$\omega$  Frequency of oscillation for second-order lag/oscillator
$\phi$  Phase angle of oscillation for second-order lag/oscillator
FORTRAN Variable Names

AA  Alphanumeric label for printout (80 characters)
DX  Initial condition on derivative when KTFTYP=5
KTFID  I.D. number of channel or variable (range is 1 to 50)
KTFTYP  Integer variable used to indicate type of transfer function (range 1 to 50)

KTFTYP = 1 is first-order lag
   Transfer function:  GAIN/(TAU1*S+1)

KTFTYP = 2 is first-order lag with differentiation
   Transfer function:  GAIN*S/(TAU1*S+1)

KTFTYP = 3 is first-order lag with integration
   Transfer function:  GAIN/[S*(TAU1*S+1)]

KTFTYP = 4 is combined lead/lag
   Transfer function:  GAIN*(TAU2*S+1)/(TAU1*S+1)

KTFTYP = 5 is second-order lag/harmonic oscillator
   Transfer function:  GAIN/(TAU1*S**2 + TAU2*S +TAU3)

T  Time
TAU1  First time constant
TAU2  Second time constant (T₁ for KTFTYP=4 or D for KTFTYP)
TAU3  Third time constant (K for KTFTYP=5)
X  Initial condition on variable
XIN  Driving term
XOUT  Output of transfer function
DISTRIBUTION LIST

Department of the Army
ATTN: DAMA-CSM
   DAMO-RQS
Washington, DC  20314

Commander
U.S. Army Materiel Command
ATTN: AMCDE-DM
5001 Eisenhower Avenue
Alexandria, VA 22304

Commander
Armament Research and Development Center
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCAR-TSS (5)
   SMCAR-LC, COL A. T. Crumpton
      T. Davidson
   SMCAR-LCS, J. Gregorits
   SMCAR-LCA, L. Ambrosini
      L. Marino
      F. Scerbo
   SMCAR-LCA-F, A. Loeb
      D. Mertz (10)
      R. Kline (3)
      E. Friedman (3)
      M. Amoruso (3)
   SMCAR-LCU, B. Bushey
Dover, NJ  07801-5001

Commander
U.S. Army Armament, Munitions and Chemical Command
ATTN: AMSMC-GCL(D)
Dover, NJ 07801-5001

Administrator
Defense Technical Information Center
ATTN: Accessions Division (12)
Cameron Station
Alexandria, VA 22304-6145

Director
U.S. Army Materiel Systems Analysis Activity
ATTN: AMXSY-MP
Aberdeen Proving Ground, MD 21005-5066