W. W. Hansen Experimental Physics Laboratory
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305 - 4085

Gravity Probe B Relativity Mission

Gravity Probe B Science Data Analysis: Covariance Matrix for Measurement Statistical Errors

S0959, Rev. -
November, 2003

Prepared by:
M. Heifetz — 11/10/2003 Date

Approved by:
G. M. Keiser — 11/10/03 Date

Approved by:
D.Ross
QA — 11/10/03 Date

Approved by:
R. Whelan
Systems Engineering — 11/10/03 Date

ITAR Assessment Performed Tom Langenstein 11.10.03

ITAR Control Req'd? ☐ Yes ☑ No

1. Introduction

The purpose of this document is to describe the Matlab program **CovarianceAnalysis.m** that has been developed to calculate statistical measurement errors in the drift rates of the GP-B gyroscopes.

The statistical measurement errors associated with parameters (or states) in the measurement model may be calculated using a covariance analysis as long as the data analysis model accurately reflects the physical processes and the statistical characteristics of the noise in measurements are known.

A. Gravity Probe B Core Measurement Model

The gyroscope readout system and the telescope readout system may be combined as described in reference [1]. This combined signal may then be used to determine the orientation and drift rate of each gyroscope relative to the true direction to the guide star. The statistical error in the gyroscope drift rate may be determined from the covariance matrix. The minimum set of parameters which must be used in the data analysis consist of

- the gyroscope drift rate relative to the guide star in the East-West and North-South directions, R_{EW} and R_{NS},
- the misalignment of the gyroscope spin axis at time $t = t_0$ in the East-West and North-South directions, EW_0 and NS_0,
- the scale factor of the gyroscope readout, C_G, and the
- the roll phase offset, $\delta\phi$.

With these parameters, the model for the combined gyroscope and telescope signals has the following structure,

$$z(t) = C_G \begin{bmatrix} (NS_0 + R_{NS}(t-t_0) + A_o\cos\phi_o + l_1\cos\phi_a + l_2\sin\phi_a)\cos(\phi_m + \delta\phi) + \\ + (EW_0 + R_{EW}(t-t_0) + l_3\cos\phi_a + l_4\sin\phi_a)\sin(\phi_m + \delta\phi) \end{bmatrix} + \text{bias} + \text{noise} \quad (1)$$

where A_0 and l_1, l_2, l_3, l_4 are respectively the amplitude of the orbital aberration signal and the components of the annual aberration signal. Explicit expressions for these constants, l_1, l_2, l_3, and l_4, are given in reference [2]. These constants depend on the location of the reference star, the tilt of the Earth's rotation axis with respect to the ecliptic, and the phase of Earth's orbital motion at the midpoint of the data acquisition period. The measured roll phase, ϕ_m, is taken to be zero when the satellite's x-axis passes through the plane which contains the Earth's rotation axis and the direction to the guide star. The orbital phase, ϕ_o, is defined to be zero when the satellite is closest to the guide star, the annual phase is defined to be zero at the midpoint of the data acquisition period, and t_0 is taken to be the time at the middle of the data acquisition period.

B. Measurement Models used in the Covariance Analysis

All measurement models explored in the Covariance analysis program are based on the core structure (1). The bias is assumed to be constant in all models and can be excluded from the state-vector.

Model 0 (basic model)

- Assumption: the scale factor of the gyroscope readout, C_G, and the the roll phase offset, $\delta\phi$, are constant during all the science data collection time (initial misalignments, NS_0 and EW_0, and the gyroscope drift rates, R_{EW} and R_{NS} are constant parameters according to their definitions).

Estimated Parameters (state-vector): C_G, $\delta\varphi$, NSo, EWo, R_{NS}, R_{EW},

Model 1

- Assumption:
  - the scale factor of the gyroscope readout, C_G, changes linearly in time: $C_G = C_{G0} + C_{Gt} * t/Ta$, (Ta is one year period);
  - the roll phase offset, $\delta\phi$, stays constant.

Estimated Parameters (state-vector): C_{G0}, $\delta\varphi$, NSo, EWo, R_{NS}, R_{EW}, C_{Gt}

Model 2

- Assumption:
  - the roll phase offset, $\delta\phi$, varies at the annual frequency:

    $$\delta\phi = \delta\phi_0 + \delta\phi_{ca}*\cos(wa*t) + \delta\phi_{sa}*\sin(wa*t)$$

  - the scale factor C_G remains constant: $C_G = C_{G0}$.

Estimated Parameters (state-vector): C_{G0}, $\delta\varphi_0$, NSo, EWo, R_{NS}, R_{EW}, $\delta\phi_{ca}$, $\delta\phi_{sa}$.

Model 3

- Assumption:
  - the scale factor, C_G, varies linearly in time and has an annual (w_a) harmonic component:

    $$C_G = C_{G0} + C_{Gt} * t/Ta + C_{Gca}*\cos(w_a*t) + C_{Gsa}*\sin(w_a*t);$$

- the roll phase offset, $\delta\phi$, varies at the annual frequency:

  $\delta\phi = \delta\phi_0 + \delta\phi_{ca}*\cos(w_a*t) + \delta\phi_{sa}*\sin(w_a*t)$.

Estimated Parameters (state-vector): C_{G0}, $\delta\varphi_0$, NSo, EWo, R_{NS}, R_{EW}, $\delta\phi_{ca}$, $\delta\phi_{sa}$, C_{Gt}, C_{Gca}, C_{Gsa}.

Model 4

- Assumption:

  - the scale factor, C_G, varies linearly in time and has both the orbital (w_o) and annual (w_a) harmonic components:

    $C_G = C_{G0} + C_{Gt} * t/Ta + C_{Gco}*\cos(w_o*t) + C_{Gso}*\sin(w_o*t) + C_{Gca}*\cos(w_a*t) + C_{Gsa}*\sin(w_a*t)$;

  - the roll phase offset, $\delta\phi$, varies at the annual frequency:

    $\delta\phi = \delta\phi_0 + \delta\phi_{ca}*\cos(w_a*t) + \delta\phi_{sa}*\sin(w_a*t)$.

Estimated Parameters (state-vector): C_{G0}, $\delta\varphi_0$, NSo, EWo, R_{NS}, R_{EW}, $\delta\phi_{ca}$, $\delta\phi_{sa}$, C_{Gt}, C_{Gco}, C_{Gso}, C_{Gco}, C_{Gso}.

2. **Calculation of the Covariance matrix**

Calculation of the covariance matrix is carried out according to the two-step filter methodology (see reference [2]):

a) for the state-vector X of the estimated parameters the first step state-vector Y = f(X) is introduced, such that the measurement equation (1) becomes a linear one: z(t) = H(t)*Y + noise;

b) the information matrix of the first step is calculated:

$$I_Y = \sum_{i=1}^{N} (H_i^T \sigma_i^{-2} H_i);$$

c) the Jacobian dY/dX of the transformation Y = f(X) is calculated (this is performed by the Matlab function **GP-B Jacobian**);

d) the information matrix of the first step is calculated:

$$I_X = (dY / dX)^T I_Y (dY / dX);$$

e) the covariance matrix of the original (second step) state-vector X is determined:

$$P_X = I_Y^{-1}.$$

The table below shows the dimension of the first step state-vector (Y) and second step state-vector (X) for different models used in calculations:

Model	dim X	dim Y
0	6	10
1	7	18
2	8	18
3	11	21
4	13	33

The dimension of the first step state-vector Y represents the number of various time signatures that are present in the analyzed signal z(t), given the structure of estimated parameters.

Below, for all four models used in the calculations, we present the state-vectors X, Y, and the corresponding time-signatures.

Model 0.

```
% Model 0: Cg, Dfi - const (Cg = Cg0, Dfi = Dfi0)
%
% State Vector          First Step States                               Time Signature
%
%   x(1) = Cg           y(1)= Cg*(NS*cos(Dfi)+EW*sin(Dfi))              cos(wr*t)
%   x(2) = deltaphi (Dfi) y(2)= Cg*(RgTa*cos(Dfi)+ RmTa*sin(Dfi))       (t/Ta)*cos(wr*t)
%   x(3) = NSo          y(3)= Cg*(11*cos(Dfi) + 13*sin(Dfi))            cos(wr*t)*cos(wa*t)
%   x(4) = EWo          y(4)= Cg*(12*cos(Dfi) + 14*sin(Dfi))            cos(wr*t)*sin(wa*t)
%   x(5) = Rg*Ta        y(5)= Cg*Ao*cos(Dfi)                            cos(wr*t)*cos(wo*t)
%   x(6) = Rm*Ta        y(6)= Cg*(-NS*sin(Dfi)+EW*cos(Dfi))             sin(wr*t)
%                       y(7)= Cg*(-RgTa*sin(Dfi)+ RmTa*cos(Dfi))        (t/Ta)*sin(wr*t)
%                       y(8)= Cg*(-11*sin(Dfi) + 13*cos(Dfi))           sin(wr*t)*cos(wa*t)
%                       y(9)= Cg*(-12*sin(Dfi) + 14*cos(Dfi))           sin(wr*t)*sin(wa*t)
%                       y(10)= -Cg*Ao*sin(Dfi)                          sin(wr*t)*cos(wo*t
```

Model 1.

```
%Model 1 (21): Cg = Cg0 + Cg_t*t/Ta ,
%
% State Vector      Time Signature              First Step States
%  x(1) = Cg0       cos(wr*t)                   y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(2) = Dfi0      (t/Ta)*cos(wr*t)            y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%  x(3) = NSo                                         Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(4) = EWo       cos(wr*t)*cos(wa*t)         y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0))
%  x(5) = Rg*Ta     cos(wr*t)*sin(wa*t)         y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0))
%  x(6) = Rm*Ta     cos(wr*t)*cos(wo*t)         y(5)= Cg0*Ao*cos(Dfi0)
%  x(7) = Cg_t      sin(wr*t)                   y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   (t/Ta)*sin(wr*t)            y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0)) +
%                                                     Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   sin(wr*t)*cos(wa*t)         y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0))
%                   sin(wr*t)*sin(wa*t)         y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0))
%                   sin(wr*t)*cos(wo*t)         y(10)= -Cg0*Ao*sin(Dfi0)
%                 (t/Ta)^2*cos(wr*t)            y(11) = Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*cos(wa*t)    y(12) = Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*sin(wa*t)    y(13)= Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*cos(wo*t)    y(14)= Cg_t*Ao*cos(Dfi0)
%                 (t/Ta)^2*sin(wr*t)            y(15)= Cg_t*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wa*t)    y(16) = Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*sin(wa*t)    y(17)= Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wo*t)    y(18)= -Cg_t*Ao*sin(Dfi0)
%
```

Model 2.

```
% Model 2 (32): Cg = Cg0;
%       Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
% State Vector      Time Signature              First Step States
%  x(1) = Cg0       cos(wr*t)                   y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%  x(2) = Dfi0                                        (1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) +
%  x(3) = NSo                                         (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0)+l4*cos(Dfi0))
%  x(4) = EWo       (t/Ta)*cos(wr*t)            y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%  x(5) = Rg*Ta     cos(wr*t)*cos(wa*t)         y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%  x(6) = Rm*Ta                                       Cg0*Dfi_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%  x(7) = Dfi_ca    cos(wr*t)*sin(wa*t)         y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%  x(8) = Dfi_sa                                      Cg0*Dfi_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   cos(wr*t)*cos(wo*t)         y(5)= Cg0*Ao*cos(Dfi0)
%                   sin(wr*t)                   y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) –
%                                                     (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) –
%                                                     (1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))
%                   (t/Ta)*sin(wr*t)            y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))
%                   sin(wr*t)*cos(wa*t)         y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) –
%                                                     Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                   sin(wr*t)*sin(wa*t)         y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) –
%                                                     Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                   sin(wr*t)*cos(wo*t)         y(10)= -Cg0*Ao*sin(Dfi0)
%                 (t/Ta)*cos(wr*t)*cos(wa*t)        y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wa*t)        y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 cos(wr*t)*cos(wo*t)*cos(wa*t)     y(13)= Cg0*Dfi_ca*Ao*sin(Dfi0)
%                 sin(wr*t)*cos(wo*t)*cos(wa*t)     y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0)
%                 (t/Ta)*cos(wr*t)*sin(wa*t)        y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*sin(wa*t)        y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 cos(wr*t)*cos(wo*t)*sin(wa*t)     y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0)
%                 sin(wr*t)*cos(wo*t)*sin(wa*t)     y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0)
```

Model 3.

```
%Model 3 (1): Cg = Cg0 + Cg_t*(t/Ta) + Cg_ca*cos(wa*t) + Cg_sa*sin(wa*t)
%             Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
% State Vector     Time Signature              First Step States

% x(1) = Cg0       cos(wr*t)                  y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +(1/2)*Cg0*Dfi_ca*
% x(2) = Dfi0                                        (l1*sin(Dfi0)+l3*cos(Dfi0)) + (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0) +
% x(3) = NSo                                         l4*cos(Dfi0))+(1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0))+
% x(4) = EWo                                         (1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))+(1/2)*Ao*Cg_co*Cos(Dfi0)
% x(5) = Rg*Ta     (t/Ta)*cos(wr*t)           y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
% x(6) = Rm*Ta                                       Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                  cos(wr*t)*cos(wa*t)        y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(NS*sin(Dfi0) +
% x(7) = Dfi_ca                                      EW*cos(Dfi0)) + Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
% x(8) = Dfi_sa    cos(wr*t)*sin(wa*t)        y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(NS*sin(Dfi0) +
%
% x(9) = Cg_t                                        EW*cos(Dfi0)) + Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
% x(10) = Cg_ca    cos(wr*t)*cos(wo*t)        y(5)= Cg0*Ao*cos(Dfi0)
% x(11) = Cg_sa    sin(wr*t)                  y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) –
%                                                    (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) –
%                                                    (1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0)) +
%                                                    (1/2)*Cg_ca*(l3*cos(Dfi0)+l1*sin(Dfi0)) +
%                                                    (1/2)*Cg_sa*(l4*cos(Dfi0)-l2*sin(Dfi0))
%                  (t/Ta)*sin(wr*t)           y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0)) +
%                                                    Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wa*t)        y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) –
%                                                    Cg0*Dfi_ca*(NS*cos(Dfi0) + EW*sin(Dfi0)) +
%                                                    Cg_ca*(-NS*sin(Dfi0) + EW*cos(Dfi0))
%                  sin(wr*t)*sin(wa*t)        y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) –
%                                                    Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                                    Cg_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wo*t)        y(10)= -Cg0*Ao*sin(Dfi0)
%           (t/Ta)*cos(wr*t)*cos(wa*t)        y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                     Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%                                                     Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%           (t/Ta)*sin(wr*t)*cos(wa*t)        y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%                                                     Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) +
%                                                     Cg_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*cos(wa*t)         y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0)
%       sin(wr*t)*cos(wo*t)*cos(wa*t)         y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0)
%       (t/Ta)*cos(wr*t)*sin(wa*t)            y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                     Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%                                                     Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wa*t)            y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%                                                     Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) +
%                                                     Cg_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*sin(wa*t)         y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0)
%       sin(wr*t)*cos(wo*t)*sin(wa*t)         y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0)
%       (t/Ta)^2*cos(wr*t)                    y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*cos(wo*t)            y(20) = Cg_t*Ao*cos(Dfi0)
%
```

Model 4.

```
% Model 4 (2) Cg = Cg0 + Cg_t*(t/Ta) + Cg_co*cos(wo*t) + Cg_so*sin(wo*t) + Cg_ca*cos(wa*t) + Cg_sa*sin(wa*t)
%       Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
%State Vector       Time Signature          First Step States
%  x(1) = Cg0       cos(wr*t)               y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%  x(2) = Dfi0                                    (1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) +
%  x(3) = NS0                                     (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0) +l4*cos(Dfi0)) +
%  x(4) = EW0                                     (1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) +
%  x(5) = Rg*Ta                                   (1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))+(1/2)*Ao*Cg_co*Cos(Dfi0)
%  x(6) = Rm*Ta     (t/Ta)*cos(wr*t)        y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%                                                 Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(7) = Dfi_ca    cos(wr*t)*cos(wa*t)     y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(-NS*sin(Dfi0) +
%  x(8) = Dfi_sa                                  EW*cos(Dfi0)) + Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(9) = Cg_t      cos(wr*t)*sin(wa*t)     y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(-NS*sin(Dfi0) +
%  x(10) = Cg_ca                                  EW*cos(Dfi0)) + Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(11) = Cg_sa    cos(wr*t)*cos(wo*t)     y(5)= Cg0*Ao*cos(Dfi0) + Cg_co(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(12) = Cg_co    sin(wr*t)               y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) –(1/2)*Cg0*Dfi_ca*
%  x(13) = Cg_so                                  (l1*cos(Dfi0) + l3*sin(Dfi0))  - (1/2)*Cg0*Dfi_sa* (l2*cos(Dfi0) +
%                                                 l4*sin(Dfi0))  + (1/2)*Cg_ca(l3*cos(Dfi0)+l1*sin(Dfi0)) +
%                                                 (1/2)*Cg_sa*(l4*cos(Dfi0)-l2*sin(Dfi0)) -  (1/2)*Ao*Cg_co*sin(Dfi0)
%                   (t/Ta)*sin(wr*t)        y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))  +
%                                                 Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   sin(wr*t)*cos(wa*t)     y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) –
%                                                 Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                                 Cg_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   sin(wr*t)*sin(wa*t)     y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) –
%                                                 Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                                 Cg_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                   sin(wr*t)*cos(wo*t)     y(10)= -Cg0*Ao*sin(Dfi0)+Cg_co*(EW*Cos(Dfio)-Ns*Sin(Dfi0))
%             (t/Ta)*cos(wr*t)*cos(wa*t)    y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                  Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%                                                  Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%             (t/Ta)*sin(wr*t)*cos(wa*t)    y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0) + RmTa*sin(Dfi0))  +
%                                                  Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) +
%                                                  Cg_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*cos(wa*t)       y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0) +
%                                                  Cg_co(l1*cos(Dfi0)+l3*sin(Dfi0))
%       sin(wr*t)*cos(wo*t)*cos(wa*t)       y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0) +
%                                                  Cg_co(l3*cos(Dfi0)-l1*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*sin(wa*t)          y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                  Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%                                                  Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wa*t)          y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0) + RmTa*sin(Dfi0))  +
%                                                  Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) +
%                                                  Cg_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*sin(wa*t)       y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0) +
%                                                  Cg_co*(l2*Cos(Dfi0)+l4*sin(Dfi0))
%       sin(wr*t)*cos(wo*t)*sin(wa*t)       y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0) +
%                                                  Cg_co(l4*cos(Dfi0)-l2*sin(Dfi0))
%       (t/Ta)^2*cos(wr*t)                  y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*cos(wo*t)          y(20)= Cg_t*Ao*cos(Dfi0)+ Cg_co*(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*cos(wo*t)          y(21)= -Cg_t*Ao*sin(Dfi0) + Cg_co(RmTa*Cos(Dfi0)-RgTa*Sin(Dfi0))
%       cos(wr*t)*sin(wo*t)                 y(22)= Cg_so(NS*cos(Dfi0)+ EW*sin(Dfi0))
%       sin(wr*t)*sin(wo*t)                 y(23)= Cg_so(EW*cos(Dfi0)- NS*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*sin(wo*t)          y(24)= Cg_so(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wo*t)          y(25)= -Cg_so(RmTa*cos(Dfi0)+RgTa*sin(Dfi0))
%       cos(wr*t)*cos(2*wo*t)               y(26)= 1/2*Cg_co*Ao*cos(Dfi0)
%       sin(wr*t)*cos(2*wo*t)               y(27)= -(1/2)*Cg_co*Ao*sin(Dfi0)
%       cos(wr*t)*sin(2*wo*t)               y(28)= (1/2)*Cg_so*Ao*cos(Dfi0)
```

```
%        sin(wr*t)*sin(2*wo*t)              y(29)= -(1/2)*Cg_so*Ao*sin(Dfi0)
%        cos(wr*t)*sin(wo*t)*cos(wa*t)  y(30)= Cg_so*(l1*cos(Dfi0)+l3*sin(Dfi0))
%        cos(wr*t)*sin(wo*t)*sin(wa*t)  y(31)= Cg_so*(l2*cos(Dfi0)+l4*sin(Dfi0))
%        sin(wr*t)*sin(wo*t)*cos(wa*t)  y(32)= Cg_so*(l3*cos(Dfi0)-l1*sin(Dfi0))
%        sin(wr*t)*sin(wo*t)*sin(wa*t)  y(33)= Cg_so*(l4*cos(Dfi0)-l2*sin(Dfi0))
%==============================================================================
```

The program **CovarianceAnalysis** returns and plots the accuracy (1-sigma) vectors for the drift rate components R_{NS} and R_{EW} versus measurement time. Calculations take into account the fact, that measurement points are 'valid' only when the Guide Star is visible (valid), and also allows to assign different noise level to any individual data point (periodic mode ECU on/off).

References:

1. Keiser, G.M., A.S. Silbergleit, and M. Heifetz, *Gravity Probe B Data Processing: Combining Gyroscope and Telescope Readouts*, GP-B, Hansen Laboratories, S0907, Rev. A,
2. Keiser, G.M., *Analytical Solution for the Gravity Probe B Covariance Matrix*, Gravity Probe B, Stanford University, S0351, October 21, 1998

Program CovarianceAnalysis.m: Script

```
% [Geod,FrameDr]=CovarianceAnalysis
% Covariance Analysis (Analytic Solution)
model = 0;

readSfromFile=0;

Dur_months = 13; % duration, months
startDay = -26; % start day, Jan 1 = 1,
Tr = 3; % Roll period, min.
durECUonHours = 6; % duration when ECU is ON, hours
periodECUswitchDays = 10; % total duration of the ECU On/Off cycle, days


dt = 2;    % data rate (sec)
To = 90*60;  % orbital period (sec)
Ta = 3.15E7;  % annual period (sec)
iocDays = 60; % IOC duration (days)
outStepMonth = 1; % output rate ( months )

fspinActual = [126.6, 151.4, 128.1, 146.9]';   % (Hz)
sigmaNoiseActual = [120, 148, 136, 124]'/1000;    % (arcsec/rt(Hz)
sigmaNoiseEcuOff = ( (sigmaNoiseActual*sqrt(Tr/3)).*(130./fspinActual) /sqrt(2*dt) )';

coefECUOff = [1, 1, 1, 1];
%kECUOn = 1;
kECUOn = 1E10;
coefECUOn = kECUOn * [1, 1, 1, 1];

DeltaFi_ann_ampl = 0;
DeltaFi_ann_ph = 0;
Cg_t = 0;
Cg_ann = 0;
Cg_ann_ph = 0;
Cg_orb = 0;
Cg_orb_ph = 0;


%SQUID Readout Signal Model
%
% z(t) = Cg*[(NSo+Rg*t-AberNS(t))*cos(wr*t+deltaphi) + ...
%            (EWo+Rf*t-AberEW(t))*sin(wr*t+deltaphi)] + ...
%             bias + v
%
% ======================================================================
% Model 0: Cg, Dfi - const (Cg = Cg0, Dfi = Dfi0)
%
% State Vector          First Step States                              Time Signature
%
%  x(1) = Cg           y(1)= Cg*(NS*cos(Dfi)+EW*sin(Dfi))              cos(wr*t)
%  x(2) = deltaphi (Dfi) y(2)= Cg*(RgTa*cos(Dfi)+ RmTa*sin(Dfi))       (t/Ta)*cos(wr*t)
%  x(3) = NSo            y(3)= Cg*(l1*cos(Dfi) + l3*sin(Dfi))          cos(wr*t)*cos(wa*t)
%  x(4) = EWo            y(4)= Cg*(l2*cos(Dfi) + l4*sin(Dfi))          cos(wr*t)*sin(wa*t)
%  x(5) = Rg*Ta        y(5)= Cg*Ao*cos(Dfi)                            cos(wr*t)*cos(wo*t)
%  x(6) = Rm*Ta        y(6)= Cg*(-NS*sin(Dfi)+EW*cos(Dfi))             sin(wr*t)
%                      y(7)= Cg*(-RgTa*sin(Dfi)+ RmTa*cos(Dfi))        (t/Ta)*sin(wr*t)
%                      y(8)= Cg*(-l1*sin(Dfi) + l3*cos(Dfi))           sin(wr*t)*cos(wa*t)
%                      y(9)= Cg*(-l2*sin(Dfi) + l4*cos(Dfi))           sin(wr*t)*sin(wa*t)
%                      y(10)= -Cg*Ao*sin(Dfi)                          sin(wr*t)*cos(wo*t)
%
```

```
%===========================================================================
% ==========================================================================
%Model 1 (21): Cg = Cg0 + Cg_t*t/Ta ,
%
% State Vector      Time Signature
%  x(1) = Cg0        cos(wr*t)                  y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(2) = Dfi0       (t/Ta)*cos(wr*t)           y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%                                                     Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(3) = NSo         cos(wr*t)*cos(wa*t)             y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0))
%  x(4) = EWo         cos(wr*t)*sin(wa*t)       y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0))
%  x(5) = Rg*Ta      cos(wr*t)*cos(wo*t)        y(5)= Cg0*Ao*cos(Dfi0)
%  x(6) = Rm*Ta      sin(wr*t)                  y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                    (t/Ta)*sin(wr*t)           y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0)) +
%  x(7) = Cg_t                                        Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                    sin(wr*t)*cos(wa*t)        y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0))
%                    sin(wr*t)*sin(wa*t)        y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0))
%                    sin(wr*t)*cos(wo*t)        y(10)= -Cg0*Ao*sin(Dfi0)
%                 (t/Ta)^2*cos(wr*t)            y(11) = Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*cos(wa*t)    y(12) = Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*sin(wa*t)    y(13)= Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0))
%                 (t/Ta)*cos(wr*t)*cos(wo*t)    y(14)= Cg_t*Ao*cos(Dfi0)
%                 (t/Ta)^2*sin(wr*t)            y(15)= Cg_t*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wa*t)    y(16) = Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*sin(wa*t)    y(17)= Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wo*t)     y(18)= -Cg_t*Ao*sin(Dfi0)
%
%=============================================================================
% Model 2 (32): Cg = Cg0;
%        Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
% State Vector      Time Signature
%  x(1) = Cg0        cos(wr*t)                y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%  x(2) = Dfi0                                     (1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) +
%  x(3) = NSo                                      (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0)+l4*cos(Dfi0))
%  x(4) = EWo        (t/Ta)*cos(wr*t)         y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%  x(5) = Rg*Ta      cos(wr*t)*cos(wa*t)      y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%  x(6) = Rm*Ta                                     Cg0*Dfi_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%  x(7) = Dfi_ca     cos(wr*t)*sin(wa*t)      y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%  x(8) = Dfi_sa                                    Cg0*Dfi_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                    cos(wr*t)*cos(wo*t)      y(5)= Cg0*Ao*cos(Dfi0)
%                    sin(wr*t)                y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) –
%                                                   (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) –
%                                                   (1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))
%                    (t/Ta)*sin(wr*t)         y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))
%                    sin(wr*t)*cos(wa*t)      y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) –
%                                                   Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                    sin(wr*t)*sin(wa*t)       y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) –
%                                                   Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                    sin(wr*t)*cos(wo*t)        y(10)= -Cg0*Ao*sin(Dfi0)
%                 (t/Ta)*cos(wr*t)*cos(wa*t)       y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*cos(wa*t)       y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 cos(wr*t)*cos(wo*t)*cos(wa*t)    y(13)= Cg0*Dfi_ca*Ao*sin(Dfi0)
%                 sin(wr*t)*cos(wo*t)*cos(wa*t)    y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0)
%                 (t/Ta)*cos(wr*t)*sin(wa*t)       y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%                 (t/Ta)*sin(wr*t)*sin(wa*t)       y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%                 cos(wr*t)*cos(wo*t)*sin(wa*t)    y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0)
%                 sin(wr*t)*cos(wo*t)*sin(wa*t)    y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0)
%=============================================================================
```

```
%Model 3 (1): Cg = Cg0 + Cg_t*(t/Ta) + Cg_ca*cos(wa*t) + Cg_sa*sin(wa*t)
%             Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
% State Vector     Time Signature        First Step States

%  x(1) = Cg0      cos(wr*t)              y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +(1/2)*Cg0*Dfi_ca*
%  x(2) = Dfi0                                  (l1*sin(Dfi0)+l3*cos(Dfi0)) + (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0) +
%  x(3) = NSo                                   l4*cos(Dfi0))+(1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0))+
%  x(4) = EWo                                   (1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))+(1/2)*Ao*Cg_co*Cos(Dfi0)
%  x(5) = Rg*Ta    (t/Ta)*cos(wr*t)       y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%  x(6) = Rm*Ta                                 Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%                  cos(wr*t)*cos(wa*t)    y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(NS*sin(Dfi0) +
%  x(7) = Dfi_ca                                EW*cos(Dfi0)) + Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(8) = Dfi_sa   cos(wr*t)*sin(wa*t)    y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(NS*sin(Dfi0) +
%  x(9) = Cg_t                                  EW*cos(Dfi0)) + Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(10) = Cg_ca   cos(wr*t)*cos(wo*t)    y(5)= Cg0*Ao*cos(Dfi0)
%  x(11) = Cg_sa   sin(wr*t)              y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) –
%                                               (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) –
%                                               (1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0)) +
%                                               (1/2)*Cg_ca*(l3*cos(Dfi0)+l1*sin(Dfi0)) +
%                                               (1/2)*Cg_sa*(l4*cos(Dfi0)-l2*sin(Dfi0))
%                  (t/Ta)*sin(wr*t)       y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0)) +
%                                               Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wa*t)    y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) –
%                                               Cg0*Dfi_ca*(NS*cos(Dfi0) + EW*sin(Dfi0)) +
%                                               Cg_ca*(-NS*sin(Dfi0) + EW*cos(Dfi0))
%                  sin(wr*t)*sin(wa*t)    y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) –
%                                               Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                               Cg_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wo*t)    y(10)= -Cg0*Ao*sin(Dfi0)
%            (t/Ta)*cos(wr*t)*cos(wa*t)   y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%                                                Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%            (t/Ta)*sin(wr*t)*cos(wa*t)   y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  +
%                                                Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) +
%                                                Cg_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*cos(wa*t)     y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0)
%       sin(wr*t)*cos(wo*t)*cos(wa*t)     y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0)
%       (t/Ta)*cos(wr*t)*sin(wa*t)        y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%                                                Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wa*t)        y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  +
%                                                Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) +
%                                                Cg_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*sin(wa*t)     y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0)
%       sin(wr*t)*cos(wo*t)*sin(wa*t)     y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0)
%       (t/Ta)^2*cos(wr*t)                y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*cos(wo*t)        y(20) = Cg_t*Ao*cos(Dfi0)
%       (t/Ta)*sin(wr*t)*cos(wo*t)        y(21) = -Cg_t*Ao*sin(Dfi0)
```

```
%==========================================================================

% Model 4 (2) Cg = Cg0 + Cg_t*(t/Ta) + Cg_co*cos(wo*t) + Cg_so*sin(wo*t) + Cg_ca*cos(wa*t) + Cg_sa*sin(wa*t)
%       Dfi = Dfi0 + Dfi_ca*cos(wa*t) + Dfi_sa*sin(wa*t)
%
%State Vector      Time Signature           First Step States
%  x(1) = Cg0      cos(wr*t)                y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%  x(2) = Dfi0                                     (1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) +
%  x(3) = NS_0                                     (1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0) +l4*cos(Dfi0)) +
%  x(4) = EW_0                                     (1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) +
%  x(5) = Rg*Ta                                    (1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))+(1/2)*Ao*Cg_co*Cos(Dfi0)
%  x(6) = Rm*Ta    (t/Ta)*cos(wr*t)         y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) +
%                                                  Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(7) = Dfi_ca   cos(wr*t)*cos(wa*t)      y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(-NS*sin(Dfi0) +
%  x(8) = Dfi_sa                                   EW*cos(Dfi0)) + Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(9) = Cg_t     cos(wr*t)*sin(wa*t)      y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(-NS*sin(Dfi0) +
%  x(10) = Cg_ca                                   EW*cos(Dfi0)) + Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(11) = Cg_sa   cos(wr*t)*cos(wo*t)      y(5)= Cg0*Ao*cos(Dfi0) + Cg_co(NS*cos(Dfi0)+EW*sin(Dfi0))
%  x(12) = Cg_co   sin(wr*t)                y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) -(1/2)*Cg0*Dfi_ca*
%  x(13) = Cg_so                                   (l1*cos(Dfi0) + l3*sin(Dfi0))  - (1/2)*Cg0*Dfi_sa* (l2*cos(Dfi0) +
%                                                  l4*sin(Dfi0))  + (1/2)*Cg_ca(l3*cos(Dfi0)+l1*sin(Dfi0)) +
%                                                  (1/2)*Cg_sa*(l4*cos(Dfi0)-l2*sin(Dfi0)) -  (1/2)*Ao*Cg_co*sin(Dfi0)
%                  (t/Ta)*sin(wr*t)         y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))  +
%                                                  Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wa*t)      y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) -
%                                                  Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                                  Cg_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*sin(wa*t)      y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) -
%                                                  Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
%                                                  Cg_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0))
%                  sin(wr*t)*cos(wo*t)      y(10)= -Cg0*Ao*sin(Dfi0)+Cg_co*(EW*Cos(Dfio)-Ns*Sin(Dfi0))
%         (t/Ta)*cos(wr*t)*cos(wa*t)        y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                   Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
%                                                   Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%         (t/Ta)*sin(wr*t)*cos(wa*t)        y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0) + RmTa*sin(Dfi0))  +
%                                                   Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) +
%                                                   Cg_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*cos(wa*t)       y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0) +
%                                                   Cg_co(l1*cos(Dfi0)+l3*sin(Dfi0))
%       sin(wr*t)*cos(wo*t)*cos(wa*t)       y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0) +
%                                                   Cg_co(l3*cos(Dfi0)-l1*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*sin(wa*t)          y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) +
%                                                   Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
%                                                   Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wa*t)          y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0) + RmTa*sin(Dfi0))  +
%                                                   Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) +
%                                                   Cg_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
%       cos(wr*t)*cos(wo*t)*sin(wa*t)       y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0) +
%                                                   Cg_co*(l2*Cos(Dfi0)+l4*sin(Dfi0))
%       sin(wr*t)*cos(wo*t)*sin(wa*t)       y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0) +
%                                                   Cg_co(l4*cos(Dfi0)-l2*sin(Dfi0))
%       (t/Ta)^2*cos(wr*t)                  y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*cos(wo*t)          y(20)= Cg_t*Ao*cos(Dfi0)+ Cg_co*(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*cos(wo*t)          y(21)= -Cg_t*Ao*sin(Dfi0) + Cg_co(RmTa*Cos(Dfi0)-RgTa*Sin(Dfi0))
%       cos(wr*t)*sin(wo*t)                 y(22)= Cg_so(NS*cos(Dfi0)+ EW*sin(Dfi0))
%       sin(wr*t)*sin(wo*t)                 y(23)= Cg_so(EW*cos(Dfi0)- NS*sin(Dfi0))
%       (t/Ta)*cos(wr*t)*sin(wo*t)          y(24)= Cg_so(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
%       (t/Ta)*sin(wr*t)*sin(wo*t)          y(25)= -Cg_so(RmTa*cos(Dfi0)+RgTa*sin(Dfi0))
%       cos(wr*t)*cos(2*wo*t)               y(26)= 1/2*Cg_co*Ao*cos(Dfi0)
%       sin(wr*t)*cos(2*wo*t)               y(27)= -(1/2)*Cg_co*Ao*sin(Dfi0)
%       cos(wr*t)*sin(2*wo*t)               y(28)= (1/2)*Cg_so*Ao*cos(Dfi0)
```

```
%       sin(wr*t)*sin(2*wo*t)           y(29)= -(1/2)*Cg_so*Ao*sin(Dfi0)
%       cos(wr*t)*sin(wo*t)*cos(wa*t)   y(30)= Cg_so*(l1*cos(Dfi0)+l3*sin(Dfi0))
%       cos(wr*t)*sin(wo*t)*sin(wa*t)   y(31)= Cg_so*(l2*cos(Dfi0)+l4*sin(Dfi0))
%       sin(wr*t)*sin(wo*t)*cos(wa*t)   y(32)= Cg_so*(l3*cos(Dfi0)-l1*sin(Dfi0))
%       sin(wr*t)*sin(wo*t)*sin(wa*t)   y(33)= Cg_so*(l4*cos(Dfi0)-l2*sin(Dfi0))
%===========================================================================


tt=clock;
cl0=tt(4:6)*[3600,60,1]';
disp(['Date: ',num2str(tt(1)),'/',num2str(tt(2)),'/',num2str(tt(3))]);

disp(['Covariance Analysis Started...']);
disp(['Model: ', num2str(model)])
disp(['Reading Data from file: ',num2str(readSfromFile)])


% Parameters

Cg0 = 1; % Scale factor (d-less)
Dfi0_arcsec = 10;
Dfi0 = Dfi0_arcsec/3600/180*pi; %(rad)

NS = 1; % arcsec
EW = 1; % arcsec
Rg = 6.6144; % arcsec/yr
Rm = 0.04093; % arcsec/yr

Cg_t = Cg_t*Cg0;
Cg_ca = Cg_ann*Cg0*sin(Cg_ann_ph/180*pi);
Cg_sa = Cg_ann*Cg0*cos(Cg_ann_ph/180*pi);
Cg_co = Cg_orb*Cg0*sin(Cg_orb_ph/180*pi);
Cg_so = Cg_orb*Cg0*cos(Cg_orb_ph/180*pi);

Dfi_t = 0;
Dfi_o = 0;
Dfi_ca_arcs = DeltaFi_ann_ampl*sin(DeltaFi_ann_ph/180*pi);
Dfi_sa_arcs = DeltaFi_ann_ampl*cos(DeltaFi_ann_ph/180*pi);
Dfi_ca = Dfi_ca_arcs*(pi/180)/3600;
Dfi_sa = Dfi_sa_arcs*(pi/180)/3600;

% flag ECUon
durECUon = durECUonHours * 3600;
periodECUswitch = periodECUswitchDays * 24 * 3600;
phaseECUswitch = periodECUswitch - durECUon;

saveFileName = ['COV_model_', num2str(model), '_Dur_', num2str(Dur_months), ...
        '_ECU_', num2str(durECUonHours), '-', num2str(periodECUswitchDays)];

disp(['Duration: ', num2str(Dur_months), ' months.']);
disp(['Start Day: ', num2str(startDay+iocDays)]);
disp(['Duration ECU On: ', num2str(durECUonHours), ' hours.']);
disp(['ECU On/Off Cycle Period: ', num2str(periodECUswitchDays), ' days.']);


disp(['Dfi_annual_amplitude: ', num2str(DeltaFi_ann_ampl),' arcsec'])
disp(['Dfi_annual_annual_phase: ', num2str(DeltaFi_ann_ph), ' degr.'])
disp(['Cg_t: ', num2str(Cg_t),' 1/yr'])
disp(['Cg_annual_amplitude: ', num2str(Cg_ann*Cg0)])
disp(['Cg_annual_phase: ', num2str(Cg_ann_ph),' degr.'])
disp(['Cg_orbital_amplitude: ', num2str(Cg_orb*Cg0)])
disp(['Cg_orbital_phase: ', num2str(Cg_orb_ph),' degr.'])
```

```
wo = 2*pi/To; % orbital frequency (rad/sec)
wa = 2*pi/Ta; % annual frequency (rad/sec)

wr = 2*pi /(Tr*60); %roll frequency

Tf = Dur_months/12*365*24*3600;
NdataPoints = Tf/dt;

outStepSec = outStepMonth/12*365*24*3600;
nStep = outStepSec/dt;
nPoints = floor(Tf/outStepSec);

Geod = zeros( nPoints,4);
FrameDr = zeros( nPoints,4);
tPlot = zeros(nPoints,1);

if (readSfromFile)
  eval(['load ', saveFileName, ' SS']);
end;

switch(model)
case 0
  Nx_aug = 0;
  Ny_aug = 0;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm]';

case 1
  Nx_aug = 5;
  Ny_aug = 11;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Dfi_ca, Dfi_sa, Cg_t, Cg_ca, Cg_sa]';

case 2
  Nx_aug = 7;
  Ny_aug = 23;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Dfi_ca, Dfi_sa, Cg_t, Cg_ca, Cg_sa, Cg_co, Cg_so]';

case 21
  Nx_aug = 1;
  Ny_aug = 8;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Cg_t]';

case 31
  Nx_aug = 1;
  Ny_aug = 4;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Dfi_ca]';

case 32
  Nx_aug = 2;
  Ny_aug = 8;
  Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Dfi_ca, Dfi_sa]';
end

S = zeros(10+Ny_aug,10+Ny_aug,4);

Iy = cell(nPoints,1);
Py = cell(nPoints,1);
DiagY = cell(nPoints,1);
Ix = cell(nPoints,1);
P2 = cell(nPoints,1);

phi0 = 0; % initial angular position of S/C on orbit (at t= 0)
```

```
% Guide Star Valid Region
gsv_sector = 0; % deg
alpha = sin(gsv_sector/180*pi);

tStart = -Tf/2;
tEnd = Tf/2;
tECUNextOn = tStart + phaseECUswitch;
tECUNextOff = tECUNextOn + durECUon;

coefSwitch = coefECUOff;

i = 0;
k = 0;
for (t = tStart : dt : tEnd)

  if ~readSfromFile

    % GS Valid flag
    phi = phi0 + wo*t; % current angular position of S/C on orbit
    gsvalid = (cos(phi) > alpha);
    if ( gsvalid )

      if (t >= tECUNextOn)
        if ( t <= tECUNextOff )
          coefSwitch = coefECUOn;
        else
          tECUNextOn = tECUNextOn + periodECUswitch;
          tECUNextOff = tECUNextOn + durECUon;
          coefSwitch = coefECUOff;
        end;
      end;

      cr = cos(wr*t);
      sr = sin(wr*t);
      tau = t/Ta;
      ca = cos(wa*t);
      sa = sin(wa*t);
      co = cos(wo*t);
      so = sin(wo*t);
      c2o = cos(2*wo*t);
      s2o = sin(2*wo*t);

      switch model

%==============================================================================
        case 0,

          H = [ cr, tau*cr,  ca*cr, sa*cr,  co*cr,  sr, tau*sr, ca*sr, sa*sr, co*sr];


%==============================================================================
        case 1

          H = [ cr,  tau*cr,  ca*cr, sa*cr,  co*cr, sr, tau*sr, ca*sr, sa*sr, co*sr,...
                tau*cr*ca, tau*sr*ca, cr*co*ca, sr*co*ca, tau*cr*sa, tau*sr*sa, cr*co*sa, sr*co*sa,...
                tau^2*cr, tau*cr*co, tau*sr*co ];

%==============================================================================
        case 2

          H = [ cr,  tau*cr,  cr*ca, sa*cr,  cr*co, sr, tau*sr, sr*ca, sr*sa, sr*co,...
```

```
            tau*cr*ca, tau*sr*ca, cr*co*ca, sr*co*ca, tau*cr*sa, tau*sr*sa, cr*co*sa, sr*co*sa,...
            tau^2*cr, tau*cr*co, tau*sr*co, cr*so, sr*so, tau*cr*so, tau*sr*so, cr*c2o, sr*c2o, cr*s2o, sr*s2o,...
            cr*so*ca, cr*so*sa, sr*so*ca, sr*so*sa];

%==========================================================================
      case 21,

        H = [ cr, tau*cr,  ca*cr, sa*cr,  co*cr,  sr, tau*sr, ca*sr, sa*sr, co*sr,...
            tau^2*cr, tau*cr*ca, tau*cr*sa, tau*cr*co, tau^2*sr, tau*sr*ca, tau*sr*sa, tau*sr*co];


%==========================================================================
      case 31,

        H = [ cr tau*cr ca*cr sa*cr co*cr, sr tau*sr ca*sr sa*sr co*sr,...
            tau*cr*ca, tau*sr*ca, cr*co*ca, sr*co*ca];

      case 32,

        H = [ cr tau*cr ca*cr sa*cr co*cr, sr tau*sr ca*sr sa*sr co*sr,...
            tau*cr*ca, tau*sr*ca, cr*co*ca, sr*co*ca, tau*cr*sa, tau*sr*sa, cr*co*sa, sr*co*sa ];
    end;  %model

%==========================================================================

    HTH = H'*H;

    for (nSquid = 1:4)
      S(:,:,nSquid) = S(:,:,nSquid) + HTH/(coefSwitch(nSquid)^2);
    end;

  end; % if gsvalid

end; % ~readSfromFile

i = i+1;
if (i==nStep)

  i = 0;
  k = k+1;

  midPointDays = startDay + iocDays;
  disp(' ');
  disp(['Month # ', num2str(k)]);

  DyDx = GradientAberr(model,Ny_aug, Nx_aug, Parameters, midPointDays);

  if (readSfromFile)
    S = SS{k};
  else
    SS{k} = S;
  end;

  for (nSquid = 1:4)
    Ix = DyDx' * S(:,:,nSquid) * DyDx;
    P2 = inv(Ix);
    Diag = sqrt(diag(P2));
    Geod(k,nSquid)      = sigmaNoiseEcuOff(nSquid)*Diag(5)*1000; %marcs/yr
    FrameDr(k,nSquid)   = sigmaNoiseEcuOff(nSquid)*Diag(6)*1000; %marcs/yr
    ScaleFactor(nSquid) = Diag(1)*sigmaNoiseEcuOff(nSquid);
    NS(nSquid) = Diag(3)*sigmaNoiseEcuOff(nSquid);
    EW(nSquid) = Diag(4)*sigmaNoiseEcuOff(nSquid);
```

```
        end;

        tPlot(k) = floor((t+Tf/2)/3600/24); %days

        disp(['Model: ', num2str(model)]);
        disp(['Geod = ', num2str(Geod(k,:),  '%10.4f')]);
        disp(['FrDr = ', num2str(FrameDr(k,:),'%10.4f')]);
        disp(['Scale factor = ', num2str(ScaleFactor)]);
        disp(['NS = ', num2str(NS,'%10.5f')]);
        disp(['EW = ', num2str(EW,'%10.5f')]);

    end;

end;

rEcuOnOff = durECUonHours / (periodECUswitchDays*24) * 100;
for (nSquid=1:4)
    figure(nSquid);

    nEnd = length(Geod);
    %nStart = 1;
    nStart = floor(nEnd/4);

    hPlot = subplot(2,1,1, 'replace');

    plot(tPlot, Geod(:,nSquid));
    xlabel('Duration (days)');
    ylabel('marcs/yr');
    grid;
    title(['Gyroscope ', num2str(nSquid), ' Geodetic Drift Rate Error. (Model ', num2str(model),')']);

    legend(['Roll ', num2str(Tr), ' min.'],['ECU On/Off ratio: ', num2str(rEcuOnOff, '%5.2f'),' %']);
    axis([200,tPlot(end), 0, 2.5]);

    hPlot = subplot(2,1,2, 'replace');

    plot(tPlot, FrameDr(:,nSquid));
    xlabel('Duration (days)');
    ylabel('marcs/yr');
    grid;
    title(['Gyroscope ', num2str(nSquid), ' Frame-Dragging Drift Rate Error']);
    legend(['Roll ', num2str(Tr), ' min.']);
    axis([200,tPlot(end), 0, 2.5]);

end;

eval(['save ', saveFileName, ' SS tPlot Geod FrameDr rEcuOnOff model Tr']);

tt=clock;
cl2=tt(4:6)*[3600,60,1]';
disp(['Elapsed time is ', num2str((cl2-cl0)/60),' min  for ',num2str(Dur_months),' months of measurements']);
```

Program GP-B Jacobian (GradientAberr.m)

```
function DyDx = GradientAberr(model,Ny_aug, Nx_aug, Parameters, midPointDays);

deltaGS = 16.8/180*pi; % Guide Star declination (rad)
alphaGS = 343.26/180*pi; % Guide Star right ascention (rad)
iE = 23.5/180*pi; % inclination of the Earth's rotation axis (relative to the ecliptic)
c = 299792.458; %speed of light (km/sec)
```

```
R_es = 1.5E8; % Radius of the Earth orbit around the Sun (km)
Ta = 3.15E7;  % annual period (sec)
wa = 2*pi/Ta; % annual frequency (rad/sec)
Ao = 5.1841; % Amplitude of orbital aberration (arcsec)

sa = sin(alphaGS);
ca = cos(alphaGS);
sd = sin(deltaGS);
cd = cos(deltaGS);
si = sin(iE);
ci = cos(iE);
k  = R_es*wa/c *( 180/pi*3600 ); %arcsec

fia = 2*pi * (midPointDays/365);
sf = sin(fia);
cf = cos(fia);

l1 = k*( cf*(sa*sd*ci-cd*si) + sf*(-ca*sd) ); %arcsec
l2 = k*( cf*(-ca*sd) + sf*(-sa*sd*ci+cd*si) );% "-"
l3 = k*( cf*(-ca*ci) + sf*(-sa) );          % "-"
l4 = k*( cf*(-sa) + sf*(ca*ci) );           % "-"

Cg = Parameters(1); % Scale factor (d-less)
Dfi0 =  Parameters(2);
NS =  Parameters(3); % arcsec
EW =  Parameters(4); % arcsec
Rg =  Parameters(5); % arcsec/yr
Rm =  Parameters(6); % arcsec/yr


sD = sin(Dfi0);
cD = cos(Dfi0);

R1 = Rg*cD + Rm*sD;
R2 = -Rg*sD + Rm*cD;
L1 = l1*cD + l3*sD;
L2 = -l1*sD + l3*cD;
L3 =  l2*cD + l4*sD;
L4 = -l2*sD + l4*cD;
N1 = NS*cD + EW*sD;
N2 = -NS*sD + EW*cD;

% Jacobian DyDxCore (10x6)
% Model 0: Cg, Dfi - const (Cg = Cg0, Dfi = Dfi0)
%
% State Vector              First Step States                         Time Signature
%
%   x(1) = Cg               y(1)= Cg*(NS*cos(Dfi)+EW*sin(Dfi))           cos(wr*t)
%   x(2) = deltaphi (Dfi)   y(2)= Cg*(RgTa*cos(Dfi)+ RmTa*sin(Dfi))      (t/Ta)*cos(wr*t)
%   x(3) = NS               y(3)= Cg*(l1*cos(Dfi) + l3*sin(Dfi))         cos(wr*t)*cos(wa*t)
%   x(4) = EW               y(4)= Cg*(l2*cos(Dfi) + l4*sin(Dfi))         cos(wr*t)*sin(wa*t)
%   x(5) = Rg*Ta            y(5)= Cg*Ao*cos(Dfi)                         cos(wr*t)*cos(wo*t)
%   x(6) = Rm*Ta            y(6)= Cg*(-NS*sin(Dfi)+EW*cos(Dfi))          sin(wr*t)
%                           y(7)= Cg*(-RgTa*sin(Dfi)+ RmTa*cos(Dfi))     (t/Ta)*sin(wr*t)
%                           y(8)= Cg*(-l1*sin(Dfi) + l3*cos(Dfi))        sin(wr*t)*cos(wa*t)
%                           y(9)= Cg*(-l2*sin(Dfi) + l4*cos(Dfi))        sin(wr*t)*sin(wa*t)
%                           y(10)= -Cg*Ao*sin(Dfi)                       sin(wr*t)*cos(wo*t)
%
%==================================================================================================
================================
%   Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm]';
```

```
DyDxCore = zeros(10,6);

DyDxCore(1,1) = N1;
DyDxCore(1,2) = Cg*N2;
DyDxCore(1,3) = Cg*cD;
DyDxCore(1,4) = Cg*sD;

DyDxCore(2,1) = R1;
DyDxCore(2,2) = Cg*R2;
DyDxCore(2,5) = Cg*cD;
DyDxCore(2,6) = Cg*sD;

DyDxCore(3,1) = L1;
DyDxCore(3,2) = Cg*L2;

DyDxCore(4,1) = L3;
DyDxCore(4,2) = Cg*L4;

DyDxCore(5,1) = Ao*cD;
DyDxCore(5,2) = -Cg*Ao*sD;

DyDxCore(6,1) = N2;
DyDxCore(6,2) = -Cg*N1;
DyDxCore(6,3) = -Cg*sD;
DyDxCore(6,4) =  Cg*cD;

DyDxCore(7,1) = R2;
DyDxCore(7,2) = -Cg*R1;
DyDxCore(7,5) = -Cg*sD;
DyDxCore(7,6) =  Cg*cD;

DyDxCore(8,1) =  L2;
DyDxCore(8,2) = -Cg*L1;

DyDxCore(9,1) =  L4;
DyDxCore(9,2) = -Cg*L3;

DyDxCore(10,1) = -Ao*sD;
DyDxCore(10,2) = -Cg*Ao*cD;

if (model==0)
   DyDx = DyDxCore;
   return;
end;

% Additional States (X: Nx_aug; Y:Ny_aug)
% Jacobian (10+Ny_aug x 6+Nx_aug)

DyDx = [DyDxCore,        zeros(10,    Nx_aug); ...
      zeros(Ny_aug,6), zeros(Ny_aug,Nx_aug)];

%==========================================================================
==========================
if (model==1)

   Dfi_ca = Parameters(7);
   Dfi_sa = Parameters(8);
   Cg_t =  Parameters(9);
   Cg_ca =  Parameters(10);
   Cg_sa =  Parameters(11);

   CgDfi_ca = Cg*Dfi_ca;
```

```
CgDfi_sa = Cg*Dfi_sa;

%y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +  (1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) +
(1/2)*Cg0*Dfi_sa*(-l2*sin(Dfi0)+l4*cos(Dfi0)) +(1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0))
+(1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))
  DyDx(1,1) = DyDxCore(1,1) + (1/2)*Dfi_ca*(-l1*sD+l3*cD) + (1/2)*Dfi_sa*(-l2*sD+l4*cD);
  DyDx(1,2) = DyDxCore(1,2) - (1/2)*CgDfi_ca*(l1*cd+l3*sd) - (1/2)*CgDfi_sa*(l2*cd+l4*sd) +(1/2)*Cg_ca*(-
l1*sd+l3*cd) + (1/2)*Cg_sa*(-l2*sd+l4*cd);
  DyDx(1,7) = (1/2)*Cg*(-l1*sD+l3*cD);
  DyDx(1,8) = (1/2)*Cg*(-l2*sD+l4*cD);

%y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) + Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
  DyDx(2,2) = DyDxCore(2,2) + Cg_t*DyDxCore(6,1);
  DyDx(2,3) = DyDxCore(2,3) + Cg_t*cD;
  DyDx(2,4) = DyDxCore(2,4) + Cg_t*sD;
  DyDx(2,9) = DyDxCore(1,1);

%y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0)) +
Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
  DyDx(3,1) = DyDxCore(3,1) + Dfi_ca*DyDxCore(6,1);
  DyDx(3,2) = DyDxCore(3,2) - CgDfi_ca*DyDxCore(1,1) + Cg_ca*DyDxCore(6,1);
  DyDx(3,3) = DyDxCore(3,3) - CgDfi_ca*sD + Cg_ca*cD;
  DyDx(3,4) = DyDxCore(3,4) + CgDfi_ca*cD + Cg_ca*sD;
  DyDx(3,7) = Cg*DyDxCore(6,1);
  DyDx(3,10) = DyDxCore(1,1);

%y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0)) +
Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
  DyDx(4,1) = DyDxCore(4,1) + Dfi_sa*DyDxCore(6,1);
  DyDx(4,2) = DyDxCore(4,2) - CgDfi_sa*DyDxCore(1,1) + Cg_sa*DyDxCore(6,1);
  DyDx(4,3) = DyDxCore(4,3) - CgDfi_sa*sD + Cg_sa*cD;;
  DyDx(4,4) = DyDxCore(4,4) + CgDfi_sa*cD + Cg_sa*sD;
  DyDx(4,8) = Cg*DyDxCore(6,1);
  DyDx(4,11) = DyDxCore(1,1);

%y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) - (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) -
(1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0)) +(1/2)*Cg_ca(l3*cos(Dfi0)+l1*sin(Dfi0))+(1/2)*Cg_sa(l4*cos(Dfi0)-
l2*sin(Dfi0))
  DyDx(6,1) = DyDxCore(6,1) - (1/2)*Dfi_ca*(l1*cD+l3*sD) - (1/2)*Dfi_sa*(l2*cD+l4*sD);
  DyDx(6,2) = DyDxCore(6,2) - (1/2)*CgDfi_ca*(-l1*sd+l3*cd) - (1/2)*CgDfi_sa*(-l2*sd+l4*cd) +(1/2)*Cg_ca*(-
l3*sd+l1*cd) - (1/2)*Cg_sa*(l4*sd+l2*cd);
  DyDx(6,7) = - (1/2)*Cg*(l1*cD+l3*sD);
  DyDx(6,8) = - (1/2)*Cg*(l2*cD+l4*sD);

%y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))  + Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
  DyDx(7,2) = DyDxCore(7,2) - Cg_t*DyDxCore(1,1);
  DyDx(7,3) = DyDxCore(7,3) - Cg_t*sD;
  DyDx(7,4) = DyDxCore(7,4) + Cg_t*cD;
  DyDx(7,9) = DyDxCore(6,1);

%y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) - Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0)) + Cg_ca*(-
NS*sin(Dfi0)+EW*cos(Dfi0))
  DyDx(8,1) = DyDxCore(8,1) - Dfi_ca*DyDxCore(1,1);
  DyDx(8,2) = DyDxCore(8,2) - CgDfi_ca*DyDxCore(6,1) - Cg_ca*DyDxCore(1,1);
  DyDx(8,3) = DyDxCore(8,3) - CgDfi_ca*cD - Cg_ca*sD;
  DyDx(8,4) = DyDxCore(8,4) - CgDfi_ca*sD + Cg_ca*cD;
  DyDx(8,7) = - Cg*DyDxCore(1,1);
  DyDx(8,10) = DyDxCore(6,1);

%y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) - Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) + Cg_sa*(-
NS*sin(Dfi0)+EW*cos(Dfi0))
  DyDx(9,1) = DyDxCore(9,1) - Dfi_sa*DyDxCore(1,1);
```

```
DyDx(9,2) = DyDxCore(9,2) - CgDfi_sa*DyDxCore(6,1) - Cg_sa*DyDxCore(1,1);
DyDx(9,3) = DyDxCore(9,3) - CgDfi_sa*cD - Cg_sa*sD;
DyDx(9,4) = DyDxCore(9,4) - CgDfi_sa*sD + Cg_sa*cD;
DyDx(9,8) = - Cg*DyDxCore(1,1);
DyDx(9,11) = DyDxCore(6,1);

%y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) + Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))

DyDx(11,1) =  Dfi_ca*R2;;
DyDx(11,2) =  -CgDfi_ca*R1 + Cg_t*L2 + Cg_ca*R2;
DyDx(11,5) =  -CgDfi_ca*sD + Cg_ca*cD;
DyDx(11,6) =   CgDfi_ca*cD + Cg_ca*sD;
DyDx(11,7) =   Cg*R2;
DyDx(11,9) =  L1;
DyDx(11,10) = R1;

%y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  + Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) + Cg_ca*(-
RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
DyDx(12,1) =  -Dfi_ca*R1;
DyDx(12,2) =  -CgDfi_ca*R2 - Cg_t*L1 - Cg_ca*R1;
DyDx(12,5) =  -CgDfi_ca*cD - Cg_ca*sD;
DyDx(12,6) =  -CgDfi_ca*sD + Cg_ca*cD;
DyDx(12,7) =  -Cg*R1;
DyDx(12,9) = L2;
DyDx(12,10) = R2;

%y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0)
DyDx(13,1) =  -Ao*Dfi_ca*sD;
DyDx(13,2) =  -Ao*(CgDfi_ca*cD + Cg_ca*sD);
DyDx(13,7) =  -Cg*Ao*sD;
DyDx(13,10) =  Ao*cD;

%y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0)
DyDx(14,1) =  -Ao*Dfi_ca*cD;
DyDx(14,2) =  Ao*(CgDfi_ca*sD - Cg_ca*cD);
DyDx(14,7) =  -Cg*Ao*cD;
DyDx(14,10) = -Ao*sD;

%y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) + Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))

DyDx(15,1) =  Dfi_sa*R2;
DyDx(15,2) = -CgDfi_sa*R1 + Cg_t*L4 + Cg_sa*R2;
DyDx(15,5) = -CgDfi_sa*sD + Cg_sa*cD;
DyDx(15,6) =  CgDfi_sa*cD +Cg_sa*sD;
DyDx(15,8) =  Cg*R2;
DyDx(15,9) = L3;
DyDx(15,11) = R1;

%y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  + Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) + Cg_sa*(-
RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
DyDx(16,1) = -Dfi_sa*R1;
DyDx(16,2) = -CgDfi_sa*R2 - Cg_t*L3 - Cg_sa*R1;
DyDx(16,5) = -CgDfi_sa*cD - Cg_sa*sD;
DyDx(16,6) = -CgDfi_sa*sD + Cg_sa*cD;
DyDx(16,8) = -Cg*R1;
DyDx(16,9) = L4;
DyDx(16,11) = R2;

%y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0)
DyDx(17,1) =  Ao*Dfi_sa*sD;
```

```
  DyDx(17,2) = -Ao*(CgDfi_sa*cD + Cg_sa*sD);
  DyDx(17,8) = -Cg*Ao*sD;
  DyDx(17,11) = Ao*cD;

 %y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0)
  DyDx(18,1) =  -Ao*Dfi_sa*cD;
  DyDx(18,2) =   Ao*(CgDfi_sa*sD - Cg_sa*cD);
  DyDx(18,8) =  -Cg*Ao*sD;
  DyDx(18,11) = Ao*sD;

 %y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
  DyDx(19,2) = Cg_t*R2;
  DyDx(19,5) = Cg_t*cD;
  DyDx(19,6) = Cg_t*sD;
  DyDx(19,9) = R1;

 %y(20) = Cg_t*Ao*cos(Dfi0)
  DyDx(20,2) = -Ao*Cg_t*sD;
  DyDx(20,9) =  Ao*cD;

 %y(21) = -Cg_t*Ao*sin(Dfi0)
  DyDx(21,2) = -Ao*Cg_t*cD;
  DyDx(20,9) = -Ao*sD;


elseif (model==2)


%   Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Dfi_ca, Dfi_sa, Cg_t, Cg_ca, Cg_sa, Cg_co, Cg_so]';

  Dfi_ca = Parameters(7);
  Dfi_sa = Parameters(8);
  Cg_t =  Parameters(9);
  Cg_ca =  Parameters(10);
  Cg_sa =  Parameters(11);
  Cg_co =  Parameters(12);
  Cg_so =  Parameters(13);

  CgDfi_ca = Cg*Dfi_ca;
  CgDfi_sa = Cg*Dfi_sa;


 %y(1)= Cg0*(NS*cos(Dfi0)+EW*sin(Dfi0)) +
 %(1/2)*Cg0*Dfi_ca*(-l1*sin(Dfi0)+l3*cos(Dfi0)) + (1/2)*Cg0*Dfi_sa*(-
l2*sin(Dfi0)+l4*cos(Dfi0))+(1/2)*Cg_ca*(l1*cos(Dfi0)+l3*sin(Dfi0))+(1/2)*Cg_sa*(l2*cos(Dfi0)+l4*sin(Dfi0))+(1/2
)*Ao*Cg_co*Cos(Dfi0)
  DyDx(1,1) = DyDxCore(1,1) + (1/2)*Dfi_ca*(-l1*sD+l3*cD) + (1/2)*Dfi_sa*(-l2*sD+l4*cD);
  DyDx(1,2) = DyDxCore(1,2) - (1/2)*CgDfi_ca*(l1*cd+l3*sd) - (1/2)*CgDfi_sa*(l2*cd+l4*sd) +(1/2)*Cg_ca*(-
l1*sd+l3*cd) + (1/2)*Cg_sa*(-l2*sd+l4*cd);
  DyDx(1,7) = (1/2)*Cg*(-l1*sD+l3*cD);
  DyDx(1,8) = (1/2)*Cg*(-l2*sD+l4*cD);
  DyDx(1,10) = (1/2)*(l1*cD+l3*sD);
  DyDx(1,11) = (1/2)*(l2*cD+l4*sD);
  DyDx(1,12) = (1/2)* Ao*cD;

 %y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) + Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
  DyDx(2,2) = DyDxCore(2,2) + Cg_t*DyDxCore(6,1);
  DyDx(2,3) = Cg_t*cD;
  DyDx(2,4) = Cg_t*sD;
  DyDx(2,9) = DyDxCore(1,1);
```

```
  %y(3)= Cg0*(l1*cos(Dfi0) + l3*sin(Dfi0)) + Cg0*Dfi_ca*(-NS*sin(Dfi0)+EW*cos(Dfi0)) +
Cg_ca*(NS*cos(Dfi0)+EW*sin(Dfi0))
   DyDx(3,1) = DyDxCore(3,1) + Dfi_ca*DyDxCore(6,1);
   DyDx(3,2) = DyDxCore(3,2) - CgDfi_ca*DyDxCore(1,1) + Cg_ca*DyDxCore(6,1);
   DyDx(3,3) = DyDxCore(3,3) - CgDfi_ca*sD + Cg_ca*cD;
   DyDx(3,4) = DyDxCore(3,4) + CgDfi_ca*cD + Cg_ca*sD;
   DyDx(3,7) = Cg*DyDxCore(6,1);
   DyDx(3,10) = DyDxCore(1,1);

  %y(4)= Cg0*(l2*cos(Dfi0) + l4*sin(Dfi0)) + Cg0*Dfi_sa*(-NS*sin(Dfi0)+EW*cos(Dfi0)) +
Cg_sa*(NS*cos(Dfi0)+EW*sin(Dfi0))
   DyDx(4,1) = DyDxCore(4,1) + Dfi_sa*DyDxCore(6,1);
   DyDx(4,2) = DyDxCore(4,2) - CgDfi_sa*DyDxCore(1,1) + Cg_sa*DyDxCore(6,1);
   DyDx(4,3) = DyDxCore(4,3) - CgDfi_sa*sD + Cg_sa*cD;;
   DyDx(4,4) = DyDxCore(4,4) + CgDfi_sa*cD + Cg_sa*sD;
   DyDx(4,8) = Cg*DyDxCore(6,1);
   DyDx(4,11) = DyDxCore(1,1);

  %y(6)= Cg0*(-NS*sin(Dfi0)+EW*cos(Dfi0)) - (1/2)*Cg0*Dfi_ca*(l1*cos(Dfi0)+l3*sin(Dfi0)) -
(1/2)*Cg0*Dfi_sa*(l2*cos(Dfi0)+l4*sin(Dfi0)) +(1/2)*Cg_ca(l3*cos(Dfi0)+l1*sin(Dfi0))+(1/2)*Cg_sa(l4*cos(Dfi0)-
l2*sin(Dfi0))
   DyDx(6,1) = DyDxCore(6,1) - (1/2)*Dfi_ca*(l1*cD+l3*sD) - (1/2)*Dfi_sa*(l2*cD+l4*sD);
   DyDx(6,2) = DyDxCore(6,2) - (1/2)*CgDfi_ca*(-l1*sd+l3*cd) - (1/2)*CgDfi_sa*(-l2*sd+l4*cd) +(1/2)*Cg_ca*(-
l3*sd+l1*cd) - (1/2)*Cg_sa*(l4*sd+l2*cd);
   DyDx(6,7) = - (1/2)*Cg*(l1*cD+l3*sD);
   DyDx(6,8) = - (1/2)*Cg*(l2*cD+l4*sD);
   DyDx(6,10) = (1/2)*(l3*cD+l1*sD);
   DyDx(6,11) = (1/2)*(l4*cD-l2*sD);
   DyDx(6,12) = -(1/2)*Ao*sD;


  %y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))  + Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
   DyDx(7,2) = DyDxCore(7,2) - Cg_t*DyDxCore(1,1);
   DyDx(7,3) = DyDxCore(7,3) - Cg_t*sD;
   DyDx(7,4) = DyDxCore(7,4) + Cg_t*cD;
   DyDx(7,9) = DyDxCore(6,1);

  %y(8)= Cg0*(-l1*sin(Dfi0) + l3*cos(Dfi0)) - Cg0*Dfi_ca*(NS*cos(Dfi0)+EW*sin(Dfi0)) + Cg_ca*(-
NS*sin(Dfi0)+EW*cos(Dfi0))
   DyDx(8,1) = DyDxCore(8,1) - Dfi_ca*DyDxCore(1,1);
   DyDx(8,2) = DyDxCore(8,2) - CgDfi_ca*DyDxCore(6,1) - Cg_ca*DyDxCore(1,1);
   DyDx(8,3) = -CgDfi_ca*cD - Cg_ca*sD;
   DyDx(8,4) = -CgDfi_ca*sD + Cg_ca*cD;
   DyDx(8,7) = - Cg*DyDxCore(1,1);
   DyDx(8,10) = DyDxCore(6,1);

  %y(9)= Cg0*(-l2*sin(Dfi0) + l4*cos(Dfi0)) - Cg0*Dfi_sa*(NS*cos(Dfi0)+EW*sin(Dfi0)) + Cg_sa*(-
NS*sin(Dfi0)+EW*cos(Dfi0))
   DyDx(9,1) = DyDxCore(9,1) - Dfi_sa*DyDxCore(1,1);
   DyDx(9,2) = DyDxCore(9,2) - CgDfi_sa*DyDxCore(6,1) - Cg_sa*DyDxCore(1,1);
   DyDx(9,3) = DyDxCore(9,3) - CgDfi_sa*cD - Cg_sa*sD;
   DyDx(9,4) = DyDxCore(9,4) - CgDfi_sa*sD + Cg_sa*cD;
   DyDx(9,8) = - Cg*DyDxCore(1,1);
   DyDx(9,11) = DyDxCore(6,1);

  %y(10)= -Cg0*Ao*sin(Dfi0)+Cg_co*(EW*Cos(Dfio)-NS*Sin(Dfi0))
  DyDx(10,2) = DyDxCore(10,2) + Cg_co*(-EW*sD-NS*cD);
  DyDx(10,3) = -Cg_co*sD;
  DyDx(10,4) = Cg_co*cD;
  DyDx(10,12)= DyDxCore(6,1);
```

```
  %y(11)= Cg0*Dfi_ca*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) + Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0)) +
Cg_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
  R1 = Rg*cD + Rm*sD;
  R2 = -Rg*sD + Rm*cD;
  L1 = l1*cD + l3*sD;
  L2 = -l1*sD + l3*cD;
   DyDx(11,1) =  Dfi_ca*R2;;
   DyDx(11,2) =  -CgDfi_ca*R1 + Cg_t*L2 + Cg_ca*R2;
   DyDx(11,5) =  -CgDfi_ca*sD + Cg_ca*cD;
   DyDx(11,6) =   CgDfi_ca*cD + Cg_ca*sD;
   DyDx(11,7) =   Cg*R2;
   DyDx(11,9) =  L1;
   DyDx(11,10) = R1;

  %y(12)= -Cg0*Dfi_ca*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  + Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0)) + Cg_ca*(-
RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
   DyDx(12,1) =  -Dfi_ca*R1;
   DyDx(12,2) =  -CgDfi_ca*R2 - Cg_t*L1 - Cg_ca*R1;
   DyDx(12,5) =  -CgDfi_ca*cD - Cg_ca*sD;
   DyDx(12,6) =  -CgDfi_ca*sD + Cg_ca*cD;
   DyDx(12,7) =  -Cg*R1;
   DyDx(12,9) = L2;
   DyDx(12,10) = R2;


  %%y(13)= -Cg0*Dfi_ca*Ao*sin(Dfi0) + Cg_ca*Ao*cos(Dfi0) + Cg_co(l1*cos(Dfi0)+l3*sin(Dfi0))
   DyDx(13,1) =  -Ao*Dfi_ca*sD;
   DyDx(13,2) =  -Ao*(CgDfi_ca*cD + Cg_ca*sD) + Cg_co*L2;
   DyDx(13,7) =  -Cg*Ao*sD;
   DyDx(13,10) =  Ao*cD;
   DyDx(13,12) = L1;


  %y(14)= -Cg0*Dfi_ca*Ao*cos(Dfi0) - Cg_ca*Ao*sin(Dfi0) + Cg_co(l3*cos(Dfi0)-l1*sin(Dfi0))
   DyDx(14,1) =  -Ao*Dfi_ca*cD;
   DyDx(14,2) =  Ao*(CgDfi_ca*sD - Cg_ca*cD) - Cg_co*L1;
   DyDx(14,7) =  -Cg*Ao*cD;
   DyDx(14,10) = -Ao*sD;
  DyDx(14,12) =  L2;

  %y(15)= Cg0*Dfi_sa*(-RgTa*sin(Dfi0) + RmTa*cos(Dfi0)) + Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0)) +
Cg_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
  L3 =  l2*cD + l4*sD;
  L4 = -l2*sD + l4*cD;
   DyDx(15,1) =  Dfi_sa*R2;
   DyDx(15,2) = -CgDfi_sa*R1 + Cg_t*L4 + Cg_sa*R2;
   DyDx(15,5) = -CgDfi_sa*sD + Cg_sa*cD;
   DyDx(15,6) =  CgDfi_sa*cD +Cg_sa*sD;
   DyDx(15,8) =  Cg*R2;
   DyDx(15,9) = L3;
   DyDx(15,11) = R1;

  %y(16)= -Cg0*Dfi_sa*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))  + Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0)) + Cg_sa*(-
RgTa*sin(Dfi0) + RmTa*cos(Dfi0))
   DyDx(16,1) = -Dfi_sa*R1;
   DyDx(16,2) = -CgDfi_sa*R2 - Cg_t*L3 - Cg_sa*R1;
   DyDx(16,5) = -CgDfi_sa*cD - Cg_sa*sD;
   DyDx(16,6) = -CgDfi_sa*sD + Cg_sa*cD;
   DyDx(16,8) = -Cg*R1;
   DyDx(16,9) = L4;
   DyDx(16,11) = R2;
```

```
%y(17)= -Cg0*Dfi_sa*Ao*sin(Dfi0) + Cg_sa*Ao*cos(Dfi0) + Cg_co*(l2*Cos(Dfi0)+l4*sin(Dfi0))
 DyDx(17,1) =  -Ao*Dfi_sa*sD;
 DyDx(17,2) = -Ao*(CgDfi_sa*cD + Cg_sa*sD) +Cg_co*L4;
 DyDx(17,8) = -Cg*Ao*sD;
 DyDx(17,11) = Ao*cD;
 DyDx(17,11) = L3;

%y(18)= -Cg0*Dfi_sa*Ao*cos(Dfi0) - Cg_sa*Ao*sin(Dfi0) + Cg_co(l4*cos(Dfi0)-l2*sin(Dfi0))
 DyDx(18,1) =  -Ao*Dfi_sa*cD;
 DyDx(18,2) =   Ao*(CgDfi_sa*sD - Cg_sa*cD) - Cg_co*L3;
 DyDx(18,8) =  -Cg*Ao*sD;
 DyDx(18,11) = -Ao*sD;
 DyDx(18,12) = L4;

%y(19)= Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
 DyDx(19,2) = Cg_t*R2;
 DyDx(19,5) = Cg_t*cD;
 DyDx(19,6) = Cg_t*sD;
 DyDx(19,9) = R1;

 %y(20) = Cg_t*Ao*cos(Dfi0)+Cg_co*(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
 DyDx(20,2) = -Ao*Cg_t*sD + Cg_co*R2;
 DyDx(20,5) = Cg_co*cD;
 DyDx(20,6) = Cg_co*sD;
 DyDx(20,9) =  Ao*cD;
 DyDx(20,12) = R1;

 %y(21) = -Cg_t*Ao*sin(Dfi0)+ Cg_co(RmTa*Cos(Dfi0)-RgTa*Sin(Dfi0))
 DyDx(21,2) = -Ao*Cg_t*cD-Cg_co*R1;
 DyDx(21,5) = -Cg_co*sD;
 DyDx(21,6) = Cg_co*cD;
 DyDx(21,9) = -Ao*sD;
 DyDx(21,12) = R2;


%y(22)= Cg_so(NS*cos(Dfi0)+ EW*sin(Dfi0))
 DyDx(22,2) = Cg_so*DyDxCore(6,1);
 DyDx(22,3) = Cg_so*cD;
 DyDx(22,4) = Cg_so*sD;
 DyDx(22,13) = DyDxCore(1,1);

 %y(23)= Cg_so(EW*cos(Dfi0)- NS*sin(Dfi0))
 DyDx(23,2) = -Cg_so*DyDxCore(1,1);
 DyDx(23,3) = -Cg_so*sD;
 DyDx(23,4) = Cg_so*cD;
 DyDx(23,13)= DyDxCore(6,1);

 %y(24)= Cg_so(RgTa*cos(Dfi0)+RmTa*sin(Dfi0))
 DyDx(24,2) = Cg_so*R2;
 DyDx(24,5) = Cg_so*cD;
 DyDx(24,6) = Cg_so*sD;
 DyDx(24,13) = R1;

 %y(25)= Cg_so(-RgTa*sin(Dfi0)+RmTa*cos(Dfi0))
 DyDx(25,2) = -Cg_so*R1;
 DyDx(25,5) = -Cg_so*sD;
 DyDx(25,6) = Cg_so*cD;
 DyDx(25,13) = R2;

 %y(26)= (1/2)*Cg_co*Ao*cos(Dfi0)
 DyDx(26,2) = -(1/2)*Cg_co*Ao*sD;
```

```
    DyDx(26,12) = (1/2)*Ao*cD;

    %y(27)= -(1/2)*Cg_co*Ao*sin(Dfi0)
    DyDx(27,2) = -(1/2)*Cg_co*Ao*cD;
    DyDx(27,12) = -(1/2)*Ao*sD;

    %y(28)= (1/2)*Cg_so*Ao*cos(Dfi0)
    DyDx(28,2) = -(1/2)*Cg_co*Ao*sD;
    DyDx(28,13) = (1/2)*Ao*cD;

    %y(29)= -(1/2)*Cg_so*Ao*sin(Dfi0)
    DyDx(29,2) = -(1/2)*Cg_so*Ao*cD;
    DyDx(29,13) = -(1/2)*Ao*sD;

    %y(30)= Cg_so*(l1*cos(Dfi0)+l3*sin(Dfi0))
    DyDx(30,2) = Cg_so*L2;
    DyDx(30,13) = L1;

    %y(31)= Cg_so*(l2*cos(Dfi0)+l4*sin(Dfi0))
    DyDx(31,2) = Cg_so*L4;
    DyDx(31,13)= L3;

    %y(32)= Cg_so*(l3*cos(Dfi0)-l1*sin(Dfi0))
    DyDx(32,2) = -Cg_so*L1;
    DyDx(32,13) = L2;

    %y(33)= Cg_so*(l4*cos(Dfi0)-l2*sin(Dfi0))
    DyDx(33,2) = -Cg_so*L3;
    DyDx(33,13) = L4;

%=============================================================================
==========================
elseif (model==21)


%   Parameters = [Cg0, Dfi0, NS, EW, Rg, Rm, Cg_t]';

    Cg_t = Parameters(7);


% y(2)= Cg0*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0)) + Cg_t*(NS*cos(Dfi0)+EW*sin(Dfi0))
    DyDx(2,2) = DyDxCore(2,2) + Cg_t*N2;
    DyDx(2,3) = DyDxCore(2,3) + Cg_t*cD;
    DyDx(2,4) = DyDxCore(2,4) + Cg_t*sD;
    DyDx(2,7) = DyDxCore(2,4) + N1;

% y(7)= Cg0*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0)) + Cg_t*(-NS*sin(Dfi0)+EW*cos(Dfi0))
    DyDx(7,2) = DyDxCore(7,2) - Cg_t*N1;
    DyDx(7,3) = DyDxCore(7,3) - Cg_t*sD;
    DyDx(7,4) = DyDxCore(7,4) + Cg_t*cD;
    DyDx(7,7) = DyDxCore(2,4) + N2;

% y(11) = Cg_t*(RgTa*cos(Dfi0)+ RmTa*sin(Dfi0))
    DyDx(11,2) = Cg_t*R2;
    DyDx(11,5) = Cg_t*cD;
    DyDx(11,6) = Cg_t*sD;
    DyDx(11,7) = R1;

% y(12) = Cg_t*(l1*cos(Dfi0) + l3*sin(Dfi0))
    DyDx(12,2) = Cg_t*L2;
    DyDx(12,7) = L1;
```

```
% y(13)= Cg_t*(l2*cos(Dfi0) + l4*sin(Dfi0))
  DyDx(13,2) = Cg_t*L4;
  DyDx(13,7) = L3;

% y(14)= Cg_t*Ao*cos(Dfi0)
  DyDx(14,2) = -Cg_t*Ao*sD;
  DyDx(13,7) = Ao*cD;

% y(15)= Cg_t*(-RgTa*sin(Dfi0)+ RmTa*cos(Dfi0))
  DyDx(15,2) = -Cg_t*R1;
  DyDx(15,5) = -Cg_t*sD;
  DyDx(15,6) = Cg_t*cD;
  DyDx(15,7) = R2;

% y(16) = Cg_t*(-l1*sin(Dfi0) + l3*cos(Dfi0))
  DyDx(16,2) = -Cg_t*L1;
  DyDx(12,7) = L2;

% y(17)= Cg_t*(-l2*sin(Dfi0) + l4*cos(Dfi0))
  DyDx(17,2) = -Cg_t*L3;
  DyDx(17,7) = L4;

% y(18)= -Cg_t*Ao*sin(Dfi0)
  DyDx(18,2) = Cg_t*Ao*cD;
  DyDx(13,7) = -Ao*sD;

%==========================================================================
elseif (model==32)

  Dfi_ca = Parameters(7);
  Dfi_sa = Parameters(8);
  CgDfi_ca = Cg*Dfi_ca;
  CgDfi_sa = Cg*Dfi_sa;

  DyDx(1,1) = DyDxCore(1,1) + (1/2)*Dfi_ca*(-l1*sD+l3*cD) + (1/2)*Dfi_sa*(-l2*sD+l4*cD);
  DyDx(1,2) = DyDxCore(1,2) - (1/2)*CgDfi_ca*(l1*cd+l3*sd) - (1/2)*CgDfi_sa*(l2*cd+l4*sd) ;
  DyDx(1,7) = (1/2)*Cg*(-l1*sD+l3*cD);
  DyDx(1,8) = (1/2)*Cg*(-l2*sD+l4*cD);

  DyDx(3,1) = DyDxCore(3,1) + Dfi_ca*(-NS*sD+EW*cD);
  DyDx(3,2) = DyDxCore(3,2) - CgDfi_ca*(NS*cD+EW*sD);
  DyDx(3,3) = DyDxCore(3,3) - CgDfi_ca*sD;
  DyDx(3,4) = DyDxCore(3,4) + CgDfi_ca*cD;
  DyDx(3,7) = Cg*(-NS*sD+EW*cD);

  DyDx(4,1) = DyDxCore(4,1) + Dfi_sa*(-NS*sD+EW*cD);
  DyDx(4,2) = DyDxCore(4,2) - CgDfi_sa*(NS*cD+EW*sD);
  DyDx(4,3) = DyDxCore(4,3) - CgDfi_sa*sD;
  DyDx(4,4) = DyDxCore(4,4) + CgDfi_sa*cD;
  DyDx(4,8) = Cg*(-NS*sD+EW*cD);

  DyDx(6,1) = DyDxCore(6,1) - (1/2)*Dfi_ca*(l1*cD+l3*sD) - (1/2)*Dfi_sa*(l2*cD+l4*sD);
  DyDx(6,2) = DyDxCore(6,2) - (1/2)*CgDfi_ca*(-l1*sd+l3*cd) - (1/2)*CgDfi_sa*(-l2*sd+l4*cd);
  DyDx(6,7) = - (1/2)*Cg*(l1*cD+l3*sD);
  DyDx(6,8) = - (1/2)*Cg*(l2*cD+l4*sD);

  DyDx(8,1) = DyDxCore(8,1) - Dfi_ca*(NS*cD+EW*sD);
  DyDx(8,2) = DyDxCore(8,2) - CgDfi_ca*(-NS*sD+EW*cD);
  DyDx(8,3) = DyDxCore(8,3) - CgDfi_ca*cD;
  DyDx(8,4) = DyDxCore(8,4) - CgDfi_ca*sD;
  DyDx(8,7) = - Cg*(NS*cD+EW*sD);
```

```
    DyDx(9,1) = DyDxCore(9,1) - Dfi_sa*(NS*cD+EW*sD);
    DyDx(9,2) = DyDxCore(9,2) - CgDfi_sa*(-NS*sD+EW*cD);
    DyDx(9,3) = DyDxCore(9,3) - CgDfi_sa*cD;
    DyDx(9,4) = DyDxCore(9,4) - CgDfi_sa*sD;
    DyDx(9,8) = - Cg*(NS*cD+EW*sD);

    DyDx(11,1) =  Dfi_ca*(-Rg*sD + Rm*cD);
    DyDx(11,2) =  -CgDfi_ca*(Rg*cD + Rm*sD);
    DyDx(11,5) =  -CgDfi_ca*sD;
    DyDx(11,6) =   CgDfi_ca*cD;
    DyDx(11,7) =   Cg*(-Rg*sD + Rm*cD);

    DyDx(12,1) =  -Dfi_ca*(Rg*cD + Rm*sD);
    DyDx(12,2) =  -CgDfi_ca*(-Rg*sD + Rm*cD);
    DyDx(12,5) =  -CgDfi_ca*cD;
    DyDx(12,6) =  -CgDfi_ca*sD;
    DyDx(12,7) =  -Cg*(Rg*cD + Rm*sD);

    DyDx(13,1) =  Dfi_ca*Ao*sD;
    DyDx(13,2) =  CgDfi_ca*Ao*cD;
    DyDx(13,7) =  Cg*Ao*sD;

    DyDx(14,1) =  -Dfi_ca*Ao*cD;
    DyDx(14,2) =   CgDfi_ca*Ao*sD;
    DyDx(14,7) =  -Cg*Ao*cD;

    DyDx(15,1) =  Dfi_sa*(-Rg*sD + Rm*cD);
    DyDx(15,2) =  -CgDfi_sa*(Rg*cD + Rm*sD);
    DyDx(15,5) =  -CgDfi_sa*sD;
    DyDx(15,6) =   CgDfi_sa*cD;
    DyDx(15,8) =   Cg*(-Rg*sD + Rm*cD);

    DyDx(16,1) =  -Dfi_sa*(Rg*cD + Rm*sD);
    DyDx(16,2) =  -CgDfi_sa*(-Rg*sD + Rm*cD);
    DyDx(16,5) =  -CgDfi_sa*cD;
    DyDx(16,6) =  -CgDfi_sa*sD;
    DyDx(16,8) =  -Cg*(Rg*cD + Rm*sD);

    DyDx(17,1) =  Dfi_sa*Ao*sD;
    DyDx(17,2) =  -CgDfi_sa*Ao*cD;
    DyDx(17,8) =  -Cg*Ao*sD;

    DyDx(18,1) =  -Dfi_sa*Ao*cD;
    DyDx(18,2) =   CgDfi_sa*Ao*sD;
    DyDx(18,8) =  -Cg*Ao*sD;

elseif (model==31)

    Dfi_ca = Parameters(7);
    CgDfi_ca = Cg*Dfi_ca;

    DyDx(1,1) = DyDxCore(1,1) + (1/2)*Dfi_ca*(-l1*sD+l3*cD);
    DyDx(1,2) = DyDxCore(1,2) - (1/2)*CgDfi_ca*(l1*cd+l3*sd);
    DyDx(1,7) = (1/2)*Cg*(-l1*sD+l3*cD);

    DyDx(3,1) = DyDxCore(3,1) + Dfi_ca*(-NS*sD+EW*cD);
    DyDx(3,2) = DyDxCore(3,2) - CgDfi_ca*(NS*cD+EW*sD);
    DyDx(3,3) = DyDxCore(3,3) - CgDfi_ca*sD;
    DyDx(3,4) = DyDxCore(3,4) + CgDfi_ca*cD;
    DyDx(3,7) = Cg*(-NS*sD+EW*cD);

    DyDx(6,1) = DyDxCore(6,1) - (1/2)*Dfi_ca*L1;
```

```
    DyDx(6,2) = DyDxCore(6,2) - (1/2)*CgDfi_ca*L2;
    DyDx(6,7) = - (1/2)*Cg*L1;

    DyDx(8,1) = DyDxCore(8,1) - Dfi_ca*(NS*cD+EW*sD);
    DyDx(8,2) = DyDxCore(8,2) - CgDfi_ca*(-NS*sD+EW*cD);
    DyDx(8,3) = DyDxCore(8,3) - CgDfi_ca*cD;
    DyDx(8,4) = DyDxCore(8,4) - CgDfi_ca*sD;
    DyDx(8,7) = - Cg*(NS*cD+EW*sD);

    DyDx(11,1) =  Dfi_ca*R2;
    DyDx(11,2) = -CgDfi_ca*R1;
    DyDx(11,5) = -CgDfi_ca*sD;
    DyDx(11,6) =  CgDfi_ca*cD;
    DyDx(11,7) =  Cg*R2;

    DyDx(12,1) = -Dfi_ca*R1;
    DyDx(12,2) = -CgDfi_ca*R2;
    DyDx(12,5) = -CgDfi_ca*cD;
    DyDx(12,6) = -CgDfi_ca*sD;
    DyDx(12,7) = -Cg*R1;

    DyDx(13,1) =  Dfi_ca*Ao*sD;
    DyDx(13,2) = -CgDfi_ca*Ao*cD;
    DyDx(13,7) = -Cg*Ao*sD;

    DyDx(14,1) = -Dfi_ca*Ao*cD;
    DyDx(14,2) =  CgDfi_ca*Ao*sD;
    DyDx(14,7) = -Cg*Ao*cD;

end
```