Estimating the frequency-dependence of t-star from spectral ratios

by Bill Menke with input from Ted Dong, April 10-12, 2017

Note that I omit printing the "star" on "t-star", because in my opinion, it confuses the mathematical notation. The model formulas for the "t-star" $t(f)$ and the difference between two "t-star" functions $\Delta t(f) = t_1(f) - t_2(f)$ are:

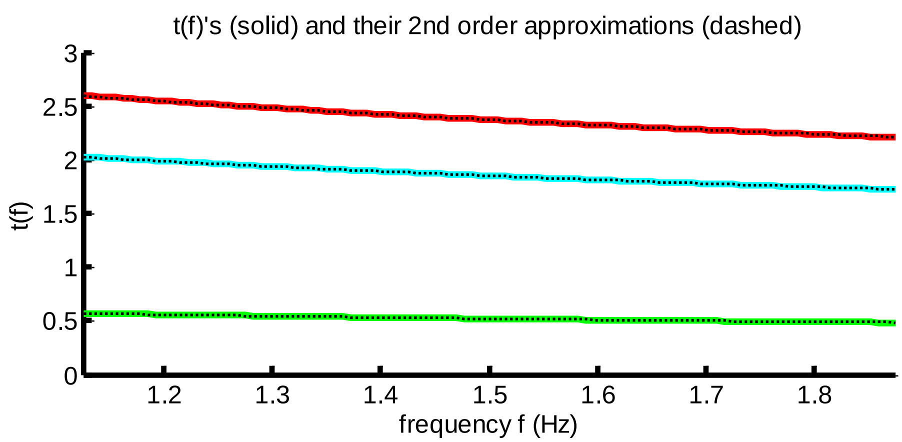
$$t(f) = t_0 \left(\frac{f}{f_0}\right)^{-\alpha} \quad \text{and} \quad \Delta t(f) = \Delta t_0 \left(\frac{f}{f_0}\right)^{-\alpha}$$

Here f_0 is a reference frequency and $\Delta t_0 = \Delta t(f_0\,)$. At the reference frequency, the derivatives of $\Delta t(f)$ are:

$$\left.\frac{d\Delta t}{df}\right|_{f_0} = -\frac{\alpha \Delta t_0}{f_0}\left(\frac{f}{f_0}\right)^{-\alpha-1}\Bigg|_{f_0} = -\frac{\alpha \Delta t_0}{f_0}$$

$$\left.\frac{d^2\Delta t}{df^2}\right|_{f_0} = \frac{\alpha(a+1)\Delta t_0}{f_0^2}\left(\frac{f}{f_0}\right)^{-\alpha-2}\Bigg|_{f_0} = \frac{\alpha(a+1)\Delta t_0}{f_0^2}$$

A second-order Taylor series representation of $\Delta t(f)$ based on these derivatives is very accurate (figure).



The spectral ratio formula is:

$$R(f) = \ln\frac{A_1 \exp(-\pi t_1 f)}{A_2 \exp(-\pi t_2 f)} = c + f\, z(f) \quad \text{with} \quad c = \ln\frac{A_1}{A_2} \text{ and } z(f) = -\pi\, \Delta t(f)$$

Method 1 (for estimating Δt_0 and α from $R(f)$): We first construct a quadratic approximation for $R(f)$ centered at $= f_0$. We need the derivatives:

$$R_0 = c + f_0\, z_0$$

$$\dot{R}_0 = z_0 + f_0\, \dot{z}_0$$

$$\ddot{R}_0 = 2\dot{z}_0 + f_0\,\ddot{z}_0$$

Then

$$R(f) = (c + f_0\,z_0) + (z_0 + f_0\,\dot{z}_0)(f - f_0) + ½(2\dot{z}_0 + f_0\,\ddot{z}_0)(f - f_0)^2$$

Collecting terms of equal order in f, we have

$$R(f) = A + Bf + Cf^2$$

with

$$A = \quad c + f_0\,z_0 - f_0 z_0 - f_0^2\dot{z}_0 + f_0^2\dot{z}_0 + ½f_0^3\ddot{z}_0 = c + ½f_0^3\ddot{z}_0$$

$$B = \; z_0 + f_0\dot{z}_0 - 2f_0\dot{z}_0 - f_0^2\ddot{z}_0 = z_0 - f_0\dot{z}_0 - f_0^2\ddot{z}_0$$

$$C = \dot{z}_0 + ½f_0\,\ddot{z}_0$$

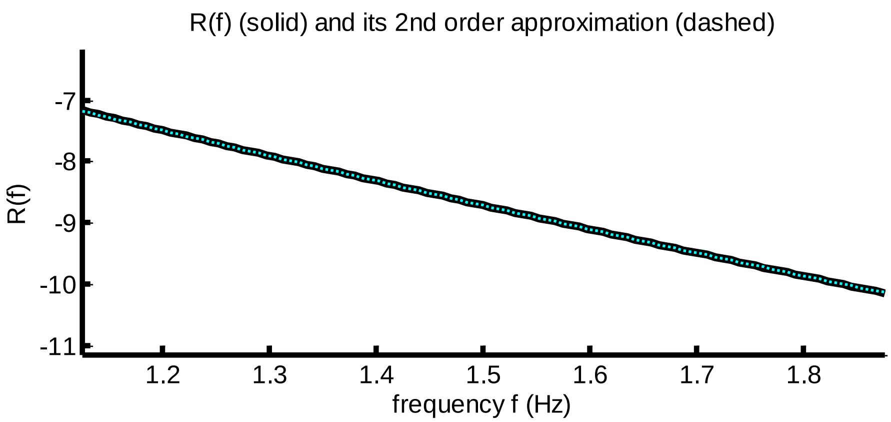
Substituting in the definition of z and its derivatives, we find

$$A = \; c + ½f_0^3\ddot{z}_0 = \; c - ½\pi f_0\alpha(a+1)\Delta t_0$$

$$B = z_0 - f_0\dot{z}_0 - f_0^2\ddot{z}_0 = -\pi\Delta t_0 - \pi\alpha\Delta t_0 + \pi\alpha(a+1)\Delta t_0 = -\pi\Delta t_0[1 + \alpha - \alpha(a+1)]$$

$$C = \; \dot{z}_0 + ½f_0\,\ddot{z}_0 = \; \frac{\pi\alpha\Delta t_0}{f_0} - \frac{\pi\alpha(a+1)\Delta t_0}{2f_0} = \frac{\pi\Delta t_0}{2f_0}\big(2\alpha - \alpha(a+1)\big)$$

The second order approximation for $R(f)$ is very accurate (figure).



The quantities A, B, and C (and their covariance, $\mathbf{C}_{A,B,C}$) can be determined through a least-squares fit to the observed $R(f)$. The unknown parameters α can be found by forming the ratio B/C (and hence eliminating Δt_0):

$$D = -\frac{B}{2f_0 C} = \frac{1 + \alpha - \alpha(a+1)}{2\alpha - \alpha(a+1)}$$

This equation can be rearranged into a quadratic for α:

$$2D\alpha - D\alpha(a+1) = 1 + \alpha - \alpha(a+1)$$

$$2D\alpha - D\alpha^2 - D\alpha = 1 + \alpha - \alpha^2 - \alpha$$

$$-D\alpha^2 + D\alpha = 1 - \alpha^2$$

$$(1-D)\,\alpha^2 + D\alpha - 1 = 0$$

It's solution is:

$$\alpha = \frac{-D \pm \sqrt{D^2 + 4(1-D)}}{2(1-D)} = \frac{-D \pm (D-2)}{2(1-D)} = \begin{cases} 1/(D-1) & \text{for } + \text{sign} \\ 1 & \text{for} - \text{sign} \end{cases}$$

The solution corresponding to the positive sign is the correct one.; the other corresponds to the trivial relation $0 = 0$. Then

$$\alpha^{est} = \left(-\frac{B}{2f_0C} - 1\right)^{-1} = -\left(\frac{2f_0C}{2f_0C} + \frac{B}{2f_0C}\right)^{-1} = -\frac{2f_0C}{B + 2f_0C}$$

and

$$(\Delta t_0)^{est} = \frac{2f_0C}{\pi\big(2\alpha - \alpha(a+1)\big)} = -\frac{(B + 2f_0C)}{\pi(1-\alpha)}$$

The sensitivities (derivatives) needed to calculate the variances are:

$$[S_{11} \quad S_{12} \quad S_{13}] = \begin{bmatrix} \frac{\partial\alpha}{\partial A} & \frac{\partial\alpha}{\partial B} & \frac{\partial\alpha}{\partial C} \end{bmatrix} = \begin{bmatrix} 0, & \frac{2f_0C}{[B + 2f_0C]^2}, & -\frac{2f_0B}{[B + 2f_0C]^2} \end{bmatrix}$$

$$[S_{21} \quad S_{22} \quad S_{23}] = \begin{bmatrix} \frac{\partial\Delta t_0}{\partial A}, & \frac{\partial\Delta t_0}{\partial B}, & \frac{\partial\Delta t_0}{\partial C} \end{bmatrix}$$

$$= \begin{bmatrix} 0, & \frac{-1}{\pi(1-\alpha)} - \frac{(B + 2f_0C)}{\pi(1-\alpha)^2}\frac{\partial\alpha}{\partial B}, & -\frac{2f_0}{\pi(1-\alpha)} - \frac{(B + 2f_0C)}{\pi(1-\alpha)^2}\frac{\partial\alpha}{\partial C} \end{bmatrix}$$

The covariance is then:

$$\mathbf{C}_{\alpha,\Delta t} = \mathbf{S}\,\mathbf{C}_{\mathrm{A,B,C}}\,\mathbf{S}^{\mathrm{T}}$$

I have verified all these formulas numerically. An exemplary calculation gives (for $\sigma_R = 0.01$):

```
a   true 0.321000 est1 0.324704 +/- 0.035034

Dt0 true 1.850000 est1 1.865267 +/- 0.096772
```

Method 2 (due to Ted Dong): Estimates of the parameters α and Δt_0 can be obtained by combining the formula for Δt_0 and $d\Delta t/df|_{f_0}$

$$\alpha = -\frac{f_0}{\Delta t_0}\frac{d\Delta t}{df}\bigg|_{f_0}$$

The spectral ratio $R(f)$ is related to $\Delta t(f)$ by:

$$R(f) = c - \pi\, f\, \Delta t(f)$$

so

$$\Delta t(f) = -\frac{R(f) - c}{\pi\, f}$$

Substituting in the quadratic fit $R(f) = A + Bf + Cf^2$ yields

$$\Delta t(f) = -\frac{(A-c)f^{-1} + B + Cf}{\pi}$$

$$(\Delta t_0)^{est} = -\frac{(A-c)f_0^{-1} + B + Cf_0}{\pi}$$

$$\frac{d\Delta t}{df}\bigg|_{f_0} = \frac{(A-c)f_0^{-2} - C}{\pi}$$

$$\alpha^{est} = -\frac{f_0}{\Delta t_0}\frac{d\Delta t}{df}\bigg|_{f_0} = \frac{(A-c)f_0^{-1} - Cf_0}{(A-c)f_0^{-1} + B + Cf_0}$$

The sensitivities (derivatives) needed to calculate the variances are:

$$\left[\frac{\partial \Delta t_0}{\partial A} \quad \frac{\partial \Delta t_0}{\partial B} \quad \frac{\partial \Delta t_0}{\partial C}\right] = \left[-\frac{1}{\pi f_0}, \quad -\frac{1}{\pi}, \quad -\frac{f_0}{\pi}\right]$$

$$\frac{\partial \alpha}{\partial A} = \frac{f_0^{-1}}{(A-c)f_0^{-1} + B + Cf_0} - \frac{(A-c)f_0^{-2} - C}{[(A-c)f_0^{-1} + B + Cf_0]^2}$$

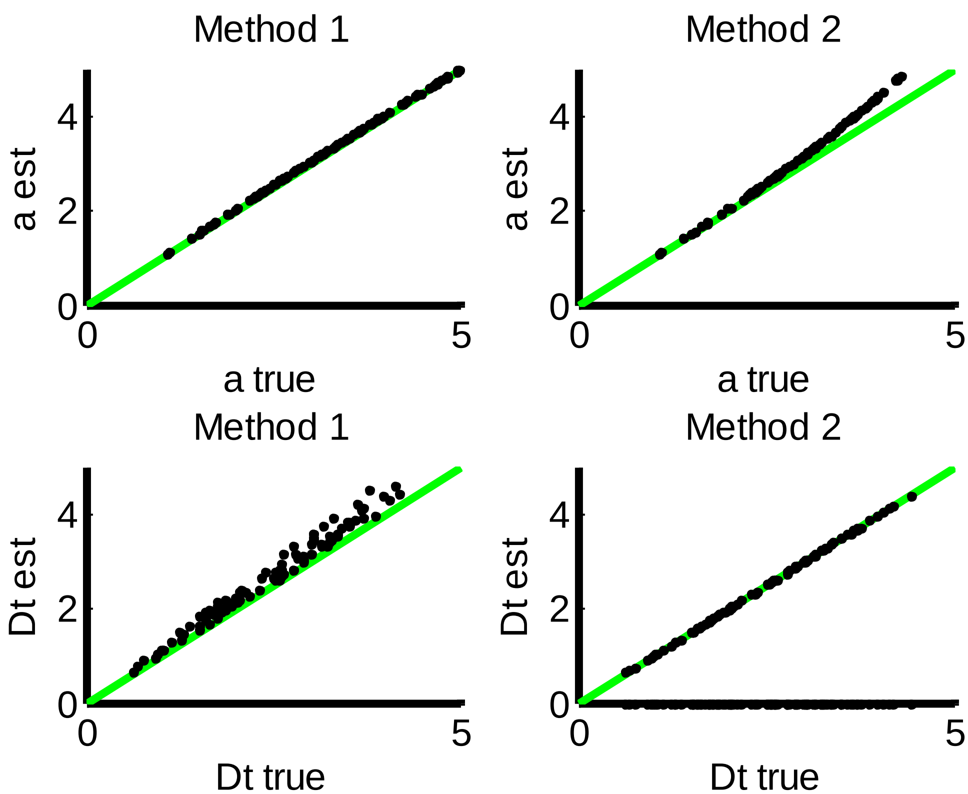
$$\frac{\partial \alpha}{\partial B} = -\frac{(A-c)f_0^{-1} - Cf_0}{[(A-c)f_0^{-1} + B + Cf_0]^2}$$

$$\frac{\partial \alpha}{\partial C} = -\frac{f_0}{(A-c)f_0^{-1} + B + Cf_0} - \frac{[(A-c) - Cf^2]}{[(A-c)f_0^{-1} + B + Cf_0]^2}$$

I have verified all these formulas numerically; they are different from the one derived in Method 1. An exemplary calculation gives:

```
a   true 0.321000 est2 0.319138 +/- 0.001582

Dt0 true 1.850000 est2 1.850019 +/- 0.000634
```



These results indicate that Method 2 is more accurate than Method 1. I think that the issue is that Method 1 relies on a second order approximation whereas Method 1 does not. On the other hand, an advantage of Method 1 is that the solution does not use $(A - c)$, a quantity which may not always be well-determined.

```
clear all
f0 = 1.5;
fmin = 0.75*f0;
fmax = 1.25*f0;
Nf = 101;
Df = (fmax-fmin)/(Nf-1);
f = fmin+Df*[0:Nf-1]';

figure(1);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis([ fmin, fmax, 0, 3 ] );
xlabel('frequency f (Hz)');
ylabel('t(f)');
```

```
title('t(f)''s (solid) and their 2nd order approximations (dashed)');

% exponent
a = 0.321;

% function t(f) at f0
t10 = 2.37;
t20 = 0.52;
Dt0 = t10-t20;

% exact formula for t(f)
t1 = t10*(f/f0).^(-a);
t2 = t20*(f/f0).^(-a);
Dt = t1-t2;

plot( f, t1, 'r-', 'LineWIdth', 4 );
plot( f, t2, 'g-', 'LineWIdth', 4 );
plot( f, Dt, 'c-', 'LineWIdth', 4 );

% first derivative of t(f) at f0
dtdf10 = -a*t10/f0;
dtdf20 = -a*t20/f0;
dDtdf0 = dtdf10 - dtdf20;

% second derivativeof t(f) at f0
d2tdf210 = a*(a+1)*t10/(f0^2);
d2tdf220 = a*(a+1)*t20/(f0^2);
d2Dtdf20 = d2tdf210 - d2tdf220;

% quadratic approximation to t(f)
t1approx = t10 + dtdf10*(f-f0) + 0.5*d2tdf210*((f-f0).^2);
t2approx = t20 + dtdf20*(f-f0) + 0.5*d2tdf220*((f-f0).^2);
Dtapprox = t1approx - t2approx;

plot( f, t1approx, 'k:', 'LineWidth', 2 );
plot( f, t2approx, 'k:', 'LineWIdth', 2 );
plot( f, Dtapprox, 'k:', 'LineWIdth', 2 );

figure(2);
clf;
hold on;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
xlabel('frequency f (Hz)');
ylabel('R(f)');
title('R(f) (solid) and its 2nd order approximation (dashed)');

A10 = 1.0;
A20 = 1.0;
```

```
A1 = A10*exp(-pi*t1.*f);
A2 = A20*exp(-pi*t2.*f);
R = log(A1./A2);
c = log(A10/A20);

axis([ fmin, fmax, min(R)-1, max(R)+1 ] );
plot( f, R, 'k-', 'LineWidth', 5 );

z0 = -pi*Dt0;
zd0 = -pi*dDtdf0;
zdd0 = -pi*d2Dtdf20;

R0 = c + f0*z0;
Rd0 = z0 + f0*zd0;
Rdd0 = 2*zd0 + f0*zdd0;

Rapprox1 =  R0 + Rd0*(f-f0) + Rdd0*(f-f0).^2;
R1max = max( abs(R-Rapprox1)./abs(R) );
fprintf('Error 1: %f\n', R1max );
% plot( f, Rapprox1, 'r-', 'LineWidth', 3 );

A = c + 0.5*(f0^3)*zdd0;
B = z0 -f0*zd0 -(f0^2)*zdd0;
C = zd0 + 0.5*f0*zdd0;

A2 = c - 0.5*pi*f0*a*(a+1)*Dt0;
B2 =  -pi*Dt0*(1+a-a*(a+1));
C2 = (pi*Dt0/(2*f0)) * (2*a - a*(a+1) );

Rapprox2 =  A + B*f + C*(f.^2);
R2max = max( abs(R-Rapprox2)./abs(R) );
fprintf('Error 2: %f\n', R2max );
% plot( f, Rapprox2, 'go', 'LineWidth', 2 );
Rapprox3 =  A2 + B2*f + C2*(f.^2);
R3max = max( abs(R-Rapprox3)./abs(R) );
fprintf('Error 3: %f\n', R3max );
% plot( f, Rapprox3, 'c:', 'LineWidth', 2 );

D = -B/(2*f0*C);
arecon = -1/(1-D);
arecon2 = -(2*f0*C)/(B+2*f0*C);
Dt0recon = ((2*f0*C)/pi) / (2*arecon-arecon*(arecon+1));

G = [ones(Nf,1), f, f.^2 ];
m = (G'*G)\(G'*R);
sigmaR = 0.01;  % prior sqrt variance
covm = (sigmaR^2)*inv(G'*G);
Rest = G*m;
R4max = max( abs(R-Rest)./abs(R) );
```

```
fprintf('Error 4: %f\n', R4max );
plot( f, Rest, 'r-', 'LineWidth', 2 );

Aest = m(1);
Best = m(2);
Cest = m(3);
fprintf('true A %f B %f C %f\n', A, B, C );
fprintf('true A %f B %f C %f\n', A2, B2, C2 );
fprintf('Fit  A %f B %f C %f\n', Aest, Best, Cest );
Dest = -Best/(2*f0*Cest);
aest = 1/(Dest-1);
aest2 = -(2*f0*Cest)/(Best+2*f0*Cest);
Dt0est = ((2*f0*Cest)/pi) / (2*aest-aest*(aest+1));
Dt0est2 = -(Best+2*f0*Cest)/(pi*(1-aest));

% sensitivities, needed for variance calculation
dadA = 0;

dadB = (2*f0*Cest)/((Best+2*f0*Cest)^2);
% dB = 1e-4;
% daB = -(2*f0*Cest)/((Best+dB)+2*f0*Cest) - aest2;
% dadB2 = daB/dB;
% fprintf('dadB %f %f\n', dadB, dadB2 );

dadC = -(2*f0*B)/((Best+2*f0*Cest)^2);
% dC = 1e-4;
% daC = -(2*f0*(Cest+dC))/(Best+2*f0*(Cest+dC)) - aest2;
% dadC2 = daC/dC;
% fprintf('dadC %f %f\n', dadC, dadC2 );

dDt0dA = 0;

dDt0dB = ((-1)/(pi*(1-aest2))) - (Best+2*f0*Cest)*dadB/(pi*((1-aest2)^2));
% apB = -(2*f0*Cest)/((Best+dB)+2*f0*Cest);
% dDt0B = -((Best+dB)+2*f0*Cest)/(pi*(1-apB))-Dt0est2;
% dDt0dB2 = dDt0B/dB;
% fprintf('dDt0dB %f %f\n', dDt0dB, dDt0dB2 );

dDt0dC = ((-2*f0)/(pi*(1-aest2))) - (Best+2*f0*Cest)*dadC/(pi*((1-aest2)^2));
% apC = -(2*f0*(Cest+dC))/(Best+2*f0*(Cest+dC));
% dDt0C = (-(Best+2*f0*(Cest+dC))/(pi*(1-apC)))-Dt0est2;
% dDt0dC2 = dDt0C/dC;
% fprintf('dDt0dC %f %f\n', dDt0dC, dDt0dC2 );

S1 = [ dadA, dadB, dadC; dDt0dA, dDt0dB, dDt0dC ];
covaDt0 = S1*covm*S1';
a951 = 2*sqrt( covaDt0(1,1) );
Dt0951 = 2*sqrt( covaDt0(2,2) );
```

```
% t(f0) and dtdf|f0 of fit
Dt0_ted = -( ((Aest-c)/f0) + Best + Cest*f0 ) / pi;
dDtdf0_ted = ( ((Aest-c)/(f0^2)) - Cest ) / pi;
a_ted = -( f0/Dt0_ted ) * dDtdf0_ted;
a_ted2 = (((Aest-c)/f0)-Cest*f0) / (((Aest-c)/f0)+Best+Cest*f0);

% sensitivities
q = ((Aest-c)/f0)+Best+Cest*f0;
dadAted = (1/(f0*q)) - (((Aest-c)/(f0^2))-Cest)/(q^2);
% dA = 1e-5;
% daA = ((((Aest+dA)-c)/f0)-Cest*f0) / ((((Aest+dA)-c)/f0)+Best+Cest*f0)-
a_ted2;
% dadAted2 = daA/dA;
% fprintf('dadA_ted %f %f\n', dadAted, dadAted2 );

dadBted = -( ((A-c)/f0) - C*f0 ) / (q^2);
% dB = 1e-4;
% daB = (((Aest-c)/f0)-Cest*f0) / (((Aest-c)/f0)+(Best+dB)+Cest*f0)-a_ted2;
% dadBted2 = daB/dB;
% fprintf('dadB_ted %f %f\n', dadBted, dadBted2 );

dadCted = (-f0/q) - ((Aest-c)-Cest*(f0^2))/(q^2);
% dC = 1e-4;
% daC=(((Aest-c)/f0)-(Cest+dC)*f0) / (((Aest-c)/f0)+Best+(Cest+dC)*f0)-
a_ted2;
% dadCted2 = daC/dC;
% fprintf('dadC_ted %f %f\n', dadCted, dadCted2 );

dDt0dAted = -1/(pi*f0);
% dDt0A=-((((Aest+dA)-c)/f0) + Best + Cest*f0 ) / pi  - Dt0_ted;
% dDt0dAted2 = dDt0A/dA;
% fprintf('dDt0dA_ted %f %f\n', dDt0dAted, dDt0dAted2 );

dDt0dBted = -1/pi;
% dDt0B=-(((Aest-c)/f0) + (Best+dB) + Cest*f0 ) / pi  - Dt0_ted;
% dDt0dBted2 = dDt0B/dB;
% fprintf('dDt0dB_ted %f %f\n', dDt0dBted, dDt0dBted2 );

dDt0dCted = -f0/pi;
% dDt0C=-(((Aest-c)/f0) + Best + (Cest+dC)*f0 ) / pi  - Dt0_ted;
% dDt0dCted2 = dDt0C/dC;
% fprintf('dDt0dC_ted %f %f\n', dDt0dCted, dDt0dCted2 );

S2 = [ dadAted, dadBted, dadCted; dDt0dAted, dDt0dBted, dDt0dCted ];
covaDt02 = S2*covm*S2';
a952 = 2*sqrt( covaDt02(1,1) );
Dt0952 = 2*sqrt( covaDt02(2,2) );

A3 = c - 0.5*pi*f0*a_ted*(a_ted+1)*Dt0_ted;
```

```
B3 =  -pi*Dt0_ted*(1+a_ted-a_ted*(a_ted+1));
C3 = (pi*Dt0_ted/(2*f0)) * (2*a_ted - a_ted*(a_ted+1) );
fprintf('Ted  A %f B %f C %f\n', A3, B3, C3 );

fprintf('a   true %f est1 %f +/- %f est2 %f +/- %f\n', a, aest2, a951, a_ted,
a952 );
fprintf('Dt0 true %f est1 %f +/- %f est2 %f +/- %f\n', Dt0, Dt0est2, Dt0951,
Dt0_ted, Dt0952 );

A3 = c - 0.5*pi*f0*a_ted*(a_ted+1)*Dt0_ted;
B3 =  -pi*Dt0_ted*(1+a_ted-a_ted*(a_ted+1));
C3 = (pi*Dt0_ted/(2*f0)) * (2*a_ted - a_ted*(a_ted+1) );

Rted =  A3 + B3*f + C3*(f.^2);
R5max = max( abs(R-Rted)./abs(R) );
fprintf('Error 5: %f\n', R5max );
plot( f, Rted, 'b-', 'LineWidth', 2 );
```