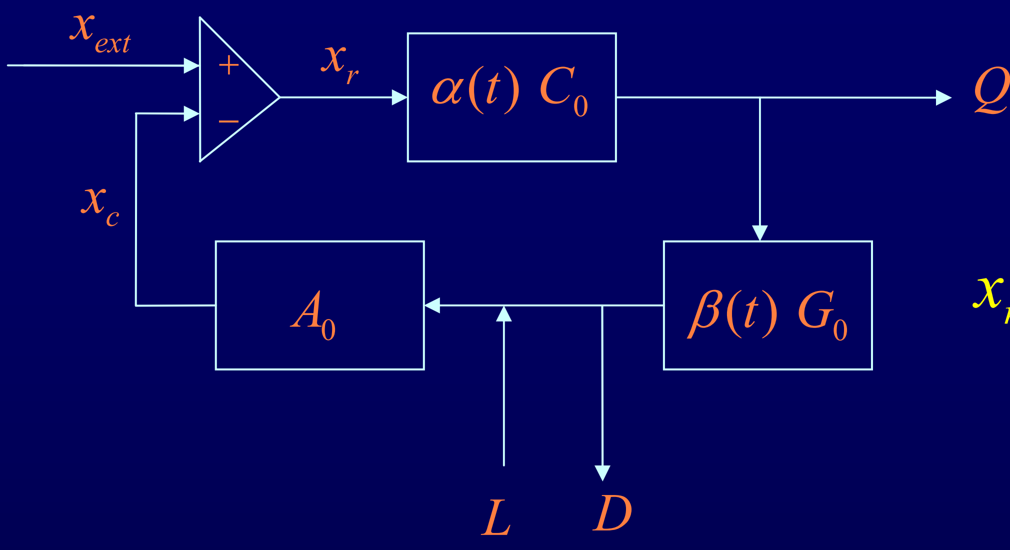
Calculation of $\alpha(t)$ and $\beta(t)$.

XS, Jolien Creighton, Mike Landry

Wanted independent method to calculate calibration factors to use in search codes.

LIGO-G030629-00-Z



At the frequency of the calibration line:

$$x_r(f_c) = x_{ext}(f_c) - x_c(f_c) \approx -x_c(f_c)$$

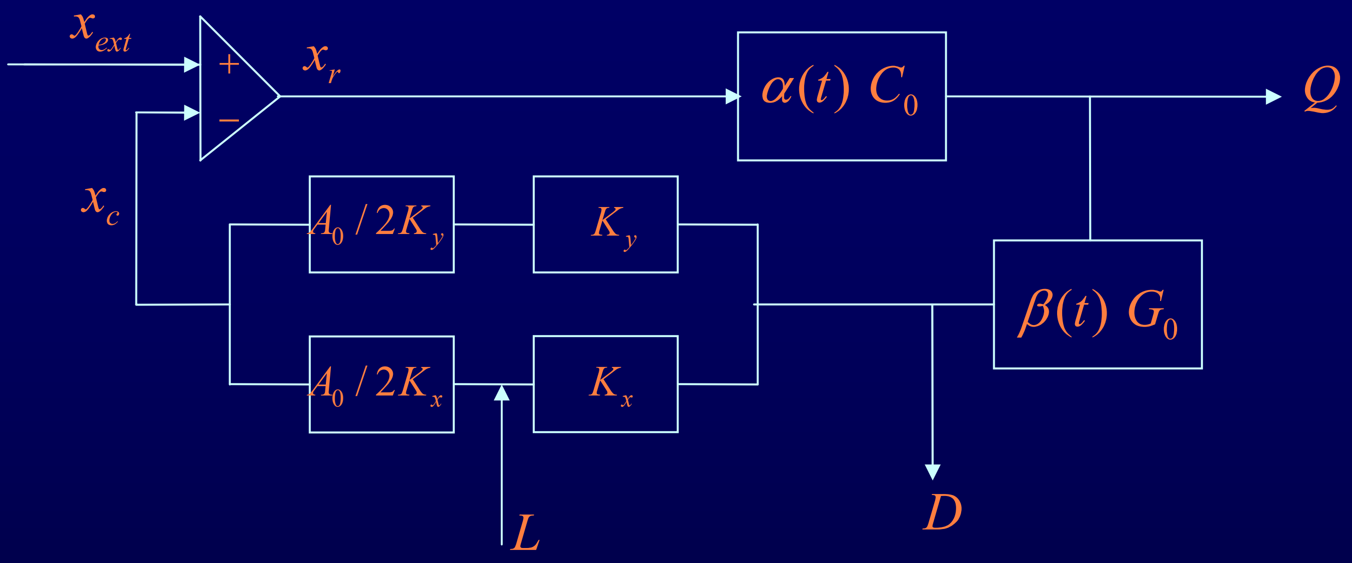
$$D(f_c) = \beta(t)\,G(f_c)\,Q(f_c) = \beta(t)\,G_0(f_c)\alpha(t)\,C_0(f_c)x_r(f_c)$$
$$\approx -\beta(t)\,G_0(f_c)\alpha(t)\,C_0(f_c)x_c(f_c) = -\alpha(t)\beta(t)\ H_0(f_c)\left[D(f_c)+L(f_c)\right]$$

$$\alpha(t)\,\beta(t) \approx -\frac{1}{H_0}\frac{D(f_c)/L(f_c)}{1-D(f_c)/L(f_c)}$$

$$\alpha(t) \approx -\frac{1}{H_0}\frac{Q(f_c)\left(1+\alpha(t)\beta(t)H_0(f_c)\right)}{C_0(f_c)A_0(f_c)L(f_c)}$$

Can independently compute $\alpha(t)$ and $\beta(t)$!

-For S2 this is, in fact, wrong. The cal lines are not injected into DARM



-Things don't change much:

$$L(f_c) \rightarrow L'(f_c) = L(f_c)/2K_x$$

$$\alpha(t)\,\beta(t) \approx -\frac{1}{H_0}\frac{D(f_c)/L'(f_c)}{1-D(f_c)/L'(f_c)}$$

$$\alpha(t) \approx -\frac{1}{H_0}\frac{Q(f_c)\left(1+\alpha(t)\beta(t)H_0(f_c)\right)}{C_0(f_c)A_0(f_c)L'(f_c)}$$

Implemented in LAL in *ComputeCalibrationFactors.c* in packages/tools

$$\alpha(t)\,\beta(t) \approx -\frac{1}{H_0}\frac{D(f_c)/L'(f_c)}{1-D(f_c)/L'(f_c)} \qquad \alpha(t) \approx -\frac{1}{H_0}\frac{Q(f_c)\left(1+\alpha(t)\beta(t)H_0(f_c)\right)}{C_0(f_c)A_0(f_c)L'(f_c)}$$

$D(f_c)$, $L(f_c)$ and $Q(f_c)$ are computed by match filtering with sinusoids

$K_x, C_0(f_c)$, $A_a(f_c)$ and $H_0(f_c)$ are taken from matlab models

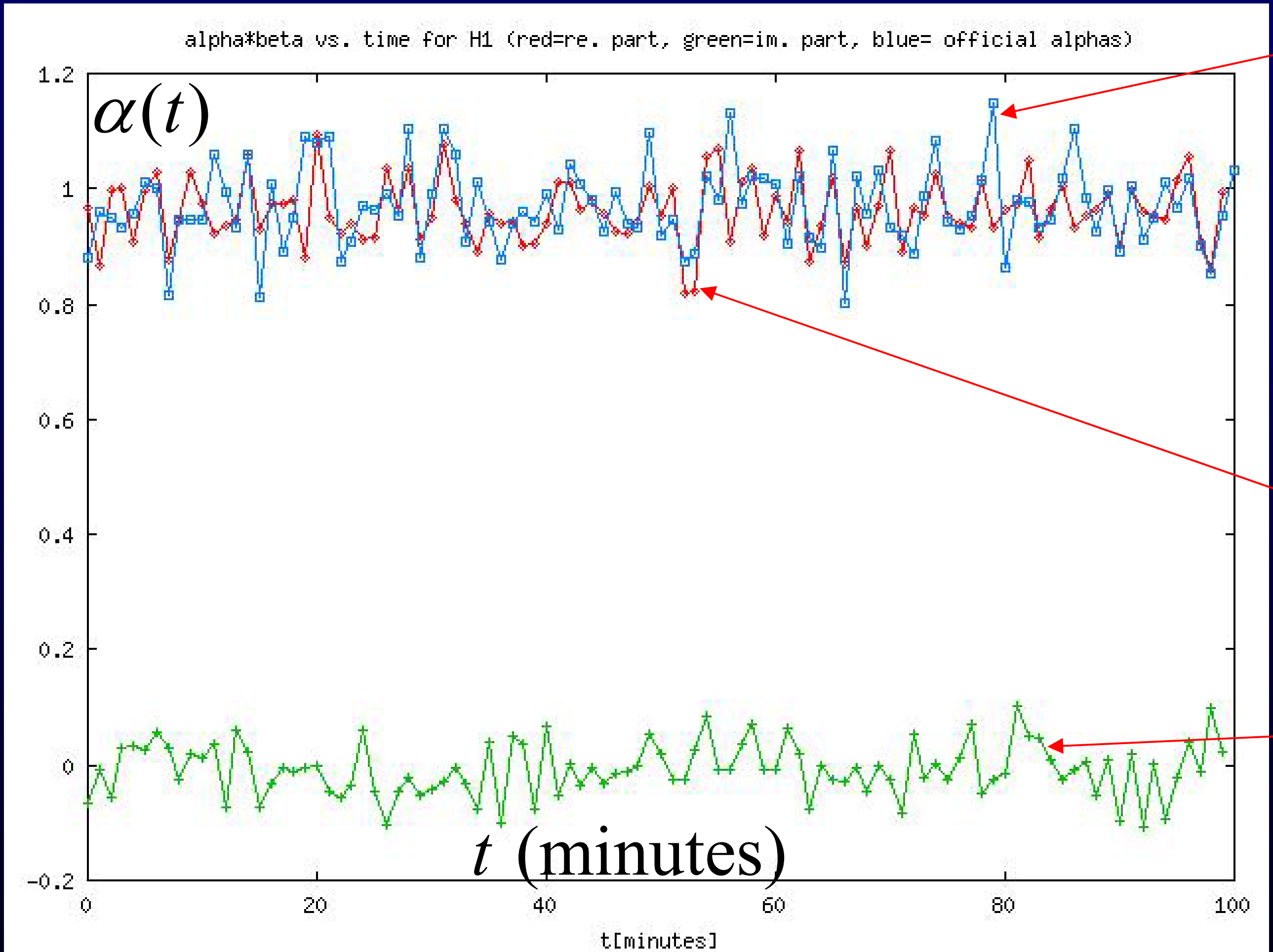
There is a test program, ComputeCalibrationFactorsTest, built with make test that is a driver for this function.

Inputs: cal. line frequency, GPS time info., frame cache file, channel names, reference files for C_0, A_a and H_0, and value of K_x

It outputs:

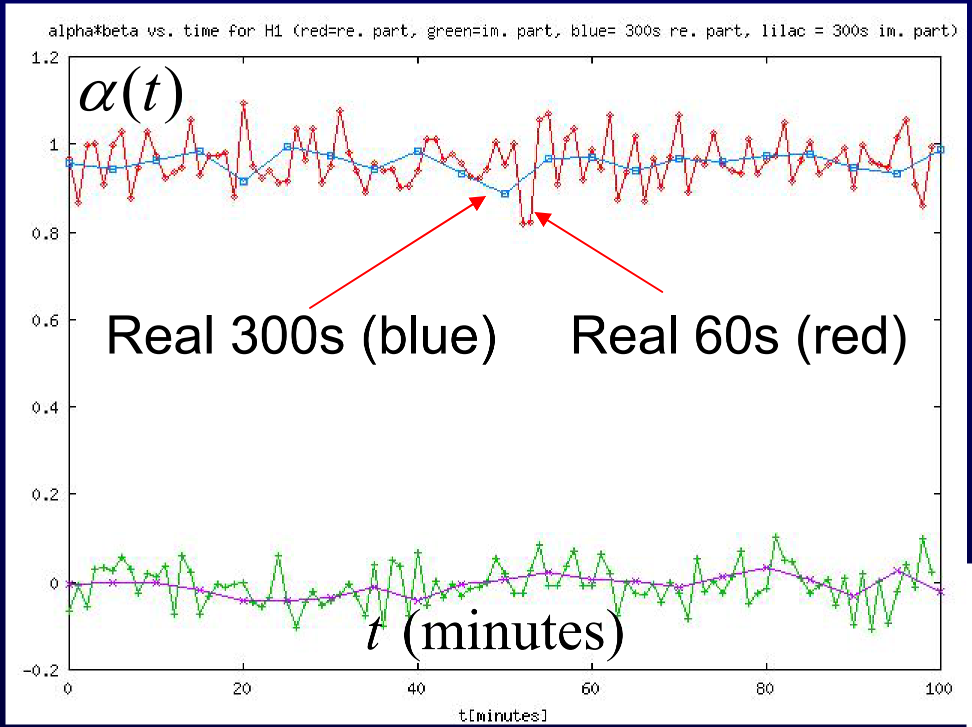
- Real and imaginary parts of $\alpha(t)$ and $\alpha(t)\,\beta(t)$
- Magnitudes of $D(f_c)$, $L(f_c)$ and $Q(f_c)$ from match filters.

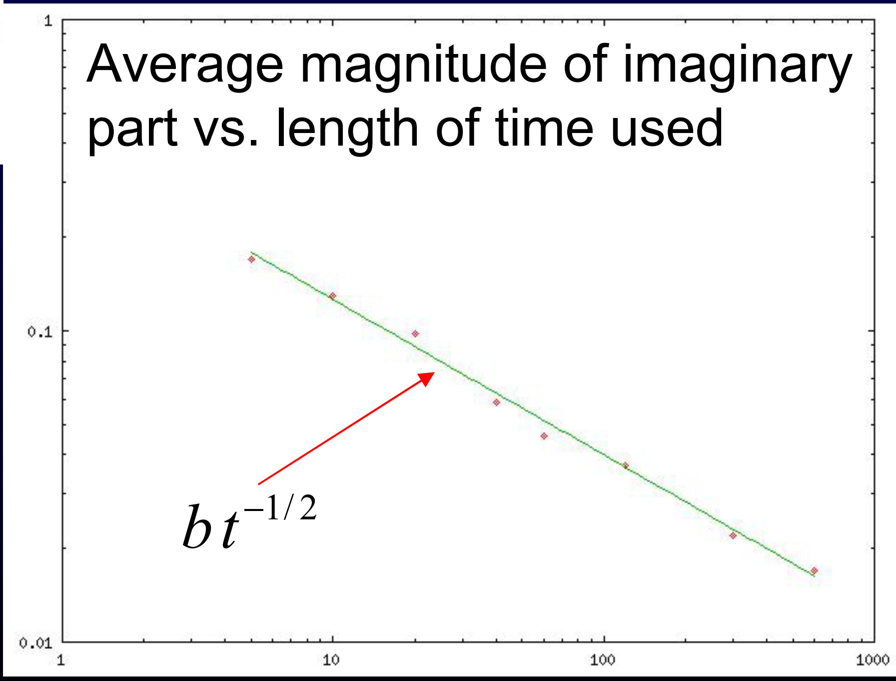
Factors given by *ComputeCalibrationFactorsTest* vs. official factors



$$x_z(f_c) = x_{ext}(f_c) - x_c(f_c) \approx -x_c(f_c)$$

Can compute factors for longer times (by match filtering longer chunks of DARM_CTRL, AS_Q and EXC):





Can find errors in reference calibration functions