LASER INTEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-E1300808-v1	*advanced LIGO*	10/22/2013

TwinCAT Library for Finding resonance in arm cavities

Sheila Dwyer

Distribution of this document:
LIGO Scientific Collaboration

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW22-295
185 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 159
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

http://www.ligo.caltech.edu/

Library	
Title	LockIN
Version	1
TwinCAT version	V2.11.0
Name space	
Author	Sheila Dwyer
Description	Lock in to adjust ALS Comm and DIFF VCO frequency to find IR resonance in arm cavities
Error Code	16#01- green not locked to arm
	16#02-IR trans PD error
	16#04-PLLforVCO error
	16#08-VCO error
	16#20-VCO frequency servo disabled
	16#40-VCO frequency off by more than FindResonance.Controls.VCOErrorMax
	16#80- dither frequency too high
	16#100- dither frequency should be well above VCO ugf (so dither is not suppressed by loop)
	16#200- Low pass frequency needs to be at least factor 2 above ugf of dither servo
	16#400- Trans PD normalization (value at last resonance was more than 1.1, normalization needs to be updated)
	16#800 last res.Trans PD below 0.9 (either normalization needs to be updated or light is too low)
Library Dependencies	Error, SaveRestore, ALSCommunication, LowNoiseVCO,PLLforVCO

State machine:

This library implements a state machine that when requested searches for the IR resonance in the arm cavities using the out of vacuum transmitted IR PD as a sensor and actuating on the LowNoiseVCO frequency. The logic becomes engaged when logic.on is set to true and there are no errors, or logic.force is set to true in which case errors are ignored (for commissioning, not normal automation). If logic on is set to true, and an error condition arises while the script is moving through the state machine, it will go back to disengaged.

Once engaged the state will become init, where the transmitted light is checked first with the current settings, then with the frequency setting found for resonance the last time the resonance was successfully found. If the transmitted light is above a dither.threshold in either case, the state will skip to dither. Otherwise, it will proceed to scan.

In the scan state the library will scan the VCO frequency from the last found resonant frequency +/- widthFirst, at a rate determined by scan.step. This changes the set frequency offset used by the PI controller built into the VCO, which has a UGF of something less than 1 Hz, so the scan rate needs to be slow enough. There is also a second scan, which could be used for a broader scan if the first scan fails. If the transmitted light goes above the dither.threshold the state machine will move on to the ditherServo state. If no transmitted light is found user intervention is needed and the state will go to failed.

In the DitherServo state there are two somewhat generic function blocks used to implement a lock in, described below. If the transmitted power drops below 90% of dither.thresh, the state will transition back to the scan state. If the threshold conditions on both the error signal and the transmitted power are met for the time FindResonance.Delay, the transmitted power and set frequency will be recorded and a the state will transition to done. Transitions out of the done state can be made by a user, or by some other state machine in twinCAT or guardian.
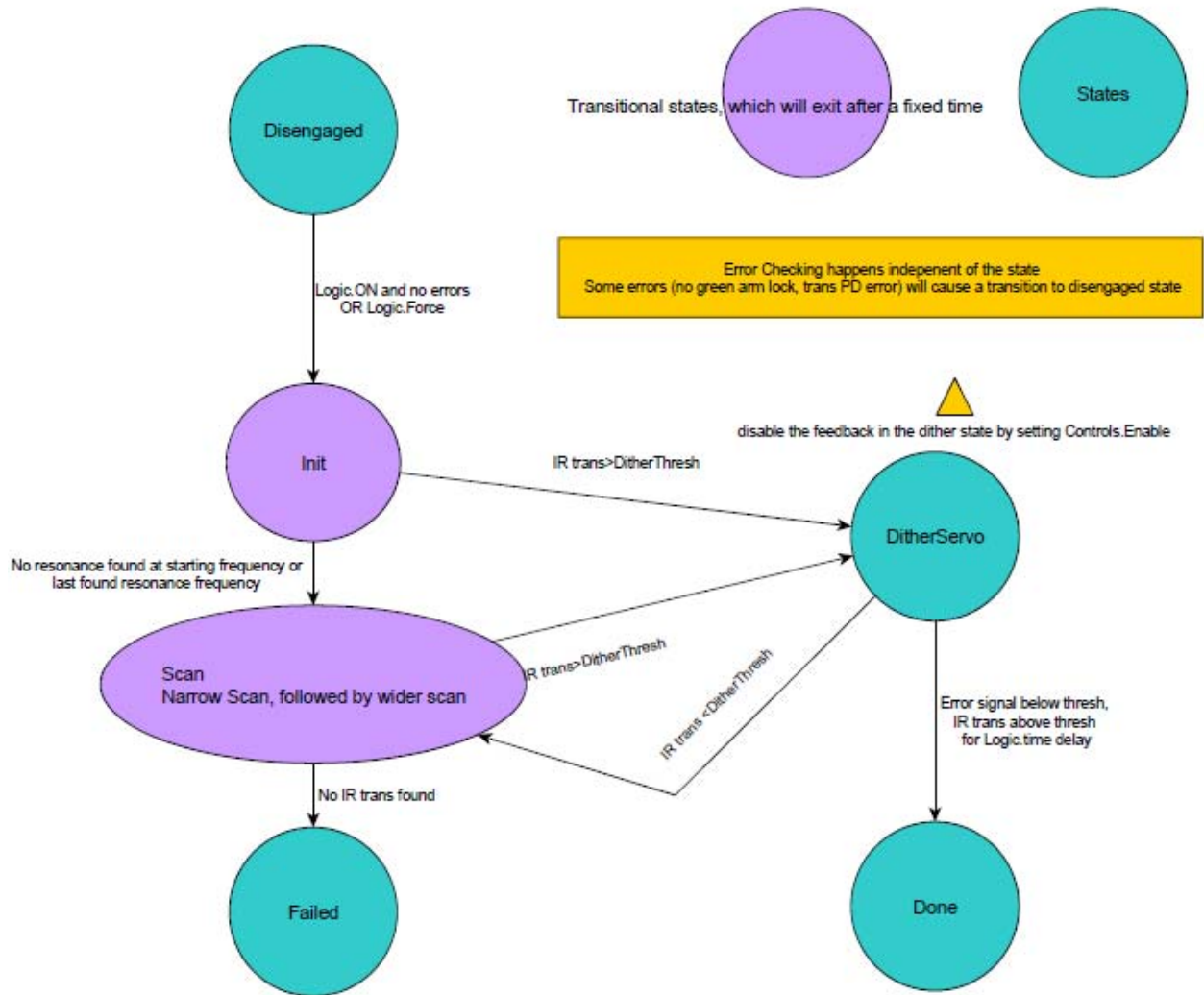
Figure 1: State diagram

DITHERFB:

Contains a generic Dither function block, which applies an excitation of amplitude and frequency specified by the user to a value (excChan), demodulates another value (DemodChan). The bekhoff ADCs have no anitaliasing as far as I know, so the modulation frequency should be less than half the Nyquist frequency to prevent aliasing of the 2f signal. Since we are using the VCO tune voltage to dither, the dither frequency should also be well above the UGF of the VCO frequency servo.

The demodulated signal is low passed with a corner frequency set to be one third of the excitation frequency (this ratio could be converted to a user set value).

$$a = \frac{2\pi ExcFreq}{3}$$

Using Euler's method, the low pass is given by:

$$H(s) = \frac{U(s)}{E(s)} = \frac{a}{s-a}$$

u(k+1)=aTe(k)-(1+aT)u(k)   where T is the sample time, e(k) is the raw demodulated signal from the last cycle and u(k) is the low passed signal from the last cycle. The dither function block produces a dither signal FindResonance.I.

We plan to demodulate the arm cavity IR transmission to adjust the VCO frequency, measuring the derivative of the transmitted intensity with respect to frequency.

$$I(f) = I_{max} \frac{1}{1 + \left(\frac{f-f_o}{\gamma}\right)^2}$$

$$\frac{dI(f)}{df} = \frac{-2I_{max}(f-f_o)}{\gamma(1+\left(\frac{f-f_o}{\gamma}\right)^2)^2} = \frac{-2(f-f_o)}{\gamma} \frac{I(f)^2}{I_{max}}$$

 Approximating the transmitted light as a lorentzian, the demodulated transmitted light only gives an error signal with the correct slope once the frequency is already within half a linewidth (as shown in figure 1). By normalizing the demodulated signal by the intensity squared, we can obtain an error signal with a much larger linear range. By using this signal for feedback instead of the dither signal alone, we can engage a PI controller as soon as we are close enough to resonance to have a good SNR on the transmitted PDs. This sensor has a response (in counts/Hz) of:

$$S(s) = \frac{-2}{\gamma I_{max}} \frac{a}{s-a}$$

The transmitted power from the PD should be normalized to one in the DC power library. The linewidth of the arm cavities should be 83.9Hz, so we can ignore the cavity pole which is above the 50Hz Nyquist frequency of the TwinCAT system.
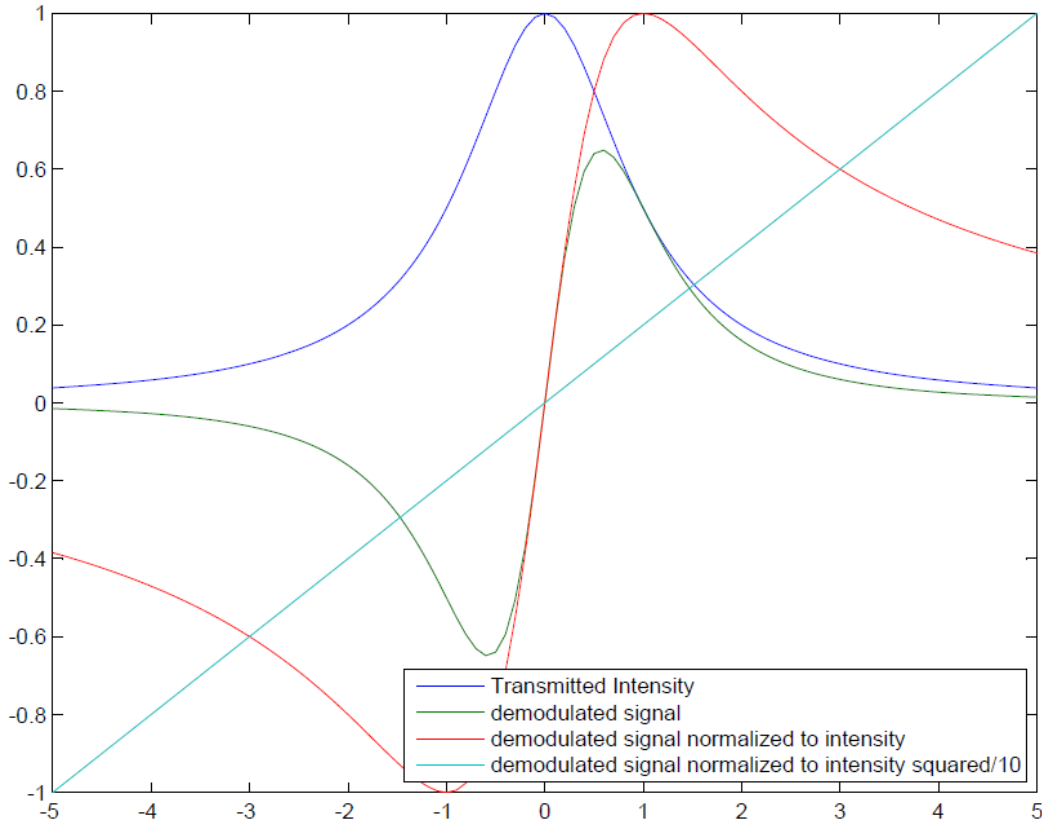
Figure 2: Normalizing the dither signal by the transmitted intensity squared will extend the linear range of the error signal.

PI control

The Low Noise VCO library implements a PI controller that keeps the VCO frequency at the set frequency using the timing system. We will feed back to the error point of this loop, so the VCO response is:

$P(s) = \frac{b}{s-b}$ where b is 2πVCO.Controls.UnityGain.

We use a PI controller:

$$C(s) = K_p \frac{s - 1/T_I}{s}$$

Implemented using the difference equation:

$$u(k) = u(k-1) + K_p \left(1 + \frac{T_I}{T} \right) e(k)$$

This gives an open loop gain of:

$$G(s) = C(s)P(s)S(s) = \frac{-2abK_p}{\gamma I_{max}} \frac{s - 1/T_I}{s(s-a)(s-b)}$$

By setting the PI corner to the VCO frequency servo UGF, we can make the loop stable for any ugf well below the low pass frequency. $(1/T_I = a)$. The ugf (well below b) is then given by: $Dither.ugf = -aK_p/\pi\gamma I_{max}$

The PIControlFB also allows the user to flip the feedback polarity and to reset the integrator.

User Interface Type

TYPE FindResonance :

STRUCT

Error:	ErrorStruct;
Logic:	FindResonanceLogicStruct;
Controls:	FindResonancePIControlsStruct;
Dither:	FindResonanceDitherStruct;
Scan:	FindResonanceScanStruct;
LastResFreq:	LREAL;
LastResTrans:	LREAL;
TransThresh:	LREAL;
ErrorThresh:	LREAL;
Delay:	INT;

END_STRUCT

END_TYPE;

Type Name	PLLforVCOStruct
Description	Structure used in the user interface
Definition	STRUCT
In/Out Tag	Name: Error Type: ErrorStruct Description: Calls error handler

---

Function Block

FUNCTIONBLOCK PLLforVCOFB:

VAR_IN_OUT

Request:	SaveRestoreEnum;
FindResonance:	FindResonanceStruct;
FindResonanceInit:	FindResonanceStruct;
VCO:	LowNoiseVCOStruct;

END_VAR

VAR_INPUT

RecieveFromEnd:	EndtoCornerPLC2Struct;
PLLforVCO:	PLLforVCOStruct;

END_VAR	
VAR	

ErrorHandler: ErrorHandlerFB;
     ErrorChecking: ErrorCheckingFB;
     DitherFB: DitherFB;
     PICntrlFB: PICntrlFB;
     Counter: INT;
     OldState: FindResonanceStateEnum;	

END_VAR	
Type Name	ALSFindResonanceFB
Description	See above
Definition	Function Block