Matlab上で動くOpticklelによる
干渉計シミュレーション

October 2, 2009 @ 重力波研究交流会

東大宇宙線研　宮川　治

この話をする目的

LCGTの干渉計、制御
の計算をできる人を育てたい

History of IFO calculation tools

時間領域と周波数領域の干渉計シミュレーションツールがある。
前者の代表はe2e。今日は後者のお話。

Twiddle(1998~) by Hiro Yamamoto, Jim Mason
- 初めてのRFも含めた実用的な干渉計周波数応答シミュレータ。Mathematica上で動く。
- 単位の定義がややこしいのと遅いのでFINESSEに取って代わられた。

FINESSE(2000~) by Andreas Freise
- 世に出た2つ目の実用的な干渉計周波数応答シミュレータ。
- Cで書かれ、Twiddleよりもかなり速い。
- Angleも計算できる

Thomasツール(2003~) by Thomas Corbitt
- 輻射圧も含めて計算できる最初のツール。ただしキャリアのみでRFはできない。
- C++で書かれていて、できることはかなり限られている。

Optickle (~2005) by Matt Evans

OpticとtickleのTITLE造語

何ができるか?

- 干渉計の周波数応答を見ることができる
- RFもふくめた輻射圧の計算
- 輻射圧ノイズを含めたquantum noise
- Angleの計算
- 複数のレーザー光源
- 制御系のサポート
- Matlab上で動く→ソースを見ることも、改変することもできる

Optickle installation

Download

http://ilog.ligo-wa.caltech.edu:7285/advligo/ISC_Modeling_Software

ただし、LVCのパスワードが必要

Recent version:

Optickle_080626.zip

もしくは坪野研のSVN

https://granite.phys.s.u-tokyo.ac.jp/svn/LCGT/trunk/isc/Optickle/

IDとパスワードは当日言います

- その他にも loopnoise (宮川), looptickle (Stefan), pickle(Lisa)などOptickleエンジンを使ったコード集がある

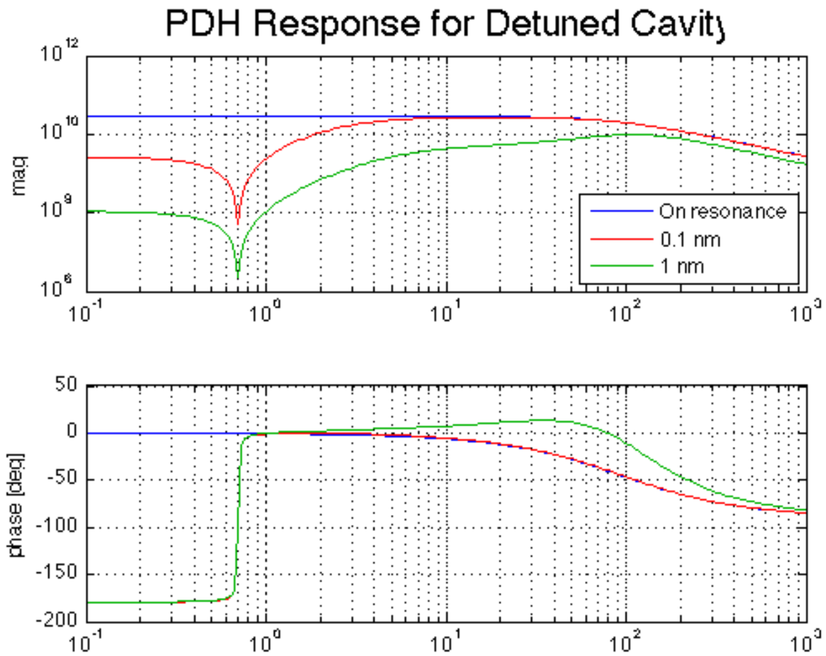
Demonstration

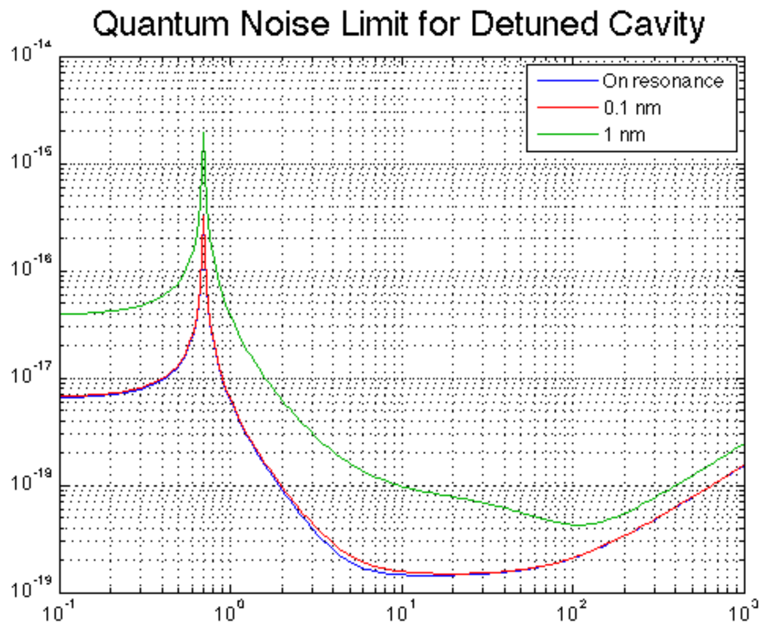
解凍した後、Matlabを起動し、@Optickleを含むディレクトリで

```
>> path(pathdef)
>> addpath(genpath(pwd))
>> cd lib
>> demoDetuneFP
```

(付属のデモプログラム、detuned FP cavityのオプティカルゲインと感度)

これらのことはreadme.txtに書かれています





demoDetuneFP.m

```
% demoDetuneFP
%   this function demonstrates the use of tickle with optFP
%

function demoDetuneFP

  % create the model
  opt = optFP;

  % get some drive indexes
  nEX = getDriveIndex(opt, 'EX');
  nIX = getDriveIndex(opt, 'IX');

  % get some probe indexes
  nREFL_DC = getProbeNum(opt, 'REFL_DC');
  nREFL_I = getProbeNum(opt, 'REFL_I');
  nREFL_Q = getProbeNum(opt, 'REFL_Q');

  nTRANSa_DC = getProbeNum(opt, 'TRANSa_DC');
  nTRANSb_DC = getProbeNum(opt, 'TRANSb_DC');

  % compute the DC signals and TFs on resonances
  f = logspace(-1, 3, 200)';
  [fDC, sigDC0, sigAC0, mMech0, noiseAC0] = tickle(opt, [], f);

  % compute the same a little off resonance
  pos = zeros(opt.Ndrive, 1);
  pos(nEX) = 0.1e-9;
  [fDC, sigDC1, sigAC1, mMech1, noiseAC1] = tickle(opt, pos, f);

  % and a lot off resonance
  pos(nEX) = 1e-9;
  [fDC, sigDC2, sigAC2, mMech2, noiseAC2] = tickle(opt, pos, f);

  % make a response plot
  h0 = getTF(sigAC0, nREFL_I, nEX);
  h1 = getTF(sigAC1, nREFL_I, nEX);
  h2 = getTF(sigAC2, nREFL_I, nEX);
```

干渉計パラメータファイル”optFP.m”の呼び出し

光学素子、ディテクタ-にアクセスするための番号の定義

具体的な干渉計応答を得る一番重要なコマンド

optFP.m : 干渉計パラメータ設定ファイル

```
% Create an Optickle Fabry-Perot

function opt = optFP

  % RF component vector
  Pin = 100;
  vMod = (-1:1)';
  fMod = 20e6;
  vFrf = fMod * vMod;

  % create model
  opt = Optickle(vFrf);

  % add a source
  opt = addSource(opt, 'Laser', sqrt(Pin) * (vMod == 0));

  % add an AM modulator (for intensity control, and intensity noise)
  %   opt = addModulator(opt, name, cMod)
  opt = addModulator(opt, 'AM', 1);
  opt = addLink(opt, 'Laser', 'out', 'AM', 'in', 0);

  % add an PM modulator (for frequency control and noise)
  opt = addModulator(opt, 'PM', i);
  opt = addLink(opt, 'AM', 'out', 'PM', 'in', 0);

  % add an RF modulator
  %   opt = addRFmodulator(opt, name, fMod, aMod)
  gamma = 0.2;
  opt = addRFmodulator(opt, 'Mod1', fMod, i * gamma);
  opt = addLink(opt, 'PM', 'out', 'Mod1', 'in', 1);

  % add mirrors
  %   opt = addMirror(opt, name, aio, Chr, Thr, Lhr, Rar, Lmd, Nmd)
  lCav = 4000;
  opt = addMirror(opt, 'IX', 0, 0, 0.03);
  opt = addMirror(opt, 'EX', 0, 0.7 / lCav, 0.001);

  opt = addLink(opt, 'Mod1', 'out', 'IX', 'bk', 2);
  opt = addLink(opt, 'IX', 'fr', 'EX', 'fr', lCav);
  opt = addLink(opt, 'EX', 'fr', 'IX', 'fr', lCav);

  % set some mechanical transfer functions
  w = 2 * pi * 0.7;      % pendulum resonance frequency
  mI = 40;              % mass of input mirror
  mE = 40;              % mass of end mirror
```

RFの定義

光学素子の定義

How to use help

例えばaddMirrorというコマンドがわからなかったら

```
>> help addMirror
 --- help for Optickle/addMirror ---

  [opt, sn] = addMirror(opt, name, aio, Chr, Thr, Lhr, Rar, Lmd, Nmd)
    Add a mirror to the model.

  aio - angle of incidence (in degrees)
  Chr - curvature of HR surface (Chr = 1 / radius of curvature)
  Thr - power transmission of HR suface
  Lhr - power loss on reflection from HR surface
  Rar - power reflection of AR surface
  Nmd - refractive index of medium (1.45 for fused silica, SiO2)
  Lmd - power loss in medium (one pass)

  see Mirror for more information
```

optFP.m

```
% add mirrors
%   opt = addMirror(opt, name, aio, Chr, Thr, Lhr, Rar, Lmd, Nmd)
lCav = 4000;
opt = addMirror(opt, 'IX', 0, 0, 0.03);
opt = addMirror(opt, 'EX', 0, 0.7 / lCav, 0.001);

opt = addLink(opt, 'Mod1', 'out', 'IX', 'bk', 2);
opt = addLink(opt, 'IX', 'fr', 'EX', 'fr', lCav);
opt = addLink(opt, 'EX', 'fr', 'IX', 'fr', lCav);

% set some mechanical transfer functions
w = 2 * pi * 0.7;      % pendulum resonance frequency
mI = 40;               % mass of input mirror
mE = 40;               % mass of end mirror

w_pit = 2 * pi * 0.5;   % pitch mode resonance frequency

rTM = 0.17;            % test-mass radius
tTM = 0.2;             % test-mass thickness
iTM = (3 * rTM^2 + tTM^2) / 12;  % TM moment / mass

iI = mE * iTM;         % moment of input mirror
iE = mE * iTM;         % moment of end mirror

dampRes = [0.01 + 1i, 0.01 - 1i];

opt = setMechTF(opt, 'IX', zpk([], -w * dampRes, 1 / mI));
opt = setMechTF(opt, 'EX', zpk([], -w * dampRes, 1 / mE));

opt = setMechTF(opt, 'IX', zpk([], -w_pit * dampRes, 1 / iI), 2);
opt = setMechTF(opt, 'EX', zpk([], -w_pit * dampRes, 1 / iE), 2);

% tell Optickle to use this cavity basis
opt = setCavityBasis(opt, 'IX', 'EX');

% add REFL optics
opt = addSink(opt, 'REFL');
opt = addLink(opt, 'IX', 'bk', 'REFL', 'in', 2);

% add REFL probes (this call adds probes REFL_DC, I and Q)
phi = 180 - 83.721;
opt = addReadout(opt, 'REFL', [fMod, phi]);

% add TRANS optics (adds telescope, splitter and sinks)
% opt = addReadoutTelescope(opt, name, f, df, ts, ds, da, db)
```

光学素子の定義

機械系の定義、輻射圧を計算する際使われる

optFP.m

```
opt = setMechTF(opt, 'IX', zpk([], -w_pit * dampRes, 1 / iI), 2);
opt = setMechTF(opt, 'EX', zpk([], -w_pit * dampRes, 1 / iE), 2);

% tell Optickle to use this cavity basis
opt = setCavityBasis(opt, 'IX', 'EX');

% add REFL optics
opt = addSink(opt, 'REFL');
opt = addLink(opt, 'IX', 'bk', 'REFL', 'in', 2);

% add REFL probes (this call adds probes REFL_DC, I and Q)
phi = 180 - 83.721;
opt = addReadout(opt, 'REFL', [fMod, phi]);

% add TRANS optics (adds telescope, splitter and sinks)
% opt = addReadoutTelescope(opt, name, f, df, ts, ds, da, db)
opt = addReadoutTelescope(opt, 'TRANS', 2, [2.2 0.19], ...
  0.5, 0.1, 0.1, 4.1);
opt = addLink(opt, 'EX', 'bk', 'TRANS_TELE', 'in', 0.3);

% add TRANS probes
opt = addProbeIn(opt, 'TRANSa_DC', 'TRANSa', 'in', 0, 0); % DC
opt = addProbeIn(opt, 'TRANSb_DC', 'TRANSb', 'in', 0, 0); % DC

% add a source at the end, just for fun
opt = addSource(opt, 'FlashLight', (1e-3)^2 * (vMod == 1));
opt = addGouyPhase(opt, 'FrenchGuy', pi / 4);
opt = addLink(opt, 'FlashLight', 'out', 'FrenchGuy', 'in', 0.1);
opt = addLink(opt, 'FrenchGuy', 'out', 'EX', 'bk', 0.1);
opt = setGouyPhase(opt, 'FrenchGuy', pi / 8);

% add unphysical intra-cavity probes
opt = addProbeIn(opt, 'IX_DC', 'IX', 'fr', 0, 0);
opt = addProbeIn(opt, 'EX_DC', 'EX', 'fr', 0, 0);
```

Photo detectorの定義

'opt' structure

optFPが何か分からないとき、マニュアルで
>> opt=optFP　と入力、
==== 3 RF frequencies
1) -20 MHz with amplitude 0
2) DC with amplitude 10
3) 20 MHz with amplitude 0
==== 13 optics

13個の光学素子

1) Laser is a Source (in: none, out: out=1)
2) AM is a Modulator (in: in=1, out: out=2)
3) PM is a Modulator (in: in=2, out: out=3)
4) Mod1 is a RFmodulator (in: in=3, out: out=4)
5) IX is a Mirror (in: fr=6 bk=4, out: fr=5 bk=7)
6) EX is a Mirror (in: fr=5 bk=13, out: fr=6 bk=11)
7) REFL is a Sink (in: in=7, out: none)
8) TRANS_TELE is a Telescope (in: in=11, out: out=8)
9) TRANS_SMIR is a Mirror (in: fr=8, out: fr=9 bk=10)
10) TRANSa is a Sink (in: in=9, out: none)
11) TRANSb is a Sink (in: in=10, out: none)
12) FlashLight is a Source (in: none, out: out=12)
13) FrenchGuy is a GouyPhase (in: in=12, out: out=13)
==== 7 drive points

7個の揺らすポイント

1) AM.drive drives AM (optic 2, drive index 1)
2) PM.drive drives PM (optic 3, drive index 1)
3) Mod1.amp drives Mod1 (optic 4, drive index 1)
4) Mod1.phase drives Mod1 (optic 4, drive index 2)
5) IX.pos drives IX (optic 5, drive index 1)
6) EX.pos drives EX (optic 6, drive index 1)
7) TRANS_SMIR.pos drives TRANS_SMIR (optic 9, drive index 1)
==== 13 links

13個のリンク（光学素子間のこと）

1) 0 meters from Laser->out to AM<-in
2) 0 meters from AM->out to PM<-in
3) 1 meters from PM->out to Mod1<-in

'opt' structure

その後できたoptをWorkspaceで見てみると
1x1のOpticckle構造体だと言うことが分かる、そこで

```
>> help optickle
Contents of Optickle:

BeamSplitter        - is a type of Optic used in Optickle
GouyPhase           - is a type of Optic used in Optickle
Mirror              - is a type of Optic used in Optickle
Modulator           - is a type of Optic used in Optickle
OpHG                - Hermite-Guass Opterator
Optic               - Optickle              - Model
RFmodulator         - is a type of Optic used in Optickle
Sink                - is a type of Optic used in Optickle
Source              - is a type of Optic used in Optickle
Telescope           - is a type of Optic used in Optickle


Optickle is both a directory and a function.

  Optickle Model

  opt = Optickle(vFrf, lambda)
  vFrf - RF frequency components
  lambda - carrier wave length (default 1064 nm)

  Class fields are:

  optic - a cell array of optics
  Noptic - number of optics
  Ndrive - number of drives (inputs to optics)
  link - an array of links
  Nlink - number of links
  probe - an array of probes
  Nprobe - number of probes
  lambda - carrier wave length
  vFrf - RF frequency components
```

‘opt’ structure

更に、

```
>> opt.lambda
ans =
  1.0640e-006

>> opt.link
ans =
    13
```

13個のリンク（光学素子間のつなぎ）があるということ

```
>> opt.probe
ans =
7x1 struct array with fields:
    sn
    name
    nField
    freq
    phase
```

7個の光検出器

```
>> opt.probe(1).name
ans =
REFL_DC
```

1個目の光検出器の名前

等でヒントが得られる

demoDetuneFP.m

```
% demoDetuneFP
%   this function demonstrates the use of tickle with optFP
%

function demoDetuneFP

  % create the model
  opt = optFP;

  % get some drive indexes
  nEX = getDriveIndex(opt, 'EX');
  nIX = getDriveIndex(opt, 'IX');

  % get some probe indexes
  nREFL_DC = getProbeNum(opt, 'REFL_DC');
  nREFL_I = getProbeNum(opt, 'REFL_I');
  nREFL_Q = getProbeNum(opt, 'REFL_Q');

  nTRANSa_DC = getProbeNum(opt, 'TRANSa_DC');
  nTRANSb_DC = getProbeNum(opt, 'TRANSb_DC');

  % compute the DC signals and TFs on resonances
  f = logspace(-1, 3, 200)';
  [fDC, sigDC0, sigAC0, mMech0, noiseAC0] = tickle(opt, [], f);

  % compute the same a little off resonance
  pos = zeros(opt.Ndrive, 1);
  pos(nEX) = 0.1e-9;
  [fDC, sigDC1, sigAC1, mMech1, noiseAC1] = tickle(opt, pos, f);

  % and a lot off resonance
  pos(nEX) = 1e-9;
  [fDC, sigDC2, sigAC2, mMech2, noiseAC2] = tickle(opt, pos, f);

  % make a response plot
  h0 = getTF(sigAC0, nREFL_I, nEX);
  h1 = getTF(sigAC1, nREFL_I, nEX);
  h2 = getTF(sigAC2, nREFL_I, nEX);
```

干渉計パラメータファイル”optFP.m”の呼び出し

光学素子、ディテクタにアクセスするための番号の定義

具体的な干渉計応答を得る一番重要なコマンド

'tickle'

```
>> help tickle
 --- help for Optickle/tickle ---

  Compute DC fields, and DC signals, and AC transfer functions

  [fDC, sigDC, sigAC, mMech, noiseAC, noiseMech] = tickle(opt, pos, f)
  opt - Optickle model
  pos - optic positions (Ndrive x 1, or empty)
  f - audio frequency vector (Naf x 1)

  fDC - DC fields at this position (Nlink x Nrf)
    where Nlink is the number of links, and Nrf
    is the number of RF frequency components.
  sigDC - DC signals for each probe (Nprobe x 1)
   where Nprobe is the number of probes.
  sigAC - transfer matrix (Nprobe x Ndrive x Naf),
    where Ndrive is the total number of optic drive
    inputs (e.g., 1 for a mirror, 2 for a RFmodulator).
    Thus, sigAC is arranged such that sigAC(n, m, :)
    is the TF from the drive m to probe n.
  mMech - modified drive transfer functions (Ndrv x Ndrv x Naf)
  noiseAC - quantum noise at each probe (Nprb x Naf)
  noiseMech - quantum noise at each drive (Ndrv x Naf)

  Example:
  f = logspace(0, 3, 300);
  opt = optFP;
  [fDC, sigDC, sigAC, mMech] = tickle(opt, [], f);
```

'fDC'

```
>> f = logspace(-1, 3, 200)';
[fDC, sigDC0, sigAC0, mMech0, noiseAC0] = tickle(opt, [], f);
```

とうって具体的に各変数を見てみよう

fDC - DC fields at this position (Nlink x Nrf) where Nlink is the number of links, and Nrf is the number of RF frequency components.

Array Editor - fDC

Nrf →

Nlink ↓

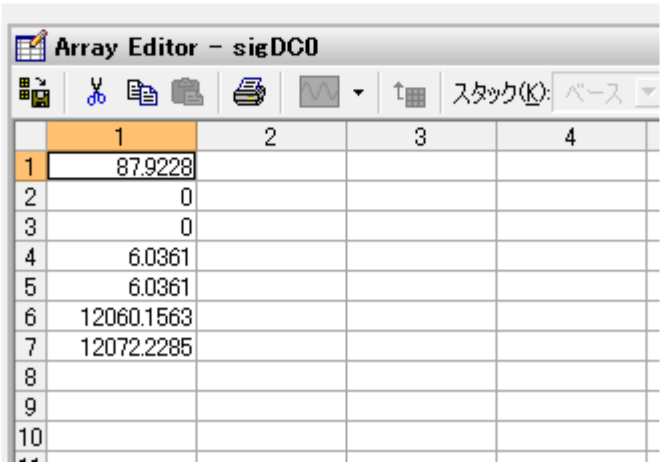
	1	2	3	4
1	0 + 0i	10 + 0i	0 + 0i	
2	1.0668e-016 - 3.4561e-017i	10 + 0i	-1.0668e-016 - 3.4561e-017i	
3	8.338e-017 - 7.4989e-017i	10 + 0i	-8.338e-017 - 7.4989e-017i	
4	0.7398 + 0.6654i	9.9002 + 1.3388e-017i	-0.7398 + 0.6654i	
5	-0.0717 + 0.0807i	109.8736 + 1.4858e-016i	0.0717 + 0.0807i	
6	0.1075 + 0.0097i	-109.8186 - 1.485e-016i	-0.1075 + 0.0097i	
7	0.9882 - 0.1163i	-9.2705 - 1.2536e-017i	-0.9882 - 0.1163i	
8	0.0019 - 0.0028i	3.4745 + 4.6984e-018i	-0.0019 - 0.0028i	
9	-0.0012 + 0.0021i	-2.4568 - 3.3223e-018i	0.0012 + 0.0021i	
10	-0.0022 - 0.001i	2.4568 + 3.3223e-018i	0.0022 - 0.001i	
11	-0.0019 + 0.0028i	3.4745 + 4.6984e-018i	0.0019 + 0.0028i	
12	0 + 0i	0 + 0i	0 + 0i	
13	0 + 0i	0 + 0i	0 + 0i	
14				
15				
16				

各項目は各リンクにおけるレーザーの振幅(パワーではない)

'sigDC'

sigDC - DC signals for each probe (Nprobe x 1) where Nprobe is the number of probes.

Nprobe



Array Editor – sigDC0

	1	2	3	4
1	87.9228			
2	0			
3	0			
4	6.0361			
5	6.0361			
6	12060.1563			
7	12072.2285			
8				
9				
10				

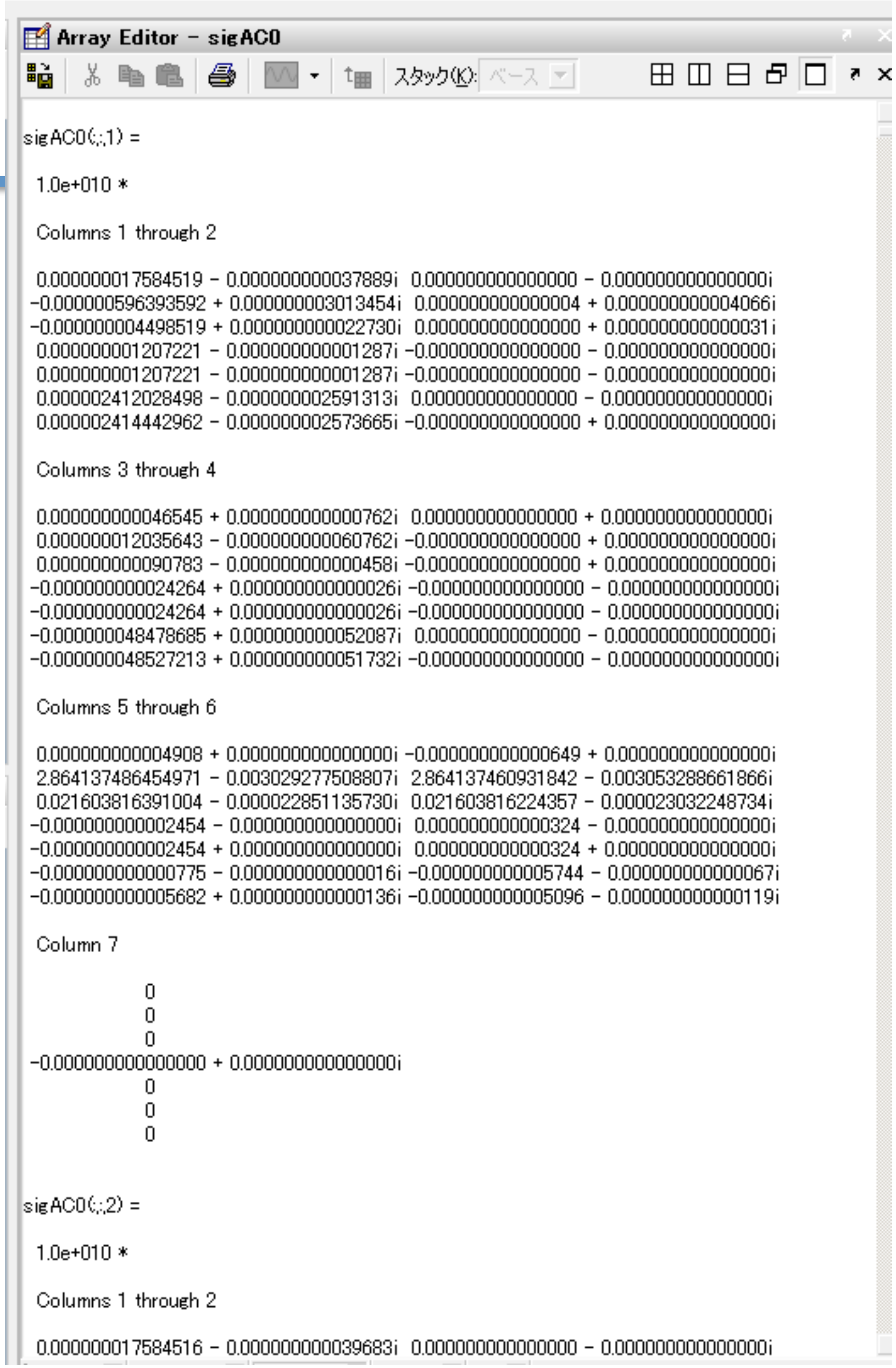
```
>> opt
:
:
==== 7 probes
1) REFL_DC probes field 7 at DC
2) REFL_I probes field 7 at 20 MHz, 96.279 degrees
3) REFL_Q probes field 7 at 20 MHz, 186.279 degrees
4) TRANSa_DC probes field 9 at DC
5) TRANSb_DC probes field 10 at DC
6) IX_DC probes field 6 at DC
7) EX_DC probes field 5 at DC
```

各項目は各PDに入射するレーザーのパワー

‘sigAC’

- sigAC - transfer matrix (Nprobe x Ndrive x Naf), where Ndrive is the total number of optic drive inputs (e.g., 1 for a mirror, 2 for a RFmodulator). Thus, sigAC is arranged such that sigAC(n, m, :) is the TF from the drive m to probe n.

- オプティカルゲインのこと。具体的にはここでは7(検出器の数)x7(振るポイントの数) x200(周波数)のマトリックス。例は最初の周波数(ここでは0.1Hz)。全部の光学素子をふってやって全部のポートに出てくる伝達関数をすべてsigACというひとつの変数に記憶させておく。



```
Array Editor - sigAC0

sigAC0(:,:,1) =

  1.0e+010 *

  Columns 1 through 2

  0.000000017584519 - 0.000000000037889i  0.000000000000000 - 0.000000000000000i
 -0.000000596393592 + 0.000000003013454i  0.000000000000004 + 0.000000000004066i
 -0.000000004498519 + 0.000000000022730i  0.000000000000000 + 0.000000000000031i
  0.000000001207221 - 0.000000000001287i -0.000000000000000 - 0.000000000000000i
  0.000000001207221 - 0.000000000001287i -0.000000000000000 - 0.000000000000000i
  0.000002412028498 - 0.000000002591313i  0.000000000000000 - 0.000000000000000i
  0.000002414442962 - 0.000000002573665i -0.000000000000000 + 0.000000000000000i

  Columns 3 through 4

  0.000000000046545 + 0.000000000000762i  0.000000000000000 + 0.000000000000000i
  0.000000012035643 - 0.000000000060762i -0.000000000000000 + 0.000000000000000i
  0.000000000090783 - 0.000000000000458i -0.000000000000000 + 0.000000000000000i
 -0.000000000024264 + 0.000000000000026i -0.000000000000000 - 0.000000000000000i
 -0.000000000024264 + 0.000000000000026i -0.000000000000000 - 0.000000000000000i
 -0.000000048478685 + 0.000000000052087i  0.000000000000000 - 0.000000000000000i
 -0.000000048527213 + 0.000000000051732i -0.000000000000000 - 0.000000000000000i

  Columns 5 through 6

  0.000000000004908 + 0.000000000000000i -0.000000000000649 + 0.000000000000000i
  2.864137486454971 - 0.003029277508807i  2.864137460931842 - 0.003053288661866i
  0.021603816391004 - 0.000022851135730i  0.021603816224357 - 0.000023032248734i
 -0.000000000002454 - 0.000000000000000i  0.000000000000324 - 0.000000000000000i
 -0.000000000002454 + 0.000000000000000i  0.000000000000324 + 0.000000000000000i
 -0.000000000000775 - 0.000000000000016i -0.000000000005744 - 0.000000000000067i
 -0.000000000005682 + 0.000000000000136i -0.000000000005096 - 0.000000000000119i

  Column 7

                  0
                  0
                  0
 -0.000000000000000 + 0.000000000000000i
                  0
                  0
                  0

sigAC0(:,:,2) =

  1.0e+010 *

  Columns 1 through 2

  0.000000017584516 - 0.000000000039683i  0.000000000000000 - 0.000000000000000i
```

各項目はDC probeの時はエラーシグナルの形、RF probeの時はエラーシグナルの傾きに相当(単位はW/m)

‘noiseAC’

noiseAC - quantum noise at each probe (Nprb x Naf)

•各プローブ(PD)で検出したQuantum ノイズスペクトラム

•ここでは具体的には7(プローブ数)x200(周波数)のマトリックス

```
>> noiseAC0

noiseAC0 =

 1.0e-005 *

 Columns 1 through 5

  0.000572951608894  0.000572951608881  0.000572951608867  0.000572951608851  0.000572951608834
  0.018712974487034  0.018750829515327  0.018792532589804  0.018838494268189  0.018889172997189
  0.000424989677426  0.000425084596132  0.000425189360981  0.000425305063740  0.000425432932093
  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077
  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077
  0.075666735429682  0.075666731260772  0.075666726687537  0.075666721670764  0.075666716167437
  0.075704595373536  0.075704591202540  0.075704586627016  0.075704581607733  0.075704576101653

 Columns 6 through 10

  0.000572951608815  0.000572951608795  0.000572951608772  0.000572951608748  0.000572951608720
  0.018945081558490  0.019006794561412  0.019074957188975  0.019150295452744  0.019233628273673
  0.000425574349691  0.000425730879791  0.000425904293283  0.000426096602082  0.000426310099108
  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077
  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077  0.000150127610077
  0.075666710130366  0.075666703507788  0.075666696242914  0.075666688273454  0.075666679531072
  0.075704570061561  0.075704563435670  0.075704556167161  0.075704548193713  0.075704539446957

 Columns 11 through 15
```

•各項目はノイズスペクトル (単位はW/rHz)

•このノイズ(noizeAC)を信号(sigAC)で割ったものが感度(signal to noise ratio、単位はm/rHz)となる

demoDetuneFP.m

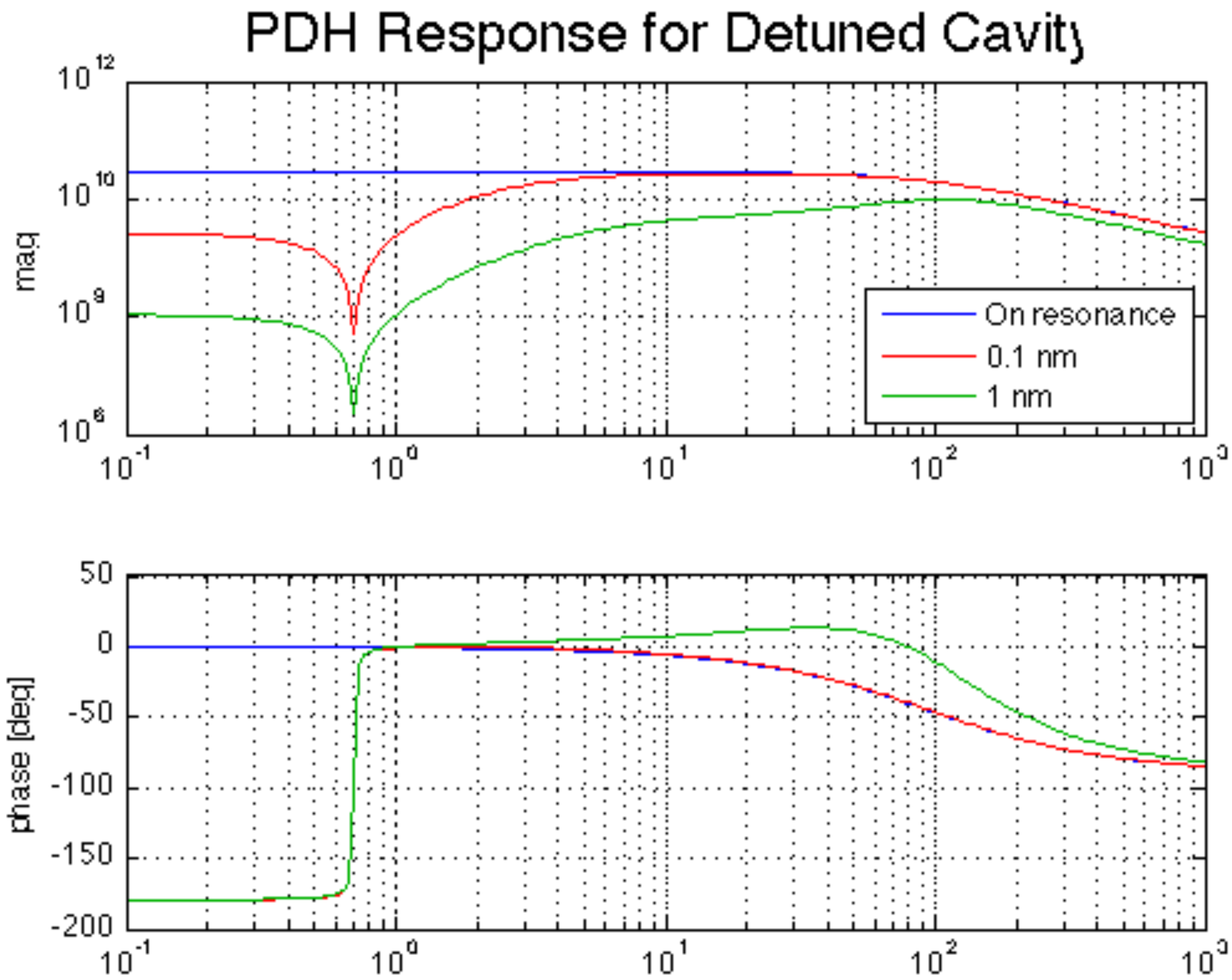
```
[fDC, sigDC1, sigAC1, mMech1, noiseAC1] = tickle(opt, pos, f);

% and a lot off resonance
pos(nEX) = 1e-9;
[fDC, sigDC2, sigAC2, mMech2, noiseAC2] = tickle(opt, pos, f);

% make a response plot
h0 = getTF(sigAC0, nREFL_I, nEX);
h1 = getTF(sigAC1, nREFL_I, nEX);
h2 = getTF(sigAC2, nREFL_I, nEX);

figure(1)
zplotlog(f, [h0, h1, h2])
title('PDH Response for Detuned Cavity', 'fontsize', 18);
legend('On resonance', '0.1 nm', '1 nm', 'Location','SouthEast');

% make a noise plot
n0 = noiseAC0(nREFL_I, :)';
n1 = noiseAC1(nREFL_I, :)';
n2 = noiseAC2(nREFL_I, :)';

figure(2)
loglog(f, abs([n0 ./ h0, n1 ./ h1, n2 ./ h2]))
title('Quantum Noise Limit for Detuned Cavity', 'fontsize', 18);
legend('On resonance', '0.1 nm', '1 nm');
grid on
```

どの周波数応答のマトリックス(signAC0~1)をどこを振って(nEX)どこで見たか(nREFL_I)を指定して各伝達関数を計算

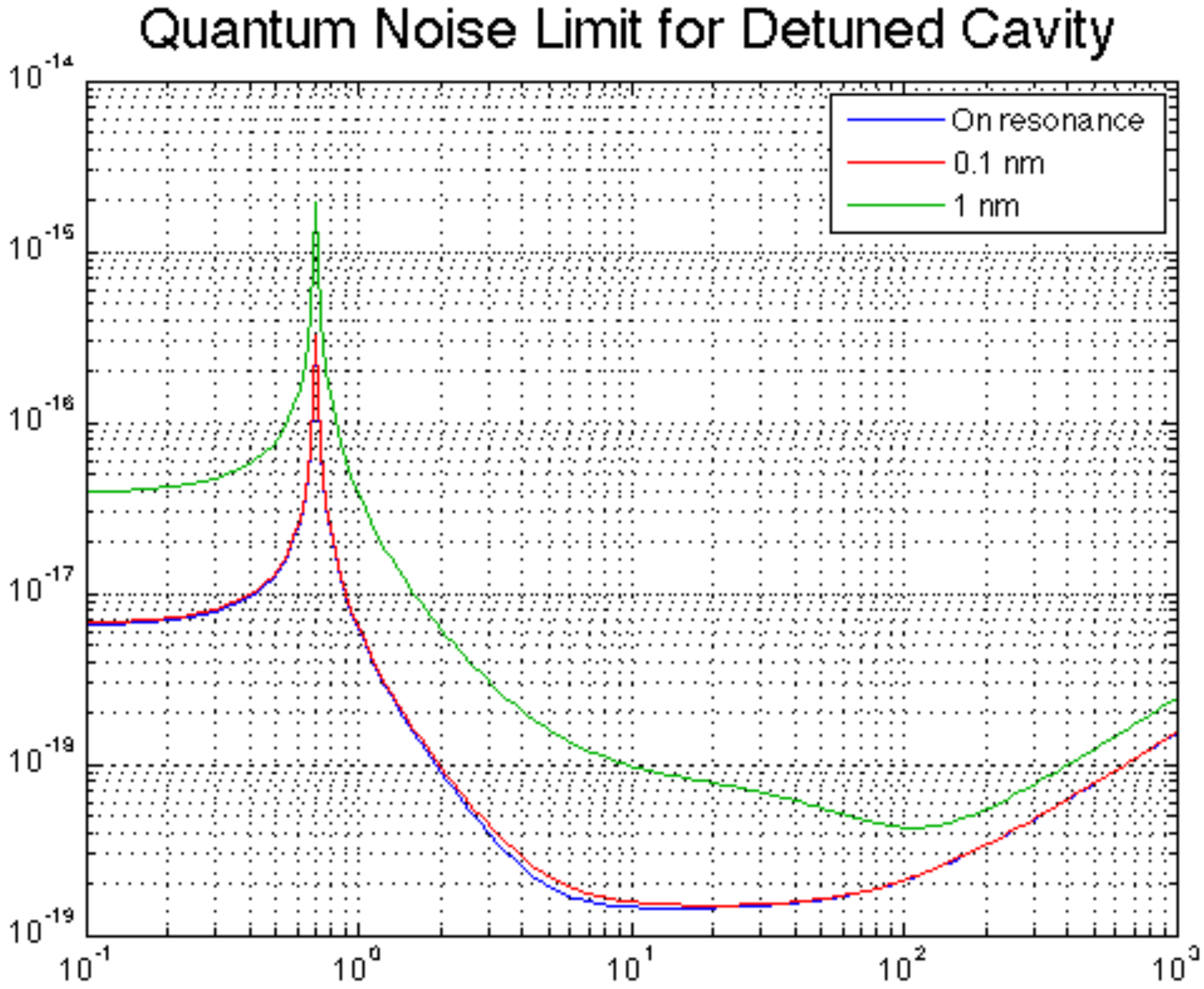
Quantum noiseを計算

Signal to noise ratio計算

‘sigAC’


PDH Response for Detuned Cavity
mag
phase [deg]
On resonance
0.1 nm
1 nm
10^{12}
10^{10}
10^{8}
10^{6}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
50
0
-50
-100
-150
-200

Signal to noise ratio



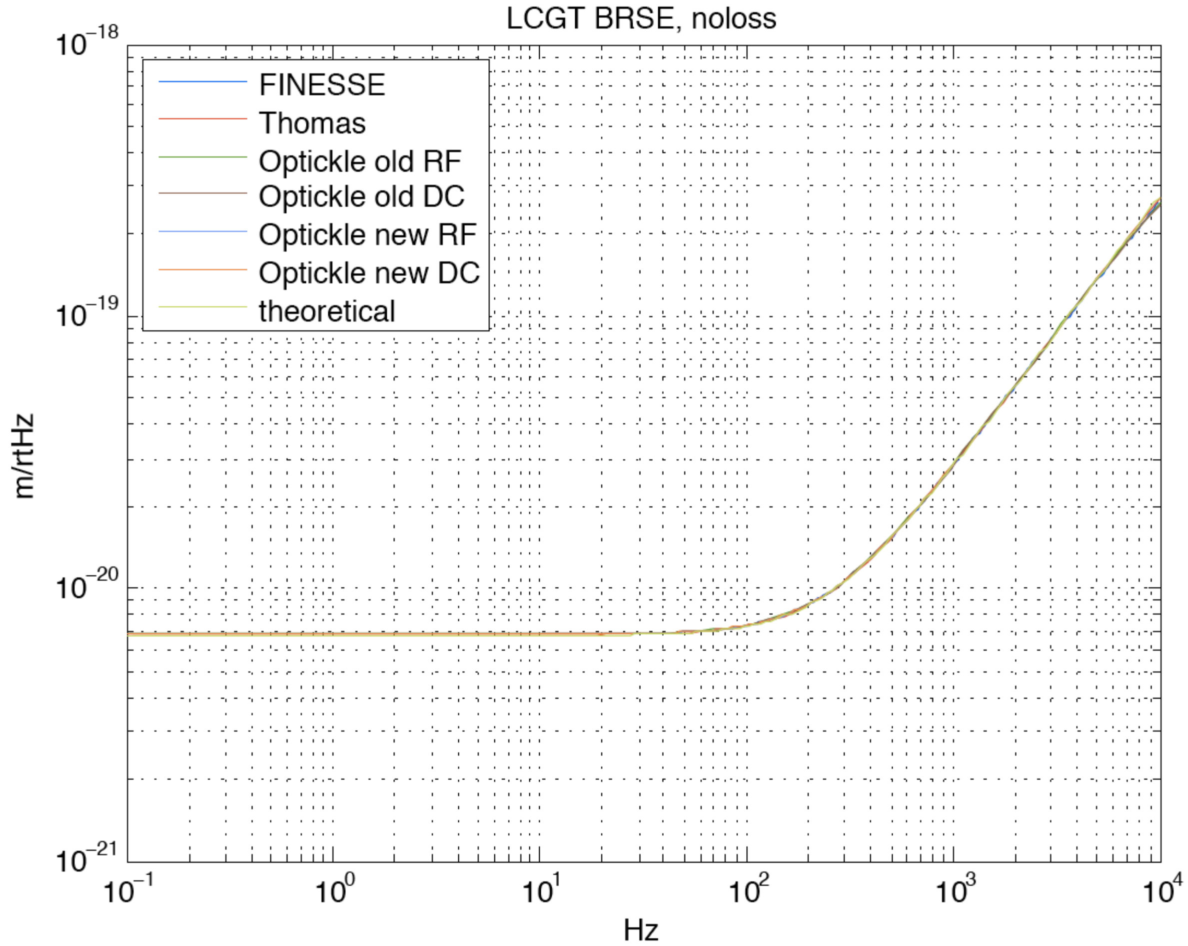
Verifications 1

LCGT BRSEでloss無し, radiation pressure 無し

1. FINESSE
2. Thomasツール
3. Optickle RF(旧バージョン)
4. Optickle DC(旧バージョン)
5. Optickle RF(現バージョン)
6. Optickle DC(現バージョン)
7. 理論曲線

を比較

だいたいOK



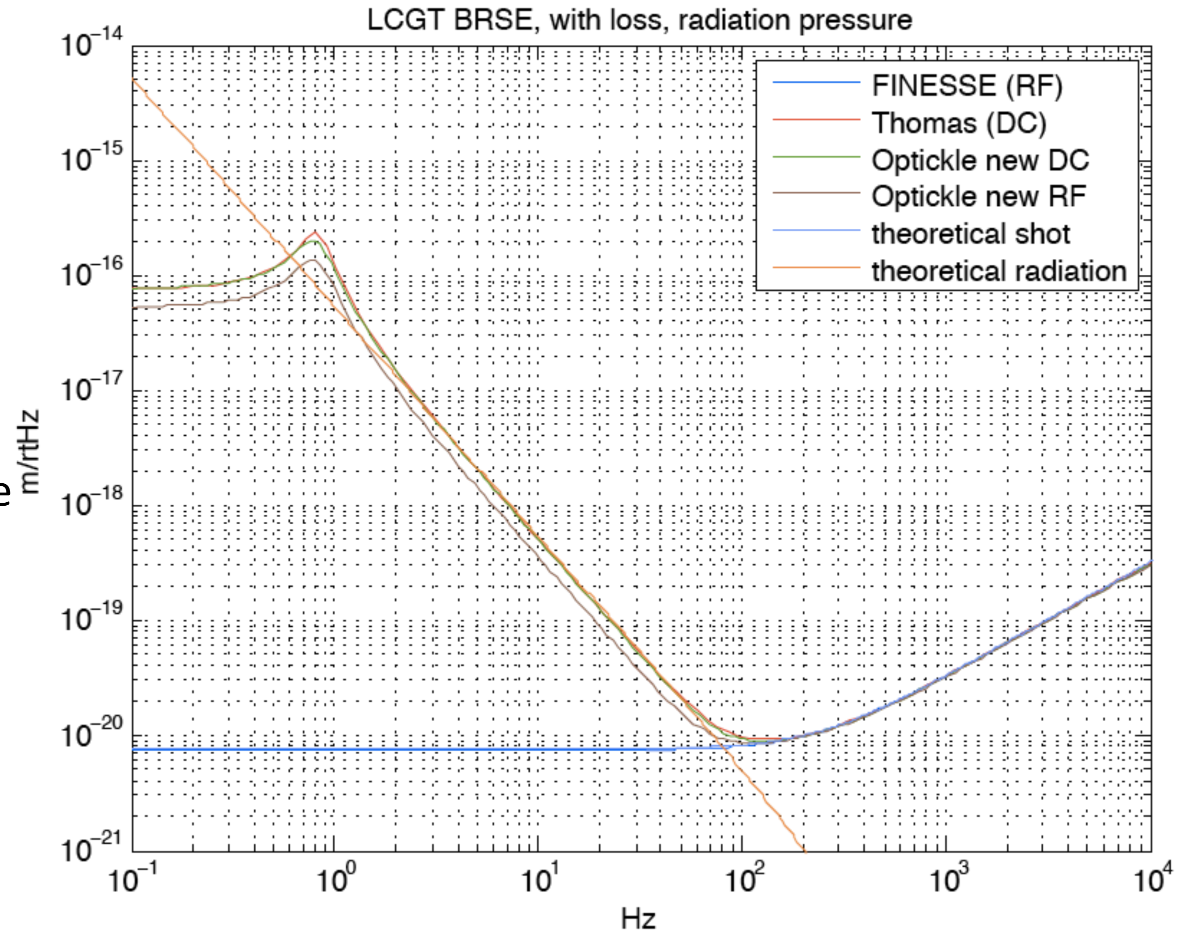
Verifications 2

LCGT BRSEでloss有り, radiation pressure 有り

1. FINESSE
2. Thomasツール
3. Optickle RF(旧バージョン)
4. Optickle DC(旧バージョン)
5. Optickle RF(現バージョン)
6. Optickle DC(現バージョン)
7. 理論曲線

を比較

OptickleのRfのradiation pressureだけsqrt(2)倍小さい、恐らくバグ



Download:
http://gw.icrr.u-tokyo.ac.jp/JGWwiki/LCGT/subgroup/ifo/ISC/Tools

マニュアルに書いていないこと

1. Wikiに上げておいたファイルの
./BRSE_loss_rad/Optickle_new/optLCGT.m

```
opt = setMechTF(opt, nIX, zpk([], wI * dampedRes, 2 / mI));
opt = setMechTF(opt, nEX, zpk([], wE * dampedRes, 2 / mE));
opt = setMechTF(opt, nIY, zpk([], wI * dampedRes, 2 / mI));
opt = setMechTF(opt, nEY, zpk([], wE * dampedRes, 2 / mE));
```

もともと付いてくるサンプルファイルでは1になっている。DC readoutでは2が正しく、RF readoutでは2*sqrt(2)が正しい

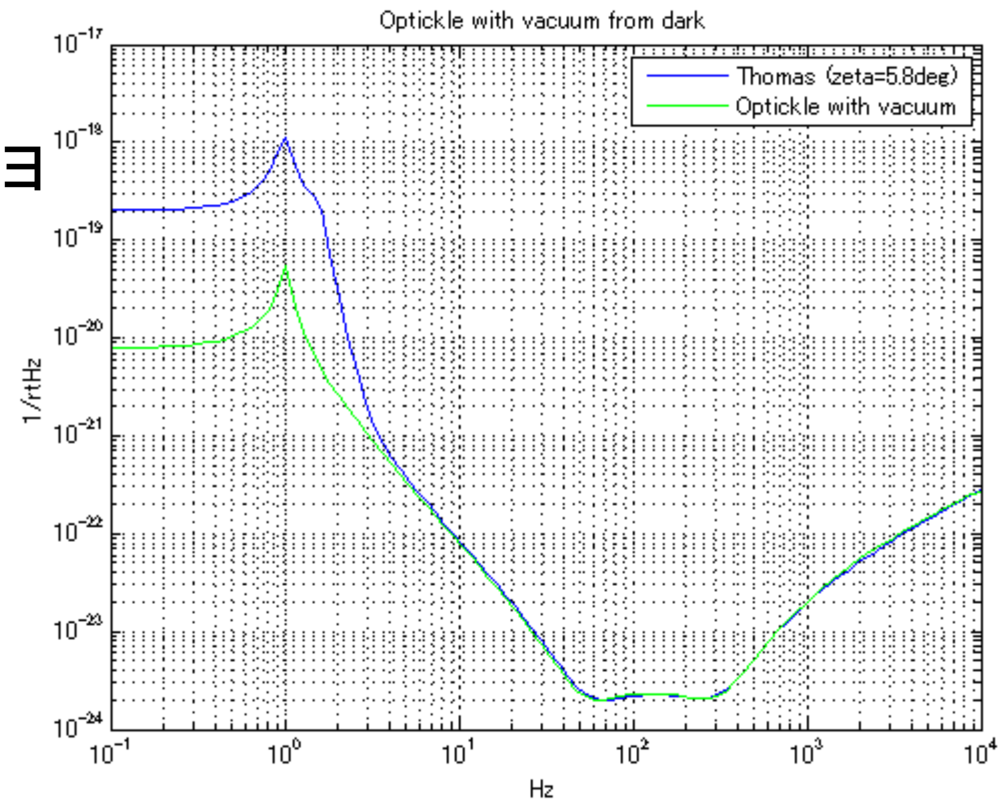
2. 同じディレクトリの
./BRSE_loss_rad/Optickle_new/respDARM_LCGT.m

```
[fDC, sigDC0, sigAC0, mMech0, noiseAC0] = tickle(opt, [], f);

% make a response plot
hXI = getTF(sigAC0, nAS_I1, nEX);
hYI = getTF(sigAC0, nAS_I1, nEY);
hXQ = getTF(sigAC0, nAS_Q1, nEX);
hYQ = getTF(sigAC0, nAS_Q1, nEY);

demph = (-90)*pi/180;
hDARMRF = (hXI - hYI)/2 * cos(demph) + (hXQ - hYQ)/2 * sin(demph);

% make a noise plot
n0 = sqrt((noiseAC0(nAS_I1, :)'.^2 + noiseAC0(nAS_Q1, :)'.^2)/2);
```

NoiseをI とQでとって、2乗和をとってやらないと正しい結果にならない、これは2-photon modeの各モードに相当する。

AdLIGO、DRSEで

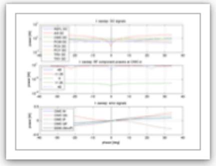
1. 理論曲線
2. Thomasツール
3. Optickle(旧バージョン) with vacuum from dark

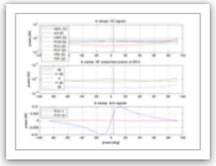


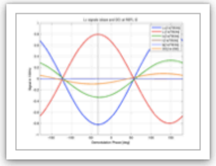
どんなことまでできるのか?

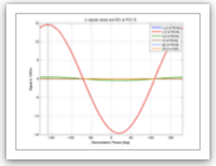
Loopnoiseパッケージ

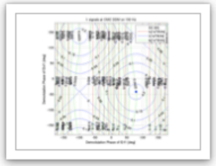
http://gw.icrr.u-tokyo.ac.jp/JGWwiki/LCGT/subgroup/ifo/ISC/TaskList/CoreIFOModel中の20090831_loopnoise274_LCGT2009_new2.zip

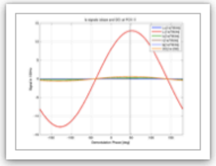

104_DCsweep_I-.png

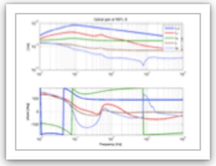

105_DCsweep_Is.png

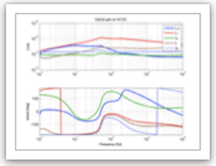

201_SDM_L+.png

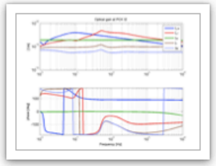

203_SDM_I+.png

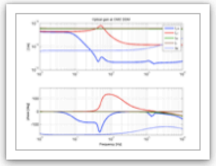

204_DDM_I-.png

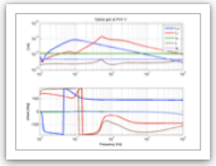

205_SDM_Is.png

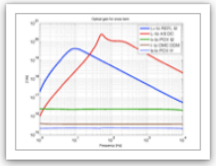

301_optgain_L+.png

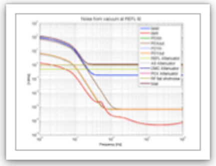

302_optgain_L-.png

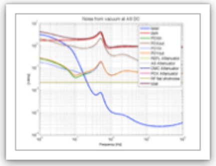

303_optgain_I+.png

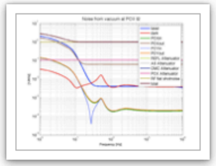

304_optgain_I-.png

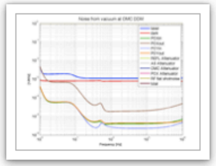

305_optgain_Is.png

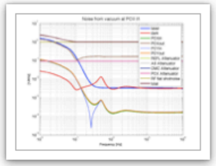

310_optgain_cross.png

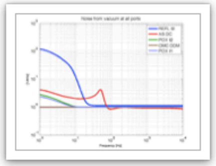

401_noise_L+.png

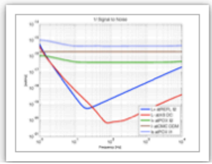

402_noise_L-.png

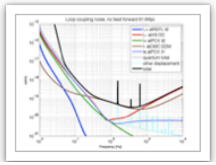

403_noise_I+.png

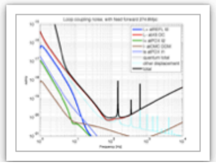

404_noise_I-.png

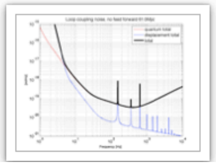

405_noise_Is.png

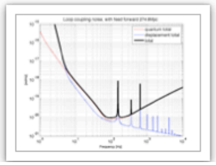

406_allnoise.png

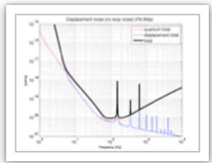






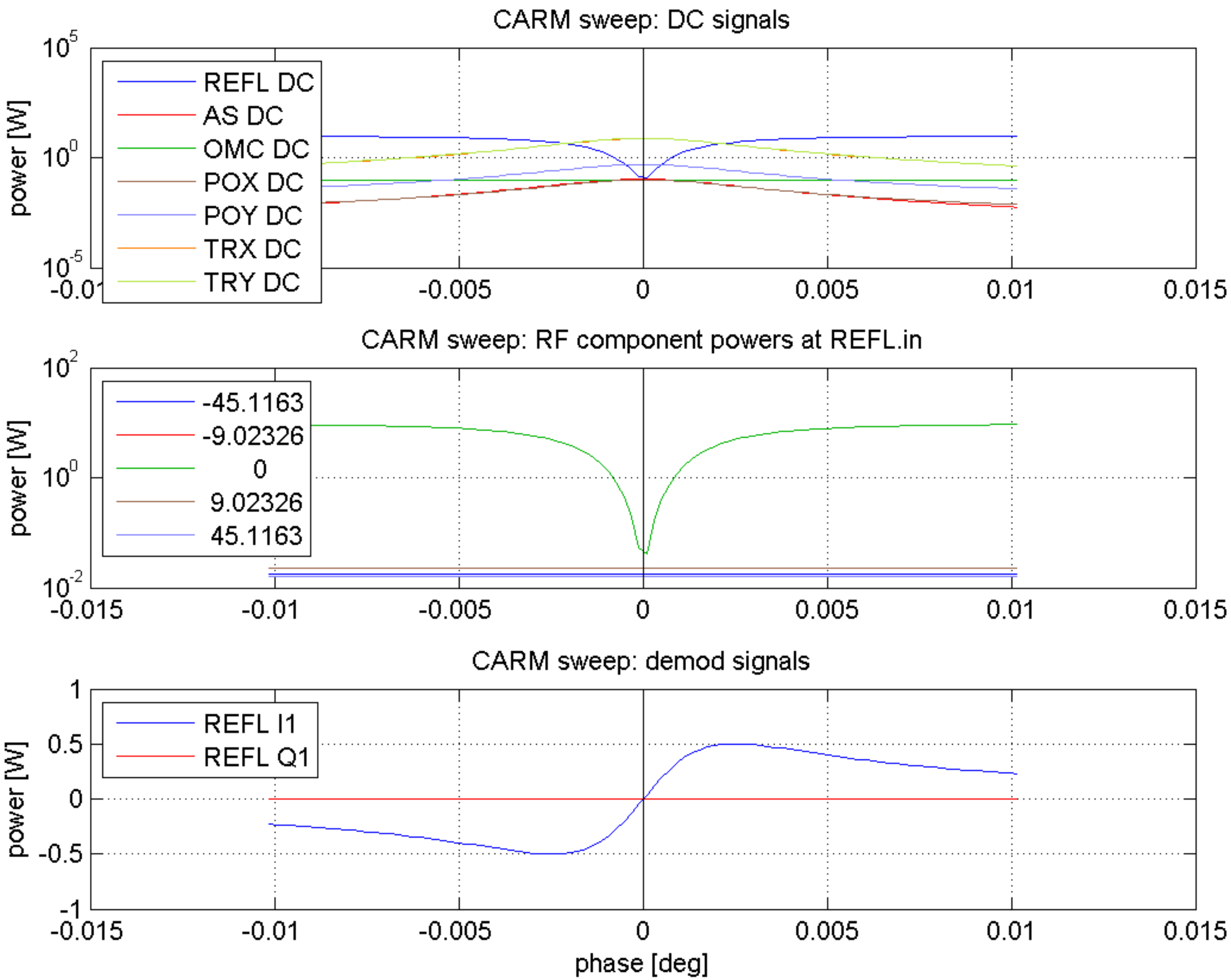






どんなことまでできるのか?

LCGTの各自由度のDC response



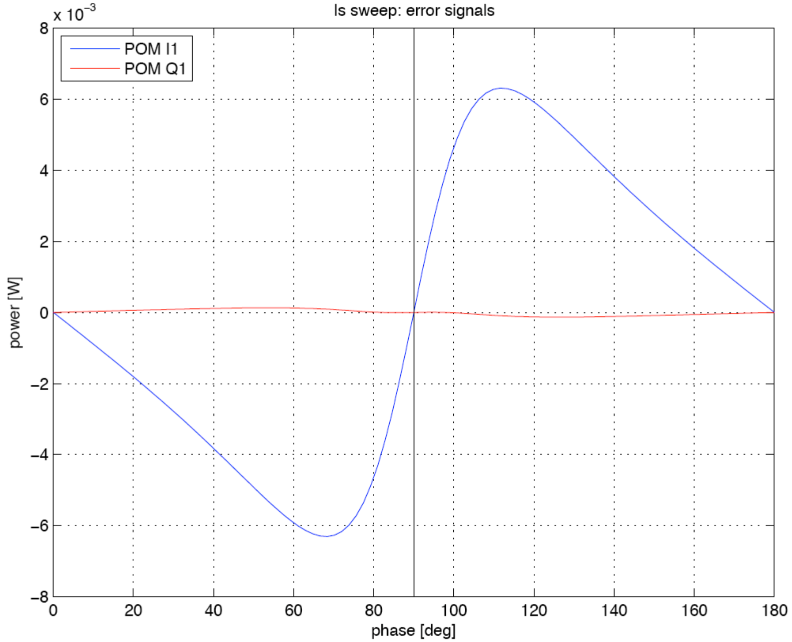
Detuningの範囲と帯域の切り替え

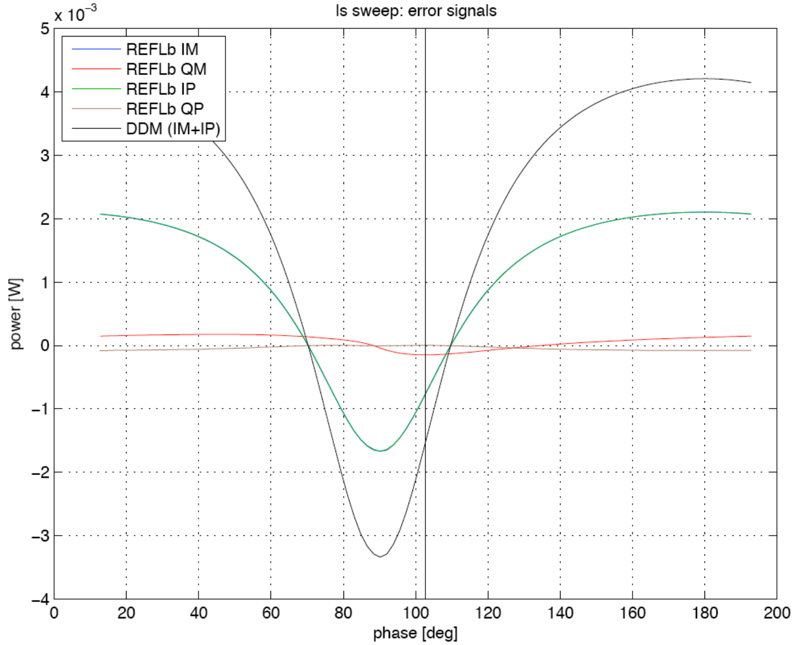
変調方式とSRC信号が線形な範囲

単位は[degree]

変調方式 Lock point	BRSE 90deg	VbBRSE 90deg	VbDRSE 97.7deg	VdBRSE 90deg	VdDRSE 102.8deg	DRSE 105.5deg
ls by SDM	90+/-10.1 ○	90+/-15.8 ○	90+/-16.2 ○	90+/-17.3 ○	90+/-17.2 △	90+/-18.2 △
ls by DDM	90+/-10.1 ○	90+/-15.8 ○	113+/-18 △	90+/-17.3 ○	113+/-18 ○	114+/-18 ○

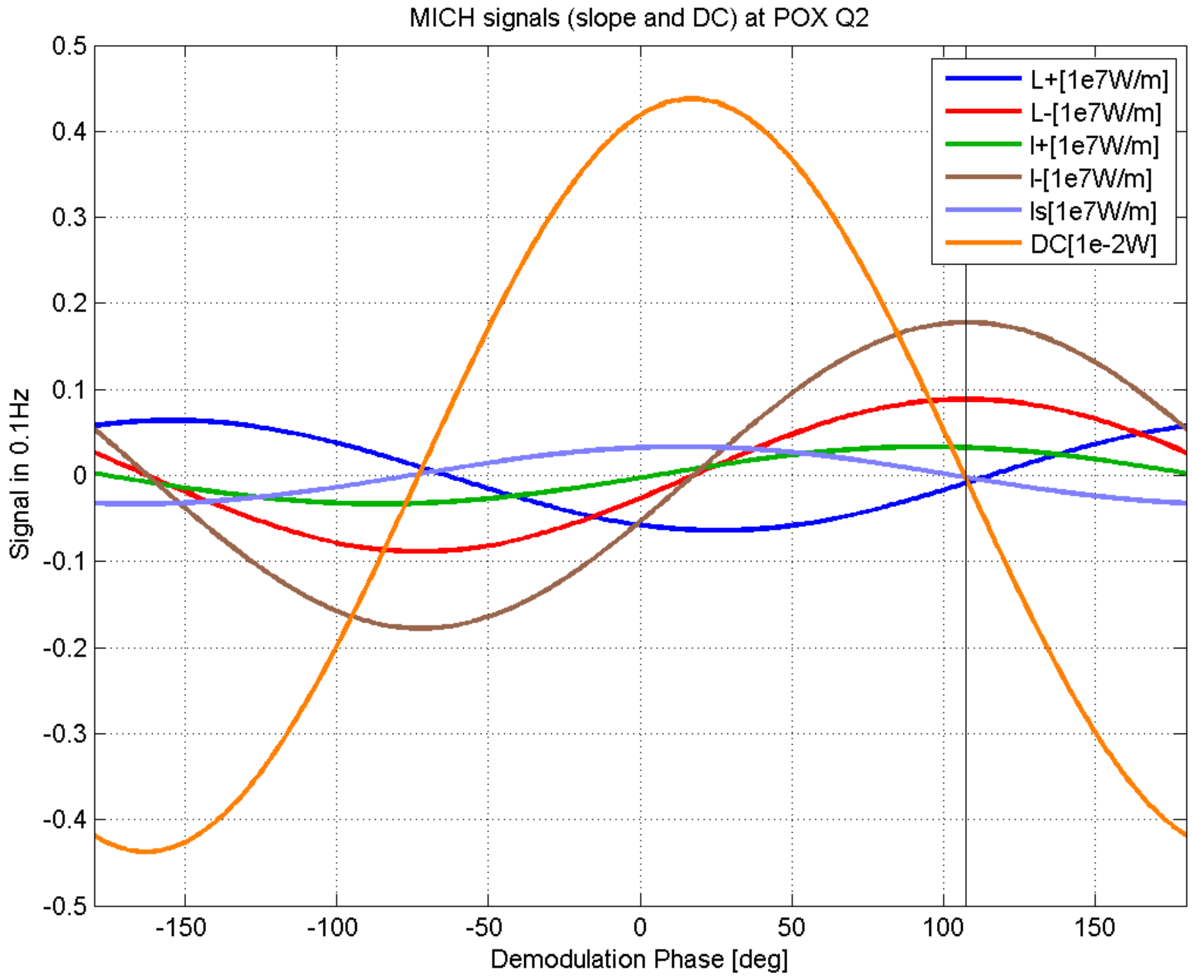
- 切り替えの例:可変狭帯域側をBRSEからDRSEへ





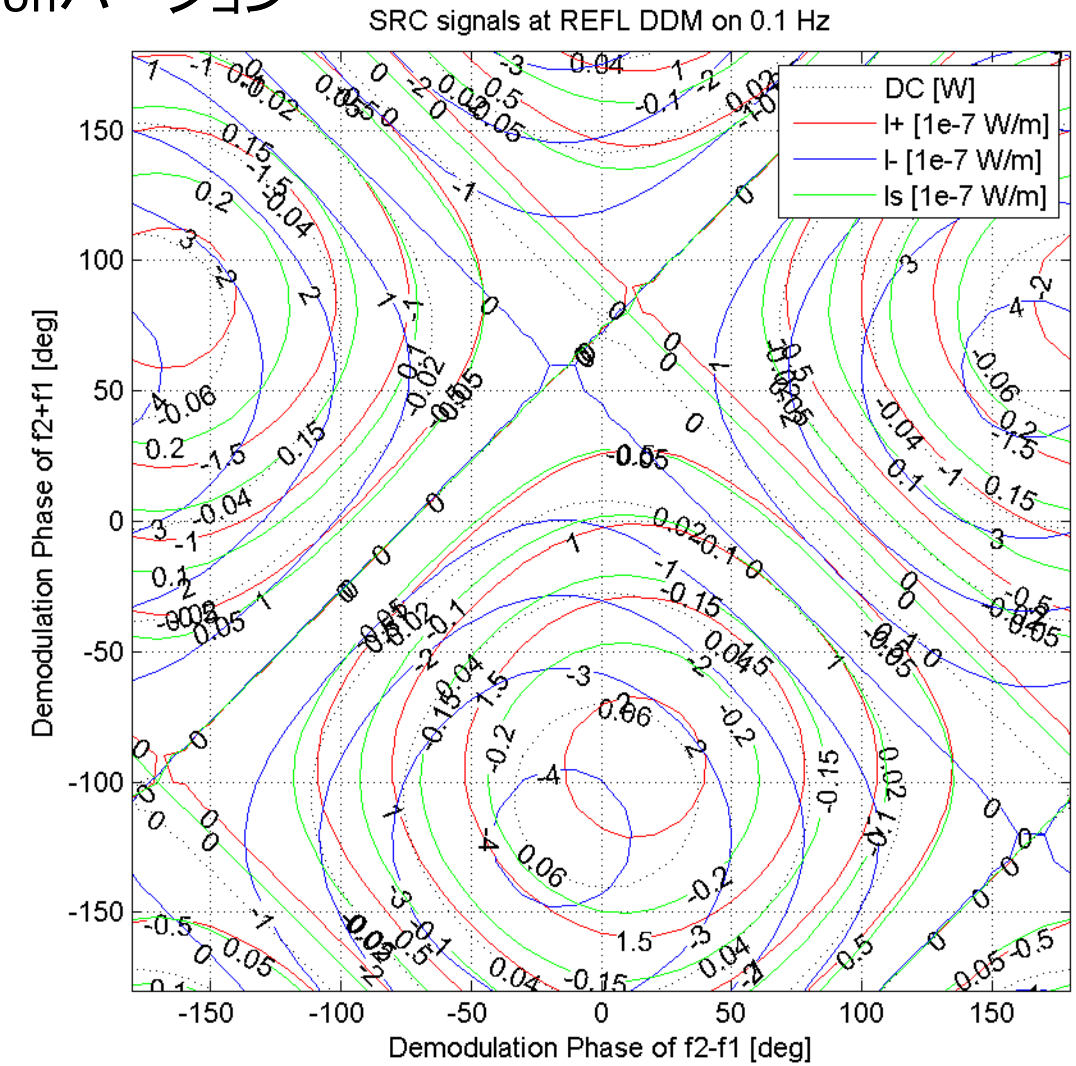
- **帯域可変については制御の面からは問題なさそう**

誤差信号のスロープの
Demoudlation phase依存性

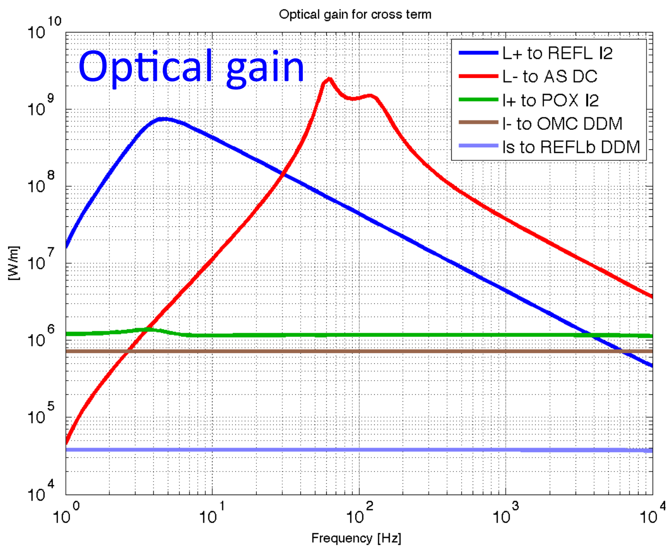


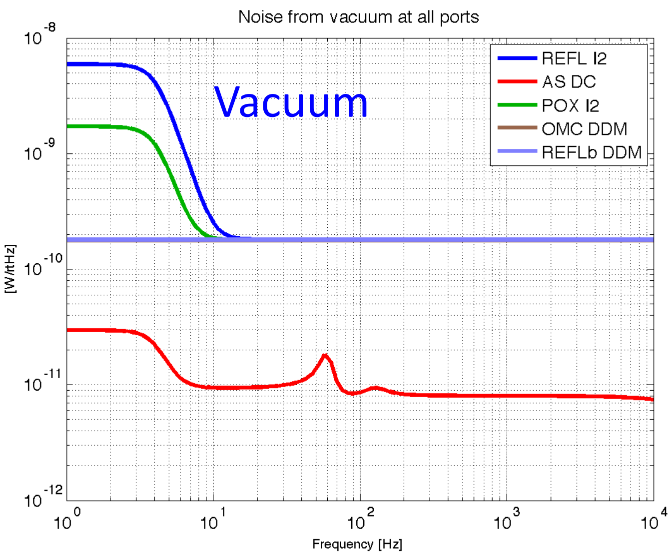
どんなことまでできるのか?

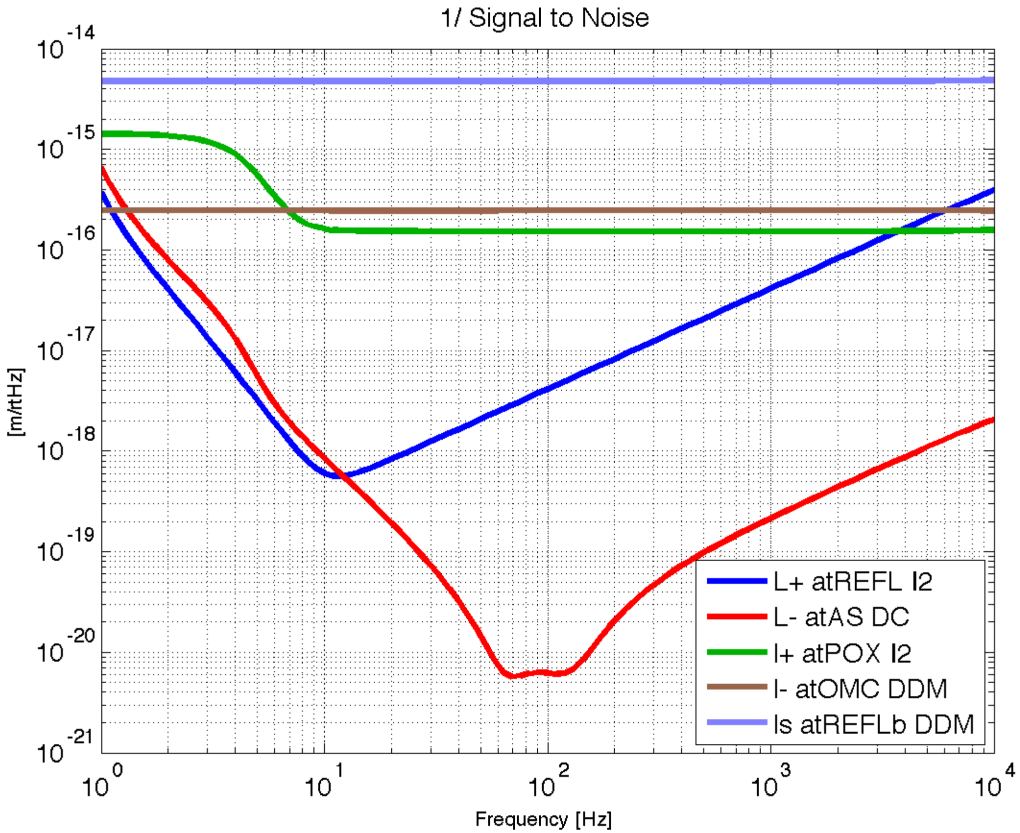
誤差信号のスロープのDemoudlation phase依存性
Double demodlationバージョン



Quantum noise limited sensitivity(例: VdDRSE)

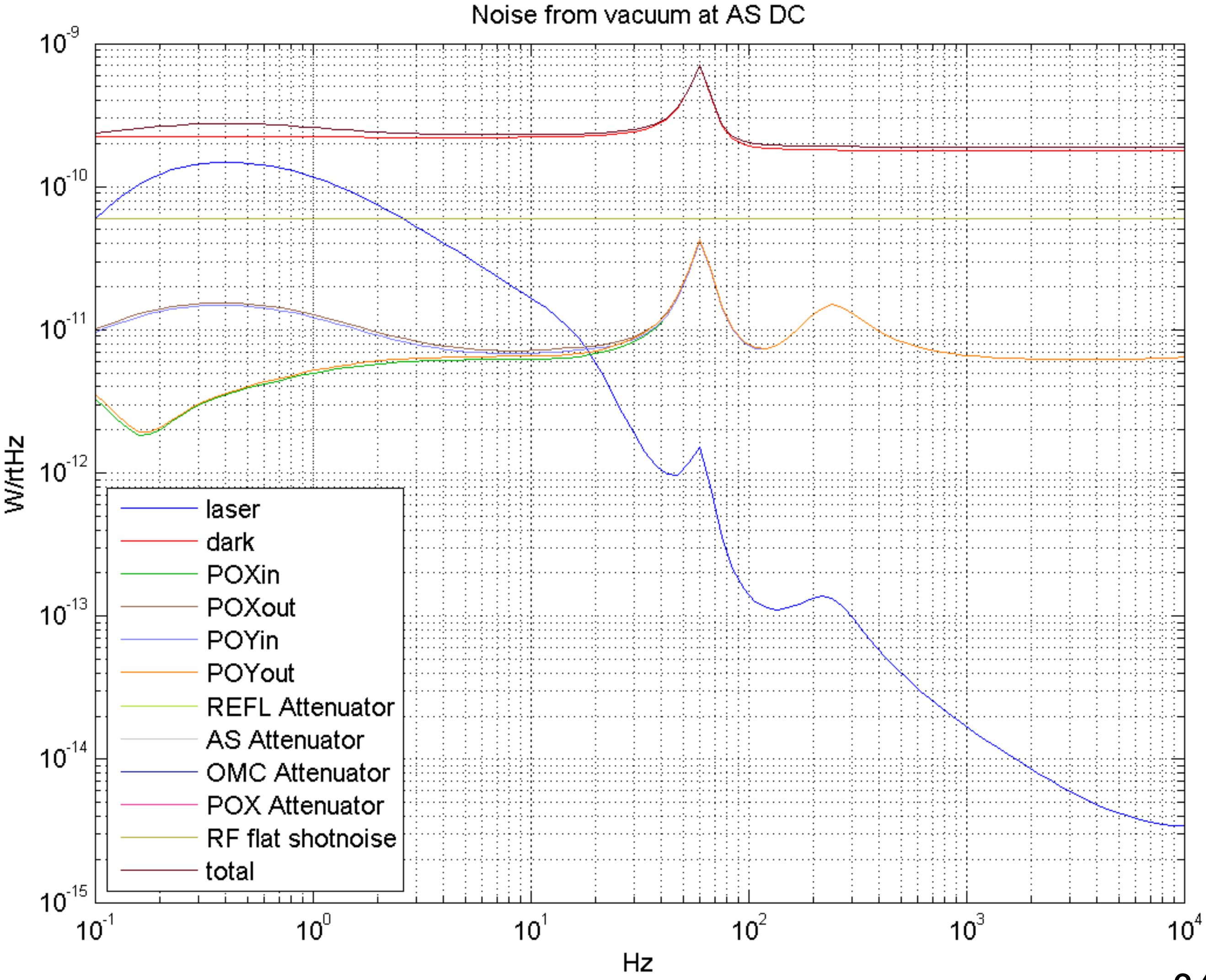




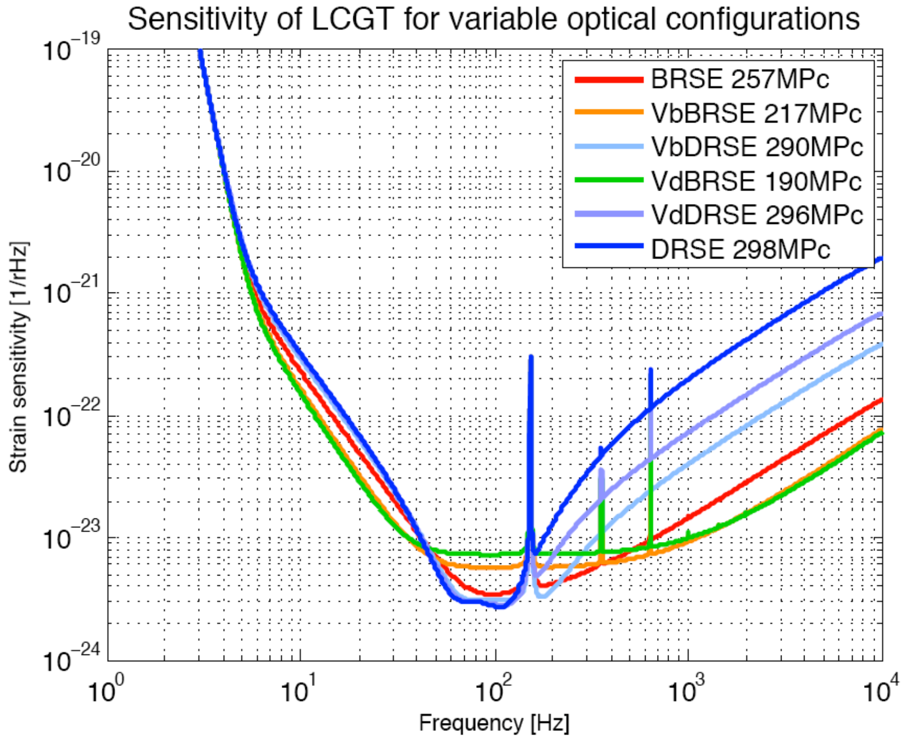


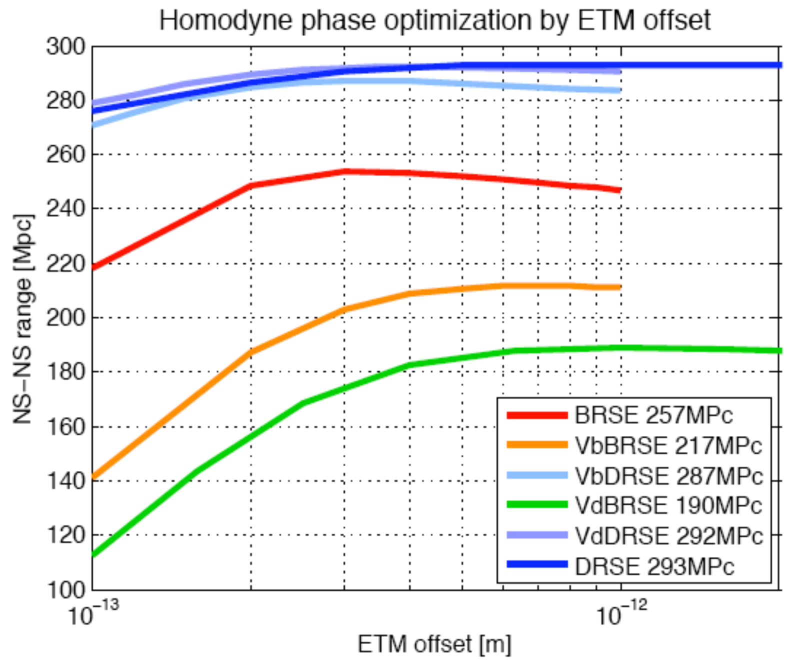
- 実際にはこのように周波数応答がある

Noise from each vacuum component



各検討対象の到達レンジ(変調無し)



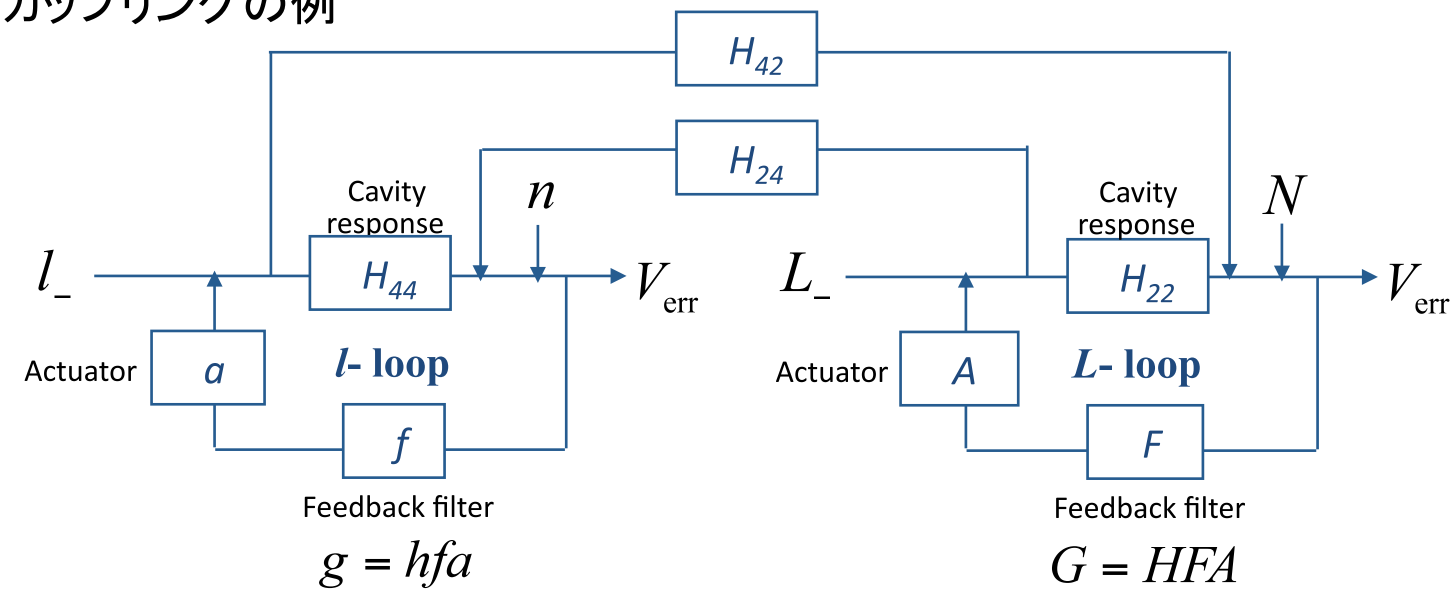


- RF 変調無しのDC readoutでNS-NSの到達レンジをOptickleで計算
- SNR=8で天頂入射（先週の会議の神田さんの定義と同じ）
- ここではDetune phase及びHomodyne phaseの最適化を実行
- 宗宮君の計算よりBRSEが1割強、DRSEでも一部少しいいのはHD phaseの最適化による効果（宗宮君も確認、宗宮計算では80度で固定）
- 原理的にはHD phase大で感度向上だが、大きすぎると非対称性から感度が悪化しだす(右上図)

- これらの感度がループカップリングを考えた場合どれくらい悪化するかを比較検討する

Cross coupling

クロスカップリングの例

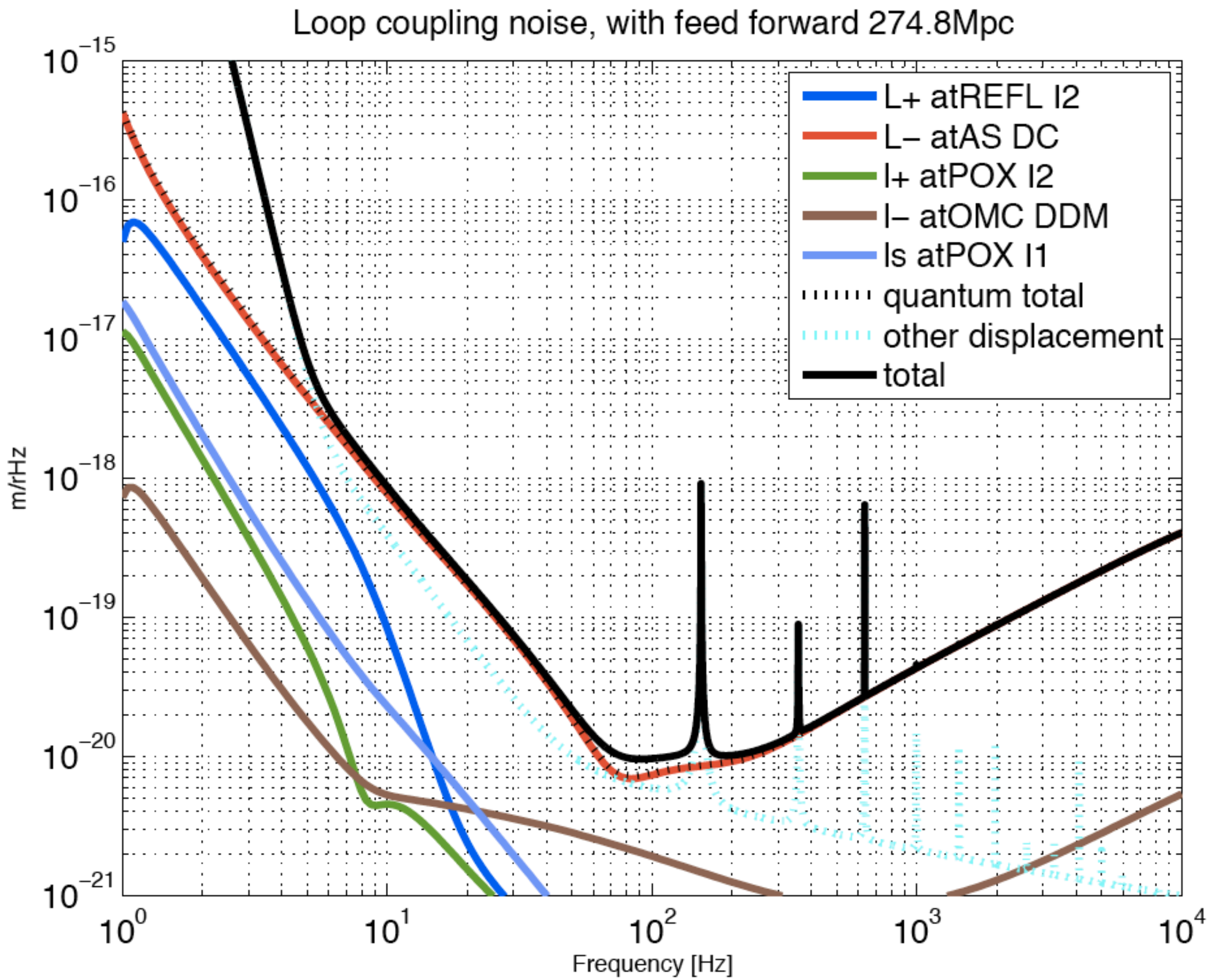


- 実際には5自由度のカップリングがある
- 1次のカップリングのみでなく、2次、3次･･･とあるので、計算では5x5のマトリックス方程式を解いて、 $L_$へのカップリングを求めている
- 輻射圧、輻射圧雑音も全自由度に考慮してある

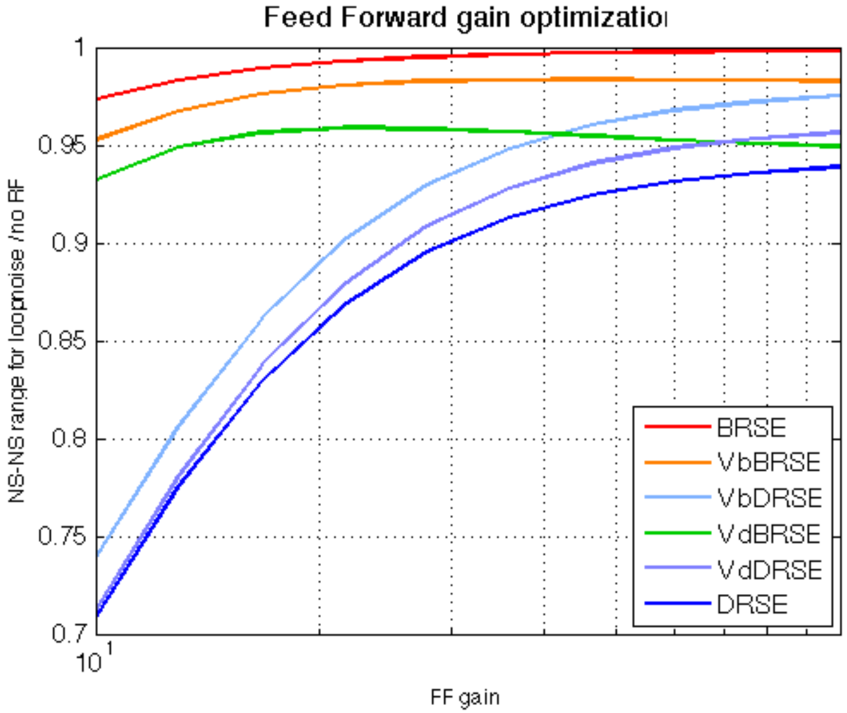
- 仮定
  - f^{-1}のフィードバックフィルター

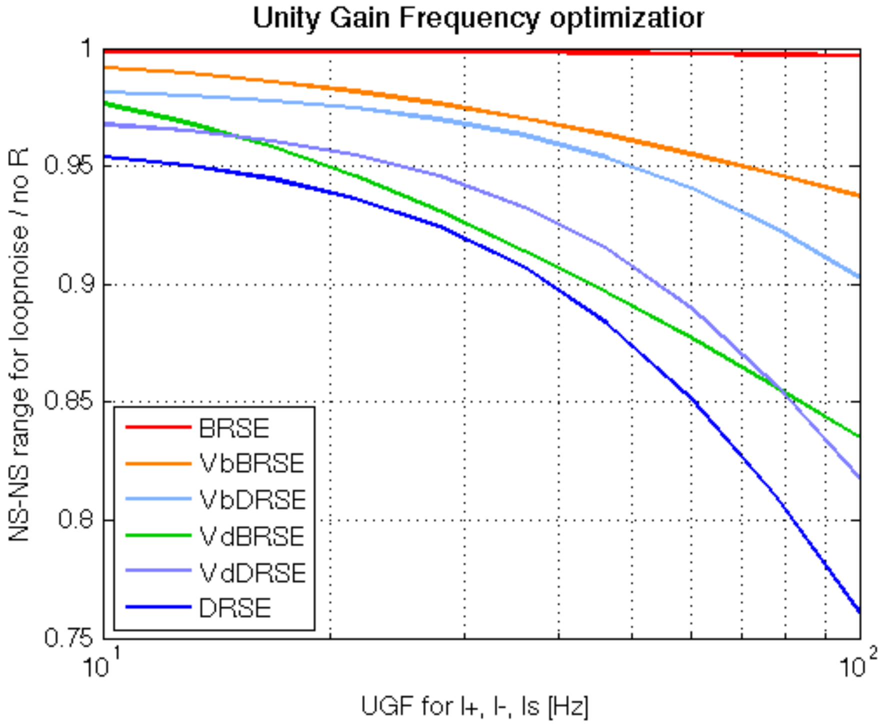
どんなことまでできるのか?

Loop coupling noise of LCGT



その他最適化



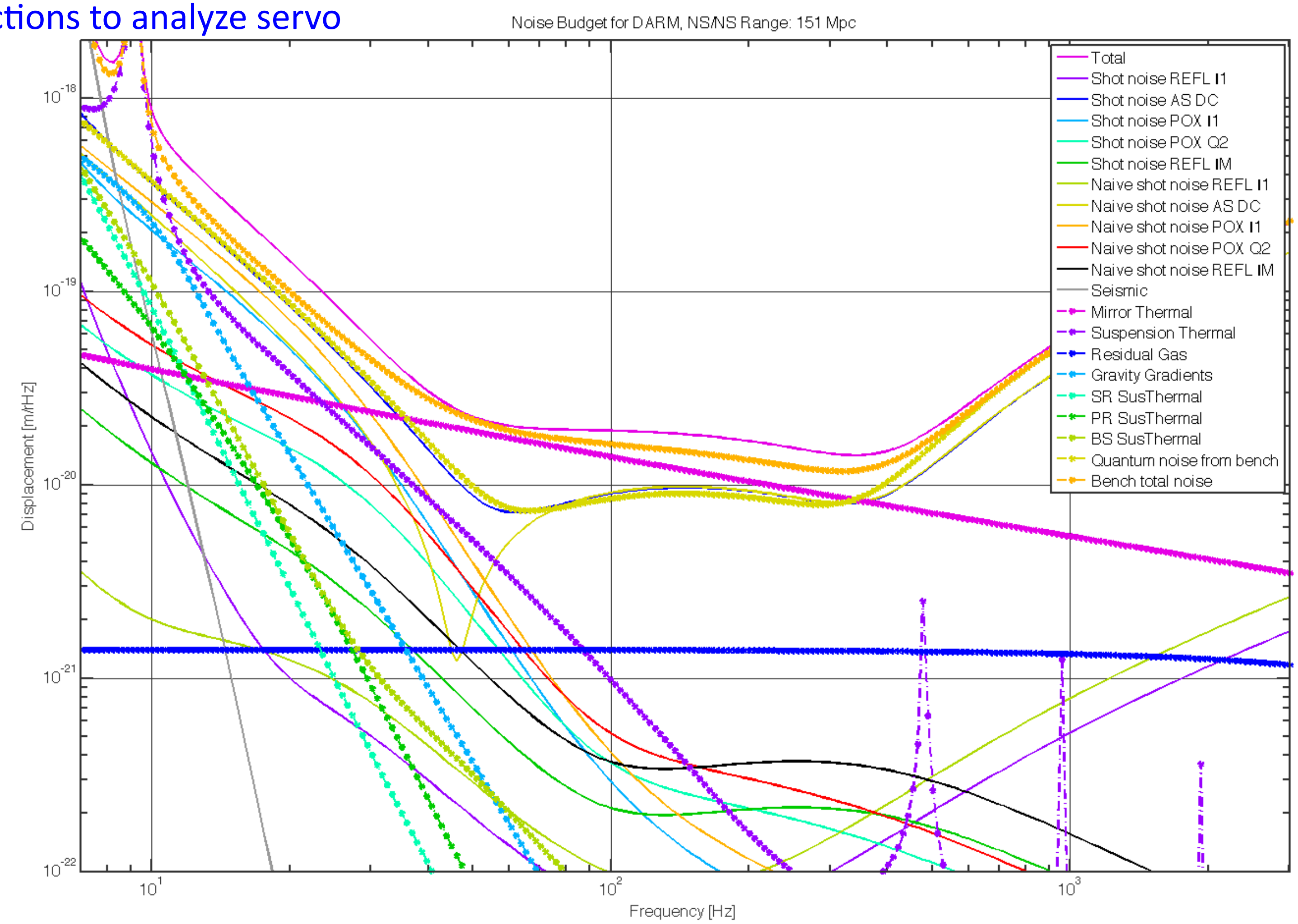


- FF gain、UGFともにBRSE、もしくは可変でも広帯域よりの方が楽

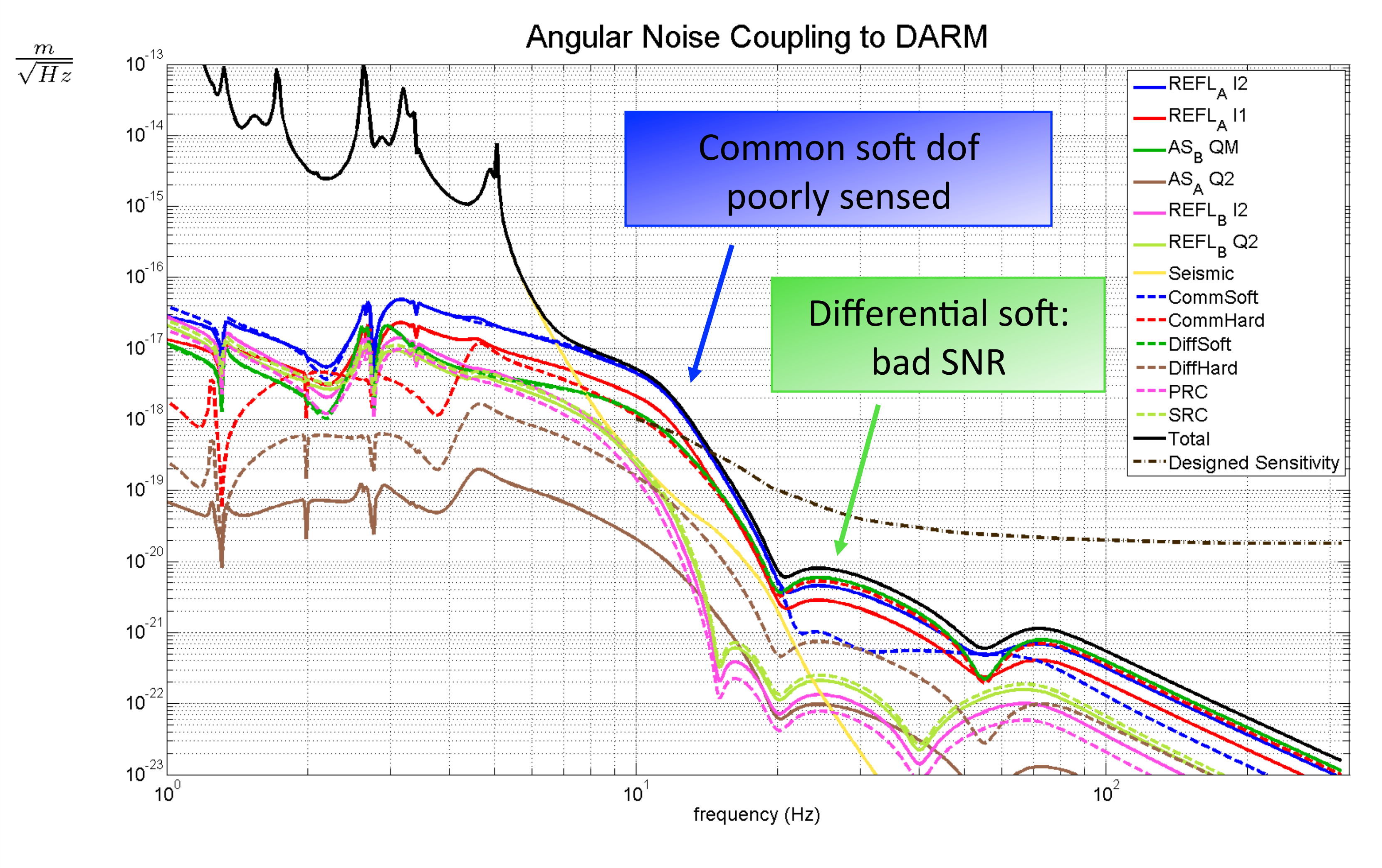
どんなことまでできるのか?

Looptickle by Stefan Balmar

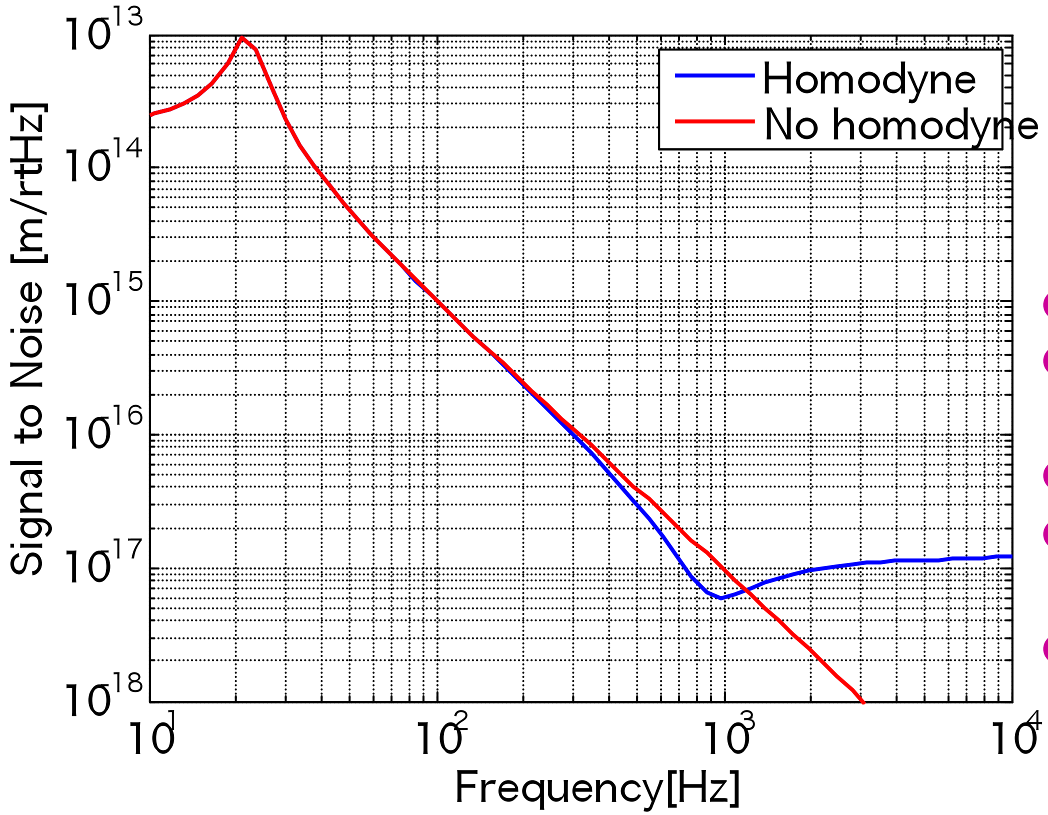
- Displacement noise
- Many functions to analyze servo



AdLIGO alignment simulation by Lisa Barsotti



QND experiment at NAOJ

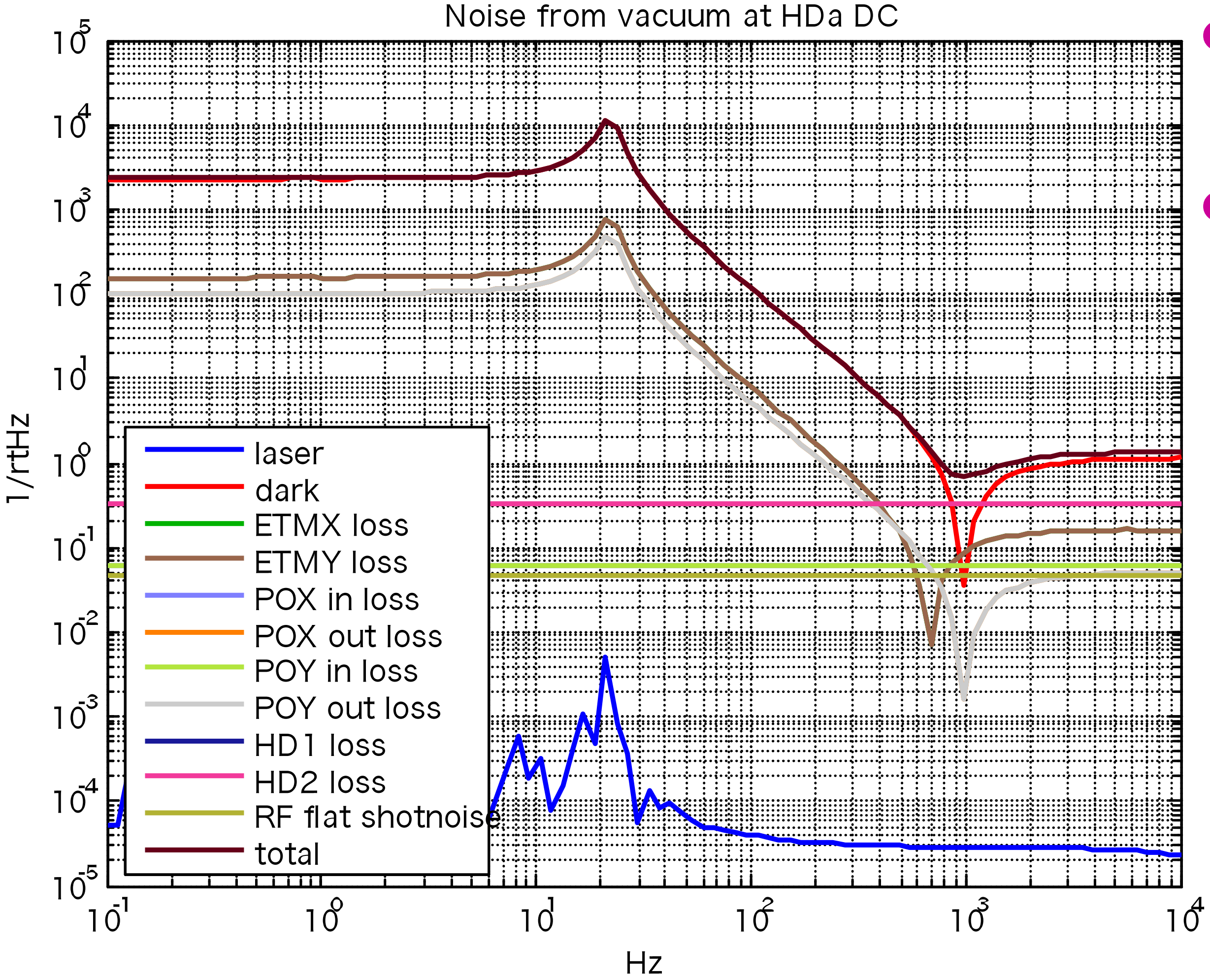


- Frequency domain interferometer simulator for quantum noise
  - Supports radiation pressure
  - Supports vacuum injection
  - Supports homodyne detection

- Front mirror loss = 30ppm
- End mirror loss (透過率含む) = 10ppm
- Loss after squeezing = 22%
- Homodyne angle = 2.5e- 4 rad

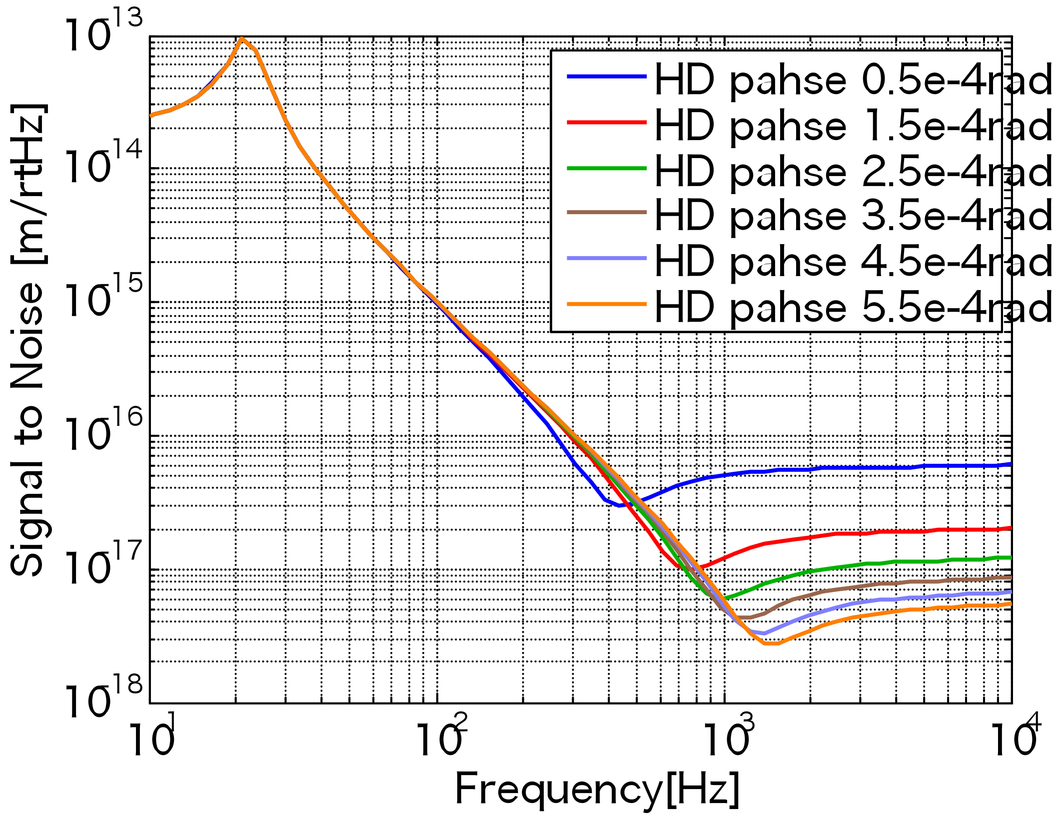
- 2-3dB squeezingが期待できる

Noise from vacuum



- 干渉計からPDまでのロスが支配的 downstream interferometer is dominant.
- RFからくるshotnoiseは無視できる
  - Output mode cleanerはいらない

Dependence on Homodyne angle



- 1e-5 ~ 1e-4 [rad]の正確さでHomodyne phaseをコントロールしなければならない

LCGTの干渉計、制御の計算をできる人を育てたい

興味のある人募集

詳しくは宮川もしくは麻生まで