

レーザー干渉計型重力波検出器のデジタル制御

東京大学理学系研究科物理学専攻

宇宙線研究所重力波グループ

中川憲保 (46072)

平成18年1月17日

重力波の研究にあたって

アインシュタインが一般相対性理論を 1916 年に発表してから 90 年ほど経とうとしており、現在その理論は中性子星やブラックホールといった半径 10km 程度のコンパクトな天体から 10^{23} km オーダーの宇宙をも支配する法則であると多くの人に信じられている。

そのアインシュタイン理論が予言するもののなかに重力波がある。理論が正しければ重力波は必ず存在するが、実験的な検証が必要である。実際に重力波を発生させて検証する事は難しい。間接的な検証としては、J.H.Taylor らの連星パルサー PSR1913+16 の観測からその存在が証明されている。[1]

重力波が何故発生しにくいかは、重力相互作用の弱さに起因している。電磁相互作用の強さは微細構造定数 $\alpha = e^2/\hbar c = 1/137$ で特徴づけられるが、これに対応する重力の強さを表す定数 $\alpha_G = Gm_p^2/\hbar c = 7 \times 10^{-39}$ となる。相互作用が弱い故に重力波は発生させにくい。これは重力波に関する物理の弱点であるが、一方で最大の長所となりうる。重力波は物質との相互作用が弱い故に、逆にいったん発生すれば途中の物質の影響をほとんど受けずに情報を伝達するはずである。例えば超新星爆発に伴う中性子星形成を考えてみる。まず、光で観測できるのは爆発前の星の表面程度でそれより内側は光が多数の電子、原子によって散乱吸収されるため情報を得る事は不可能である。しかし、ニュートリノならば星の内部のような低密度層は突き抜ける。実際 SN1987A では中性子星表面からの情報をその外側に存在する $\sim 15M_\odot$ のガスの存在にもかかわらず我々にもたらした。しかし、ニュートリノを使っても中性子星の中心の情報、例えば中心密度を知る事は難しい。あまりにも密度、温度が高いためにニュートリノといえども散乱吸収を受けるからである。しかし、重力波なら、中性子星の中心で発生すればほとんど散乱吸収を受けない。[3]

このような点から、重力波がもたらすこれまでにない情報の獲得を目指して、そして物理、天文に新たな世界が開かれる事を夢見て、まずはその検出を目指してレーザー干渉計型重力波検出器の研究を進めていく事にする。

デジタルについて

今回制御にデジタルを用いるにあたって、まずはじめに、デジタルがアナログに対し全てに於いて勝るといえるようには考えていない。ダイナミックレンジについては、ノイズレベルが数 nV である回路が十数 V まで、つまり 10 桁に及ぶレンジを問題なく扱う事が出来る。これに対して、デジタルは内部演算に関しては double による 64bit 浮動小数点演算が出来ることで、十数桁のダイナミックレンジが保証されるが、アナログからデジタルに変換する際に 16bit で変換され、そのレンジは 5 桁となる。この点ではアナログの方が優秀である。

しかしフィルタを設計する際に、アナログが決まった値の抵抗等の部品から設計しなければならないのに対し、デジタルでは自由に値を設定できる（サンプリング定理等を満たす必要はあるが）。またフィルタの設計変更の際も、アナログではハンダやコテを使い装着し直さなければならないが、デジタルではプログラムの数字を書き換えるだけで済む。他にもデジタルでは複雑な演算が組みやすいことや、その都度変わるゲイン等の記録をすることが可能であったりと、アナログにはない良さがある。

最終的に再びアナログに置き換えることになる場合も、それまでの実験にデジタルを用いることは開発の手間から考えると有効である。また、デジタルに関する世の中の技術も未だ発展に陰りが見えず、干渉計を開発する手段の一つとしてデジタルを用いて研究することにした。

本論文の構成

本論文の構成について、概略を書いておく。

第 1 章では、レーザー干渉計の重力波検出に関して、重力波の導出と、Michelson 干渉計による検出について書いてある。

第 2、3 章では Fabry-Perot 共振器についてその性質と制御、特に長さ方向の制御について書いてある。

第 4、5 章ではデジタル信号処理とそのシステムについてフィルタの設計、今回用いるシステムのノイズレベルや遅延、デジタル特有の問題点について書いてある。

第 6 章では、デジタルシステムによる Fabry-Perot 共振器の制御について書いてあり、この章が本論文の主となる部分である。

そして、第 7 章で結論と今後についてまとめてある。

最後には補足の章を載せて以上が本論文の内容となる。

目次

第 1 章	レーザー干渉計による重力波検出	6
1.1	重力波	6
1.2	レーザー干渉計	8
1.2.1	Michelson 干渉計	8
1.2.2	干渉計による重力波検出	9
第 2 章	Fabry-Perot 共振器の性質	11
2.1	Fabry-Perot 共振器の応答	11
2.2	鏡の変位と光量	12
2.3	制御時の信号	15
第 3 章	Fabry-Perot 共振器の制御	16
3.1	鏡の変位と速度	16
3.1.1	鏡の変位	17
3.1.2	鏡の速度	19
3.2	コイルとアクチュエーター	21
3.2.1	アクチュエーター、コイルのレンジ	21
3.2.2	feedback のゲインと、返す時間	22
3.2.3	コイルでの過渡現象	22
3.3	lock acquisition	24
3.3.1	lock acquisition とは	24
3.4	Fabry-Perot 共振器を lock する手段	24
3.4.1	error signal の広帯域化	25
3.5	現実の Fabry-Perot 共振器で起きる複雑な現象	26
3.5.1	共振器内のドップラー効果	26
3.5.2	規格化による弊害	27
3.6	Fabry-Perot 共振器の制御	28
第 4 章	デジタル信号処理	29
4.1	デジタルとアナログ	29
4.2	演算法のまとめ	30

4.2.1	z 変換	30
4.2.2	Fourier 変換	30
4.3	IIR フィルタの設計	31
4.3.1	S-Z 変換	31
4.3.2	直接型	33
4.3.3	縦続型	35
4.4	遅延時間	36
4.4.1	位相遅延特性	37
4.4.2	群遅延特性	37
4.5	デジタル信号処理	38
第 5 章	デジタルシステム	39
5.1	DSP の特徴	39
5.1.1	積和演算器	39
5.1.2	ソフトとハード	40
5.2	DSP の評価	40
5.2.1	用いたデジタルシステム	40
5.2.2	ノイズレベル	41
5.2.3	位相遅延	41
5.3	時間に関する問題点	44
5.3.1	サンプリング定理	44
5.3.2	遅延による位相の遅れ	44
5.3.3	演算速度	45
5.4	レンジに関する問題点	46
5.4.1	ダイナミックレンジ	46
5.4.2	桁落ち	46
5.4.3	量子化雑音 (丸め誤差)	47
5.4.4	オーバーフロー	49
5.5	デジタルシステム	50
第 6 章	TAMA300 の Fabry-Perot 共振器片腕 mass lock	51
6.1	実験部干渉計構成	51
6.2	NSmass lock 用の error signal 規格化プログラミング	53
6.2.1	フィルタ部分の設計	53
6.2.2	規格化部分の設計	54
6.2.3	トリガーの追加	57
6.2.4	オーバーフロー対策	58
6.2.5	演算速度	58
6.3	Fabry-Perot 共振器 lock テスト	59

6.3.1	lock acquisition 測定	59
6.3.2	測定結果	60
6.3.3	デジタルフィルタによる lock acquisition	67
6.3.4	再現性	67
6.4	デジタルフィルタによる lock acquisition	69
第 7 章	結論	70
7.1	デジタルシステムの評価	70
7.2	規格化を用いた lock acquisition	71
7.3	干渉計制御に際して	71
7.4	最後に	72
付 録 A	Feed-back range の変更	75
A.1	range 変更の設定	75
A.2	range による lock 成功率	75
付 録 B	デジタルシステムに用いたボード	77
B.1	TMS320C6713DSK	77
B.2	DSK6000IFA	79
B.3	他デジタルシステム	79
B.3.1	汎用プロセッサ	80
B.3.2	FPGA や CPLD	80
付 録 C	Fabry-Perot 共振器ロック用フィルタ	81
付 録 D	使用した各種プログラム	85
D.1	桁落ち現象検証用プログラム	88
D.2	演算時間測定用プログラム	88
D.3	規格化、トリガー組み込み済みデジタルフィルタプログラム	88

第1章 レーザー干渉計による重力波 検出

重力波検出の原理は、長さを測る事である。重力波は時空の波であり、時空の変動となって現れ、2点間の距離を変化させる。その長さの変化を捕らえる事で検出する事が出来る。

現在研究されている重力波検出器として、主には弾性体振動子をアンテナとし、伸縮モードの変化を測定する共振型と呼ばれるタイプと、レーザー干渉計を用いたタイプ(自由質点型)がある。共振型検出器は弾性体の共振周波数付近でのみ感度を持つ一方、レーザー干渉計はレーザーや光学部品の技術進歩もあり、広い帯域での観測が可能なため、現在はレーザー干渉計の研究が盛んになっている。

この章では、

1. Einstein 方程式の線形化による重力波の導出とその偏向。[2]
2. レーザー干渉計による重力波の検出。[3]

について述べる。

1.1 重力波

重力波は一般相対論からアインシュタイン方程式を解く事により求められる。
まずはじめにメトリックテンソルを

$$g_{\alpha\beta} = \eta_{\alpha\beta} + h_{\alpha\beta} \quad (1.1)$$

とする¹。そして h の trace reverse テンソル \bar{h} を次のように定義する。

$$\bar{h}^{\alpha\beta} \equiv h^{\alpha\beta} - \frac{1}{2}\eta^{\alpha\beta}h \quad (1.2)$$

$$h^{\alpha\beta} \equiv \bar{h}^{\alpha\beta} - \frac{1}{2}\eta^{\alpha\beta}\bar{h} \quad (1.3)$$

$$(h \equiv h^{\alpha}_{\alpha}, \bar{h} \equiv \bar{h}^{\alpha}_{\alpha})$$

¹ここでは、 $\eta_{\alpha\beta} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ である。

これらを用いてアインシュタイン方程式にゲージ条件としてローレンツゲージを課すと、真空中で線形化されたアインシュタイン方程式は

$$\square \bar{h}_{\alpha\beta} = 0 \quad (1.4)$$

である²。この方程式の解として、

$$\begin{aligned} \bar{h}_{\alpha\beta} &= A_{\alpha\beta} \exp(ik_{\mu}x^{\mu}) \\ (\text{ただし、} k_{\alpha}k^{\alpha} &= 0, A^{\alpha\beta}k_{\beta} = 0) \end{aligned} \quad (1.5)$$

なる波動解があり³、これが重力波の解となる。さらにこの解にゲージ条件を課すと、

$$A^{\alpha}_{\alpha} = 0 \quad (1.6)$$

$$A_{\alpha\beta}U^{\beta} = 0 \quad (1.7)$$

(U^{β} は時間的単位ベクトル)

このゲージのもとでは $\bar{h}_{\alpha\beta} = h_{\alpha\beta}$ である。

この時 U^{β} を Background Minkowski 時空の時間基底として、重力波の進行方向を Z 軸にとると、

$$A_{\alpha\beta} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_{xx} & A_{xy} & 0 \\ 0 & A_{xy} & -A_{xx} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.8)$$

$$h_{\alpha\beta} = A_{\alpha\beta} \exp(ik_{\mu}x^{\mu}) \quad (1.9)$$

となる。 A_{xx} と A_{xy} は重力波の直交する2つの偏光を表している。

²ここでは、 c を真空中の光速として $\square = -\frac{\partial^2}{c^2 \partial t^2} + \Delta$ である。

³この論文では虚数単位を i とする。

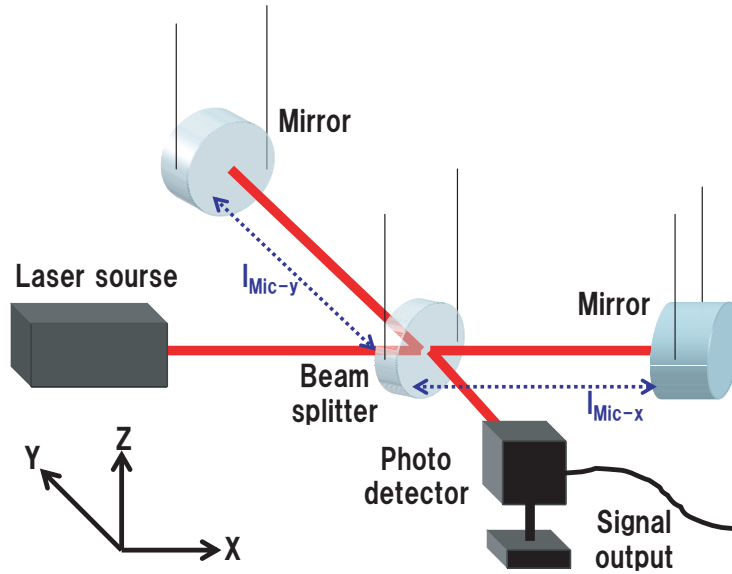


図 1.1: Michelson 干渉計の図。Beam Splitter とそれぞれの鏡は吊されている。ここでは図にあるように各座標軸を設定する。

1.2 レーザー干渉計

1.2.1 Michelson 干渉計

レーザー干渉計は図 1.1 のような Michelson 干渉計を基本として作られたものである。レーザー光源から出た光は、Beam Splitter によって直交する 2 つの方向 (図 1.1 では x 軸方向と y 軸方向) に分けられた後、それぞれ鏡によって反射され、Beam Splitter 上で再結合して干渉する。干渉計が更正する平面に対して垂直に重力波が入射すると、beam splitter から鏡までの固有距離が x 軸方向と y 軸方向とは逆相で変化するため、鏡からの反射光に位相差が生じ干渉縞が変化する。レーザー干渉計重力波検出器は、この干渉縞から重力波の信号を検出するものである。

Michelson 干渉計は 2 つに分けられた光が再結合するまでの位相差の変化を、出力光の強度変化として検出するものである。入射レーザー光の電場をレーザーの角周波数を Ω として、

$$E_{in} = E_0 e^{i\Omega t} \quad (1.10)$$

とする⁴。分けられた 2 つの光はそれぞれ ϕ_x 、 ϕ_y だけの位相変化をして再結合する。このとき、Photo Detector での電場 E_{out} は、

$$E_{out} = \frac{1}{2} E_0 e^{i(\Omega t - \phi_x)} - \frac{1}{2} E_0 e^{i(\Omega t - \phi_y)} \quad (1.11)$$

⁴ t は時間である。

となる。よって、Photo Detector での強度 P_{out} は、

$$\begin{aligned} P_{out} &= |E_{out}|^2 \\ &= \frac{1}{2}|E_0|^2(1 - \cos(\phi_x - \phi_y)) \\ &= \frac{1}{2}P_{in}(1 - \cos \phi_-) \end{aligned} \quad (1.12)$$

となる。ここでは $\phi_- = (\phi_x - \phi_y)$ とした。これより、各腕の位相変化が同相である時には Michelson 干渉計の出力は変化しないが、逆相に変化した時は出力の変化を生じさせる。

1.2.2 干渉計による重力波検出

図 1.1 の Michelson 干渉計に z 軸方向に進行する A_{xx} 偏光の重力波が入射し、4 次元線素が

$$ds^2 = -c^2 dt^2 + (1 + A_{xx})dx^2 + (1 - A_{xx})dy^2 + dz^2 \quad (1.13)$$

で与えられたとする。Beam splitter(BS) や鏡などは水平面内には自由質点にしてあるとすると、これらの座標は変化しないままである。光は $ds^2 = 0$ の道筋を通るので、 x 軸に沿って往復してきた光の位相は、 $\omega = \frac{2\pi c}{\lambda}$ と光の角周波数において、

$$\phi_x(t) = \Omega t_1 \quad (1.14)$$

$$\int_t^{t_1} \frac{dt'}{\sqrt{1 + A_{xx}(t')}} = \frac{2l_{\text{Mic-x}}}{c} \quad (1.15)$$

により与えられる。 A_{xx} の 1 次までで近似すると、

$$\phi_x(t) = \Omega \left(t - \frac{2l_{\text{Mic-x}}}{c} - \frac{1}{2} \int_{t - \frac{2l_{\text{Mic-x}}}{c}}^t A_{xx}(t') dt' \right) \quad (1.16)$$

である。 y 軸方向も同様に計算し、位相差を求めると、

$$\phi_- = \phi_x - \phi_y = -\frac{2\Omega(l_{\text{Mic-x}} - l_{\text{Mic-y}})}{c} - \Delta\phi_{\text{GR}}(t) \quad (1.17)$$

$$\Delta\phi_{\text{GR}}(t) = \Omega \int_{t - \frac{2l}{c}}^t A_{xx}(t') dt' \quad (l = l_{\text{Mic-x}} \simeq l_{\text{Mic-y}}) \quad (1.18)$$

となる。式 (1.17) の第 2 項が干渉計の重力波に対する応答となる。今、重力波を次のように

$$A_{xx}(t) = h_0 \sin(\omega t) \quad (1.19)$$

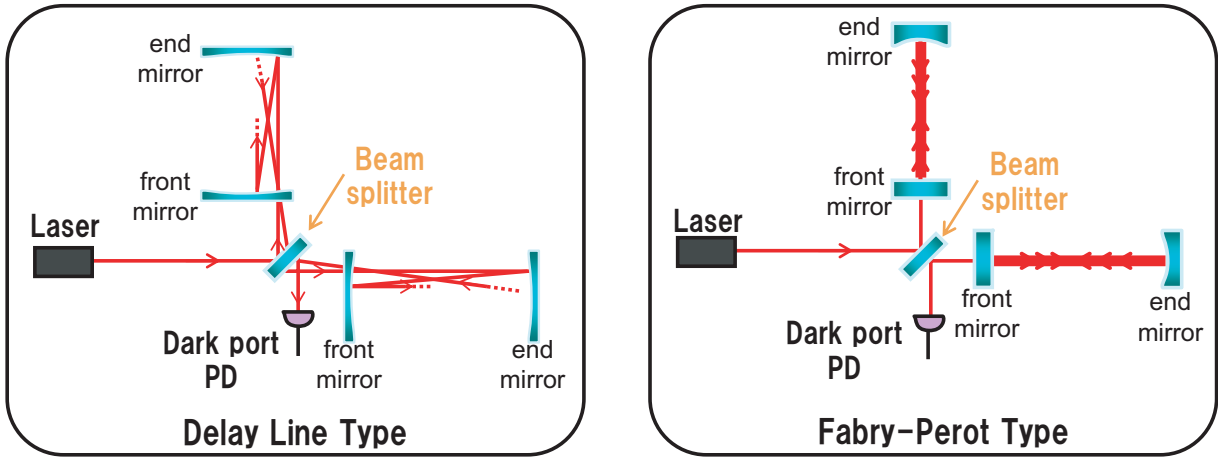


図 1.2: Delay-Line 方式 (左) と、Fabry-Perot 方式 (右) 干渉計のモデル。DL 方式の例としては、front mirror の一点に穴が空いており、光はそこから入射し、多重反射した後その穴から出ていく。FP 方式では透過型の鏡が用いられており、光は front mirror から入射し、共振器で多重干渉した後、front mirror を透過して出ていく。

と仮定して、式 (1.18) の位相変化を計算すると、

$$\Delta\phi_{GR} = 2h_0 \frac{\Omega}{\omega} \sin\left(\frac{\omega l}{c}\right) \cos\left(t - \frac{l}{c}\right) \quad (1.20)$$

となり、この応答の振幅は、

$$\phi_{GR} = 2h_0 \frac{\Omega}{\omega} \sin\left(\frac{\omega l}{c}\right) \quad (1.21)$$

で表される。この式は周波数 ω の重力波に対して

$$\frac{\omega l}{c} = \frac{\pi}{2} \quad (1.22)$$

のとき最大値を取ることになる。例えば、1kHz の重力波に対しては、 $l = 75\text{km}$ の時に感度が最大になる。しかし、実際作る事の出来る干渉計の大きさには限度があるので、光路長を稼ぐための手段として図 (1.2) のような鏡の多重反射を利用した Delay-Line 方式 (DL 方式) と多重干渉を利用した Fabry-Perot 方式 (FP 方式) が考えられている。現在、世界の大型レーザー干渉計では FP 方式が主流になってきている。

第2章 Fabry-Perot 共振器の性質

干渉計型重力波検出器において、構成する鏡は振り子状に吊られている。鏡を完全に宙に浮かせて自由質点とするのは不可能なため、地面振動により常に揺らされてしまうことになる。Fabry-Perot-Michelson (FPM) 干渉計では、Fabry-Perot 共振器内で光が共振した状態でそれぞれの腕共振器から出てきたレーザー光が、beam splitter を経て再び dark-fringe で干渉することでその性能を発揮する。よって、このような動作条件を満たすために干渉計の光路長について常に制御する必要がある。

この章では、

1. Fabry-Perot 共振器の応答。
2. 共振点付近での透過光量と反射光量。
3. 制御時の信号。

について述べる。ただし本論文において、光学系は全て理想的にアラインメント¹がされているものとする。また、この章では鏡が非常にゆっくりゆれているものとする。

2.1 Fabry-Perot 共振器の応答

Fabry-Perot 共振器とは図 2.1 にあるような高反射率の鏡を 2 枚対向させ、現実には地面振動等で鏡が揺らされているので、feedback をかけて共振させる。そうしてレーザー光を共振器内に閉じ込めることで、内部パワーを増幅させる光学系のことである。ここでは共振器長を L 、光源側の鏡の反射率と透過率を r_F 、 t_F 、向かい合う鏡を r_E 、 t_E とする。また、共振器の内側での反射率を正、外側を負とする²。また、こ

¹アラインメントとは対向する鏡の向きのことである。共振器の制御ではこれも大きな課題となってくるが、本論文では理想的であるとしてこのことには触れないでおく。

²Fabry-Perot 共振器に用いられるのは多層膜ミラーであり反射面がどこにあるか考える事により、反射率の表現に違いは出るが、ここでは反射面が誘電体多層膜の $\frac{\lambda}{4}$ (λ : レーザー波長) の境界面の一つの面であるとする。そうすることで、反射面の表と裏で反射率は同じ絶対値で正負が逆転したものになる。



図 2.1: Fabry-Perot 共振器の概略図

こでの反射率、透過率は振幅反射率、振幅透過率とする。計算の簡略化のために

$$R \equiv r_F r_E$$

$$T \equiv t_F^2$$

$$T' \equiv t_F t_E$$

と定義し、レーザーの角周波数から波数 k を $k = \Omega/c$ とする。光源からの単色平面波の式 (1.10) に対して、共振器の透過係数 t_{cav} 反射係数 r_{cav} はそれぞれ

$$t_{\text{cav}}(\Omega, L) = \frac{T'}{1 - R \exp(-2ikL)} \quad (2.1)$$

$$r_{\text{cav}}(\Omega, L) = -r_F + r_E T \frac{\exp(-2ikL)}{1 - R \exp(-2ikL)} \quad (2.2)$$

となる。透過光と反射光の強度 I_t 、 I_r は、 $I_0 = |E_0|^2$ として

$$I_t = |t_{\text{cav}}(\Omega, L)|^2 I_0 = \frac{T'^2}{1 + R^2 - 2R \cos 2kL} I_0 \quad (2.3)$$

$$I_r = |r_{\text{cav}}(\Omega, L)|^2 I_0 = \frac{r_F^2 + r_E^2 (T + r_F^2)^2 - 2R(T + r_F^2) \cos 2kL}{1 + R^2 - 2R \cos 2kL} I_0 \quad (2.4)$$

となる。TAMA300 での値等を表 2.1 を用いた時の透過光、反射光強度は図 2.2、2.3 のようになる³。横軸はある共振点からのずれを位相で表しており、縦軸は入射してきた光の強度を 1 として規格化し、鏡でのロスは無しと計算してある。

2.2 鏡の変位と光量

式 (2.3)、(2.4) の係数は k 、 L が共振条件

$$kL = n\pi (n : \text{正の整数}) \quad (2.5)$$

³光速 c は 2.99792458×10^8 [m/sec]。

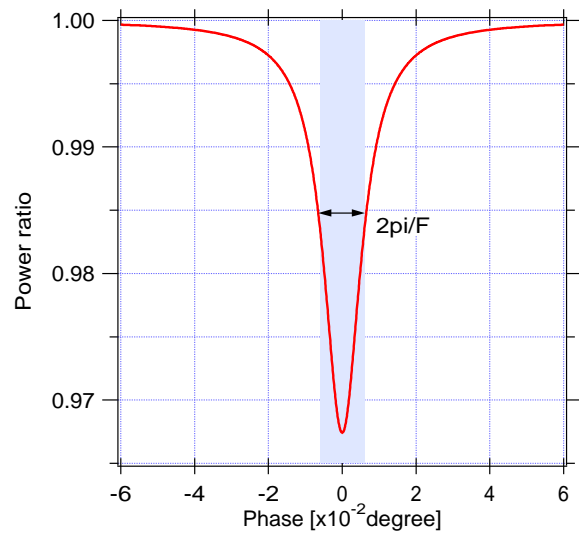
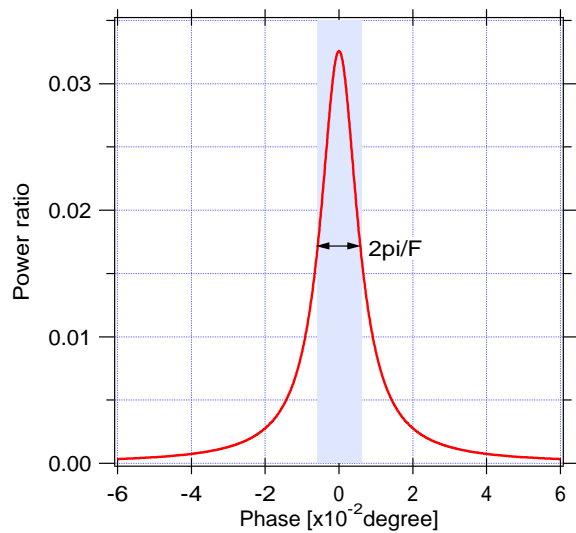


図 2.2: Fabry-Perot 共振器の透過光強度と変位 図 2.3: Fabry-Perot 共振器の反射光強度と変位

を満たす時にそれぞれ最大値、最小値を取る。Fabry-Perot 共振器の共振点は周期的に ($\Delta(kL) = \pi$ ごとに) 存在し、そのピーク間の間隔とピークの半値全幅 ($\Delta(kL)_{\text{FWHM}}$) の比はフィネス F と呼ばれて、

$$F \equiv \frac{\pi}{\Delta(kL)_{\text{FWHM}}} = \frac{\pi\sqrt{r_F r_E}}{1 - r_F r_E} \quad (2.6)$$

と表される。この式に表 2.1 の値を代入すると、516 という結果が得られる。また TAMA300 の Fabry-Perot 共振器においては実際に測定もされており、その値は 459 ± 19 という結果が得られてる。[5] この章の計算においては、表 2.1 の値および、 $F = 516$ という値を用いることにする。

名称	記号	数値
共振器長	L	300 m
front mirror 反射率	r_F^2	0.988
front mirror 透過率	t_F^2	0.012
end mirror 反射率	r_E^2	0.9999
end mirror 透過率	t_E^2	0.0001
レーザー波長	λ	1064 nm

表 2.1: 透過光強度、反射光強度を求めるための係数

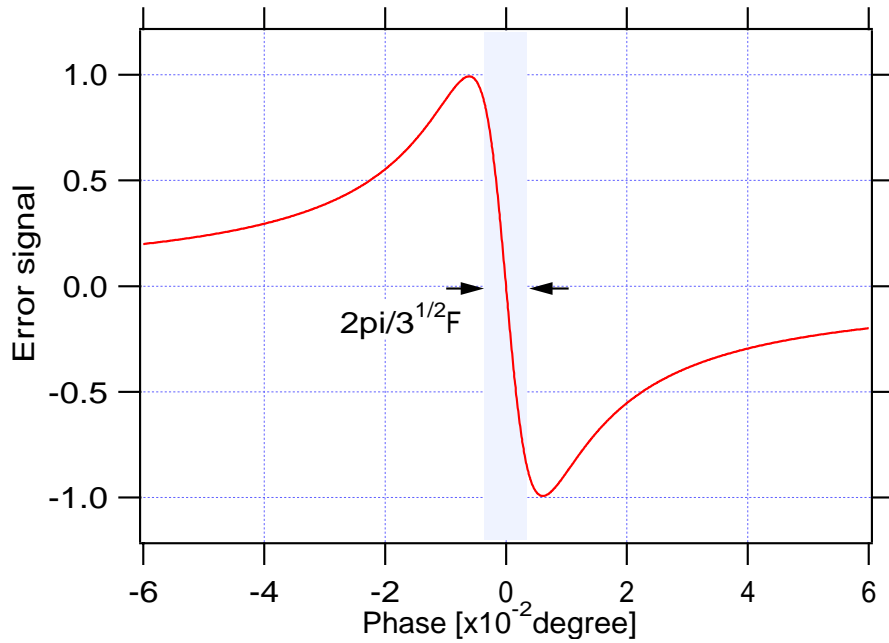


図 2.4: Fabry-Perot 共振器の反射光の error signal

式 (2.6) より共振条件付近で

$$\Delta(kL) < \frac{\pi}{F} \quad (2.7)$$

$$(2.8)$$

を満たす時、共振器は位相 kL に対して鋭敏な位相感度を持ち、そのことは図 2.2、図 2.3 から確認出来る。共振条件から外れてしまうと位相感度がほとんど無くなるので、Fabry-Perot 共振器を位相検出器として用いるには共振器長 L 、またはレーザーの位相を制御して、位相変化が悪くても

$$\Delta(kL) < \frac{\pi}{2F} \quad (2.9)$$

に抑えなくてはならない。今、レーザー周波数が一定だとすると、

$$\Delta L < \frac{\pi}{kF} = \frac{\lambda}{2F} \simeq 10^{-9}[\text{m}] \quad (2.10)$$

となる。この ΔL が共振器を位相検出器として用いる時の変位の許容範囲となる。

2.3 制御時の信号

変調法で共振器長を $kL = n\pi$ を満たすように制御する場合、実際には透過光や反射光から DC 成分を取り除いた成分

$$\frac{T'}{1 - R \exp(-2ikL)} E_0 \quad (2.11)$$

$$r_E T \frac{\exp(-2ikL)}{1 - R \exp(-2ikL)} E_0 \quad (2.12)$$

の虚部 t_{err} 、 r_{err} を求める事になる。それぞれ $E_0 = 1$ とすると、次のようになる⁴。

$$t_{\text{err}} = \Im \left[\left(\frac{T'}{1 - R \exp(-2ikL)} \right) \right] = \frac{T'(1 - R \cos(2kL))}{1 + R^2 - 2R \cos 2kL} \quad (2.13)$$

$$r_{\text{err}} = \Im \left[\left(\frac{r_E T \exp(-2ikL)}{1 - R \exp(-2ikL)} \right) \right] = -\frac{r_E T \sin(2kL)}{1 + R^2 - 2R \cos 2kL} \quad (2.14)$$

この時の反射光の r_{err} と変位は図 2.4 のようになる。横軸は kL による位相を表している。 r_{err} は error signal と呼び、その error signal と共振器長のずれが線形対応している幅は、 $r_E^2 = R \sim 1$ と考えた時、

$$\Delta(2kL) \sim \frac{\pi}{\sqrt{3F}} \quad (2.15)$$

となる。[4] 実際これは図 2.4 から確認できる。 k が一定である時は次のように書き換えられて、

$$\Delta L \sim \frac{\lambda}{2\sqrt{3F}} \quad (2.16)$$

今 TAMA300 で考えた場合、レーザーの周波数が安定であることに相当し、その時

$$\Delta L \sim 6 \times 10^{-10} = 0.6\text{nm} \quad (2.17)$$

となる。つまり、実際に Fabry-Perot 共振器を制御しようとする場合、鏡を 1nm にも満たない変位内に引き込まなければならず、そこに干渉計を制御する難しさがある。

⁴ここでは $\Im[Z]$ は Z の虚部を表す。

第3章 Fabry-Perot 共振器の制御

Fabry-Perot 共振器は、現実には鏡が地面振動などにより常に揺らされている。その為に Fabry-Perot 共振器からの信号をもとに feedback を返して制御する手法がよく取られている。共振状態を保つためには常に長さが一定になるように制御しなくてはならない。

この章では

1. Fabry-Perot 共振器の鏡の変位と速度。
2. feedback 制御に用いるコイル・アクチュエーターについて。
3. error signal と lock acquisition。
4. 現実の Fabry-Perot 共振器で起きる複雑な現象。

について述べてゆく。

3.1 鏡の変位と速度

ある系を特定の領域に閉じこめようとする場合、その特定の領域からの誤差が無限に線形であれば、その誤差分を feedback で返せばいずれその領域まで引き込む事が出来る。

しかし、レーザー干渉計において、測定できるその誤差の error signal は図 2.4 のように、狭い変位の間でだけ線形な領域になっている。

制御有効範囲に引き込むためには以下のようないくつかの条件を考えなくてはならない。

- 鏡の動く変位
- 鏡に装着されたアクチュエーター、コイルのレンジ
- feedback を返すまでにかかる時間
- feedback をかけてからコイルに目的の電圧がかかるまでの過渡現象

これらを順に考え、この節ではまず鏡について考察する。

3.1.1 鏡の変位

レーザー干渉計型重力波検出器は、吊された鏡の距離変化を精密に測定する事で重力波の影響を検出しようとする。一方で、鏡は吊されているといえども地面と完全に分離されているわけではないので、地面振動によって揺れている。地面振動のような外乱に起因する振動は重力波検出器のSN比 (signal-to-noise-ratio) を悪くし、共振点への引き込みの妨げにもなる。したがって、鏡には吊すこと以外にも防振が行われているが、それでも完全に防振する事は不可能である。

ここでは実際の地面振動と鏡を吊す懸架系の防振特性から揺れの振幅の見積を行っていく。

地面振動の振幅

地面振動には地震のような突発的な大振幅の振動のほかにも、定常的な微小振幅の振動が存在する。このような定常的な振動は常微動と呼ばれている。干渉計の定常的な性能を与えるのはこの常微動である。以降とくに断らない限り、地面振動といえば常微動を示すものとする。

地面振動の典型的な振幅をパワースペクトル密度 $y_{\text{seis}}(f)$ で表すと f の単位を Hz として、

$$y_{\text{seis}}(f) = \left(\frac{1\text{Hz}}{f}\right)^2 \times 10^{-7} \text{ [m}/\sqrt{\text{Hz}}] \quad (3.1)$$

という形の f^2 に反比例するということが広い周波数範囲にわたって成立する事が知られている [6][7][8]。

次に低周波まで含めた全周波数帯での RMS(Root Mean Square) 振幅を見積もってみる。周波数が 0.1Hz 以上の領域では逆 2 乗則が良く成り立っているのが知られている。(参照：図 3.1) 0.1Hz 以下の周波数帯域での RMS 振幅は 10^{-5} で一定であるものとして考える。地面振動の RMS 振幅 y_{RMS} を求めるために式 (3.1) の 2 乗を全周波数帯域で積分してその平方根を取ると、

$$\begin{aligned} y_{\text{RMS}} &= \left[\int_0^{\infty} |y_{\text{seis}}(f)|^2 df \right]^{1/2} \\ &= \left[\int_0^{0.1} |10^{-5}|^2 df + \int_0^{\infty} \left| \frac{10^{-7}}{f^2} \right|^2 df \right]^{1/2} \\ &\sim 4\mu\text{m} \end{aligned} \quad (3.2)$$

となる。つまり地面は少なくとも数 μm は常に振動している事になる。また、式 (3.1) から推測されるように、低い周波数にパワーが集中している。

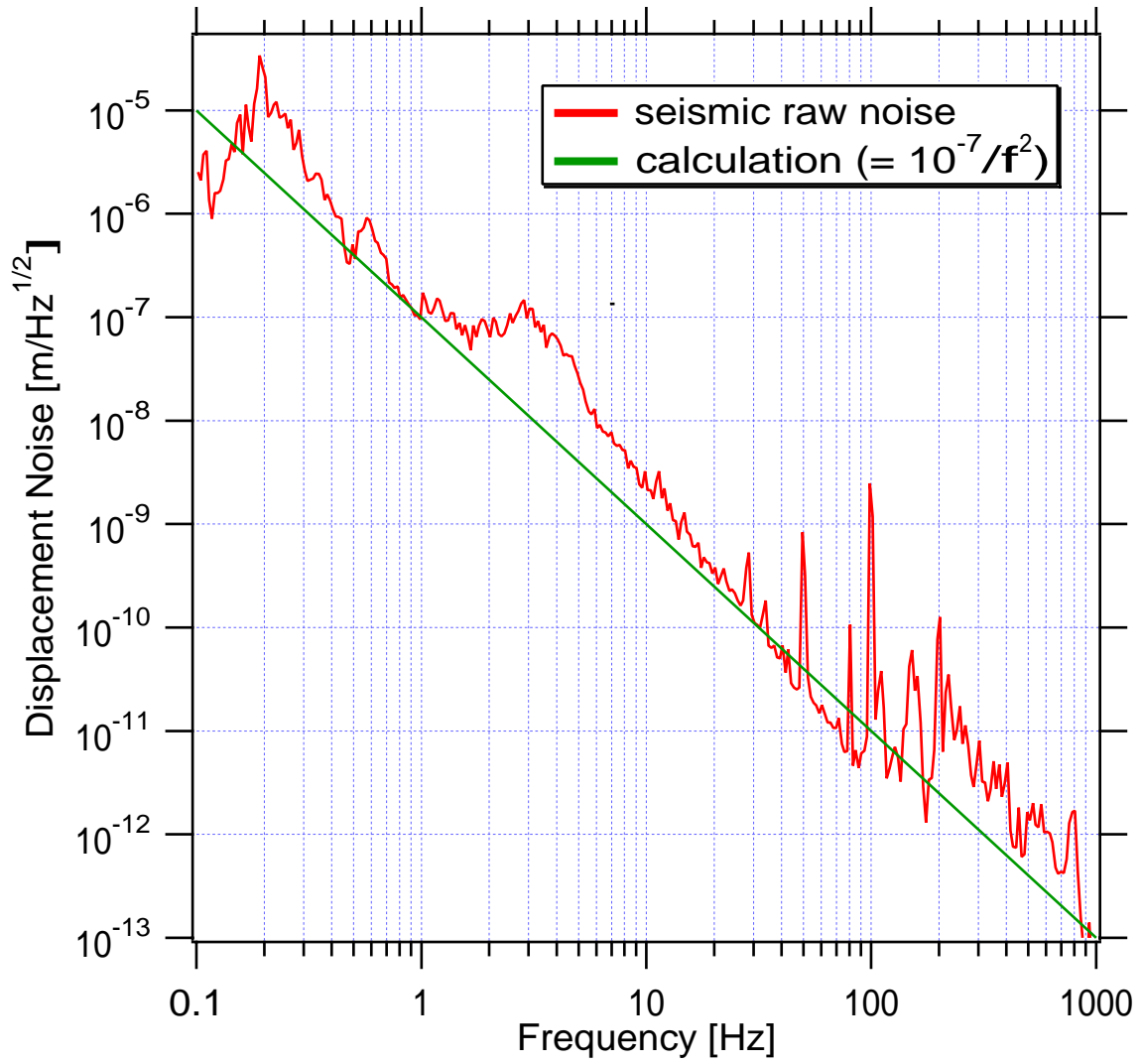


図 3.1: 国立天文台三鷹キャンパス地面振動変位スペクトル。直線はこの論文で用いている式 (3.1) 地面振動のモデル。[9]

懸架系防振比

地面振動を抑えるために、干渉計の鏡には防振が施されている。地面振動による鏡の振幅 $x_{\text{seis}}(f)$ は地面振動の振幅 $y_{\text{seis}}(f)$ と線形な関係にあるものとする、

$$x_{\text{seis}}(f) = H(f)y_{\text{seis}}(f) \quad (3.3)$$

このように書き表せる。鏡の振幅と地面振動の振幅の比 $H(f)$ を地面振動に対する鏡の懸架システムの伝達関数と呼ぶ。また、伝達関数の絶対値 $|H(f)|$ を懸架システムの防振比と呼ぶ。実際の TAMA300 の防振比のパワースペクトラムは図 3.2 のようになる。これより懸架系の特性から 1Hz あたりに bump があるが、そこから 20Hz にかけて減衰率は大きくなり、20Hz を超えたところでは -80dB 以上の減衰を得られている。簡単のためにこの鏡での減衰率を図から

$$H(f) = \begin{cases} 1 & (f \leq 2\text{Hz}) \\ 1/(\frac{f}{2})^4 & (2\text{Hz} < f \leq 20\text{Hz}) \\ 10^{-4} & (20\text{Hz} \leq f) \end{cases} \quad (3.4)$$

のようにおく。したがって $x_{\text{seis}}(f)$ は

$$x_{\text{seis}}(f) = \begin{cases} 10^{-5} & (f \leq 0.1\text{Hz}) \\ 10^{-7}/f^2 & (0.1\text{Hz} < f \leq 2\text{Hz}) \\ 2^4 \cdot 10^{-7}/f^6 & (2\text{Hz} < f \leq 20\text{Hz}) \\ 10^{-11}/f^2 & (20\text{Hz} \leq f) \end{cases} \quad (3.5)$$

のように表されて、鏡の RMS 振幅を求めるために式 (3.5) の 2 乗を全周波数帯域で積分してその平方根を取ると、

$$x_{\text{RMS}} = \left[\int_0^{\infty} |x_{\text{seis}}(f)|^2 df \right]^{1/2} \\ \sim 4\mu\text{m} \quad (3.6)$$

となる。高い周波数では防振が効いているので、一度干渉計が共振点に引き込まれると、感度を測定する上では要求を満たしている。しかし、 x_{RMS} と y_{RMS} の差はほとんど無く、地面振動から来る低周波の大きなパワーで鏡の RMS 振幅がほとんど決まってしまうことがわかる。これより、共振器の鏡は lock するまでは数 μm の変位で揺れていると考えられる。

3.1.2 鏡の速度

鏡が実際に揺れている速度は、日によってや時間帯によっても異なってくる。鏡の速度を常時測定するシステムが確立していないので、統計的に述べることは出来ない

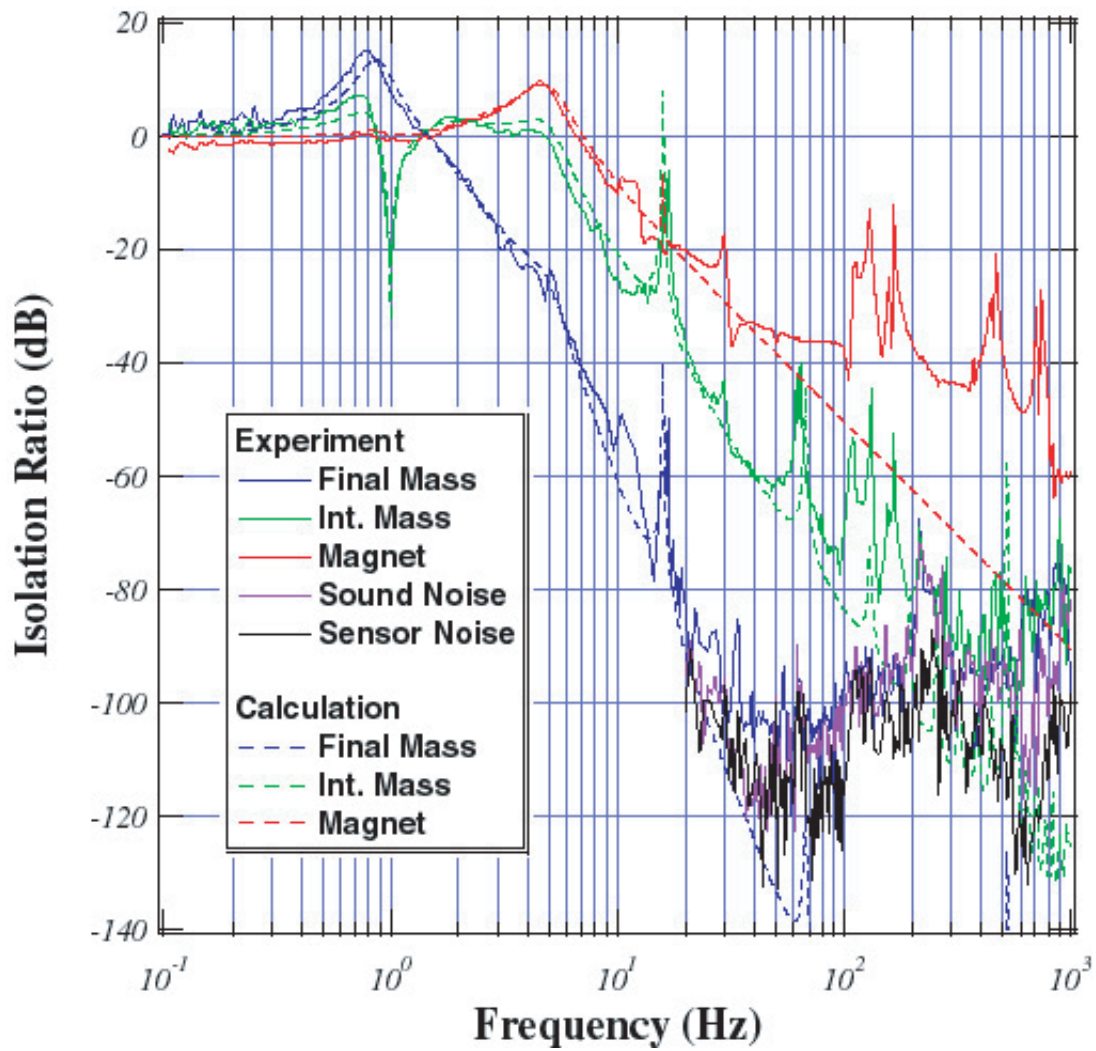


図 3.2: TAMA300 懸架系の防振比。[9] このグラフの Final Mass にあたる青色の線が鏡の防振比になる。

記号	名称	値
C_c	コイルアクチュエーターのカップリング	$4.9\text{kg m /s}^2\text{A}$
V_c	コイルにかかる電圧	10 V
R_c	コイルアクチュエーターのインピーダンス	50Ω
m	鏡の重さ	1kg

表 3.1: コイルアクチュエータおよび鏡のパラメータ

が、経験上、lock acquisition 可能な変位を透過光が出てくる時間で割ることにより、等速で運動していた場合の速度を求めることが出来る。この最大速度が数 $\mu\text{m}/\text{sec}$ となる。平日の日中などは人間の活動も激しく早く揺れ、平日でも深夜また年末年始ではその揺れ方もゆっくりとなる。

3.2 コイルとアクチュエーター

このように揺れる鏡を TAMA300 では feedback により制御している。ここではその feedback 制御で用いられているコイルとアクチュエーターに関連したことを述べていく。

3.2.1 アクチュエーター、コイルのレンジ

TAMA300 で実際に鏡に使われているアクチュエーター、コイルから鏡に加えられる力積を考える。

まず一般に、コイル・アクチュエーターの力を F_{coil} 、lock acquisition にかける時間を Δt 、鏡の質量を m 、鏡の速度を v_m とすると、以下の条件を

$$F_{\text{coil}}\Delta t > mv_m \quad (3.7)$$

を満たす時に鏡を lock することが出来る。ここで実際に TAMA300 で使われている表 3.1 のパラメータを用いて計算していく。鏡には4つのコイルが取り付けられており、それぞれ C_c のカップリングを持っている。そのコイルには最大 10V の電圧がかかるようになっており、回路のインピーダンスは 50Ω であるので、鏡に加えられる最大の力積は

$$\begin{aligned} F_{\text{coil}}\Delta t &= 4 \times 0.5 \times 9.8 \times 10^{-3} \times \frac{10}{50} \times \Delta t \\ &\simeq 4 \times 10^{-3} \times \Delta t \end{aligned} \quad (3.8)$$

となる。これより lock acquisition により鏡を共振点に引き込める最大速度との関係を

求めることが出来て、式 (3.7) より、

$$\begin{aligned} 4 \times 10^{-3} \times \Delta t &> 1 \times v_m \\ \therefore v_m &< 4 \times 10^{-3} \times \Delta t \end{aligned} \quad (3.9)$$

有効な error signal が出てくるのは 1msec 程度なので、鏡が $4\mu\text{m}/\text{sec}$ 以上の速さで揺れていたら、捕らえる事が出来ないということになる。また今回の計算は error signal を捕らえたら、一瞬でコイルに最大電圧がかかると仮定しているうえに、実際は feedback を返すまでに遅れがあったりするのでより厳しくなる。

3.2.2 feedback のゲインと、返す時間

error signal を検知してから feedback を返すまでの時間について考えてみる。まず信号がケーブルを伝わる速さはケーブルの品質により異なるが、おおよそ光の速さの $2/3$ 程度である。TAMA300 の場合、長く見積もっても feedback 制御のケーブル総延長 l_{cable} が 100m を超えることはないので、その 100m で feedback を返すまでの時間 t_{feed} を計算すると、

$$t_{\text{feed}} = \frac{l_{\text{cable}}}{\frac{2}{3}c} \simeq 0.5 \quad [\mu\text{sec}] \quad (3.10)$$

であるから信号が伝わるまでの時間自体は lock acquisition のタイムスケールが 1msec 程度なので問題ではない。

鏡を lock する時、まずは共振ポイントを通り過ぎようとする鏡の error signal を捕らえた瞬間に feedback のゲインを高くして、その瞬間に全力で引き戻せばいい。ただ、問題は feedback のゲインを高くしてしまうと、いったん共振ポイントに引き込んだ後はより細かい制御で最も共振する状態に持って行かなくてはならないところにある。feedback が強すぎると共振ポイントに引き込めても、今度は feedback に揺らされてなかなか安定しなかったり、もともと鏡がゆっくり共振ポイントを通り過ぎようとしていたときは、逆に鏡がはじかれてしまう。

feedback のゲインはこれらのことを考慮して高すぎず低すぎない設定をしなければならない。

3.2.3 コイルでの過渡現象

干渉計の lock acquisition にかかる時間が 1msec 前後という事で、電氣的にも過渡応答を考える必要が出てくる。ここでは簡単なモデルとして図 3.3 のようなものを考える。ここでのインダクタンス、抵抗値は TAMA300 で実際に使われているコイルの値を参考にしてそれぞれ、 $L = 31[\mu\text{H}]$ 、 $R = 1.8\Omega$ とする。[12] ここで回路の起電力

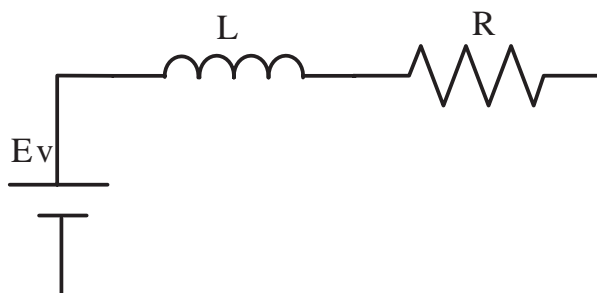


図 3.3: コイルの過渡現象計算モデル

E_V を 10V、流れる電流を $I_{ec}(t)$ として、回路方程式を立てると

$$L \frac{dI_{ec}(t)}{dt} + RI_{ec}(t) = E_V \quad (3.11)$$

のようになる。これより回路に流れる電流の時間を変数とした解が求まる。抵抗値、起電力が一定であるから、コイルにかかる力は電流に比例する。定常に電圧がかかる時を 1 に規格化した時の過渡現象の様子は図 3.4 のようになる。

この過渡現象の計算から、90%の電圧がかかるまでには 0.04msec ほどかかることになってしまう。図 3.4 の規格化した関数を $y(t)$ 、有効な error signal の時間を T_l として、過渡現象が無い時に対して、過渡現象を含めた場合コイルにかかる事の変位の割合 D_x を求めると、

$$D_x = \int_0^{T_l - T_{fb}} y(t) dt \quad (3.12)$$

となる。今、 $T_{fb} = 0$ とした時に有効な error signal の時間と、lock acquisition により加えられる変位の関係は表 3.2 のようになった。この結果より、実際には T_{fb} による遅延もあり、積分時間が短くなるため、過渡現象による引き込み変位の減少を考慮しなくてはならない場合も出てくると言える。

$T_l - T_{fb}$ [msec]	過渡現象がある時の変位 [m]	過渡現象が無い時の変位 [m]
1	0.983	1
0.5	0.966	1
0.1	0.828	1

表 3.2: 過渡現象による引き込み変位の減少率

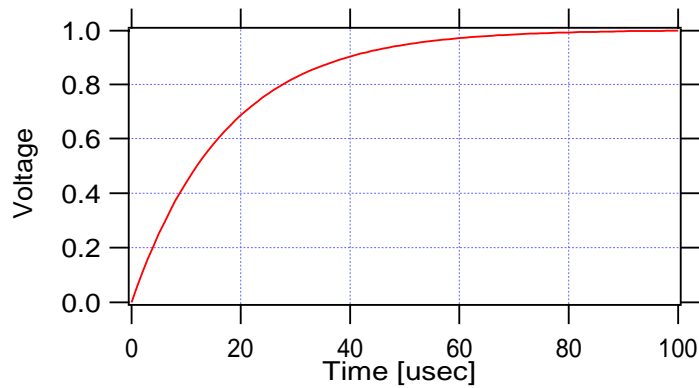


図 3.4: コイルの過渡現象計算モデル

3.3 lock acquisition

3.3.1 lock acquisition とは

これまでの章で計算してきたように、Fabry-Perot 共振器を共振状態を保つためには鏡を $\Delta L \sim 0.6\text{nm}$ に保たなければならない。一方で鏡は日にもよるが、およそ 10^4nm のオーダーでゆれており、制御の第一段階としてまず、その鏡を $\Delta L \sim 0.6\text{nm}$ の変位内の領域に引き込まなければならない。これを lock acquisition という。共振点付近で制御系が反応できる変位を増幅させる事が出来れば、それだけ lock しやすくなる。そこで error signal に演算を加える事で、反応できる変位を増幅出来るのでその事について説明していく。

3.4 Fabry-Perot 共振器を lock する手段

Fabry-Perot 共振器を lock させるには、TAMA300 の場合 feedback を共振器の鏡に返す方法と、レーザー系に返す方法がある。feedback により Fabry-Perot 共振器や、モードクリーナーの長さを制御することで光の周波数安定化を行う。この時制御すべき長さを波長で割った数だけ波の個数が存在する。つまり TAMA300 その波数比は

$$\text{Fabry - Perot 共振器} : \text{モードクリーナー} = 1 : 30$$

である。光の周波数変化する、もしくは変化させると波数に比例して位相が変化することになる。よって Fabry-Perot 共振器で制御する場合、モードクリーナーに対して 30 倍揺れていることになる。それだけ feedback でかける力が必要になってくる。

3.4.1 error signalの広帯域化

3.2節で考えてきたように、いかにlock acquisitionを成功させるかはlock acquisitionの時にかけることの出来る変位に依存している。加える変位を増やせる方法としては以下のような可能性が考えられる。

遅延 T_b の減少

サーボのフィルタを設計し直す事でUGF¹を上げる事が出来るが、その為にはフィルタのゲインマージンや位相余裕、懸架系の共振周波数、制御帯域など考慮する必要のある要素が多く、今回は既存のフィルタと同じ物を用いる。

コイル・アクチュエータ力 F_{act} の強化

短い時間でも、コイル・アクチュエータの力を強化する事で大きな力積を与える事が出来るが、鏡の変位として重力波の信号が現れてくるために、もともとコイル・アクチュエータ系の電気ノイズが問題となることもありえるので、干渉計感度の点からいけばむしろこの力を落とさなくてはならない。

error signalの演算

フィルタの設計をしないしたり、コイル・アクチュエータを交換したりするのではなく、装置自体は同じものを使い、出てくる信号の演算をすることによって広帯域化を目指す。

ここでもう一度、 $r_F = r_E$ の時のerror signal E_{er} と Fabry-Perot 共振器の透過光量 I_t を見直してみると、

$$E_{er} = -\frac{r_E T \sin(2kL)}{1 + R^2 - 2R \cos 2kL} \quad (3.13)$$

$$I_t = \frac{T'^2}{1 + R^2 - 2R \cos 2kL} I_0 \quad (3.14)$$

となっている。ここで、error signalを透過光量で規格化すると、

$$\frac{E_{er}}{I_t} = -\frac{r_E T \sin(2kL)}{T'^2 I_0} \quad (3.15)$$

となり \sin の1次の関数となる。そうすることでerror signalの傾きが緩くなり、線形域が広がることが期待できる。[13]

実際に図 2.2の透過光量で図 2.4のerror signalを規格化したものが図 3.5のようになる。もともとの透過光量が少なかったため、規格化すると規格化すると値も増幅さ

¹後述の 6.3.2 参照。

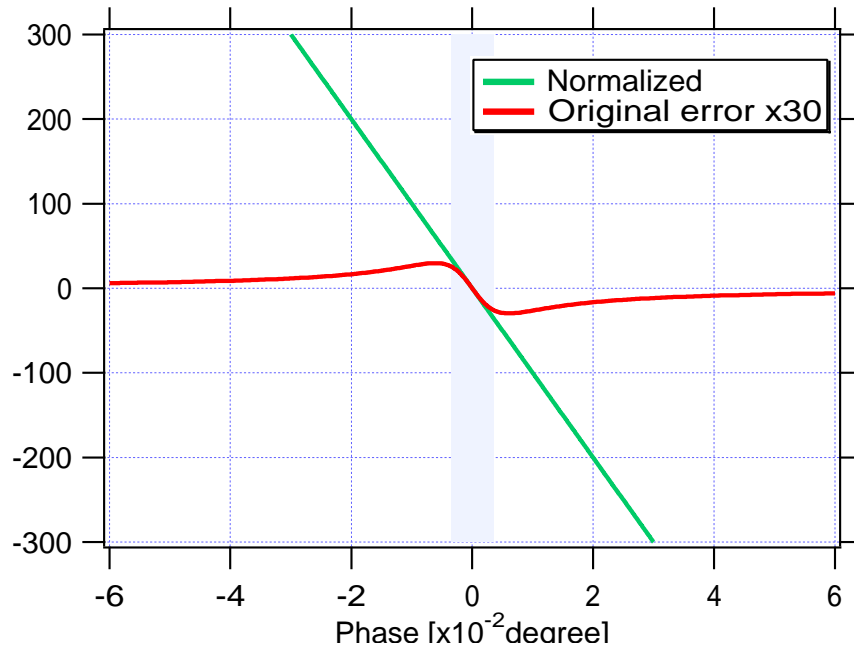


図 3.5: 透過光量により規格化された error signal

れるので、比較するためにオリジナルの error signal は 30 倍して書いてある。この図より、確かに線形域が広がっている事が確認できる。それにより鏡に加える事の出来る力積も増加し、lock acquisition を成功しやすくなるということになる。

3.5 現実の Fabry-Perot 共振器で起きる複雑な現象

ここでは Fabry-Perot 共振器を lock しようとした時に出てくる、他の現象について述べる。

3.5.1 共振器内のドップラー効果

これまで共振器内の透過光量や error signal を描いてきたが、これは鏡が非常にゆっくりと揺れているという仮定のもとドップラー効果を考慮に入れていないものである。

しかし、実際は鏡がある程度の速度を超えて揺れだすと、ドップラー効果が無視できないような状況になってくる。[5]

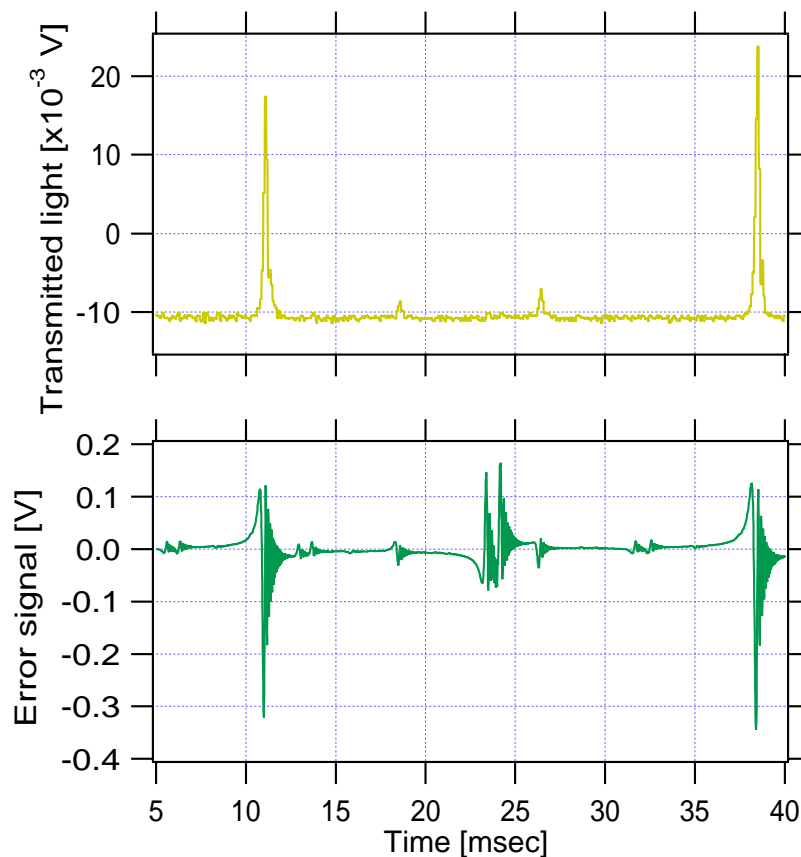


図 3.6: ある日の TAMA300 における透過光量と error signal。

3.5.2 規格化による弊害

規格化する事で error signal の線形域を増やせることはわかったが、透過光量が少ない時ほどシグナルが増幅される事で lock acquisition に関係のないシグナルが増幅されてしまう。その error signal にはサイドバンドや、発生源はわからないが混入してしまったシグナルが含まれている。

ある日の TAMA300 における透過光量と error signal は図 3.6 のようになる。11msec と 38msec あたりにあるのが共振状態を示す error signal であり、23~25msec あたりにあるのがサイドバンドによるシグナルである。これらのシグナルにはドップラー効果による影響もみることができる。透過光量を併せてみるとわかるように、共振状態に無い時はほとんど透過光量もないことがわかる。透過光量で規格化する際はオフセットを打ち消すようにこの場合、実際には $12 \times 10^{-3} \text{V}$ ほど加算して規格化する事になる。

不要なシグナルに反応してしまわないために、この透過光量を使ったトリガーを設定する。そうすることで、lock acquisition に関係ないシグナルが来た時に間違っ

feedback を返す事も無くなる。また、規格化無し時には反応してしまっていたサイドバンドに対しても反応しにくくなる。

3.6 Fabry-Perot 共振器の制御

Fabry-Perot 共振器を lock するために考慮する内容として、

鏡の変位 防振系を用いても低周波のパワーが大きく数 μm の変位で動いている。

feedback 計算上、鏡が $4\mu\text{m}/\text{sec}$ 以下の速度であれば lock 出来るが、ゲインや応答までの時間を考慮するとさらにゆっくりでないと lock できない。

error signal 透過光量で規格化することにより線形な領域が増える。

lock する手段 Fabry-Perot 共振器の鏡を用いて lock する時、TAMA300 の場合モードクリーナーに比べて 30 倍の変位と速さを制御しなくてはならない。。

透過光量による規格化 トリガーをしないと共振以外の信号も増幅させてしまう。

これらのことに注意しながら Fabry-Perot 共振器の制御を考えていかなければならない。

第4章 デジタル信号処理

大型の干渉計である TAMA300 を建設した当初から比べると技術進歩が進み、当時では実現が厳しかった速度でのデジタル処理が可能となってきた。今回、そうした高速処理が可能となったデジタルシステムを導入するのだが、まずはその基本となるデジタル信号処理についてこの章では、

1. デジタルとアナログについて。
2. アナログからデジタルへの変換に必要な演算法のまとめ。[14]
3. デジタルフィルタの設計法。[14, 15]
4. デジタルシステムの位相遅延。[16]

について調べていく。

4.1 デジタルとアナログ

アナログが連続時間システム (continuous time system) を扱うのことに對して、デジタルでは離散時間システム (discrete time system) を扱っている。アナログとデジタルは相互に変換できるが、その際に ADC(analog-to-digital combert) や DAC(digital-to-analog combert) による量子化、またサンプリング定理¹を満たしていないとならないという問題があり、アナログと同じように扱うことはできない。しかし、ADC や DAC の組み方や、サンプリング周波数を適切に設定することで、問題なく変換できるようになる。

一方で、デジタルで扱う離散時間信号は、フーリエ変換、逆フーリエ変換を使って周波数領域の信号に変換する事が出来る。離散時間システムでは、連続時間システムではなしえない周波数領域でのさまざまな処理が出来る事が特徴であり利点である。また、アナログでは複雑で作るのが難しいようなフィルタも、デジタルでは楽に作れてしまうことも多いというのはメリットである。

¹サンプリング定理とは、サンプリング周波数が信号の帯域幅の2倍以上でなくてはならない、というもの

4.2 演算法のまとめ

ここではアナログフィルタからデジタルフィルタに変換する際に必要な演算を復習しておく。

4.2.1 z 変換

連続時間システム的设计・解析にはラプラス変換が用いられているが、離散時間システムについては z 変換が用いられる。この z 変換の定義を示しておく。

離散時間信号が $x(n), n = 0, 1, \dots$ と与えられたとすると、複素変数 z を用いて次式のように定義される関数 $X(z)$ のことを $x(n)$ の両側 z 変換と呼ぶ。

$$X(z) = \sum x(n) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (4.1)$$

通常用いられている信号は $n < 0$ の場合は零であるので、

$$X(z) = \sum x(n) = \sum_{n=0}^{\infty} x(n)z^{-n} \quad (4.2)$$

となる片側 z 変換を用いて、この論文では式(4.2)で定義される $X(z)$ を $x(n)$ の z 変換と呼ぶ。

またデジタル演算において z^{-1} は離散時間信号を1サンプル遅らせる演算子と考える事が出来る。これはデジタルフィルタを設計する上で重要な概念となる。

4.2.2 Fourier 変換

時間 t の関数で与えられている信号をフーリエ変換すると、その信号に含まれている周波数成分を求める事が出来る。これを一般にスペクトル解析と呼ぶ。

離散時間信号 $x(n)$ の離散時間フーリエ変換 $X(\omega)$ は次式のように定義される。

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-i\omega n} \quad (4.3)$$

また、離散時間フーリエ逆変換は次のように定義される。

$$x(n) = \frac{1}{2\pi} \int_{\pi}^{-\pi} X(\omega)e^{i\omega n} d\omega \quad (4.4)$$

式(4.3)より $X(\omega)$ が連続変数 ω の関数であることが、式(4.4)より $X(\omega)$ が周期 2π の関数であることがわかる。

また両側 z 変換の式 (4.1) で、 $z = e^{i\omega}$ とおくと、

$$\begin{aligned} X(z)|_{z=e^{i\omega}} &= \sum_{n=-\infty}^{\infty} x(n)z^{-n}|_{z=e^{i\omega}} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{in\omega} \\ &= X(\omega) \end{aligned} \tag{4.5}$$

となり、フーリエ変換に等しくなる。

4.3 IIR フィルタの設計

デジタルフィルタをインパルス応答で次の2種類に分類することが出来る。インパルス応答とは、1個のパルスをデジタル信号システムに入力したときの出力のことである。

IIR フィルタ infinite impulse response filter のことで、インパルス応答が無限に続き得るフィルタ

FIR フィルタ finite impulse response filter のことで、インパルス応答がある時間になれば必ず零に収束するフィルタ

演算という点から、同じ程度の特性のフィルタを実現する場合、FIR フィルタはIIR フィルタに比べて演算誤差の影響も小さく、理論的には必ず安定になるので、比較的扱いやすいフィルタだと言える。ここで言うフィルタが安定とは、有限の振幅の入力信号に対して、出力信号の振幅も有限になることをいう。不安定であると、出力の振幅が無限大になる。実際には扱う事の出来るデータ範囲は限られているので、無限大でなく、オーバーフローが発生することになる。

しかし、FIR フィルタはIIR フィルタに比べて、必要な計算量が1桁多くなってしまふ。今回は演算速度も要求されるフィルタの設計となるので、IIR フィルタで設計を行っていくことにする。

4.3.1 S-Z 変換

IIR フィルタの設計法は、アナログフィルタの伝達関数を z 変換した式に変換する方法が一般的で、 s - z 変換法と呼ばれている。アナログフィルタの伝達関数はインパルス応答をラプラス変換したもので、例えばインパルス応答を $h(t)$ とすると、アナログフィルタの伝達関数 $H(s)$ は

$$H(s) = \int_0^{\infty} h(t)e^{-st} dt$$

で与えられる。これに対応する離散時間信号の表現が z 変換である。ここではその s - z 変換法の1つである、双1次 z 変換法について示していく。

双1次変換法の原理は、 s 領域での積分を z 領域での積分に近似することによっている。連続信号 $s(t)$ とこれを1階積分した $x(t)$ を考える。このとき、 $x(t)$ を時間間隔 T でサンプリングした $x[n]$ は

$$\begin{aligned} x[n] &= x(nT) \\ &= \int_{(n-1)T}^{nT} s(t)dt + x((n-1)T) \end{aligned} \quad (4.6)$$

となる。この式の積分に対して台形公式²を適用すると、時間 T が微小区間であれば良好な近似値が得られる。したがって、

$$\begin{aligned} x[n] &= \frac{T(s[n] + s[n-1])}{2} + x[n-1] \\ \therefore x[n] - x[n-1] &= \frac{T}{2}(s[n] + s[n-1]) \end{aligned} \quad (4.7)$$

となる。ここで $x[n]$ 、 $s[n]$ を

$$X(z) = \sum x[n] \quad , \quad S(z) = \sum s[n]$$

として z 平面に変換して、この式の両辺を z 変換すると

$$\begin{aligned} X(z)(1 - z^{-1}) &= \frac{T}{2}S(z)(1 + z^{-1}) \\ \therefore \frac{X(z)}{S(z)} &= \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \end{aligned} \quad (4.8)$$

が得られる。式(4.8)左辺は式(4.6)より、 $S(z)$ を1階積分して $X(z)$ を得るための伝達関数と考えることが出来る。積分を s により表すと $1/s$ なので、

$$\begin{aligned} \frac{1}{s} &= \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \\ \therefore s &= \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \end{aligned} \quad (4.9)$$

が得られる。これが双1次 z 変換における z と s の関係式になる。

まずここでは、双1次 z 変換の性質を考えていく。式(4.9)に、

$$s = i\omega_s \quad (\omega_s : s \text{ 領域における角周波数 [rad/sec]}) \quad (4.10)$$

$$z = \exp(p \cdot T) \quad (4.11)$$

$$p = i\omega_z \quad (\omega_z : z \text{ 領域における角周波数 [rad/sec]}) \quad (4.12)$$

²非積分関数 $y=f(x)$ を区間 (a,b) で積分するとき、台形公式とはその区間での積分を両端の値 $f(a)$, $f(b)$ を用いて、台形の面積を求める要領で積分値 S を計算する方法である。区間の幅を $h (=b-a)$ とすれば、面積 S は $S = (f(a)+f(b)) \cdot h / 2$ で求めることが出来る。

を代入すると、

$$\begin{aligned}
 i\omega_s &= \frac{2}{T} \frac{1 - \exp(-p \cdot T)}{1 + \exp(-p \cdot T)} \\
 &= \frac{2i}{T} \tanh\left(\frac{p \cdot T}{2}\right) \\
 &= \frac{2i}{T} \tanh\left(\frac{i\omega_z \cdot T}{2}\right) \\
 \therefore \omega_s &= \frac{2}{T} \tan\left(\frac{\omega_z \cdot T}{2}\right)
 \end{aligned} \tag{4.13}$$

となり、 s 領域での角周波数 ω_s と z 領域での角周波数 ω_z との関係が得られる。

式 (4.13) から ω_s と ω_z との関係が非線形になることがわかる。このため双 1 次 z 変換では、所望の角周波数 ω_z が決まったとき、まず式 (4.13) により ω_s を求め、この ω_s についてアナログフィルタを設計する必要がある。

つぎに式 (4.9) を用いて、アナログフィルタの伝達関数をデジタルフィルタの伝達関数に変換する方法を示していく。アナログフィルタの伝達関数 $H(s)$ を次式のように極、零点および系の利得に分解する³。

$$H(s) = \frac{g \cdot \prod_{a=0}^{m_{\text{zero}}-1} s - \Re[\mathcal{Z}(a)] + i\Im[\mathcal{Z}(a)]}{\prod_{b=0}^{m_{\text{pole}}-1} s - \Re[\mathcal{P}(b)] + i\Im[\mathcal{P}(b)]} \tag{4.14}$$

(

g	系の利得
$\mathcal{Z}(a)$	$a = 0, 1, \dots, m_{\text{zero}} - 1$: 伝達関数の零点
$\mathcal{P}(b)$	$b = 0, 1, \dots, m_{\text{pole}} - 1$: 伝達関数の極
$\mathcal{Z}(\cdot), \mathcal{P}(\cdot)$	実根または共役対を持つ複素根

)

その上で、極および零点を 1 つずつ双 1 次 z 変換することでデジタルフィルタの伝達関数が得られる。

4.3.2 直接型

ここでは IIR フィルタを具体的に実行する方法を書いていく。フィルタの構成に必要な要素は乗算器 (multiplier)、加算器 (adder)、単位遅延素子 (unit delay element) の 3 種類である。デジタルフィルタはこれらの組み合わせにより、直接型、縦続型、並列型などに分類される。

このようにフィルタの構成法が分類されているのには、計算機側の問題で、以下の条件のどれかが欠ける場合、構成法の特有の性質が生じてしまうためである。

³ここでは複素数の実部を $\Re[\mathcal{Z}]$ 、虚部を $\Im[\mathcal{Z}]$ のように表す。

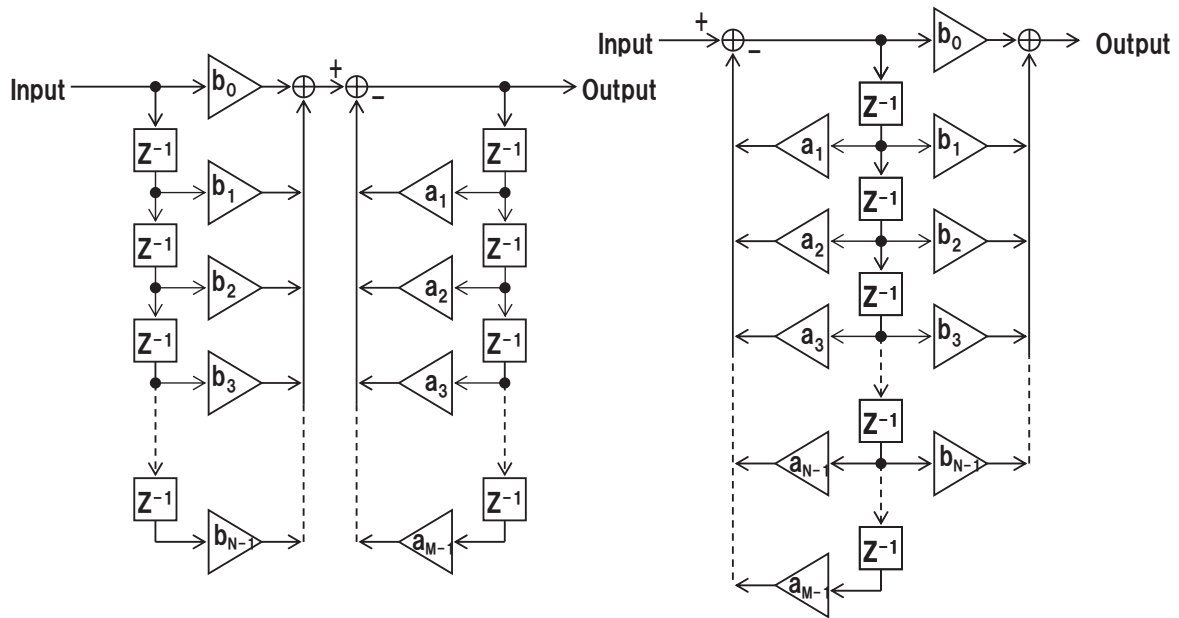


図 4.1: 直接型 I 型デジタルフィルタ構成図

図 4.2: 直接型 II 型デジタルフィルタ構成図

1. 演算精度が極めて高い
2. 計算速度が十分に速い
3. 演算に用いる変数を多く取れる

まずはじめにデジタルフィルタの最も一般的な形である直接型について述べていく。直接型を構成するとき、伝達関数は入力を $A(z)$ 、出力を $B(z)$ として、

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^{N-1} b(n) \cdot z^{-n}}{1 + \sum_{n=0}^{M-1} a(n) \cdot z^{-n}} \quad (4.15)$$

となり、フィルタを構成すると図 4.1、4.2 のようになる。直接 I 型は図 4.1 でいうと大きく左右 2 つのブロックに分けることが出来て、伝達関数を計算してわかるのだが、これは左右を入れ替えることが出来る。そしてそれらを入れ替えることにより遅延器を共有することが出来る。さらに遅延器を共有することで直接 II 型が得られる。

この構成法の特徴は次のようになる。

- 式 (4.15) が定まると、その式の係数 $a(n)$, $b(n)$ がそのままフィルタの乗算器の係数になる。
- 分母多項式 (図 4.1 の右ブロック) と分子多項式 (図 4.1 の左ブロック) を分離して考えることが出来る。

- フィルタ計算における変数の個数 (= 遅延素子個数) は I 型が $N + M$ 個、II 型が $\max(N, M)$ 個となる。I 型の方が作りやすいが、II 型の方が効率的ということである。
- フィルタの係数感度が高い。各係数 $a(n), b(n)$ の精度にシビア。
- 各係数の取り得る値の範囲が広い。
- 上 2 つの特性から、低ビット数では計算精度の向上、オーバーフロー防止の観点から実現が困難になる場合がある。

このように直接型は速度と精度はあるが、計算機への要求も厳しいものになってくる。直接型を基本に計算機側の要求も和らぐよう設計されているのが次の構成法となる。

4.3.3 縦続型

縦続型構成は、デジタルフィルタの伝達関数を因数分解し、得られた極および零点を 1 つずつ直接型によるフィルタで構成して、これらをカスケード接続する。この構成法では次式の経緯式で示される伝達関数をもとに作られる。

$$H(z) = g \frac{\prod_{n=0}^{N-1} (1 - \text{zero}[n] \cdot z^{-1})}{\prod_{n=0}^{M-1} (1 - \text{pole}[n] \cdot z^{-1})} \quad (4.16)$$

ここで g は系の利得、 $\text{zero}[n]$ 、 $\text{pole}[n]$ は零点および極であり、実数または共役複素数で示される。今回のデジタルシステムではフィルタの係数に複素数が許されていないので、共役対となる 2 つの複素根を掛け合わせ 2 次のフィルタとした次式を用いる。

$$H(z) = g \frac{\prod_{n=0}^{N_r-1} (1 - r_z[n] \cdot z^{-1}) \prod_{n=0}^{N_i-1} (1 + c_{z1}[n] \cdot z^{-1} + c_{z2}[n] \cdot z^{-2})}{\prod_{n=0}^{M_r-1} (1 - r_p[n] \cdot z^{-1}) \prod_{n=0}^{M_i-1} (1 + c_{p1}[n] \cdot z^{-1} + c_{p2}[n] \cdot z^{-2})} \quad (4.17)$$

ただし、ここでは $r_z[n]$ 、 $r_p[n]$ は実数の零点および極であり、

$$\begin{aligned} c_{z1}[n] &= -2\Re[\text{zero}[n]] \\ c_{z2}[n] &= \Re[\text{zero}[n]]^2 + \Im[\text{zero}[n]]^2 \\ c_{p1}[n] &= -2\Re[\text{pole}[n]] \\ c_{p2}[n] &= \Re[\text{pole}[n]]^2 + \Im[\text{pole}[n]]^2 \\ N &= N_r + 2N_i \quad , \quad M = M_r + 2M_i \end{aligned}$$

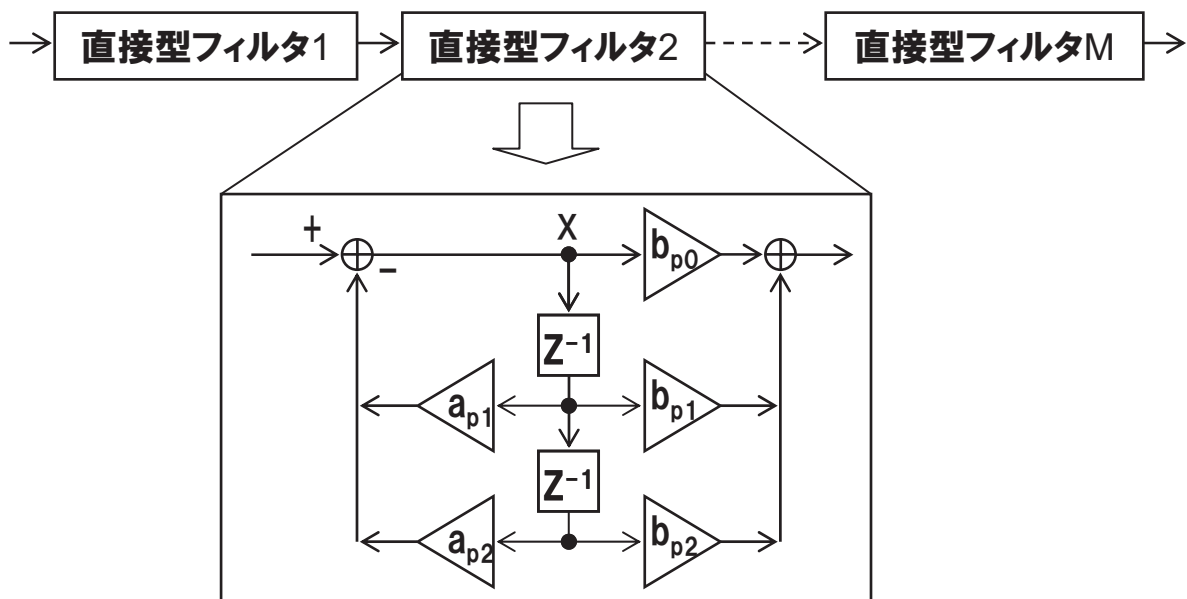


図 4.3: 縦続型デジタルフィルタ構成図

を満たす。また、 $\text{zero}[n]$ 、 $\text{pole}[n]$ は共役対を持つ複素根。縦続型フィルタの構成図は図 4.3 のようになる。この構成法の特徴は次のようになる。

- 伝達関数に含まれる極および零点の情報からフィルタ構成できる。
- 高次のフィルタが 1 次または 2 次のフィルタの集合として扱うことができる。そのため、演算誤差の累積が抑えやすく、直接型に比べてフィルタの計数感度の改善が可能。
- フィルタにより極の個数 M および零点の個数 N は必ずしも同じでないために、1 次および 2 次のフィルタからなる各部分のフィルタを直接 II 型の形で構成する汎用プログラムは作りにくい。よって、 M 、 N の値によりフィルタの構成をその都度考える必要がある。

この構成法は演算速度が直接型より遅くなってしまいが、演算誤差を抑えることを優先した構成になる。今回の実験で用いるフィルタ設計にはこの縦続型を用いることにする。

また、他には並列型による構成法などもあるがここでは詳しく述べないでおく。

4.4 遅延時間

デジタル信号処理においては ADC、DAC、それに CPU での演算などによりアナログ処理に比べて時間の遅延が避けられない。この遅延がデジタルの制御帯域を決定

することになるので、遅延時間を調べる事は非常に重要である。

正弦波信号の時間的なずれは位相特性に現れ、信号に遅延時間 t_ϕ を与えると、信号の角周波数 ω に応じて位相が $-t_\phi\omega$ だけ変化する。すなわち、位相 $\phi(\omega) = -t_\phi\omega$ は、遅延時間に相当する位相だといえる。

これより位相から遅延時間に相当するものを求めよる事が出来て、位相遅延から求める方法と、群遅延から求めるものの2通りの方法がある。

4.4.1 位相遅延特性

遅延を求める前に、位相には $2\pi n$ (n :整数) の不定性があり、位相 $\phi(\omega)$ と $\phi(\omega) + 2\pi n$ の間には区別が付かないので、まず、位相遅延特性を求める準備としてアンラップを行っておく⁴。そして先ほど述べたとおり、周波数によらない一定の遅延を与えた場合には、

$$\phi(\omega) = -t_\phi\omega \quad (4.18)$$

という関係式が成り立つ。そこで、この両辺を ω で割って、

$$t_\phi = -\frac{\phi(\omega)}{\omega} \quad (4.19)$$

という式で遅延を定義するのが位相遅延特性 (phase delay property) である。位相遅延特性は、直感的には分かりやすく、オリジナルの信号と測定している信号とのピーク位置の時間差・遅延を表すものであり、角周波数 ω の2つの正弦波の間の位相遅延差が t_ϕ だということは、2つの正弦波のピークの位置の時間差が t_ϕ であるということになる。

4.4.2 群遅延特性

群遅延から遅延を求める方法も書いておく。位相遅延が ω で割るという方法を取るのに対して、

$$t_{g\phi} = -\frac{d}{d\omega}\phi(\omega) \quad (4.20)$$

というように、 ω で微分するのが群遅延特性 (group delay property) である。位相遅延が単純に2つの正弦波の「ピークの差」なのに対して、群遅延は「うなりのピークの差」になる。周波数がほんの少しだけ違う2つの波を重ねると、うなりが生じてそれを、式で表すと

$$\cos((\omega + \Delta\omega)t + (\phi + \Delta\phi)) + \cos((\omega - \Delta\omega)t + (\phi - \Delta\phi)) \quad (4.21)$$

⁴アンラップとは 2π ごとに回っていた位相を $2\pi n$ 足し合わせる事で連続的に位相をつなげる操作である。

と書き表せて、加法定理を用いて書き換えると

$$2 \cos(\Delta\omega t + \Delta\phi) \cos(\omega t + \phi) \quad (4.22)$$

となり、 $\cos(\Delta\omega t + \Delta\phi)$ の部分がうなりとなる。そして、うなりの部分の位相遅延に相当するものが、 $t_\phi = -\frac{\Delta\phi(\omega)}{\Delta\omega}$ だが、この式に対して、 $\Delta\phi \rightarrow 0$ の極限を取ると、群遅延特性が得られる。群遅延には計算上、位相の不定性の影響を受けないという利点があり、アンラップを行わなくても遅延時間を求める事が出来る。

4.5 デジタル信号処理

デジタル信号の処理についてこの章では、

離散時間システム デジタルでは、アナログによる連続時間システムではなしえない周波数領域でのさまざまな処理が出来る。

IIR IIR フィルタの方が FIR フィルタより演算速度が速いが、精度に注意が必要。

縦続型 直接型よりやや演算はかかるが、係数感度の点から、縦続型の方が安定。

位相遅延 ADC や DAC、演算によりデジタルでは避けることの出来ない弱点。システムを設計する上でまず測っておく必要がある。

という結論をえることが出来た。

第5章 デジタルシステム

今回、干渉計の要請からも遅延の少ない高速 (UGF¹で ~ 1kHz) のサーボの構成が必要であるために、Digital Signal Processor(DSP) を用いて制御実験を行っていく事に
する。この章ではデジタルシステムに関する評価を、

1. DSP の特徴。 [17]
2. DSP のノイズレベルと遅延。
3. 時間に関するデジタル特有の問題。
4. レンジに関するデジタル特有の問題。

の順に調べていく。

5.1 DSP の特徴

この節では、デジタルシステムとして今回用いることにした DSP について述べる。

5.1.1 積和演算器

DSP の特徴は積和演算の演算速度にある。積和演算そのものは普通のプロセッサでも実行可能だが、DSP はこれを 1 クロックで高速処理するために次に示すような特徴のあるハードウェア構成をしている。

- クロックで乗算が可能な乗算器 (積和演算器) を内蔵
- ハーバード・アーキテクチャをもつ

図 5.1 のように積和演算を高速処理するために、1 クロックで処理が可能な乗算器と加算器を内蔵しているのが特徴である。

積和演算器には 2 つの入力 a と b があるが、1 クロックで演算を行うためには、2 つの入力データの読み込みも 1 クロックで同時に行わなければならない。そのため、プログラム・メモリとデータ・メモリおよびそれぞれのバスが分離独立したハーバード・アーキテクチャと呼ばれる構成を DSP はもっている。

¹後述の 6.3.2 参照。

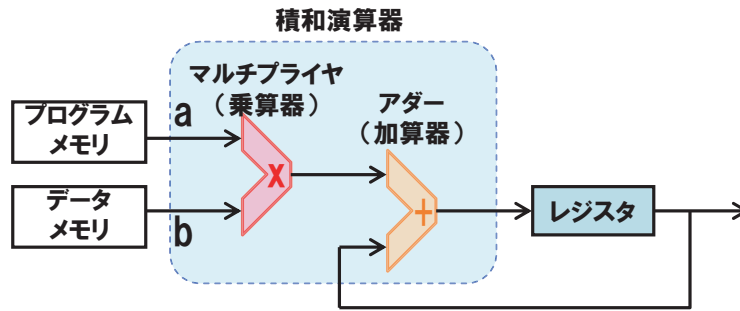


図 5.1: DSP のハードウェア構成

5.1.2 ソフトとハード

積和演算だけに関して言うと、たとえ CPU が積和演算に多くのクロックを消費するとしてもそれだけ早いコアを使えば DSP と同等の処理は出来ることになる²。

DSP を採用した他の理由には、テキサスインストゥルメンツ社が出しているスターターキットのような安価な評価ボードがあることや、C 言語で書いたプログラムを DSP 用のアセンブラに書き換えるコードコンポーザーというコンパイラがあるという使いやすさも基準となった。

5.2 DSP の評価

デジタルシステムとして DSP を用いるにあたって、この節ではその基本的な性能をしらべていく。本実験で用いる DSP のクロック周波数は 225MHz で、またこれ以降ことわらない限り、サンプリング周波数は 200kHz で用いる。

5.2.1 用いたデジタルシステム

用いたシステムは主に、DSP が搭載されたボードテキサスインストゥルメンツ社製 TMS320C6713DSK と ADC と DAC が搭載されている平塚エンジニアリング社製の IO ボード DSK6000IFA から構成されている。それぞれ実物の写真、スペック等は補足に載せておく。デジタルシステムの構成は図 5.2 のようになっている。DSP 搭載ボードの上に IO ボードが装着されており、シグナルはまず IO ボードで A/D 変換され、下の DSP ボードに送られる。そして、DSP で演算処理を終えた後に IO ボードに返されて、D/A 変換されて出力される。DSP での演算はスタンドアローンで行わ

²実際、今回使う DSP のクロック周波数は浮動小数点演算可能な 225MHz である。それに対して CPU ではクロック周波数が 3GHz を超えるコアもあり、1 回の浮動小数点積和演算にどれだけクロックを消費するかによっては CPU の方が早いことも理論上ありえる。ただ、現状では汎用の構成や用いるプログラムから、DSP の方が今は早い。

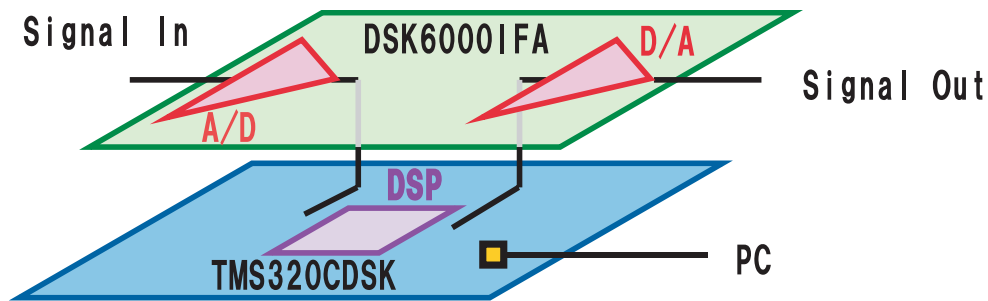


図 5.2: デジタルシステムの概略図

れており、PCから一度演算プログラムを送り込んでしまえば、PCとの接続を遮断しても動作し続ける。そのため、PCによる電氣的なノイズを防ぐ事が出来る。

5.2.2 ノイズレベル

デジタル制御により、高速で複雑な演算が可能な制御ループが実現できたとしても、そこでノイズが大きく発生してしまえば実際に使う事が出来ない。まずそのノイズレベルを測定しておく。測定方法は、今回用いるデジタルシステムのInputは 50Ω で終端しておき、デジタルフィルタの部分はスルー(ADCで変換された信号をそのままDACに返すプログラム)として、サンプリング周波数は 200kHz としておく。測定結果が図5.3の通りで、赤色がDSPからPCを切り離れた時の測定結果、水色がPCと接続したままの状態での測定結果である。緑の線は今回の測定を行ったFFTアナライザのInputを 50Ω で終端しておいた時のノイズフロアである。今回の測定により、PCと接続したままではノイズが多すぎる事が分かり、また切り離れた時でも今回のデジタルシステムのノイズフロアがおよそ $300\text{nV}/\sqrt{\text{Hz}}$ であることがわかった。

5.2.3 位相遅延

今回はこの位相遅延特性を用いて今回用いるデジタルシステムの評価を行う。デジタルフィルタの部分は同じくスルーとしておく。サンプリング周波数は、これからの実験においてでは全て 200kHz で行っていくが、今回は参考に 100kHz での測定も行った。図5.4がスルーのデジタルシステムの伝達関数、図5.5がデジタルシステムの位相遅延特性になっている。

この測定結果よりそれぞれの遅延時間は表5.1のようになり、併せて 1kHz での位相の遅延、 5° 、 10° 、 15° 遅れる時の周波数を載せておいた。これらの値はデジタルでフィルタを設計する時に参考になるものである。干渉計に用いられる制御サーボの位相余裕はUGF付近で 45° 以上あれば安心だと言われている。実際には位相余裕

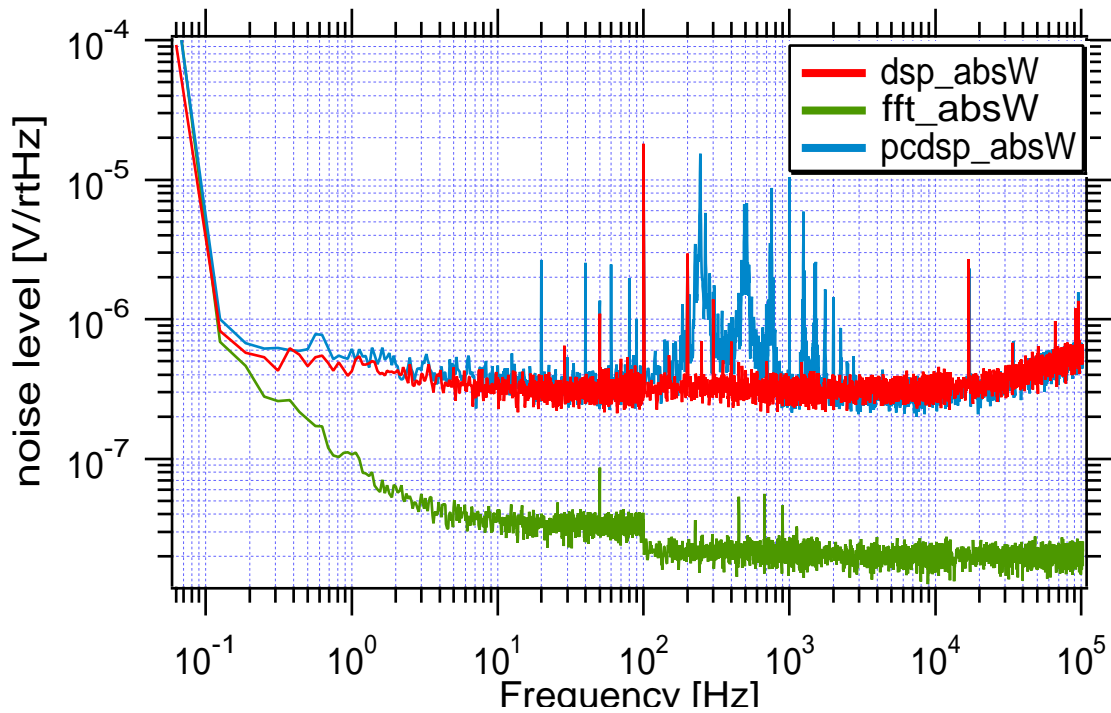


図 5.3: デジタルシステムのノイズフロア

が 45° ないサーボも存在し、そこでデジタルによる遅延で 10° 遅れることになると制御が厳しくなる可能性が出てくる。

サンプリング 周波数 (kHz)	遅延時間 (sec)	1kHz の位相遅れ (degree)	5° 遅れる 周波数 (Hz)	10° 遅れる 周波数 (Hz)	15° 遅れる 周波数 (Hz)
200	22.0×10^{-6}	7.92	631	1263	1893
100	39.5×10^{-6}	14.2	352	703	1054

表 5.1: デジタルシステムの遅延時間と位相遅れ。 10° 遅れてしまうと、アナログと同じフィルタ設計の場合制御が厳しくなる。

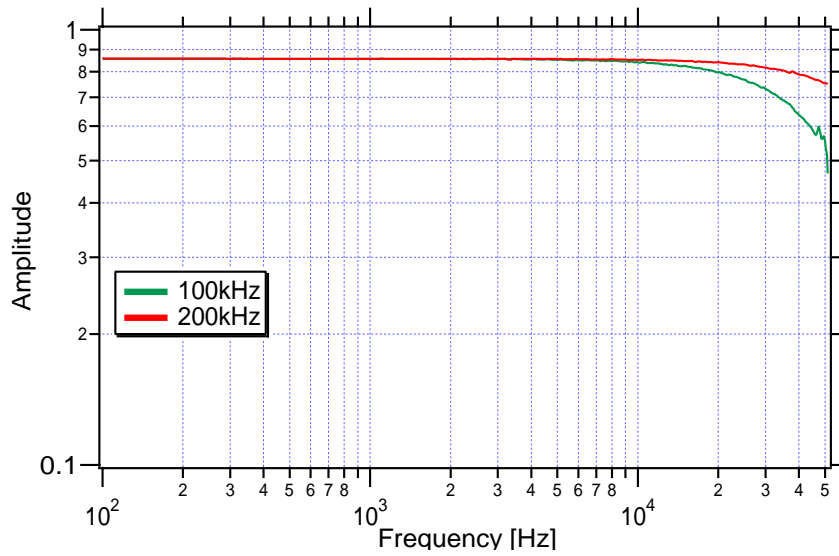


図 5.4: スルーフィルタの伝達関数絶対値。サンプリング周波数が 100kHz の場合、50kHz あたりではサンプリング定理の影響が見られる。

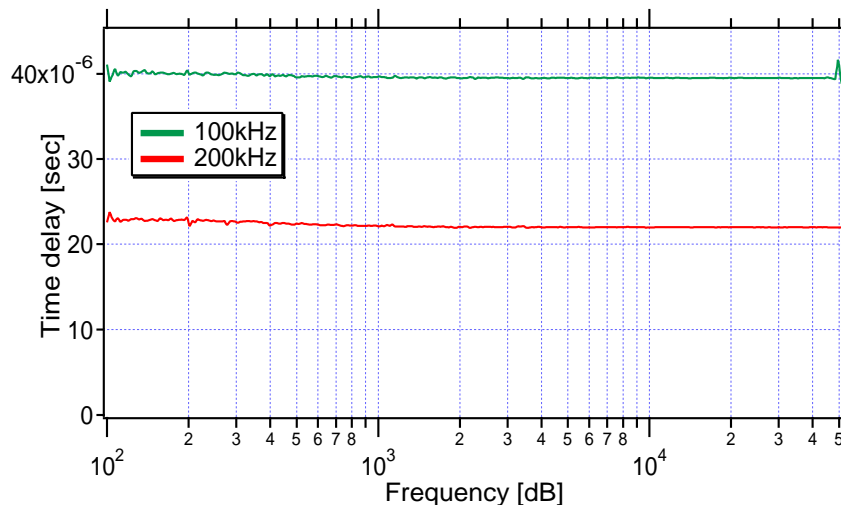


図 5.5: スルーフィルタの位相遅延特性

5.3 時間に関する問題点

ここではデジタルシステムを扱ううえで、アナログシステムには無い、時間に関する問題点について挙げていく。

5.3.1 サンプリング定理

サンプリング周波数は、アナログ波形を、デジタルデータにする際の標本化（サンプリング）で、1秒間における標本を採る頻度である。サンプリング定理とは、入力信号として最大周波数 f_{MAX} までの交流信号を A-D 変換したのち、これを D-A 変換して戻したとき、原信号が忠実に再生できるためには、少なくとも $2f_{MAX}$ 以上のサンプリング周波数 f_S でサンプリングする必要があるという定理である。サンプリング周波数 f_S の $1/2$ である周波数をナイキスト周波数と呼び、元信号にナイキスト周波数より高周波の成分が含まれていると、スペクトルにエイリアシングノイズ³ が生じる。

デジタルフィルタを設計するうえでは、ナイキスト周波数を越えたフィルタの設計は出来ない。またナイキスト周波数以上の信号が入るような場合には、サンプリングする前にアンチエイリアシングフィルタ⁴ を置く必要がある。

また出力の際には DAC の bit 階調により出力されるので、スムーズなサインカーブなどを得たい場合には、スムージングフィルタを追加する必要がある。

5.3.2 遅延による位相の遅れ

デジタルシステムで信号を処理する場合、アナログ信号をデジタル変換し、そのデジタル信号を演算してから再びアナログ変換するという過程をとるために変換や演算でアナログ信号処理に比べると時間がかかってしまう。今回使うデジタルシステムの信号処理ダイヤグラムは図 5.6 のようになる。

今回のシステムは複数チャンネルを同時サンプリング、同時処理するので、サンプリング周期をトリガーにして変換が進められる。まずはじめのサンプリングで取ってきたデータを ADC でデジタルに変換し、DSP のメモリに送る。次のサンプリングで演算を行い結果をメモリに送る。次のサンプリングで DAC がメモリに送る。さらに次のサンプリングで複数のチャンネルで処理された結果を出力用のメモリにまとめる。そして、5 つめのサンプリングで演算結果を同時に出力する。よって、デジタル処理においてはおよそ 4 サンプルの遅延が発生することになる。これは 200kHz サンプリングの場合、 $20\mu\text{sec}$ に相当する。これは前節

³エイリアシングノイズとは、折り返し現象により信号の帯域内にナイキスト周波数以上の信号が入り込んでしまい、それが雑音となってしまうこと。標本化された信号は、サンプリング周波数 f_S を搬送波とした DSB(double side band) スペクトラムをもつ。信号周波数がナイキスト周波数より大きくなると、DSB スペクトラムが信号の帯域内に入り込み、エイリアシング・ノイズが発生する。

⁴折り返し雑音をなくす目的で、ナイキスト周波数以上の信号を除去するためのフィルタ。

で実際に測定した $22.0\mu\text{sec}$ とほぼ一致している。

このシステムでは平行に信号を処理して、同時に出力することに主眼を置いているために、処理に余裕がとってある。デジタル制御システムの演算内容が確定すればもう少し速い処理システムに変更することは十分に可能である⁵。

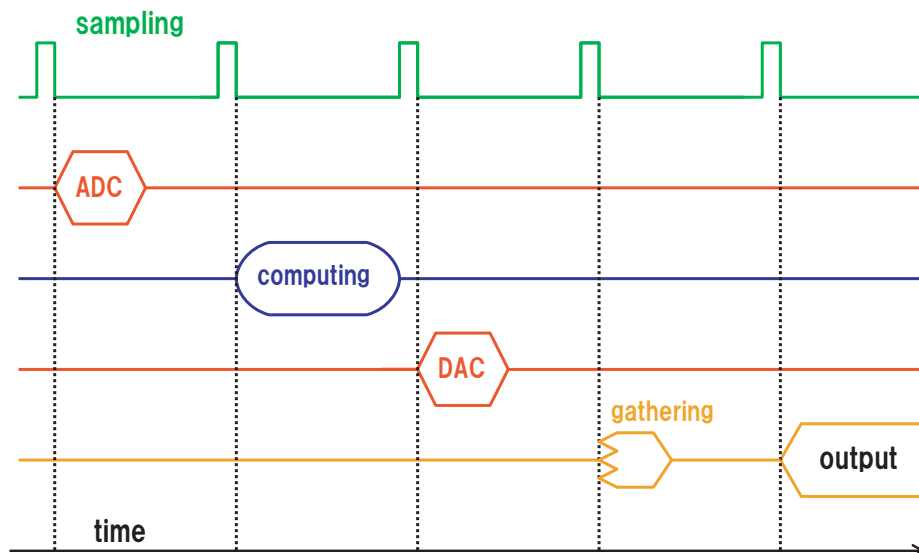


図 5.6: 今回使うデジタルシステムの信号処理ダイヤグラム

5.3.3 演算速度

信号のリアルタイム処理をしているために、演算に時間がかかり、次のデータが来る前に演算結果を吐き出せない場合、次のデータの処理が出来ずにシステムとして破綻してしまう。

デジタルシステムの組み方にもよるが、今回のシステムでは1 サンプル以内に演算は終えなければならない。サンプリング周波数 f_s と DSP のクロック周波数 f_{clock} から演算出来るサイクル数 N_{cycle} が次式のように求めることが出来る。

$$N_{\text{cycle}} < \frac{f_{\text{clock}}}{f_s} \quad (5.1)$$

今回、サンプリング周波数は 200kHz 、クロック周波数は 225MHz で使っているのので、 N_{cycle} は 1125cycle で $5\mu\text{sec}$ となる。ただ、1 サンプルの間の時間全てを演算に用いられるわけではないので、プログラムを設計する際には 10 % ほど余裕をみた 1000cycle 以内、 $4.5\mu\text{sec}$ に収まるように考えるべきである。

⁵実際、さらに速く 500kHz でサンプリング出来る IO ボードが存在し、本実験では登場しないが入手済みである。

5.4 レンジに関する問題点

ここではデジタルシステムを扱ううえで、アナログシステムには無い、レンジに関する問題点について挙げていく。

5.4.1 ダイナミックレンジ

ダイナミックレンジについて考えた時、アナログ回路の場合、例えば AD797 だとノイズレベルがおよそ $1.7\text{nV}/\text{rtHz}$ 、出力が $\pm 15\text{V}$ であるのでそのレンジは $10^9 \sim 10^{10}$ となる。一方デジタル回路の場合、今回使う ADC や DAC の場合 16bit であり、そのレンジとしては $-32768 \sim +32767$ でレンジは 10^4 のオーダーである。よってレンジの広い信号を扱い、高い SN 比が要求される場合にはホワイトニングやデホワイトニングが欠かせない。

5.4.2 桁落ち

計算機内の演算においては float や double といったデータ型を用いることでレンジは広がる。今回使った DSP のデータ型については表 5.2 のようになる。これより、float の場合小数部が $2^{23} \sim 8 \times 10^6$ なので、有効数字が 6 桁である。6 桁の差がある値どうしの、演算結果が間違ってくる可能性があり、double の場合 $2^{52} \sim 7 \times 10^{16}$ なので、有効数字が 16 桁あり、16 桁の差がある値どうしの演算結果が間違ってくる可能性がある。ただ、現実問題として、16 桁の精度あればたいいていのレンジはカバーできるので double を使っている分にはたいがい問題は無いと考えられる。ただ、double を用いるとそれだけ扱うデータ bit 数が増えるので、演算時間が増えることになり、精度と演算時間の兼ね合いで使い分ける必要が出てくる [18]。

型	サイズ	表現	ビット割り当て数 (符号、指数、小数)
float	32 ビット	IEEE 32 ビット	(1, 8, 23)
double	64 ビット	IEEE 64 ビット	(1, 11, 52)

表 5.2: TMS320C6000 C/C++ のデータ型

桁落ち現象の検証実験

桁落ちの現象は低周波側で現れることが多い。それは低周波信号ほどデジタルフィルタ内での帰還の影響を受けるからである。その例が図 5.7 になる。ここでは float で

演算を行った時と、計算値を載せてある。このフィルタは4.4Hzに極を持ち、48Hzに零点を持つ2次の低周波ゲインアップ用のフィルタである。このフィルタは今回実験を行う Fabry-Perot 共振器の制御にもつかわれているものになる。float による測定結果は低周波側でゲインが頭打ちになっており、位相もだいが計算値とは異なっている。

桁落ちについて検証実験を行った。検証方法はこのデジタルフィルタの input にプログラムで同じ1という値を与え続ける。低周波の信号ほどこのような条件に近くなる。このフィルタはIIR 縦続型で構成されており、図 4.3 でいう x の点で演算中に一番値が大きくなる。double を用いてループをある回数回した時、各点での値を示したのが表 5.3 になる。出力を見ると低周波側でゲインアップしていることが読み取れる。演算過程の x 点においては 10^4 回を超え出すとすでに float の有効数字である 6 桁を超えてしまい、これ以上のループだと float の場合、桁落ちが起きてしまうことがわかった。

これがどれくらいの周期に相当するかというと、サンプリング周波数が 200kHz で 10^4 回分だと考えると、20Hz に相当する。実際図 5.7 を見てみると、20Hz あたりから位相が計算値からはずれてきている。

以上のことから、フィルタのゲインそのものが高なくても、演算途中で値が大きくなることがあり、フィルタの設計次第ではそれが桁落ちを起こすということがわかった。また、このフィルタに関しては double を用いることで桁落ちが無くなり計算値と一致することは確認済みである。

ループ回数	入力	x 点での値	出力
1×10^0	1	1.000×10^0	1.000×10^0
1×10^1	1	5.495×10^1	1.025×10^0
1×10^2	1	5.004×10^3	1.280×10^0
1×10^3	1	4.569×10^5	4.431×10^0
1×10^4	1	1.983×10^7	5.259×10^1
1×10^5	1	2.320×10^8	5.349×10^2

表 5.3: 桁落ち現象の検証

5.4.3 量子化雑音 (丸め誤差)

標本化され、量子化、符号化されたデジタル値を再びアナログ信号に戻したとき、入力信号との間で生じる差のことである。

理想的な A-D コンバータでも、連続量が不連続なデジタル・データに変換されるので、量子化による誤差が生じる。n ビットの A-D コンバータで変換する場合、合

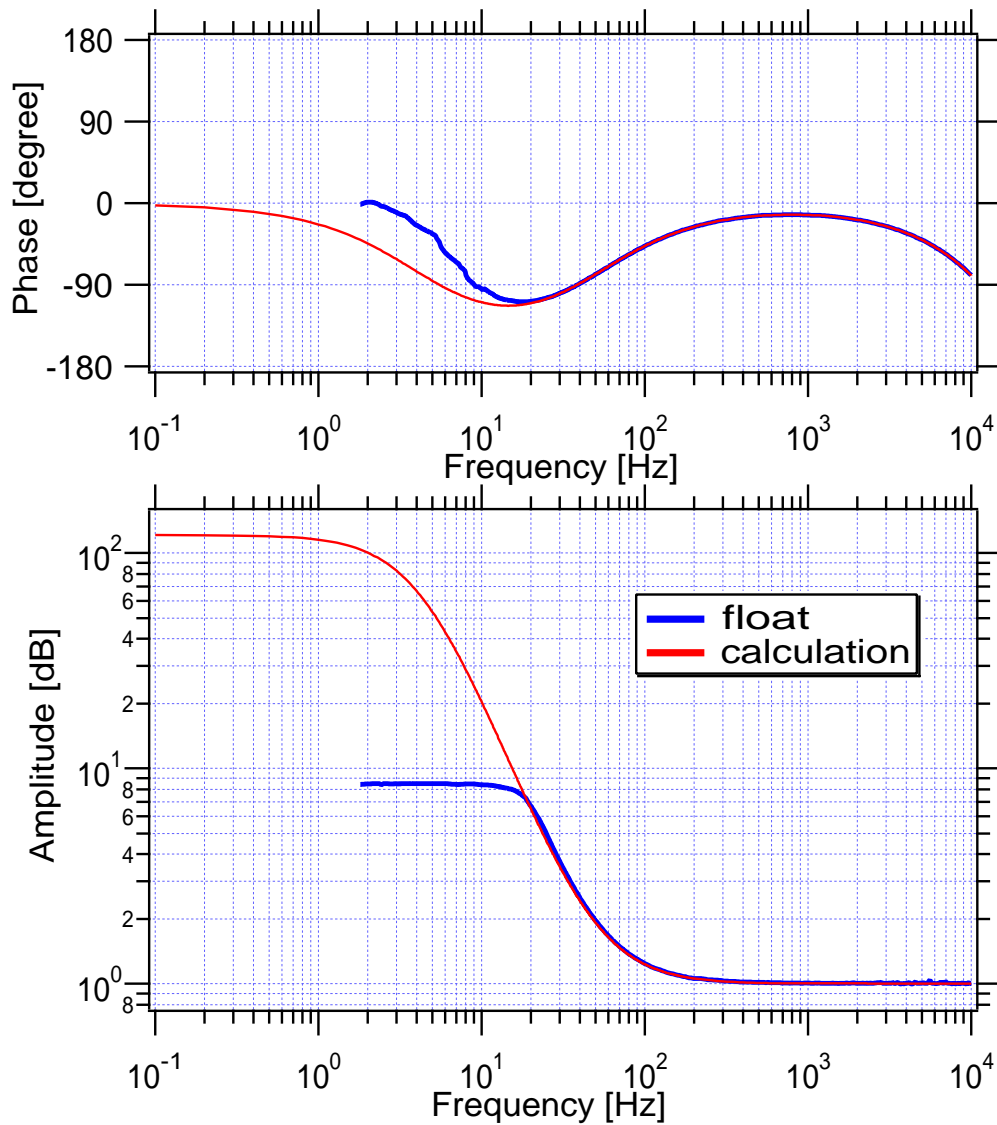


図 5.7: 低周波域における桁落ち現象。float を用いた測定データが2Hz 以上にしかないのは、そこまでの周波数で確認されたためである。double を用いて測定した結果は、ここには載せていないが、きれいに計算値と一致した。

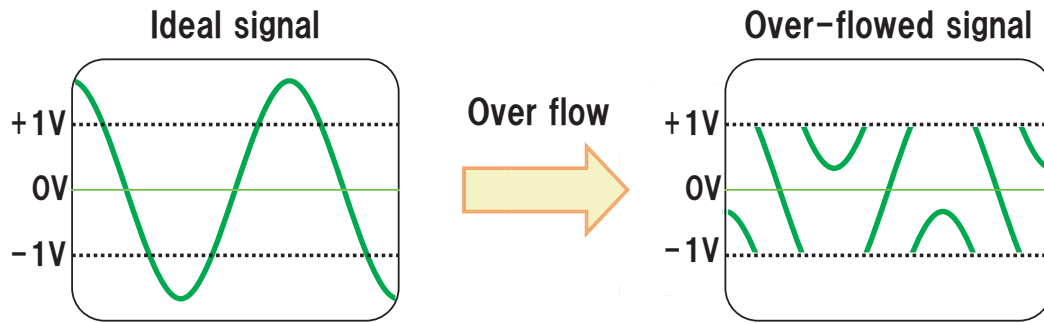


図 5.8: オーバーフローによる信号の折り返し。

計 2^n 個の段階のデータに分けられることになり、各段階の途中の値は隣り合う上か下の値に丸められる。したがって量子化には本質的に $1/2 \times 2^{-n}$ の誤差が生じる⁶。

5.4.4 オーバーフロー

コンピュータが数値演算を行なった結果が、扱える数値の最大値を超えること。桁あふれともいう。一つの数値を表現するために割り当てられた記憶容量は決まっているため、扱える数値には上限がある。演算結果がこの上限を超えるのがオーバーフロー⁷である。例えば、16 ビットの long の場合、-32768 から 32767 までの整数を扱うことができるが、ある演算を行なった結果が 50000 になったら、16 ビット long の枠内では扱うことができないため、オーバーフローとなる。

実際にどのようなことが起きるかということ、今回用いるシステムの long では bit の割り当てが、16 進数表示で 0000 ~ 7FFF が正領域の +0 ~ +1 に、8000 ~ FFFF が負領域の -1 ~ -0 に対応している。オーバーフローが起きた場合のシグナルの様子は図 5.8 のようになり、オーバーフローが起きてしまえばもはや全く違ったシグナルとなってしまうので、オーバーフローする時にはサチレーションするようなど対策を取らなければならない。

⁶ 2^{-n} のことを Least Significant Bit (LSB) ともいう。最下位ビットで 2^{-n} の重みをもつビット。n ビット符号のもっとも下位にあたる。デジタル値のバイナリ・データの LSB は D-A コンバータや A-D コンバータの最小分解能である。D-A コンバータでは、この値より細かく出力を可変することはできない。

⁷バッファオーバーフローとは、また別である。C++ 言語などのプログラミング言語で書かれたプログラムでは、プログラムが確保したメモリサイズを越えて文字列が入力されると領域があふれて (オーバーフロー) しまい、予期しない動作が起きる。これはバッファオーバーフローと呼ぶ。

5.5 デジタルシステム

この章の結論として、

DSP ハーバードアーキテクチャを持ち、1クロックで積和演算するリアルタイム信号処理に特化したマイクロプロセッサである。

DSP の特性 ノイズレベルは $300\text{nV}/\sqrt{\text{Hz}}$ で、遅延が $22.0\mu\text{sec}$ である。

演算速度 今回の設定では1演算を $1000\text{cycle}(4.5\mu\text{sec})$ 以下に終えなければならない。

レンジ 桁落ちやオーバーフローには注意してプログラムを設計しなければならない。

ということが得られた。

第6章 TAMA300のFabry-Perot共振器片腕mass lock

デジタルフィルタによるサーボでのTAMA300制御の第一歩としてFabry-Perot共振器の片腕を鏡(mass)による制御に取り組んだ。

アナログフィルタによる鏡の制御による片腕のみのlockの場合、lockするまでに数分かかる。地面振動が激しい時などは1時間かかってもlockしないことがある。このlock acquisitionをデジタルフィルタを用いて行うことで、lockまでの時間を短縮すれば、制御システムとして有効であることを証明することが出来る。

また、Fabry-Perot共振器片腕のlockを成功させることによって、ここでの手法は他の制御ループにも応用が可能となり、結果として干涉計全体の制御の向上につながることになる。その可能性を見極めるためにも今回の実験は重要だと考える。

この章では

1. 実験のセットアップ。
2. デジタルフィルタのプログラミング。[19]
3. Fabry-Perot共振器のlock acquisition。

について述べてゆく。

6.1 実験部干涉計構成

今回はTAMA300で実際に使われているFabry-Perot共振器を用いて実験を行った。TAMA300は基線長300mの腕を2本持ち、モードクリーナーが10mある国内最大のレーザー干涉計型重力波検出器である。今回の実験ではそのFabry-Perot共振器の片腕を使いlock acquisitionに取り組んだ。干涉計にはリサイクリングミラーやもう一本のFabry-Perot共振器もあるが、今回はまずデジタルフィルタによるサーボのテストを行うため、それらは使わずに実験を行った。干涉計の構成は図6.1のようになり、今回使わなかった方のFabry-Perot共振器は何も制御を行っていない。

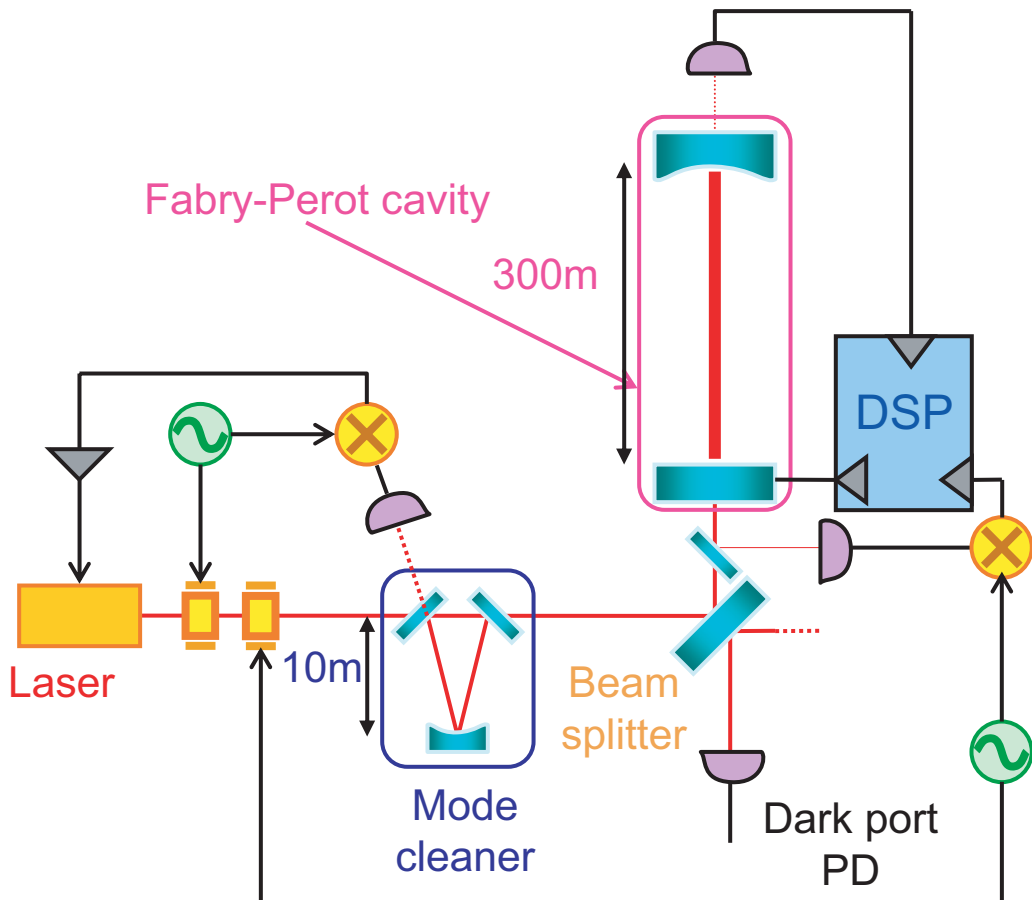


図 6.1: 干渉計の構成図。アナログフィルタで干渉計を制御する時には DSP の部分をアナログフィルタに置き換え、透過光は制御に用いない。

6.2 NSmass lock用の error signal 規格化プログラミング

デジタルフィルタのプログラムに関してここでは各演算を順番に説明していく。

6.2.1 フィルタ部分の設計

フィルタ部分に関しては、改良できる可能性もあるが、今回は規格化による lock acquisition の改善を調べるためにデジタルサーボ用のフィルタの形はアナログフィルタと伝達関数の絶対値が同等になるフィルタを作成した。位相に関してはデジタルとアナログの相互の変換等があるため、デジタルの方が位相遅れが大きくなってしまう。アナログフィルタの回路図は後ろに図 C.1 としてのせておいた。

また、Fabry-Perot 共振器の制御においては、lock した後に共振が安定に保たれるよう低周波側でゲインアップを行っている。これもアナログフィルタと同等の操作をデジタルで行うことにする。

デジタルフィルタの設計は式 (C.5)、式 (C.6) のようにアナログフィルタの伝達関数を式 (4.17) にそって s-z 変換する。フィルタの構成は、演算誤差を抑えることを優先するため、IIR フィルタの縦続型で作成した。デジタルフィルタの伝達関数は

$$H(z) = g \cdot \frac{\prod_{n=1}^N (b_{0n} + b_{1n} \cdot z^{-1} + b_{2n} \cdot z^{-2})}{\prod_{n=1}^N (1 - a_{1n} \cdot z^{-1} - a_{2n} \cdot z^{-2})} \quad (6.1)$$

のようになる。 g はゲインであり、 a_{*n} と b_{*n} (*は 1 or 2) は定数になる。lock 前では $N = 5$ 、lock 後では $N = 6$ である。

プログラムの構成としてフィルタ部分は、定数の定義部、メイン文中のバッファクリア、ループ中のフィルタ演算の大きく 3 つの部分に分かれ、それぞれプログラムを図 D.1、D.2、D.3 に載せておく。プログラムの書き方としては、まだまだ改善の余地はあると思われるが、現在演算する上では支障は出ていない。

定数の定義部のプログラムでは変数が am、bm、cm、dm と 4 組定義されているが、cm、dm はそれぞれ am、bm のフィルタ変数に lock 後のゲインアップフィルタ変数を加えたただけのものである。実際に IIR フィルタの演算を行っているのは D.3 のなかで、桃色に網掛けしてある部分になる。またこのプログラムでは lock したかどうかのシグナルはアナログフィルタの時とタイミングをそろえるために外部からの信号を頼りにしている。その外部からの信号が LS で表される変数になっている。

アナログフィルタとデジタルフィルタの lock 前と後の伝達関数の測定を行った結果がそれぞれ図 6.2、6.3 のようになっている。デジタルフィルタの伝達関数計算値

$H_{cal}(\omega)$ は、デジタルフィルタ作成に用いた変数と、through で測定した時の遅延の値から次式のようになる。

$$H_{cal}(\omega) = H(z)|_{z=\exp[i\omega T_s]} \times \exp[i\omega t_{delay}] \quad (6.2)$$

ここで T_s はサンプリングタイム、 t_{delay} は through フィルタでの遅延、 ω が角周波数ある。今回のデジタルシステムでは $T_s = 5 \times 10^{-6} \text{sec}$ 、 $t_{delay} = 22.0 \times 10^{-6} \text{sec}$ という値を代入することになる。

その測定結果を比較してみると、伝達関数の絶対値についてはアナログフィルタもデジタルフィルタの実測値と計算値の全てが一致している。位相に関して、デジタルフィルタの方が遅れが見られるが、実測値と計算値が一致していることから、これはデジタルシステムそのものの遅延に由来するもので、プログラムには関係ないということがわかる。

6.2.2 規格化部分の設計

デジタルフィルタを使うことの今回の目的がフィルタに透過光量によるこの規格化を取り込むことにある。規格化を行う際に避けることが出来ないのが除算というプロセスになり、かつての計算機には除算は演算時間がかかり高速処理は出来なかったが、技術の進歩によりその演算も高速処理出来るようになった。

ただ、高速処理できるようになったといえども、加算器と乗算器しか搭載していないチップにとっては簡単ではない。

除算をするにもいくつかのプログラムがあり、

A/B 単に '/' 記号を使った除算。整数除算における商を求める。

`__divd` ランタイム・サポート算術ルーチンで、double の除算を行う。プログラム中では `__divd(double, double)` のように使う。

`__rcpdp` TMS320C67x C/C++ コンパイラの組み込み関数で、64 ビット double の逆数近似値を計算する。プログラム中では `__rcpdp(double)` のように使う。

などがある。まず、浮動小数点を扱うので '/' は使うことが出来ず、また演算速度の問題から `__rcpdp` の方が処理が速いのでこれを使うことにした。

プログラム中で規格化の演算を行う場所については入力してきたシグナルがフィルタ演算される前に設定した。後に述べるトリガーについても同一の演算式に

$$(\text{エラーシグナル}) = (\text{入力シグナル}) \times (\text{透過光量の逆数}) \times (\text{トリガー})$$

という演算を加えた。透過光量はこの演算を行う前に `__rcpdp` により逆数を求めている。ここでのトリガーは 1 or 0 である。

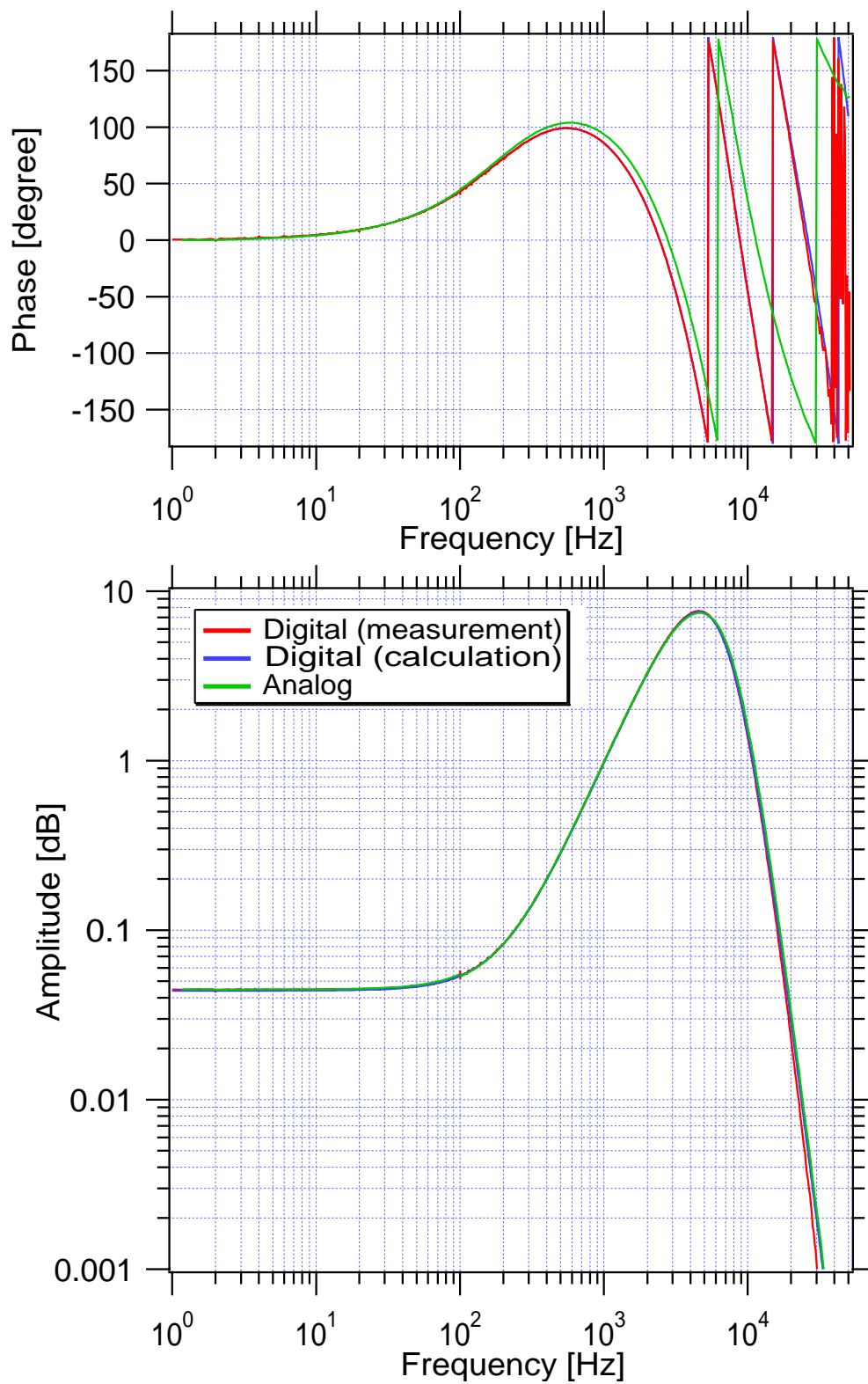


図 6.2: lock 前の Fabry-Perot 共振器制御用フィルタの伝達関数

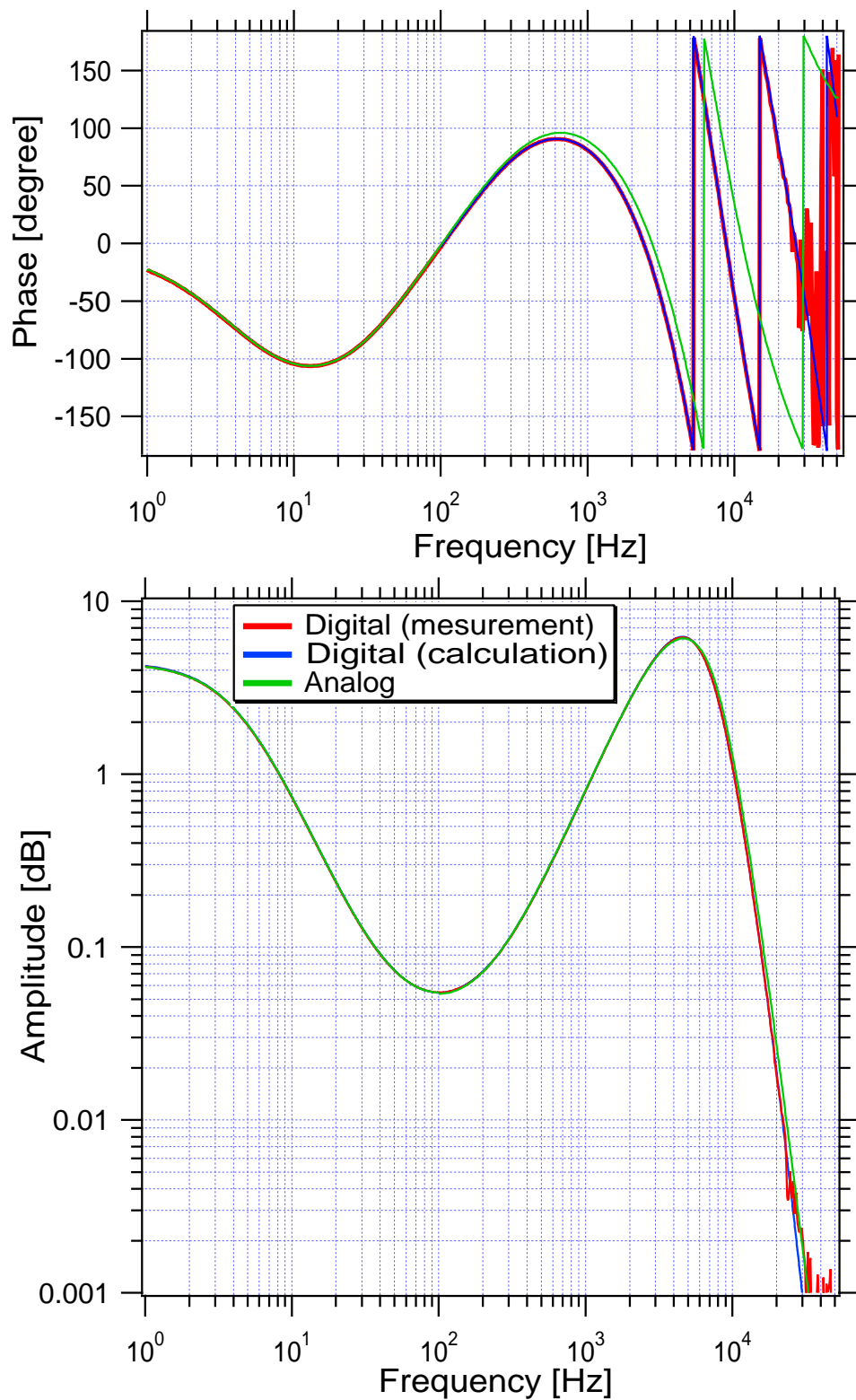


図 6.3: lock 後の Fabry-Perot 共振器制御用フィルタの伝達関数

6.2.3 トリガーの追加

除算を行うときに透過光量が0の場合、不可能な演算になってしまうという理由と、error signal には共振状態を示すシグナル以外のものも混ざっているということから、トリガーを演算プログラムに組み込む。プログラムは図 6.4 のようになる。入力された透過光量 Tr0 を short から double に変換する際に併せてオフセットが0になるような調整と、最大値を1に規格化するように計算している。

トリガーを設定するスレッシュホールドについては今のところはその時の透過光量と共振時の error signal のタイミングから決めているが、将来的には自動で設定されるようにする予定である。

```
interrupt void int4(void)
{
  /*define variables*/
  double Tg;
  short Tr0;

  Tr0 = *(volatile short *)DSKIF_AD3; /*Transmitted light*/
  Tr = (double)Tr0*3.051757e-5 + 0.1; /*modified Trs. light*/

  /******Triger setting [Tg=Triger]******/
  if (Tr>0.03)
  {
    Tg = 1.;
  }
  else
  {
    Tg = 0.;
  }
  :
  :
  :
}
```

図 6.4: トリガー設定のプログラム

6.2.4 オーバーフロー対策

short(16bit) でインプットしたシグナルは直後にオフセットなどの補正を受けて、double(浮動小数点 64bit) に変換され演算が進められていく。演算中は double を使っている分には問題は無いが、最後に D/A で出力する際に再び short に変換しなくてはならない。その際に short のレンジをオーバーしてしまうと図 5.8 のようにシグナルが折り返してしまうので、図 6.5 にあるプログラムで折り返し起きないように上限と下限でリミットをつける。また、このレンジを変えることで feedback のレンジも変えることができる。

```
/******Over flow problem******/  
  
    SI= SI*G;  
  
        if (SI>=32767) So = 32767;  
    else if (SI<=-32768) So = -32768;  
    else                So = (short)SI;
```

図 6.5: オーバーフロー対策

6.2.5 演算速度

複雑な演算プログラムを作っても、演算が 1 サンプル以内には収まらなければそれは使うことが出来ないので、演算速度の測定を行った。測定方法は、プログラム中で信号を読み込み、演算し、結果を書き出すこのループを 10^6 回連続で行わせてその合計時間を計り、平均時間を出した。また、変数および定数については double と float の 2 通りで定義し演算時間を比較した。その結果が表 6.1 のようになった。

これより、実際 float の方が演算時間がかからないとわかったが、桁落ちが確認された件や double を使っても演算時間にまだ余裕があるということからも double で定義しプログラムを設計することにする。

double	float	200KHz sampling
3.47 μ sec	2.95 μ sec	5.00 μ sec

表 6.1: 演算速度

6.3 Fabry-Perot 共振器 lock テスト

TAMA300 の片腕 300mFabry-Perot 共振器を用いて、lock acquisition を行い、フィルタの形式によりその lock acquisition にどれだけ差が出てくるかを調べた。調べた制御法は、

1. 従来の TAMA300 アナログフィルタ。
2. アナログと同じ伝達関数（絶対値）を持つデジタルフィルタ。
3. 2 のデジタルフィルタに規格化の演算を付加する。
4. 2 のデジタルフィルタにトリガーの演算を付加する。
5. 4 のフィルタにさらに規格化の演算を付加する。

となる 5 種類の方法で実験を行った。

実験で測定したものは lock acquisition にかかる時間とその時のオープンループ伝達関数である。手法はある一定時間 lock acquisition を行い、共振器が lock したら手動により feedback を切って lock を落として、また lock acquisition を行うという操作を繰り返した。lock がかかってから 3 秒ほど経ってから落とすようにし、それより短い時間しか lock が保てない場合はカウントしない。実験を行う前に各フィルタが十数分かそれ以上安定に lock し続けることは確認してある。また、feedback のゲインに関しては最適なゲイン設定法が確立されておらず、各フィルタでそのとき一番 lock しやすかったゲインを採用し、その後はゲイン一定のまま測定を行った。

6.3.1 lock acquisition 測定

デジタルフィルタによる lock acquisition のうちでどの操作が一番効果的であるかを調べるために、先に述べた 5 つのフィルタタイプで測定を行った。それぞれのフィルタにおける特徴と期待する結果は次のようになる。

1, アナログフィルタ

まずはじめに、現在 TAMA300 でサーボに使われているアナログのフィルタを用いて、片腕の Fabry-Perot 共振器がどれくらいの時間で lock するのか統計をとった。このフィルタでの結果が現状の TAMA300 で、改良を行う前にあたる。

2, デジタルフィルタ（アナログ同形式サーボ）

このフィルタは伝達関数の絶対値がアナログフィルタと同じになるようにしたもので、基本的にはアナログフィルタとの違いがそこまで出ないことを期待する。デジタルで構成したために若干の位相遅れがあるので、その違いが出る可能性はある。

3, デジタルフィルタ (規格化付加)

デジタルフィルタ (アナログ同形式サーボ) に、透過光量による規格化のプログラムを加えた。error signal における共振時のシグナルは増幅されるが、規格化の場合透過光量が少ない場合における他のシグナルも増幅されるため、lock acquisition が改善するのか、それ以前に lock 出来るのかどうか未知である。

4, デジタルフィルタ (トリガー付加)

デジタルフィルタ (アナログ同形式サーボ) に、透過光量によるトリガーのプログラムを加えた。error signal 自体が大きくなるわけではないが、トリガーを加えることにより lock acquisition と関係ないシグナルの影響が減るため、feedback のゲインを大幅に上げることが出来る。

5, デジタルフィルタ (トリガー、規格化付加)

デジタルフィルタ (アナログ同形式サーボ) に、透過光量による規格化とトリガーのプログラムを加えた。今回の実験における完成形であり、計算通りであれば lock acquisition に要する時間が減少し、lock acquisition の成功率も上がるはずである。

6.3.2 測定結果

まず、'3, デジタルフィルタ (規格化付加)' の結果に関しては共振状態を示す信号以外も増幅されてしまい、lock することはなかった。

それ以外の結果としては、'1, アナログフィルタ' と '2, デジタルフィルタ (アナログ同形式サーボ)' は lock acquisition に要する時間についてはほぼ同じような結果が出たので、lock acquisition のみに関してはアナログフィルタとデジタルフィルタに大差はないと言える。

また、トリガーを加えることで lock acquisition に要する時間が減少し、さらに規格化を加えることによって一層の lock acquisition 短縮化に成功した。

各フィルタ形式の測定結果は表 6.2 のようになった。それぞれの指標は次のように求めている。

フィルタ形式	FB ゲイン [倍]	UGF [Hz]	Gain margin [dB]	Phase margin [degree]	average time [sec]	図番号	
アナログ	-	268	10.3	40.0	70.0	6.6	
デジタル	アナログ同形式	1	546	4.51	41.6	65.1	6.7
	+トリガー	34.1	617	3.36	38.8	12.9	6.8
	+トリガー規格化	7.9	575	4.60	40.5	5.59	6.9

表 6.2: 各フィルタ形式による lock acquisition 測定結果

FB ゲイン アナログフィルタに関しては今回比較していない。デジタルフィルタについては、アナログ同形式サーボを基準に feedback ゲインを何倍にしているかの値である。

UGF Unity Gain Frequency の略で、オープンループ伝達関数のゲインが 0dB になる周波数。これ以上の周波数では制御が加えられなくなる周波数。

Gain margin オープンループ伝達関数の位相が 0 °に達した時のマイナスゲインの幅をゲイン余裕 (Gain margin) という。ゲイン又は負の feedback 量がマージン分だけ増えれば発振してしまう。

Phase margin ゲインが 0dB 迄下がった時の位相角と 0 °の差を位相余裕 (Phase Margin) という。つまり、発振に至るまでには、これだけの位相角の余裕があるという値。

average time feedback をオンにしてから Fabry-Perot 共振器が lock するまでの平均時間。この実験で一番注目する指標。

図番号 各フィルタで lock を行った時のオープンループ伝達関数 (上段:位相、中段:絶対値) と、下段には lock acquisition に要した時間のヒストグラムを載せている。

オープンループ伝達関数の図に関して、1.3kHz あたりに見える共振は Fabry-Perot 共振器の鏡でなんらかの共振¹が生じているため、フィルタ自体に問題はない。そのため、その共振周波数以降はやや計算値とずれてしまっている。伝達関数を測定するうえで、100Hz 以下および 4kHz くらいから上の周波数ではうまくデータを取ることが出来ないが、その間の周波数では計算値と実測値がよく一致している。

¹干渉計の鏡は重力波という非常に小さな変動を捕らえるために繊細に作られている。この年は度重なる地震のためにダメージを受けてしまったのではないかと考えられている。

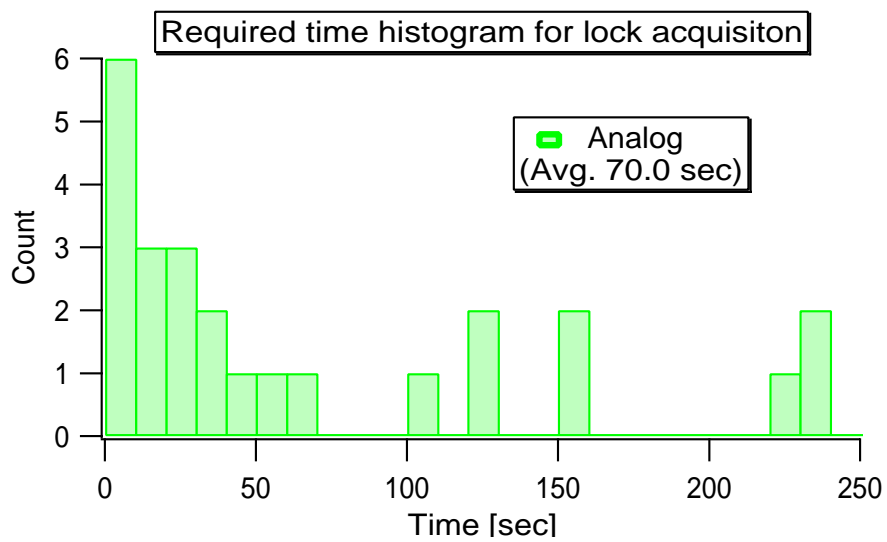
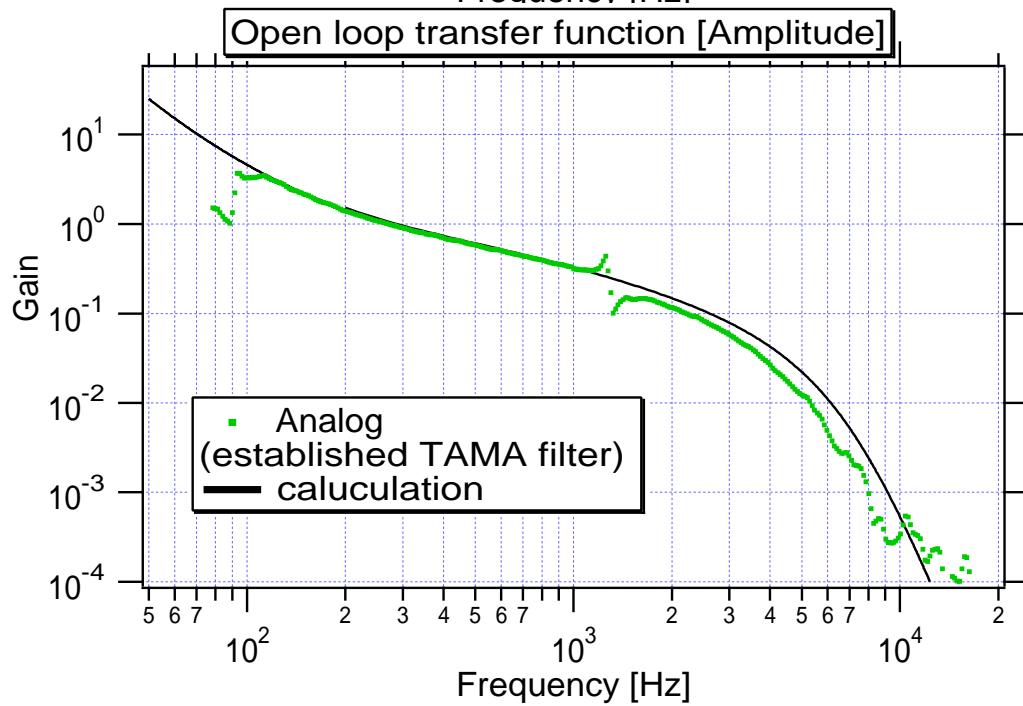
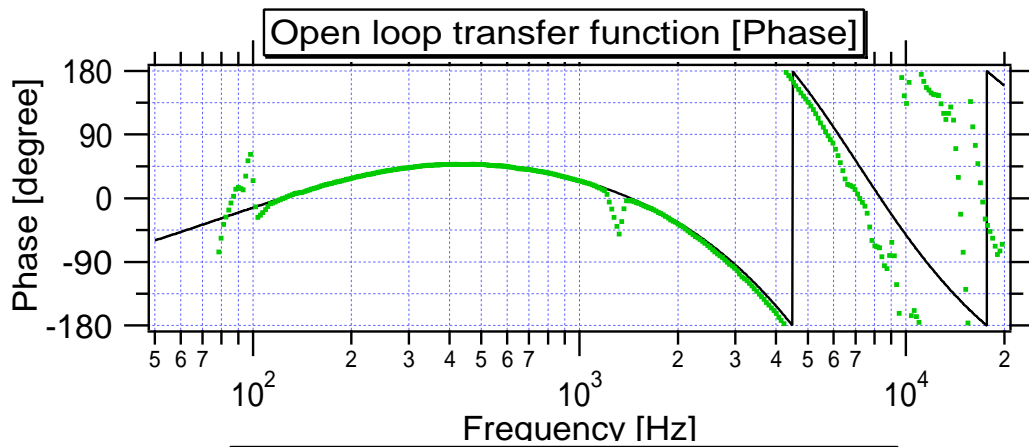


図 6.6: アナログフィルタによる lock acquisition

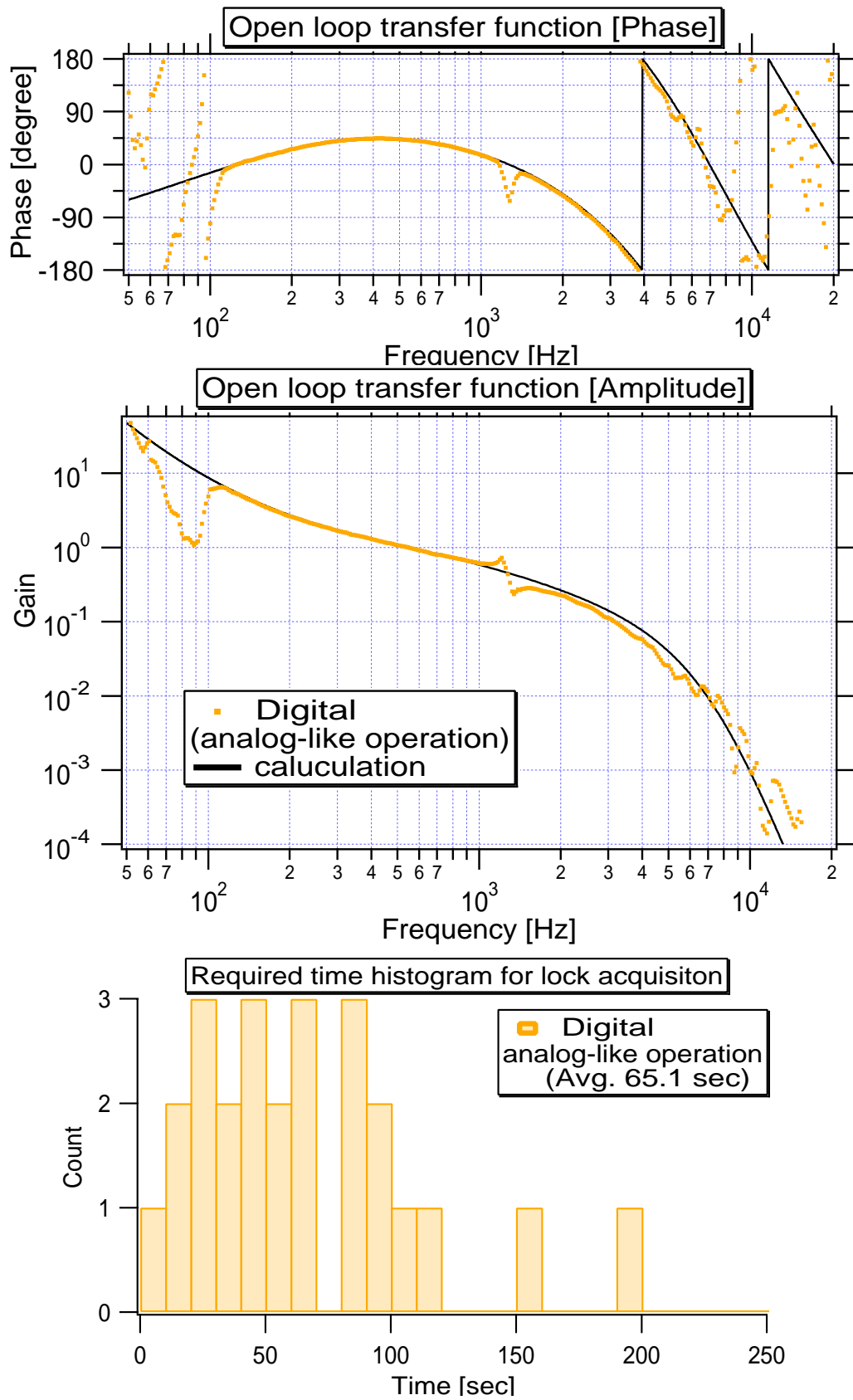


図 6.7: デジタルフィルタ (アナログ形式サーボ) による lock acquisition

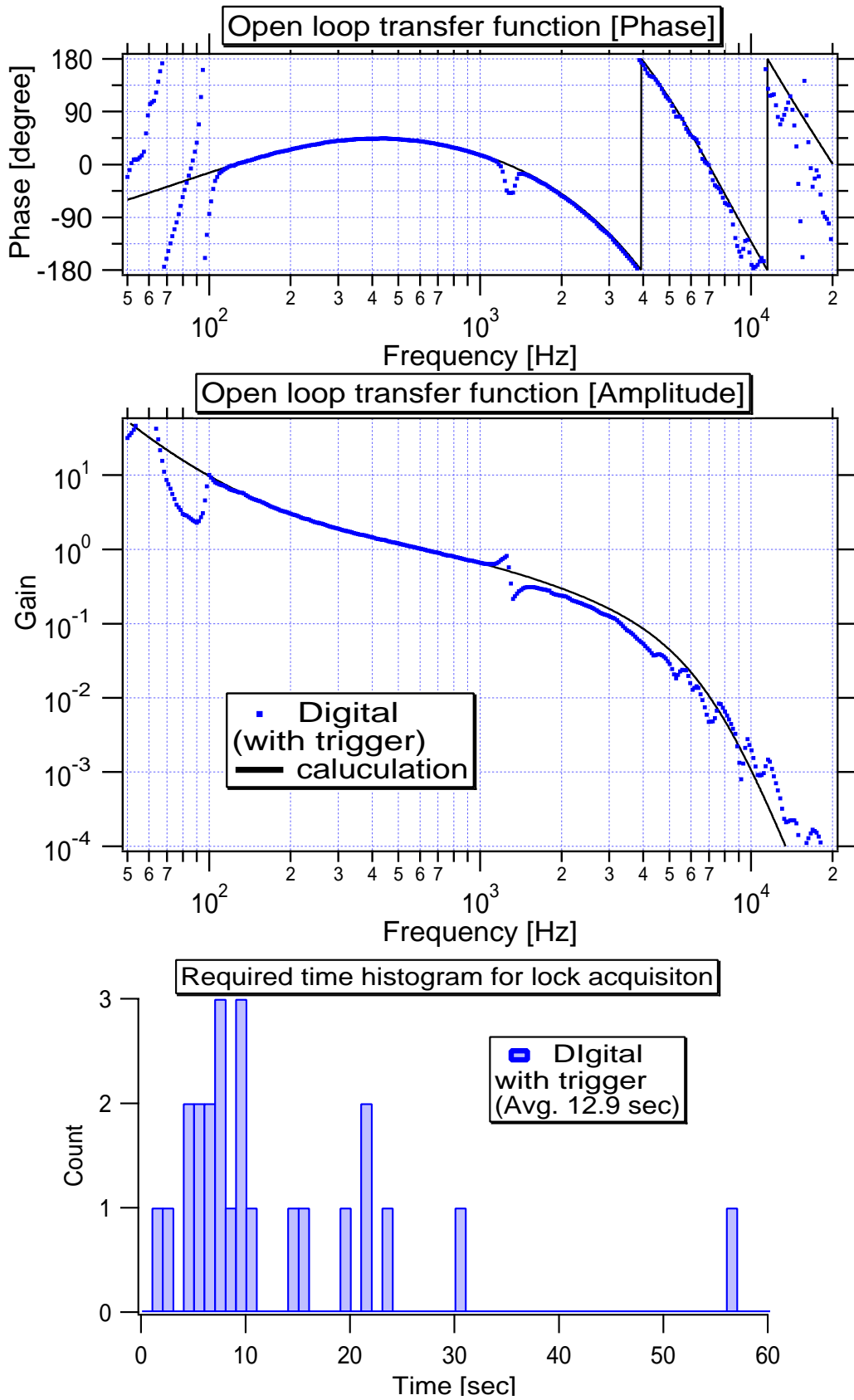


図 6.8: デジタルフィルタ (トリガー付加) による lock acquisition

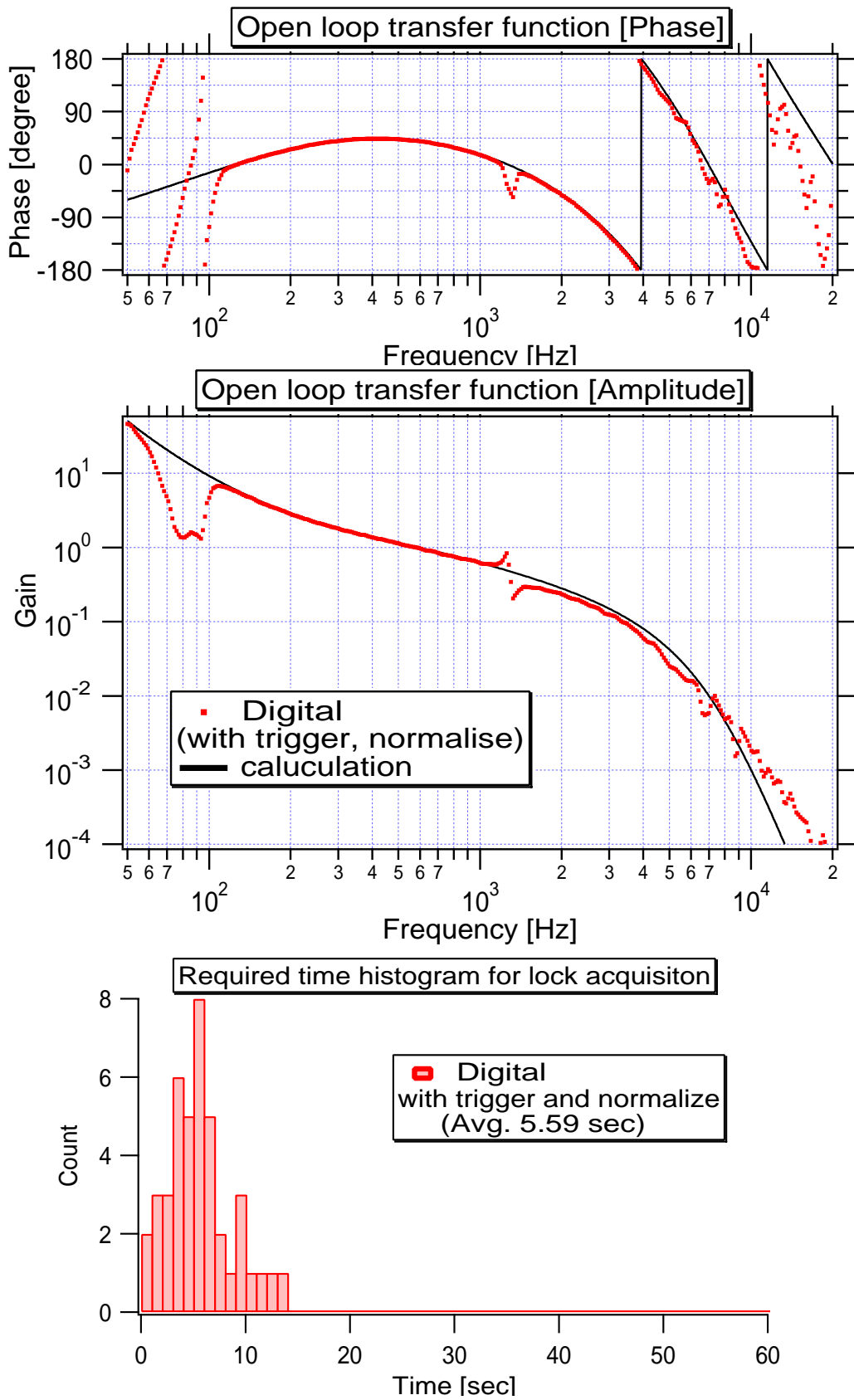


図 6.9: デジタルフィルタ (トリガー₆₅ 規格化付加) による lock acquisition

これらの測定結果からわかったことは以下ようになる。

- アナログフィルタと同等のものがデジタルフィルタにより再現できる。
- 透過光量による規格化は単独では効力を発揮しない。
- トリガーを加え feedback ゲインを上げるだけでも lock acquisition は改善する。
- トリガーに加え規格化の操作を加えると、より一層 lock acquisition が改善する。

ただ、lock acquisition というものは不定的な要素も多いため再現性を確かめる必要がある。そのため、別の日に同様な実験を行った結果を次のセクションで示すことにする。

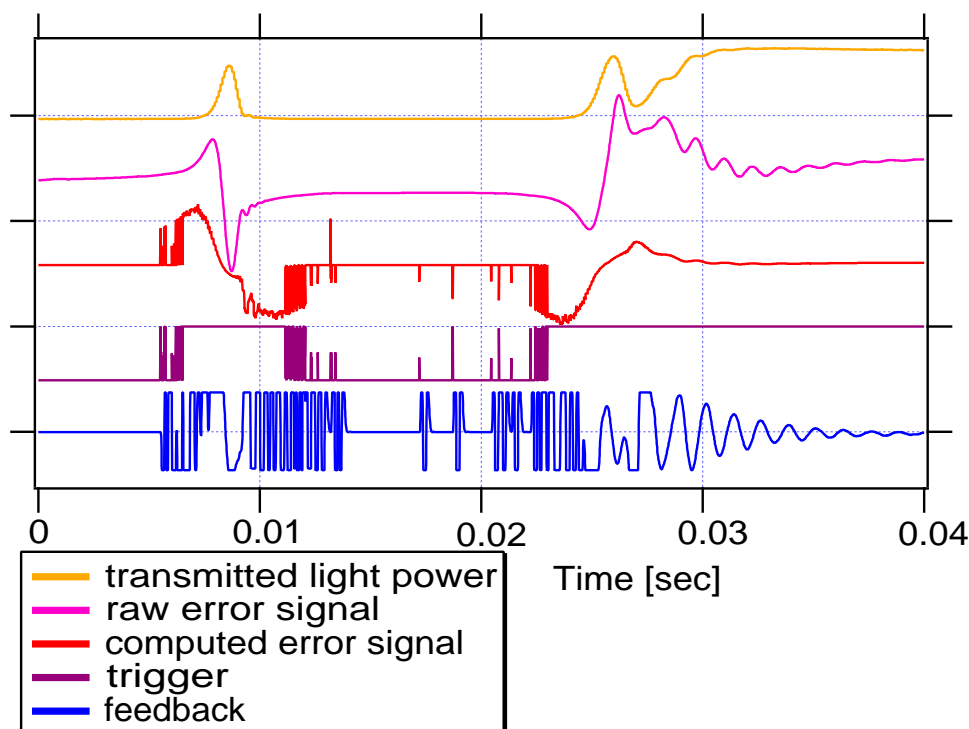


図 6.10: デジタルフィルタによる lock acquisition

6.3.3 デジタルフィルタによる lock acquisition

lock acquisition を行い、デジタルフィルタによる lock を成功させた時の各信号のグラフが図 6.10 である。上から順に透過光量のパワー、干渉計からの生の error signal、DSP で透過光量により規格化した error signal、トリガーのオンオフ、feedback でかけた電圧値となっている。規格化により線形域が広がっていることが確認できる。一つ目の error signal では lock に失敗しているが、その次の error signal により lock に成功している。

6.3.4 再現性

干渉計を lock させるとき、地面振動の影響を非常に受けやすくその地面振動は日や時間帯によっても異なるので同様の実験を別の日にも行った。複数日行った測定のある日の例として lock acquisition に要した時間のヒストグラムと平均時間が図 6.11 のようになる。

この日は地面振動が大きかったのか、アナログフィルタでは lock するまでに時間がかかったが、規格化とトリガーの演算を加えたデジタルフィルタは前回同様に平均 5 秒程度で lock することができた。

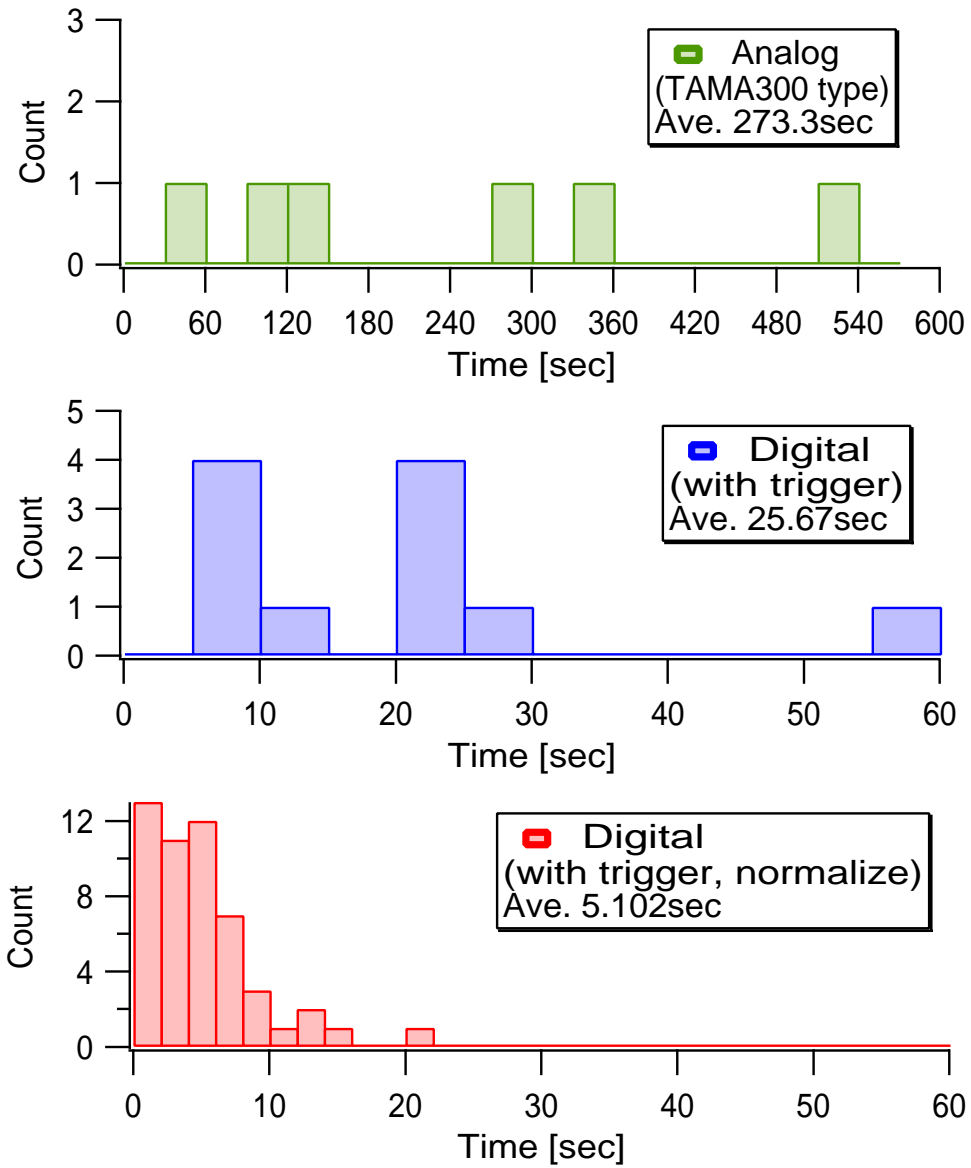


図 6.11: デジタルフィルタ (トリガー、規格化付加) による lock acquisition

6.4 デジタルフィルタによる lock acquisition

デジタルフィルタを用いた Fabry-Perot 共振器の mass による lock acquisition の実験から以下のことが得られた。

フィルタのプログラム プログラムはトリガーや規格化も含めて $3.47\mu\text{sec}$ で 1 演算を終えることが出来た。

デジタルフィルタによる lock acquisition 従来のアナログフィルタでは、lock までに 70sec かかっていたものが 5.6sec まで短縮することが出来た。また、再現性も確認できた。

今回の実験結果より、デジタルフィルタを用いることは干渉計制御において有効であると確認できた。

第7章 結論

TAMA300 における Fabry-Perot 共振器の鏡による長さ制御にデジタルフィルタを採用し、lock acquisition を改善することが出来た。

最後に結論として、本論文のテーマであるデジタル制御について

- 評価 ~ デジタルシステムの基本性能 ~
- 利用 ~ 規格化を用いた lock acquisition ~
- 今後 ~ 干渉計全体の制御に向けて ~

の3点についてまとめていく。

7.1 デジタルシステムの評価

まずデジタルシステムの評価として、フィルタを構成するのに必要な基本的スペックとしては、

- クロック周波数は 225MHz、サンプリング周波数は 200kHz
- 今回のシステムでは 22.0 μ sec の位相遅延
- ノイズレベルは 300nV/ \sqrt{Hz}
- ADC と DAC は 16bit 変換

のようであった。

これらは今回の実験から Fabry-Perot 共振器の lock acquisition に関しては問題ないということがわかった。

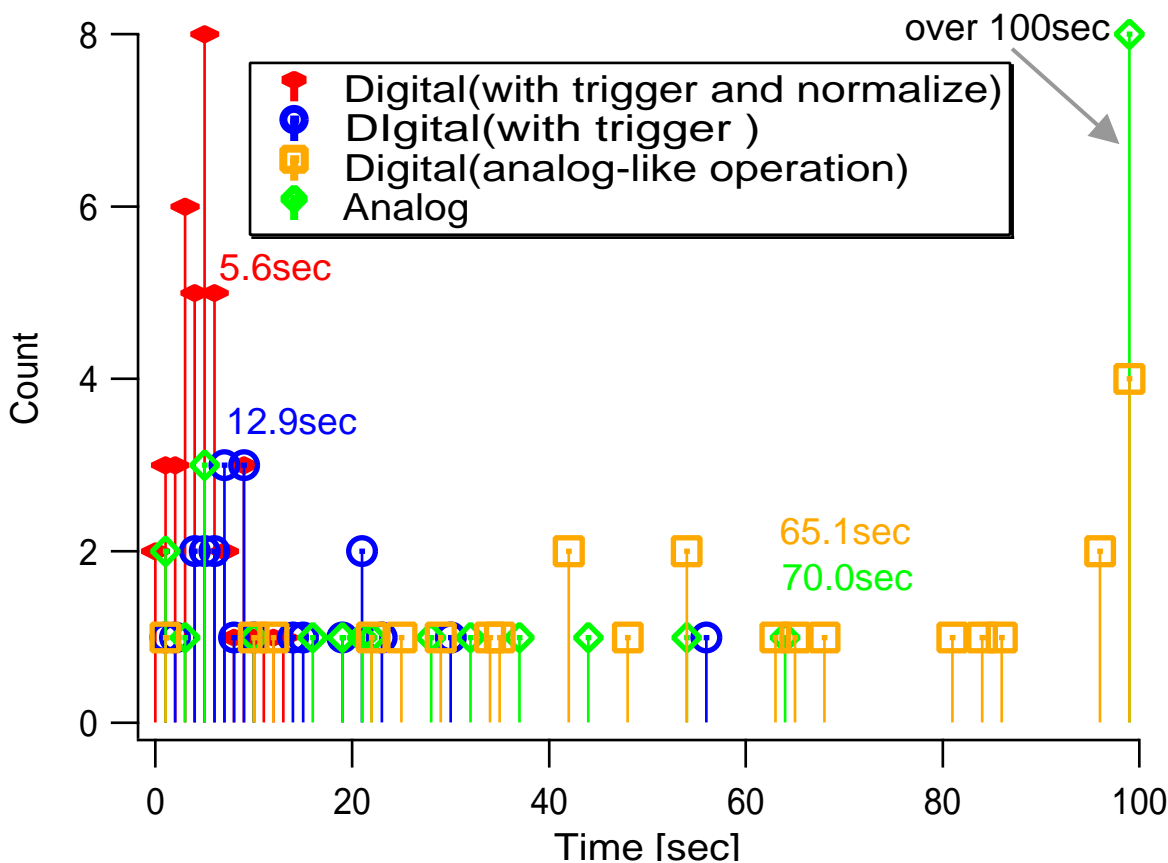


図 7.1: hstresult

7.2 規格化を用いた lock acquisition

デジタルシステムを用いて制御を行った Fabry-Perot 共振器の lock acquisition では規格化とトリガーを組み込むことにより、lock するまでの時間がアナログフィルタに比べ大幅に改善した。

アナログ (従来 TAMA300) ⇒ デジタル (規格化、トリガー)
 70.0sec ⇒ 5.6sec

その比較したヒストグラムは図 7.1 のようになる。

7.3 干渉計制御に際して

今回の実験では干渉計感度を測定するような実験は行わなかったため、観測時の制御においてはまだ課題が残されている。干渉計の感度曲線は数桁に及ぶため、ダイナ

ミックレンジの狭さはデジタルの弱点となる。また、現状ではプログラムを直接書き換えながらオフセットなどの調整を行っているため、ユーザーにとって使いやすいとは言えない。今後の課題としては、

- ホワイトニング、デホワイトニングの製作
- ユーザーが使いやすいシステムの構築

となってくる。

7.4 最後に

デジタル制御は、システムの組み方次第でデメリットも克服でき、非常に有効な道具であることが証明された。デジタル制御そのものが干渉計感度を向上できるかは未知であるが、それを道具として将来計画である km におよぶ大型干渉計 LCGT では

- ロスなしの伝送
- 規格化、多段階フィルタ等トリッキーな演算
- フィルタ変更の簡易化
- 制御情報の詳細なログ記録
- 使いやすいシステム

といったところで大いに貢献できる可能性を秘めており、TAMA300 へのインストールも含め研究を進めていく予定である。

参考文献

- [1] A.Einstein, 'Die Grundlage der allgemeinen Relativitätstheorie', Annalen der Physik, 49 (1916)
- [2] 中村卓史・三尾典克・大橋正健、重力波をとらえる, 京都大学出版会 (1998)
- [3] 三尾典克、大橋正健編; 重力波アンテナ技術検討書 (1992)
- [4] 河邊径太、'Fabry-Perot 型重力波検出器の制御', 東京大学修士論文 (1992)
- [5] Kuniharu Tochikubo, Development of 300-m Fabry-Perot cavity with automatic alignment control, doctor thesis(1999)
- [6] Shoemaker,D., et.al, 'Noise behavior of the Garching 30-m prototype gravitational-wave detector.', Phys. Rev., D38: 423-432, (1988)
- [7] Del,Fabro, et.al., 'Three-dimensional seismic super-attenuator for low frequency gravitational wave detection' Phys. Lett. A,124:253-257, (1987)
- [8] 新谷昌人、'Fabry-Perot 共振器のアラインメント制御', 東京大学修士論文 (1992)
- [9] <http://t-munu.phys.s.u-tokyo.ac.jp/>
- [10] Koji Arai, Recombine3: Suspension Transfer Functions, National Astronomical Observatory, TAMA project
- [11] 新井宏二、TAMA300 用マグネット・コイル・アクチュエータのカップリング測定、内部文書 (2005)
- [12] http://tamago.mtk.nao.ac.jp/tama/recom/www_suspension_install_status/coil_property.html
- [13] M.Evance, 'Lock Acquisition in Resonant Optical Interferometers', Doctor thesis, California Institute of Technology(2002)
- [14] 石田義久・鎌田弘之、デジタル信号処理のポイント Turbo C プログラムによる信号処理技術入門、産業図書 (1989)

- [15] 中村尚吾、ビギナーズデジタルフィルタ、東京電機大学出版局 (1989)
- [16] <http://www-ise2.ise.eng.osaka-u.ac.jp/iwanaga/study/index.html>
- [17] トランジスタ技術 2002.10 月号、CQ 出版社 (2002)
- [18] Texas Instruments、TMS320C6000 オプティマイジング (最適化) C/C++ コンパイラ
- [19] 三上直樹、C 言語によるデジタル信号処理入門、CQ 出版社 (2002)

付録A Feed-back rangeの変更

規格化とトリガーを用いたデジタルフィルタ（以降最適化フィルタと呼ぶ）による lock acquisition の改善を確かめることが出来たので、同じフィルタ設計で feedback のレンジを変えて lock acquisition に取り組む実験を行った。

この実験により現在用いられているコイルアクチュエータで、どれほどの力を加えればどれくらい lock するのかがわかる。

A.1 range 変更の設定

feedback レンジの変更はプログラム中で行った。最適化フィルタでは出力時に long のレンジを全て使った $-32768 \sim 32767$ でオーバーフロー対策のプログラムを組んでいたが、その値を変更した。測定した範囲は表 A.1 のようになる。最適化フィルタを基準に、おおよそ $1/2$ 、 $1/3$ 、 $1/5$ したフィルタで測定を行った。プログラム中でレンジを減らしたときに、実際 feedback ではどれくらいレンジが変わっているのかについても、生 feedback の peak-to-peak 電圧値をみることで確認した。

A.2 range による lock 成功率

レンジを変更したときの lock に要する時間のヒストグラムは図 A.1 のようになった。

ただし、レンジを $1/5$ にした時の結果は参考値である。このレンジの時は lock をしても、レンジが狭いために lock 後の安定化に対応しきれず 1 秒もたずに落ちてしまうことが多かった。少なくとも 1 秒以上 lock していた時を対象とした。また、長くても 10 秒程度しか lock を保ち続けることは出来なかった。

残りのレンジについては lock を保つことは出来た。レンジを狭めるにつれて lock acquisition の成功率が下がり、lock までに時間がかかるようになっていくことがわかる。

この測定結果から、feedback レンジを半分にしても lock acquisition に関しては、多少 lock までには時間がかかるがほとんど影響が無いと考えられる。また、feedback レンジを $1/5$ にしても lock だけは出来ることがわかった。

	オーバーフローレンジ (long)	feedback レンジ (V_{pp} [V])
最適化版	-32768 ~ 32767	17.5
×1/2	-16000 ~ 16000	8.95
×1/3	-10000 ~ 10000	5.63
×1/5	-6000 ~ 6000	3.75

表 A.1: feedback レンジの変更

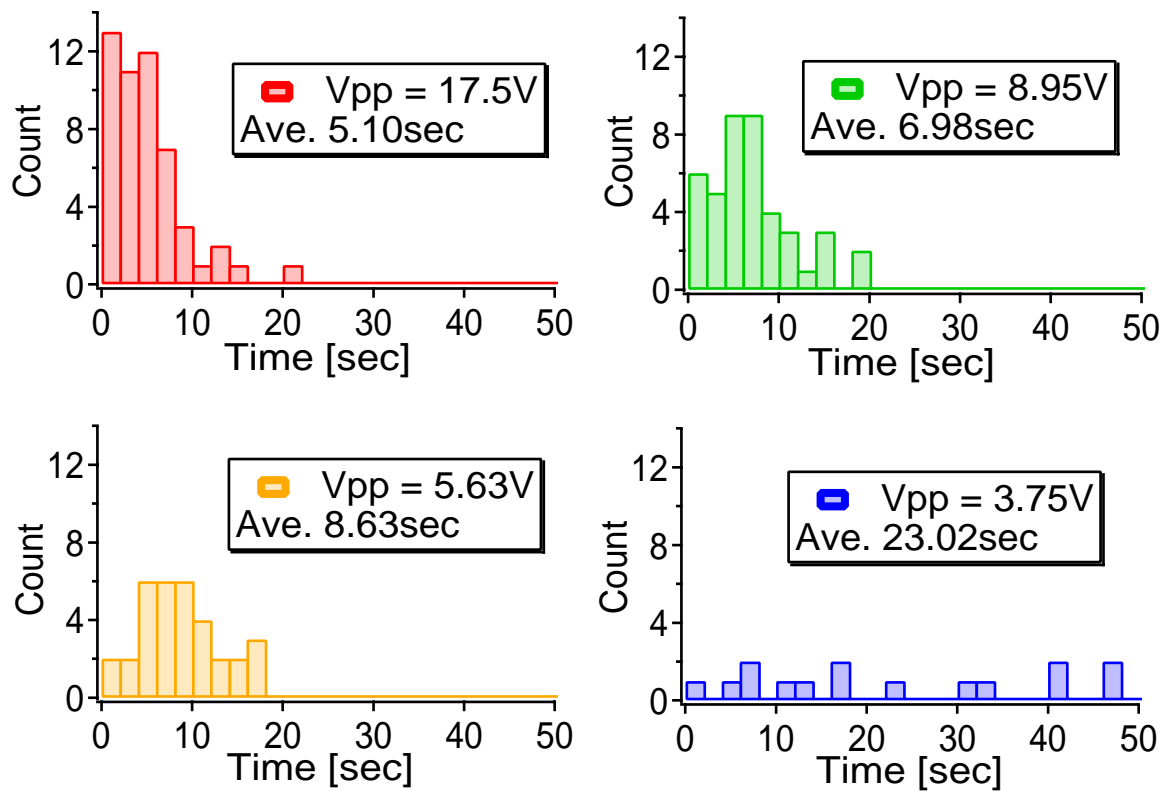


図 A.1: レンジ変更時の lock acquisition

付録B デジタルシステムに用いた ボード

今回デジタルシステムを構築するのに用いた DSP ボードと IO ボードについて書いておく。

B.1 TMS320C6713DSK

TMS320C6713 DSK はテキサスインストゥルメンツ社製 C67x™ シリーズ浮動小数点 DSP を搭載した DSP スターター・キットである。この DSK には、225 MHz で動作する TMS320C6713 DSP が搭載されており、McASP および IIC の評価を容易に行なうことが可能である。ボード上には DSP 以外に、SDRAM、フラッシュ・メモリ、ステレオ・コーデック等、アプリケーションを実装、評価する為の環境が一通り用意されている。また拡張コネクタを搭載しているで、ドータ・ボードによる拡張を容易に行なうことが可能である。USB 経由 JTAG コントロールによるエミュレーションを採用されており、PC との接続が容易で、ノート PC 等でも簡単に開発、評価を行なうことができる。実物は図 B.1 のようになる。

TMS320C6713 DSK の主な特徴をまとめておくと次のようになる。

- C6713 225MHz 搭載
- オンボード USB JTAG コントローラによるエミュレーション
- 8MB SDRAM、512KB フラッシュ・メモリ搭載
- ドータ・ボード拡張用コネクタ
- JTAG ヘッド
- +5V 動作

TMS320C6713 のチップの特徴をまとめておくと次のようになる。

- 225MHz の周波数動作で 1350MFLOPS
- IEEE 準拠の 32bit 単精度浮動小数点演算と 64bit 倍精度浮動小数点演算



図 B.1: TMS320C6713DSK

- 性能を落とさずに実行できるキャッシュ・ベース
 - 4KByte の一次データ・キャッシュ(L1D)(2WaySetAssociative)
 - 4KByte の一次プログラム・キャッシュ(L1P)(DirectMapped)
 - 192KB の L2 SRAM と 64KByte の L2 キャッシュ/SRAM(4バンク)(計 256KB)
- 2つのマルチチャンネル・オーディオ・シリアル・ポート (McASP)
- 16ステレオ・チャンネルのインター・ICサウンド (IIS)
- S/PDIF 通信プロトコルとの互換
- 2つのインター IC コントロール (IIC) ポート
- T1/E1、MVIP、IOM-2 などと直結可能な 2チャンネルのマルチチャンネル・バッファード・シリアルポート (McBSP)
- SBSRAM、SDRAM を直結
- 非同期メモリ (SRAM や FIFO)、ブート用の 8/16bitROM を接続可能
- 16チャンネルのエンハンスド DMA

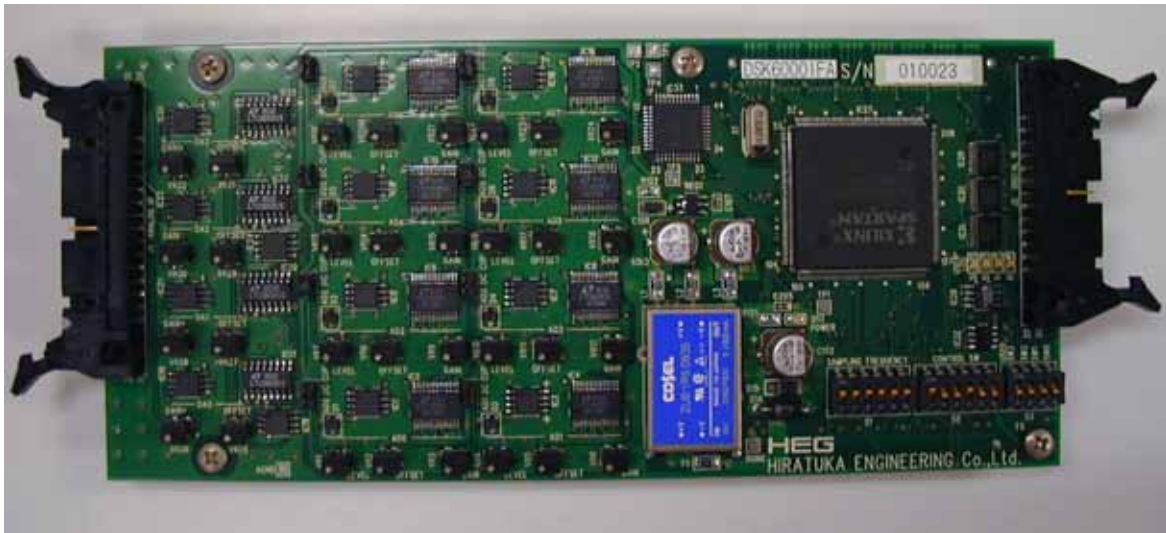


図 B.2: DSK6000IFA

B.2 DSK6000IFA

DSP多チャンネルアナログ入出力拡張ボード DSK6000IFA は、TI 製 DSK スタータキット (TI 製 DSK) に接続して、高精度な多チャンネルデジタル信号処理技術の研究、高機能な DSP アプリケーションの開発などに幅広く利用出来る IO ボードとなっている。ADC と DAC が搭載されており、サンプリング周波数の選択もこのボードから行う事が出来る。実物は図 B.2 のようになる。

DSK6000IFA の特徴をまとめておくと次のようになる。

- A/D:8 チャンネル同時サンプリング
- D/A:4 チャンネル同時変換
- 最高サンプリング周波数 200kHz、16bit
- 入出力レンジ $\pm 1V$
- デジタル外部入力、出力 : 8bit TTL レベル
- 電源: +5 (DSK より供給)

B.3 他デジタルシステム

参考に DSP 以外のデジタルシステムについてもしるしておく。

B.3.1 汎用プロセッサ

DSP が汎用プロセッサの特徴を取り入れる一方、一般の CPU にも乗算器を搭載しているものが増えてきた。例えばインテル社ではプロセッサのようなハードだけでなく、ソフトウェアも販売しており、その中にインテル・インテグレートッド・パフォーマンス・プリミティブ (インテル IPP) という名のインテル製プロセッサ向けの信号処理ライブラリがある。このコンパイラを使えば、複数の演算器を内蔵した Pentium の性能をフルに発揮でき、場合によれば DSP をしのぐ処理能力が得られる。ただし、今回行うような低処理遅延が要求される設計にはパソコンは向いていない。サンプリングが周波数が数 ~ 10 数 kHz 程度で済む機械系制御の処理は可能である。

B.3.2 FPGA や CPLD

集積度と速度が大幅に向上した FPGA や CPLD を使っても、デジタル信号処理システムを実現可能である。アプリケーションによっては DSP でソフト処理するよりも FPGA や CPLD でハードウェア化する方が効率的な場合もある。

ただ、現在は固定小数点演算であり、ダイナミックレンジの広い干渉計の制御には向かない。将来的にはどうなっていくかわからないので、考慮に入れておく必要がある。

付録C Fabry-Perot共振器ロック用 フィルタ

アナログフィルタは1つのアンプ、4種類7つのフィルタから構成されており、それぞれフィルタ部分の伝達関数は次のようになる。

$$T1 = \frac{R2}{R1(C0R2s + 1)} \quad (C.1)$$

$$T2 = \frac{1}{C1C2R3R4 \left(s^2 + \frac{\left(\frac{1}{R4} + \frac{1}{R5} + \frac{1}{R3} \right) s}{C1} + \frac{1}{C1C2R4R5} \right)} \quad (C.2)$$

$$T3 = \frac{\frac{1}{R6 + \frac{1}{C3s}} + \frac{1}{R7}}{C4s + \frac{1}{R8}} \quad (C.3)$$

$$T4 = \frac{1}{R9 \left(\frac{1}{R10 + \frac{1}{C5s}} + \frac{1}{R11} \right)} \quad (C.4)$$

よって、アナログフィルタの伝達関数 T_{analog} はロック前では、

$$\begin{aligned} T_{\text{analog}} &= T1 \times T2^2 \times T3^2 \\ &= \frac{\sum_{n=0}^2 b_s(n) \cdot s^n}{9} \\ &= \frac{\sum_{n=0} a_s(n) \cdot s^n}{9} \end{aligned} \quad (C.5)$$

となる。それぞれ抵抗とコンデンサーに回路中の値を代入することで、係数の値が求められ表C.1のようになる。ロック後では

$$\begin{aligned} T_{\text{analog}} &= T1 \times T2^2 \times T3^2 \times T4^2 \\ &= \frac{\sum_{n=0}^4 b_s(n) \cdot s^n}{11} \\ &= \frac{\sum_{n=0} a_s(n) \cdot s^n}{11} \end{aligned} \quad (C.6)$$

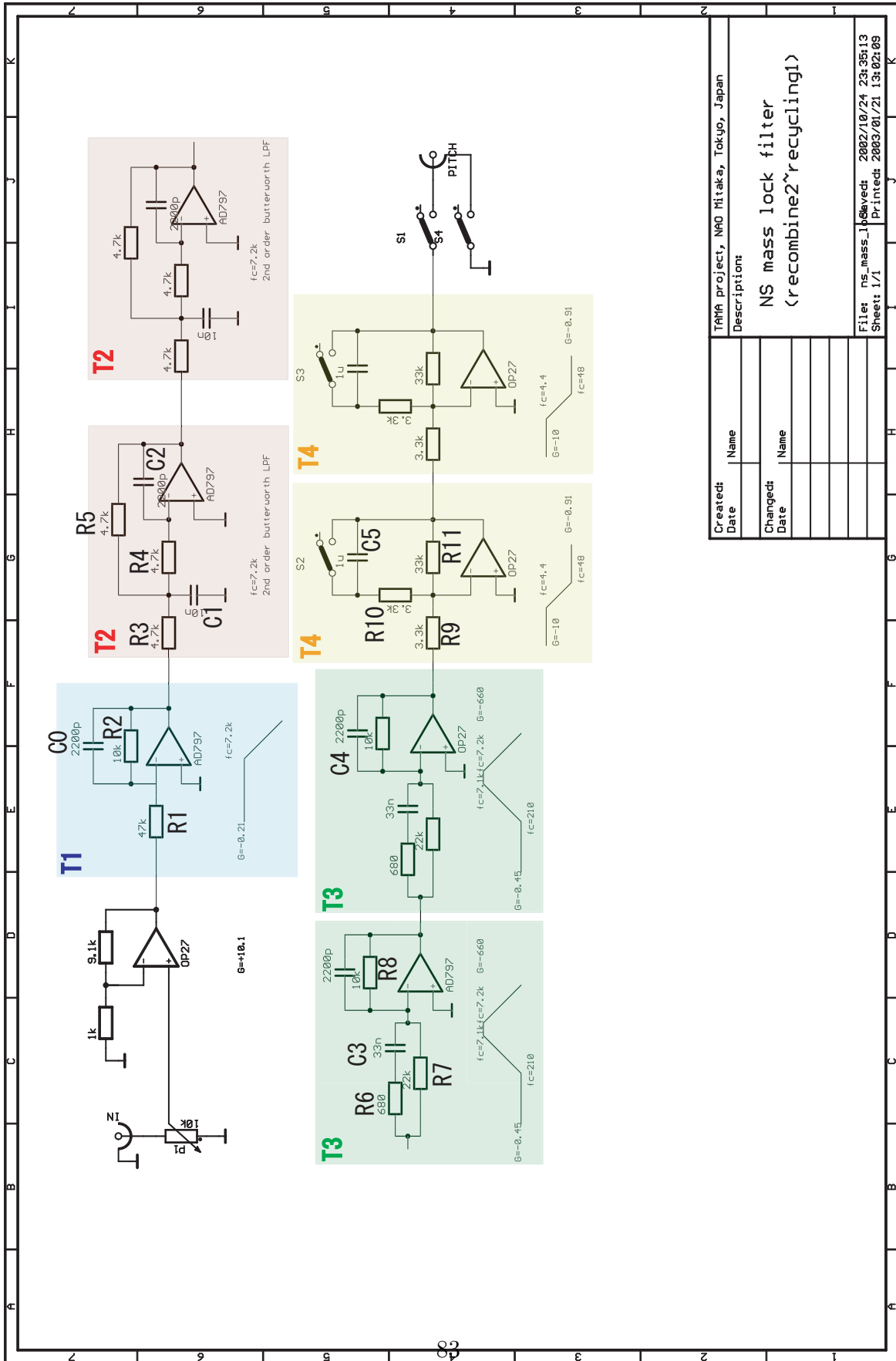
となる。それぞれ抵抗とコンデンサーに回路中の値を代入することで、係数の値が求められ表C.2のようになる。

n	$a_s(n)$	$b_s(n)$
0	$3.888081204972841 \times 10^{25}$	$1.7091969425764198 \times 10^{24}$
1	$6.723270019639035 \times 10^{22}$	$2.558462719403792 \times 10^{21}$
2	$5.338657894121227 \times 10^{27}$	$9.57427918855287 \times 10^{17}$
3	$2.561283702695126 \times 10^{13}$	
4	$8.190366689216031 \times 10^8$	
5	$1.8108619978290146 \times 10^4$	
6	$2.768483264016644 \times 10^{-1}$	
7	$2.821906415170263 \times 10^{-6}$	
8	$1.738780641134752 \times 10^{-11}$	
9	$4.9236352 \times 10^{-17}$	

表 C.1: ロック前アナログフィルタ伝達関数の係数

n	$a_s(n)$	$b_s(n)$
0	$3.5703225022707435 \times 10^{28}$	$1.5695105074163636 \times 10^{29}$
1	$2.5982279383194875 \times 10^{27}$	$1.2708138237289404 \times 10^{27}$
2	$4.749449081636532 \times 10^{25}$	$3.34769849142126 \times 10^{24}$
3	$8.170771295983939 \times 10^{21}$	$3.138722064164572 \times 10^{21}$
4	$6.476858797569198 \times 10^{17}$	$9.57427918855287 \times 10^{17}$
5	$3.104615187587429 \times 10^{13}$	
6	$9.922418649495478 \times 10^8$	
7	$2.1929889320106813 \times 10^4$	
8	$3.35174618007123 \times 10^{-1}$	
9	$3.415665994662553 \times 10^{-6}$	
10	$2.104252818119717 \times 10^{-11}$	
11	$5.95759859200000 \times 10^{-17}$	

表 C.2: ロック後アナログフィルタ伝達関数の係数



Created Date	Name
Changed Date	Name

TAMA project, NAO Mitaka, Tokyo, Japan

Description

NS mass lock filter
(recombine2~recycling1)

File: ns_mass_1

Sheet: 1/1

Released: 2002/10/24 23:35:13

Printed: 2003/01/21 13:02:09

図 C.1: Fabry-Perot 共振器ロック用フィルタ

(a_{1n}, a_{2n})	(b_{0n}, b_{1n}, b_{2n})
(0.8056852074, 0)	(1, 1.095069074, 0)
(1.581653030, -0.6254259562)	(1, 2.111187955, 1.119910158)
(1.599454328, -0.6396193565)	(1, 1.952492589, 0.9605736906)
(1.683871152, -0.7277703518)	(1, 1.841250384, 0.848876703)
(1.683955253, -0.727809165)	(1, -1.986683357, 0.9867276901)
(1.999724537, -0.9997245557)	(1, -1.996971989, 0.9969742812)

表 C.3: デジタルフィルタ伝達関数の係数

付録D 使用した各種プログラム

使用した各種プログラムを記しておく。

```
#define ORDER 12

double u[ORDER/2][2];          /* intermediate buffer */
const double am[ORDER/2-1][2] =
{
  { 0.8056852074,          0. },          /* a11, a21 */
  { 1.581653030,         -0.6254259562},    /* a12, a22 */
  { 1.599454328,         -0.6396193565},    /* a13, a23 */
  { 1.683871152,         -0.7277703518},    /* a14, a24 */
  { 1.683955253,         -0.727809165}];    /* a15, a25 */

const double bm[ORDER/2-1][3] =
{
  { 1.,          1.095069074,          0. }, /* b01, b11, b21 */
  { 1.,          2.111187955,          1.119910158}, /* b02, b12, b22 */
  { 1.,          1.952492589,          0.9605736906}, /* b03, b13, b23 */
  { 1.,          1.841250384,          0.848876703}, /* b04, b14, b24 */
  { 1.,          -1.986683357,          0.9867276901}]; /* b05, b15, b25 */

const double cm[ORDER/2][2] =
{
  { 0.8056852074,          0. },          /* a11, a21 */
  { 1.581653030,         -0.6254259562},    /* a12, a22 */
  { 1.599454328,         -0.6396193565},    /* a13, a23 */
  { 1.683871152,         -0.7277703518},    /* a14, a24 */
  { 1.683955253,         -0.727809165},    /* a15, a25 */
  { 1.999724537,         -0.9997245557}];    /* a16, a26 */

const double dm[ORDER/2][3] =
{
  { 1.,          1.095069074,          0. }, /* b01, b11, b21 */
  { 1.,          2.111187955,          1.119910158}, /* b02, b12, b22 */
  { 1.,          1.952492589,          0.9605736906}, /* b03, b13, b23 */
  { 1.,          1.841250384,          0.848876703}, /* b04, b14, b24 */
  { 1.,          -1.986683357,          0.9867276901}, /* b05, b15, b25 */
  { 1.,          -1.996971989,          0.9969742812}]; /* b06, b16, b26 */
```

図 D.1: 定数定義のプログラム

```

void main(void)
{
    /*define variables*/
    int k,m;

    /*buffer clear*/
    for (m=0; m<ORDER/2; m++)
    for (k=0; k<2; k++)
    {
        u[m][k] = 0.0;
    }
        :
        :
    for (;;)
    {
        /* Keep waiting for Timer0 Interrupt */
    }
}

```

図 D.2: メイン文のバッファクリアのプログラム

```

interrupt void int4(void)
{
    /*define variables*/
    double SI, LS;
    int m;
    short SIO;

    SIO = *(volatile short *)DSKIF_ADO; /*Signal Input*/
    SI = (double)SIO; /*modified Signal Input*/

    /******Error signal calculation******/
    if (LS<.3)/******Lock Before*****/
    {
        for (m=0; m<ORDER/2-1; m++)
        {
            un = am[m][0]*u[m][0] + am[m][1]*u[m][1] + SI;
            SI = bm[m][0]*un + bm[m][1]*u[m][0] + bm[m][2]*u[m][1];
            u[m][1] = u[m][0]; /* shift of data */
            u[m][0] = un; /* shift of data */
        }
        SI = SI*5.094971566e-006*32.;
    }

    else/******Lock After*****/
    {
        for (m=0; m<ORDER/2; m++)
        {
            un = cm[m][0]*u[m][0] + cm[m][1]*u[m][1] + SI;
            SI = dm[m][0]*un + dm[m][1]*u[m][0] + dm[m][2]*u[m][1];
            u[m][1] = u[m][0]; /* shift of data */
            u[m][0] = un; /* shift of data */
        }
        SI = SI*4.216521791e-006*10.;
    }

    :
    :
    :
}

```

図 D.3: ループ文の IIR フィルタ構成部

D.1 桁落ち現象検証用プログラム

桁落ち現象を確認した時のプログラムが図 D.4 のようになる。これはセクション 5.4.2 で実際に使ったプログラムである。測定する時は j の値を変えることで、そのプログラムのループを何回転するか決められる。

プログラム中で赤字のところは、桁落ちを検証するためのプログラムとなっている。ループを回り終えた後には最後に各ポイントでの値を表示するようにしてある。結果についてはセクション 5.4.2 を参照して欲しい。

D.2 演算時間測定用プログラム

演算時間を測定した時のプログラムが図 D.5 のようになる。プログラム中で赤字のところは、検証演算時間を測定するためのプログラムとなっている。 j の値を 1000000 とする事で、連続で 100 万回演算を行った時にかかった時間を表示するようにしてある。このプログラムでは、変数、定数共に `float` で定義してある。

D.3 規格化、トリガー組み込み済みデジタルフィルタプログラム

今回デジタルフィルタを用いた lock acquisition の実験で、規格化とトリガーを取り入れることにより lock までの時間を短縮することに成功したフィルタのプログラムを載せておく。図 D.6、図 D.7、図 D.8 を順につないだものがそのプログラムである。

```

#include <stdio.h>
#include "c6711dsk.h"
#include <c6x.h>
#include "c6xdskif.h"

#define ORDER 12

float u[ORDER/2][2];          /* intermediate buffer */

const double cm[ORDER/2-5][2] =
{ { 1.999724537,   -0.9997245557} /* a12, a22   G-up_filter */
};
const double dm[ORDER/2-5][3] =
{ { 1.,   -1.996971989,   0.9969742812} /* b02, b12, b22 G-up_filter */
};

void enableGlobalINT(void);      /* enable GIE          */
void enableNMI(void);           /* enable NMIE        */
void enableSpecificINT(int INTnumber); /* enable specific interrupt */

void main(void)
{
    double S1, un, S0;
    int j, m;

    S1=1.;
    un=0;
    u[0][0]=0;
    u[0][1]=0;

    for (j=0; j<100000; j++)
    {
        for (m=0; m<ORDER/2-5; m++)
        {
            un = cm[m][0]*u[m][0] + cm[m][1]*u[m][1] + S1;
            S0 = dm[m][0]*un + dm[m][1]*u[m][0] + dm[m][2]*u[m][1];
            u[m][1] = u[m][0]; /* shift of data */
            u[m][0] = un;      /* shift of data */
        }
        printf("un = %e , u0 =%e, u1 =%e , S0 =%e\n", (double)un, (double)u[0][0], (double)u[0][1], (double)S0)
    }
}

```

図 D.4: 桁落ち現象検証用プログラム

```

#include <stdio.h>
#include "c6711dsk.h"
#include "c6x.h"
#include "c6rdskf.h"
#include <time.h>

#define ORDER 12

double u[ORDER/2][2]; /* intermidiate buffer */
const double am[ORDER/2-1][2] =
{
  { 0.8056852074, 0. }, /* a11, a21
  { 1.581653030, -0.6254259562}, /* a12, a22
  { 1.59454328, -0.6396193565}, /* a13, a23
  { 1.683871152, -0.727703518}, /* a14, a24
  { 1.68395253, -0.727809165}}; /* a15, a25

const double bm[ORDER/2-1][3] =
{
  { 1., 1.095069074, 0. }, /* b01, b11, b21
  { 1., 2.11187955, 1.119910158}, /* b02, b12, b22
  { 1., 1.952492589, 0.9605738906}, /* b03, b13, b23
  { 1., 1.841250384, 0.848876703}, /* b04, b14, b24
  { 1., -1.986883357, 0.9967276901}}; /* b05, b15, b25

const double cm[ORDER/2][2] =
{
  { 0.8056852074, 0. }, /* a11, a21
  { 1.581653030, -0.6254259562}, /* a12, a22
  { 1.59454328, -0.6396193565}, /* a13, a23
  { 1.683871152, -0.727703518}, /* a14, a24
  { 1.68395253, -0.727809165}, /* a15, a25
  { 1.999724537, -0.9997245557}}; /* a16, a26

const double dm[ORDER/2][3] =
{
  { 1., 1.095069074, 0. }, /* b01, b11, b21
  { 1., 2.11187955, 1.119910158}, /* b02, b12, b22
  { 1., 1.952492589, 0.9605738906}, /* b03, b13, b23
  { 1., 1.841250384, 0.848876703}, /* b04, b14, b24
  { 1., -1.986883357, 0.9967276901}, /* b05, b15, b25
  { 1., -1.996971989, 0.9969742812}}; /* b06, b16, b26

void enableGlobalINT(void); /* enable GIE
void enableINT(void); /* enable IMIE
void enableSpecificINT(int INumber); /* enable specific interrupt */

void main(void)
{
  /*Define variables*/
  int k,m,j;
  clock_t t1,t2,t3;

  /*Program*/
  /*buffer clear*/
  for (m=0; m<ORDER/2; m++)
  for (k=0; k<K+1; k++)
  {
    u[m][k] = 0.0;
  }

  /*(unsigned volatile int *)EMIF_GE2 = 0x01200422; /* GE2 Bus timing control */

  enableSpecificINT(4); /* enable INT4(External INT)
  enableAMU(0); /* enable AMU (Ona, Maskable Interrupt) */
  enableGlobalINT(0); /* set GIE(Global Interrupt Enable) */

  t1=clock(0);
  for (j=0; j<1000000; j++)
  {
    /*define variables*/
    float S1, G, LS, Tr, Tg, un, cal;
    int m;
    short S10, LS0, G0, Tr0, So, DA1, DA2, DA3;

    S10 = *(volatile short *)DSKIF_A00; /*Signal Input*/
    LS0 = *(volatile short *)DSKIF_A01; /*Locked sign*/
    G0 = *(volatile short *)DSKIF_A02; /*Gain for feedback*/

```

```

Tr0 = *(volatile short *)DSKIF_A03; /*Transmitted light*/
S1 = (float)S10; /*modified Signal Input*/
G = (float)G0; /*modified Gain for feedback*/
LS = (float)LS0; /*modified Locked sign*/
Tr = (float)Tr0; /*modified Trs. light original0.015*/

DA3 = Tr * 8192; /*Trans cal. light monitor*/

/******Gain value modify (G -> 0<G<1)*****
if (G<0)
{
  G = 0. ;
}
else
{
  G = G+4.8828125e-4;
}

/******Trigger setting [Tr=Trigger]*****
if (Tr>0.03) /*0.005*/
{
  Tg = 1. ;
}
else
{
  Tg = 0. ;
}

DA2 = (short)Tg * 8192; /*Trigger monitor*/

/******Error signal calculation*****
if (LS<3) /*****Lock Before****
{
  cal = rcpdq(Tr);
  S1 = S1 * 0.1 * cal * Tg; /*only first BEST gain10 0912*/
  DA1 = (short)S1;
  for (m=0; m<ORDER/2-1; m++)
  {
    un = am[m][0]*u[m][0] + am[m][1]*u[m][1] + S1;
    S1 = bm[m][0]*un + bm[m][1]*u[m][0] + bm[m][2]*u[m][1];
    u[m][1] = u[m][0]; /* shift of data */
    u[m][0] = un; /* shift of data */
  }
  S1 = S1+4.216521791e-006*10. ;
}
else /*****Lock After****
{
  DA1 = (short)S1;
  for (m=0; m<ORDER/2; m++)
  {
    un = am[m][0]*u[m][0] + am[m][1]*u[m][1] + S1;
    S1 = dm[m][0]*un + dm[m][1]*u[m][0] + dm[m][2]*u[m][1];
    u[m][1] = u[m][0]; /* shift of data */
    u[m][0] = un; /* shift of data */
  }
  S1 = S1+4.216521791e-006*10. ;
}

/******Over flow problem*****
if(S1>32767) So = 32767;
else if (S1<-32768) So=-32768;
else So=(short)S1;

/* (unsigned volatile short *)DSKIF_DA0 = So; /*Signal Output*/
/* (unsigned volatile short *)DSKIF_DA1 = DA1; /*Output*/
/* (unsigned volatile short *)DSKIF_DA2 = DA2; /*Output*/
/* (unsigned volatile short *)DSKIF_DA3 = (short)DA3; /*Output*/

t2=clock(0);
printf("Time = %ld cycles %ld cycles %e sec\n", (long)t2, (long)t3, (float)t3/(CLOCKS_PER_SEC);
}

```

演算メイン文

図 D.5: 演算時間測定用プログラム

```

#include <stdio.h>
#include "c6711dsk.h"
#include <c6x.h>
#include "c6xdskif.h"
#include <time.h>
#include "fastrts67x.h"

#define ORDER 12

double u[ORDER/2][2]; /* intermediate buffer */
const double am[ORDER/2-1][2] =
{
  { 0.8056852074, 0. }, /* a11, a21 */
  { 1.581653030, -0.6254259562}, /* a12, a22 */
  { 1.599454328, -0.6396193565}, /* a13, a23 */
  { 1.683871152, -0.7277703518}, /* a14, a24 */
  { 1.683955253, -0.727809165}}; /* a15, a25 */
const double bm[ORDER/2-1][3] =
{
  { 1., 1.095069074, 0. }, /* b01, b11, b21 */
  { 1., 2.111187955, 1.119910158}, /* b02, b12, b22 */
  { 1., 1.952492589, 0.9605736906}, /* b03, b13, b23 */
  { 1., 1.841250384, 0.848876703}, /* b04, b14, b24 */
  { 1., -1.986683357, 0.9867276901}}; /* b05, b15, b25 */
const double cm[ORDER/2][2] =
{
  { 0.8056852074, 0. }, /* a11, a21 */
  { 1.581653030, -0.6254259562}, /* a12, a22 */
  { 1.599454328, -0.6396193565}, /* a13, a23 */
  { 1.683871152, -0.7277703518}, /* a14, a24 */
  { 1.683955253, -0.727809165}, /* a15, a25 */
  { 1.999724537, -0.9997245557}}; /* a16, a26 */
const double dm[ORDER/2][3] =
{
  { 1., 1.095069074, 0. }, /* b01, b11, b21 */
  { 1., 2.111187955, 1.119910158}, /* b02, b12, b22 */
  { 1., 1.952492589, 0.9605736906}, /* b03, b13, b23 */
  { 1., 1.841250384, 0.848876703}, /* b04, b14, b24 */
  { 1., -1.986683357, 0.9867276901}, /* b05, b15, b25 */
  { 1., -1.996971989, 0.9969742812}}; /* b06, b16, b26 */

void enableGlobalINT(void); /* enable GIE */
void enableNMI(void); /* enable NMIE */
void enableSpecificINT(int INTnumber); /* enable specific interrupt */

void main(void)
{
  /*define variables*/
  int k, m;

  /*program*/
  /*buffer clear*/
  for (m=0; m<ORDER/2; m++)
  for (k=0; k<2; k++)
  {
    u[m][k] = 0.0;
  }

  *(unsigned volatile int *)EMIF_CE2 = 0x0120C422; /* CE2 Bus timing control */

  enableSpecificINT(4); /* enable INT4(External INT) */
  enableNMI(); /* enable NMI(Non Maskable Interrupt)*/
  enableGlobalINT(); /* set GIE(Global Interrupt Enable) */

  for (;;)
  {
    /* Keep waiting for Timer0 Interrupt */
  }
}

```

図 D.6: 規格化、トリガー組み込み済みデジタルフィルタプログラム (その1)

```

interrupt void int4(void)
{
    /*define variables*/
    double S1, G, LS, Tr, Tg, un, cal;
    int m;
    short S10, LS0, G0, Tr0, So, DA1, DA2, DA3;

    S10 = *(volatile short *)DSKIF_A0; /* Signal Input */
    LS0 = *(volatile short *)DSKIF_AD1; /* Locked sign */
    G0 = *(volatile short *)DSKIF_AD2; /* Gain for feedback */
    Tr0 = *(volatile short *)DSKIF_AD3; /* Transmitted light */
    S1 = (double)S10; /* modified Signal Input */
    G = (double)G0; /* modified Gain for feedback*/
    LS = (double)LS0*3.051757e-5; /* modified Locked sign */
    Tr = (double)Tr0*3.051757e-5 + 0.1; /* modified Trs. light */

    DA3 = Tr * 8192 ; /*Trans cal. light monitor*/

    /******Gain value modify (G -> 0<G<1)******/
    if (G<0)
    {
        G = 0. ;
    }
    else
    {
        G = G*4.8828125e-4;
    }

    /******Triger setting [Tg=Triger]******/
    if (Tr>0.03) /*decide threshold of trigger*/
    {
        Tg = 1. ;
    }
    else
    {
        Tg = 0. ;
    }

    DA2 = (short)Tg * 8192 ; /*Triger monitor*/
}

```

図 D.7: 規格化、トリガー組み込み済みデジタルフィルタプログラム (その2)

```

/*****Error signal calculation*****/
if (LS<.3)/****Lock Before****/
{
cal = _rcpdp(Tr);
SI = SI * 0.1 * cal * Tg;      /*normalize and trigger*/

    DA1 = (short)SI ;
for (m=0; m<ORDER/2-1; m++)
    {
        un = am[m][0]*u[m][0] + am[m][1]*u[m][1] + SI;
        SI = bm[m][0]*un + bm[m][1]*u[m][0] + bm[m][2]*u[m][1];
        u[m][1] = u[m][0];      /* shift of data */
        u[m][0] = un;          /* shift of data */
    }
    SI = SI*5.094971566e-006*32. ;
}

else/****Lock After****/
{
    DA1 = (short)SI ;
for (m=0; m<ORDER/2; m++)
    {
        un = cm[m][0]*u[m][0] + cm[m][1]*u[m][1] + SI;
        SI = dm[m][0]*un + dm[m][1]*u[m][0] + dm[m][2]*u[m][1];
        u[m][1] = u[m][0];      /* shift of data */
        u[m][0] = un;          /* shift of data */
    }
    SI = SI*4.216521791e-006*10. ;
}

/*****Over flow problem*****/

SI= SI*G;

if(SI>=32767) So = 32767;
else if (SI<=-32768) So=-32768;
else So=(short)SI;

/*****Out put data*****/

*(unsigned volatile short *)DSKIF_DA0 = So;      /* Signal Output */
*(unsigned volatile short *)DSKIF_DA1 = DA1;     /* Normalized, triggered input*/
*(unsigned volatile short *)DSKIF_DA2 = DA2;     /* Triger monitor Output */
*(unsigned volatile short *)DSKIF_DA3 = (short)DA3; /* Modified transmitted light */
}

```

図 D.8: 規格化、トリガー組み込み済みデジタルフィルタプログラム (その3)

謝辞

本論文の実験は、多くの方々の御指南、助言、協力のおかげでここまで仕上げることが出来ました。大変感謝しております。

指導教官である黒田和明教授には、国内最大である干渉計 TAMA300 で研究する機会を与えて頂き、今年度は TAMA300 のある国立天文台に移って研究を行いたいと言った時にも快く了承して頂きました。実際の重力波検出器に触れ、開発する機会を与えて頂いたことに非常に感謝しています。

国立天文台教授の藤本眞克氏には天文台への受け入れ手続きや、DSP 寄贈の際にもお世話になりました。

同助教授川村静児氏にはデジタルフィルタでロックする過程における鋭い指摘を頂き、デジタルフィルタによる演算と lock acquisition における理解が深まりました。

同主任研究員の新井宏二氏には、実験以前のことから最後に至るまで本当にお世話になりました。本論文のテーマであるデジタル制御の提案から、プログラミング、計算機の使い方、難解複雑な TAMA300 のしくみ、干渉計の世界情勢、他にも膨大な知識を与えて頂きました。自分では扱いきれない TAMA300 を制御するために深夜、時には日の出まで実験に付き合ってもらったことは2度や3度ではありません。研究に対する姿勢を学ばして頂きました。

同主任研究員の辰巳大輔氏にはC言語でつまづいた時にはお世話になりました。考え方が偏ってしまっている時にも新たな方向性を示して頂きました。

同主任研究員の高橋竜太郎氏にはデジタルフィルタ設計の際、デジタル特有の問題に関して教えて頂きました。

同研究員の佐藤修一氏には lock acquisition に関する指摘を頂き、考えをまとめることが出来ました。

東京大学理学部の阿久津智忠氏には、天文台での先輩として TAMA300 に関する情報をいろいろと教わりました。

東京大学宇宙線研究所重力波グループの方々にも研究の基礎や、干渉計についていろいろと教わりました。

ここには書ききれませんが、研究会や会議等でいろいろと指摘して頂いた重力波グループの方々に感謝しております。

また本実験において、日本テキサスインスツルメンツ社からは、DSP スターターキット、およびコードコンポーザースタジオの提供をして頂きありがとうございました。

最後になりましたが、研究を行うことを理解して、応援してくれた親族や友人に感謝いたします。ありがとうございます。