TOKYO TECH
Pursuing Excellence

アナログ・ミクストシグナル回路設計とシミュレーション技術

松澤　昭

東京工業大学

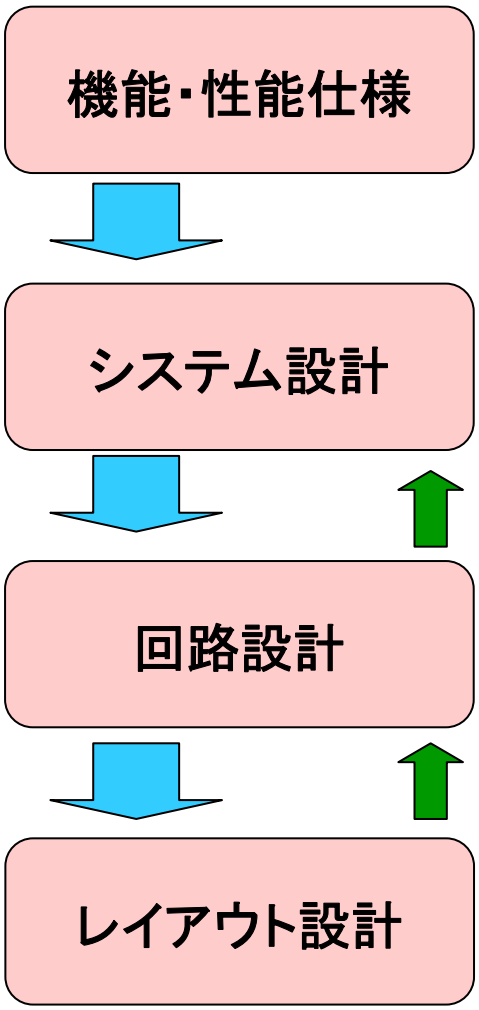
内容

TOKYO TECH Pursuing Excellence

- アナログ回路設計の概要
- パイプライン型ADCの設計
  - 性能と回路仕様
  - 性能関数
  - シミュレーション技術
- ΔΣ型ADCの設計
  - システムレベルの検討
  - トランジスタレベルの検討
- まとめ

最新資料は松澤・岡田研究室ホームページ
パブリケーションからダウンロードしてください
http://www.ssc.pe.titech.ac.jp

アナログ回路設計の流れ



作業内容

- 要求される機能
- 要求される性能

- アーキテクチャの検討
- 安定性の検討
- トレードオフの検討
- 回路形式の検討

TRキャラクタライズ

- トランジスタレベルの設計
- PVT検討
- バラツキ・ノイズ検討
- インターフェース検討

配線キャラクタライズ

- レイアウト制約
- LPE
- ポストレイアウトSim

ポイント

- インターフェースの明確化
- プロセスの決定
- デバイスパラメータの明確化
- 実装技術の明確化

- システムとしての妥当性
- トレードオフの許容度
- 回路が成り立つか？
- デジタルアシストの適用？

- レイアウト情報の組込み
- テストベンチ

- 制約の明確化
- LPE精度
- Sim時間

機能・性能仕様

TOKYO TECH Pursuing Excellence

- 分解能・直線性
  - 実効分解能
  - INL DNL (どちらを重視)
  - SNR, DFDR, SNDR (どれを重視？)
- 変換周波数
  - 固定か可変か？
- 信号帯域
  - ERB
- 電源電圧
- PVTマージン
- 消費電力
- 入力信号電圧範囲
- シングル・差動
- 入力インピーダンス
  - 容量
  - 抵抗(周波数依存)
- クロックレイテンシー
- I/O形式
- PSRR
- デザインルール
- 面積
- オプションプロセス
  - MIMが可能か？

ADCの開発でもこれぐらいの仕様が必要

カスタマーに聞いても全ては答えられない

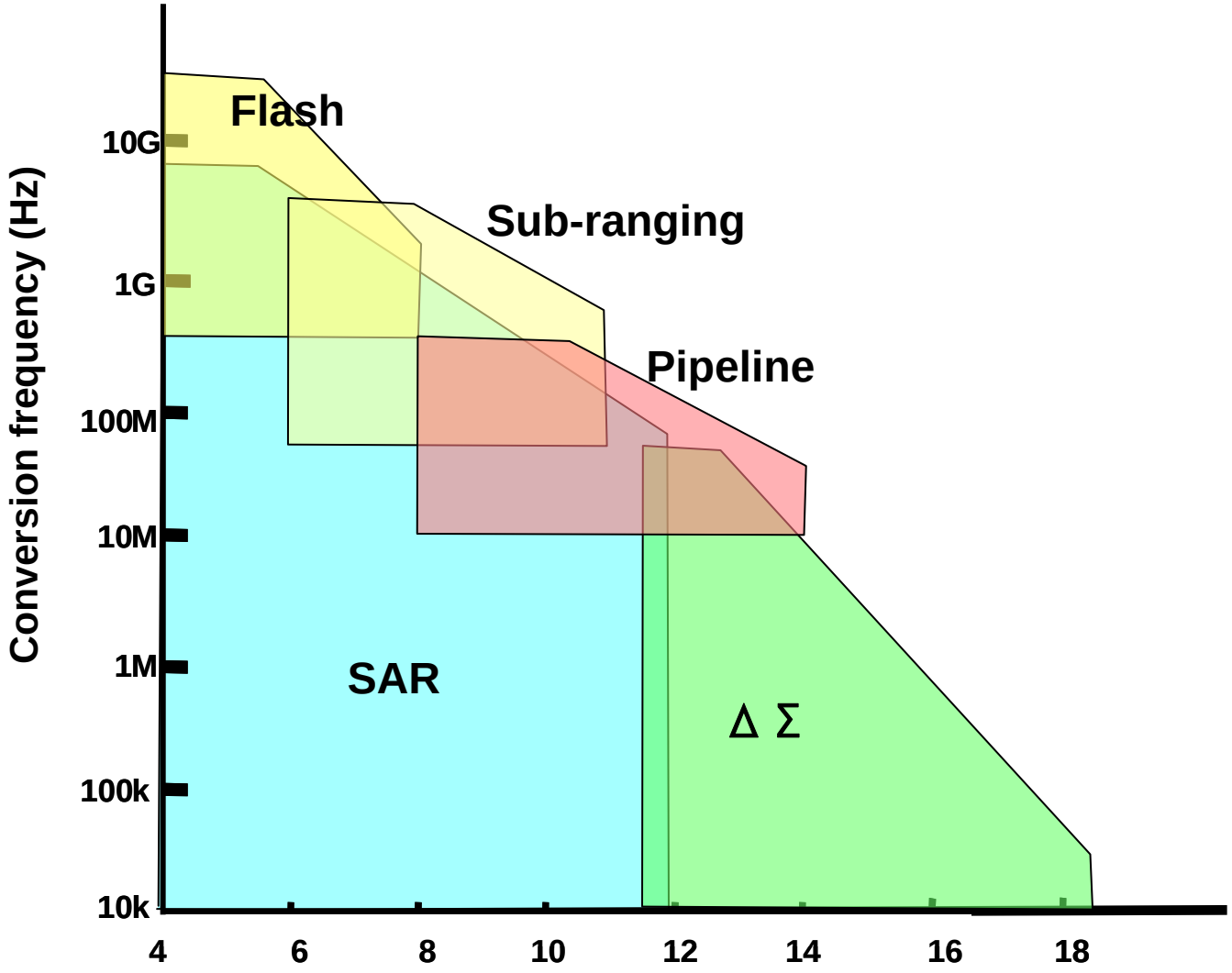
実際は回路設計者のアドバイスが必要

これら仕様により変換形式まで変わる

ADCの分解能・変換周波数・変換方式

TOKYO TECH Pursuing Excellence

ADCは分解能･変換周波数･変換方式により最適な変換方式がある。



各A/D変換方式の特徴

TOKYO TECH Pursuing Excellence

各変換方式で設計の重要ポイントが変わる

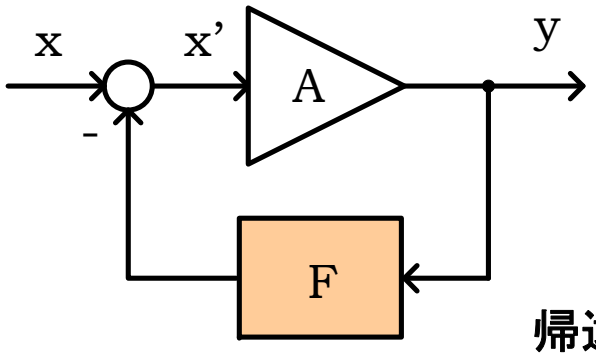
	比較器	OP アンプ	容量 DAC	電流 DAC	フィードバック系	微細化 低電圧化	設計ポイント
Flash	◎					◎	・比較器のオフセット ・クロック・ロジック
SAR	○		◎			◎	・容量DACの精度 ・比較器のノイズ
Pipe		◎	◎		○（部分）	×	・容量DACの精度 ・OPアンプ性能
Δ Σ		○	○	◎	◎（部分全体）	△	・DACの精度 ・帰還回路の設定 ・OPアンプノイズ

Matsuzawa & Okada Lab.

負帰還技術

負帰還回路は高精度受動素子による帰還回路と高利得増幅器を用いる。
入出力特性が帰還回路だけで決まるようにできるので、高精度である。

高利得増幅器



帰還回路

$$y = A(x - Fy)$$

$$\therefore y = \frac{A}{1+AF}x$$

$$y = \frac{1}{F}x \quad (A \to \infty)$$

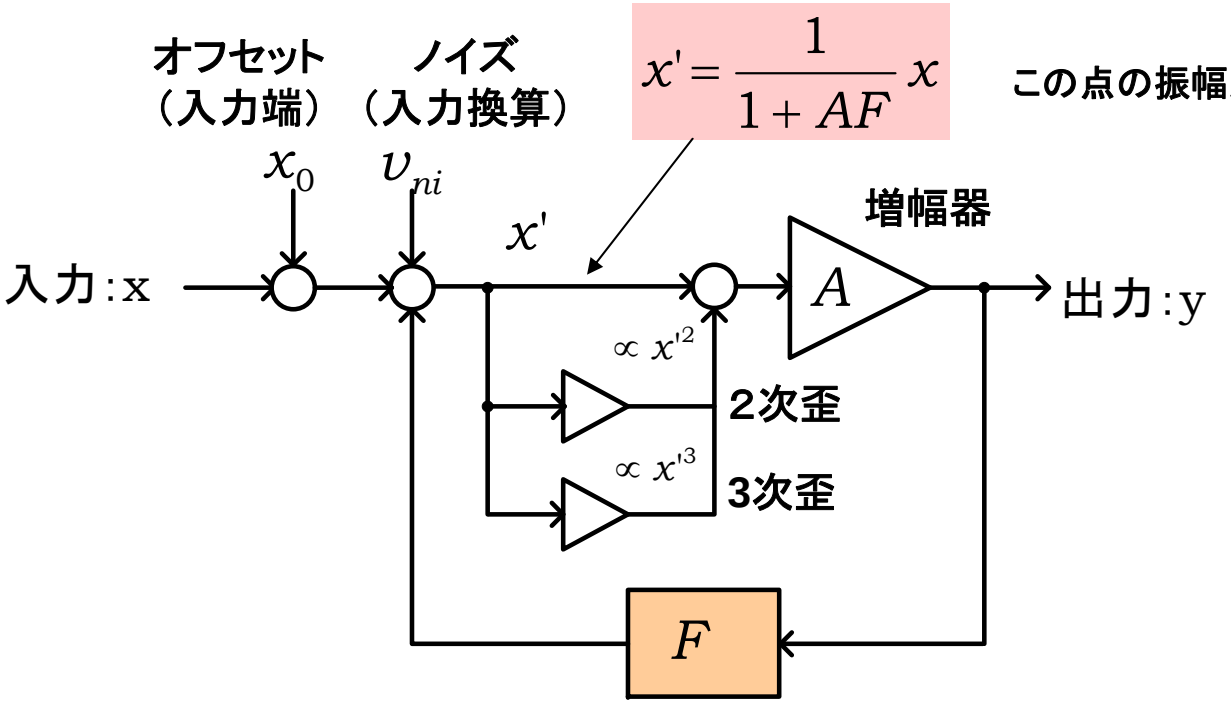
$$x' = \frac{1}{1+AF}x$$

負帰還技術における特性

TOKYO TECH
Pursuing Excellence

負帰還技術は歪み（非直線性）の抑制に効果的である

ノイズ：入力換算ノイズは減少しない
オフセット電圧：入力換算オフセット電圧は減少しない
歪み：増幅器に入力される信号振幅が低化するので減少する



安定か発振か？

負帰還をかけると不安定になり発振に至ることもあるので、十分なチェックが必要。

ポールの位置により応答の形が決まる。
ポールが左反面（実数部が負）で安定。右反面（実数部が正）で発振。
虚数部を持つと振動成分が発生する。

$$H(s) = \frac{A(s)}{1 + A(s)F(s)}$$

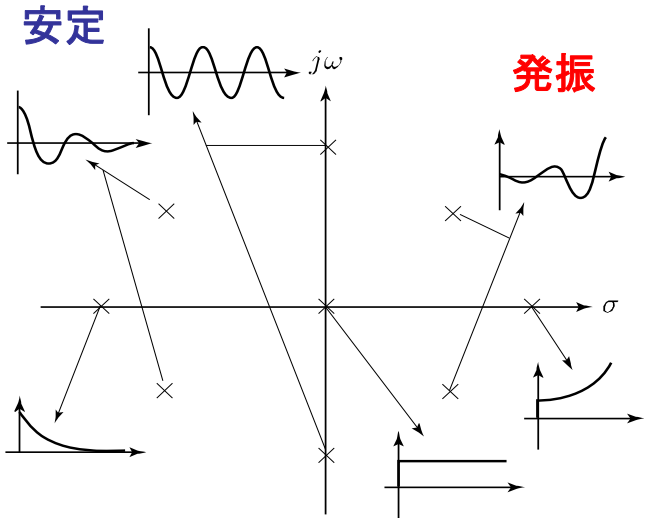
$$H(s) = K\frac{(s - z_1)(s - z_1).....(s - z_m)}{(s - p_1)(s - p_1).....(s - p_n)}$$

$$y(t) = \sum_{i=1}^{n} k_i \exp(p_i t)$$

$$\exp(p_i t) = \exp((\sigma_i + j\omega_i)t) = \underbrace{\exp\sigma_i t}_{\text{利得の項}} \cdot \underbrace{\exp(j\omega_i t)}_{\text{振動の項}}$$

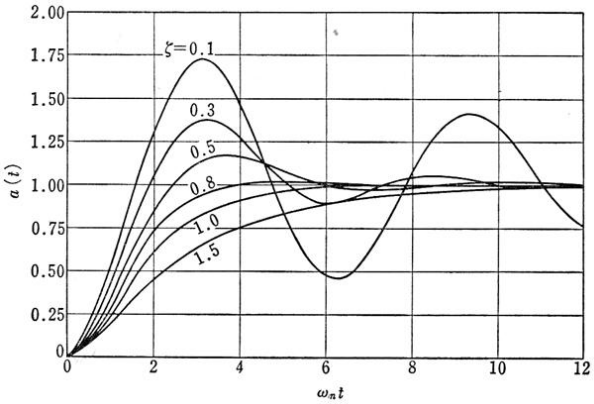
2次系のとき

$$H(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$



$\zeta > 1$: 虚数成分は発生しない → 振動しない

$\zeta < 1$: 虚数成分が発生 → 振動する



離散時間系：Z平面

標本化とはある周波数に対する位相回転作用であることを表している

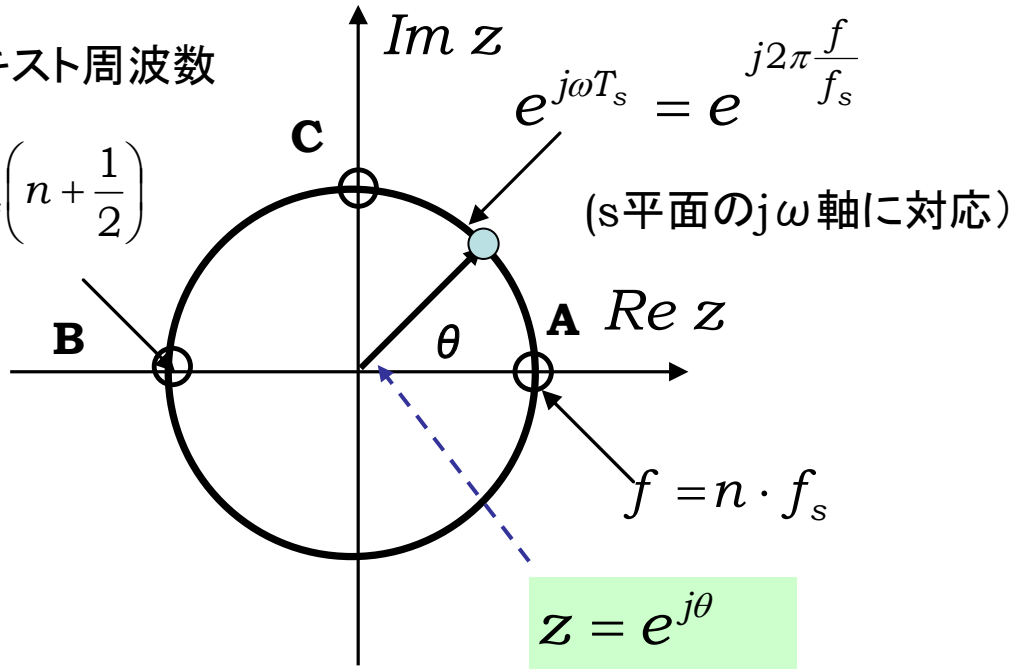
$z = e^{sT_s}$ において

$s = j\omega$ を代入すると、

$z = e^{j\omega T_s}$

z平面上で単位円の軌跡を表している

$$z = e^{j2\pi \frac{f}{f_s}}$$



$$z = e^{(\sigma + j\omega)T_s} = e^{\sigma T_s} e^{j\omega T_s} = e^{\sigma T_s}\left(\cos \omega T_s + j \sin \omega T_s\right)$$

$\sigma > 1$: 単位円の外側→ 発散

$\sigma < 1$: 単位円の内側→ 収束

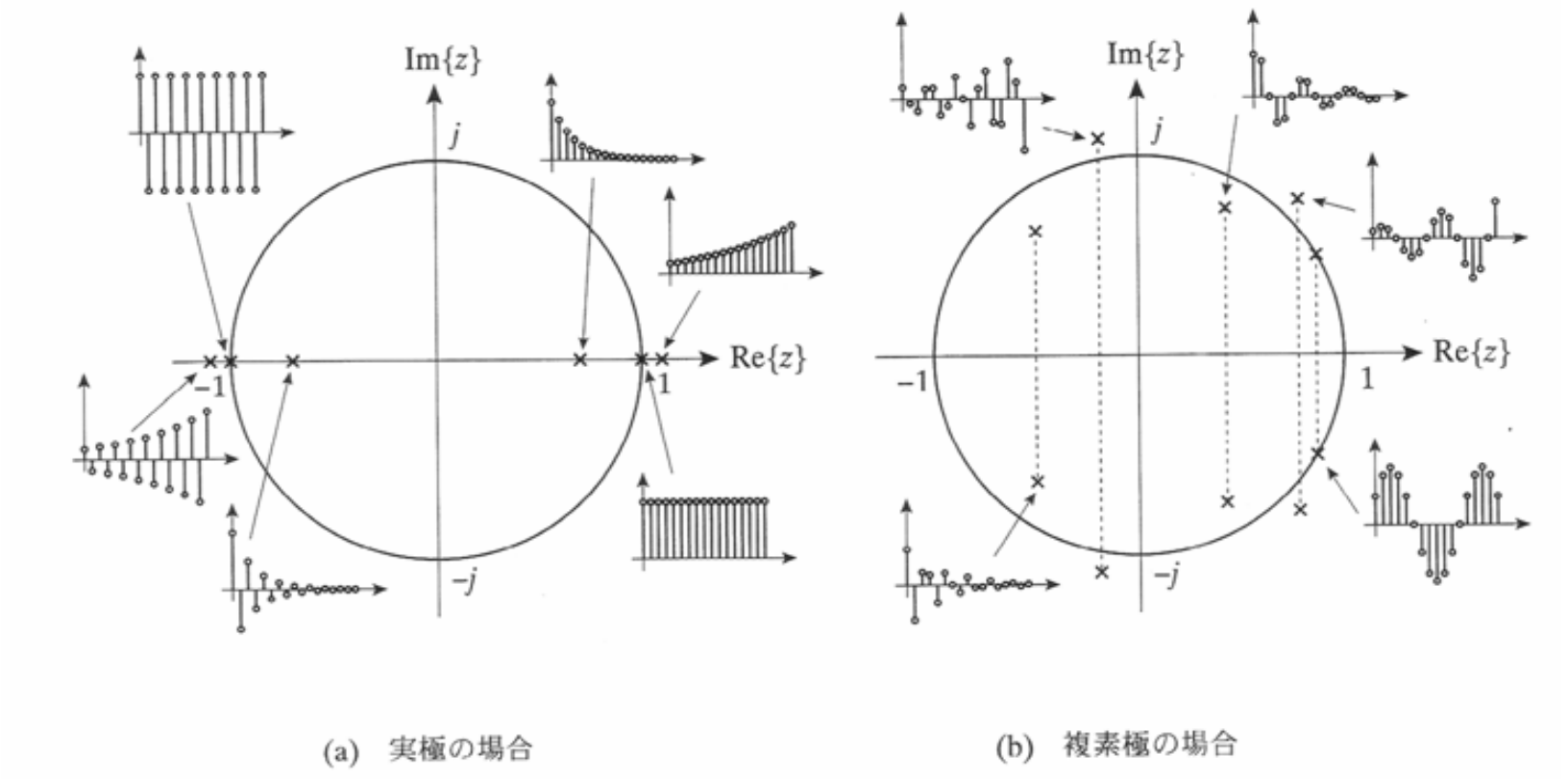
$$z = e^{j\theta}$$

$$\theta = 2\pi \frac{f}{f_s}$$

極の位置とシステムの安定

TOKYO TECH Pursuing Excellence

ポールが単位円の内側にあれば安定、外側では不安定である。



(a) 実極の場合　　(b) 複素極の場合

「デジタル信号処理の基礎」 三上直樹　CQ出版

TOKYO TECH
Pursuing Excellence

パイプライン型ADCの設計

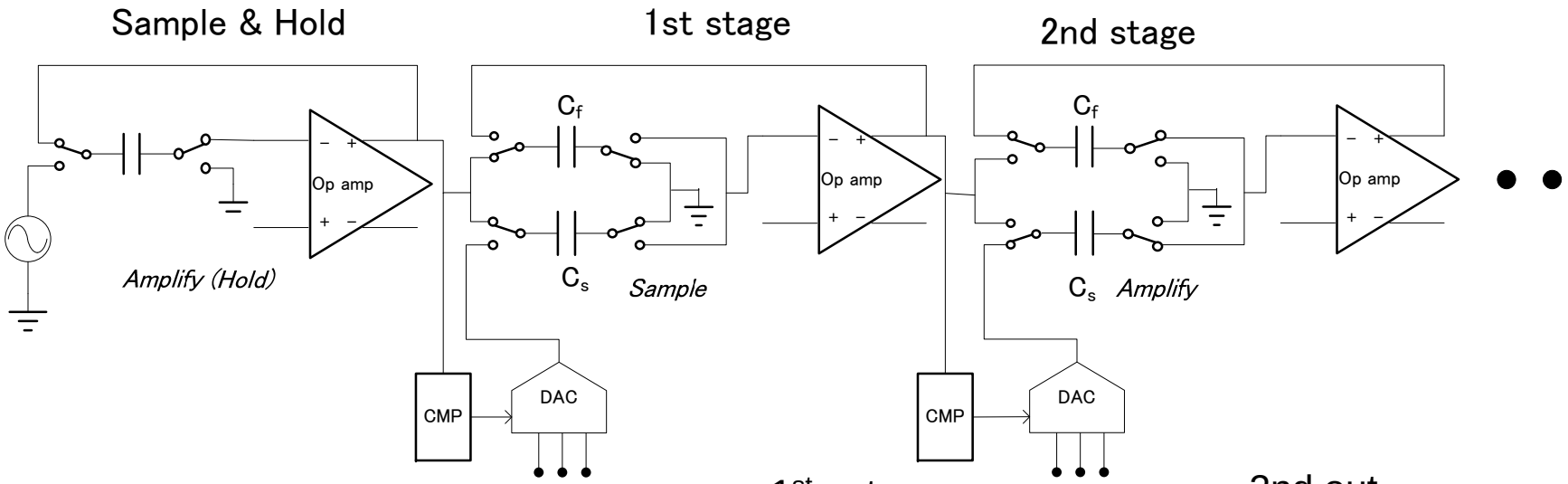
Matsuzawa & Okada Lab.

パイプライン型ADCの基本構成

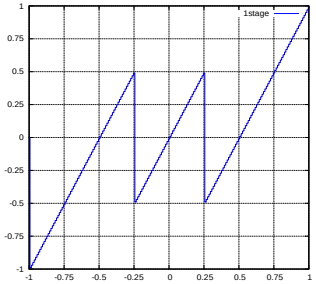
TOKYO TECH Pursuing Excellence

パイプライン型ADCは
- 標本化
- 電圧比較(ADC)
- 比較結果に応じたDAC電圧設定
- 増幅(通常2倍)

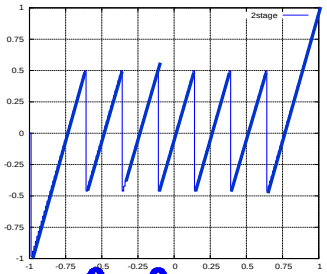
をパイプライン的に行う



1st out

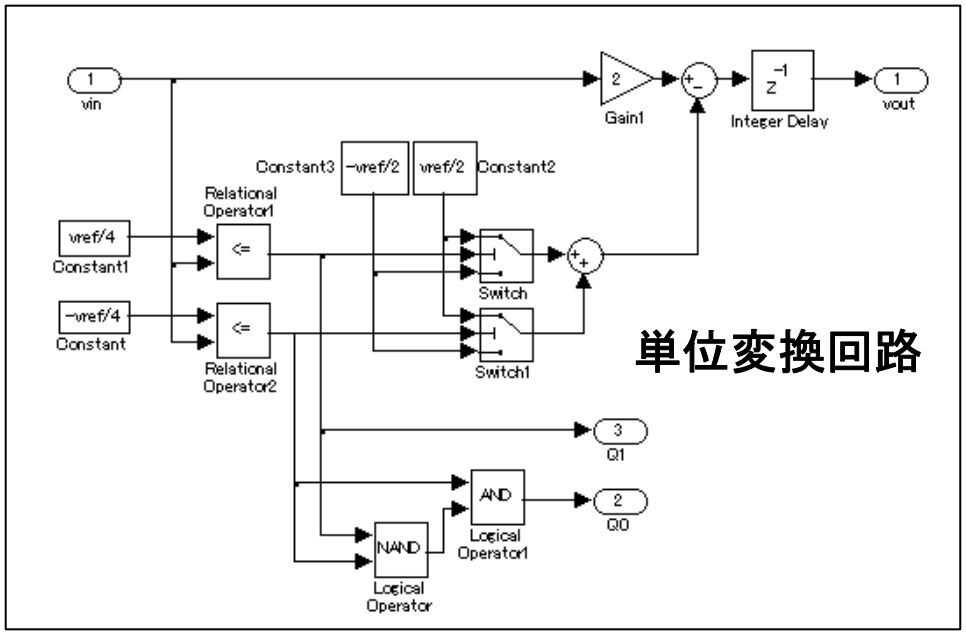


2nd out



Matsuzawa & Okada Lab.

MatLabによる機能シミュレーション

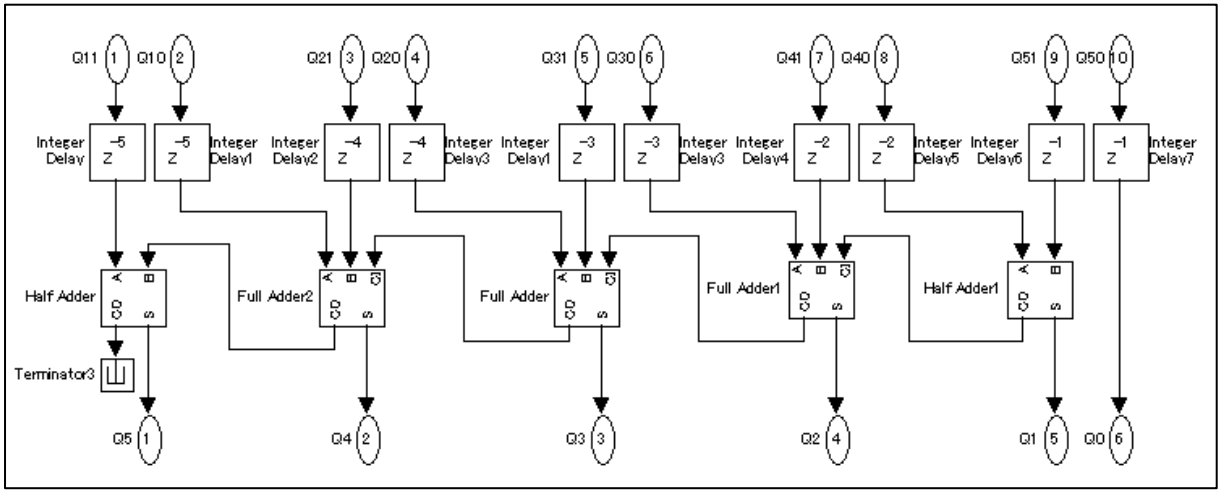


単位変換回路

容量ミスマッチ
利得
歪み
帯域
ノイズ
比較器オフセット

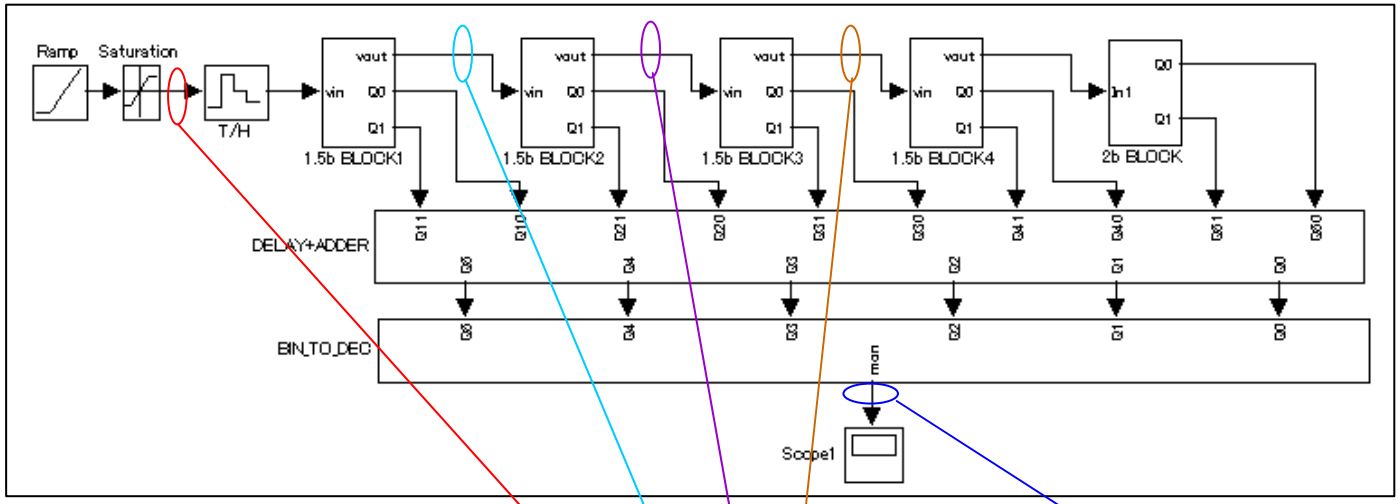
などをモデル化することができるが
どんなモデルが必要かは設計知識が必要

論理回路

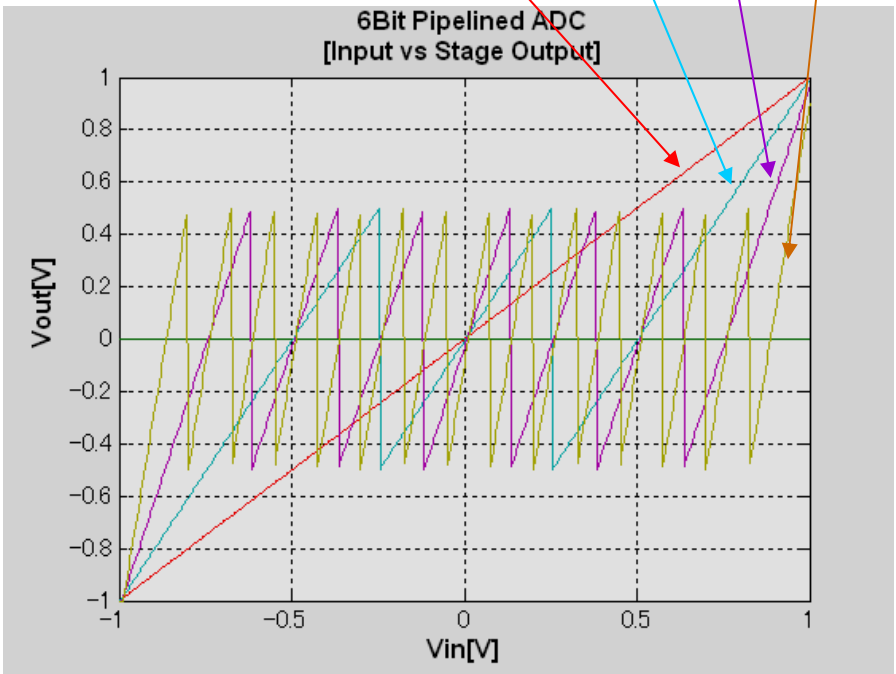


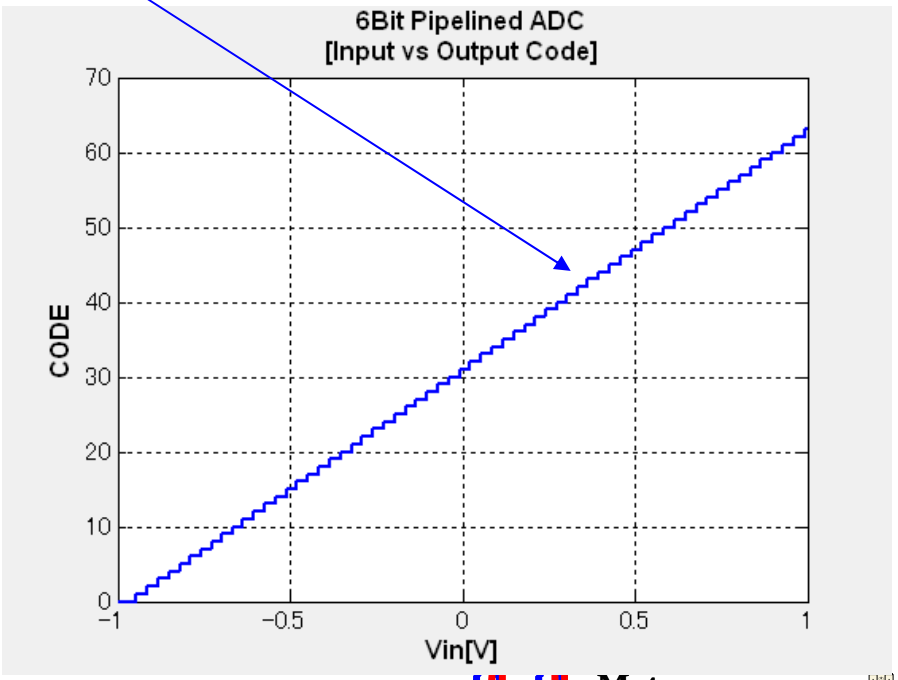
動作確認

TOKYO TECH Pursuing Excellence



SIM条件:
サンプリング周波数
100MHz
SIM時間
5us
Ramp波形スロープ
$4*10^5$ V/s





Matsuzawa & Okada Lab.

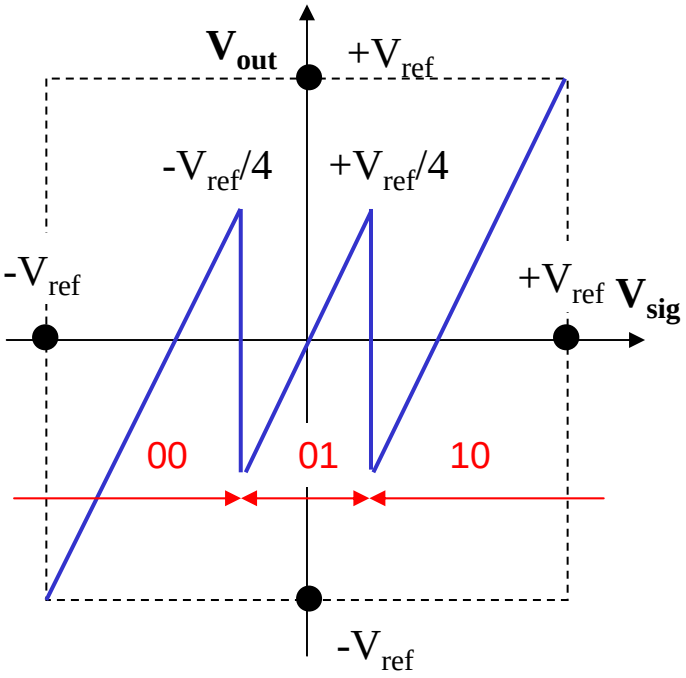
1.5ビット冗長構成

冗長構成により比較器と増幅器のオフセット電圧は変換特性に影響を与えない。

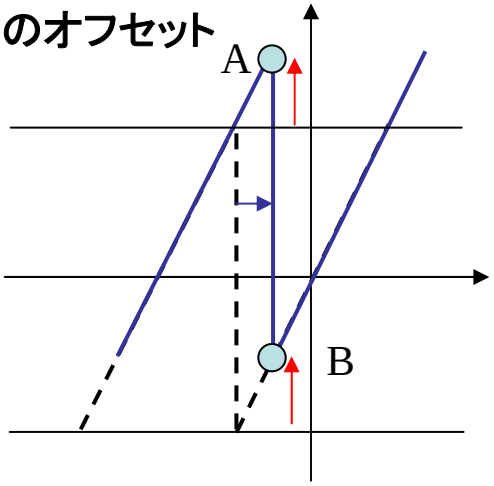
→精度はOpAmp周りで決まる

1.5ビット冗長構成の変換特性

変換範囲の充分内側で折れ返す特性



比較器のオフセット

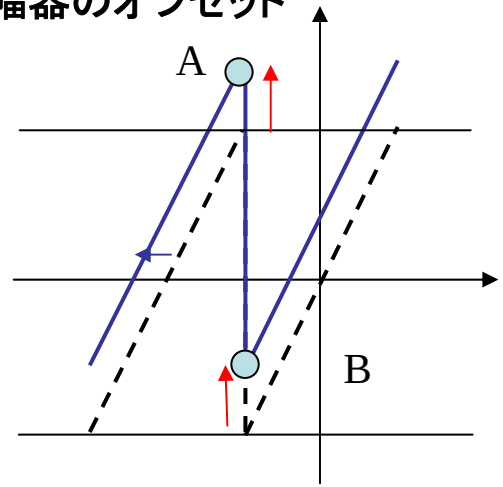


比較器のオフセットで
切り替わり点はずれる

利得が正確な場合
A点とB点は値として
つながる

**比較器のオフセットは
補正可能**

増幅器のオフセット



増幅器のオフセットで
変換特性は上下にシフト

利得が正確な場合
A点とB点は値として
つながる

**増幅器のオフセットは
補正可能**

システム解析から要求回路性能を導出

何が性能を決め、何が決めないかを知ることが重要

- 比較器とOpAmpのオフセット電圧は直線性に殆ど影響を与えない

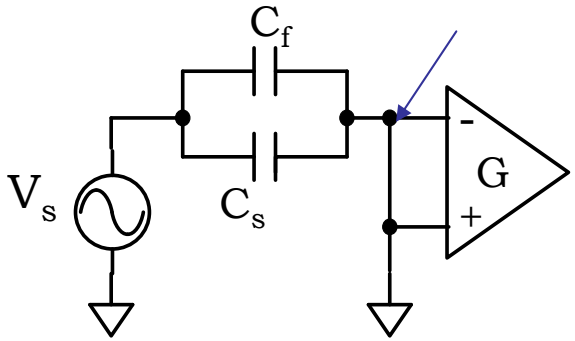
- 直線性・SNRへの影響要因
  - OpAmp利得
  - 容量ミスマッチ
  - ノイズ

- 変換速度
  - OpAmp帯域
  - スイッチ・比較器・ロジックの速度

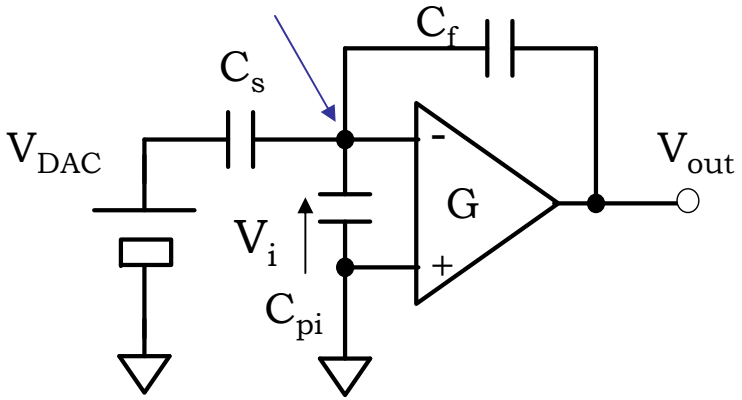
伝達関数の計算

OpAmpを用いたスイッチトキャパシタ回路から回路の伝達関数を導出

サンプルモード



増幅モード



注目ノードの電荷は

$$Q = -V_s\left(C_f + C_s\right)$$

$$Q' = -C_s\left(V_{DAC} - V_i\right) - C_f\left(V_{out} - V_i\right) + C_{pi}V_i$$

$$Q = Q' \qquad V_{out} = -GV_i$$

$$V_{out} = \frac{V_s\left(1 + \frac{C_s}{C_f}\right) - \frac{C_s}{C_f}V_{DAC}}{1 + \frac{\left(1 + \frac{C_s}{C_f} + \frac{C_{pi}}{C_f}\right)}{G}}$$

必要利得の計算

$$V_{out} = \frac{2V_s - V_{DAC}}{1+\frac{3}{G}} \approx (2V_s - V_{DAC})\left(1-\frac{3}{G}\right)$$

$$\therefore \delta V_{out} \approx \frac{3(2V_s - V_{DAC})}{G}$$

$$\delta_1 \approx \frac{3\left(-2\cdot\frac{V_{ref}}{4}+V_{ref}\right)}{G} = -\frac{3}{2}\frac{V_{ref}}{G}$$

$$\delta_2 \approx \frac{3\left(-2\cdot\frac{V_{ref}}{4}\right)}{G} = -\frac{3}{2}\frac{V_{ref}}{G}$$

$$\frac{|\delta_1|+|\delta_2|}{2} = \frac{3}{2}\frac{V_{ref}}{G} < \frac{2V_{ref}}{2^N\cdot 8}$$

$$\therefore G > 6\times 2^N$$

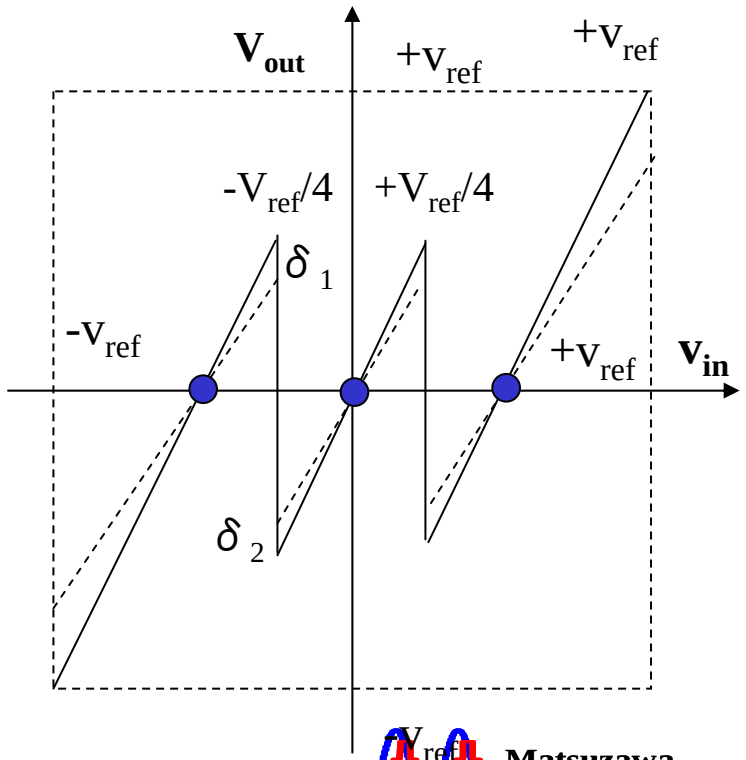
$$G > 6N+16(dB) \quad \frac{1}{8}LSB\text{基準の場合}$$

$$G > 6N+10(dB) \quad \frac{1}{4}LSB\text{基準の場合}$$

N=10を代入して　76dB

$\frac{C_{pi}}{C_f} = 1$と仮定する

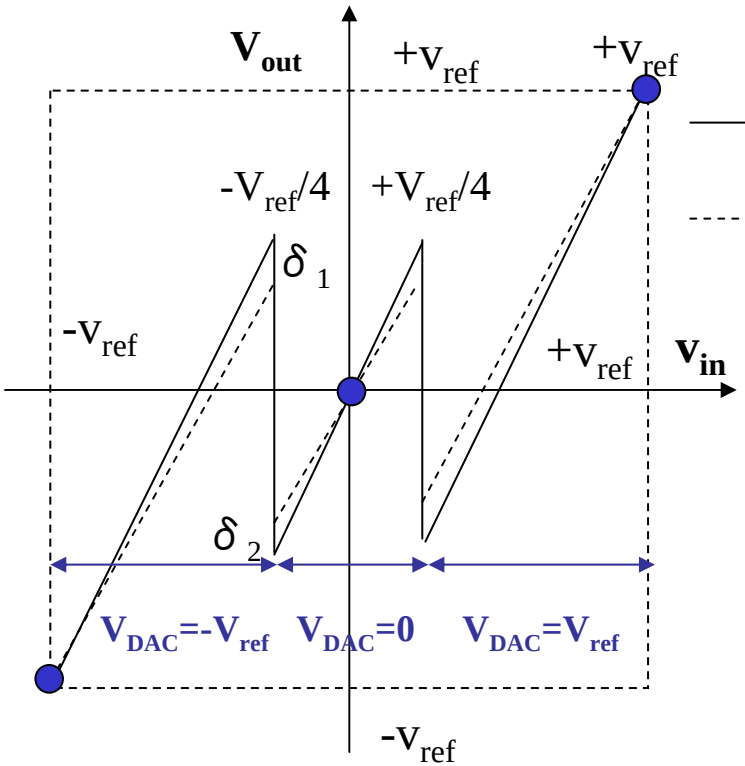
有限利得による誤差



容量ミスマッチの計算　(1.5b)

$$V_{out} \approx V_s\left(1+\frac{C_s}{C_f}\right)-\frac{C_s}{C_f}V_{DAC} \approx V_s(2+\Delta_c)-(1+\Delta_c)V_{DAC} \qquad C_s=(1+\Delta_c)C_f$$

$$\delta=\Delta_c(V_s-V_{DAC})$$



——$C_f=C_s$　ミスマッチによる変換誤差を1/4 LSBとする

------ $C_f \neq C_s$　OPアンプ利得は無限大

$$\delta_1=\Delta_c\left(-\frac{V_{ref}}{4}+V_{ref}\right)=\frac{3}{4}\Delta_c V_{ref}$$

$$\delta_2=\Delta_c\left(-\frac{V_{ref}}{4}\right)=-\frac{1}{4}\Delta_c V_{ref}$$

$$\frac{|\delta_1|+|\delta_2|}{2}<\frac{LSB}{4}=\frac{2V_{ref}}{2^N\cdot 4}$$

$$\Delta_c<\frac{1}{2^N}$$

容量の決定

TOKYO TECH Pursuing Excellence

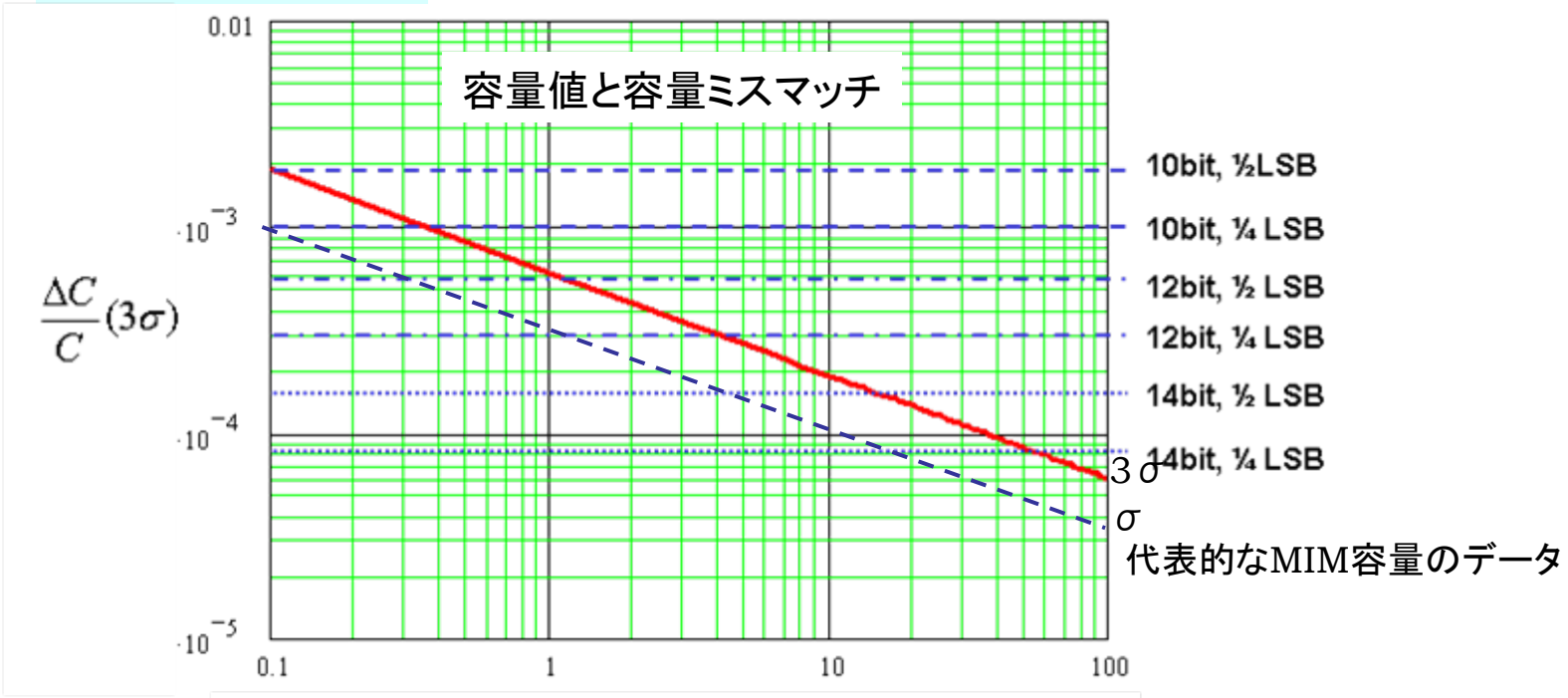
容量ミスマッチの一般的な性質により、必要容量は分解能の2乗に比例する

高精度ADCには大きな容量が必要となる

$$\frac{\Delta C}{C}(3\sigma) = \frac{6\times10^{-4}}{\sqrt{C\,(pF)}}$$

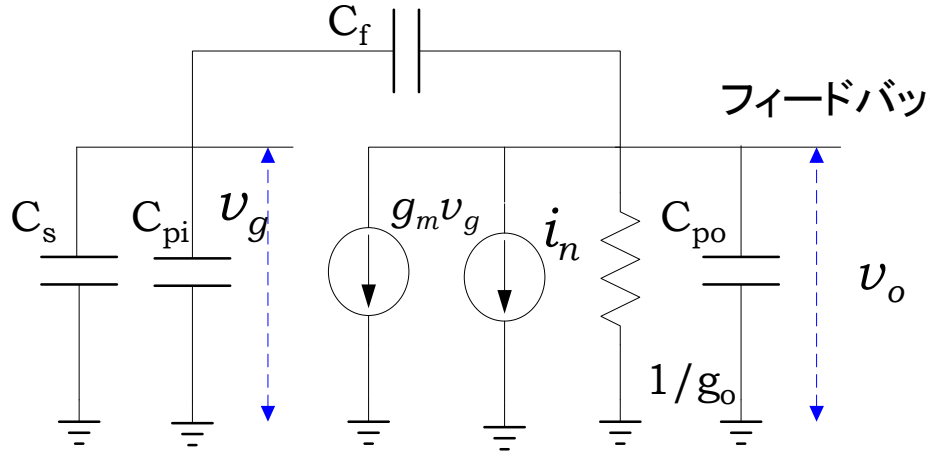
$$C \geq 3.6\times10^{-19}2^{2N}$$

$$C \propto 2^{2N}$$



増幅時の出力ノイズ

OPアンプの増幅時におけるノイズを計算する



フィードバックファクター

$$\beta \equiv \frac{v_g}{v_o}$$

これが小さいとノイズが増加し
応答速度が遅くなる

ノイズ源の数
- n=2: Cascode
- n=4: Folded Cascode

γ : ノイズ係数 $(\frac{2}{3} \sim 2)$

C_L:出力端から見える全容量

$$v_o = \frac{-g_m v_g + i_n}{g_o + sC_L},\ v_g = \beta v_o \ \therefore v_o = \frac{i_n}{g_o + g_m\beta + sC_L} \approx \frac{i_n}{g_m\beta + sC_L}$$

$$v_{no}^2 / Hz = \frac{i_n^2}{(g_m\beta)^2 + (\omega C_L)^2},\ i_n^2 = 4\gamma \cdot nkTg_m$$

$$\therefore v_{no}^2 = \int_0^\infty \frac{i_n^2}{(g_m\beta)^2 + (\omega C_L)^2} df = \frac{\gamma \cdot n \cdot kT}{\beta C_L}$$

$$\because \int_0^\infty \frac{1}{a^2 + (2\pi f)^2} df = \frac{1}{4a} \qquad C_s = C_f = C_0$$

$$v_{no}^2 = \frac{\gamma \cdot n \cdot kT}{\beta C_L}$$

$$v_{ni}^2 = \frac{2}{4}\frac{\gamma \cdot n \cdot kT}{\beta C_L} = \frac{\gamma \cdot n \cdot kT}{2\beta C_L}$$

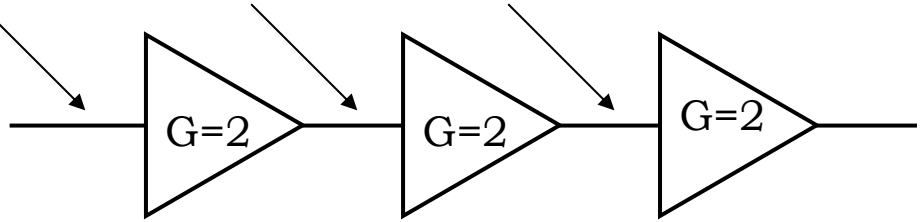
差動なので2倍
入力換算は2^2分の1

パイプライン全体のノイズ

熱雑音は信号系の容量とフィードバックファクター、回路形式で決まる。

$$v_{in1}^2 = \left(\frac{\gamma \cdot n \cdot}{2\beta C_L} + \frac{1}{C_0}\right) kT$$

$$v_{in2}^2 = 2v_{in1}^2 \qquad v_{in3}^2 = 4v_{in1}^2$$



1段目　2段目　3段目

$$\beta \to \beta$$

$$C_L \to \frac{C_L}{2}$$

$$v_{ni}^2 = 2 \cdot v_{ni-1}^2$$

i段目のノイズ

$$v_{in\_i}^2 = 2^{i-1} v_{in1}^2$$

$$v_{in\_nt}^2 = \sum_{i=1}^{N} \frac{1}{2^{i-1}} v_{in1}^2 = v_{in1}^2 \left(1 + \frac{1}{2} + \frac{1}{4} \ldots\right) \approx 2v_{in1}^2$$

入力換算でのi段目のノイズ

$$v_{in\_i}^2 = \frac{2^{i-1}}{4^{i-1}} v_{in1}^2 = \frac{1}{2^{i-1}} v_{in1}^2$$

$$C_0 > \left(\frac{9}{8}\gamma n + 2\right) \frac{kT}{v_{nt}^2}$$

$$C_0 > 3\left(2 + \frac{9}{8}\gamma n\right)\left(\frac{2^N}{V_{ref}}\right)^2 kT$$

$$v_{nt}^2 = \left(\frac{2}{C_0} + \frac{\gamma n kT}{\beta C_L}\right) kT$$

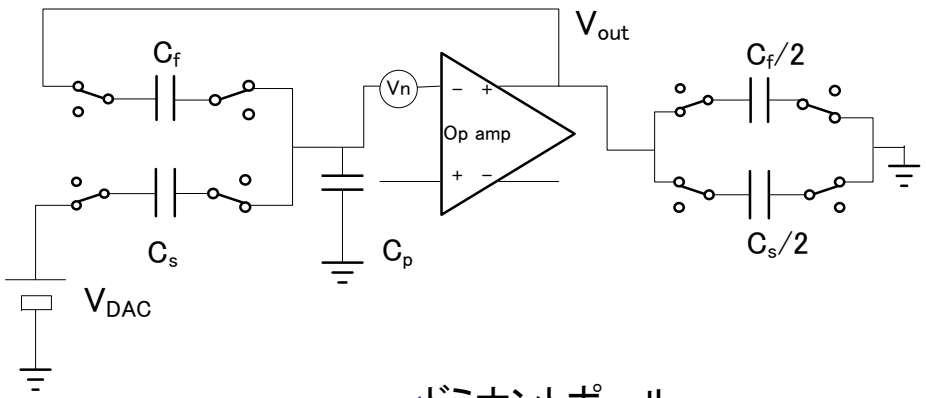
$\beta = \frac{1}{3}$　を代入

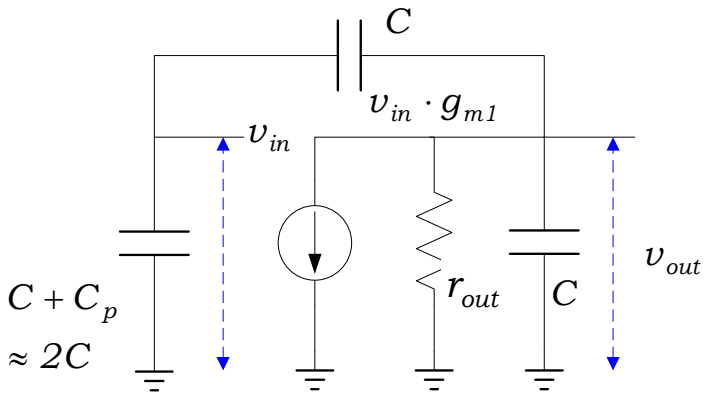
$C_L = \frac{8}{3} C_0$

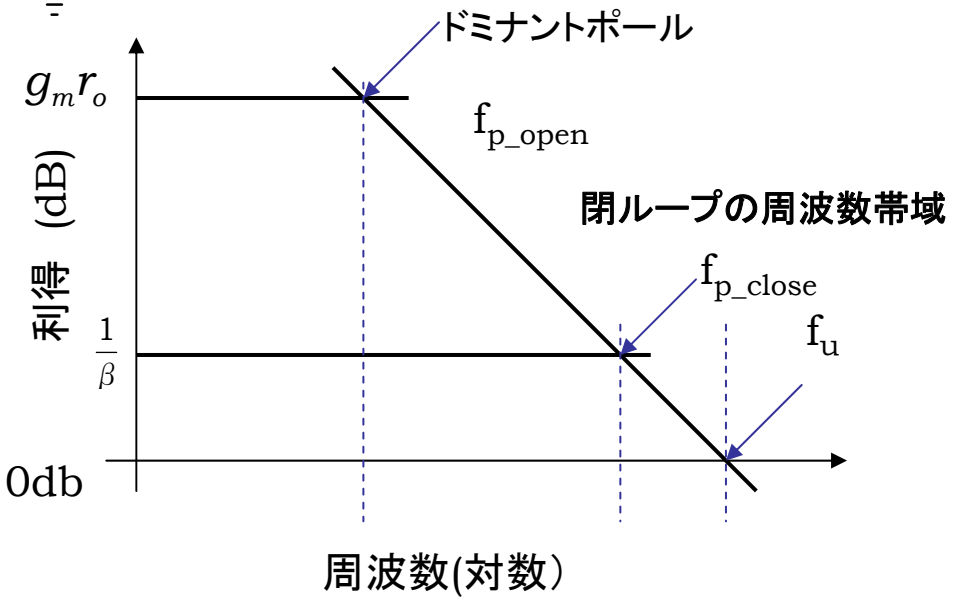
$$v_{nt}^2 \le \frac{1}{3}\left(\frac{V_{ref}}{2^N}\right)^2$$

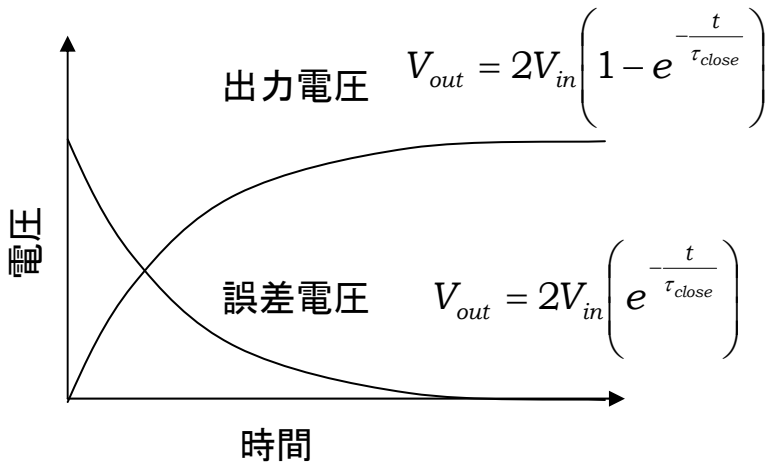
変換周波数

閉ループでのステップ応答で変換周波数が決まる





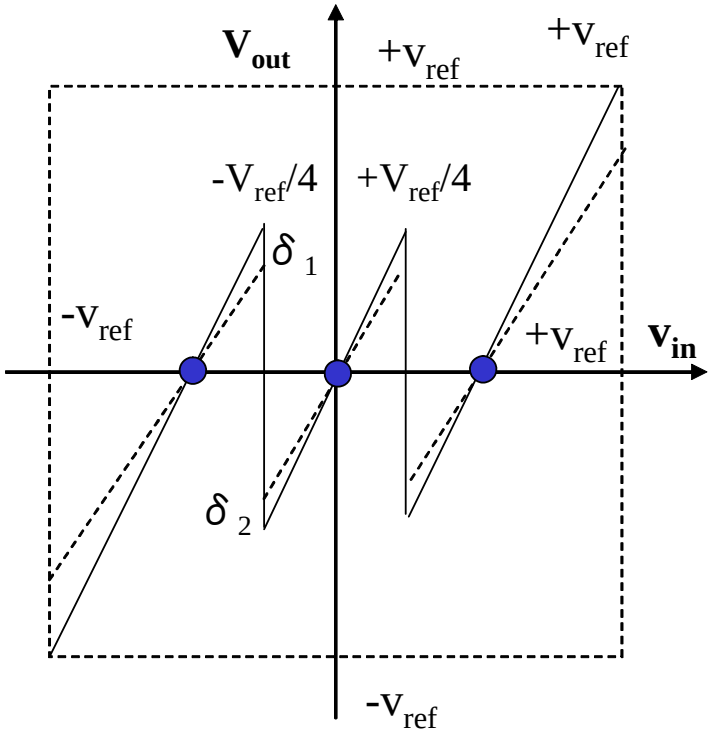




変換周波数

TOKYO TECH Pursuing Excellence

変換周波数は閉ループバンド幅と分解能で決まる



$$\delta_1 = \delta_2 = 2\frac{V_{ref}}{4}e^{-\frac{t_{ss}}{\tau}}$$

$$\frac{\delta_1 + \delta_2}{2} = \frac{V_{ref}}{2}e^{-\frac{t_{ss}}{\tau}} < \frac{2V_{ref}}{2^N \cdot 4}$$

$$\omega_{close} = \frac{g_m\beta}{C_L}$$

$$GBW_{close} = \frac{g_m\beta}{2\pi C_L}$$

セットリング誤差をLSB/4と仮定

$$\tau < \frac{t_{ss}}{0.7N} \quad t_{ss} = \frac{1}{3f_c} \quad \tau = \frac{1}{2\pi GBW_{close}} = \frac{1}{\omega_{close}}$$

セットリング時間は変換周器の1/3仮定

$$GBW_{close} > \frac{3f_c \cdot 0.7N}{2\pi} \approx \frac{Nf_c}{3}$$

要求回路性能に基づく回路設計

- 要求回路性能
  - 信号振幅
  - 容量値
  - ノイズ電圧
  - 閉ループ帯域
- 回路設計
  - 回路形式
  - 動作電流
  - トランジスタサイズ

フォールデ゛ィッド゛カスコード゛型OPアンプ゛

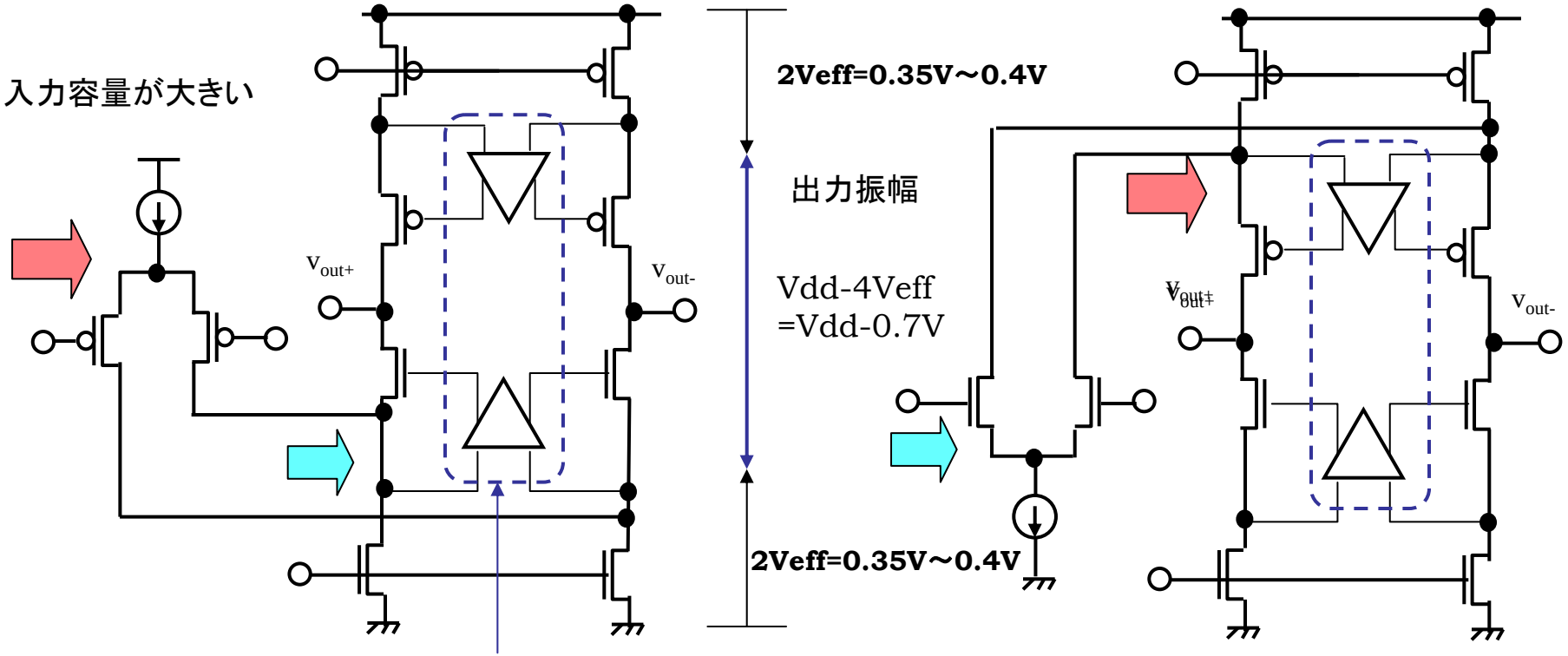
OpAmpの形式として使いやすいフォールディッドカスコード型を選択した場合

PMOS入力　　**通常NMOS入力型が高速、低電力に有利である。**　　NMOS入力

入力容量がNMOS入力の3倍程度
→β が下がり、閉ループでの帯域が低下
第2ポールが高く、位相余裕が大きい

カスコードだけでは低電圧で40dB程度しか利得が取れない
→ゲインブースト回路で利得を稼ぐ

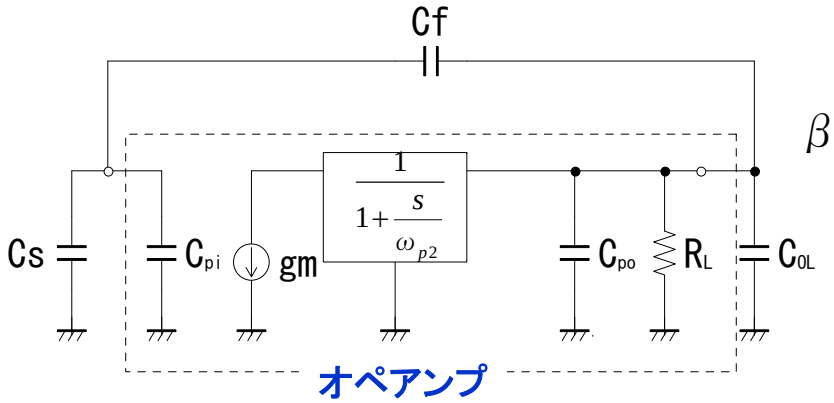
入力容量がPMOS入力の1/3倍程度
→β が高く、閉ループでの帯域が高い
第2ポールが低く、位相余裕が小さい



ゲインブースト回路を入れても消費電力は20%程度の増加で済む

閉ループ帯域の算出

閉ループ帯域はg_mと容量（信号系容量とトランジスタ容量）で決まる



単位変換回路の等価回路

g_m : 入力部トランジスタのトランスコンダクタンス

C_s, C_f : 帰還容量

C_{oL} : 次段の帰還容量の和

C_{pi}, C_{po} : オペアンプの入力,出力寄生容量

R_L : オペアンプ出力抵抗

ω_{p2} : オペアンプの第2ポール

クローズドループにおける $GBW_{\_close}$ は

$$GBW_{\_close} = \frac{g_m}{2\pi C_L}\beta$$

β は帰還係数、C_L は実効負荷容量を表し、

$$\beta = \frac{C_f}{C_f + C_s + C_{pi}}$$

$$C_L = C_{po} + C_{oL} + \frac{C_f\left(C_s + C_{pi}\right)}{C_f + C_s + C_{pi}}$$

次段の帰還容量を1/2ずつ減ずると仮定すると、

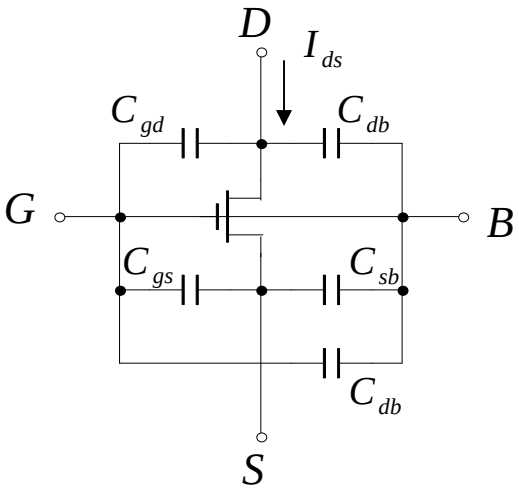
$$C_{oL} = \frac{C_s + C_f}{2}$$

$$C_o = C_s = C_f = C_{oL}$$

これより、

$$GBW_{\_close} = \frac{g_m}{2\pi C_o}\frac{1}{\left(2+\frac{C_{pi}}{C_o}\right)\left(1+\frac{C_{po}}{C_o}\right)+\left(1+\frac{C_{pi}}{C_o}\right)}$$

トランジスタのキャラクタライズ



アナログ回路ではV_{eff} ($=V_{GS}-V_T$)を
一定にして動作電流を変化させるので
V_{eff}, Lをいくつか想定し、それ毎にI_Dに対する
W、各容量などをキャラクタライズしておくと設計が簡単になる

トランジスタ容量はI_Dに比例する

$$g_m = \frac{2I_{ds}}{V_{eff}}$$

$$W = \frac{2L}{\mu C_{ox} V_{eff}^{\ 2}} I_{ds}$$

代表的プロセスでのMOSのキャラクタライズ　V_{eff}=0.175V

(a)W_N,W_P[μm/mA],V_{A_N}, V_{A_P}[V]

ルール	W_N	W_P	V_{A_N}	V_{A_P}
90nm	24.3	74.9	0.82	0.69
0.13μm	37.5	147	0.82	0.64
0.18μm	54.8	219	0.99	0.93
0.25μm	116.0	396	0.78	0.97
0.35μm	162.0	603	1.01	0.86

(b)C_{pi_N}, C_{pi_P},C_{po}[fF/mA], ω_{p2_N}, ω_{p2_P}[GHz]

ルール	C_{pi_N}	C_{pi_P}	C_{po}	ω_{p2_N}	ω_{p2_P}
90nm	23.7	93.4	94.5	9.35	15.4
0.13μm	65.5	249	168	7.7	10.3
0.18μm	115	475	340	2.06	4.7
0.25μm	236	662	832	0.83	1.7
0.35μm	303	1034	892	0.54	1.7

動作電流と変換速度の推定

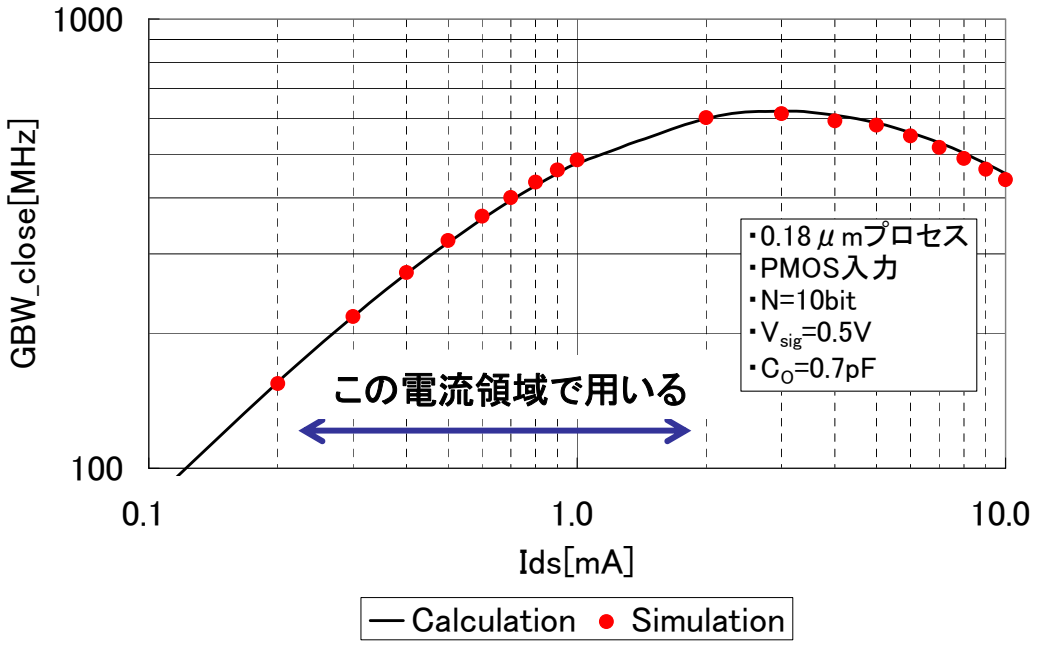
TOKYO TECH Pursuing Excellence

閉ループ帯域は動作電流の関数となる

電流 I_{ds} で規格化した寄生容量 C_{pi}, C_{po} をもとに $GBW_{\_close}$ を推定する。

$$GBW_{\_close} = \frac{g_m}{2\pi C_o}\frac{1}{\left(2+\frac{C_{pi}}{C_o}\right)\left(1+\frac{C_{po}}{C_o}\right)+\left(1+\frac{C_{pi}}{C_o}\right)} = \frac{I_{ds}}{\pi C_o V_{eff}}\frac{1}{\left(2+\frac{\alpha_{pi}I_{ds}}{C_o}\right)\left(1+\frac{\alpha_{po}I_{ds}}{C_o}\right)+\left(1+\frac{\alpha_{pi}I_{ds}}{C_o}\right)}$$

計算値とシミュレーション結果の比較



$$g_m = \frac{2I_{ds}}{V_{eff}} \quad C_{pi} = \alpha_{pi} I_{ds}, \quad C_{po} = \alpha_{po} I_{ds}$$

α_{pi}, α_{po} はデザインルールに依存

C_o は熱雑音などを考慮して、

$$C_o \geq 1.66 \times 10^{-19}\left(\frac{2^N}{V_{sig}}\right)^2$$

理論値とSim結果は5%以内で一致
（入力の寄生容量 C_{gd} の
ミラー効果を2倍として計算）

Ids:各トランジスタの動作電流（全体では $4I_{ds}$）

Matsuzawa & Okada Lab.

CMOS OpAmpの利得

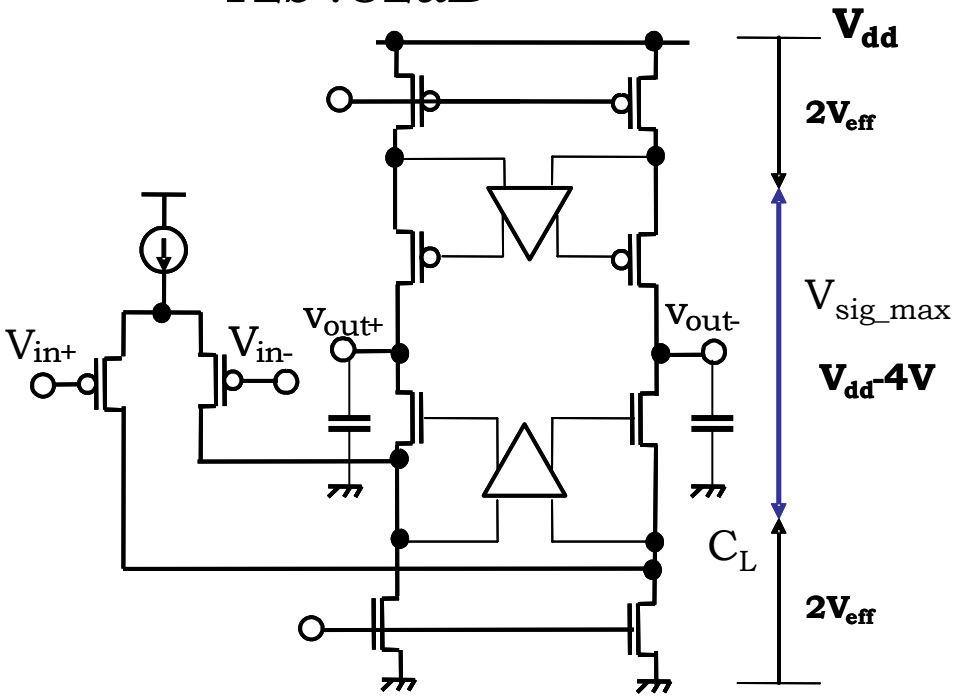
トランジスタのキャラクタライズから、利得を見積もる

パイプライン型ADCの性能はOPアンプに依存しているが、
微細化が進むと必要な利得が取れない。

$$G_{DC}(dB) > 6N + 10$$

$10b : 70dB$

$12b : 82dB$



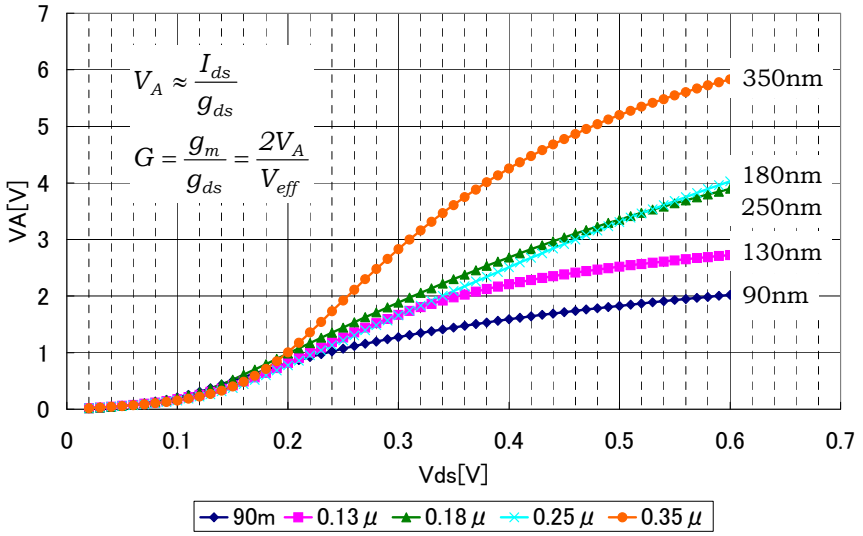
Sub-100nm CMOS

$$G_{DC} \approx \left(\frac{V_A}{V_{eff}}\right)^n \approx \left(\frac{1}{0.15}\right)^n \approx 16dB \times n$$

$n < 5$

$G_{DC} < 80dB$

微細化とV_A



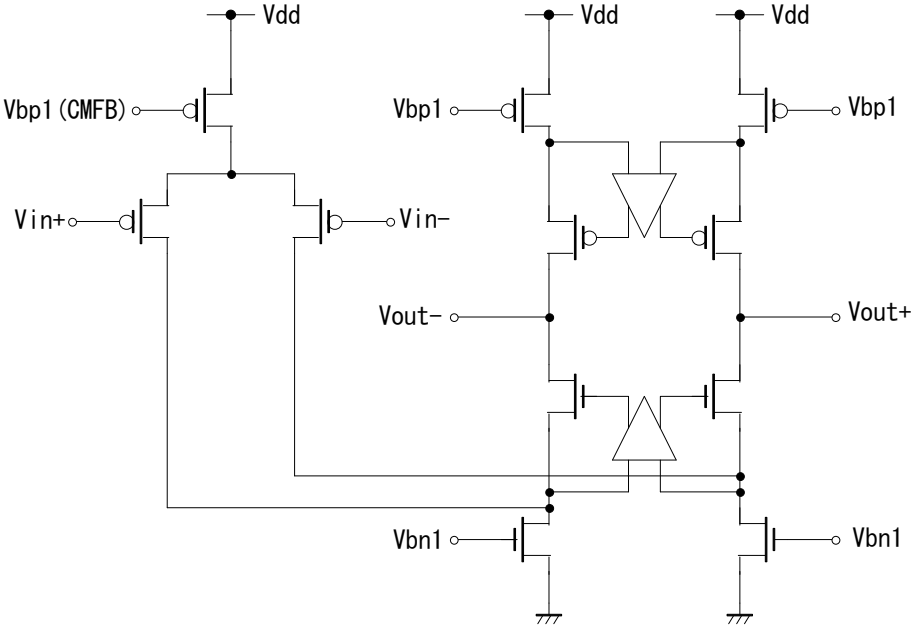
ゲインブースト

TOKYO TECH Pursuing Excellence

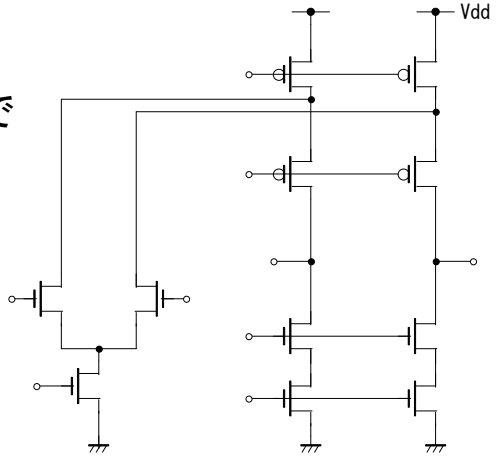
高いDCゲインが必要な場合はゲインブースト回路を用いる

※90nmではメインアンプ単体で
35～40dB程度しか出ない

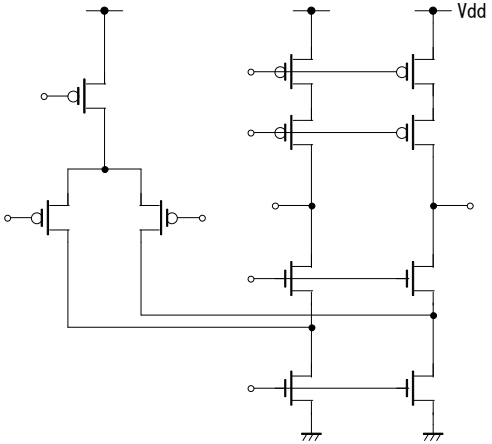
N,PMOS側それぞれにゲインブーストアンプを用いて
出力抵抗を上げてゲインを増す。



メインアンプ



PMOS側ブーストアンプ



NMOS側ブーストアンプ

Matsuzawa & Okada Lab.

MatLabとCadence ADE

TOKYO TECH Pursuing Excellence

- **MatLab**
  - モデルベース(プログラムベース)Sim
  - SimLinkによりモデル構築が容易
  - 豊富なデータ処理機能(FFT, フィルタ、ヒストグラム, ポールゼロなど)と可視化機能
  - タイムベースの可変化が困難
  - SPICEとの混在Simが困難
- **Cadence ADE**
  - モデルベース+トランジスタベース
  - タイムベースの可変化が容易
  - SPICEとの混在Simが可能
  - 貧弱なデータ処理機能と可視化機能

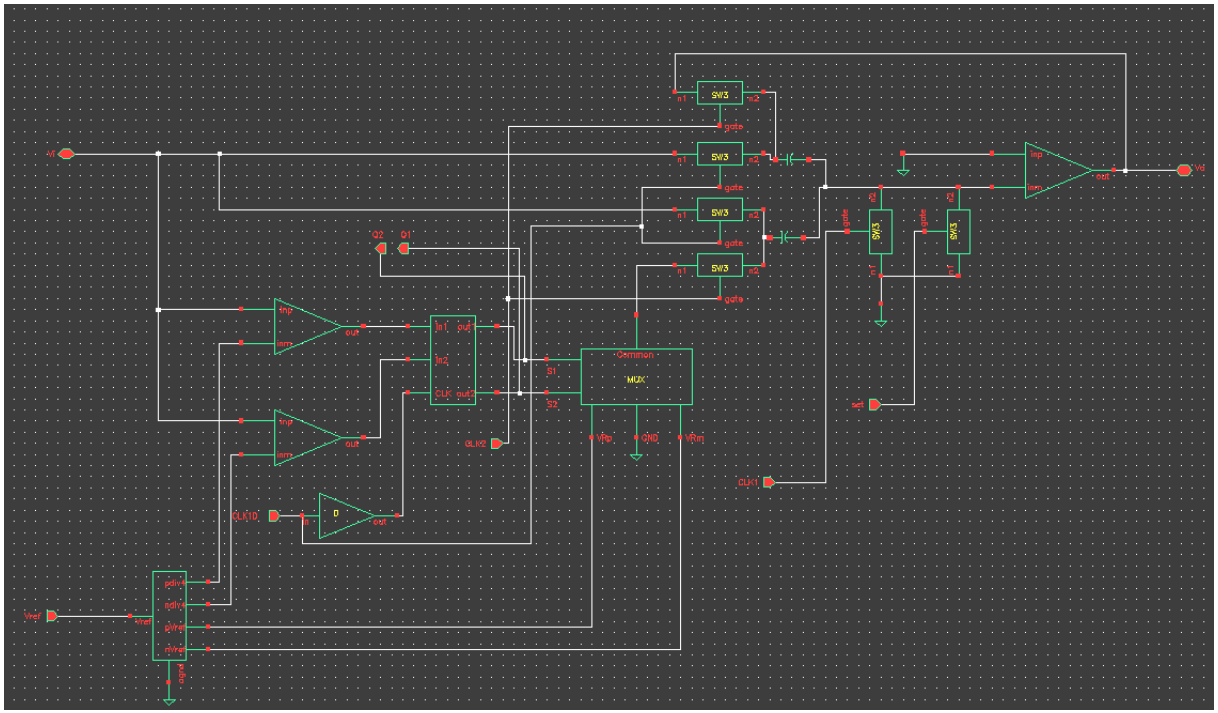
互いに補う関係

Verilog-AとSpiceの混在シミュレーション

TOKYO TECH Pursuing Excellence

回路設計においてはVerilog-AとSpice (SPECTRE)の
混在シミュレーションが有効である

初めはVerilog-Aで記述し、機能を確認してから、
ブロックごとにトランジスタレベルに変更し、性能を確認してゆく。

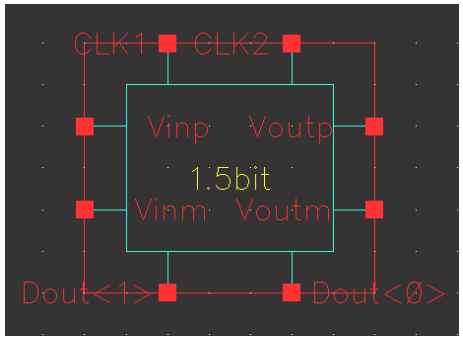


Matsuzawa & Okada Lab.

単位変換回路の記述

TOKYO TECH Pursuing Excellence

```
//Verilog-AMS HDL for "seminar", "idealstage"
"verilogams"

`include "constants.vams"
`include "disciplines.vams"
`timescale 10ps / 10ps

module idealstage(Dout, Vinm, Vinp, Voutm, Voutp,
CLK1,CLK2);
inout Vinm, Vinp, Voutm, Voutp, CLK1, CLK2;
output [1:0] Dout;
electrical Vinm, Vinp, Voutm, Voutp, CLK1, CLK2, dclk;

reg [1:0] Dint;
wire [1:0] Dout = Dint;

parameter real
                vref = 0.25,
                vcom = 0.6,
                Cf = 1.0p;
                Cs = 1.0p;
                trise = 0.1n,
                tfall = 0.1n,
                dt = 0.5n;

real vtd1,vtd2;
real vinp,vinm,vin;
integer D;


analog begin
                vtd1 = -vref/2;
                vtd2 = vref/2;

                V(dclk) <+ absdelay(V(CLK1),dt);

                @(cross(V(dclk)-vcom,+1.0)) begin
                                vinp = V(Vinp);
                                vinm = V(Vinm);
                end
```




```
vin = vinp - vinm;

if(vin < vtd1 && V(CLK2) > vcom ) begin
        V(Voutp) <+ ((Cs+Cf)*(vinp-vcom)+Cs*vref)/Cf+vcom;
        V(Voutm) <+ ((Cs+Cf)*(vinm-vcom)-Cs*vref)/Cf+vcom;
end
else if(vin >= vtd2 && V(CLK2) > vcom) begin
        V(Voutp) <+ ((Cs+Cf)*(vinp-vcom)-Cs*vref)/Cf+vcom;
        V(Voutm) <+ ((Cs+Cf)*(vinm-vcom)+Cs*vref)/Cf+vcom;
end
else if(vin >= vtd1 && vin < vtd2 && V(CLK2) > vcom) begin
        V(Voutp) <+ ((Cs+Cf)*(vinp-vcom))/Cf+vcom;
        V(Voutm) <+ ((Cs+Cf)*(vinm-vcom))/Cf+vcom;
end
else begin
                V(Voutp) <+ vcom;
                V(Voutm) <+ vcom;
end

if(vin < vtd1) begin //digital
                D = 0;
end
else if(vin >= vtd1 && vin < vtd2) begin
                D = 1;
end
else if(vin >= vtd2) begin
                D = 2;
end
else begin
                D = 3;
end
end

always @(cross(V(CLK2)-vcom,+1.0)) begin
                Dint <= #1 D;
end
endmodule
```


Matsuzawa & Okada Lab.

D/A及びDNL、INLデータ取得

TOKYO TECH Pursuing Excellence

得られたデータをMatLab, Excelに取り込む

パイプラインADCからのCodeをD/Aする。元の信号との比較を行いDNL,INLのデータを得る

```
//Verilog-AMS HDL for "seminar", "DtoA"
"verilogams"

`include "constants.vams"
`include "disciplines.vams"

module DtoA ( Aout, Q, CLK, Ref_p,
Ref_m);

output Aout;
input  [9:0] Q;
input CLK;
input Ref_p, Ref_m;
electrical Aout, CLK, Ref_p, Ref_m;
wire [9:0] Q;

//file
integer f1,f2; //f1=DNL,f2=INL file

//DtoA
parameter real
                vref = 0.25,
                vcom = 0.6,
                trise = 0.1n,
                tfall = 0.1n;

parameter integer
                bit = 10;

real dq, f_s, vout0, vout1;
integer digit, i;
```

```
//Ref in
parameter integer count = 0;
real ref[0:20];
real refdata0, refdata1;
integer j;
//DNL
real DNL;
integer qdata;
analog begin
@(initial_step) begin
      f1=$fopen("~/DNL.dat");
      f2=$fopen("~/INL.dat");
      vout0=0;
      vout1=0;
      f_s= 4*vref;
      dq = f_s/pow(2,bit);
end
@(final_step) begin
      $fclose(f1);
      $fclose(f2);
end
//DtoA
@(cross(V(CLK)-vcom,+1.0)) begin
      digit=0;
      for(i=0; i < bit; i=i+1) begin
            if(Q[i] == 1) begin
                digit = digit+pow(2,i);
            end
      end
end
vout0 = dq*digit;
V(Aout) <+ transition(vout0, 0.0, trise,
tfall)+dq/2;
```

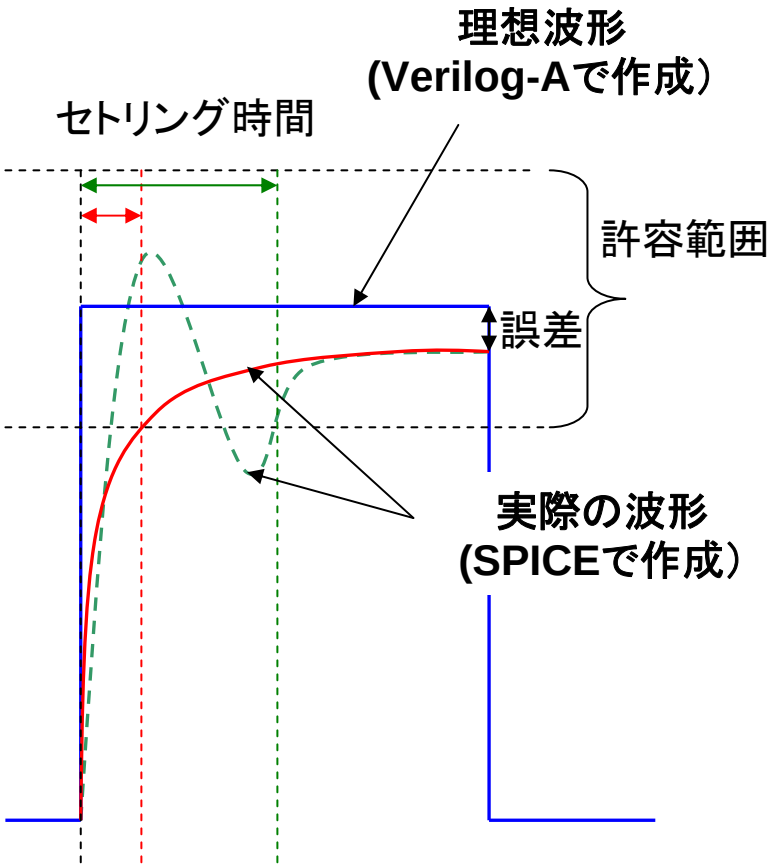
```
//Ref_in
@(cross(V(CLK)-vcom,+1.0)) begin

      ref[0] = V(Ref_p)-V(Ref_m);

      for(j=count; j>0; j=j-1) begin
                  ref[j] = ref[j-1];
      end
      refdata0 = ref[count]+2*vref;
end

//DNL,INL
if(vout0 != vout1 && V(CLK) < vcom) begin
      qdata = Q;
      DNL = (refdata0-refdata1)/dq -1;

$fstrobe(f1,"%d %f %f",qdata,DNL,refdata0
);

$fstrobe(f2,"%d %f %f",qdata,refdata0,vout
0);
      refdata1 = refdata0;
      vout1 = vout0;
end

end

endmodule
```

セトリング時間＆誤差算出方法

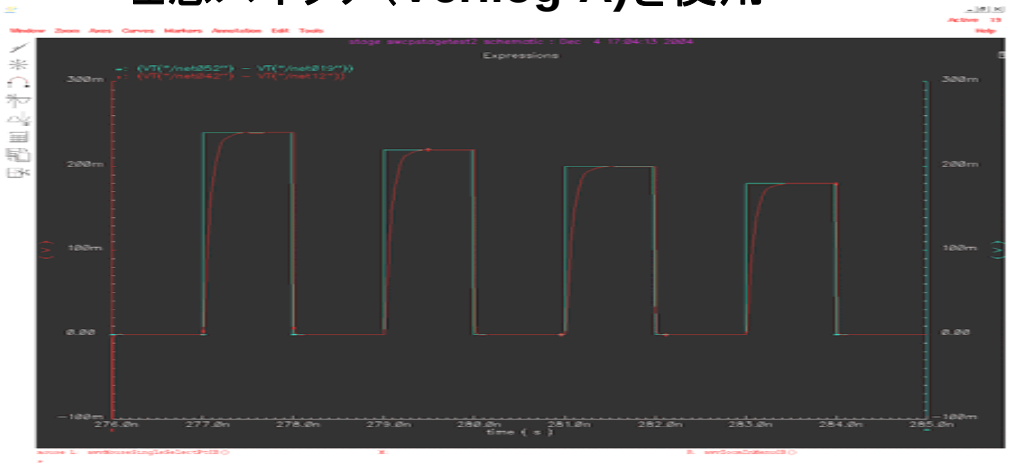
TOKYO TECH Pursuing Excellence

Verilog-AとSPECTREの混在シミュレーションの実際

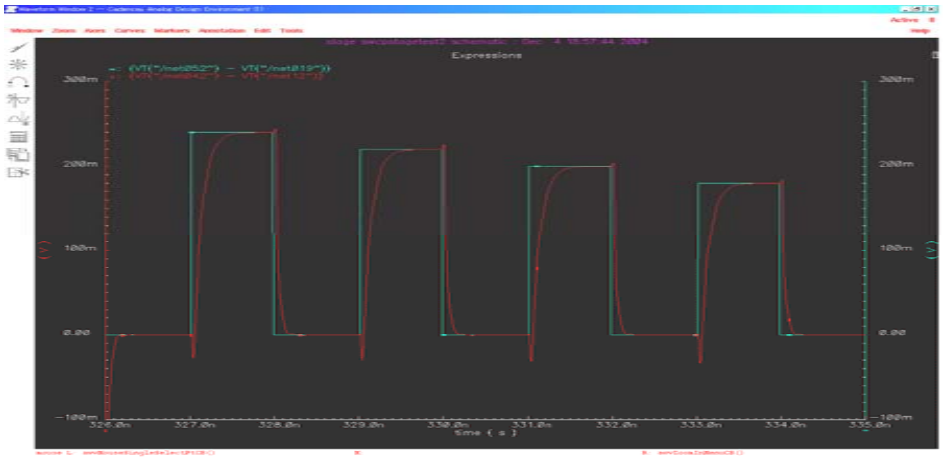
・理想スイッチではOpAmpで応答が決定
・MOSスイッチにより応答が劣化
→スイッチを最適化する



理想スイッチ（Verilog-A)を使用



MOSスイッチ（SPICE)を使用

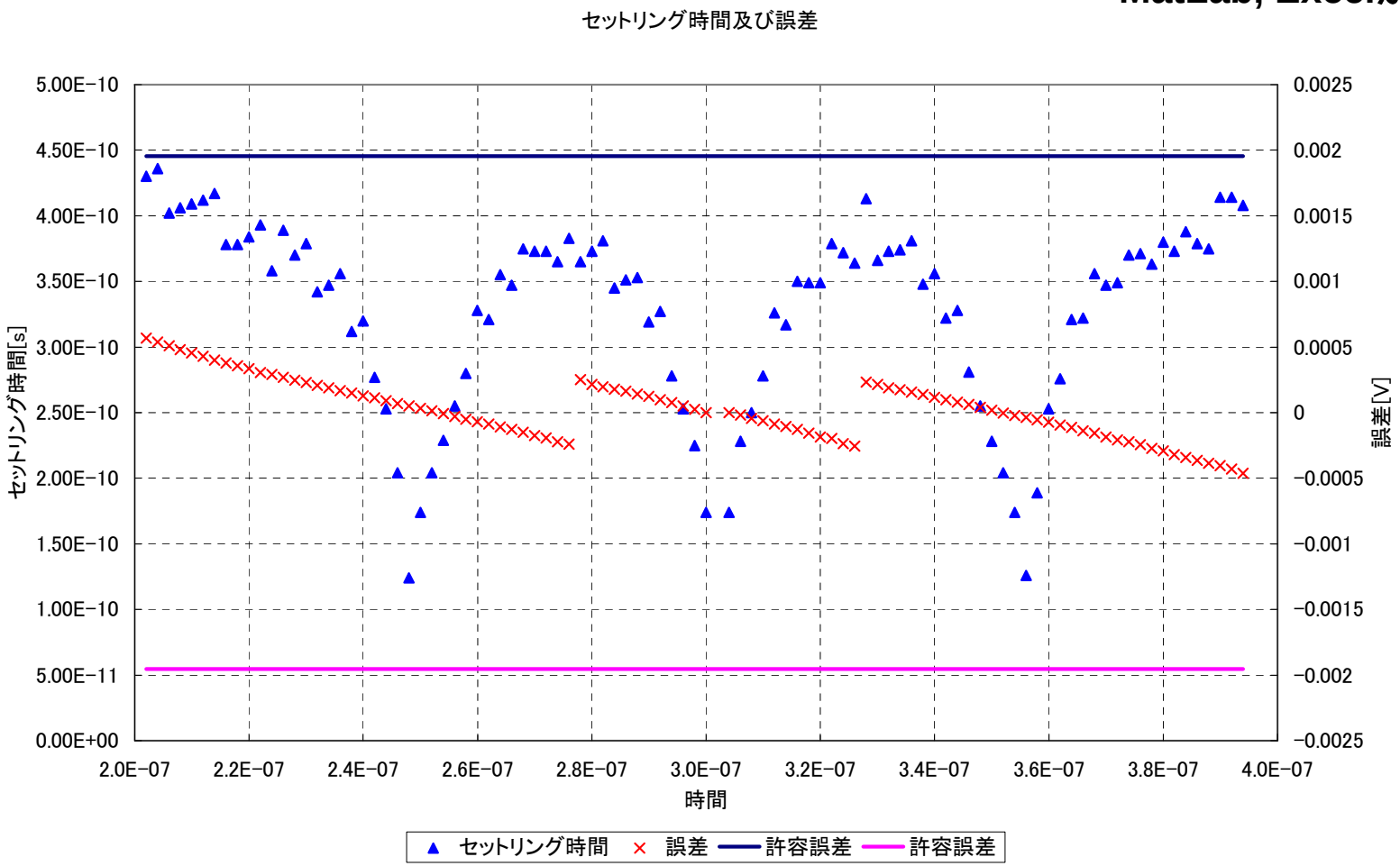


セットリング時間＆誤差の導出

TOKYO TECH Pursuing Excellence

得られたセットリング時間から誤差を算出

MatLab, Excelが便利



Matsuzawa & Okada Lab.

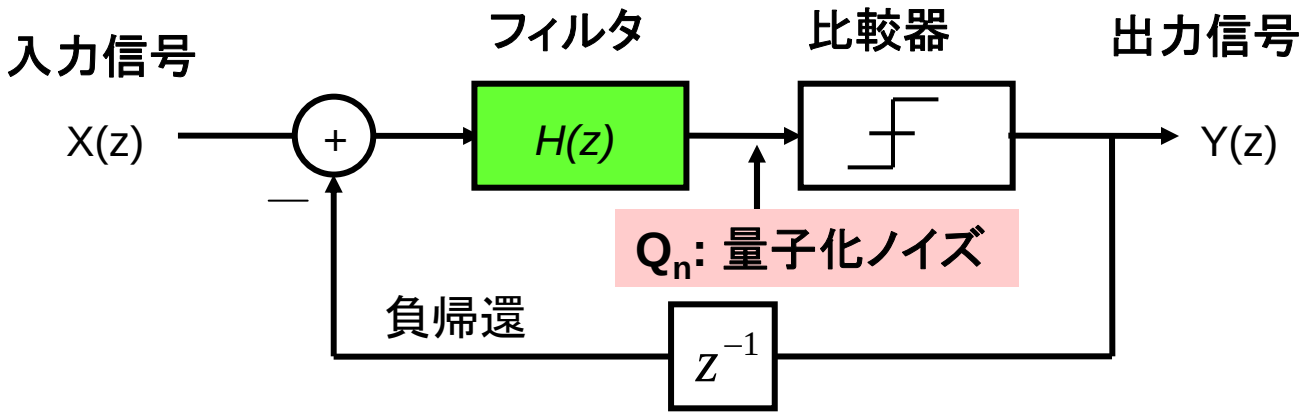
TOKYO TECH
Pursuing Excellence

ΔΣ型ADCの設計

Matsuzawa & Okada Lab.

Δ Σ 変調技術

Δ Σ 変調技術により比較器のノイズを効果的に抑制できる。



$$Y(z) = \frac{H(z)}{1 + H(z)z^{-1}} X(z) + \frac{1}{1 + H(z)z^{-1}} Q_n(z)$$

STF (Signal Transfer)　　NTF (Noise transfer)

$$Y(z) \approx X(z) + \frac{1}{H(z)} Q_n(z)$$

量子化ノイズにハイパスフィルタがかかっている

Ex. $H(z) = \frac{1}{1 - z^{-1}}$　$STF(z) = 1,$　$NTF(z) = 1 - z^{-1}$

No filter　　High pass filter

ΔΣ変調器の周波数特性

TOKYO TECH Pursuing Excellence

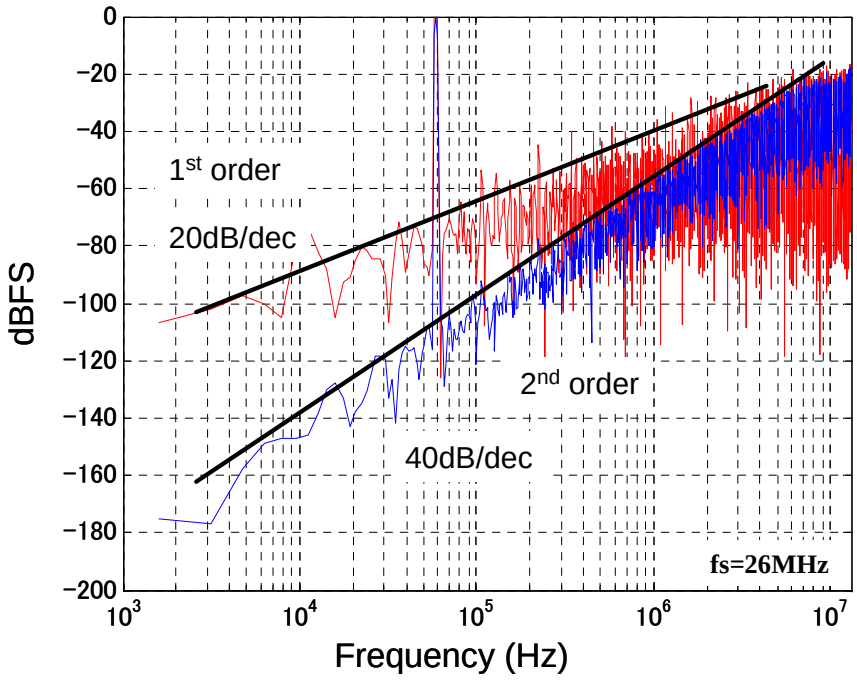
低周波ノイズは効果的に抑圧されている。

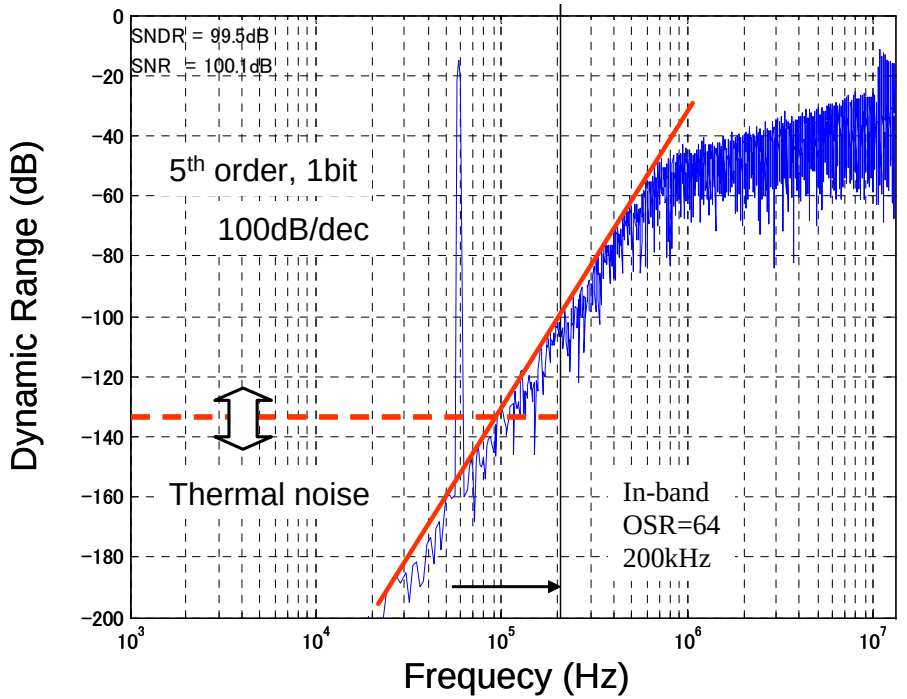
$$Y(z) = X(z) + \left(1 - z^{-1}\right)^L Q(z) \qquad N_q = \int_{-f_b}^{+f_b} h_q^2(f) \left|1 - z^{-1}\right|_{z=e^{j2\pi f/fs}}^{2L}$$

L：積分次数
M: オーバーサンプリング比

$$h_q^2(f) = \frac{\Delta^2}{12 f_s}$$

$$\approx \int_{-f_b}^{+f_b} \frac{\Delta^2}{12 f_s} \left| \frac{j2\pi f}{f_s} \right|^{2L} df = \left(\frac{\Delta}{2}\right)^2 \frac{1}{3\pi(2L+1)} \left(\frac{\pi}{M}\right)^{2L+1}$$





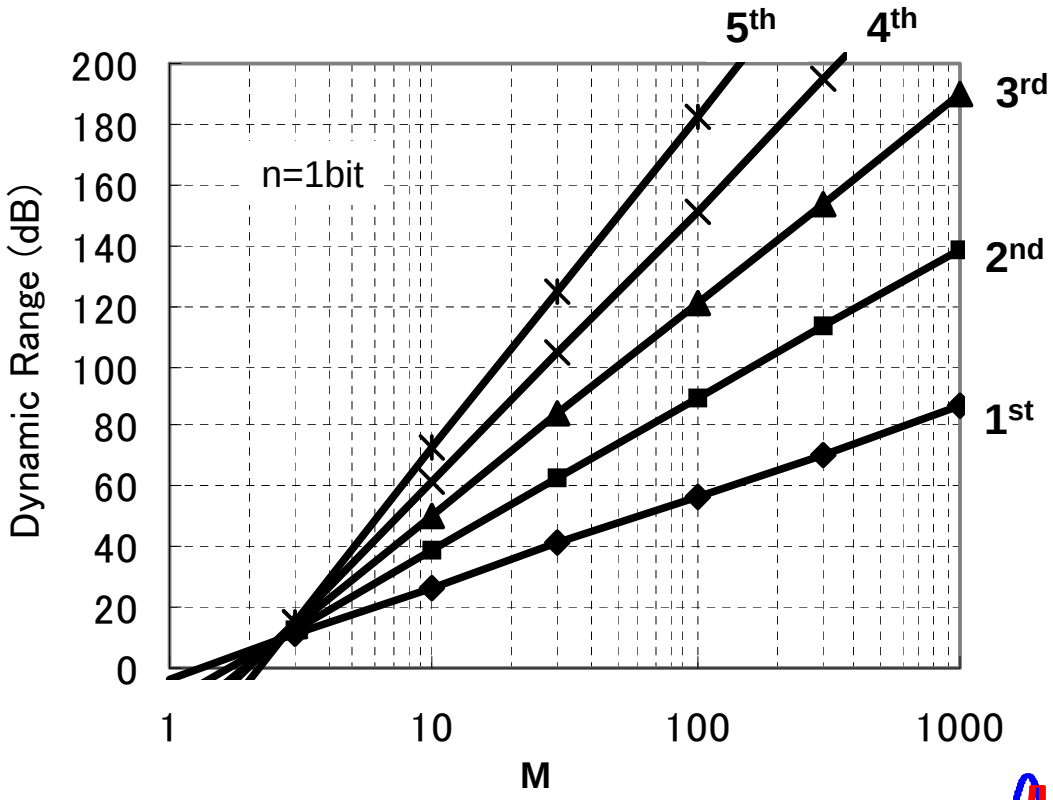
Matsuzawa & Okada Lab.

ΔΣ変調とSNR

ΔΣ変調では積分次数Lとオーバーサンプリング比Mを上げればSNRは上がる。
しかし、、、、、

$$SNR = \frac{3\pi}{2}\left(2^N - 1\right)^2 (2L+1)\left(\frac{M}{\pi}\right)^{2L+1}$$

N：比較器の分解能
L：積分次数
M: オーバーサンプリング比



Δ Σ 型ADCの設計

Δ Σ 型ADCは負帰還技術を用いているので、以下のシステム検討が必要である。

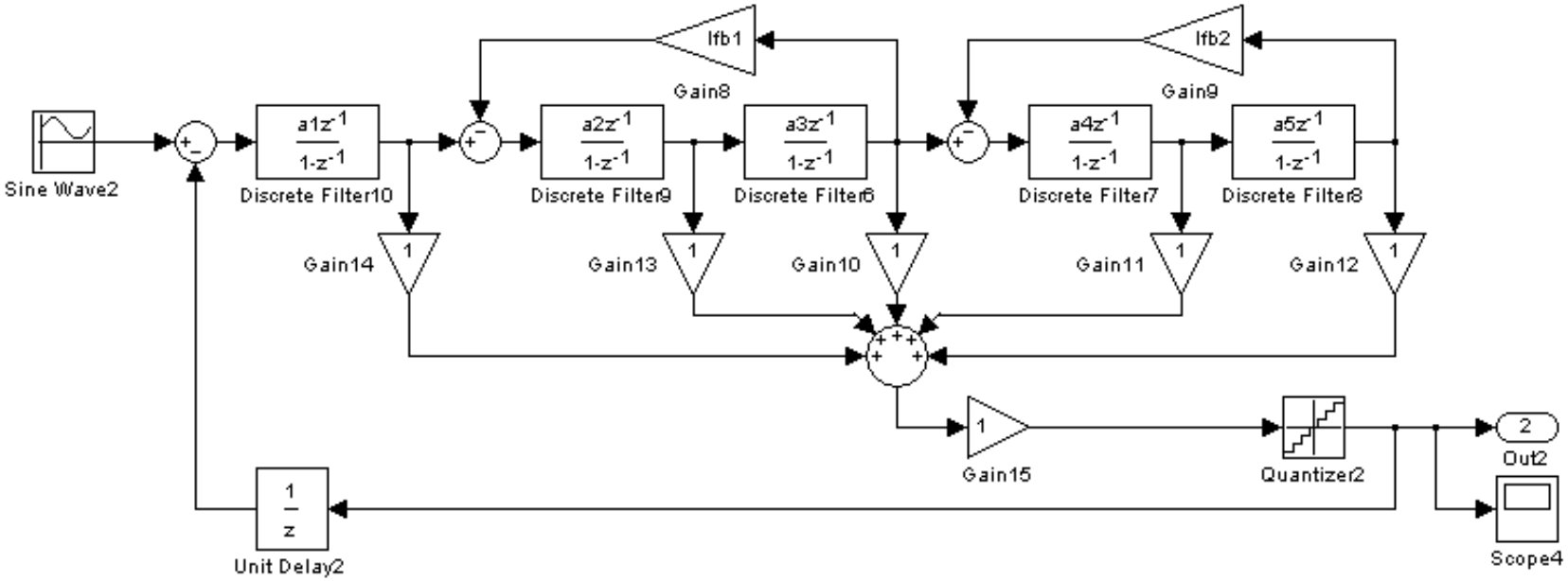
安定動作： ポール位置の確認
周波数特性：NTF, STFの確認
積分器の飽和：各部の信号が飽和しないこと
基本SNRの確認

→基本的に利得定数の設定

演算器：利得、GBW
比較器：量子化ノイズ、遅延
DAC：誤差

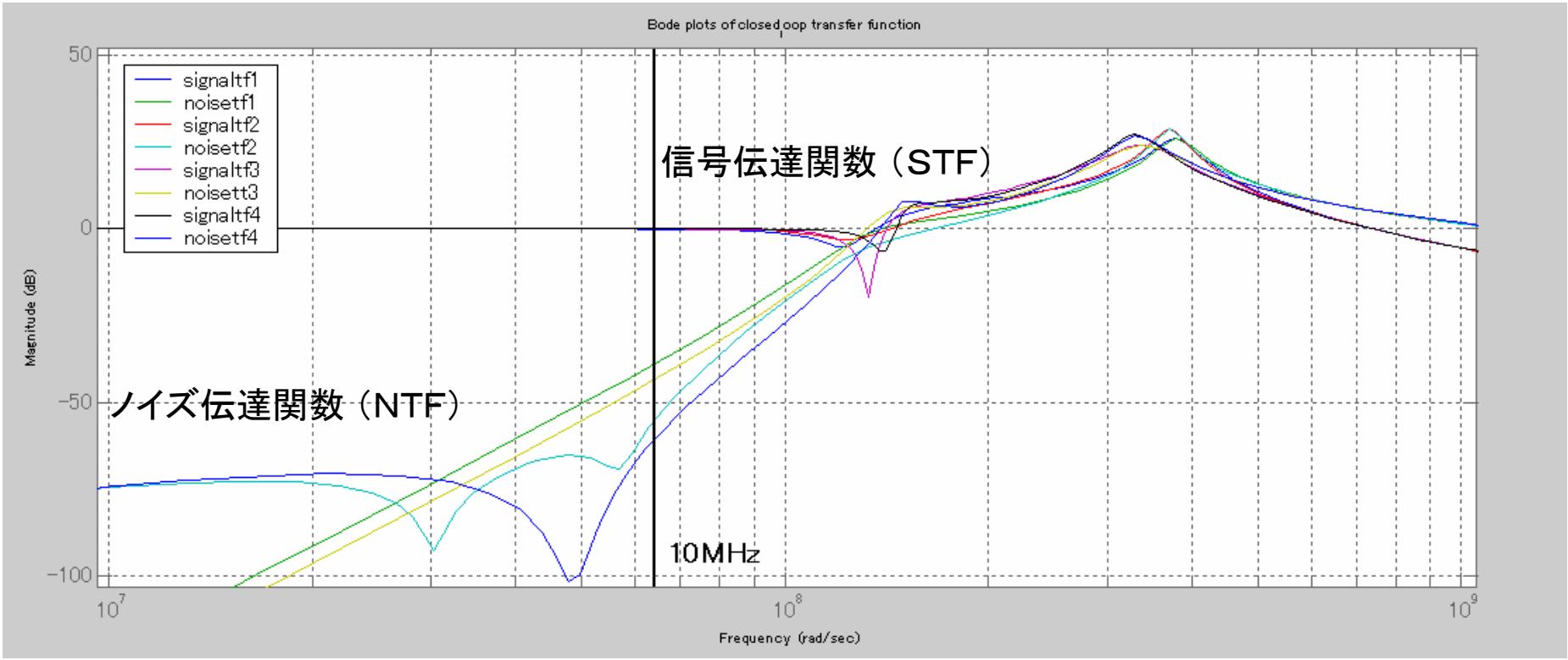
→利得と誤差で検討可能

MatLabが最適



周波数特性

STF: ピークが許容範囲か？
NTF : 低域で十分に減衰しているか？



周波数特性と安定性確認

TOKYO TECH Pursuing Excellence

- ボード線図と極・ゼロ点プロットを描くコマンド (Control System Toolbox)

```
>> A = zpk([-1],[-2 -3],10)
零点/極/ゲイン:
 10 (s+1)
-----------
(s+2) (s+3)

>> bode(A)
>> pzmap(A)
>> [poles zeros] = pzmap(A)
poles =
    -2
    -3
zeros =
    -1
>> range = 2*pi*10.^[5.0:0.01:8.5];
>>[mag,phase,w]=bode(A,range);
```

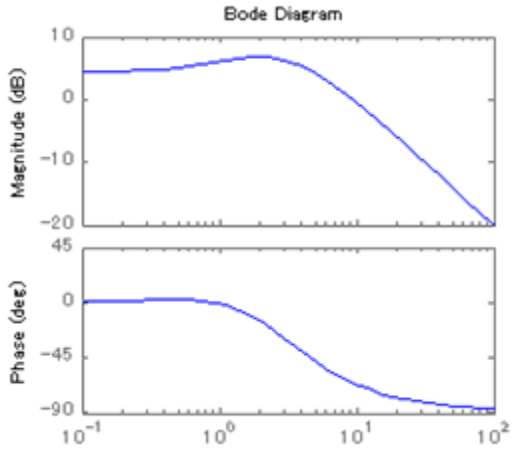
ゼロ点，極，ゲインからの伝達関数作成

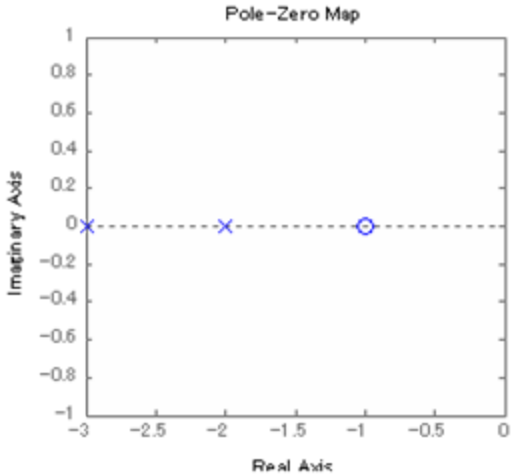
ボード線図の描画

極・ゼロ点をプロット

極・ゼロ点を導出

ボード線図の値を行列で出力



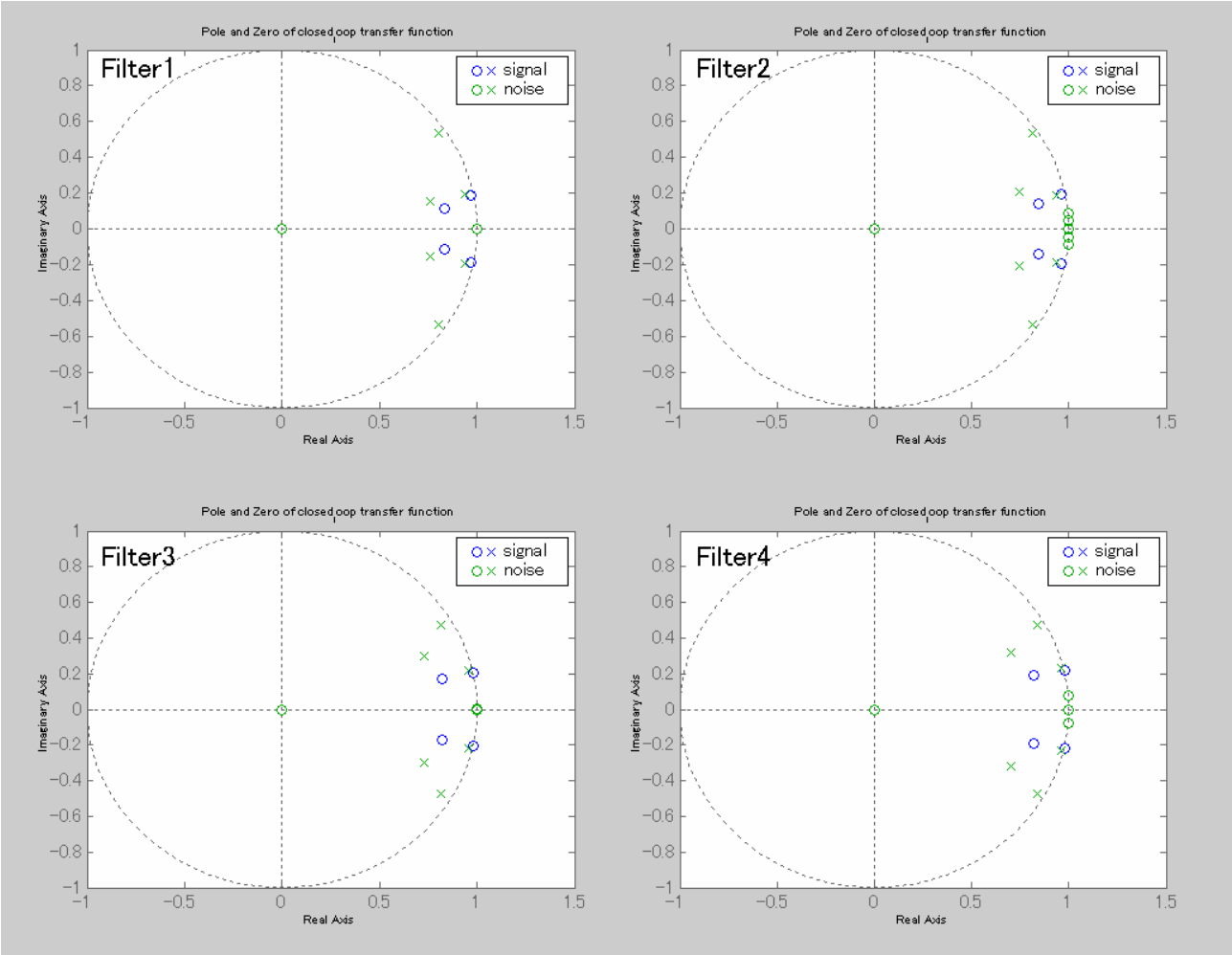


Matsuzawa & Okada Lab.

安定性 （極・ゼロ点）

TOKYO TECH Pursuing Excellence

ポール (x)：単位円の内側にあること
ゼロ(○)：NTFは単位円上にあること

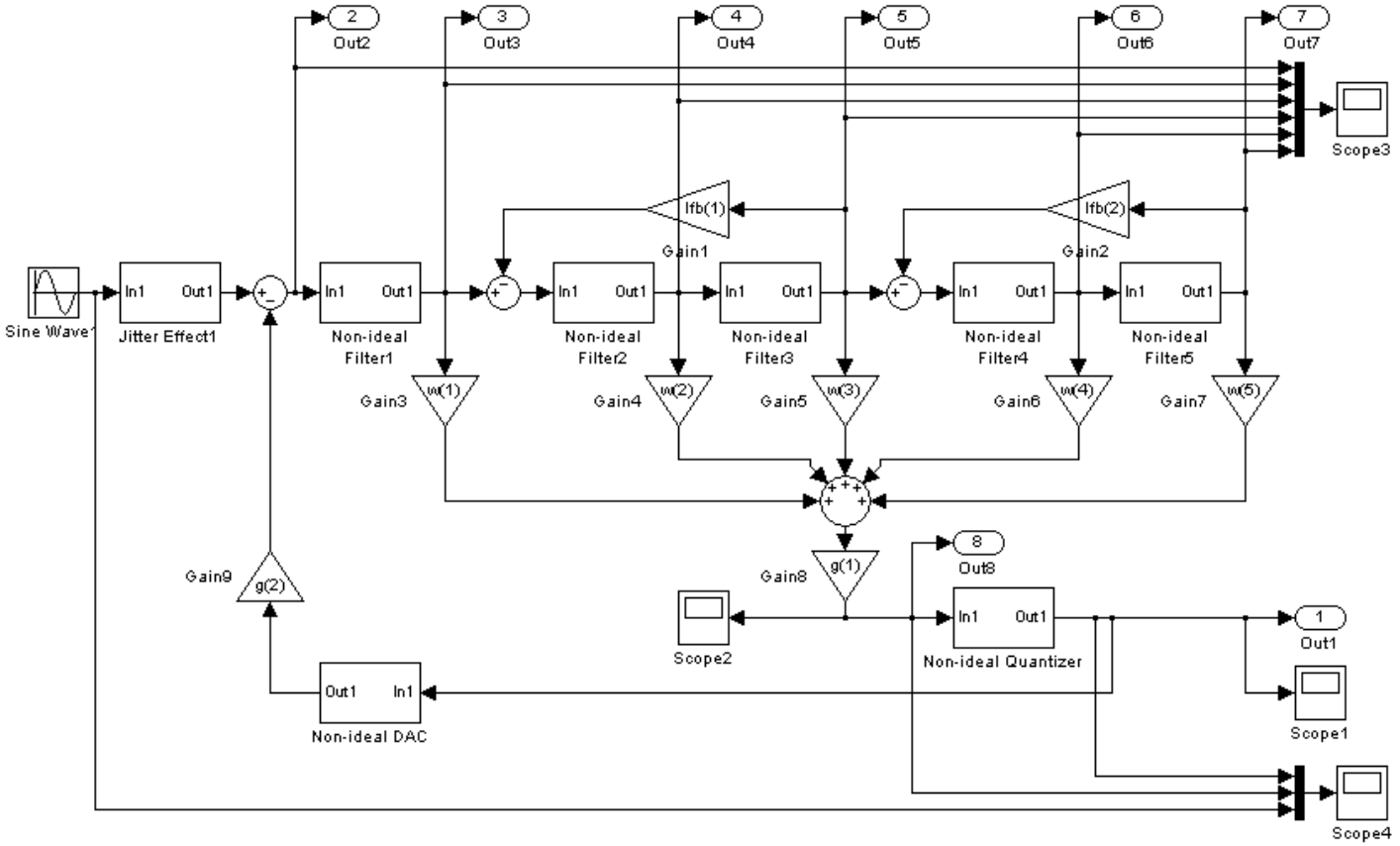


Matsuzawa & Okada Lab.

積分器の飽和の確認

伝達関数が同じでも各積分出力振幅は異なる。

伝達関数を変えずに、各利得定数を変化させて、各積分器の出力をモニターし、ヒストグラムを取る。

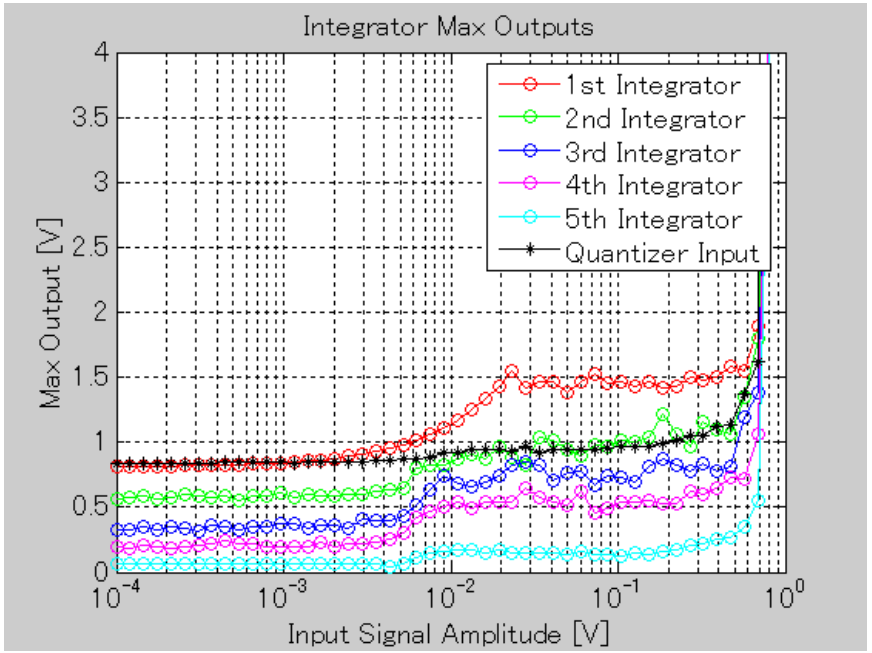


スケーリングと積分器の振幅

TOKYO TECH Pursuing Excellence

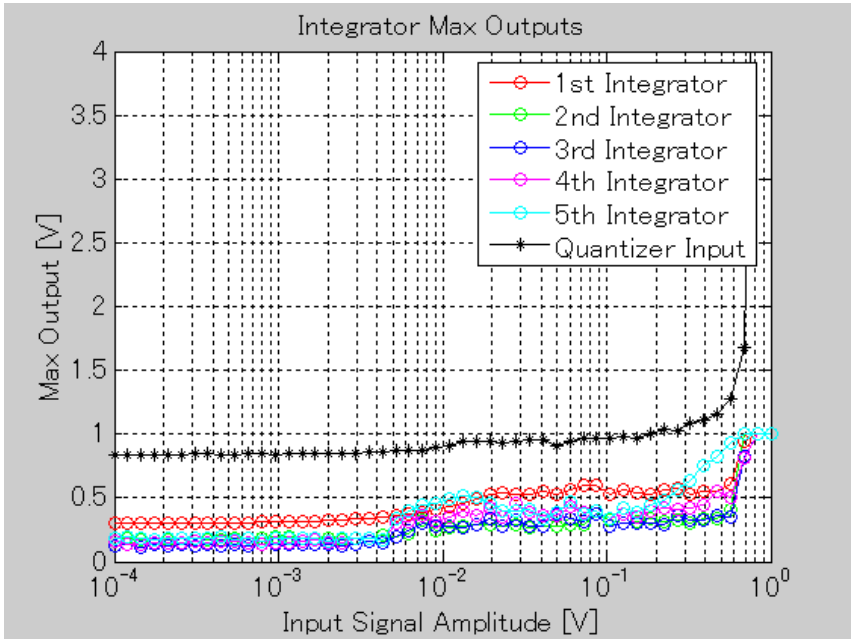
係数を最適化することで伝達関数を変えずに各積分器が飽和しないように構成できる

Scaling前



a5 = 0.8106 0.3861 0.2236 0.1280 0.0401
lfb5 = 0.0177 0.8412
w = 1.0 1.0 1.0 1.0 1.0

Scaling後



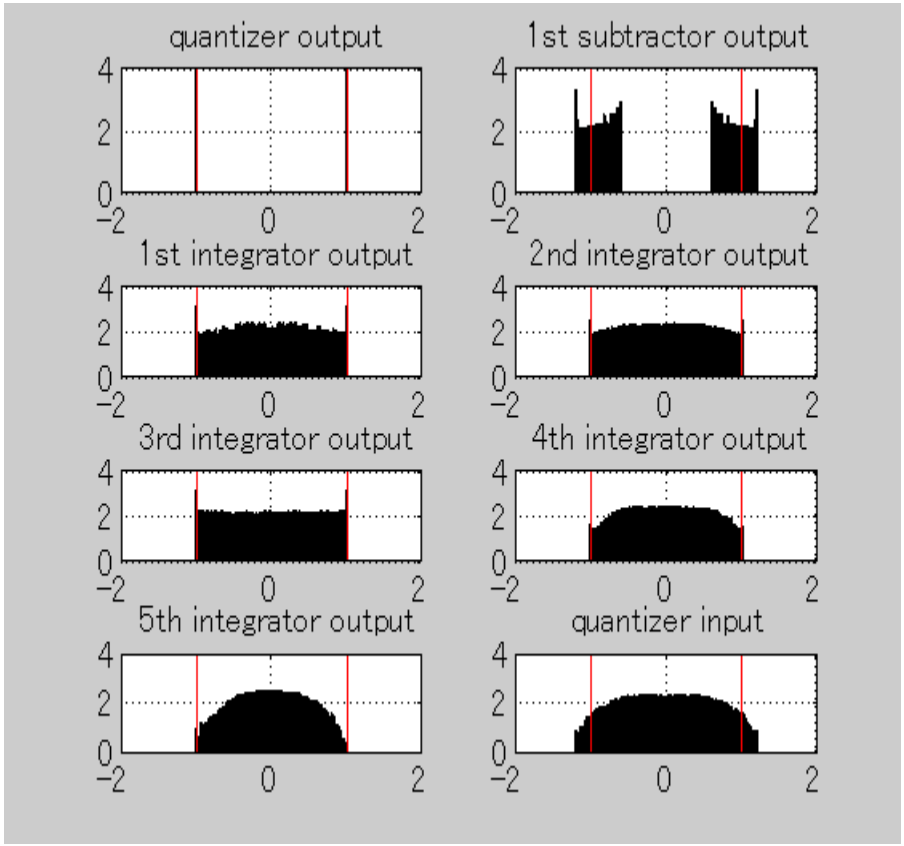
a5 = 0.2996 0.3219 0.2811 0.2417 0.1565
lfb5 = 0.0169 0.1141
w = 2.7054 3.2444 2.5804 1.3670 0.3500

スケーリングの効果

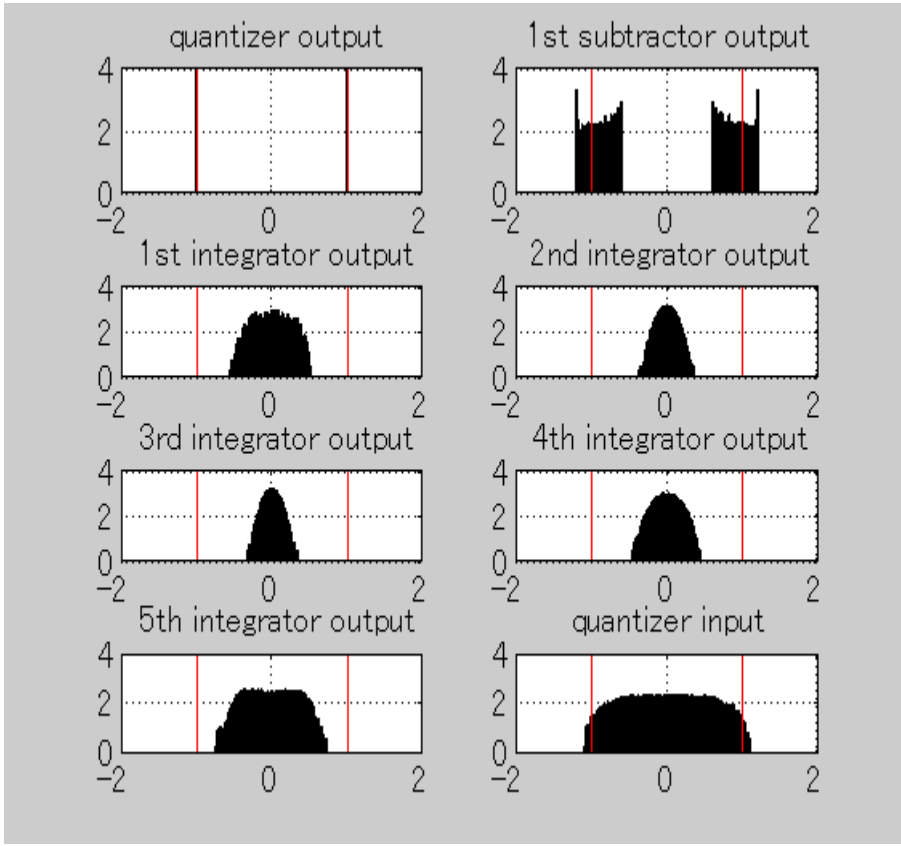
TOKYO TECH Pursuing Excellence

1bit　量子化

スケーリング前(積分器が飽和している)



スケーリング後(積分器が飽和していない)



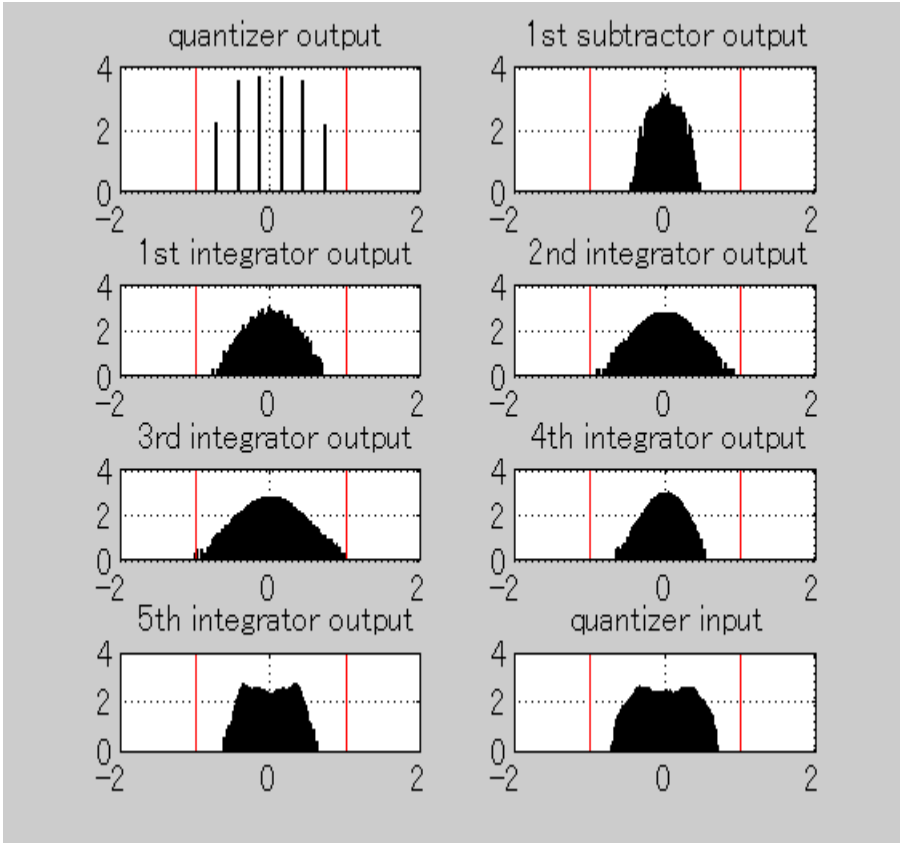
スケーリングの効果

TOKYO TECH Pursuing Excellence

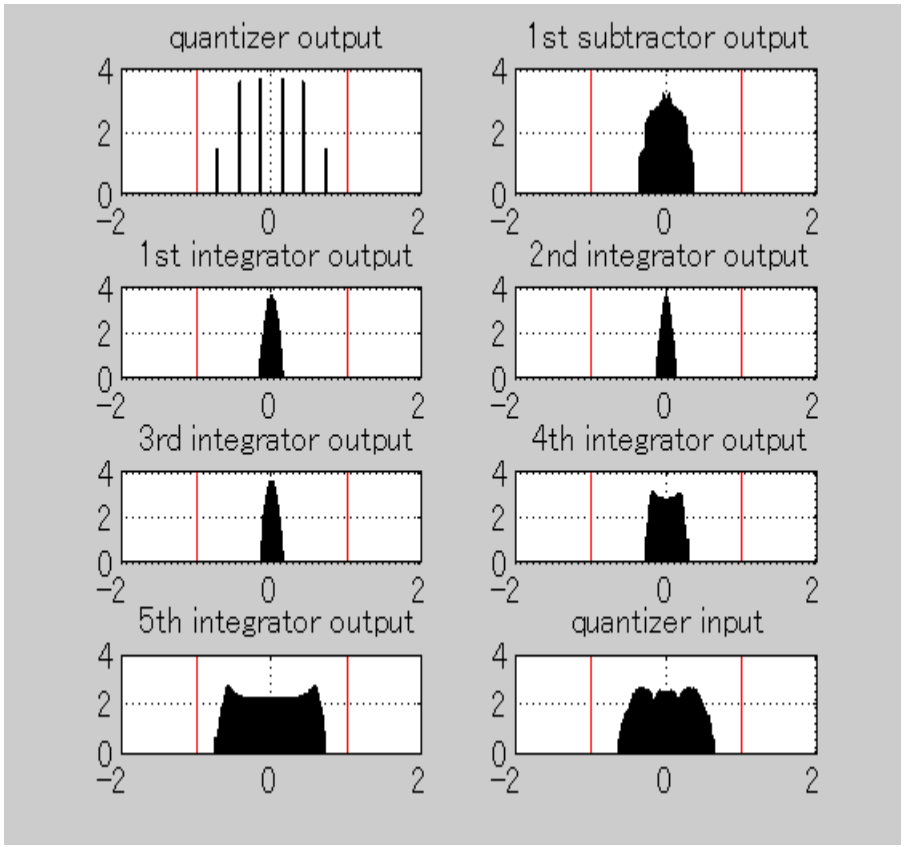
5bit 量子化

5bitの場合は信号の強度分散が小さい

スケーリング前(信号強度の分散が大きい)



スケーリング後(分散が小さい)

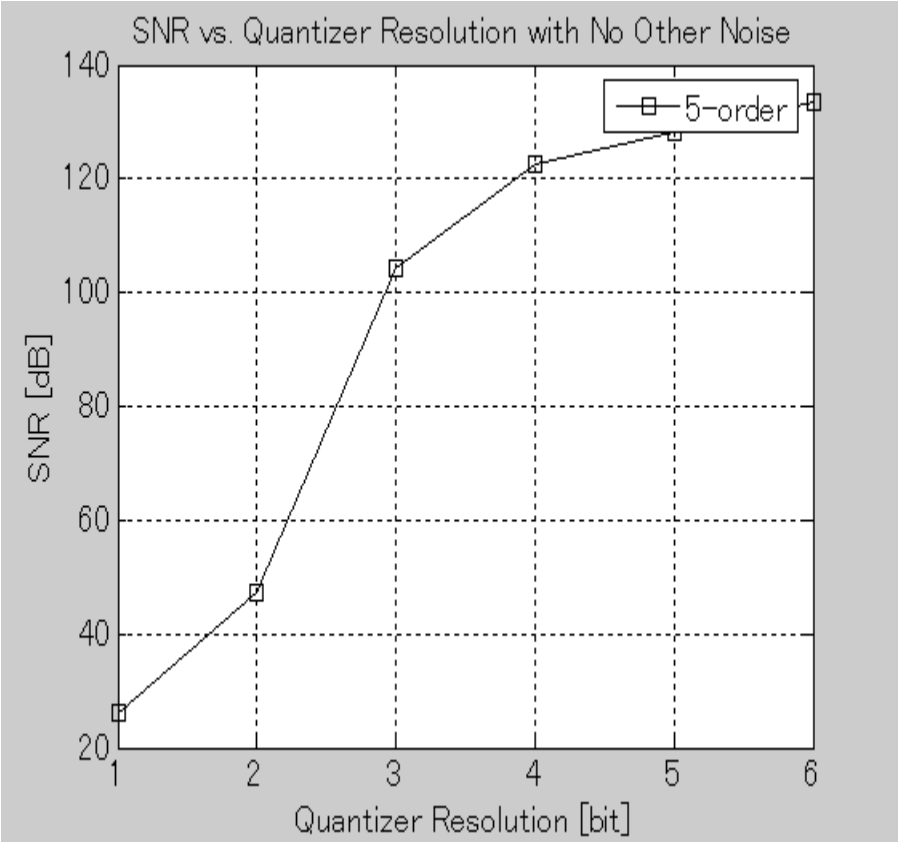


Matsuzawa & Okada Lab.

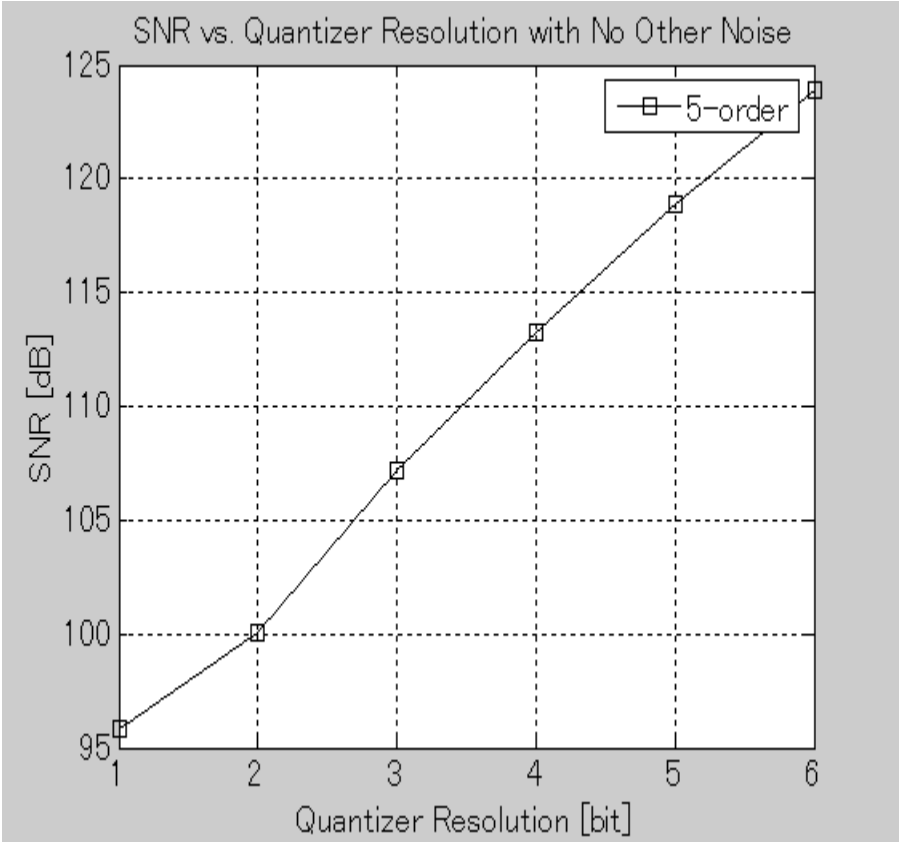
スケーリングの効果

積分器を飽和させると、正常な負帰還が掛からなくなり
実質的にループ利得が大幅低下、SNRが極端に劣化する

スケーリング前(量子化ビットが小さいとSNRが劣化)

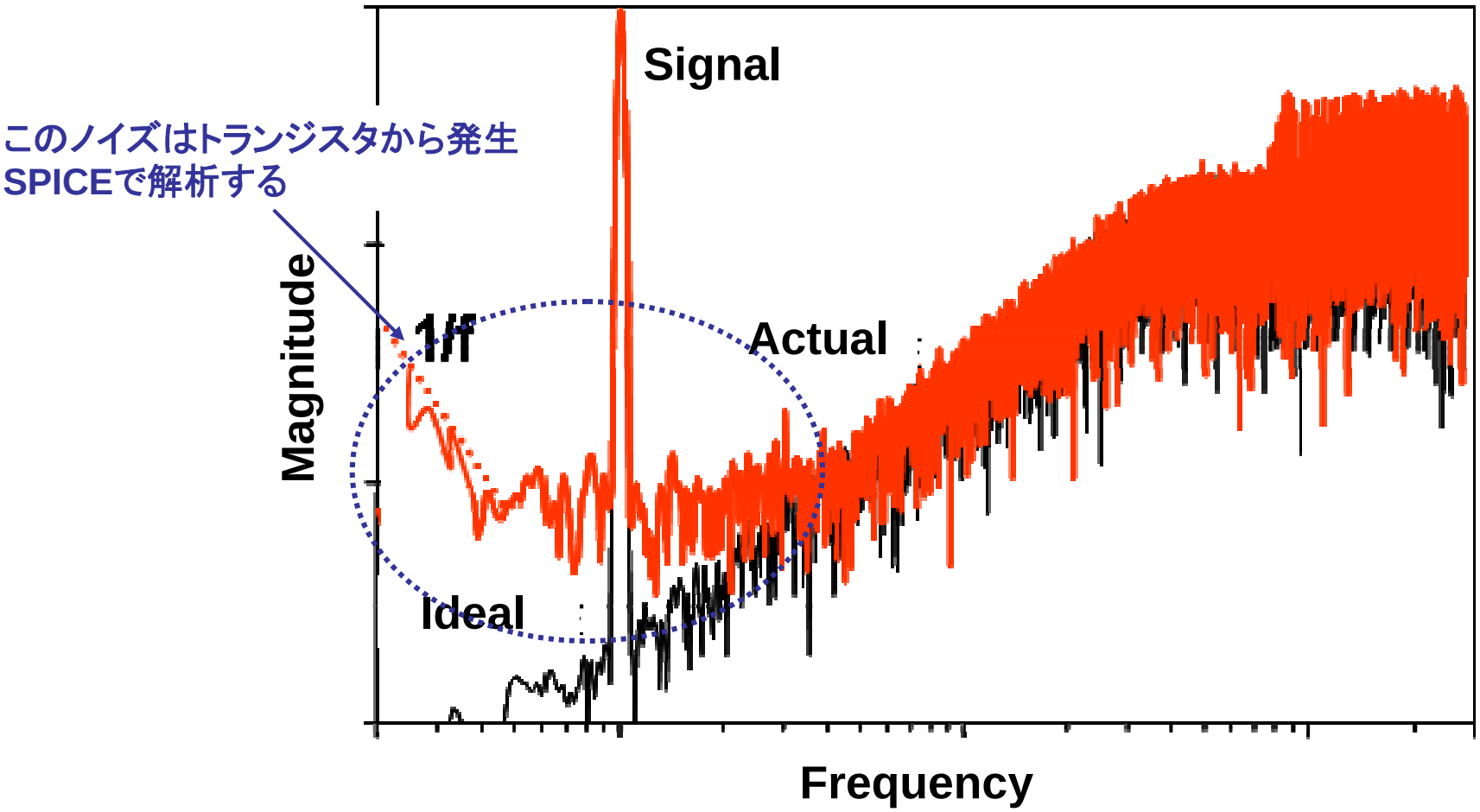


スケーリング後(素直な特性)



回路の熱雑音と1/f 雑音の影響

たとえΔΣ変調技術を用いて量子化雑音を抑圧できたとしても、
回路の熱雑音と1/f 雑音の影響は免れない

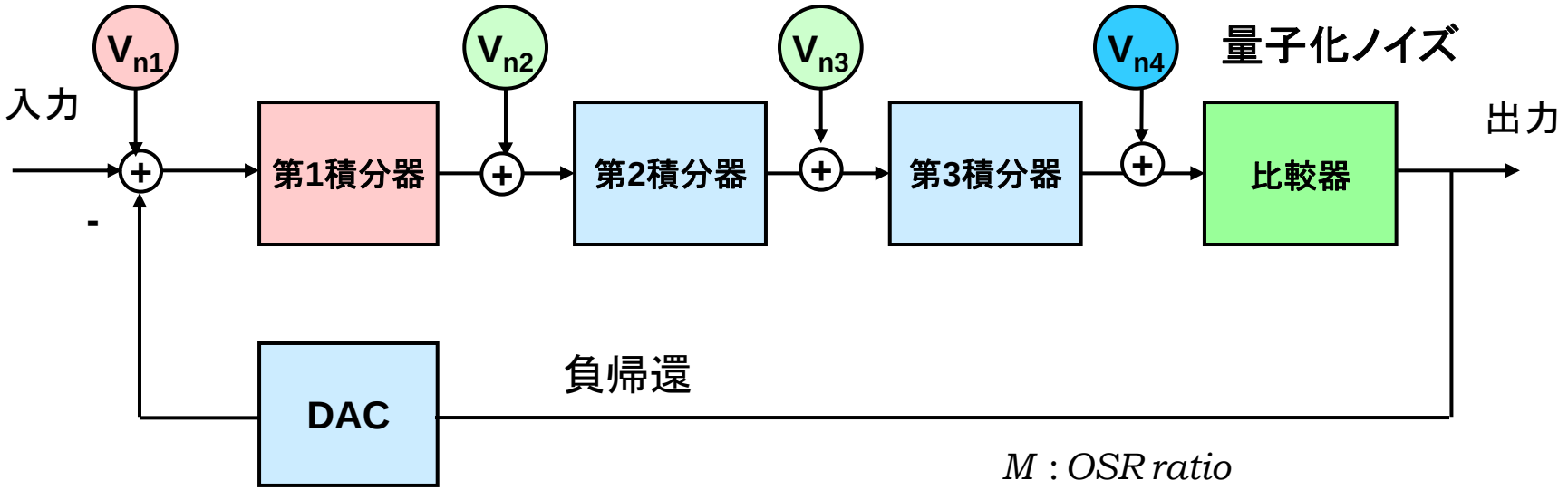


ノイズの寄与

負帰還回路の最後尾で発生する量子化ノイズはΔΣ変調で抑圧できるが、
初段の積分回路の入力換算ノイズは低減できない。

・初段の積分器のノイズ
・S/Hのノイズ
・DACのノイズ

SPICEでノイズシミュレーションを行うべき



$M : OSR\ ratio$

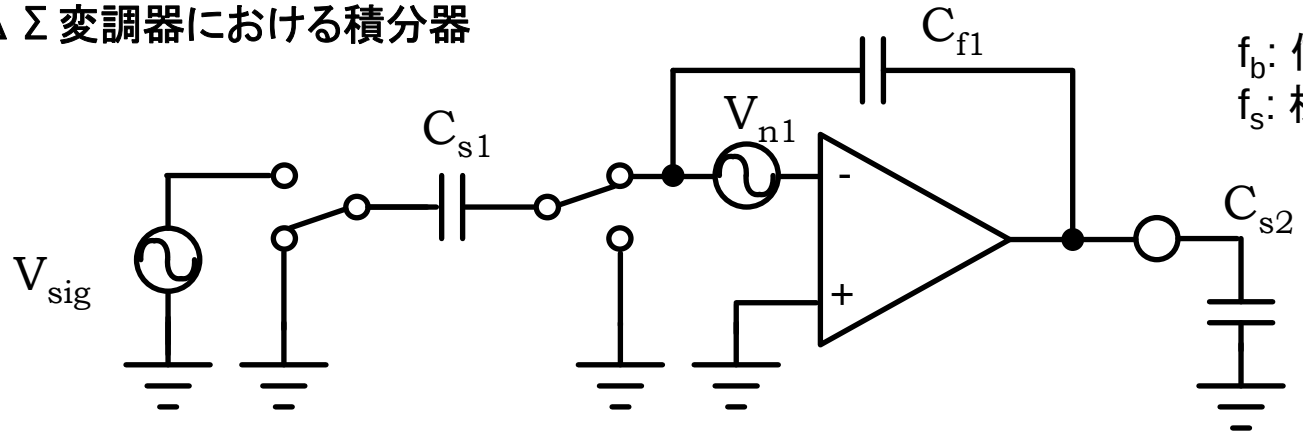
$A_i : Gain\ upto\ i^{th}\ integrator$

$$P_{N\_tot} = \frac{1}{M}\left(P_{N1} + P_{N2}\frac{\pi^2}{3A_2{}^2M^2} + P_{N3}\frac{\pi^4}{5A_3{}^2M^4} + P_{N4}\frac{\pi^6}{7A_4{}^2M^6} \right)$$

増幅器のノイズ

スイッチトキャパシタ回路のノイズ電圧は基本的に容量とオーバーサンプリング比で決まる

例：Δ Σ 変調器における積分器



f_b: 信号帯域
f_s: 標本化周波数

サンプリング時のノイズ: $v_{ns}^2 = \frac{kT}{C_{s1}} \rightarrow \frac{kT}{C_{s1}}\frac{1}{M}$

全体のノイズ $v_{ntot}^2 = v_{ns}^2 + v_{ni}^2 \approx \frac{kT}{C_{s1}M} + \frac{4kT\gamma n}{g_m} f_b$

増幅時のノイズ: $v_{ni}^2 \approx \frac{4kT\gamma n}{g_m} f_b$

$GBW = \frac{g_m}{2\pi C_L} = \frac{g_m}{2\pi\alpha C_{s1}} \geq 4f_s$

$\alpha = \frac{1+a_2}{1+a_1}$

$g_m = 8\pi\alpha C_{s1} f_s \quad @GBW = 4f_s$

γ： ノイズ係数
n： ノイズ源の数

Total noise

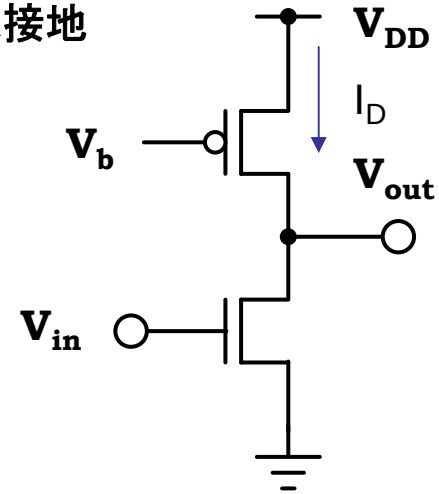
$$v_{ntot}^2 \approx \frac{kT}{C_{s1}M}\left(1 + \frac{\gamma n}{4\pi\alpha}\right)$$

a_1: gain of 1st stage
a_2: gain of 2nd stage

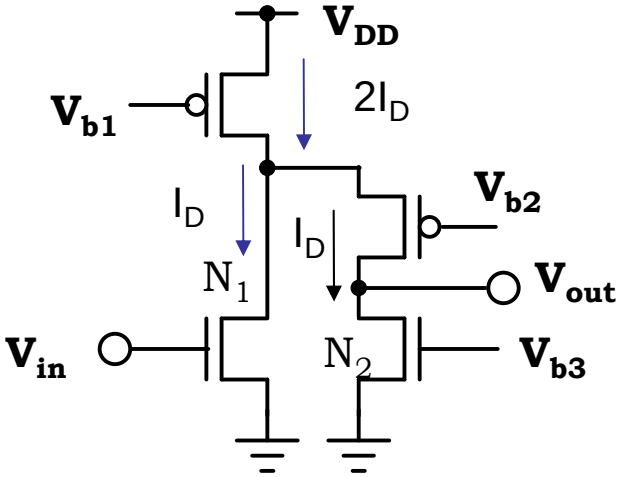
OpAmpのノイズ (1)

回路によって単位電流あたりのノイズ電力が異なる

(1) ソース接地



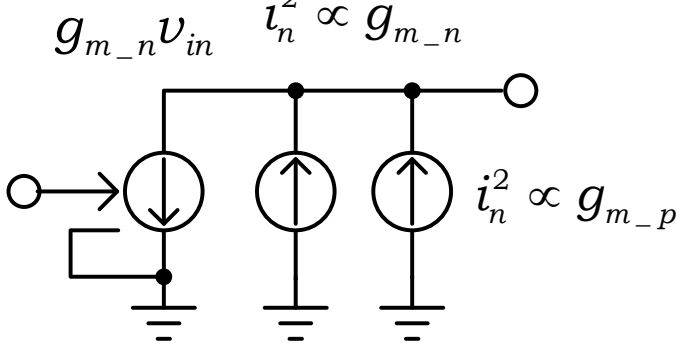
(2) フォールデッドカスコード

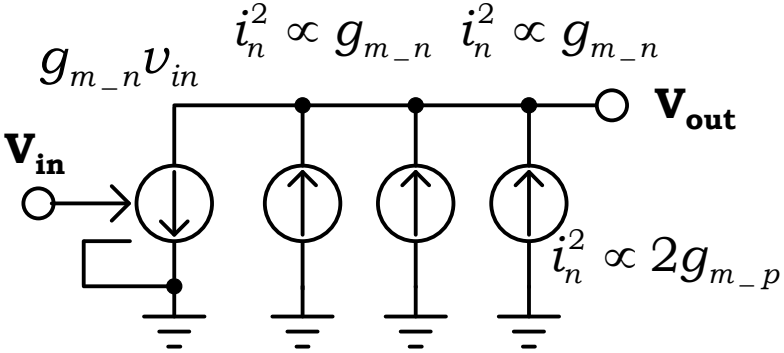


ノイズFoMをノイズ電力と動作電流の積とする

簡単化のためにシングルエンドで記述

仮定：g_m/I_D が等しい





$$FoM \equiv \frac{v_{ni}^2 I_D}{4kT}$$

$$v_{ni}^2 \approx 2 \times \frac{4kT}{g_{m\_n}}$$

$$FoM = V_{eff}$$

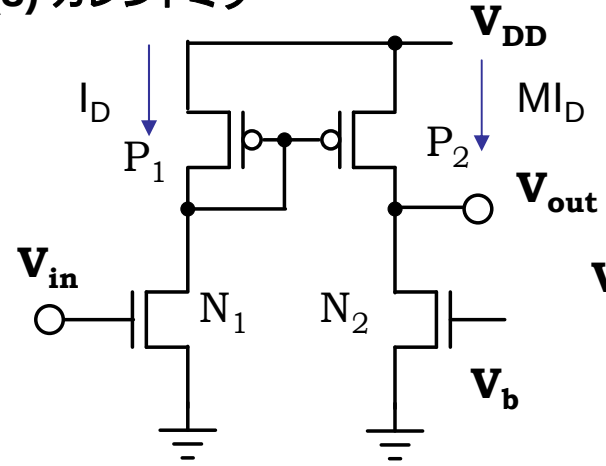
$$v_{ni}^2 \approx 4 \times \frac{4kT}{g_{m\_n}}$$

$$FoM = 4V_{eff}$$

OpAmpのノイズ(2)

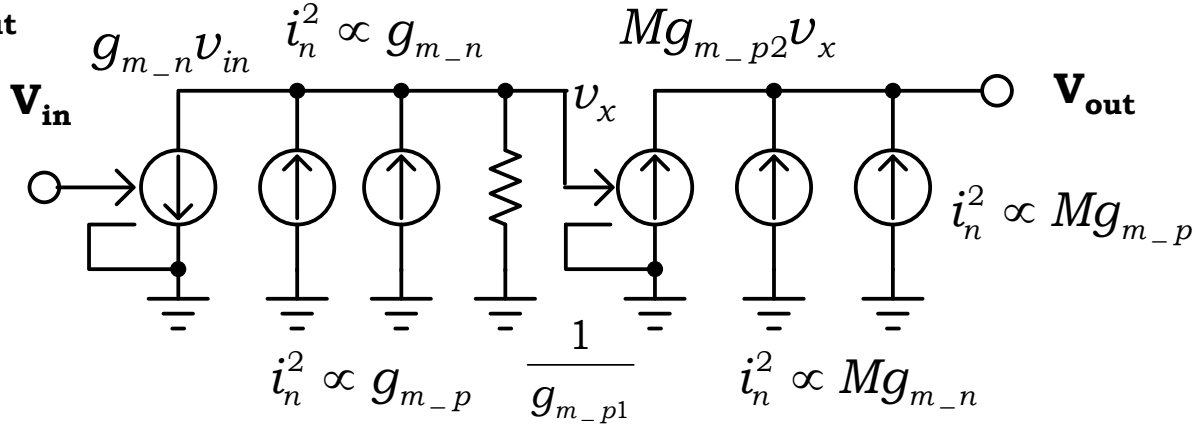
回路形式により、同一ノイズ電力でも、動作電流は16倍の差が出る。

(3) カレントミラー

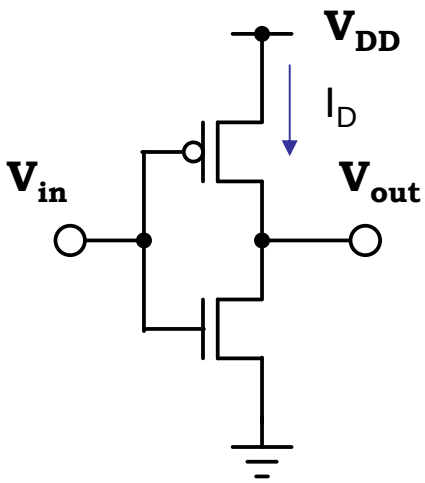


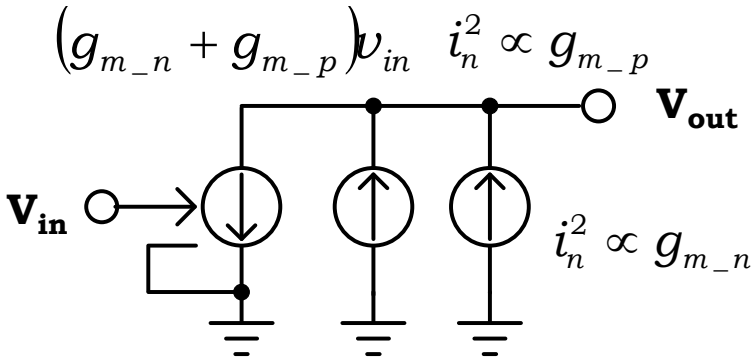
$$v_{ni}^2 \approx 2\left(1+\frac{1}{M}\right)\times\frac{4kT}{g_m}$$

$$FoM = \frac{(1+M)^2}{M}V_{eff}$$



(3) 相補型





$$v_{ni}^2 \approx \frac{1}{2}\times\frac{4kT}{g_m}$$

$$FoM = \frac{V_{eff}}{4}$$

MATLAB+Simulinkを用いたノイズSim

TOKYO TECH Pursuing Excellence

```
% ノイズを再現した2出力のsimulinkモデルからPSDを表示するスクリプトM-file

Fs  = 640e6;       % サンプリング周波数
Ts  = 1/Fs;        % サンプリング周期(Simulinkはベースワークスペース内の変数を参照できる)
N   = 16384;       % FFTの長さ
simtime = Ts*N*1.2;     % シミュレーション時間(最小限の時間の1.2倍)
TotalNoise = 1000;      % Simulink内で使う変数  全ノイズの大きさ
WhiteNoise = 1;         % Simulink内で使う変数  ホワイトノイズの大きさ

[t,x,y] = sim('NS',simtime);    % Simulinkモデル NS.mdl をsimtimeだけ動かし, 出力をyに格納

output1 = y(end-N+1:end,1);   % out1ブロックの出力はyの1列目, out2ブロックの出力はyの2列目
output2 = y(end-N+1:end,2);   % end-N+1:end,2 で2列目の最後からN行分を指定し別の変数に代入

psd1 = 2*((abs(fft(output1,N))/N).^2)/(Fs/N);   % 関数fft()を用いてPSDを計算(結果も行列)
psd2 = 2*((abs(fft(output2,N))/N).^2)/(Fs/N);   % .^ で行列の要素ごとのべき乗になる
f    = Fs*(0:(N/2-1))/N;                        % ナイキスト周波数までの横軸行列の作成

loglog(f,psd1(1:(N/2)),'b',f,psd2(1:(N/2)),'r');  % 両対数スケールで2つのPSDをプロット
                                                  % 'b'は青,'r'は赤
ylabel('PSD [V^2/Hz]');
xlabel('Frequency [Hz]');            % タイトルや軸はプロット後にメニューからも編集できる
title('Power Spectrum Density of Noise Sources');
legend('PSD of Out1','PSD of Out2');       % プロットの順に凡例の設定
grid on;                                   % グリッドの表示
axis([Fs/N Fs/2 1e-10 1e4]);               % プロット範囲の指定

sum(psd1)*(Fs/N)/2  % 全ノイズの大きさの確認(文の最後に;がない場合は計算結果が表示される)
mean(psd2)          % ホワイトノイズの大きさの確認
```

MATLABからSimulinkモデルを動作させ，さらにその出力を処理する例

Simulinkによるノイズのモデリング

TOKYO TECH Pursuing Excellence

- Total Noise と White Noise のモデル化

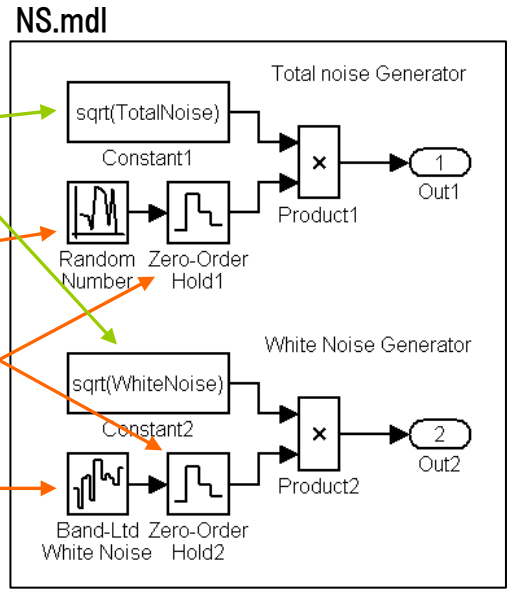
パワーを設定しているので
ルートをとってかける

ダブルクリックして分散を1,
サンプル時間をTsに設定

サンプル時間をTsまたは -1
(前のブロックと同じ)に設定

ノイズパワーを0.5,
サンプル時間をTsに設定

ノイズ源を複数使うときは,
相関をなくすため, Seedを
別々にする

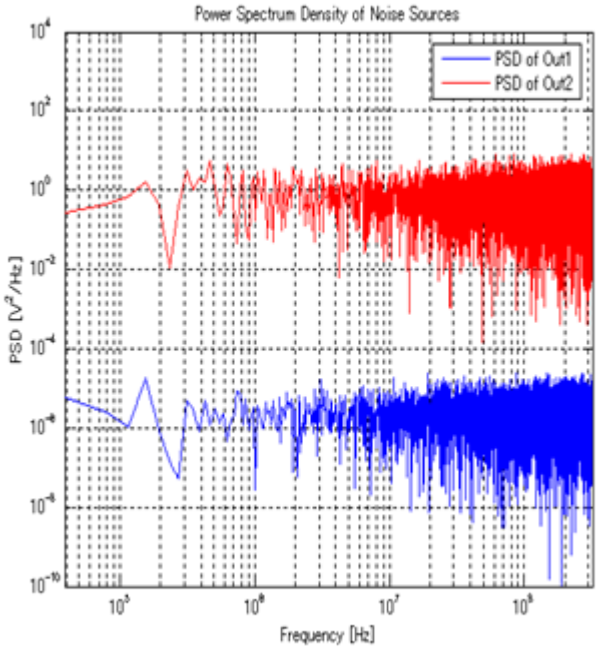


Total Noise はナイキスト周波数まで
積分したノイズパワーを設定
(kT/Cノイズのモデルに使用)
White NoiseはPSDを直接設定
(熱雑音のモデルに使用)

- M-fileの実行結果

ほぼM-file内で設定
した通りの大きさの
PSDが得られている
ことを確認

```
>> NSexe
ans =
   1.0078e+003
ans =
    0.9983
```



Matsuzawa & Okada Lab.

1/f ノイズのモデリング

TOKYO TECH Pursuing Excellence

- White Noise モデルと LPF を使った1/f ノイズのモデル化

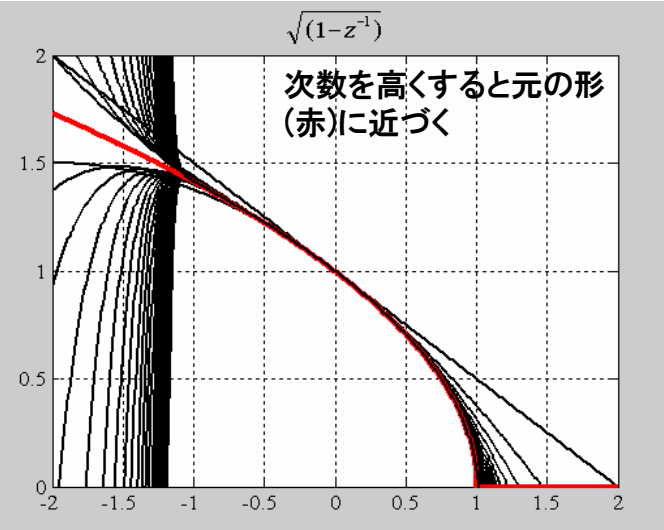
白色ノイズに1次のLPFをかけると，周波数10倍につき10dBでなく20dB落ちてしまう
傾きを1/f にするには伝達関数をルートにしなくてはならない

LPFの分母の $\sqrt{(1-z^{-1})}$ をテーラー展開

・低域でスペクトルが平らになる
・テーラー展開の次数を増やすと改善されるが限界がある

各次の項の式（n>1）

$$coeff(n) = -\frac{1}{2^n}\cdot\frac{1}{n!}\cdot\prod_{i=2}^{n}(2i-3)\cdot z^{-n}$$

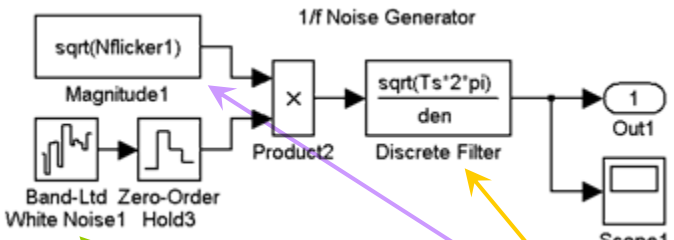


テーラー展開のM-file（係数の配列を出力）

```
function [coeff] = Taylor_01(order)
  % Taylor expansion of (1-x)^0.5

  clear coeff

  coeff(1) = 1;
  coeff(2) = -0.5;
  for i = 3:(order+2)
    coeff(i) = coeff(i-1)/2/(i-1)*(2*(i-1)-3);
  end
  coeff = coeff(1:order);

  % check
  x = -2:0.01:2;
  y = 0;
  for i = 1:order
    y = y + coeff(i)*(x.^(i-1));
  end

  plot(x,(1-x).^0.5,x,y);
  grid on
  axis([-2 2 0 2])
```


Matsuzawa & Okada Lab.

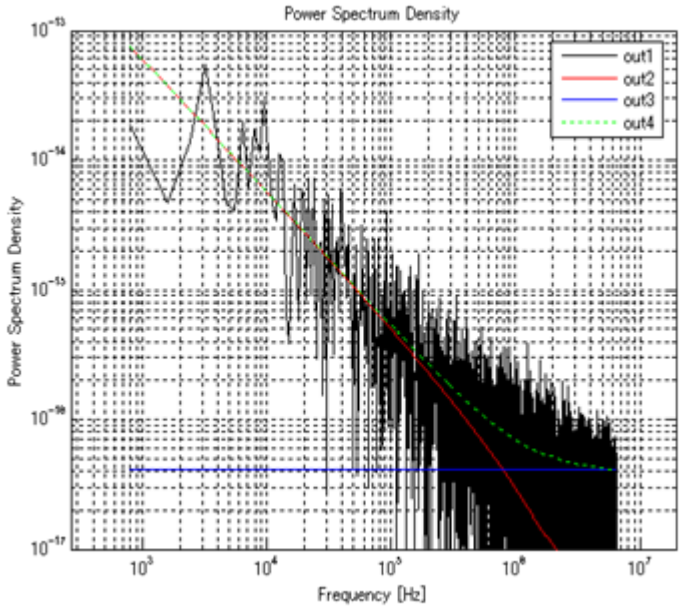
1/f ノイズのモデリング



ノイズパワーを0.5,
サンプル時間をTsに設定

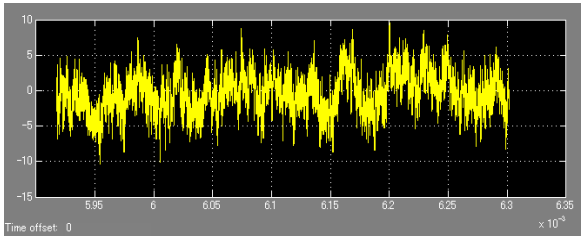
Simulinkモデルのパラメータ設定

```
epsilon = (8.8542149e-12) * 3.9;
toxn    = 5.00e-9;
toxp    = 5.00e-9;
Coxn    = epsilon / toxn;
Coxp    = epsilon / toxp;
Ln  = 0.90e-6;
Lp  = 0.90e-6;
Wn1 = 35*1e-6;
Wp1 = 140*1e-6;
Kn  = 30e-25;
Kp  = 30e-25;
Nflicker1 = (2*Kn/(Wn1*Ln*Coxn)+2*Kp/(Wp1*Lp*Coxp))
% 1/f ノイズの大きさの計算

den = Taylor_01(300)
% 作成した関数M-fileでテーラー展開の係数の計算
```

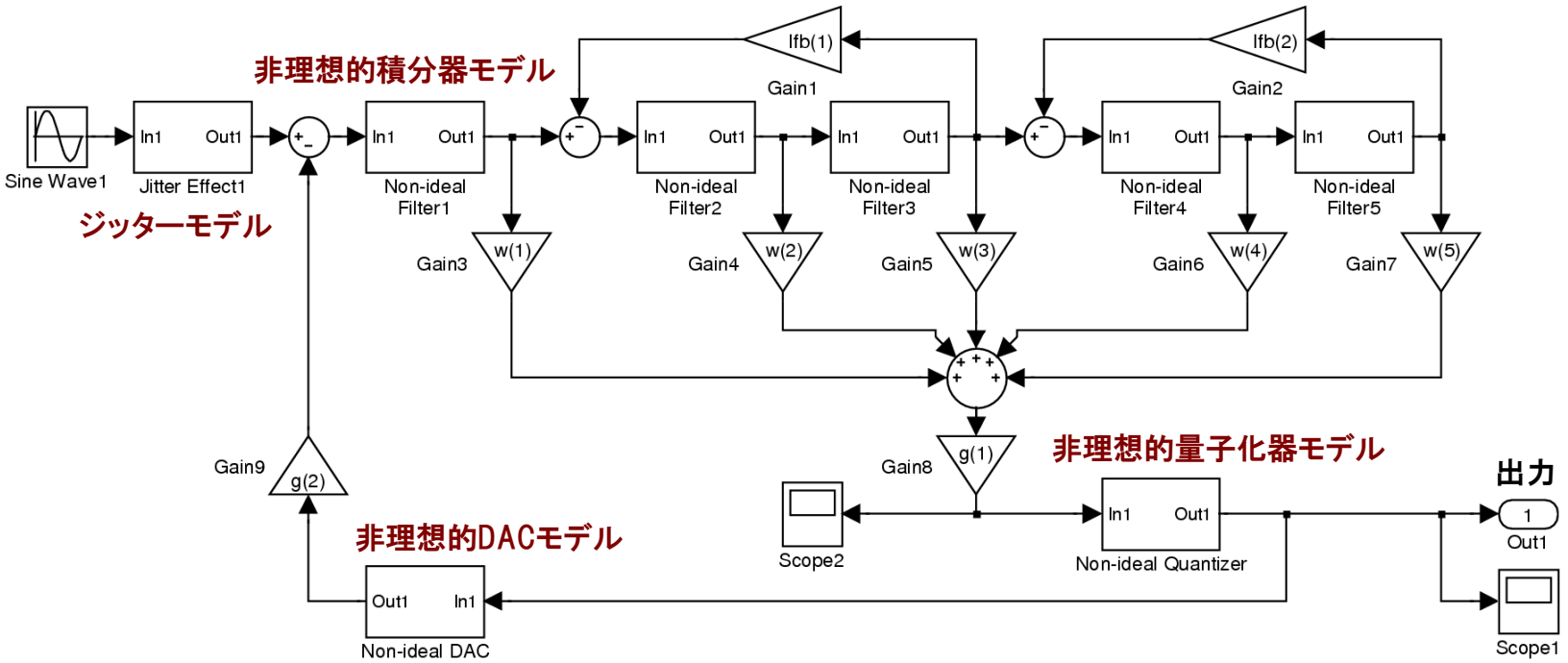


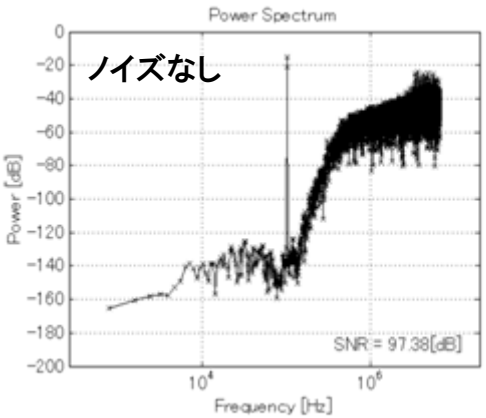
黒:LPFの1/fノイズモデル
赤:逆フーリエ変換のモデル
青:熱雑音レベル
緑:合計

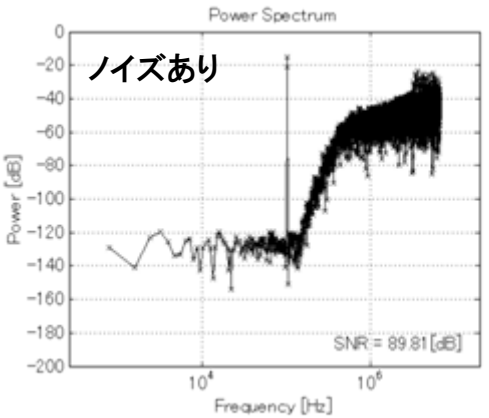
時間波形



非理想的動作を含むΣΔADCモデル





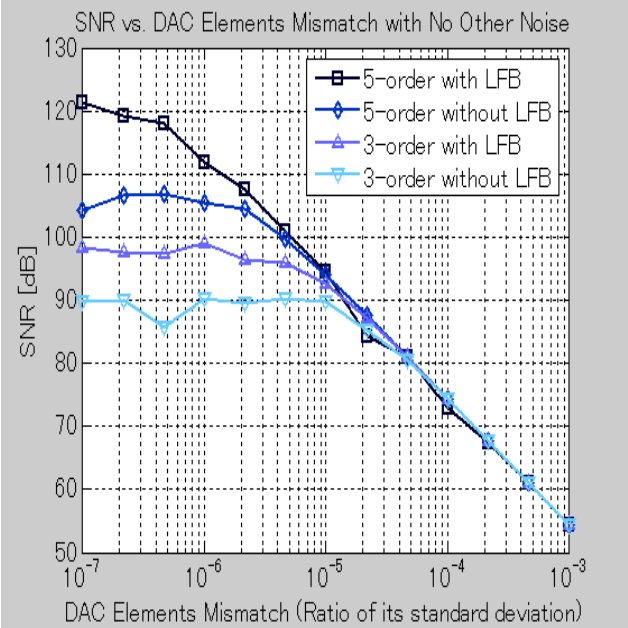


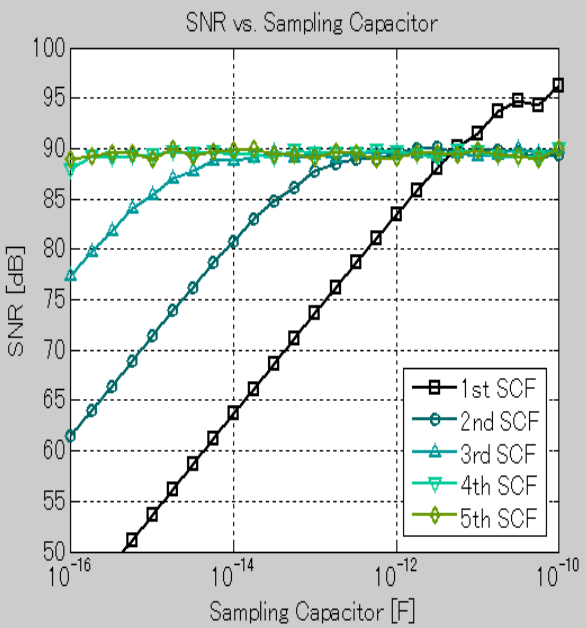
MATLABでこのモデルを動作させ，
出力からパワースペクトルを求め
てSNRを算出する
パラメータを振ってこれを繰り返す

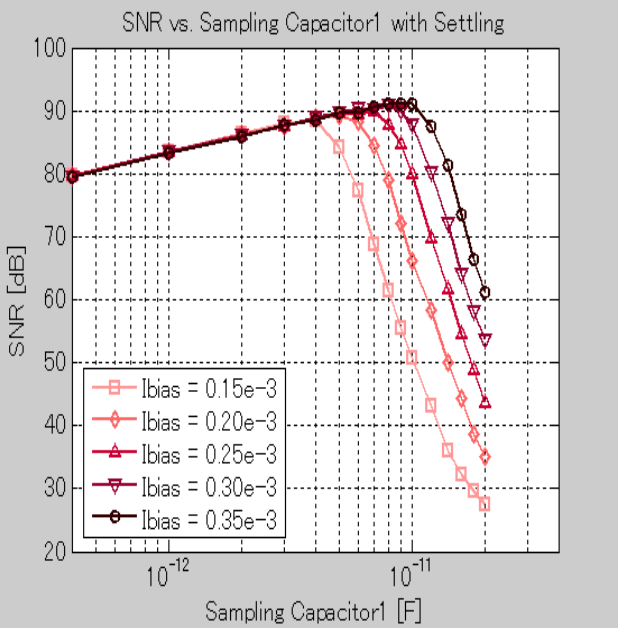
MATLAB / Simulink Simulation Results

TOKYO TECH Pursuing Excellence

Non-ideality を考慮し，パラメータを変えてシミュレーションを行った結果の例







DAC素子のミスマッチ vs. SNR

DAC素子のミスマッチがノイズ成分となり，これは Noise Shaping されないので影響が大きい
DEM (Dynamic Element Matching) を使うことで改善できる

サンプリング容量 vs. SNR

初段のサンプリング容量によるkT/Cノイズは Noise Shaping されないので影響が大きい
後段になるほどサンプリング容量が小さくできるのでオペアンプの消費電力を低くできる

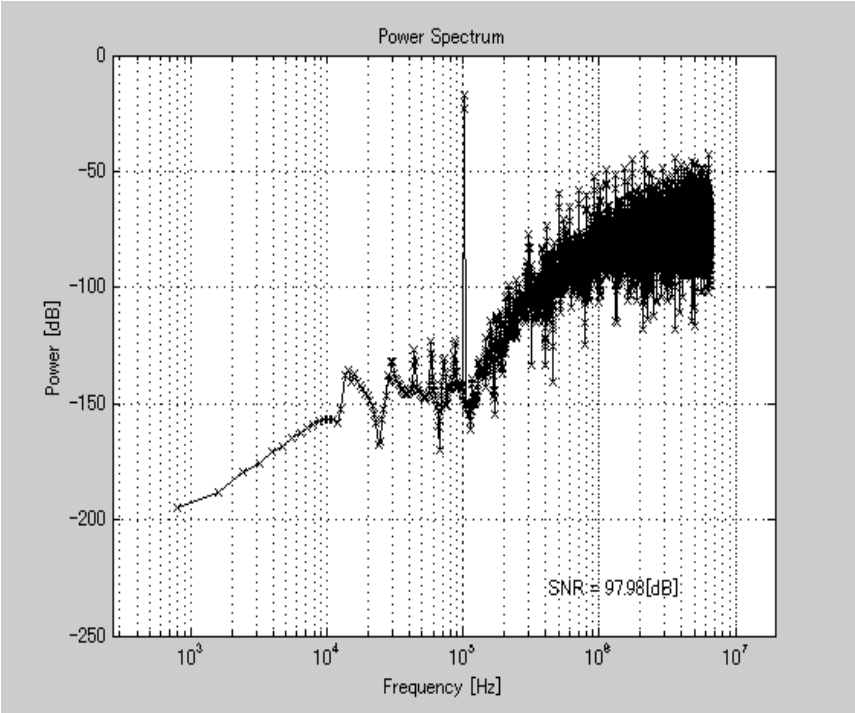
初段サンプリング容量 vs. SNR

サンプリング容量が大きすぎるとsettling の問題が発生し，g_mを大きくするために電流が増える
計算のみで求めたバイアス電流と比較する

スペクトラム

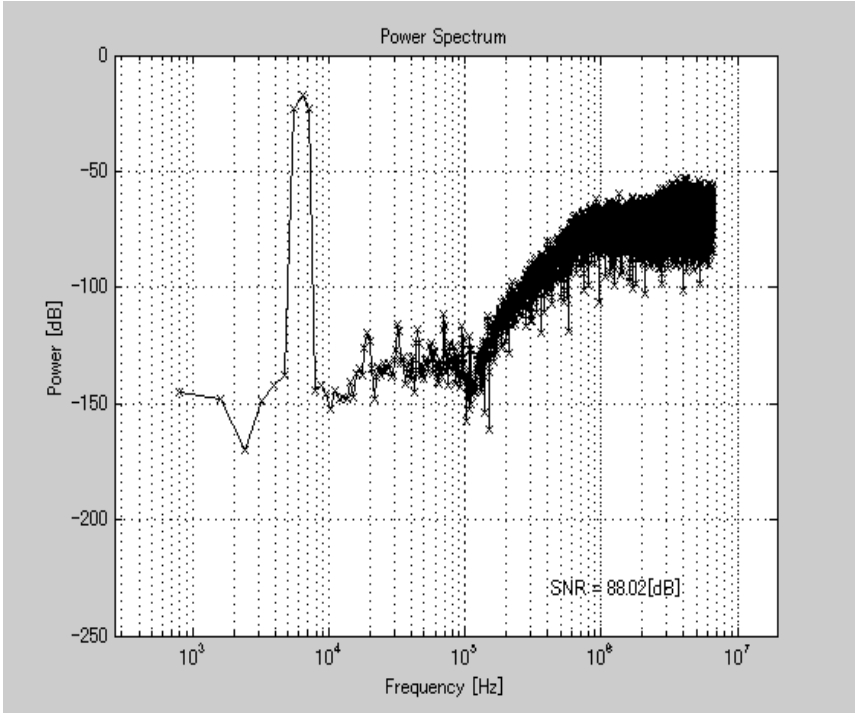
トランジスタノイズを考慮
しないときのスペクトラム

MatLab



トランジスタノイズを考慮
したときのスペクトラム

Verilog-A+SPECTRE



トランジエントノイズ解析

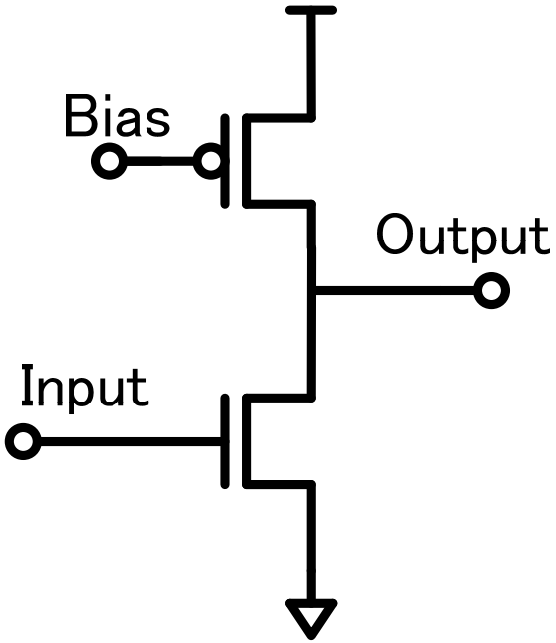
TOKYO TECH Pursuing Excellence

これまでサンプリング系においてはノイズのシミュレーションはできなかった。
最近、SPECTREで可能になった。

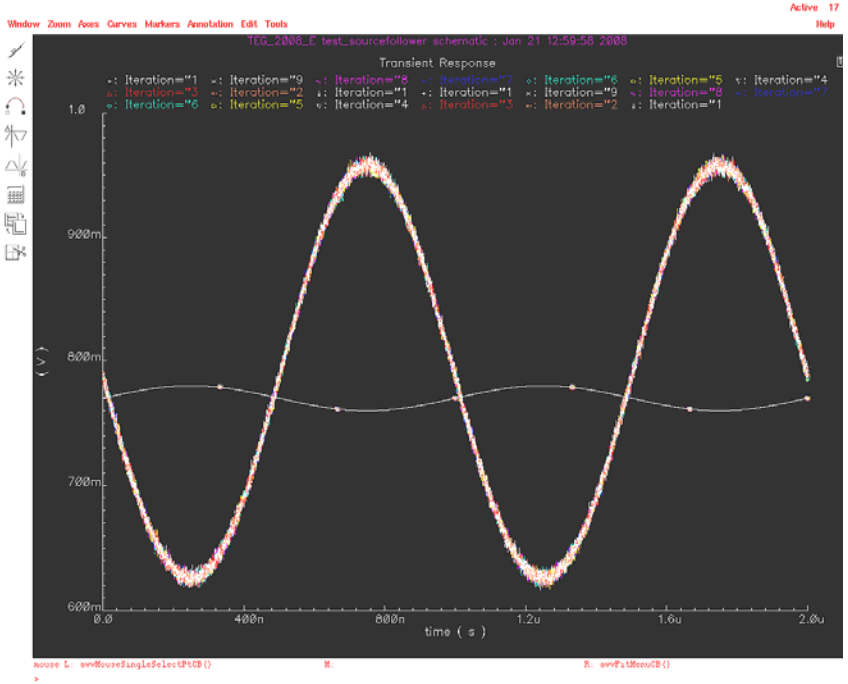
SPECTREにおけるノイズを含んだ過渡解析シミュレーション

SCF回路ではTノイズ解析を行った後、必要に応じてFFTで周波数領域で評価する

・シミュレーション例
ソース接地アンプに正弦波を入力

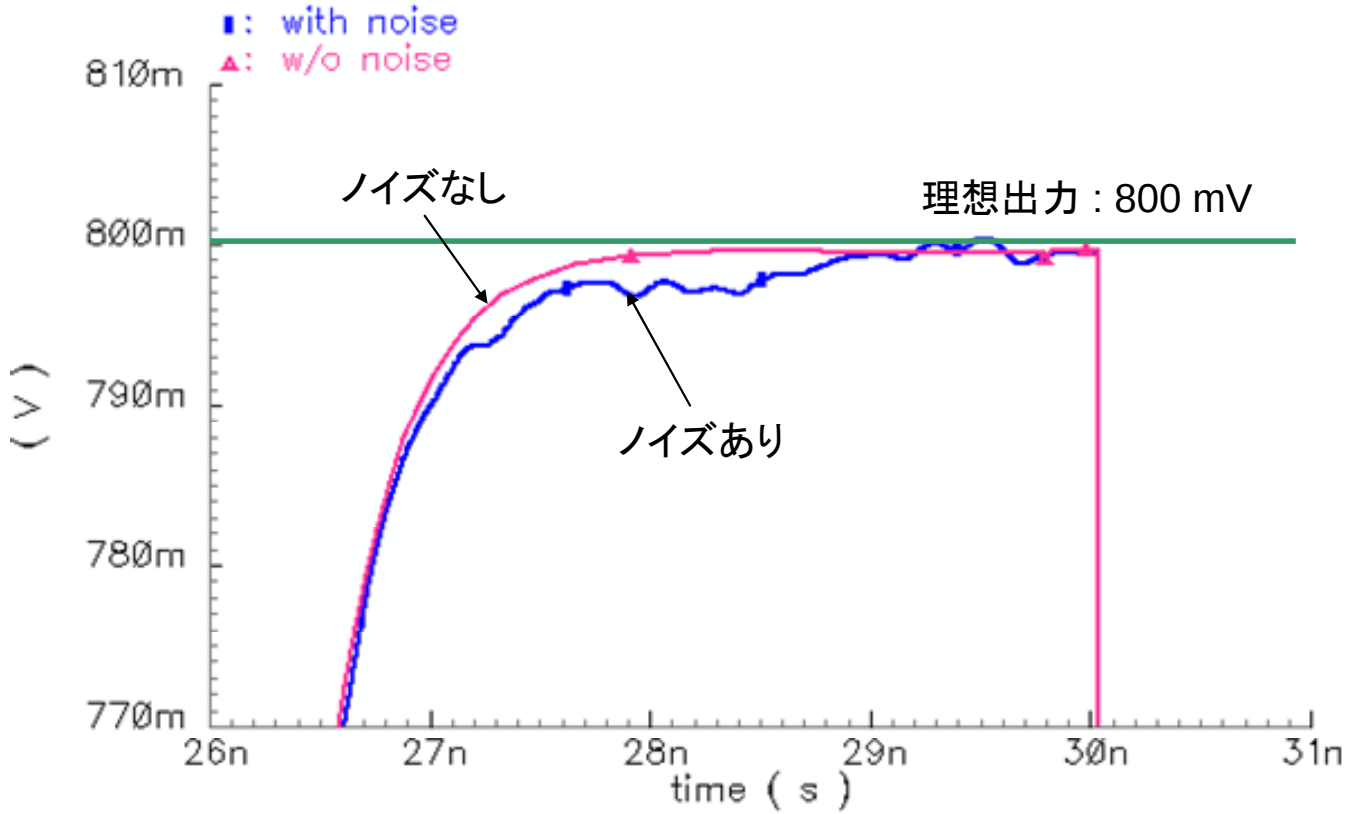


出力にノイズが現れている　　試行回数：10回



過渡信号波形

Tノイズを用いると、ノイズがあるときのセットリング時間をシミュレーションできる。



MatLabとCadence ADE

TOKYO TECH Pursuing Excellence

- **MatLab**
  - モデルベース(プログラムベース)Sim
  - SimLinkによりモデル構築が容易
  - 豊富なデータ処理機能(FFT, フィルタ、ヒストグラム, ポールゼロなど)と可視化機能
  - タイムベースの可変化が困難
  - SPICEとの混在Simが困難

- **Cadence ADE**
  - モデルベース+トランジスタベース
  - タイムベースの可変化が容易
  - SPICEとの混在Simが可能
  - 貧弱なデータ処理機能と可視化機能

まとめ

TOKYO TECH Pursuing Excellence

- 性能・仕様の明確化
- 性能を決定する要因を明らかにしてモデル化を行う
  - 性能を決めない要因も明らかにする
- 回路以前にシステムとしての解析が重要
  - 特に負帰還回路：安定性　信号の飽和
- システム解析から回路への要求仕様を明らかにする
- トランジスタのキャラクタライズにより回路設計は容易になる
- シミュレータは特徴を把握して活用する
  - MatLab：システム解析、信号・データ処理
  - Cadence: 機能・トランジスタ混在Sim
    タイムベース変動、過渡ノイズ解析