

Tip-Tilt補償光学系の性能評価

東北大学 修士 1 年 赤澤拓海

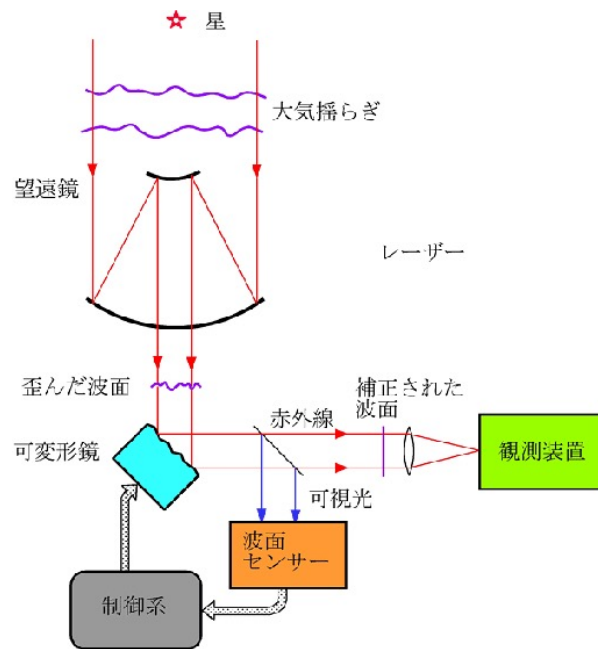
東北大学 秋山正幸、金沢大学 軸屋一郎

補償光学とTip-Tilt

地上観測の場合、観測天体からの光は大気を通過することによって波面が歪んでしまう。

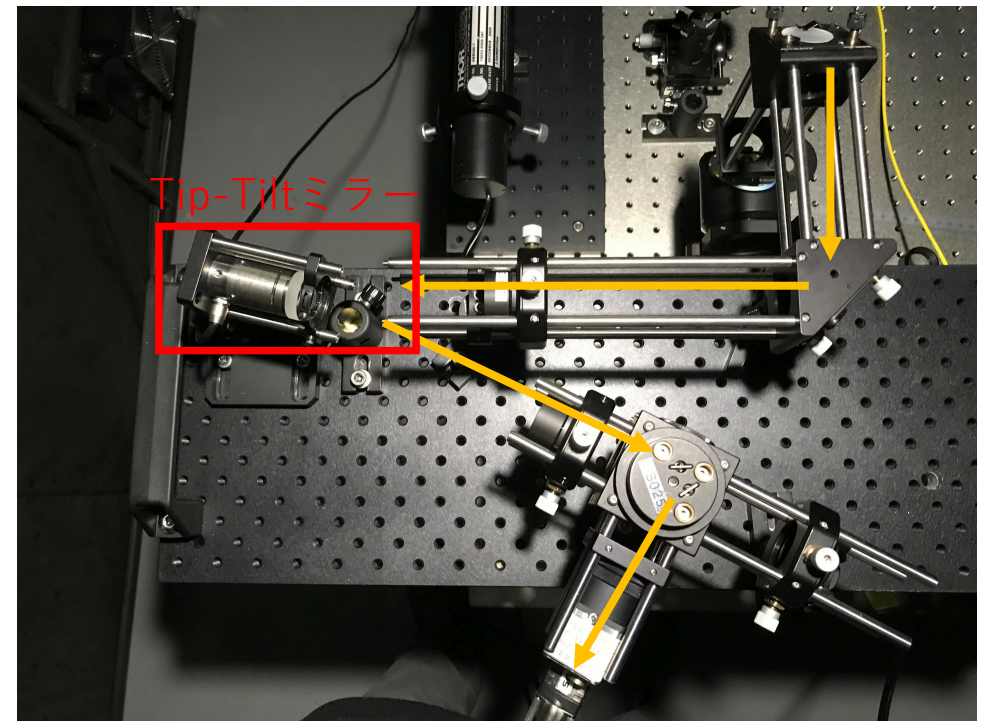


補償光学を用いて歪みを補正する。

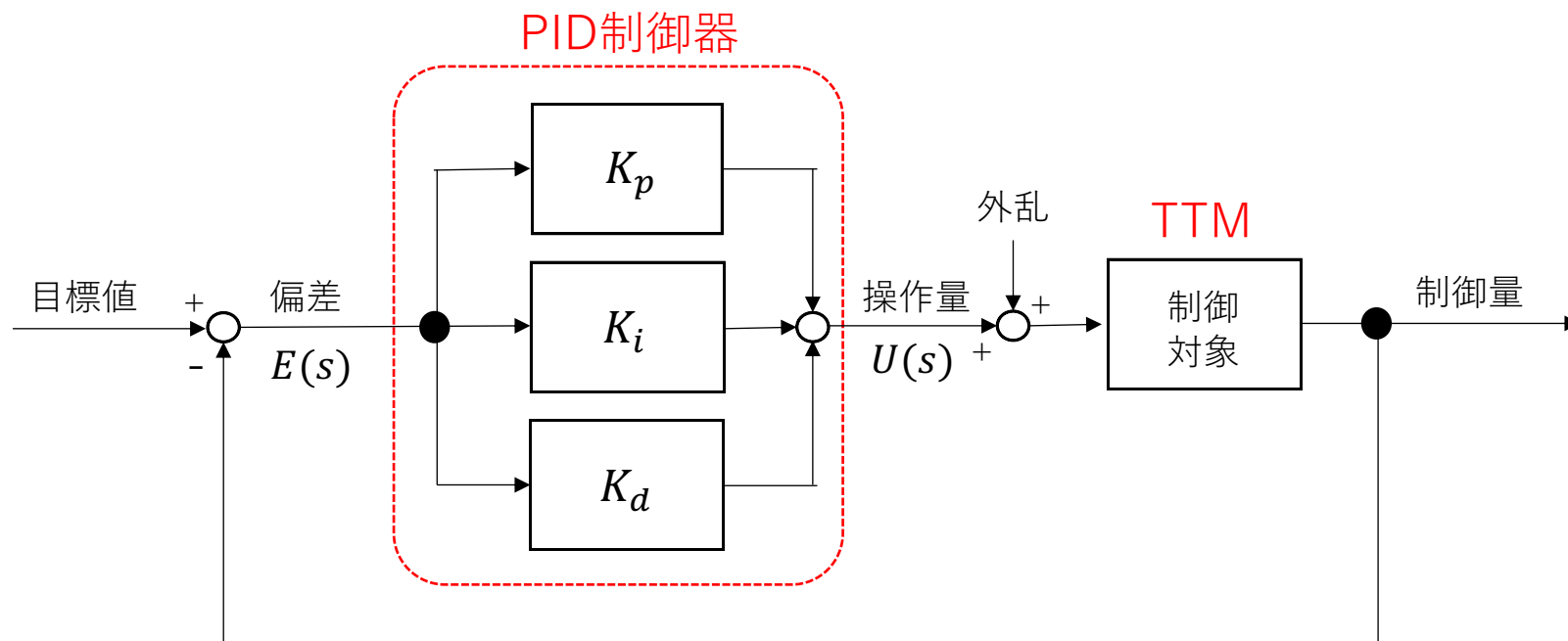


補償光学系の概念図

- 波面の傾きの x, y 成分をそれぞれTip成分、Tilt成分と呼ぶ。
- 像で見ると重心位置のズレに相当する。



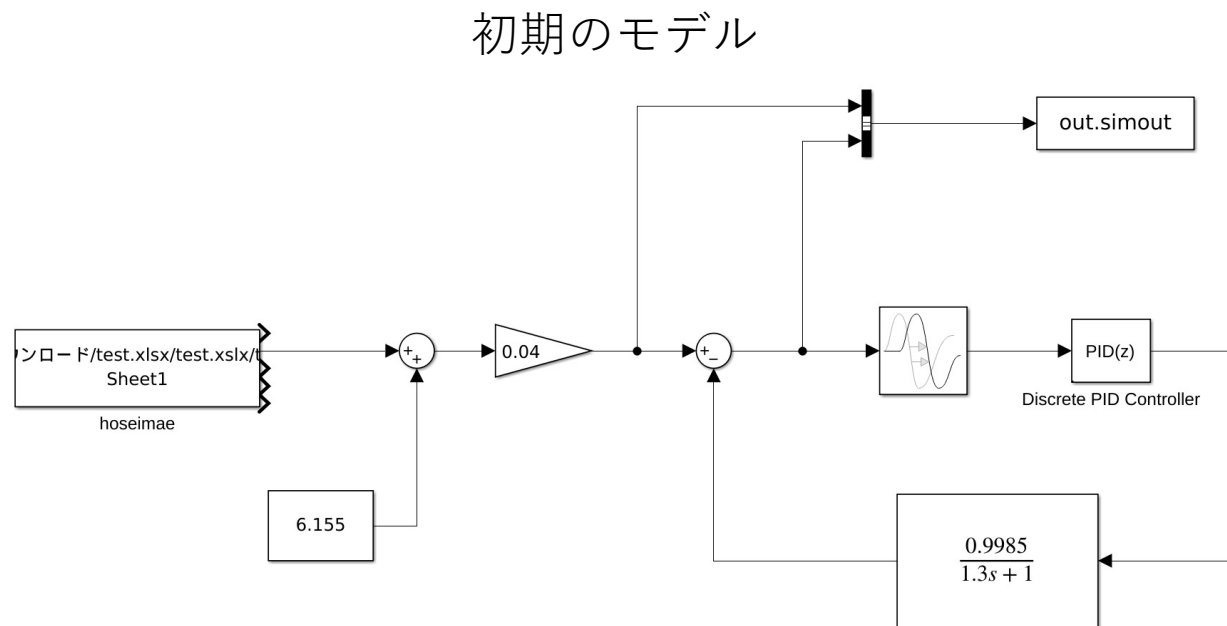
PID制御



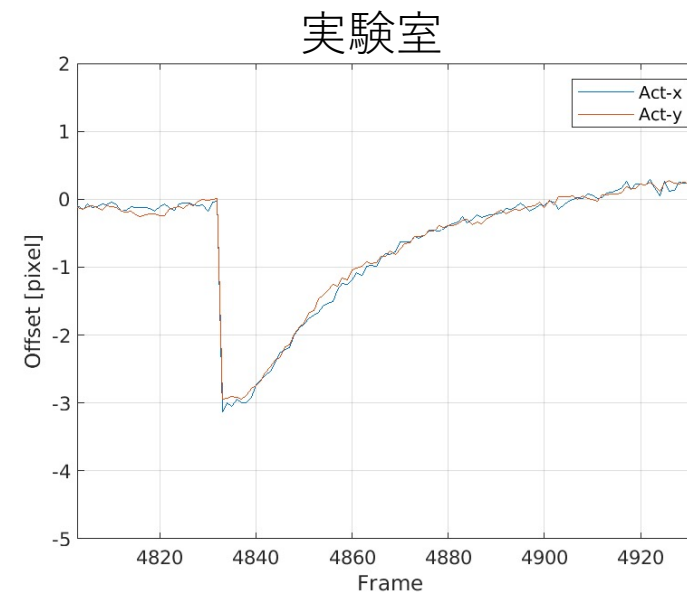
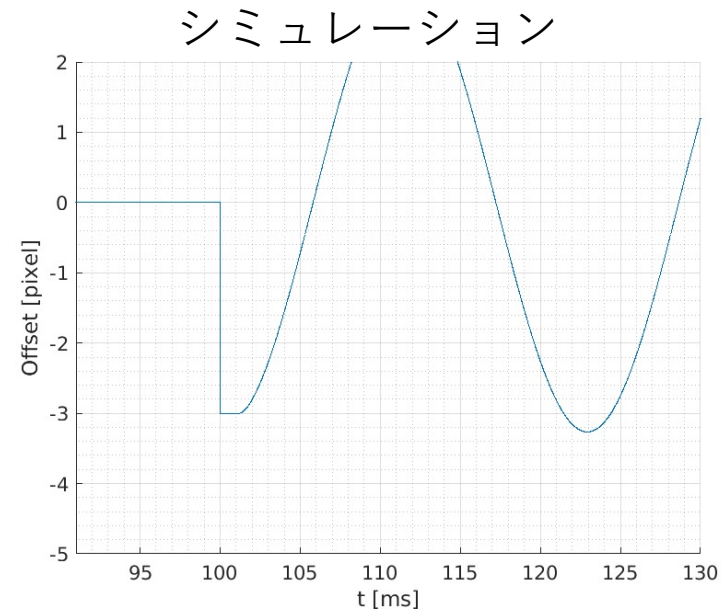
$$u(t) = \underline{K_p} e(t) + \underline{K_i} \int_0^t e(\tau) d\tau + \underline{K_d} \dot{e}(t)$$

P: 比例(proportional)
I: 積分(integrational)
D: 微分(differential)

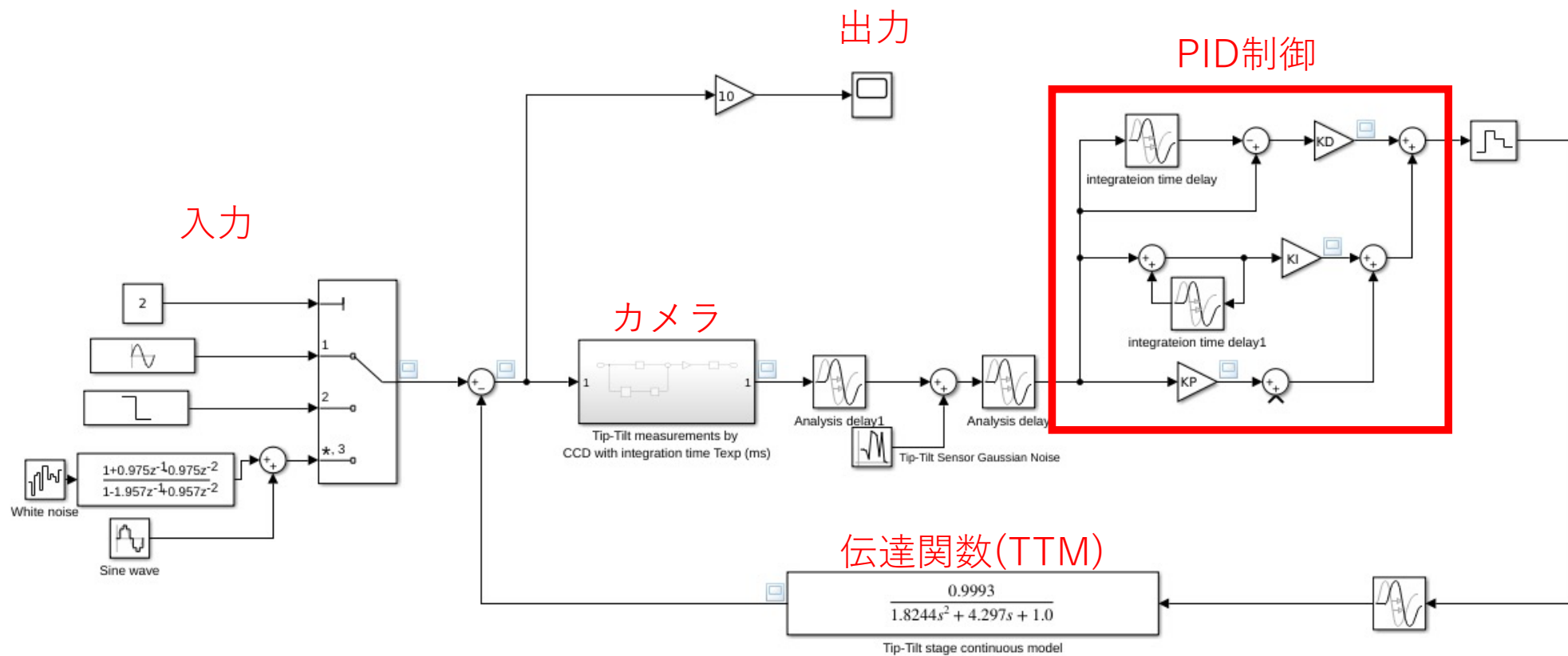
モデル化とその目的



正確なモデル化をすることでパラメータの最適化が行える。

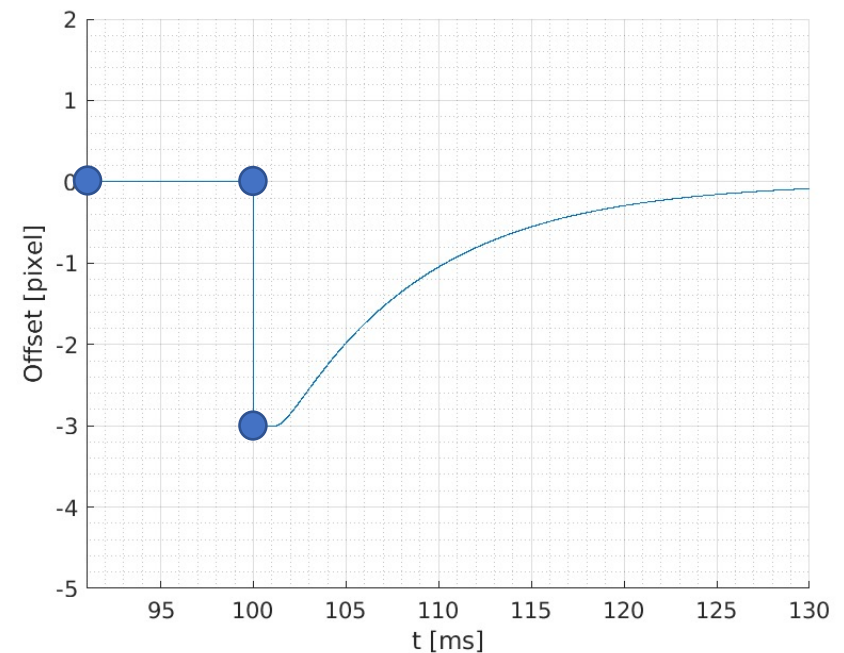
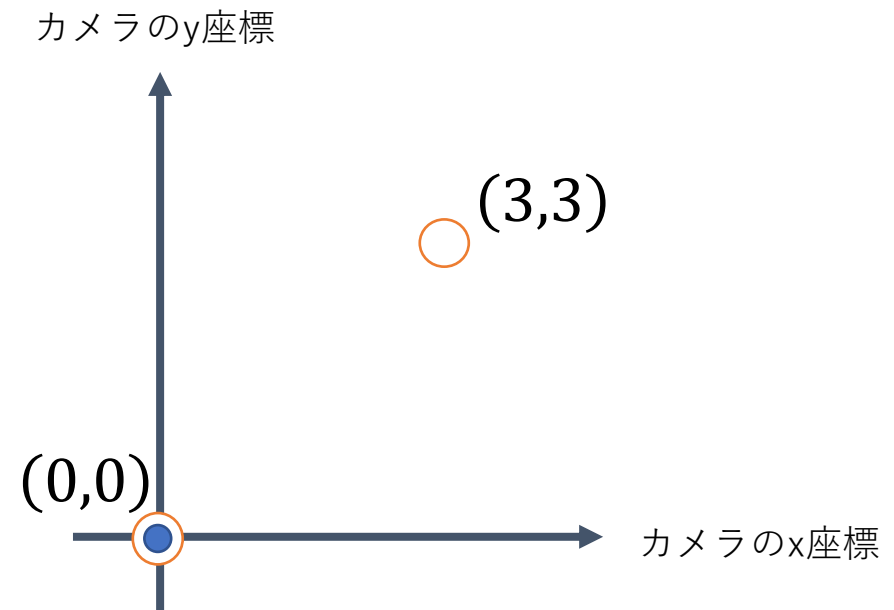


ブロック線図 (MATLAB, SIMULINK)



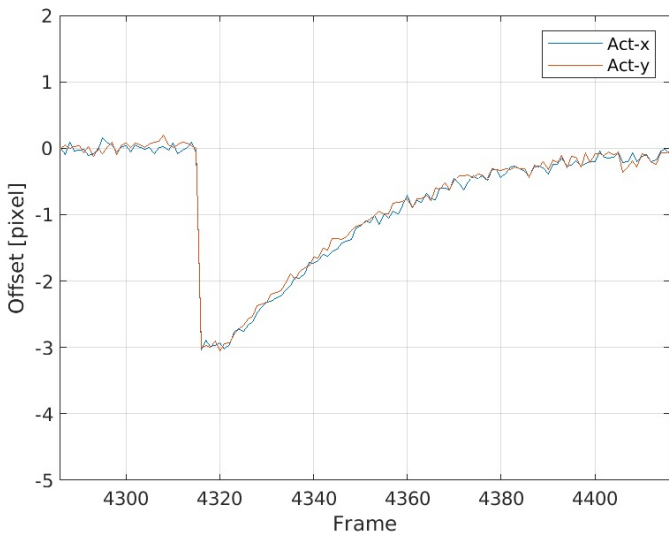
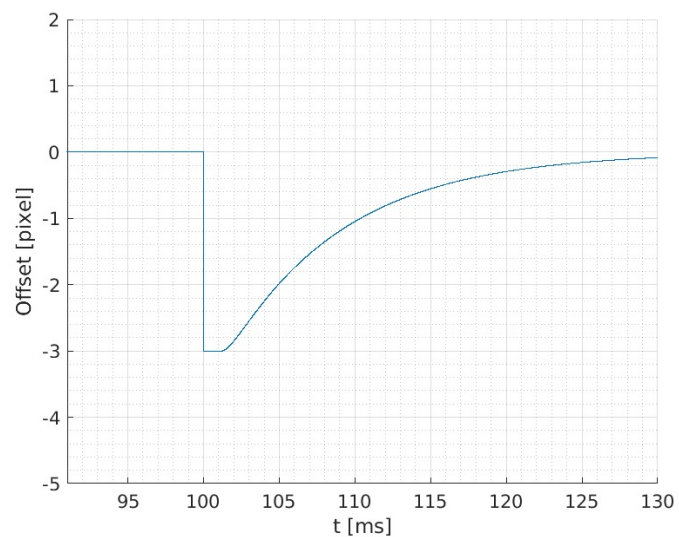
実験概要

- ① 制御パラメータ(K_p, K_i, K_d)をセットする。
- ② カメラの(0,0)の光が集まるように補正をかける。
- ③ ある瞬間（下図の $t=100\text{ms}$ ）にリファレンス位置（補正中心）を(3,3)[pix]に変更する。
- ④ 制御系は(3,3)に光を集めようと補正をかける
- ⑤ その時の光の位置とリファレンス位置の差をオフセットとして測定する。
- ⑥ シミュレーションと実験室の光学系で同じ実験を行い、振る舞いを比較する。

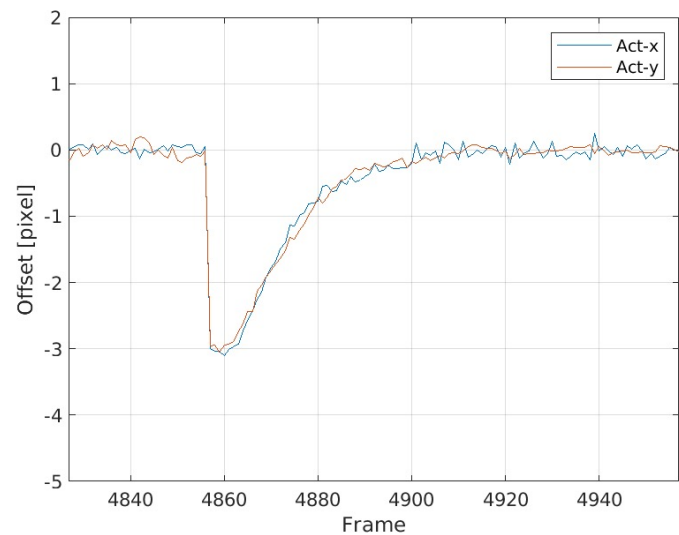
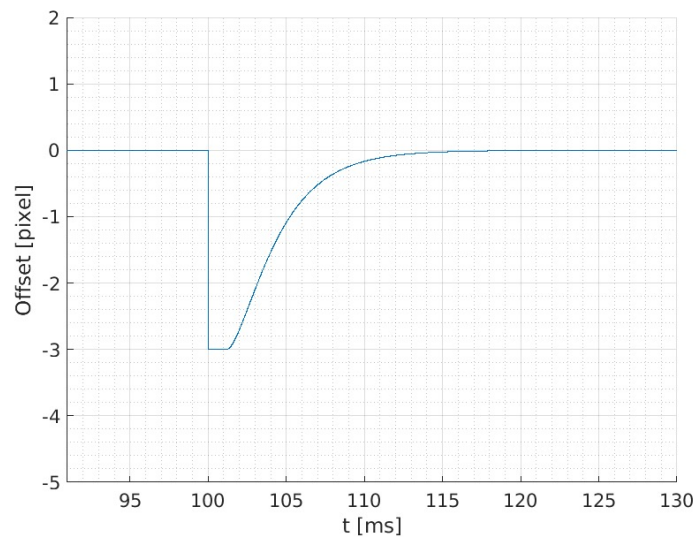


P制御の再現

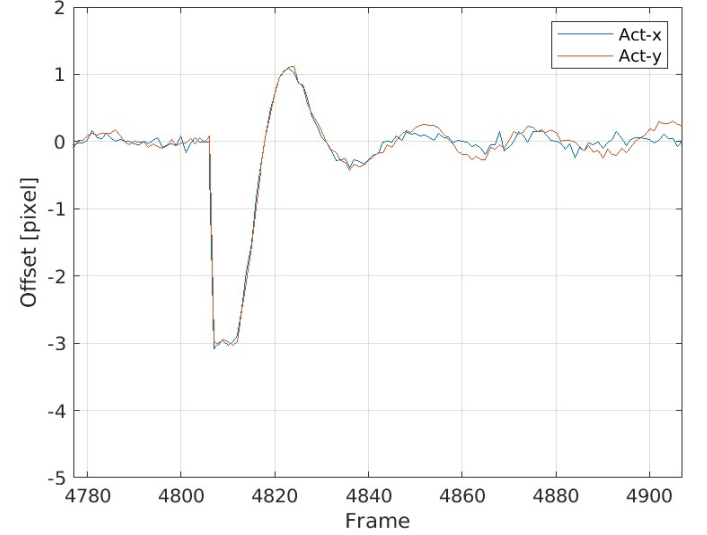
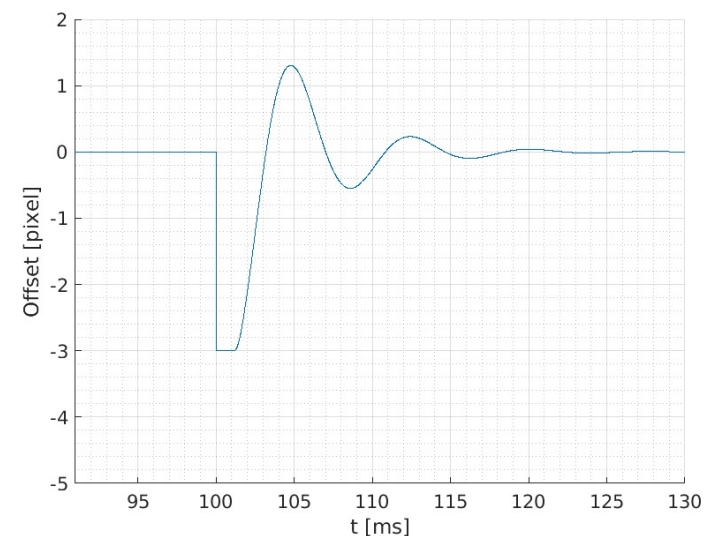
(0.5, 0, 0)



(1.0, 0, 0)

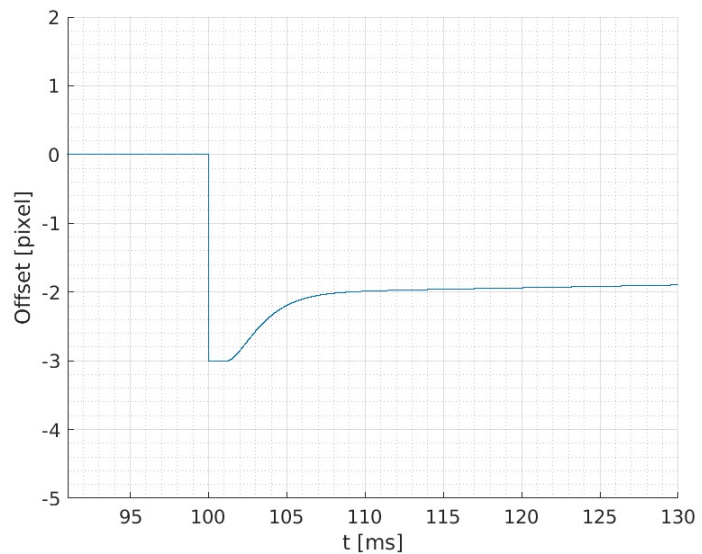


(3.0, 0, 0)

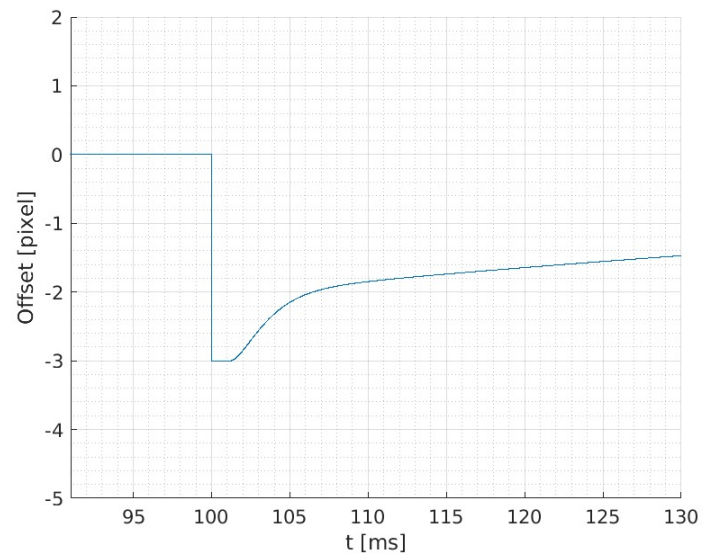


PI制御の再現

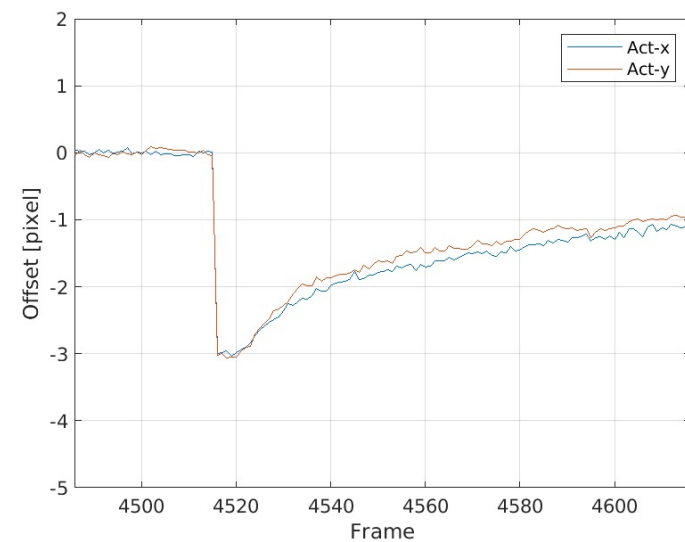
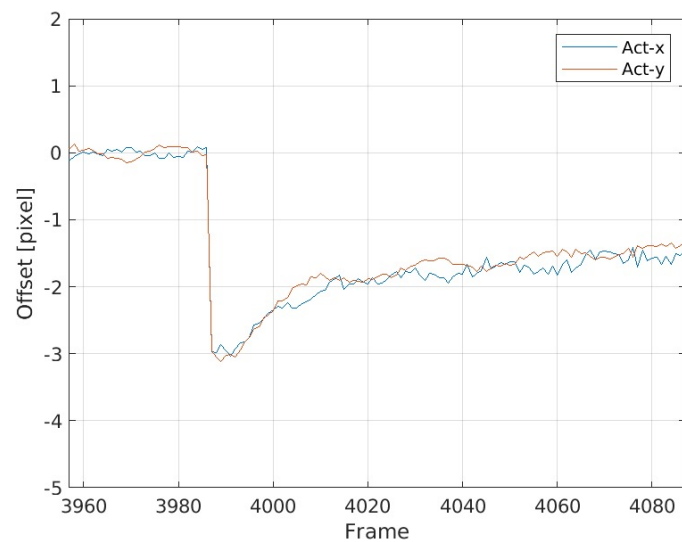
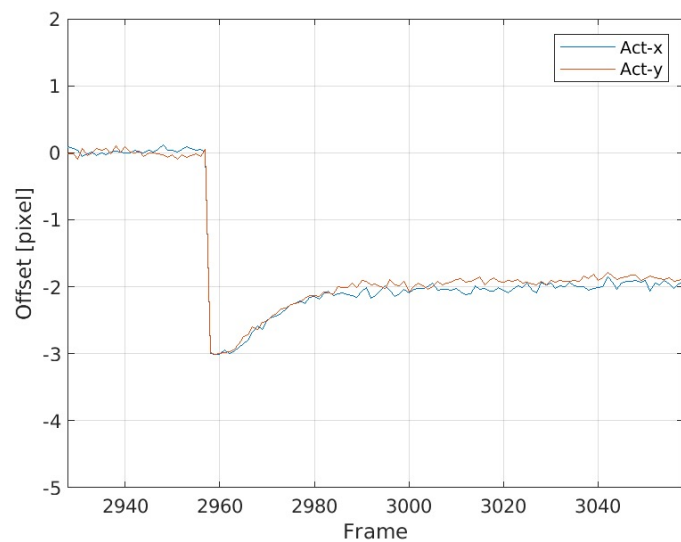
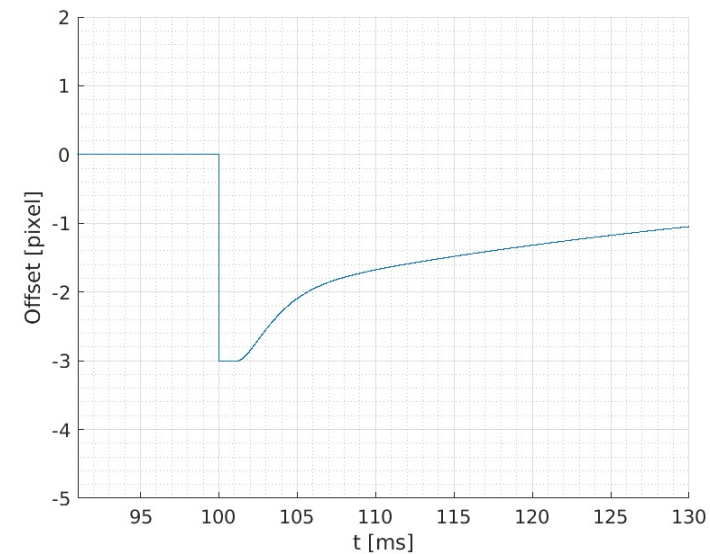
(0.5, 0.001, 0)



(0.5, 0.005, 0)

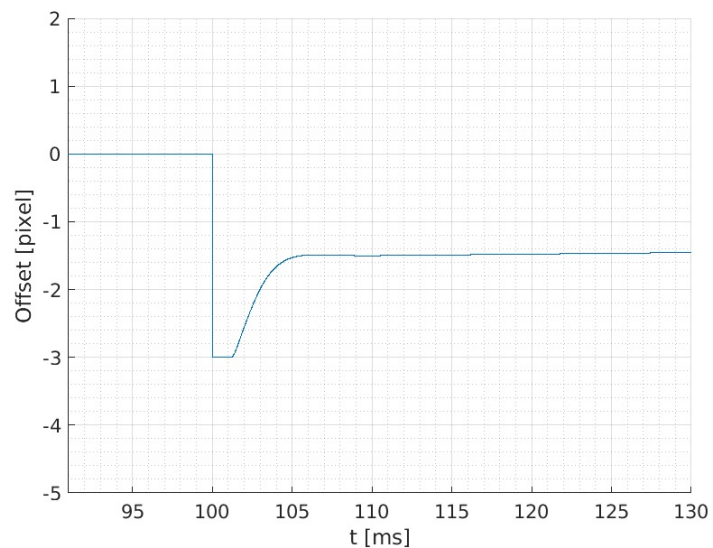


(0.5, 0.01, 0)

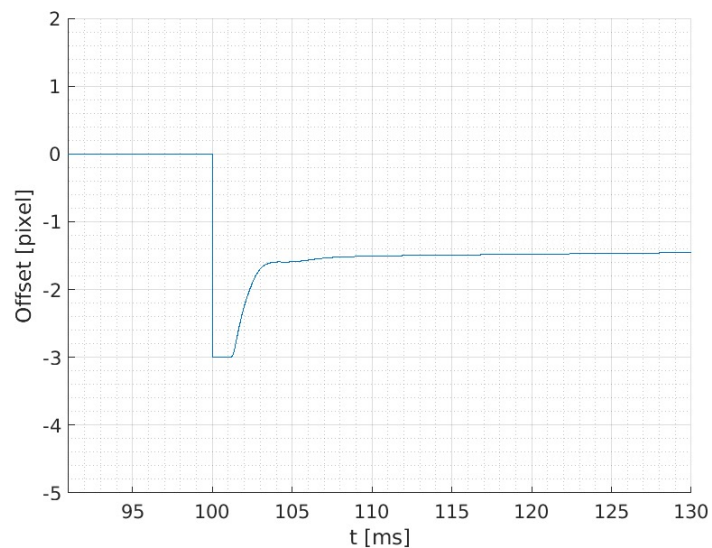


PID制御の再現

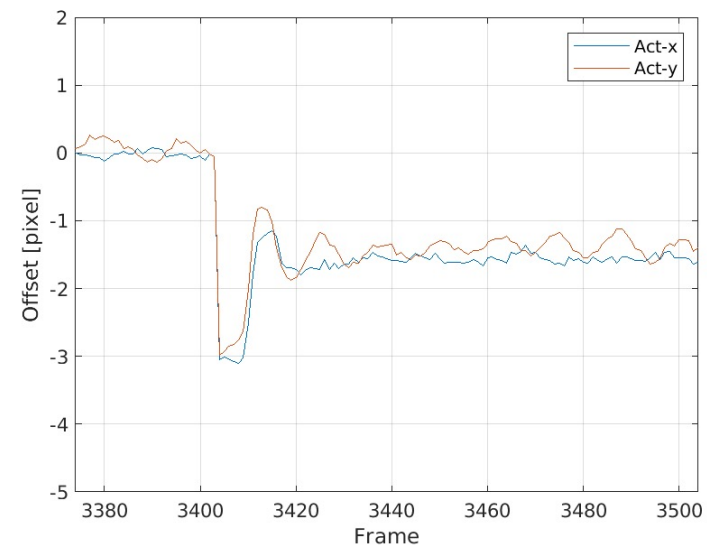
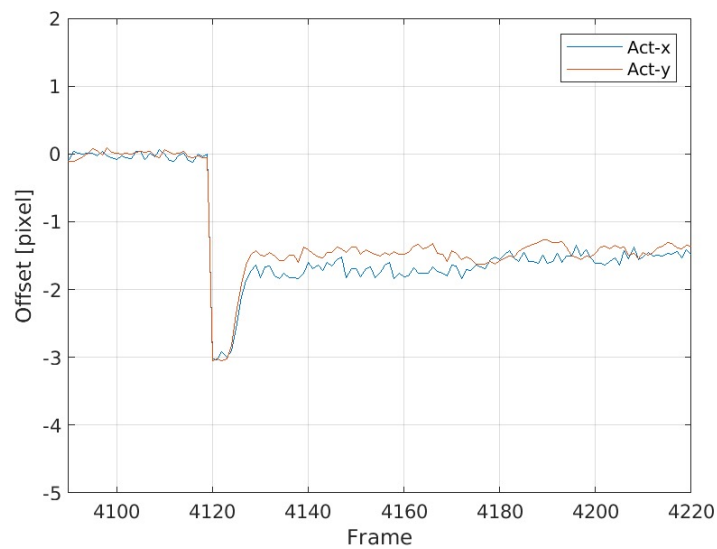
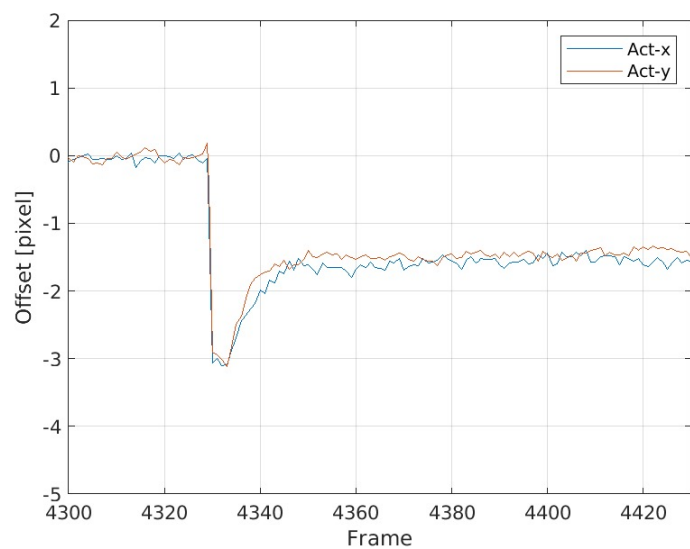
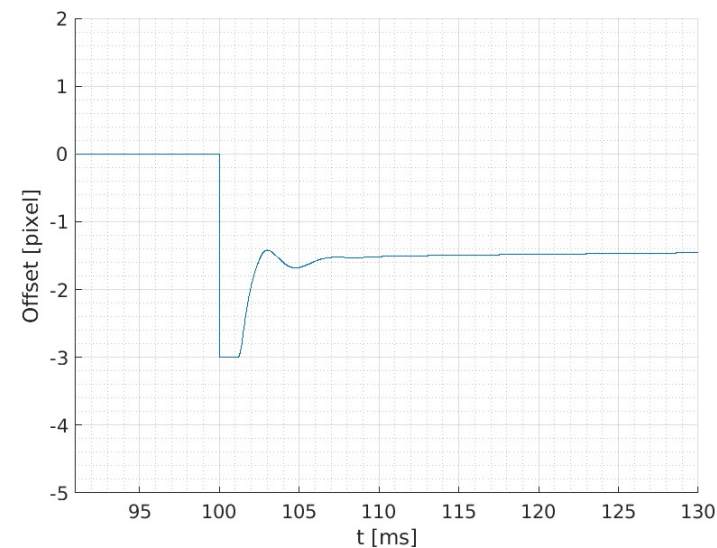
(1.0, 0.001, 1.0)



(1.0, 0.001, 3.0)



(1.0, 0.001, 5.0)



まとめと今後の展望

- 実験室でTip-Tilt補償光学系を組み上げて、その制御系のモデル化を目指した。
- P, PI, PID制御について、いずれの振る舞いもシミュレーションで再現することができた。
- 制御系の安定性を保ちつつ、補正精度が最も良くなるパラメータセットを探していく。

伝達関数

- 入力と出力の関係を表す関数
- 今回は与えた電圧とアクチュエータの長さの変化量の関係

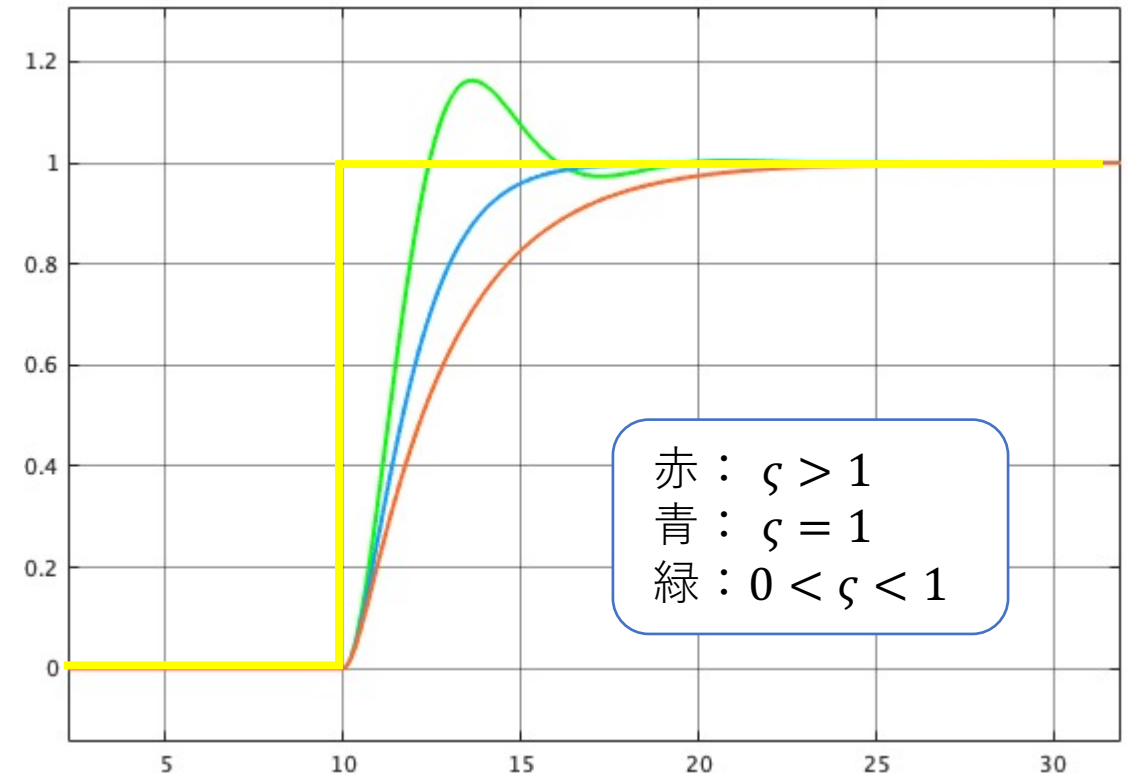
- 二次遅れ系の伝達関数の一般形は

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

- ζ は振動の特性を、 ω_n は速応性を定める。

- 本研究では

$$G(s) = \frac{0.9993}{1.8244s^2 + 4.297s + 1.0}$$



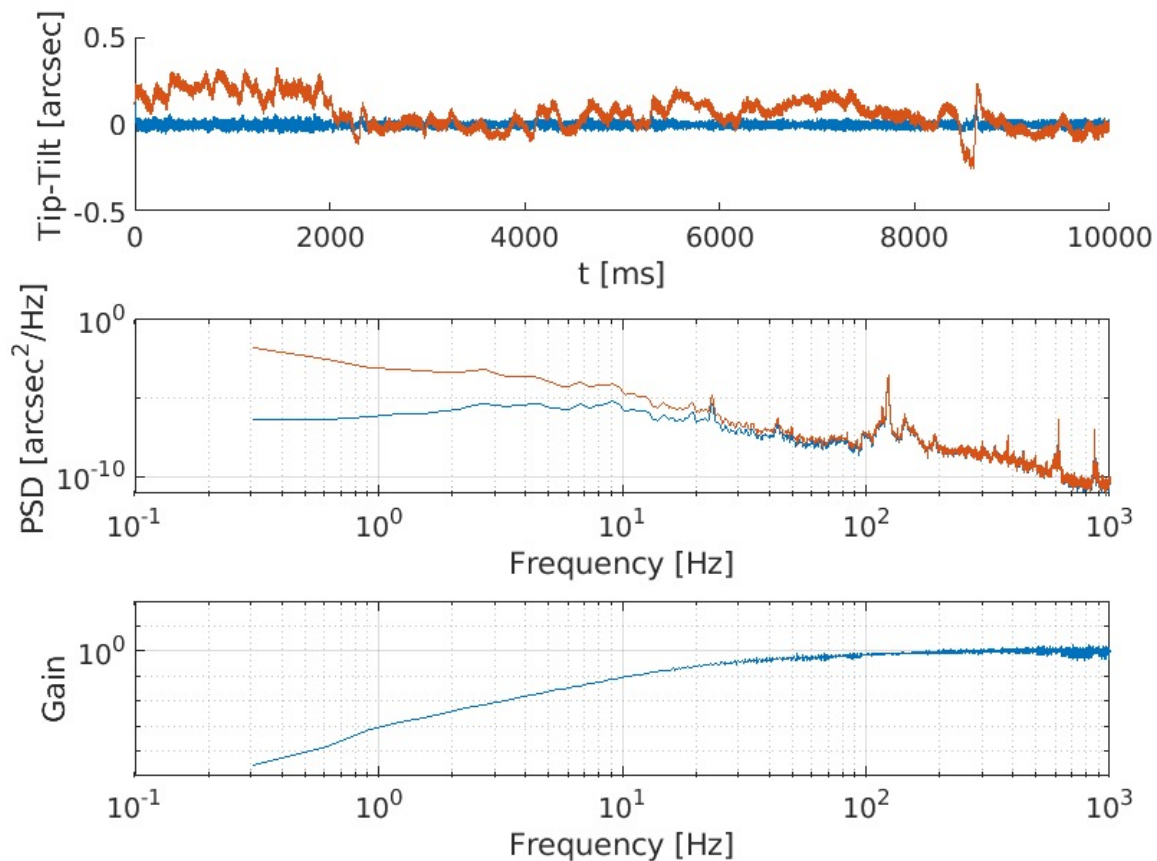
ステップ信号(黄)を入力したときの応答

次のスライドの図の見方

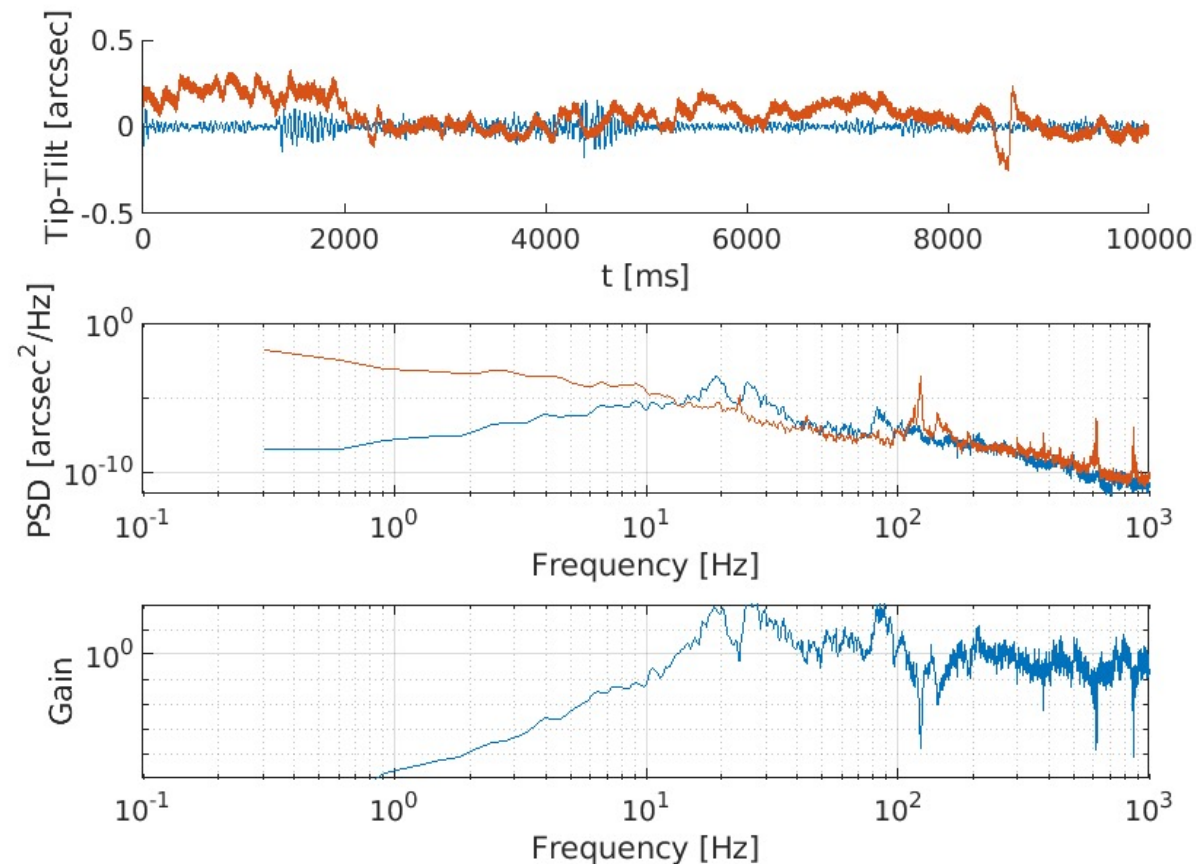
- 左のセットがシミュレーションで得られた補正結果、右のセットが実験室で組み上げた光学系の補正結果を表す。
1. 上段：揺らぎの大きさ[arcsec]の時間変化を表す。赤線は補正前、青線は補正後。赤線の揺らぎを入れてそれを補正すると青線の揺らぎが残ったということになる。
 2. 中段：揺らぎに各周波数成分がどれだけ含まれているかを表す。赤線は補正前、青線は補正後。PSDの値が小さいほどその周波数成分が少ない。また赤線より青線の方が値が小さくなるほどその周波数成分の補正がよく効いていると言える。
 3. 下段：中段の赤線と青線の値の比(Gain)を表す。値が小さいほどその周波数成分の補正がよく効いていると言える。

シミュレーションと実験室の補正結果の比較

シミュレーション



実験室



実験室の方が低周波数の補正精度が良いが、~10-100Hz付近で悪化する

PID制御のプログラム

IDScamera_CAIO::MeasureCalc() 補正計算部分

```
vCOP(0) = ReferencePosition.x - Centroid.x; リファレンス位置と測定された位置のずれ  
vCOP(1) = ReferencePosition.y - Centroid.y;  
vCOI = vCOI + vCOP; 位置ずれの積分量の計算  
vCOD = vCOP - vCOPpre; 位置ずれの微分量の計算
```

```
vVOP = mA * vCOP; 行列をかけて位置ずれの値を補正電圧値に変換  
vVOI = mA * vCOI;  
vVOD = mA * vCOD;
```

```
vCOPpre = vCOP; 次の微分量の計算のために値を保存
```

```
if( iFlag_CAIOupdate.load() == 0 && Centroid.peak > PeakThreshold*ReferencePosition.peak){  
    CAIOstatus.fTargetPos[0] = fKP*vVOP(0) + fKI*vVOI(0) + fKD*vVOD(0) + CAIOstatus.fCurrentPos[0];  
    加える
```

```
    CAIOstatus.fTargetPos[1] = fKP*vVOP(1) + fKI*vVOI(1) + fKD*vVOD(1) + CAIOstatus.fCurrentPos[1];  
    iFlag_CAIOupdate.store(1);  
}
```

```
if( bSaveData ){  
    DataFile << " updateTT " << vCOI(0) << " " << vCOI(1);  
}
```

```
// reset integrated error  
if( vCOI(0) > 10 || vCOI(1) > 10 ){  
    vCOI(0) = 0.0;  
    vCOI(1) = 0.0;  
}
```

9/21

カメラで見た時の位置ズレをvCOPとし、それらの総和をvCOI、一つ前との差分をvCODとする。

行列mAをかけることで、位置ズレから電圧値に変換する。

設定したゲイン(fKPなど)をかけて補正值を出す。

次のスライドでmAを求める。
(mA→Aとする)

行列Aを求めるプログラム

$$A\vec{c} = \vec{v}$$

$\vec{v}$; アクチュエータの各チャンネルに与えた電圧
 $\vec{c}$; カメラで見た時の移動量



何回か測定してベクトルを行列にする

$$AC = V$$



C は正方行列ではないため逆行列が取れない。
 C^T をかけることで正方行列にして、逆行列を取れるようにする。

$$ACC^T = VC^T$$

$$\therefore A = VC^T(CC^T)^{-1}$$

$$V = \begin{pmatrix} \boxed{\Delta V} & \boxed{0} & \cdots \\ \boxed{0} & \boxed{\Delta V} & \cdots \end{pmatrix}$$

ΔV ; アクチュエータ (2つ) に与える電圧の上限一下限 (設定値)

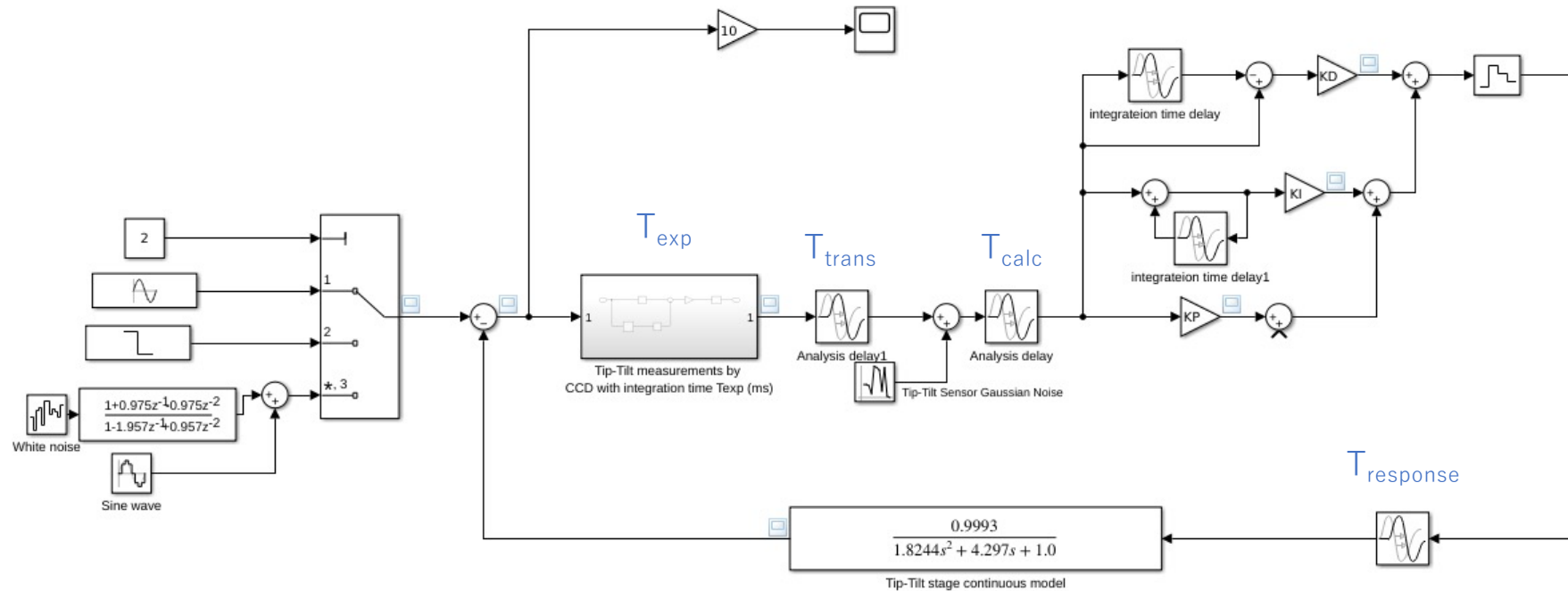
↓ 対応 ↓

$$C = \begin{pmatrix} \boxed{\Delta x} & \boxed{\Delta x} & \cdots \\ \boxed{\Delta y} & \boxed{\Delta y} & \cdots \end{pmatrix}$$

$\Delta x, \Delta y$; ΔV の変化を与えたときの移動量

1 セット (何セットかデータを取る)

各所のむだ時間（遅延）



T_{exp}	0.302ms	CCDカメラの露出時間
T_{trans}	0.3ms	制御系へのデータの転送時間
T_{calc}	0.05ms	データの処理時間
T_{response}	0.3ms	電圧の伝達時間