信号依存ノイズによる終点分散最小規範の検証

上野, 秀俊 / UENO, Hidetoshi

(発行年 / Year)
2012-03-24
(学位授与年月日 / Date of Granted)
2012-03-24
(学位名 / Degree Name)
修士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

信号依存ノイズによる終点分散最小規範の検証

10R6203　上野秀俊

法政大学大学院　工学研究科　システム工学専攻　人間情報研究室
担当教員　藤田昌彦　教授

あらまし

サッカードのような到達運動は，釣鐘型の滑らかな軌跡になる速度特性をとる．多くの学者がこの特徴を解明するために研究をしてきた．1998 年に Harris と Wolpert は「到達運動の目的は終点での位置の分散を小さくすること」として、脳からの指令の大きさに比例するノイズの存在を仮定した「終点分散最小規範」を提唱した．確かに，この規範によって得られる到達運動は，実際に観測して得られる速度波形とほぼ一致した．しかし，この理論は極値問題の解であるため，その周辺の軌道は観測される運動と大きく異なる可能性がある．そこで，非現実的な速度特性を持つような拘束条件を追加して，この規範を用いたところ，終点分散は追加前と大差ない結果が得られた．よって，実際とは異なる運動軌道にも終点分散が小さい軌道が多く存在することが示された．

目次

1　序論

人間が注視位置を変更する時に生じる高速な眼球運動（Saccade，サッカード，跳躍性眼球運動）のような目標点への到達運動をする時，脳から発せられた運動指令信号が神経系を通じて，各筋肉に伝えられて一連の動作が生成される．最終的に目標点の近傍へと導かれることになる．この時，眼が取りうる運動の軌跡は無限に存在する．しかし，実際には滑らかで釣鐘型の軌跡を持つ速度特性をとる．そのため，到達運動を最適化するための何らかの規範に基づき，脳が軌道計画を決定すると考えられる．

最適化規範の一つに， Harris と Wolpert [1] が提案した「終点分散最小規範」がある．この規範で作られる速度軌跡は現実の人間の運動特性とほぼ一致した．特に，振幅 20 deg 以上の運動で起こる非対称な軌跡をも見事に導きだしている．

ところで，終点分散最小規範は極値問題の解であるから，その周辺の軌道は観測される運動と大きく異なる可能性がある．そこで本研究では速度特性の拘束条件を様々に設定してシミュレーションを行い，最終規範最小規範の妥当性を検討することにした．

2　終点分散最小規範

同じ到達運動でも図 1 のように終端点にばらつきが生じる．

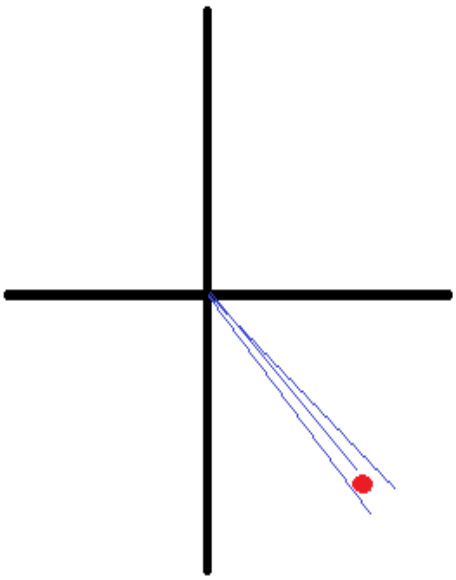

図 1. 終端点のばらつき

終点分散最小規範を数式に表すと (1) 式のようになる．

$$\sum_i \int_T^{T+R} (x_{i,t} - \bar{x_t})^2 dt \quad \rightarrow Min \tag{1}$$

i は試行番号，$x_{i,t}$ は i 番目の試行の時刻 t における眼の位置，$\bar{x_t}$ は時刻 t における眼の平均位置である．また，T は運動到達時間，R は運動終了後の停留時間である．

Harris と Wolpert [1] は到達運動において，脳から送られる運動指令信号に避けられない生体ノイズが加わることによって到達点が分散すると考え，到達点での分散を最小にするような軌道計画が最適であるとした．ここでノイズは運動指令信号の大きさに応じて増大する信号依存ノイズ（signal-dependent noise）を仮定した．これにより，素早い運動は大きな運動指令信号を必要とするが，その分，終点での分散を増大させてしまうことになる．信号依存ノイズ w_t はガウス性白色ノイズであり，平均 0 で分散 $\sigma^2[w_t]$ は運動指令が u_t であるとき，(2) 式のように仮定した．

$$\sigma^2[w_t] = k|u_t|^2 \tag{2}$$

3　計算モデル

3.1　眼球プラント

検証は Harris と Wolpert [1] の設定に従って図 2 の眼球プラントを用いた．図 2 において U は運動指令信号，W はノイズ，X は眼の位置，T_1, T_2 は時定数である．

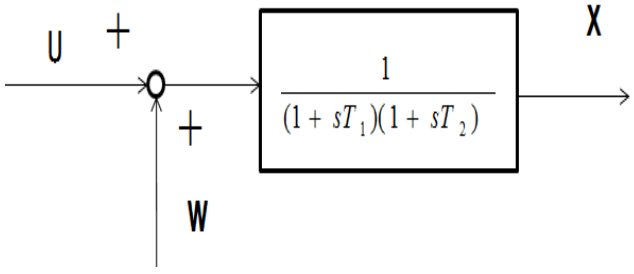

図 2．眼球プラント

眼球プラントを微分方程式にすると

$$\dot{\boldsymbol{x_t}} = \boldsymbol{C}\boldsymbol{x_t} + \boldsymbol{b}(u_t + w_t) \qquad (t = 1, \ldots, T+R) \tag{3}$$

離散近似すると

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{b}(u_t + w_t) \quad (t = 1, \ldots, T+R) \tag{4}$$

$\boldsymbol{x}_t$: $\begin{pmatrix} \text{位置} \\ \text{速度} \end{pmatrix}$ からなる時刻 t での眼の 2 次元ベクトル

$\boldsymbol{A}$: $\boldsymbol{I}$ を単位行列として， $\boldsymbol{A} = \boldsymbol{I} + \boldsymbol{C}$ である 2 次の正方行列

$$\begin{pmatrix} 1 & 1 \\ -\frac{1}{T_1 T_2} & 1 - \frac{T_1 + T_2}{T_1 T_2} \end{pmatrix}$$

b: $\begin{pmatrix} 0 \\ \frac{1}{T_1 T_2} \end{pmatrix}$ からなる 2 次元ベクトル

u_t : 時刻 t における運動指令信号（スカラー）

w_t : 時刻 t におけるノイズ（スカラー）

3.2 費用関数と制約条件式

時刻 t での共分散は

$$Cov[x_t] = k \sum_{i=0}^{t-1} (\boldsymbol{A}^{t-1-i}\boldsymbol{b})(\boldsymbol{A}^{t-1-i}\boldsymbol{b})^T u_i^2 \tag{5}$$

$$(t = 1, \ldots, T+R)$$

$Cov[x_t]$ の (1,1) 要素が時刻 t での分散 V_t である.

制約条件は期待値が到達点で停留期間中，その位置を維持するという条件とするので

$$E[x_t] = \sum_{i=0}^{t-1} \boldsymbol{A}^{t-1-i}\boldsymbol{b}u_i \quad (t = 1, \ldots, T+R) \tag{6}$$

が停留期間中，一定値を取るように設定した.

最小化される費用関数 f は，時刻 $t = T$ で到達後の一定期間 R にわたって，合計された位置の分散として

$$f = \sum_{t=T+1}^{T+R} V_t \tag{7}$$

と定義する．

4 実験

(4) 式の計算モデルをつかって [1] を参考にシミュレーションにより検証を行った．最小分散を求める計算は [2] を参考に MATLAB 最適化ツールボックスの制約条件付き非線形最小化 (fmincon) を用いた．サッカードの振幅と到達時間は [2],[3] を参考に表 1 のように設定した．

表 1 . 振幅と到達時間

振幅 r (deg)	到達時間 T (ms)
5	40
10	50
20	70
30	90
40	120
50	160

運動終了後の停留時間 R を 50 ms，時定数 T_1,T_2 をそれぞれ 224 ms, 13 ms とし， 1 ms の時間ステップで検証を行った．

4.1 実験１：Harris と Wolpert [1] の再現

Harris と Wolpert [1] と同条件で釣鐘型の速度波形が描けるかを検証した（実験 1 ）.

4.2 実験２：最適解の近傍

実験 1 で作成された最適な運動指令信号を $\tilde{u}$ として，（8）式のようにランダムの値 a を加え， f の値を計算した.

$$u_t = \tilde{u}_t + a \qquad (t = 1, \ldots, T-3) \tag{8}$$

ここで a は $-0.5, \ldots, 0.5$ の間で 0.1 刻みの値をランダムに加え u の値を変更した．また到達点、停留期間は一定の値を取る必要があるため，値の変更は，$u_1, \ldots, u_{T-3}$ までとし，u_{T-2}, u_{T-1} では目標に到達するように調整した値とし，残りは無修正とした（実験 2 ）．このような運動指令信号 1000 サンプルでの f をプロットした（図 5 ）．ここに横幅は

$$|\Delta u| = \sqrt{(\tilde{u}_1 - u_1)^2 + \cdots + (\tilde{u}_T - u_T)^2} \tag{9}$$

である.

4.3 実験３：対称条件を加えて

速度波形が対称な波形をとるように 計算モデルの制約条件に (10) 式を追加して最適条件と比較した (実験 3)．なお，5 deg, 10 deg では最適条件で，ほぼ対称波形であるため対称条件は加えなかった.

$$\boldsymbol{x}(t) = \begin{pmatrix} p(t) \\ v(t) \end{pmatrix} \text{ として}$$

$$v(T/2 - c) = v(T/2 + c) \qquad (0 \le c \le T/2) \tag{10}$$

4.4 実験４：一定区間同値条件を加えて

実験 1 の制約条件に速度の最大値（ピーク）を中心に一定期間 D で同じ数値（速度）をとるように (11) 式を追加してシミュレーションを行った (実験 4)．この検証での振幅は 5 deg, 20 deg, 40 deg で行った.

$$v(m) \equiv v(m+c) \tag{11}$$

ここに m は速度が最大となる時刻であり，全ての $|c| \leq D/2$. 実験 4 を図で示すと，図 3 のようになる.

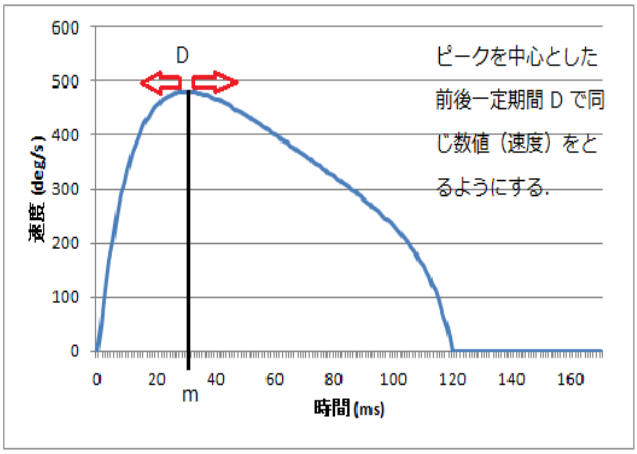

図 3. 実験 4 の説明

5　結果と考察

5.1　実験 1 ：Harris と Wolpert [1] の再現

結果は図 4 となり、彼らと同じ結果が得られることを確認した.

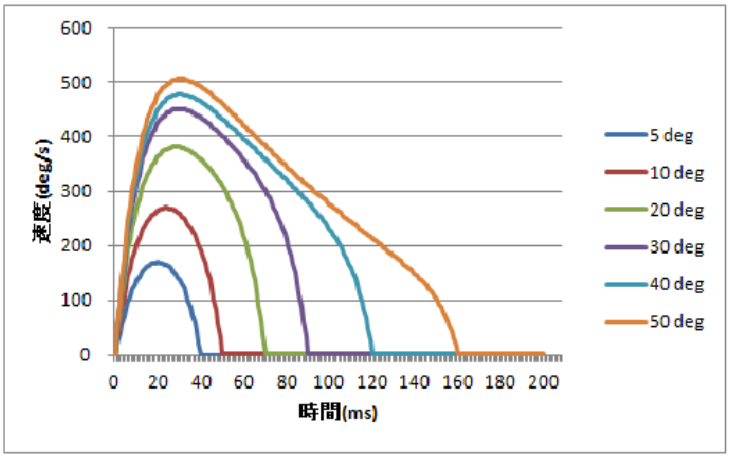

図 4. サッカードの最適速度波形

5.2　実験２：最適解の近傍

結果は，図 5 となり，振幅に関わらずほぼ同じグラフとなり，$|\Delta u|$ の値が大きくなると f の値は急速に大きくなる．

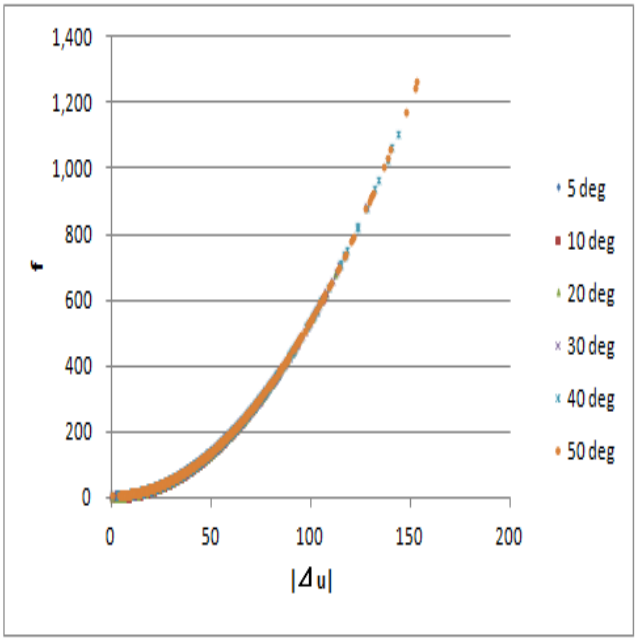

図 5. $f - |\Delta u|$ グラフ

5.3　実験３：対称条件を加えて

対称条件を加えて求まる速度波形は図 6 のようになった．

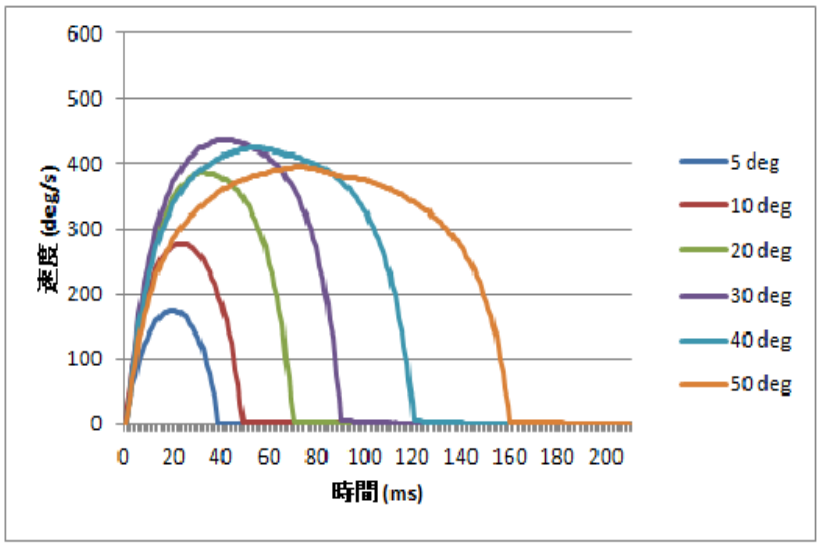

図 6. 対称条件を加えての速度波形

また，２つの運動指令信号を比較すると図 7，8 となった．

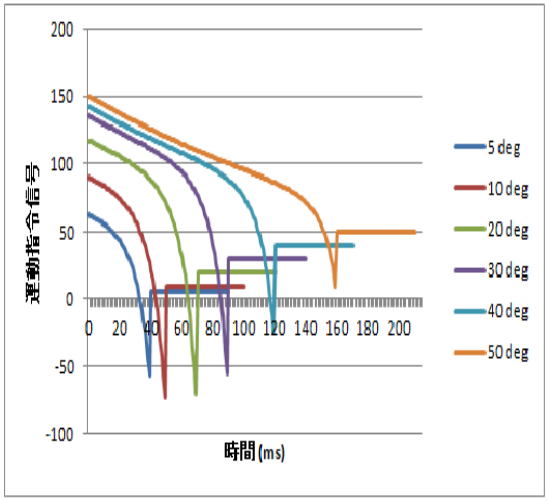

図 7. 最適条件での運動指令信号

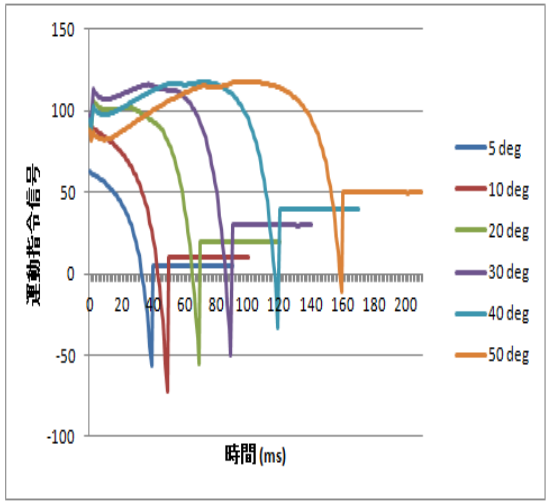

図 8. 対称条件での運動指令信号

２つの条件を比較すると，表２のようになり最適条件と異なる条件でも終点分散 f の値に大差がなかった．

表 2 . 最適条件と対称条件の f の比較

f	最適条件	対称条件
20 deg	3.2448	3.5680
30 deg	5.1760	5.5600
40 deg	6.4567	6.8943
50 deg	7.3059	7.9193

最適条件での運動指令信号を $\tilde{u}$ ，対称条件での運動指令信号を $\check{u}$ として

$$u_t = \tilde{u_t} + \lambda(\check{u_t} - \tilde{u_t}) \qquad (t = 1, \ldots, T + R) \tag{12}$$

とし， f の値，速度波形を出し，２つの条件の間でどのような変化が起きているのかを調べた．ここで λ は 0〜1 まで 0.1 を刻み幅とした．また図では， $\lambda = 0, 0.3, 0.7, 1$ での運動指令信号と速度波形を記した．

結果から最適条件から対称条件への変化は単調で，準最適のような特殊ケースは見られなかった．

5.3.1　20 deg での比較

表 3. 20 deg での f の変化

λ	f
0 （最適）	3.2448
0.1	3.2482
0.2	3.2580
0.3	3.2743
0.4	3.2970
0.5	3.3262
0.6	3.3619
0.7	3.4041
0.8	3.4527
0.9	3.5078
1 （対称）	3.5680

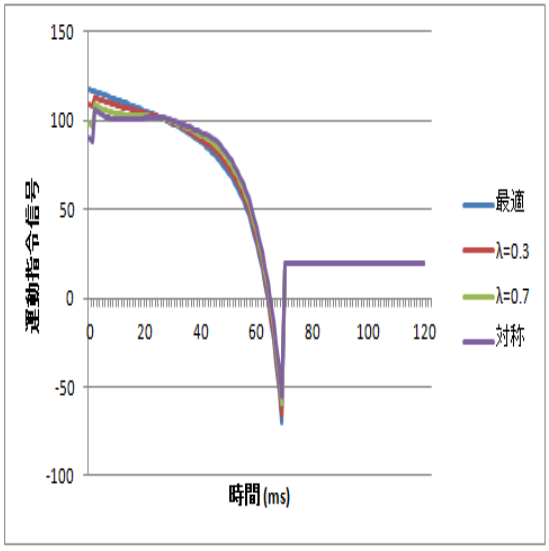

図 9. 20 deg 運動指令の比較

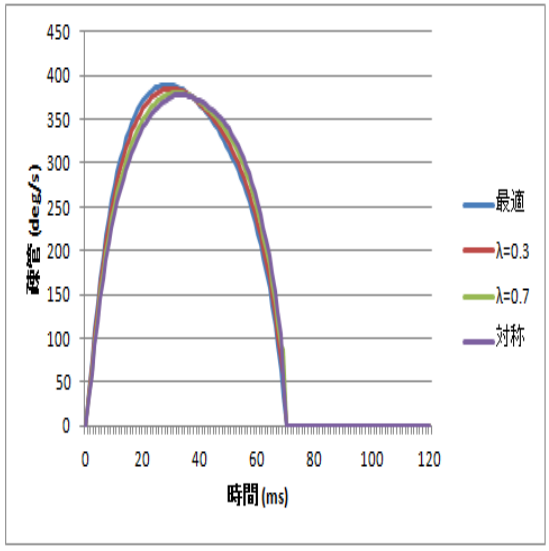

図 10. 20 deg 速度波形の比較

5.3.2　30 deg での比較

表 4. 30 deg での f の変化

λ	f
0 （最適）	5.1760
0.1	5.1799
0.2	5.1914
0.3	5.2104
0.4	5.2369
0.5	5.2710
0.6	5.3126
0.7	5.3618
0.8	5.4184
0.9	5.4827
1 （対称）	5.5600

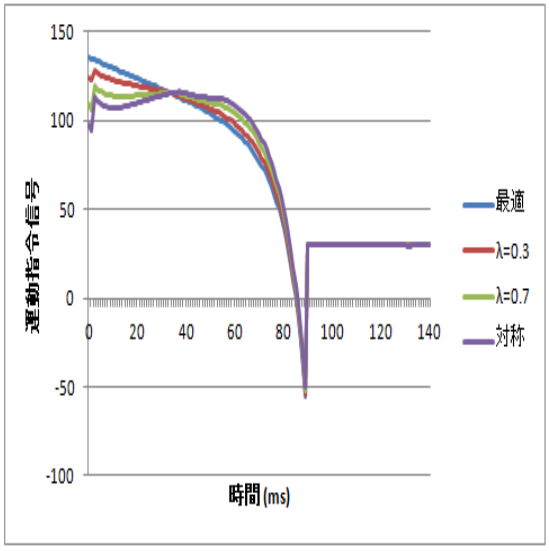

図 11. 30 deg 運動指令の比較

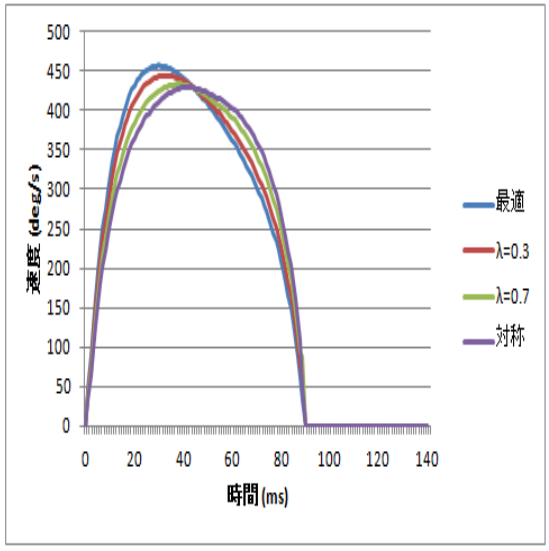

図 12. 30 deg 速度波形の比較

5.3.3　40 deg での比較

表 5. 40 deg での f の変化

λ	f
0 （最適）	6.4567
0.1	6.4610
0.2	6.4737
0.3	6.4947
0.4	6.5241
0.5	6.5618
0.6	6.6079
0.7	6.6623
0.8	6.7250
0.9	6.7961
1 （対称）	6.8943

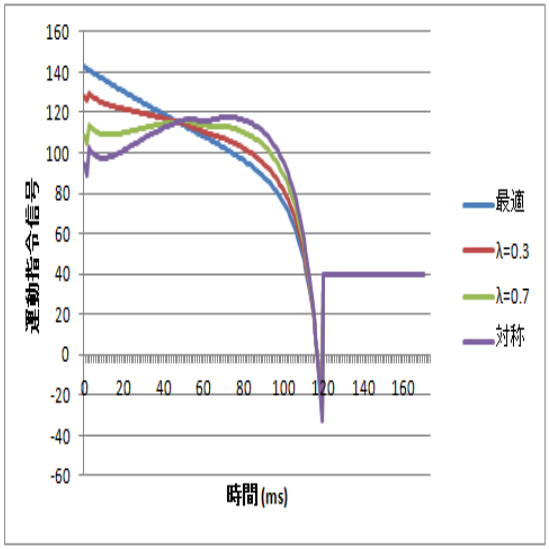

図 13. 40 deg 運動指令の比較

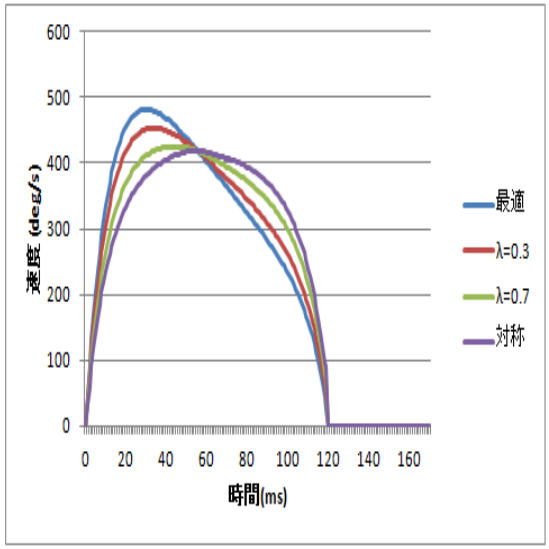

図 14. 40 deg 速度波形の比較

5.3.4　50 deg での比較

表 6. 50 deg での f の変化

λ	f
0 （最適）	7.3059
0.1	7.3119
0.2	7.3298
0.3	7.3595
0.4	7.4010
0.5	7.4543
0.6	7.5195
0.7	7.5965
0.8	7.6854
0.9	7.7861
1 （対称）	7.9193

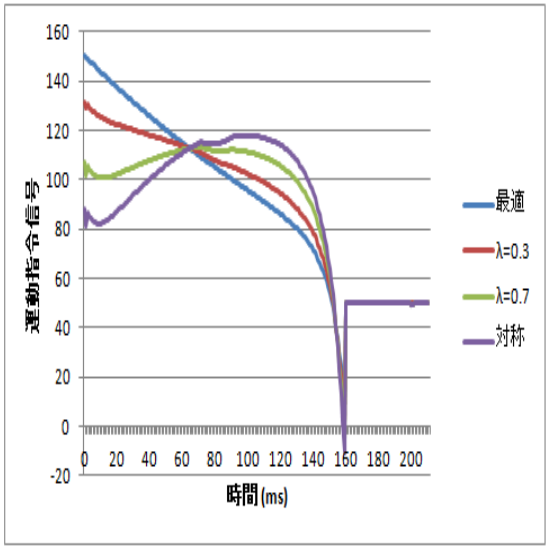

図 15. 50 deg 運動指令の比較

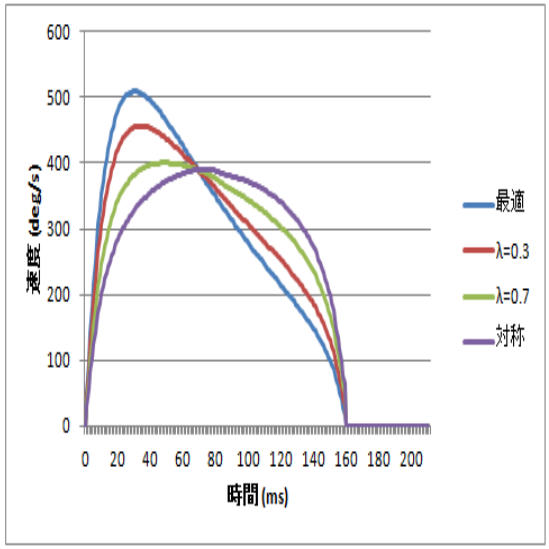

図 16. 50 deg 速度波形の比較

5.4　実験４：一定区間同値条件を加えて

5.4.1　f の値

結果をまとめると表 7〜表 9 のようになった．やはり最適条件と f に大差なかった．

表 7. 5 deg での実験 4 の f

一定期間 D (ms)	f
0	0.5140
10	0.6080
20	0.6427

表 8. 20 deg での実験 4 の f

一定期間 D (ms)	f
0	3.2448
10	3.4144
20	3.4217
30	3.4464

表 9. 40 deg での実験 4 の f

一定期間 D (ms)	f
0	6.4567
10	6.5050
20	6.5085
30	6.5236

より最適解近傍となるようにランダムな値 a を $-0.1, \ldots, 0.1$ （刻み幅は 0.1） と $-0.01, \ldots, 0.01$ （刻み幅は 0.01）に変更して行った実験 2 と実験 4 の f を比較する． f の値がほぼ等しい時の実験 2 と実験 4 の運動指令信号と速度波形を比較をするために各図，それぞれ A,B,C の場合を調べた．

5.4.2　5 deg での比較（ケース A）

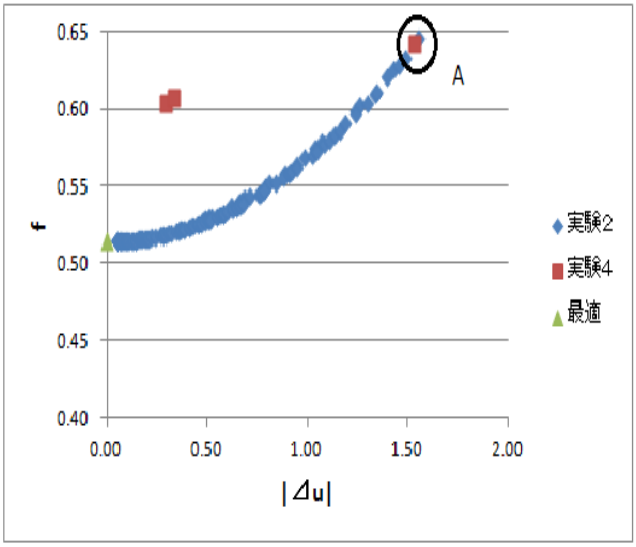

図 17. 5 deg での実験 2 と実験 4 の f の比較

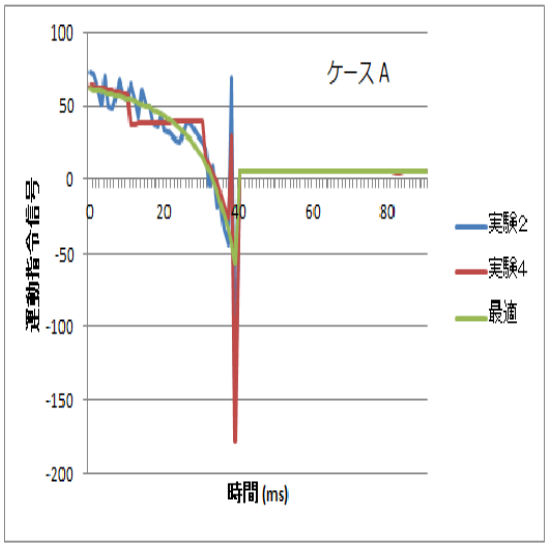

図 18. 5 deg 運動指令の比較（A の場合）

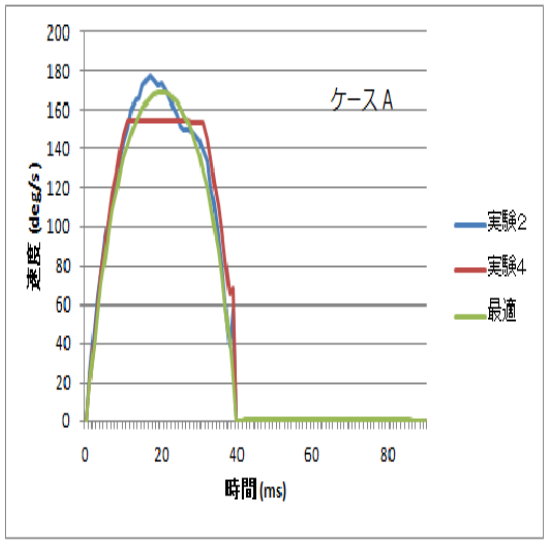

図 19. 5 deg 速度波形の比較（A の場合）

5.4.3　20 deg での比較（ケース B）

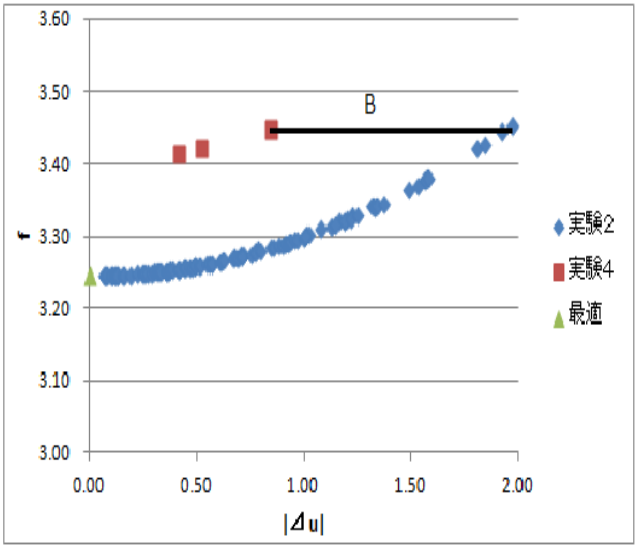

図 20. 20 deg での実験 2 と実験 4 の f の比較

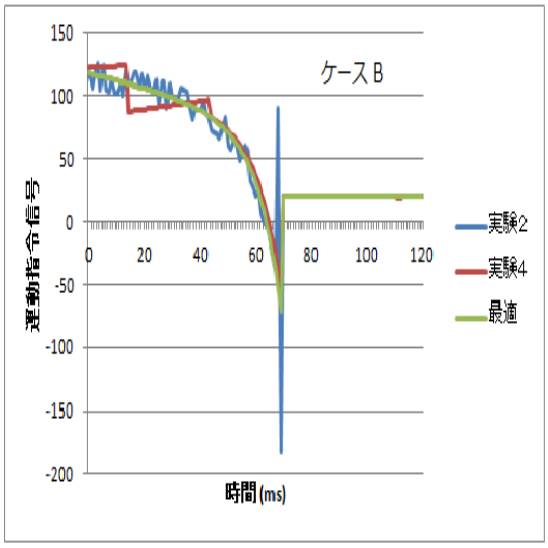

図 21. 20 deg 運動指令の比較（B の場合）

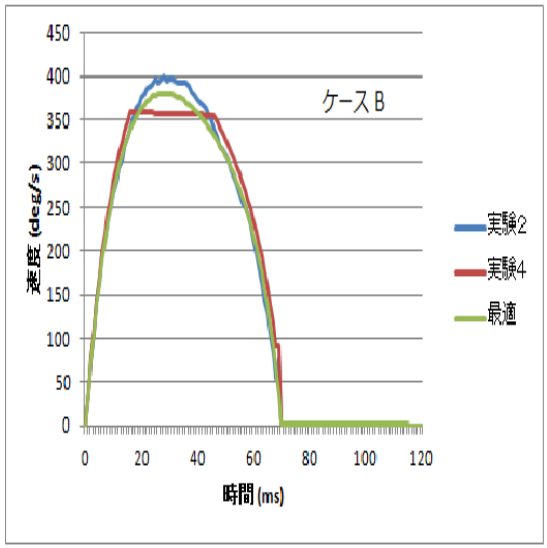

図 22. 20 deg 速度波形の比較（B の場合）

5.4.4　40 deg での比較（ケース C）

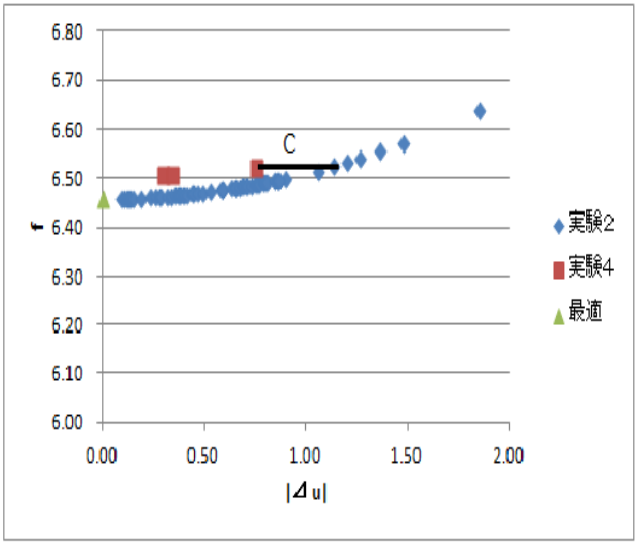

図 23. 40 deg での実験 2 と実験 4 の f の比較

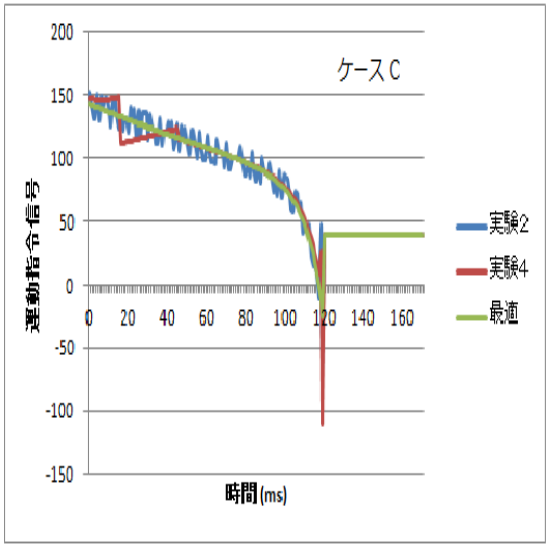

図 24. 40 deg 運動指令の比較（C の場合）

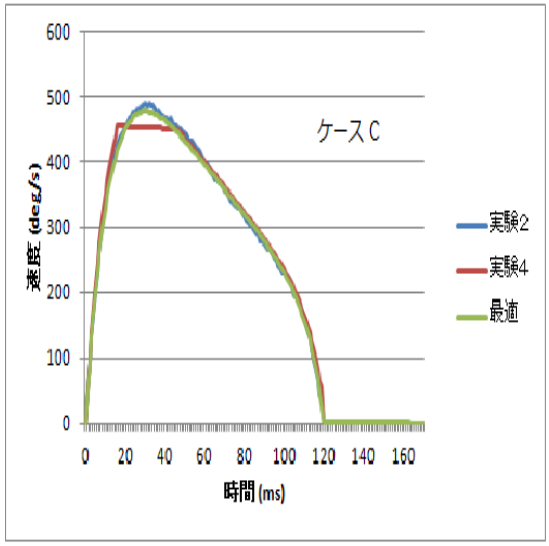

図 25. 40 deg 速度波形の比較（C の場合）

以上より，最適値の近傍では 終点分散 f が小さくなる，現実離れしている速度軌跡になるような運動指令信号が存在する．

6　結論

Harris と Wolpert の信号依存ノイズの存在を仮定した「終点分散最小規範」は実際の到達運動を表現できる．しかし，速度波形を現実離れした条件に変更しても，特に大きな振幅において終点分散 f は大差のない結果となった．よって彼らの仮定「終点での分散を小さくする」規範だけでは不十分で，他にも要素が必要ではないかと考えられる．今後は，この他の何かの要素を導きだしていくことが課題である．

謝辞

本研究について，様々な助言，指導をして下さった藤田昌彦教授に深く感謝の意を述べます．また，基礎となる MATLAB プログラムを提供して頂いた井口尚彦博士，線形システムの基礎を教授して頂いた檜山隼人氏に深く御礼いたします．

参考文献

1) C. Harris and D. Wolpert : “ Signal-dependent noise determines motor planning, ” Nature, vol. 394, pp. 780 − 784, 1998.
2) 井口尚彦，阪口豊，石田文彦：生体ノイズ特性に基づく終点分散最小規範の再検討，電子情報通信学会論文誌， Vol. J87-D-2 No.4, pp.999 − 1007, 2004.
3) H. Collewijn, J. Erkelens and M. Steinman :“Binocular co-ordination of human horizontal saccadic eye movements,” Journal of physiology, vol. 404, pp. 157 − 182, 1988.

付録１：線形回帰直線の傾き

実験２で得られた図５に

$$y = x^a$$

$$\log y = a \log x$$

ここで $Y = \log y, X = \log x$ とすると

$$Y = aX$$

として， a の値を求める線形回帰直線の傾きを調べると

表 A1. 振幅と線形回帰直線の傾き

振幅 r (deg)	線形回帰直線の傾き a
5	1.1995
10	1.0864
20	1.0829
30	1.0644
40	1.1100
50	1.0765

表 A1 になり，平均は $\bar{a} = 1.1033$ となった．

付録２：$f-|\Delta u|$ 近傍

実験２と実験４の f を比較するために行った，ランダムな値 a を変更した時の $f-|\Delta u|$ のグラフを図 A1，図 A2 に記す．なおこの検証のサンプル数は２つとも 100 で，実験４と同じ振幅で行った．

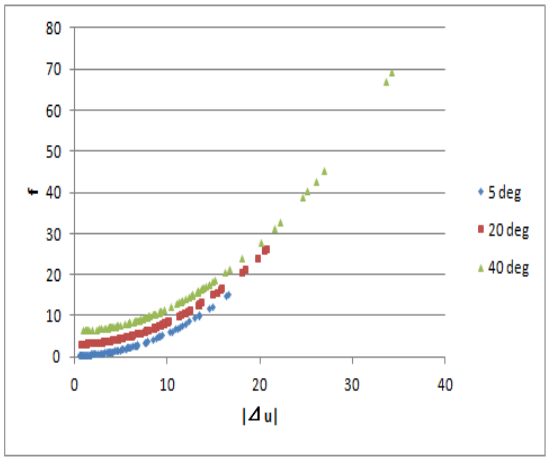

図 A1. $a=\pm 0.1$ での $f-|\Delta u|$ グラフ

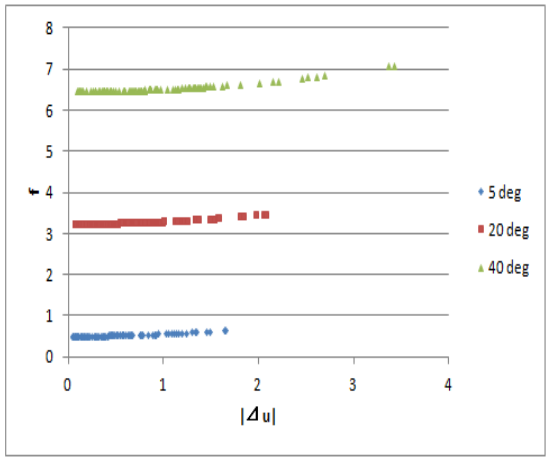

図 A2. $a=\pm 0.01$ での $f-|\Delta u|$ グラフ

付録３：u_{T-2},u_{T-1} の調整

実験２において，u_{T-2}, u_{T-1} をどのように調整したかを記す．(4) 式において

$\boldsymbol{A}=\begin{pmatrix}1 & 1\\ -a & c\end{pmatrix}$，$\boldsymbol{b}=\begin{pmatrix}0\\ a\end{pmatrix}$，$\boldsymbol{x}_{T-1}=\begin{pmatrix}x_1\\ x_2\end{pmatrix}$，$\boldsymbol{x}_{T-2}=\begin{pmatrix}\acute{x}_1\\ \acute{x}_2\end{pmatrix}$ として，到達点を r とすると

$$\begin{aligned}x_T &= \begin{pmatrix}1 & 1\\ -a & c\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}+\begin{pmatrix}0\\ a\end{pmatrix}u_{T-1}\\ &= \begin{pmatrix}x_1+x_2\\ -ax_1+cx_2+au_{T-1}\end{pmatrix}\\ &= \begin{pmatrix}r\\ 0\end{pmatrix}\end{aligned}$$

よって， $u_{T-1}=\frac{1}{a}\left(ax_1-cx_2\right)$ となる．同様に

$$x_{T-1}=\begin{pmatrix}\acute{x}_1+\acute{x}_2\\ -a\acute{x}_1+c\acute{x}_2+au_{T-2}\end{pmatrix}$$

$$= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

よって， $u_{T-2} = \frac{1}{a}\,[r - (1-a)\acute{x}_1 - (1+c)\acute{x}_2]$ となるように調整した．

付録４：使用プログラム

最適化計算を行った MATLAB の計算プログラムと実行プログラムを記す．論文での表記の都合上，「」 で括ってある部分は実際のプログラムではコメントである．同様の理由で一部表記を変更している．

計算プログラム

```
function eo= calculation( fname, d, T0, R0, tick, Nfmax, num)
global N T f1 Aeq beq
filename = strcat('./data/', fname, 'cond.mat');

save(filename, 'R0');
save(filename, 'tick', '-append');

T = round(T0 / tick); 「————————step number」
R = round(R0 / tick); 「————————step number」
N = T + R; 「————————total step number」
N0= T0+R0;

「———————— model paramater」
T1= 224;
T2= 13;
k= 0.01;

T12 = 1.0 / (T1 * T2);

Ad= [ 0          1; ... 「————first component gives position.」
-T12 -(T1+T2) * T12]; 「————————second component gives velocity.」

B= tick * [ 0; T12];

order = size(Ad) * [0; 1];
```

A= expm(tick * Ad); 「———————— A= tick * Ad + eye(order)」;

save(filename, 'T1', '-append');
save(filename, 'T2', '-append');
save(filename, 'A', '-append');
save(filename, 'Ad', '-append');
save(filename, 'B', '-append');
save(filename, 'k', '-append');
save(filename, 'T0', '-append');
save(filename, 'R0', '-append');
save(filename, 'd', '-append');

「————————————————関数 fmincon に使う引数の初期設定」
f1= zeros(N,N);
Aeq= zeros(N,N);
beq= zeros(N,1);
Veq= zeros(N,N);
Peq= zeros(N,N);
vel= zeros(N,1);
pos= zeros(N,1);
V= zeros(N,1);
Vt= zeros(N,1);
LB = [];
UB = [];

「———————————————— options of fmincon」
options= optimset('MaxIter',Nfmax);
options= optimset('Algorithm','sqp');
options= optimset(options, 'MaxFunEvals',Nfmax);
options= optimset(options, 'TolFun', 3e-6);

「————————————————行列 H の作成 H=k*Σ (A^{nt-i}B)($A^{nt-i}B$).$'$ $*$ $q(i)$」
for nt= T:N 「———————— nt=T で視標に到達して nt=N まで位置を keep する.」
for i= 1:nt-1

C= k*(A$^{nt-i} * B) * (A^{nt-i} * B).'$;
f1(nt,i) = C(1,1); 「———————— scalar」
end
C= 0.25*k* B * B.';
f1(nt,nt) = C(1,1); 「———————— scalar」
end

「————————————終点が d になるように設定」
for nt= T:N 「———————— T2:N2」
for i= 1:nt-1
C= A$^{nt-i} * B$;
Aeq(nt,i)= C(1,1);
end
C= 0.5*B;
Aeq(nt,nt)= C(1,1);
end
for nt= T:N
beq(nt)= d; 「————— T〜N の間、眼球を位置 d に保持する beq= zeros(N2,1)」
end

「———————— u0 は運動指令ベクトルの初期値」
u0= zeros(N,1);
u1= zeros(N,1);

[u1,op]= fmincon(@funV0, u0, [], [], [], [], LB, UB, @funcon0, options);
Nit= op.iterations;
Nfun= op.funcCount;

pause(1);

「———————————— step nt における分散の計算」
for nt= 1:N
V(nt) = f1(nt,1:nt); 「———— V(nt) は列ベクトル．ただし f1(1:T-1,i)=0 に注意」
end

「——————終点分散の計算」
sigV= sum(V(T:N)); 「————————— tick^{2}は $B*B.'$の中に入っている」

fprintf(1);
filename = strcat('. data ; fname, 'A', num2str(d), 'T', num2str(T0), 'P');
save(filename, 'u1');
save(filename, 'Nit', '-append');
save(filename, 'Nfun', '-append');
u0 = u1;

「——————振幅（5, 10, 20, 40 deg）毎のグラフ表示　横軸は step number」
ut=zeros(N,1);
ut=1:N;
yt=zeros(N0,1);

figure(num);
subplot(3,1,1);
plot(ut,u0);「——— hold on」
axis([0 180/tick -100 150]);
hold on

for nt=1:N 「——— 1:N2」
for i= 1:nt-1
C= A$^{nt-i}*B$;
Veq(nt,i)= C(2,1);
Peq(nt,i)= C(1,1);
end
C= B;
Veq(nt,nt)= C(2,1);
Peq(nt,nt)= C(1,1);
end
vel=Veq * u0;
pos=Peq * u0;
for nt=1:T

```
fprintf('f',vel(nt));
end
subplot(3,1,2)
「————————— figure(2)];
plot(ut,vel);「————————— hold on」;
axis([0 180/tick -0.1 0.6])
hold on;

subplot(3,1,3)
「————————— figure(3);」
plot(ut,pos);「————————— hold on;」
axis([0 180/tick 0 60])
hold on;

fclose('all');

「————— Target function」

function f= funV0(u)
global f1 N T
「————————————各時間 step の分散」
for nt= T:N
Vt(nt)= f1(nt,1:nt);
end
「————————————終点分散の算出」
f= sum(Vt(T:N));

「————— Terminal Condition」

function [cineq, ceq] = funcon0(u)
global N Aeq beq
cineq = zeros(N,1);
ceq = Aeq * u - beq; 「—— Aeq= zeros(N,N); beq= zeros(N,1);」
```

実行プログラム

```
delta=1.0;
calculation('P0.dat', 5, 40, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 1);
calculation('P1.dat', 10, 50, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 2);
calculation('P2.dat', 20, 70, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 3);
calculation('P3.dat', 30, 90, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 4);
calculation('P4.dat', 40, 120, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 5);
calculation('P5.dat', 50, 170, 50, delta, 2.0, -0.1, 2.0, 100000/delta, 6);
```