UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE SISTEMAS E ENERGIA

IA–856 Exercício Computacional
Controle Linear Quadrático

Giovanni Chemello Caprio
RA: 211483

4. (Exercício Computacional) Considere o mesmo sistema proposto por você para a identificação não-paramétrica, com função de transferência discreta $Y(z)/U(z) = G_u(z)$. Considere duas possibilidades para a função de transferência entre o ruído e a saída Y, $Y(z)/E(z) = G_e(z)$:
I) $G_e(z) = \sigma_e$, isto é, a representação completa é $Y(z) = G_u(z)U(z) + \sigma_e E(z)$,
II) $G_e(z) = \sigma_e G_u(z)$, isto é, $Y(z) = G_u(z)(U(z) + \sigma_e E(z))$
em que $e_k, k \geq 0$ forma uma sequência de ruído branco gaussiano padrão.

   (a) Adote uma representação de estados para este sistema considerando: y – saída; u – entrada de controle; e – ruído a ser considerado como nos casos I) ou II) acima, com variância $\sigma_e^2 = 0{,}04$.

   (b) Adote um custo por estágio na forma $\|y_k + f u_k\|^2$ com $f = 1$. Para a solução de horizonte infinito, verifique antes as condições para a existência e unicidade das equações algébricas de Riccati do controle.

   (c) (observação completa de estados) Obtenha as soluções de controle e o custo ótimo para os problemas LQ determinístico ($e_k = 0, \forall k \geq 0$) e LQG com os ruídos nos casos I) e II) acima nas seguintes situações:

      i. Horizonte $N = 3$ adotando-se $Q = I$;
      ii. Horizonte infinito para o LQ;
      iii. Horizonte infinito e custo descontado para o LQG (adote $\rho = 0{,}9$);
      iv. Horizonte infinito e custo médio por unidade de tempo para o LQG.

      Dica – Para resolver a equação algébrica de Riccati utilize a rotina `dare` do Matlab.

   (d) Examine o impacto em algumas das soluções de controle ao se escolher $f = 0{,}001$ e $f = 1000$ no custo por estágio.

   (e) (observação parcial de estados) Supondo agora que só a saída y_k a cada instante k seja disponível para o controle, resolva o problema de controle ótimo neste cenário, detalhando-o completamente.

Linear Quadrático Determinístico

Dado a Gu escolhida sendo: `Gu = tf(540, conv([1 6],[1 5 72])`. Foi transformada em discreta, amostrada e gerada em espaço de estados a fim de análises posteriores. Sendo:

- A

-0.0389	-0.0531	0.0171
0.1250	0	0
0	0.0078	0

- B

4
0
0

- C

0.2947	0.2407	1.9391

Assim, para nossas analises, testamos a controlabilidade entre (A,B) e observabilidade de (A,C) e foi visto que o sistema era controlável e observável. Partindo então para os cenários de controle.

i. N = 3 e Q = I

Para esse cenário, geramos um x_0 inicial com valor de [1;2;3]. Com isso, tivemos que fazer a relação de Filtro de Kalman, para gerar os x estimados futuros e gerar o controle ótimo u* e os custos ótimos associados por estágio. Obtendo:

u* = (-1.0336 , 0.2033)

Custos = (28.6085, 14.3674 , 1.18)

Custo Total = 44.1558 Custo Médio = 14.71

ii. Horizonte Infinito

Caso não tenhamos os N's que levarão nosso controle ótimo a cada iteração, podemos calcular o controle que gera o menor custo em relação ao infinito.

Para isso, vamos gerar a matriz S via Eq. Algébricas de Riccati. Mas para isso, devemos provar que (A,B) é estabilizável e (D*,A*) é detectável. (A,B) é detectável, justamente por (A,B) ser controlável. O que foi calculador anteriormente. (D*,A*) não é observável, mas podemos calcular o autovalor que não está no circulo unitário e verificar se é observável, o que foi provado. Assim, temos as condições de existência e unicidade de solução.

Com isso, calculamos a solução pelo código de matlab dare, que devolve a matriz S que minimiza o custo no infinito.

Os resultados obtidos foram:

$$u^1 = (-0.3257)$$

$$\text{Custo} = (14{,}0159)$$

iii. Variação de f

- Para f=0.001

Observamos um que para:

- N = 3: u* = (-0.0114 , 0.000237) / Custo = (12.1926, 0.0474, 0.00014);

- Infinito: u^1 = (0.0231) / Custo = (14.0159);

Assim, os custos e ações de controles foram diminuidos.

- Para f=1000

Observamos um que para:

- N = 3: u* = (-0.0121, 1.5259e-06) / Custo = (24.3873, 0.0124, 0.0001);

- Infinito: u^1 = (-0.0060) / Custo = (14.0159);

Assim, os custos e ações de controles foram aumentados.

Linear Quadrático Gaussiano

Adicionado o ruído gaussiano com variância 0.04. Percebe-se que a planta continuou observável e controlável. Assim, os resultados para o controle finito com N= 3 e Q = I, adotando um x inicial na origem:

$$u^*=(0,0)$$

$$\text{Custo} = (200.26,\ 9.377,\ 1.7583)$$

Utilizando o código do LQ para determinação do Infinito, percebeu-se que a modificação gerou um (A,B) estabilizável e um (D*,A*) detectável. Assim,

$$u^1 = (-0.3257)$$

$$\text{Custo} = (14{,}0159)$$

Para a mesma planta, o sistema não apresentou mudanças no custo no infinito para ruídos, isso deve-se ao fato do ruído ser gaussiano com média nula.

- Para f=0.001

Observamos um que para:

- N = 3: u* = (-0.0121, 1.5259e-06) / Custo = (294.8, 21.31 , 0.0890);

- Infinito: u^1 = (0.0231) / Custo = (14,0159);

Assim, o custo no caso finito aumentou.

- Para f=1000

Observamos um que para:

- N = 3: u* = (-0.0121, 1.5259e-06) / Custo = (329.88, 33.432, 2.2560);

- Infinito: u^1 = (-0.0060) / Custo = (14,0159);

Para o infinito, foi caculado com um p=0.9. Isso ocorre para dar menos importancia ao futuro das predições. Com isso, obtivemos:

u^1 = (-0.3263)

Custo = (14,0143)

Devido ser muito próximo de 1, não obteve tanta diferença.

Código LQ

```
% Exercicio 4 - LQG
clc, close all, clear all

%% Sistema Escolhido não-paramétrico
Ts = 1; % Tempo a cada amostra
Gu = tf(540, conv([1 6],[1 5 72]));     % F.T. continua
Gud = c2d(Gu,Ts);                          % F.T. discreta
sys = ss(Gud);       % Espaço de Estados

Co = ctrb(sys.A,sys.B)
rank(Co) % testa controlabilidade
Ob = obsv(sys.A,sys.C)
rank(Ob) % testa observabilidade

%% LQ Deterministico (wk=0) (x_k+1 = sys.a * x_k + sys.b * u_k)

% i)(Finito com N=3)

N = 3;
Q = eye(3); %Q = I
```

```
% cálculo dos ganhos de controle ótimo M(j) e S(N)
S{N} = Q;
A = sys.a; B=sys.b; F = 1; D = 1;

for j=N-1:-1:1
    M{j} = ((B'*S{j+1}*A + F'*D)/(B'*S{j+1}*B + F'*F));
    S{j} = A'*S{j+1}*A + D'*D - (A'*S{j+1}*B + D'*F)*(B'*S{j+1}*A + F'*D);
end

% cálculo do filtro de Kalman
F = sys.a; G = sys.b; H = sys.C; D =  sys.D;
P{1} = F*Q*F'; % covariancia inicial
L=[0 0 1; 0 1 1; 1 1 1];
R=1;
x{1} = sqrt(5)*L*randn(3,1); % gera condição inicial
y{1} = H*x{1}; % gera observação inicial
xa = [1 2 3]'; % valor do estado a priori [k|k-1]
xe{1} = xa;

for k=1:N-1
    % Measurement update
    K{k} = P{k}*H'/(H*P{k}*H'+R); % Ganhos de kalman
    P{k+1} = (eye(3) - K{k}*H)*P{k}; % Proxima Cov

    % evolução do sistema verdadeiro:
    u{k} = -M{k}*xa; % controle ótimo que minimiza o custo
    x{k+1} = F*x{k} + G*u{k};
    y{k+1} = H*x{k+1};

    %valor estimado:
    xa = F*xe{k} + G*u{k};     % x[k+1|k]
    xe{k+1}= xa + K{k}*(y{k+1}-H*xa);  % x[k+1|k+1]

    % custo no estágio
    c(k) = x{k}'*Q*x{k} + u{k}'*R*u{k};
end
% custo final
c(N) = x{N}'*S{N}*x{N};

% ii)(Infinito)

A = sys.a; B=sys.b; D = 1; F =  1;
% teste de estabilizavel e detectavel
%(A,B) é estabilizavel pois (A,B) é controlável
D_ = (eye(3)-F*inv(F'*F)*F')*D;
A_ = A-(B*inv(F'*F)*F'*D);
a = A_(1,:);
Ob2 = obsv(D_,a)
rank(Ob2) % testa observabilidade para verificar detecbilidade
eig(A_) % verifica o pólo não observável para analise
aa = [-400*eye(3) - a; D_]% verifica a detectabilidade no
rank(aa)                   % no pólo fora do circulo unitario
[X,L,G] = dare(A,B,Q,R) % Valor do Ricatti
Mx=inv(B'*X*B+F'*F)*(B'*X*A+F'*D);
x_inf = [1 2 3]';
u_inf = -Mx*x_inf %controle ótimo infinito
V_inf = x_inf'*X*x_inf %custo no infinito
```

Código LQG

```
% Exercicio 4 - LQG
clc, close all, clear all

%% Sistema Escolhido não-paramétrico
Ts = 1; % Tempo a cada amostra
Gu = tf(540, conv([1 6],[1 5 72]));     % F.T. continua
Gud = c2d(Gu,Ts);                          % F.T. discreta
sys = ss(Gud);         % Espaço de Estados

Co = ctrb(sys.A,sys.B)
rank(Co) % testa controlabilidade
Ob = obsv(sys.A,sys.C)
rank(Ob) % testa observabilidade

%% Caso Ruido I)

% ruido de medida y_k = C*x_k + sqrt(var_y)* e_k  com e_k ~ N(0,1)
var_y = 0.04;

% ruido do sistema x_k+1 = sys.a * x_k + sys.b * u_k + F*w_k
F = [1 0 0;0 1 0;0 0 1];
Cor = ctrb(sys.A,F); rank(Cor) % testa controlabilidade através do ruído

% desempenho E[ sum_1^(N-1) [x_k'*Qc*x_k +u_k*Rc*u_k] + x_N'*Sf*x_N]
% problema inicia-se no instante k=1 com x_1 ~ N( m_1, P_1)
L = [1 0 0; 1 1 0;1 1 1];
P_1 = 500*L*L';
m_1 = [0 0 0]';

% i)

N = 3;
R = var_y;
Q = eye(3); %Q = I

% cálculo dos ganhos de controle ótimo M(j) e S(N)
S{N} = Q;
A = sys.a; B=sys.b; D = 1; F=1; C=sys.c;

for j=N-1:-1:1
    M{j} = ((B'*S{j+1}*A + F'*D)/(B'*S{j+1}*B + F'*F));
    S{j} = A'*S{j+1}*A + D'*D - (A'*S{j+1}*B + D'*F)*(B'*S{j+1}*A + F'*D);
end

% cálculo do ganho do filtro de Kalman
P{1} = P_1;
for k=1:N-1
    % Measurement update
    K{k} = A*P{k}*C'/(C*P{k}*C'+R);
    P{k+1} = A*P{k}*A'+ F*Q*F' - K{k}*(C*P{k}*C'+R)*K{k}';
end
```

```
% Controle
x{1} = sqrt(500)*L*randn(3,1); % gera condição inicial
y{1} = C*x{1} + sqrt(var_y)*randn; % gera observação inicial
xa = m_1; % valor do estado a priori [k|k-1]
xe{1} = xa;

for k=1:N-1
    % evolução do sistema verdadeiro:
    u{k} = -M{k}*xa;
    x{k+1} = A*x{k} + B*u{k} + F* randn(3,1);
    y{k+1} = C*x{k+1} + sqrt(var_y)*randn;

    %valor estimado:
    xa = A*xe{k} + B*u{k};     % x[k+1|k]
    xe{k+1}= xa + K{k}*(y{k+1}-C*xa);  % x[k+1|k+1]

    % custo no estágio
    c(k) = x{k}'*Q*x{k} + u{k}'*R*u{k};
end
% custo final
c(N) = x{N}'*S{N}*x{N};
custo = cumsum(c);
plot(c,'*')

%% ii)(Infinito)

A = sys.a; B=sys.b; D = 1; F = 1;
% teste de estabilizavel e detectavel
%(A,B) é estabilizavel pois (A,B) é controlável
D_ = (eye(3)-F*inv(F'*F)*F')*D;
A_ = A-(B*inv(F'*F)*F'*D);
a=A_(1,:);
Ob2 = obsv(D_,a)
rank(Ob2) % testa observabilidade para verificar detecbilidade
eig(A_) % verifica o pólo não observável para analise
aa = [-3.9*eye(3) - a; D_]% verifica a detectabilidade no
rank(aa)                    % no pólo fora do circulo unitario
[X,L,G] = dare(A,B,Q,R) % Valor do Ricatti
Mx=inv(B'*X*B+F'*F)*(B'*X*A+F'*D);
x_inf = [1 2 3]';
u_inf = -Mx*x_inf %controle ótimo infinito
V_inf = x_inf'*X*x_inf %custo no infinito

%% descontado(Descontado)

p = 0.0001;
p2 = p^(1/2);

A = p2*A; B = p2*B; F = p2*F; D= p2*D;
[Xp,L,G] = dare(A,B,Q,R) % Valor do Ricatti
Mxp=inv(B'*Xp*B+F'*F)*(B'*Xp*A+F'*D);
x_infp = [1 2 3]';
u_infp = -Mxp*x_infp %controle ótimo infinito
V_infp = x_infp'*Xp*x_infp %custo no infinito
```