UDESC – Universidade do Estado de Santa Catarina PPGEEL – Programa de Pós-Graduação em Engenharia Elétrica CDI - Controle Digital Professor: Antonio Silveira (antonios@joinville.udesc.br) Aula 10

## REGULADOR LINEAR QUADRÁTICO

#### 1. Sistemas discretos no espaço de estados

Estivemos estudando técnicas de controle no domínio de funções de transferência e via representações polinomiais com base tanto em modelos determinísticos como estocásticos pelo uso das representações ARX, ARMAX e ARIMAX. Estas mesmas descrições podem ser representadas também no domínio do espaço de estados, levando assim a característica estocástica para as representações matriciais. Como exemplo, consideremos um sistema ARMAX dado por

$$(1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}) y(k) =$$

$$z^{-1} (b_0 + b_1 z^{-1} + \dots + b_{n_a} z^{-n_b}) u(k) + (1 + c_1 z^{-1} + \dots + c_n z^{-n_c}) \xi(k)$$
(1)

onde y(k), u(k),  $\xi(k)$ , são, respectivamente, a saída, a entrada e uma sequência do tipo ruído branco.

Uma descrição no espaço de estados do modelo ARMAX mostrado em (1) pode ser representada pelas  $n_a$  variáveis de estado

$$x_{1}(k) = -a_{1}x_{1}(k-1) + x_{2}(k-1) + b_{0}u(k-d) + (c_{1} - a_{1})\xi(k-1)$$

$$x_{2}(k) = -a_{2}x_{1}(k-1) + x_{3}(k-1) + b_{1}u(k-d) + (c_{2} - a_{2})\xi(k-1)$$

$$\vdots$$

$$x_{n_{a}}(k) = -a_{n_{a}}x_{1}(k-1) + b_{(n_{a}-1)}u(k-d) + (c_{n_{a}} - a_{n_{a}})\xi(k-1)$$

tal que  $y(k) = x_1(k) + \xi(k)$  seja equivalente a (1) e o modelo no espaço de estados dado por

$$x(k) = \mathbf{A}x(k-1) + \mathbf{B}u(k-d) + \Gamma \xi(k-1),$$
  

$$y(k) = \mathbf{C}x(k) + \xi(k),$$
(2)

onde as matrizes A, B, Γ, C, são descritas por

$$\mathbf{A} = \begin{bmatrix} -\mathbf{a} & \mathbf{I}_{(n_a-1)} \\ \mathbf{0}_{1 \times (n_a-1)} \end{bmatrix}, \tag{3}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0}_{(n_a - n_b) \times 1} \end{bmatrix},\tag{4}$$

$$\Gamma = \begin{bmatrix} (c_1 - a_1) & (c_2 - a_2) & \cdots & (c_{n_a} - a_{n_a}) \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 1 & \mathbf{0}_{1 \times (n_a - 1)} \end{bmatrix},$$
(5)

$$\mathbf{C} = \begin{bmatrix} 1 & \mathbf{0}_{1 \times (n_{\alpha} - 1)} \end{bmatrix},\tag{6}$$

sendo 
$$\mathbf{a}^{\mathsf{T}} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_a} \end{bmatrix}, \ \mathbf{b}^{\mathsf{T}} = \begin{bmatrix} b_0 & b_1 & \cdots & b_{n_b} \end{bmatrix}$$
.

Do ponto de vista matemático, uma descrição ARMAX no espaço de estados é bastante conveniente para, por exemplo, elaborarmos um código de simulação de um sistema discreto ou mesmo para viabilizar outras formas de solução, análise e projeto de sistemas de controle digitais em que as descrições matriciais possam trazer benefícios. No entanto, propõe-se o seguinte experimento de análise:

- 1) Obtenha o equivalente discreto via ZOH do modelo no espaço de estados do Pêndulo Amortecido da aula passada que foi baseado em uma descrição Posição-Velocidade. Simule este sistema para uma entrada degrau e analise as variáveis de estado;
- 2) Obtenha o modelo ARMAX do Pêndulo Amortecido a partir do modelo contínuo por função de transferência e depois reescreva o sistema ARMAX polinomial para o ARMAX no espaço de estados apresentado em (2). Simule este sistema para uma entrada degrau e observe suas variáveis de estado;

Com base nos testes (1) e (2), avalie a diferença do comportamento das variáveis de estado. No teste (1), diga quem são (posição? Velocidade?) as variáveis de estado envolvidas e, depois, faça o mesmo com as variáveis do teste (2).

Os resultados dos testes propostos nos permitem evidenciar que a representação ARMAX no espaço de estados nos gera variáveis de estado que não têm comportamento assintoticamente estável, ou seja, não tendem a zero quando a saída medida torna-se constante em regime permanente. Usaremos esse resultado como uma observação antes de iniciarmos nossos projetos com base no LQR discreto, pois se lembrarmos da revisão do caso contínuo, vimos que nosso problema de controle se resumia em conduzir a variável medida a um valor de referência e as demais variáveis de estado eram conduzidas a zero, como em um problema de regulação. Além disso, a forma de sintonia (ponderação) do controlador, feita por tentativa e erro, exige que se conheça de maneira razoável o comportamento das variáveis envolvidas no problema, ou então não há como adotar um critério coerente de sintonia e o resultado é obtido simplesmente pelo mero acaso de testes sem sentido.

### 2. Projeto do LQR discreto: tratamento do modelo de projeto

Em toda atividade de projeto de controlador baseado em modelo, nosso projeto se inicia exatamente nessa base, ou seja, o modelo. Este precisa "explicar" o se quer controlar e nada mais. Isso se deve tanto a tentativa de conter a complexidade do controlador resultante como também para não incluir informações desnecessárias no modelo de projeto e que são posteriormente desprezadas no momento da sintonia do controlador.

Deste modo, vamos estabelecer uma divisão entre modelos de simulação e modelos para o projeto de controladores. Faremos então o desenvolvimento do nosso projeto diretamente sobre um exemplo de controle de posição do sistema pendular estudado anteriormente. Vamos tomar como objetivo o amortecimento das oscilações e o rastreamento da posição com erro nulo em regime permanente.

O modelo contínuo por função de transferência do Pêndulo Amortecido é descrito por

$$G(s) = \frac{Y(s)}{U(s)} = \frac{3.48}{s^2 + 0.8s + 4} \tag{7}$$

Como um dos nossos interesses nesse projeto é o de conter as características oscilatórias do sistema, a representação de segunda-ordem mostrada em (7) é suficiente para caracterizar o

comportamento das variações da posição, ou seja, transportar a informação da velocidade de resposta à lei de controle resultante. A obtenção da descrição no espaço de estados, ao menos para este caso SISO, é bastante simples de ser conduzida garantindo o conhecimento total sobre o que cada variável de estado resultante irá representar, ou seja, podemos reescrever (7) da seguinte maneira:

$$\ddot{y} + 0.8\dot{y} + 4y = 3.48u \tag{8}$$

A partir dessa nova representação mostrada em (8) podemos caracterizar que  $x_1 = y$  seria a resposta e que  $x_2 = \dot{x}_1$  seria a variação da resposta. Note que propositalmente estamos endereçando essas definições não como posição ou velocidade, mas como resposta e variação da resposta, justamente para enfatizar que isto pode ser feito para qualquer sistema dinâmico.

Além da compensação das oscilações do pêndulo, queremos garantir o rastreamento de referência com erro nulo. Como não forçamos uma estrutura de controlador fixa, mas sim estamos considerando que o controlador final é função do modelo nominal de projeto, a característica integradora também precisa estar presente no modelo ou então não existiria tal ação de controle na lei resultante. Faremos então a inclusão do integrador diretamente sobre o sistema representado em (8), de tal forma que este passa a ser representado por

$$\ddot{y} + 0.8 \, \ddot{y} + 4 \, \dot{y} = 3.48 \, \dot{u} \tag{9}$$

onde  $x_1=y,\ x_2=\dot{y},\ x_3=\ddot{y}$ , que em outras palavras explica o aparecimento da variação de u que já nos remete a lembrança do nosso operador  $\Delta=1-z^{-1}$  que nos permite utilizar a forma incremental no domínio discreto, ou seja,  $\Delta u(k)$ .

Tenha em mente que os passos que realizamos até este momento não são as únicas formas para a inclusão do integrador. Na verdade existem muitas outras maneiras, mas nem todas resultam na mesma representação que queremos obter, tal que seja possível escrever nosso modelo nominal de projeto discreto em função de  $\Delta u(k)$ . Mais adiante veremos uma forma mais simples de inclusão do integrador e que é feita diretamente com a manipulação das matrizes do sistema no espaço de estados e que torna mais simples a inclusão para o caso MIMO. No entanto, a forma realizada neste momento do nosso estudo favorece o entendimento de que estamos literalmente manipulando o modelo de projeto para que ele possua as características que o nosso problema requer.

Reescrevendo-se o sistema em (9) após a substituição das variáveis de estado fornece:

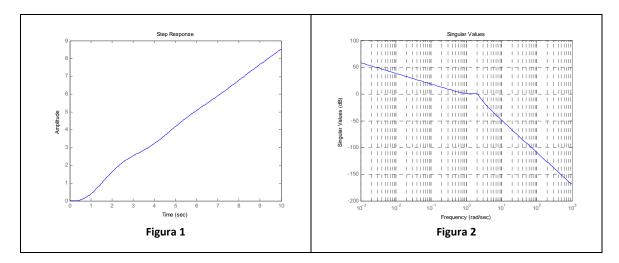
$$\dot{x}_1 = x_2 
\dot{x}_2 = x_3 
\dot{x}_3 = -0.8x_3 - 4x_2 + 3.48\dot{u}$$
(10)

Escrevendo (10) na forma matricial obtém-se

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & -0.8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 3.48 \end{bmatrix} \dot{u}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{C} x$$
(11)

Note que se simularmos o sistema mostrado em (11) para uma entrada degrau veremos a saída em rampa devido a presença do integrador (Fig. 1). Da mesma maneira, visualizando o diagrama de magnitude em frequência do sistema em (11), veremos a queda na taxa de 20dB por década (Fig. 2). Esses resultados são mostrados nas figuras 1 e 2 a seguir:

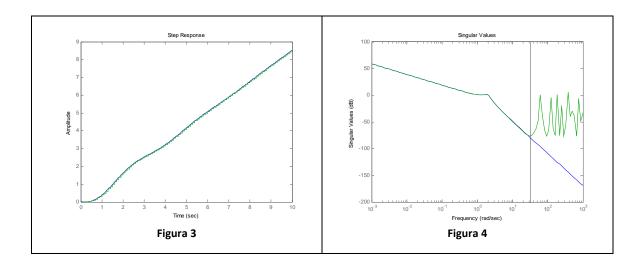


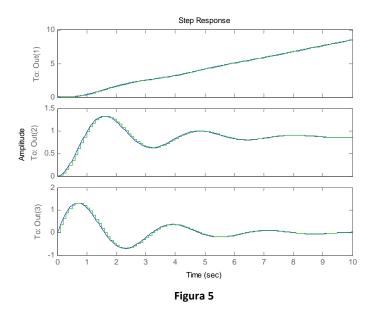
Se discretizarmos o sistema em (11) teremos a garantia de que sua estrutura será a que projetamos. Para um  $t_s=0.1$  segundos o equivalente discreto via ZOH do sistema em (11) é descrito por

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0.09935 & 0.004853 \\ 0 & 0.9806 & 0.09547 \\ 0 & -0.3819 & 0.9042 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \end{bmatrix} + \begin{bmatrix} 0.0005675 \\ 0.01689 \\ 0.3322 \end{bmatrix} \Delta u(k-d)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(k)$$
(12)

Nas figuras 3 e 4 são mostrados os equivalentes aos gráficos das figuras 1 e 2 confrontando o sistema discreto com o sistema contínuo. Na Figura 5 são apresentadas as sequências das variáveis de estado no caso contínuo e discreto para enfatizar a equivalência. Note que há um pequeno defasamento entre os gráficos devido ao atraso de amostragem existente no sistema discreto.





# 3. LQR discreto com realimentação total de estados

De maneira similar ao caso contínuo, o problema do LQR discreto se baseia em encontrar o ganho ótimo, K, do controlador, cuja lei de controle

$$u(k) = -Kx(k) + y_r(k)$$
 (13)

minimiza a função custo quadrática discreta, dada por

$$J = \sum_{0}^{\infty} \left[ x^{T}(k)Qx(k) + u^{T}(k)Ru(k) \right]$$
(14)

considerando o caso estático ou de horizonte infinito, que em outras palavras estabelece que os estados tendem a zero quando o tempo tende a infinito.

A solução do problema de otimização envolvendo as equações (13) e (14) é obtida a partir da solução da Equação a Diferenças de Riccati de Controlador, descrita por

$$P(k+1) = A^{T} P(k) A - A^{T} P(k) B \left( B^{T} P(k) B + R \right)^{-1} B^{T} P(k) A + Q$$
(15)

que está relacionada ao problema dos Mínimos Quadrados conforme vimos nas aulas anteriores. De maneira semelhante ao problema de otimização via MQR, normalmente a matriz P é inicializada com valores elevados quando não se conhece valores adequados na vizinhança da condição ótima para P. Como este procedimento é feito de maneira off-line e assim como no MQR, tem estabilidade e convergência garantida para modelos de projeto controláveis e estáveis, é possível iniciar o algoritmo com valores na ordem de 100 ou maiores e obter a solução ótima para P com apenas algumas centenas de iterações.

O resultado da Eq. (15) é então utilizado para calcular o ganho ótimo da seguinte forma:

$$K = \left[ A^T P B \left( B^T P B + R \right)^{-1} \right]^T \tag{16}$$

No software MATLAB e similares é comum se encontrar funções nativas que resolvem o problema do LQR contínuo e discreto. No MATLAB, isto é feito pela função dlqr para o caso discreto. Os únicos parâmetros de projeto são as matrizes Q e R que operam de maneira idêntica ao caso contínuo apresentado anteriormente.

#### 3.1. Observações sobre o LQR

A estrutura de um controlador LQR com realimentação total de estados é idêntica a de um controlador no espaço de estados projetado por alocação de pólos. A diferença está justamente no método de obtenção dos ganhos e principalmente nos resultados alcançáveis.

Devido a essa similaridade estrutural os requisitos de projeto também se mantêm, como: o modelo de projeto precisa ser Controlável e a matriz dinâmica deve ser não-singular.

A grande diferença é que não se faz necessário ter de arbitrar a localização dos pólos e zeros de malha fechada e, toda a solução está atrelada ao compromisso entre velocidade de resposta e energia de controle para garantir esse desempenho de velocidade.

#### 3.2. Condições para a convergência do algoritmo LQR

- Existe um ganho K tal que a matriz A-BK seja estável;
- A matriz de saída  $\,C\,$  é de rank cheio e igual a ordem do modelo de projeto, ou seja, todos os estados estão disponíveis;
- A matriz de ponderação do sinal de controle, R, é definida positiva. Isto significa que, no caso MIMO, todas as entradas devem ser ponderadas;
- A matriz de ponderação dos estados, Q, deve ser semidefinida positiva e o par  $(\sqrt{Q},A)$  deve ser detectável i.e., os pólos instáveis precisam ser observáveis.

## 4. Exemplo de projeto

Projete um controlador LQR discreto com realimentação total de estados para o sistema mostrado a seguir:

$$G(s) = \frac{0.8 \times 2^2}{s^2 + 2 \times 0.5 \times 2s + 2^2} = \frac{K_p \omega_n^2}{s^2 + 2 \xi \omega_n s + \omega_n^2}$$
(17)

Execute suas simulações no ambiente Simulink para estudar a estrutura do controlador e avaliar a simplicidade de um controlador composto simplesmente por ganhos que operam sobre os estados do sistema.

Projete o controlador considerando a ação incremental para o rastreamento de referências com erro nulo em regime.

Obs: material de apoio para esse exemplo encontra-se na pasta da disciplina (Aula 09) no Dropbox.

#### 4.1. Resultados do exemplo sobre o sistema em (17)

Reescrevendo-se o sistema em (17) em função das derivadas da saída medida, tem-se que

$$\ddot{y} + (2\xi\omega_n)\dot{y} + (\omega_n^2)y = K_p\omega_n^2 u$$

$$x_1 = y$$

$$x_2 = \dot{y}$$
(18)

que fornece o seguinte modelo no espaço de estados:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 3.2 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
(19)

Note que em (19) estamos considerando que todos os estados são mensuráveis, isto é evidente pela inclusão  $ad\ hoc$  da matriz identidade na matriz C.

Para que o controlador resultante seja capaz de garantir o rastreamento assintótico de referência com erro de regime nulo, precisamos aumentar a descrição em (19) com a inclusão de um integrador. No entanto, devemos destacar que isto só é recomendável para sistemas do Tipo-0, que é o caso em questão. Caso contrário, deve-se aproveitar o integrador natural do processo.

Se recordarmos o que foi feito sobre o primeiro exemplo, na equação (9) para se chegar em (11), podemos generalizar uma forma canônica para a inclusão do integrador em sistemas baseados na forma posição-velocidade como a (19). Essa forma canônica é dada por

$$\dot{x}_{a} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & A \end{bmatrix} x_{a} + \begin{bmatrix} 0 \\ B \end{bmatrix} \dot{u} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & -2 \end{bmatrix} x_{a} + \begin{bmatrix} 0 \\ 0 \\ 3.2 \end{bmatrix} \dot{u}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & C \end{bmatrix} x_{a} = I_{3}x_{a}$$
(20)

Antes de obtermos um equivalente discreto para o modelo aumentado mostrado em (20), temos que decidir qual será o período de amostragem  $T_s$ . Sabemos que a frequência

natural é  $\omega_n = 2$  rad/s. Vamos optar por uma frequência de amostragem pelo menos 10 vezes mais veloz. Deste modo,

$$f_s = 10 \left( \frac{\omega_n}{2\pi} \right) \approx 3.1831 \text{ Hz}$$
 (21)

$$T_s = \frac{1}{f_s} = 0.3 \text{ s}$$
 (22)

O equivalente discreto de (20) para o período de amostragem de 0.3 s é

$$x(k+1) = \begin{bmatrix} 1 & 0.2847 & 0.0361 \\ 0 & 0.8554 & 0.2124 \\ 0 & -0.8495 & 0.4307 \end{bmatrix} x(k) + \begin{bmatrix} 0.0123 \\ 0.1157 \\ 0.6796 \end{bmatrix} \Delta u(k)$$

$$y(k) = I_3 x(k)$$
(23)

Sem considerar restrições ou critério de desempenho neste exemplo, mas apenas para testar o algoritmo do controlador LQR, definimos as matrizes de ponderação como sendo:

$$Q = diag(1 \ 1 \ 1), \quad R = 1$$
 (24)

Execute o m-file respectivo a esta aula para verificar os resultados do cálculo da matriz P feito pela função dlqr do MATLAB e também pela utilização da equação recursiva de Riccati. Note que o resultado é idêntico e igual a

$$P = \begin{bmatrix} 8.3180 & 4.8598 & 1.1396 \\ 4.8598 & 8.1893 & 1.3264 \\ 1.1396 & 1.3264 & 1.5532 \end{bmatrix}$$
 (25)

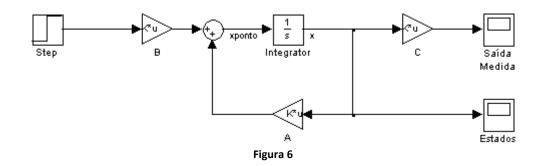
e que possibilitou o cálculo do ganho ótimo

$$K = \begin{bmatrix} 0.6951 & 0.4846 & 0.4755 \end{bmatrix} = \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix}$$
 (26)

Para enfatizar a estrutura do controlador LQR e sua semelhança com um convencional controlador por realimentação de estados com integrador, vamos fazer a simulação desta vez utilizando o Simulink, já que este favorece a análise estrutural pelos diagramas de blocos.

O sistema contínuo a ser simulado, ainda em malha aberta, está representado na Figura 6, enquanto que o diagrama com o controlador LQR discreto conectado ao processo é apresentado na Figura 7. Note que tudo o que não é "planta", obviamente é controlador. Note também que o ganho do controlador, mostrado em (26), foi "desmembrado" e que  $K_{\rm I}$  opera como o ganho da ação incremental do controlador, semelhante a uma estrutura do tipo I+PD.

Na Figura 7 destaca-se também a presença dos blocos ZOH enfatizando a inclusão dos componentes A/D e D/A no controle de um sistema contínuo e que os dados que chegam e saem do controlador digital são de fato amostrados, representados nas figuras 8 e 9.



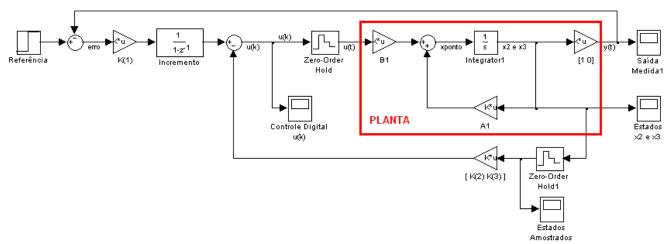
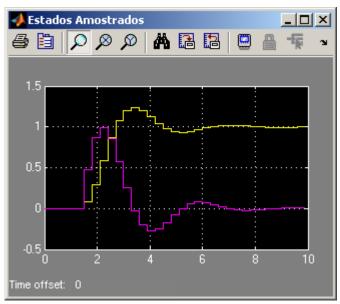


Figura 7



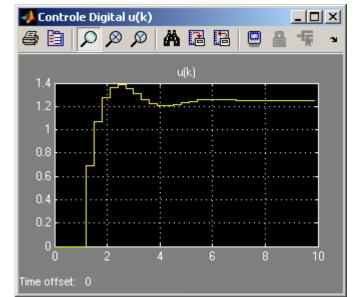


Figura 8 Figura 9

## 5. Bibliografia

- ANDERSON, B. D. O.; MOORE, J. B. Optimal filtering. Englewood Cliffs, New Jersey, USA, Prentice-Hall, Inc, 1979.
- ASTROM, K. J; WITTENMARK, B. Computer-controlled systems: Theory and Design. Mineola, New York, USA: 3rd Ed, Dover Publications, 1997.
- COELHO, A. A. R.; COELHO, L. S. Identificação de Sistemas Dinâmicos Lineares. Florianópolis, SC, Brasil: Editora da Universidade Federal de Santa Catarina, 2004.
- STEVENS, B. L.; LEWIS, F. L. Aircraft control and simulation. Hoboken, New Jersey, USA: 2nd Ed, John Wiley and Sons, Inc, 2003.