



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

PMR 3408 Instrumentação

Pré Relatório para a experiência de *Ball and Beam*

Este pré-relatório tem objetivo de fornecer uma revisão de controle linear, bem como uma introdução à controle moderno e discretização e à modelagem de sistemas dinâmicos para a bancada experimental do laboratório de Instrumentação e valerá 2 pontos da nota total da experiência.

Atenção: Este pré-relatório utilizará o MatLab. O script e o simulink gerados deverão ser levados ao laboratório no dia da experiência.

Nome _____	n°USP _____

Introdução:

Há alguns métodos diferentes para se projetar controladores de sistemas lineares. Dentre os mais comuns na indústria estão o PID e o controle por realimentação de estados.

O PID é o método mais utilizado atualmente. Uma das vantagens do PID é que ele pode ser facilmente implementado com ou sem o conhecimento prévio da planta do sistema a ser controlado, mas necessita de um processo iterativo que pode ser trabalhoso ou até perigoso em alguns casos. Para estes casos é recomendado que se tenha um conhecimento prévio mínimo do modelo para determinar ganhos para um chute inicial, evitando movimentos bruscos ou impossíveis para os atuadores.

Já o método de controle por espaço de estados (controle moderno) vem ganhando espaço e é tradicionalmente utilizado com o modelo do sistema. A grande vantagem deste método é possibilitar o controle de sistemas MIMO. Além disso, este método possibilita uma visualização explícita e no domínio do tempo para cada um dos estados.

Para esta experiência de laboratório iremos implementar um PID sem considerar o modelo e uma lei de controle por realimentação de estados em uma plataforma conhecida como *Ball and Beam* como a representada na figura 1. O objetivo da experiência é posicionar uma bolinha de tênis de mesa em um ponto desejado e calibrar dois tipos de sensores de forma a obter melhores resultados.

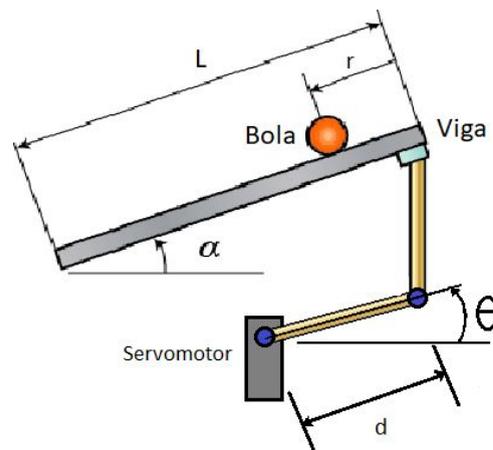


Figura 1: Esquemático do Ball and Beam

As tarefas de calibração serão realizadas durante o laboratório, entretanto para otimizar o tempo restrito das aulas, optamos por realizar o projeto do controlador em um pré-relatório.

Parte I: Resumo de controle linear

Tanto o método de PID quanto o controle por espaço de estados utilizam lógicas parecidas. Para fazer um projeto de PID sem modelo, será necessária a utilização de um sistema possuindo um feedback direto do estado que se deseja controlar. Assim, será obtido um erro e que deseja-se minimizar.

Inicialmente vamos avaliar a influência de cada parcela do PID, e o fenômeno físico correspondente. Para tanto, utilizaremos o método sem considerar o modelo do sistema.

O controlador proporcional visa zerar o erro (e) entre o efetuator e o setpoint, isto é, a plataforma irá atuar proporcionalmente entre o ângulo do motor e o erro de posição da bolinha fazendo ela se aproximar do *setpoint* determinado. Entretanto, como o atrito entre a

bolinha e a plataforma são muito pequenos, a inércia da bolinha fará com que ela continue o movimento aumentando novamente o erro e o resultado deverá ser similar ao movimento de gangorra. A expressão correspondente para o controle proporcional é portanto:

$$P = K_p * e$$

O controlador derivativo visa desacelerar a bolinha até que sua velocidade seja nula. Portanto pode-se fazer uma analogia à criação artificial de um atrito no sistema. Dessa forma, um controlador PD seria suficiente para alcançar e manter o efetuator um determinado setpoint. Com isso, o controlador fica:

$$PD = K_p * e + K_d * \frac{e - e_{ant}}{T_s}$$

Entretanto alguns erros em regime podem surgir decorrentes do sistema ou de dinâmicas externas e pelo esforço do controlador apresentar valores muito pequenos. Para garantir que o sistema atinja realmente o setpoint desejado, é introduzido um integrador que atua somando o erro ao longo do tempo. Assim, após determinado tempo o esforço do controlador é aumentado mesmo que o controlador proporcional e o derivativo sejam pouco expressivos, até que o sistema volte a atuar. A expressão do integrador (v) e do controlador PID são apresentados a seguir:

$$v = v + e$$

$$PID = K_p * e + K_d * \frac{e - e_{ant}}{T_s} + K_i * v$$

Agora que entendemos o funcionamento do PID, vamos relembrar como determinar os ganhos em um sistema com modelo dinâmico conhecido.

Cada sistema possui uma ou mais dinâmicas próprias, cada uma dessas dinâmicas podem ser representadas em uma função de transferência $G(s)$ na forma de pólos (denominador) e zeros (numerador).

$$G(s) = \frac{Y(s)}{X(s)} = \frac{(s-a_1)(s-a_2)...(s-a_n)}{(s-b_1)(s-b_2)...(s-b_k)}$$

Entretanto algumas dessas dinâmicas podem ser desprezadas, uma vez que elas ocorrem e se encerram muito mais rápido que as demais.

Ao se fechar uma malha de controle utilizando um feedback, como no exemplo da figura 2, a função de transferência correspondente do sistema deixa de possuir as dinâmicas de $G(s)$ e passa a possuir de dinâmica de $G_{mf}(s)$

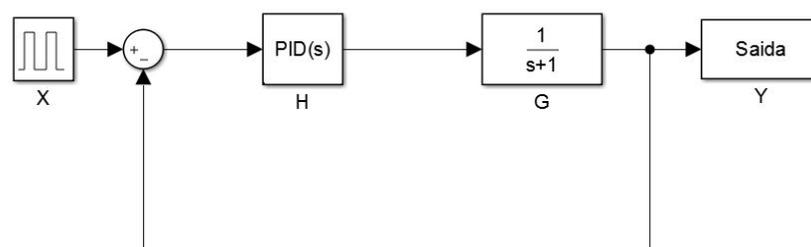


Figura 2: Diagrama de Blocos de um sistema contínuo genérico

$$G_{mf}(s) = \frac{G(s)H(s)}{1+G(s)H(s)}$$

Note que agora os pólos podem ser regulados variando os ganhos do controlador $H(s)$, e portanto a dinâmica do sistema também varia.

Uma forma de determinar os ganhos do sistema é alocando os pólos de um controlador arbitrário para que ele corresponda a alguns parâmetros desejados. Para alocar os pólos utiliza-se o método do lugar das raízes.

Vale lembrar que, pelo lugar das raízes, os pólos de primeira ordem podem ser descritos por:

$$s_n = -\zeta * \omega_n$$

Já um par de pólos de segunda ordem podem ser descritos por

$$s_n = -\zeta * \omega_n + \omega_d i$$

Sendo a constante de amortecimento ζ calculada para sistemas subamortecidos por:

$$\zeta = \sqrt{\frac{\ln(Mp)^2}{\pi^2 + \ln(Mp)^2}}$$

Para sistemas não amortecidos $\zeta = 0$, e para sistemas criticamente amortecidos $\zeta = 1$. A frequência natural ω_n pode ser calculada com base no tempo de estabilização t_s em 98% e a frequência amortecida ω_d é calculada em sistemas subamortecidos por:

$$\omega_n = \frac{4}{\zeta * t_s}$$
$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

Após determinar os pólos a serem alocados, basta resolver o sistema forçando os pólos de $G_{m.f}$ a serem iguais aos pólos determinados, obtendo os ganhos do controlador.

Parte II: Resumo de controle digital

Essa experiência será realizada utilizando um arduino, a frequência de processamento deste microcontrolador não é muito alta, além disso, há limitações de frequência de sensores e outros recursos que serão utilizados. Esses fatos impedem que um controle contínuo possa ser utilizado corretamente.

Para resolver o problema, será necessário fazer a discretização do controlador, isto é, cada ciclo do loop de controle deverá ocorrer em um tempo pré-determinado. Este tempo pré-determinado será chamado de tempo de amostragem (T_s). Perceba que, caso o sistema seja muito mais lento que o loop de controle, o controle passa a atuar de forma praticamente contínua, obtendo-se um controle mais suave dos movimentos.

A malha fechada em controle digital de um sistema mecânico consiste em amostrar os sinais contínuos transformando-os em sinais discretos para leitura e processamento no microcontrolador e em seguida é feita a reconstrução do sinal de saída do microcontrolador para atuar o sistema mecânico. Um exemplo está representado na figura 3:

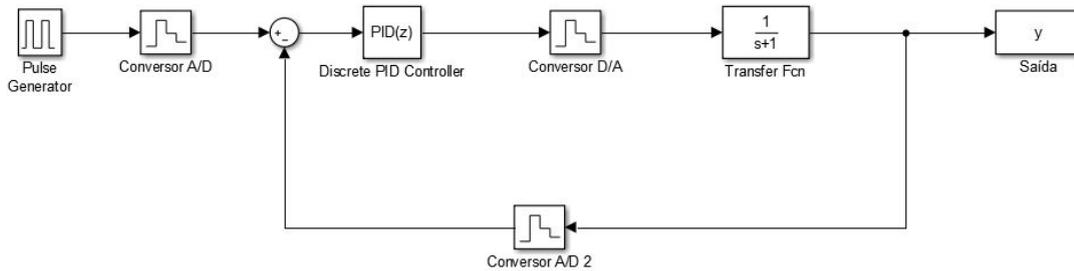


Figura 3: Diagrama de Blocos de um sistema Real Genérico

A amostragem pode ser representada do ponto de vista matemático pela multiplicação do sinal por um trem de impulsos. Já a reconstrução pode ser feita pelo método do reconstrutor (ou segurador) de ordem zero (ZOH) que consiste em manter a amplitude do sinal amostrado até a amostragem seguinte, como pode ser visto nas figuras 4 e 5. As expressões matemáticas não serão abordadas nesta experiência, já que o foco da experiência são os fenômenos físicos.

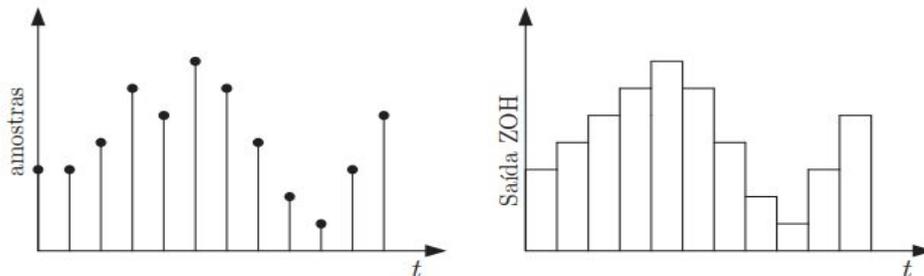


Figure 4: Gráficos de amostragem e reconstrução por ZOH

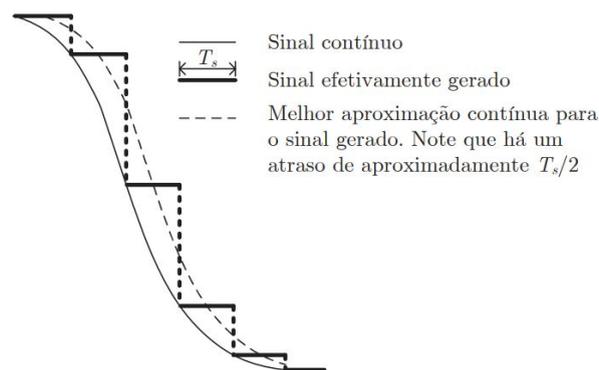


Figure 5: Gráfico de reconstrução por ZOH

Existem diversas formas de fazer a digitalização dos sistemas que poderiam inclusive fornecer resultados melhores em alguns casos, mas o ZOH será suficiente para alcançarmos os objetivos dessa experiência. No MATLAB o comando `c2d` faz a conversão de sistema contínuo para discreto.

Agora que já sabemos como discretizar a planta de um sistema, resta saber como discretizar os pólos e zeros. O método mais comum utilizado é o de mapeamento casado de pólos e zeros, dado pela expressão:

$$z = e^{sT_s}$$

Esta expressão é resultado da equivalência entre o lugar das raízes determinada por laplace e o lugar das raízes determinado pela transformada-z.

Parte III: Controle por realimentação de estados

O método de controle por realimentação de estados não foge muito ao que já foi visto anteriormente. Um sistema dinâmico pode ser representado em variáveis de estado na forma:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Já o sistema digitalizado terá a forma:

$$\begin{aligned}x[n+1] &= \Phi x[n] + \Gamma u[n] \\ y[n] &= Cx[n] + Du[n]\end{aligned}$$

A saída y deve representar apenas as saídas de interesse, dessa forma a matriz C possui dimensão n x m onde n é o número de variáveis de interesse e m o número de estados do sistema

Perceba que, embora a representação matemática do sistema tenha sido digitalizada, o sistema físico ainda é contínuo. Portanto, a representação mais real do sistema controlado é similar ao representado na figura 3. A forma digitalizada será utilizada apenas para determinação da lei de controle por meio do método do lugar das raízes.

Diferentemente dos métodos já vistos anteriormente, ao invés de realimentarmos a saída, iremos realimentar cada um dos estados. Se o sistema é controlável, podemos determinar uma matriz de ganhos K em que a lei de controle $u = -Kx$ possa alocar os polos do sistema em malha fechada. Assim, uma expressão de seguimento de trajetória para a lei de controle é:

$$u = -k_1 * x_1 - k_2 * x_2 + r$$

Enquanto no controlador PID bastava que conhecêssemos o erro em relação ao setpoint, neste método será necessário conhecer todos os demais estados. Entretanto o acesso à medição de estados pode ser impraticável ou muito precária, necessitando do projeto de observadores ou estimadores, o que está fora do escopo da disciplina.

Embora esta experiência não possua uma medição direta da velocidade angular do motor, a aproximação de derivada da posição apresenta resultados bastante satisfatórios, permitindo a utilização do método de realimentação de estados.

De forma análoga ao PID descrito na parte I, o sistema possui um ganho atuando na parcela proporcional ao primeiro estado e um ganho atuando na parcela derivativa representada pelo segundo estado. Entretanto ainda falta adicionar um integrador ao sistema para garantir que o erro em regime vá a zero.

A expressão do sistema em malha fechada com integrador é conhecida como sistema aumentado. O procedimento consiste em acrescentar um novo estado e um feedback do estado a ser integrado, como o estado a ser controlado é representado pela matriz de saídas C o sistema com integrador pode ser escrito como:

$$\Phi_A = \begin{bmatrix} \Phi & 0 \\ -C & I \end{bmatrix}$$

$$\Gamma_A = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$

Parte IV: Modelagem do sistema

A modelagem dinâmica para a bola foi realizada através do método de lagrange, obtendo-se a seguinte equação:

$$\left(\frac{J}{R^2} + m\right) \ddot{r} + mg \sin \alpha - mr \dot{\alpha}^2 = 0$$

Como se pode notar, há duas não linearidades na equação dinâmica. Para esta experiência, serão aplicados métodos de controle linear, como aquelas vistas nas disciplinas de controle 1 e controle 2, portanto precisaremos linearizar a equação.

Para pequenos ângulos de α , pode-se linearizar a equação em torno de $\alpha = 0$. A equação linearizada e a relação entre os ângulos são:

$$\left(\frac{J}{R^2} + m\right) \ddot{r} + mg \alpha = 0$$

$$\alpha = \frac{d}{L} \theta$$

Com isso, obtemos a equação dinâmica linearizada:

$$\left(\frac{J}{R^2} + m\right) \ddot{r} + mg \frac{d}{L} \theta = 0$$

Exercício 1: Transforme a equação anterior para a forma de espaço de estados e enumere os estados que precisam ser medidos.

Parte IV: Controle digital

Como a experiência será implementada em um microcontrolador de baixo custo (arduino), as frequências de operação são muito baixas, portanto é necessário realizar a digitalização da equação dinâmica. Os parâmetros do sistema são dados na tabela abaixo:

Parâmetro	Descrição	Valores
m	Massa da Bola	0.0274 (Kg)
R	Raio da Bola	0.015 (m)
d	Comprimento da Alavanca	0.05 (m)
g	Gravidade	-9.8 (m/s)
L	Comprimento da viga	0.3 (m)
J_b	Momento de inércia da bola	$\frac{2}{5} mR^2$ (kgm^2)
r	Coordenada da bola	-
α	Ângulo da viga	-
θ	Ângulo do motor	-

Exercício 2: Implemente no MatLab a equação do espaço de estados e faça a discretização utilizando o método de reconstrutor de ordem zero (zoh) com um período de amostragem de T_s . Para ganharmos tempo, utilize o script a seguir:

```
clear all; close all; clc;
t_run = 40;
m = 0.00274; R = 0.015; g = -9.8; L = 0.3; d = 0.05; J = 2*m*R^2/3;
fs = 50; Ts = 1/fs;

%% Faça aqui os exercícios
```

Exercício 3: Para um sobressinal de até 20% e tempo de estabilização de 4s, calcule os valores da constante de amortecimento, frequência natural e frequência natural amortecida. A partir destes valores, determine os pólos em s e transforme para z utilizando o método de mapeamento de pólos e zeros. **Obs:** os números imaginários no MATLAB são representados por 1i

Parte V: Simulink

Abra o esquema do Simulink fornecido no moodle. Um bloco de MATLAB function foi adicionado no lugar do controlador e será programado com a mesma equação que será utilizada no arduino, basta abrir o bloco. Verifique também os outros blocos e nomes de variáveis utilizadas.

Para que o sistema fique mais real, foi adicionada uma saturação (calculada em radianos). Essa saturação representa uma limitação do motor e das simplificações adotadas, como a linearização.

O simulink irá se comunicar com o script que será criado. Adicione ao final do seu script o seguinte código:

```
%% Simulação
sim('Diagrama_de_Blocos');
n = 0:Ts:Ts*(length(u)-1);

figure('units','normalized','outerposition',[0 0 1 1]);
title('Output');grid on; hold on;
subplot(121); grid on; hold on;
plot(tout,r,'-k'); plot(tout,y,'-b'); hold off;
legend('Reference Signal', 'Output Signal', 'Location','NorthWest');
xlabel('Time [s]'); ylabel('Amplitude');
subplot(122); grid on; hold on;
stairs(n,u,'-b');
title('Control Signal'); xlabel('Time [s]'); ylabel('Amplitude')

fprintf('Alocação de polos\n k1 = %f;\n k2 = %f;\n', [K(1), K(2)])
```

Exercício 4: Faça a alocação dos pólos (utilize o place(Phi, Gamma, Ts) do MATLAB) e determine os ganhos para a realimentação de estados. Simule utilizando o simulink com uma saturação de 10° e plote a saída e o esforço do controlador.

Obs: Note que a matriz C teórica da saída possui dimensão 1x2 e é diferente da matriz C do Simulink de dimensão 2x2. Isso ocorre pois é necessário fazer a realimentação de todos os estados, mas nem todos os estados serão controlados.

Obs2: A função place funciona apenas se todos os polos forem diferentes, caso queiram alocar os polos com multiplicidade maior que 1 utilizem a função acker.

Parte VI: Adição de integrador

Exercício 5: Visto que o sistema possui erro em regime não nulo, refaça a simulação utilizando uma nova equação para o espaço de estados, mas adicionando um estado de integração. **Note que agora há um estado a mais, portanto é necessário alocar um polo extra.** O polo do integrador deverá agir de forma mais rápida do que a dinâmica mais lenta do sistema, mas também deve ser lento o suficiente para ter uma dinâmica relevante. O Ogata determina que as dinâmicas começam a ser menos relevantes quando são de 5 a 8 vezes mais rápidas do que a dinâmica mais lenta do sistema

Faça a alocação do integrador com 3 valores diferentes:

- a) Mesma velocidade do polo mais lento que os polos alocados anteriormente.
- b) 2x mais rápido que o polo mais lento
- c) 5x mais rápido que o polo mais lento

Simule utilizando o simulink. O sistema continua estável para os 3 casos anteriores? Houve degradação na dinâmica do sistema para algum deles? Proponha o motivo da diferença entre os resultados e da instabilidade, caso haja.

Obs: O integrador é gravado na memória e não é realimentado. Utilizem da mesma forma que o descrito no PID com a variável v. Para tanto, no MATLAB fcn acrescentem uma variável global v.