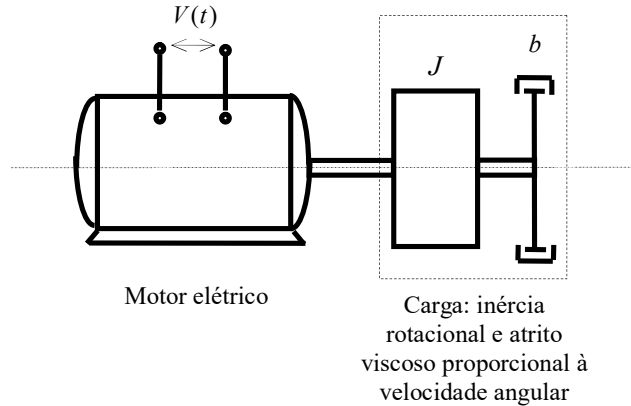


### Sistema a ser controlado

No motor elétrico, a conversão eletromecânica de energia obedece à seguinte relação:  $T = Ki$ , onde  $T$  é torque e  $i$  é a corrente elétrica no enrolamento do motor. O enrolamento do motor possui resistência elétrica  $R$  e indutância  $L$ . A inércia total (eixo do motor e carga) é  $J$ , e há atrito viscoso (coeficiente  $b$ ). Considere como saída a velocidade angular da carga.

$V(t)$  : entrada  
 $\omega$  : saída  
 $J = 0,01 \text{ kgm}^2$   
 $B = 0,1 \text{ Nms}$   
 $K = 0,01 \text{ Nm/A}$   
 $L = 0,5 \text{ H}$   
 $R = 1 \Omega$



Modelo dinâmico no espaço de estados:

$$\begin{cases} \frac{d}{dt}\omega = -\frac{b}{J}\omega + \frac{K}{J}i \\ \frac{d}{dt}i = -\frac{K}{L}\omega - \frac{R}{L}i + \frac{1}{L}V(t) \end{cases}$$

**Tarefa 0** – não acrescenta nota pois o programa está apresentado aqui neste arquivo, mas a não realização da Tarefa 0 acarreta a diminuição da nota da Tarefa 1 (obs.: copiar e colar pode não funcionar no Scilab, ou no Arduino):

Em tempo contínuo, e usando o Scilab, aplique e simule um compensador PID com os seguintes ganhos (é permitido usar o comando “*csim*” e similares):

$$\frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} + K_D s$$

$$K_p = 100 \quad K_I = 200 \quad K_D = 10$$

Tempo de simulação: 10 s.

### Tarefa 1:

Obtenha uma aproximação em tempo discreto do compensador PID mostrado na Tarefa 0, usando transformada Z e o método do trapézio (à mão), e, usando o Scilab, aplique no motor e simule calculando diretamente pelas equações de diferenças (**sem usar o comando “*flts*”** ou similares). A aproximação do motor usando o método do segurador de ordem zero (comando “*dscr*” no Scilab) já está apresentada neste arquivo. Use os seguintes períodos de amostragem:  $T=0,25 \text{ s}$ ;  $T=0,1 \text{ s}$ ;  $T=0,05 \text{ s}$ . Compare as respostas do sistema com esses períodos de amostragem entre si e com a obtida na simulação do sistema contínuo (Tarefa 0).

### Tarefa 2:

Repita a Tarefa 1 usando a regra “para trás” (“backward rule”).

Obs.: não é preciso obter a aproximação em tempo discreto do compensador PID usando a regra “para trás”, pode ser usado o resultado já mostrado na página 4 da apostila “PME3402\_TOPICO\_06\_PID\_DIGITAL\_2020.pdf”.

**O relatório deverá conter o registro das tarefas acima e ser efetuado no próprio código do Scinotes na forma de texto comentado.**

Os valores numéricos dos parâmetros foram retirados da seguinte fonte:

<http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=ControlPID>

```

// Use este programa como ponto de partida.
// A Tarefa 0 já está pronta, e a Tarefa 1 iniciada.

// Obs.: esse programa foi escrito para o Scilab 5.5.2.
// Adaptações podem ser necessárias se este programa for usado no Scilab 6.1.

// Importante: os gráficos devem estar em sua própria janela, evitando
// que a próxima figura se sobreponha à figura anterior, exceto se for esse
// o objetivo, no caso de se querer fazer comparações dos resultados.

// Tarefa 0
clear;clf;
// Parâmetros do motor de corrente contínua:
J=0.01;
b=0.1;
K=0.01;
L=0.5;
R=1;

// Modelo no espaço de estados:
A=[-b/J K/J; -K/L -R/L];
B=[0;1/L];
C=[1 0];
D=0;
motor=syslin('c',A,B,C,D);

// Função de transferência do motor:
Gmotor=ss2tf(motor);

// Compensador PID
KP=100;
KI=200;
KD=10;
s=poly(0,'s');
Gpid=syslin('c',KP+(KI/s)+KD*s);

// Conexão em série do compensador PID e do motor (malha aberta):
Gma=Gpid*Gmotor;

// Fechamento da malha (feedback), com feedback unitário:
Gfb=syslin('c',s/s); // feedback unitário

// Fechando a malha (Gmf é a função de transferência de malha fechada):
Gmf=Gma/.Gfb;

// Simulação para entrada degrau unitário:
DT=0.001;
Tf=10;
t=0:DT:(Tf-DT);
u=ones(t);
x0=[0;0;0];
y=csim(u,t,Gmf,x0);

plot2d(t,y);
xtitle('Saida controlada por PID: tempo contínuo - linha preta','t (s)','y (rad/s)');

```

```

// Tarefa 1
// Modelo em tempo discreto do motor de corrente contínua usando o
// segurador de ordem zero (ZOH):
T=0.25 // Período de amostragem

//dscr obtém o modelo em tempo discreto de uma planta no espaço de estado
// usando o ZOH.
motorD=dscr(motor,T);

// função de transferência do motor em tempo discreto (ZOH):
GmotorD=ss2tf(motorD);

// Simulando o sistema com compensador PID usando as equações de diferenças:

// Equações de diferenças para o modelo em tempo discreto do motor de corrente
// contínua:
nMD=coeff(numer(GmotorD));
dMD=coeff(denom(GmotorD));
n=length(nMD);
d=length(dMD);
if d>n then
p=d;
else
p=n;
end

// Condições iniciais - motor
for i=1:(p-1)
um(i)=0;
ym(i)=0;
e(i)=0;
end

// Equações de diferenças:

// O restante do programa deve ser desenvolvido pelo grupo.
// É preciso escrever as equações de diferenças do motor e desenvolver
// e escrever as equações de diferenças do PID.

// IMPORTANTE: nas Tarefas 1 e 2 as simulações devem ser feitas por meio de
// equações de diferenças, não podem ser usadas funções "prontas" do Scilab para
// simulação de sistemas discretos.

```