

Laboratório de controle

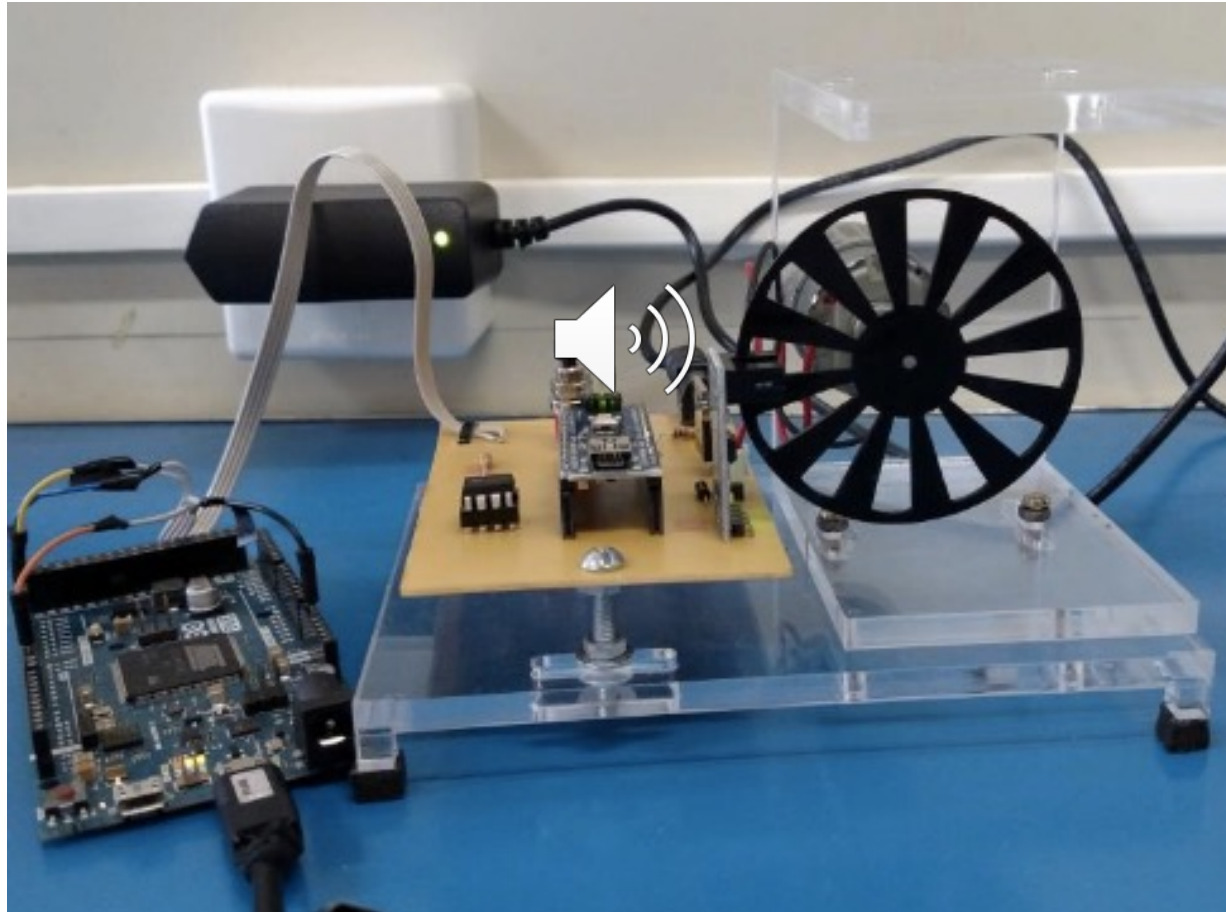
Aula 7

Modelagem de um motor de corrente continua
e sintonia de controladores PID

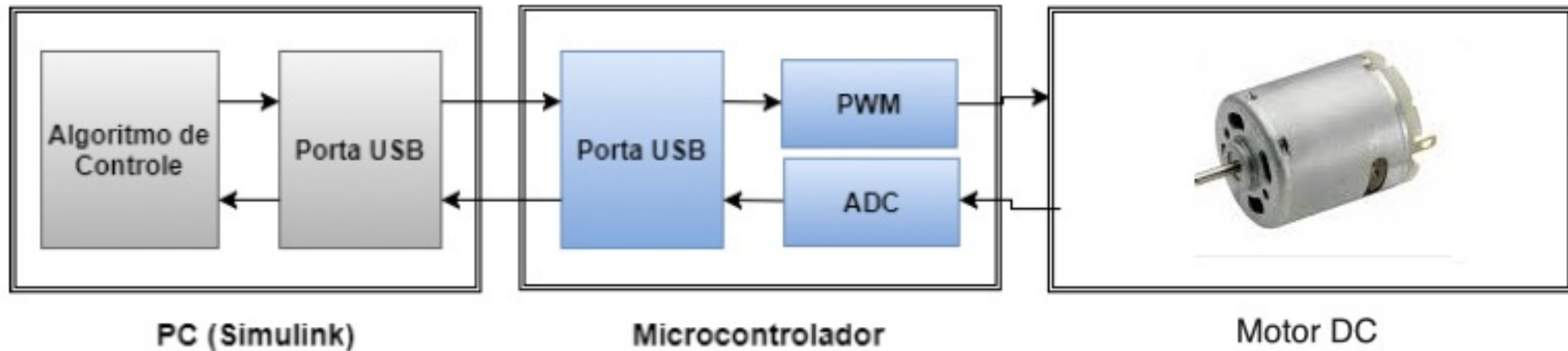
Objetivos

- Estudar o motor de corrente contínua e seus componentes
- Obter modelos para o motor
- Sintonizar controladores por métodos simplificados

Motor de CC utilizado



Sistema de controle digital usado com o motor CC



O modelo é obtido aplicando sinais adequados

- Sinal de rampa
- Sinal degrau

Resposta a uma entrada PWM tipo rampa

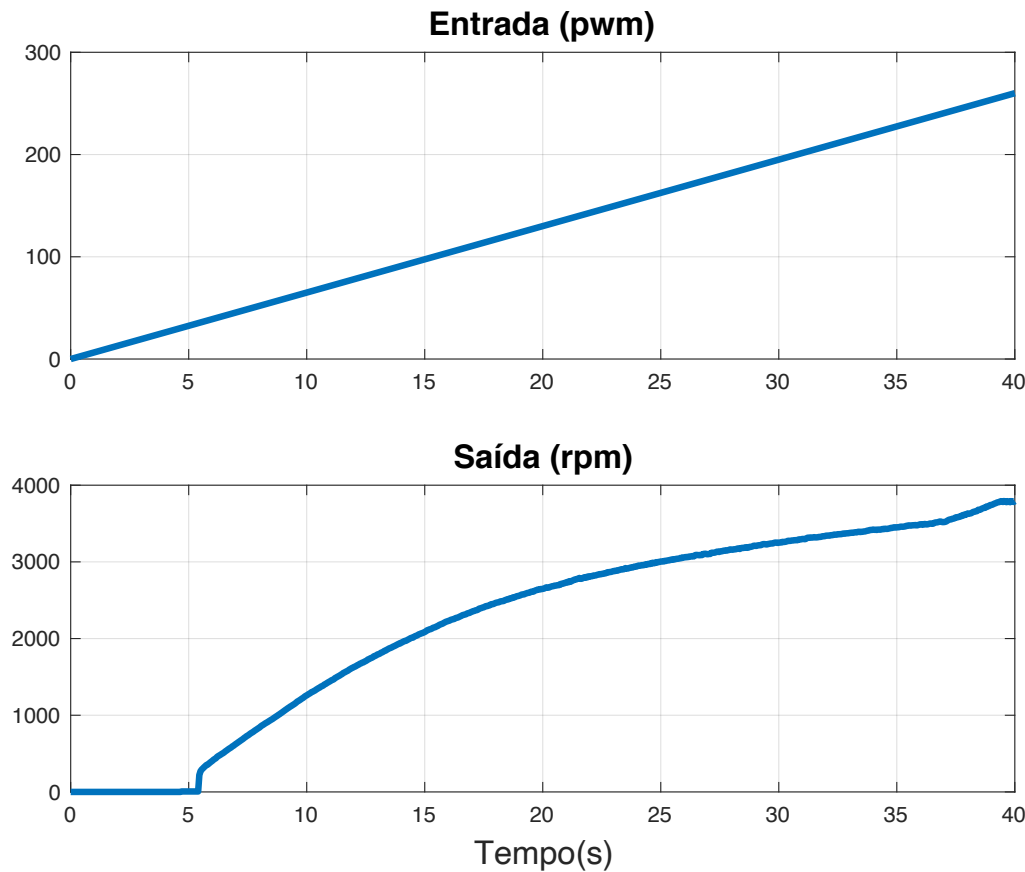


Fig1

Resposta a uma entrada PWM tipo rampa

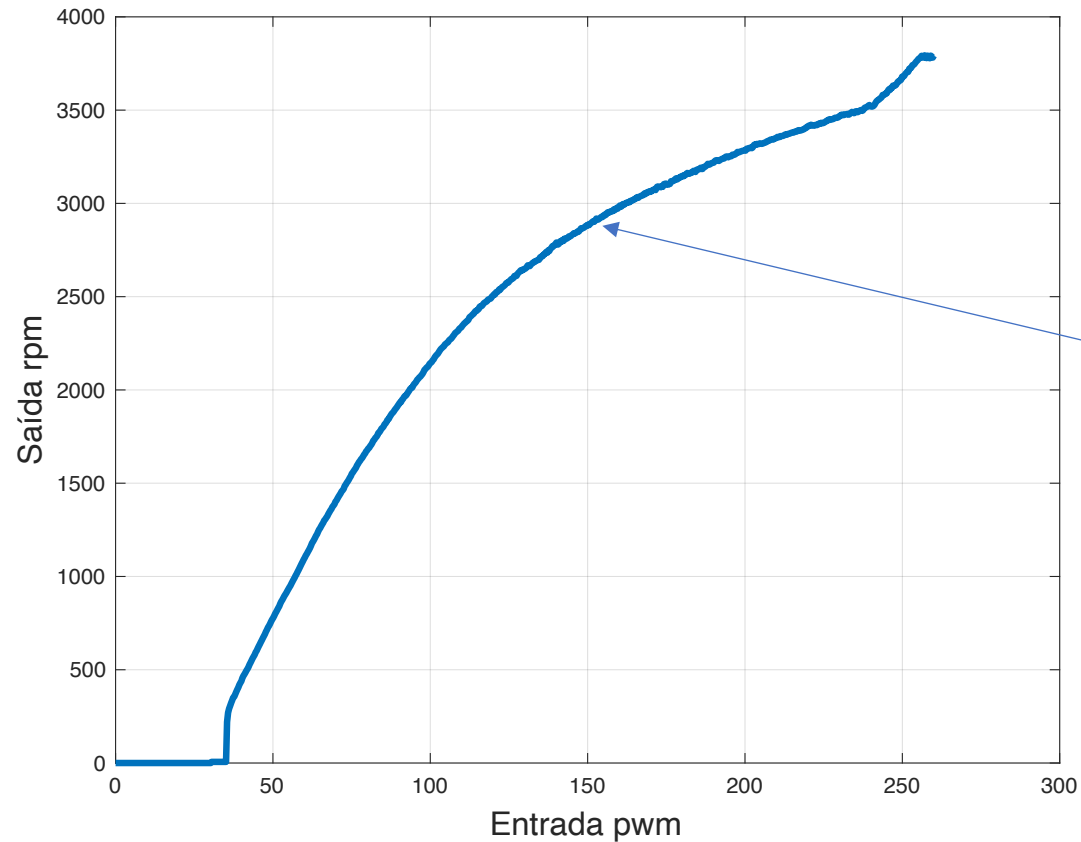


Fig2

Nesta figura se observa que a entrada PWM for 150, a rotação será aproximadamente 3000RPM.

Resposta a uma sequência de degraus

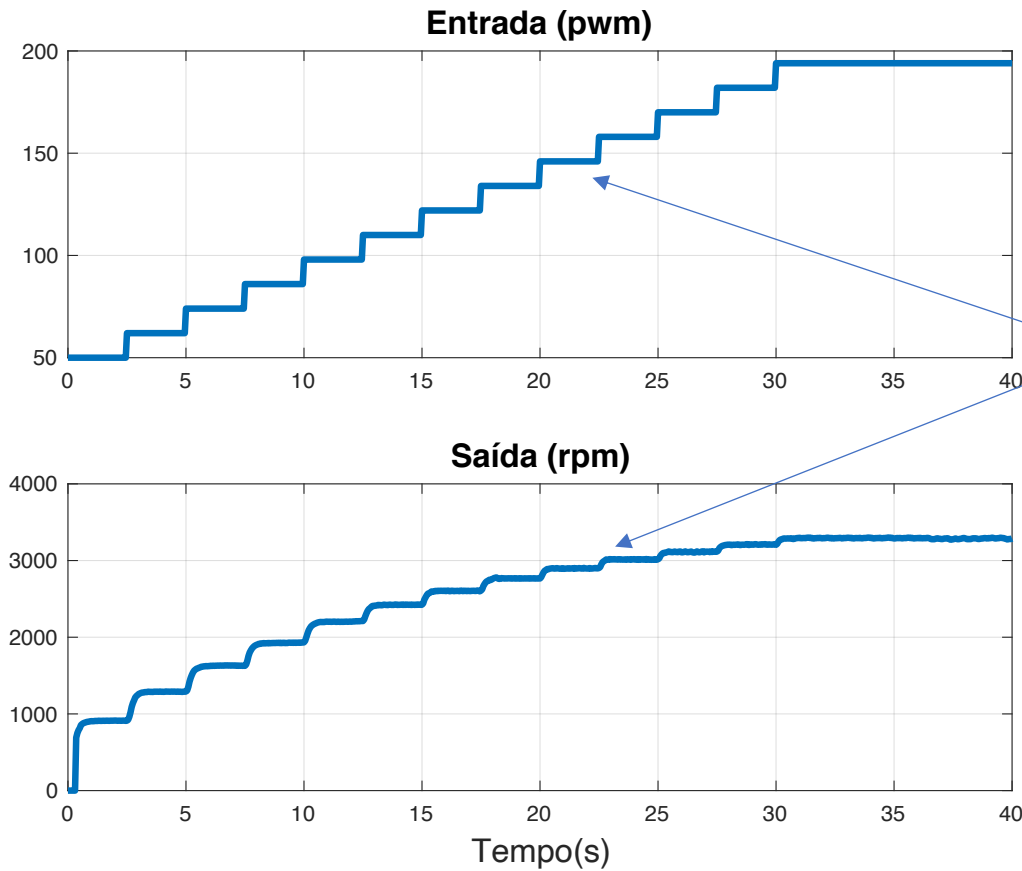


Fig3

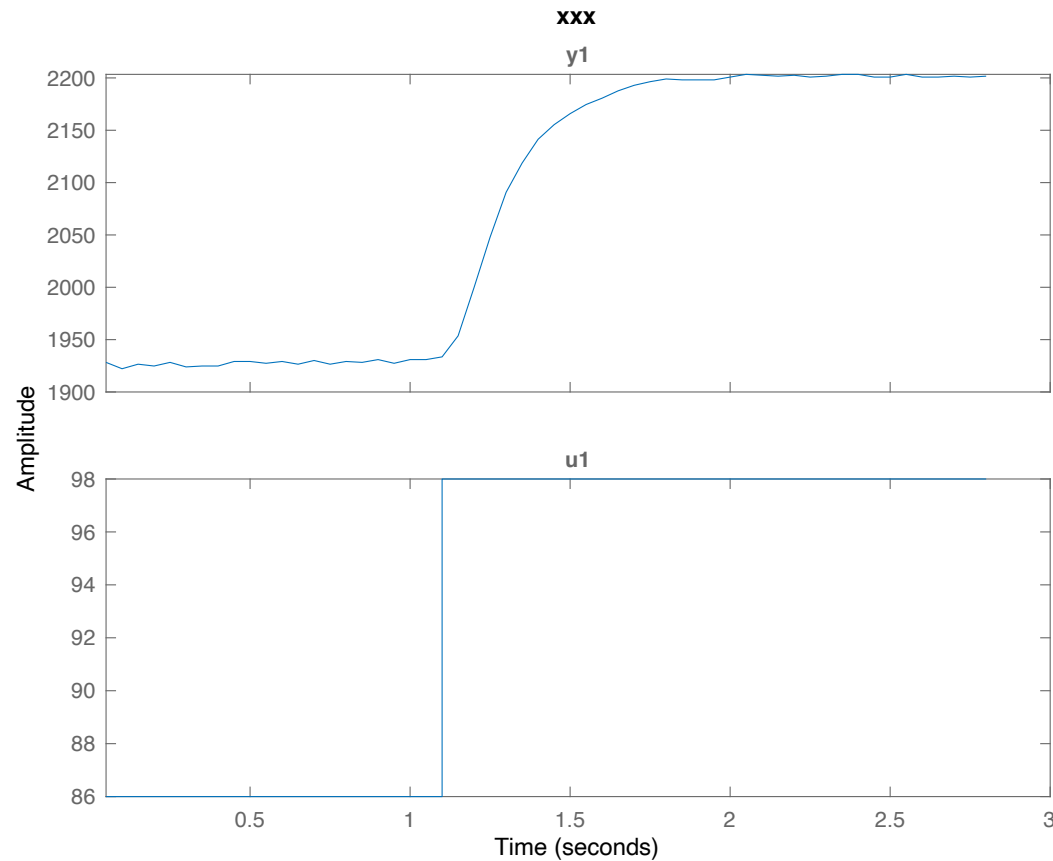
Aqui também se observa que 150RPM na entrada fornecem 3000 rpm de velocidade

Modelo do motor CC

O modelo representa a relação entre o sinal PWM aplicado e a velocidade medida em RPM.

Observa-se que o comportamento depende da rotação. Deve-se obter um modelo que represente adequadamente a região na qual o motor vai operar.

Modelo do motor CC

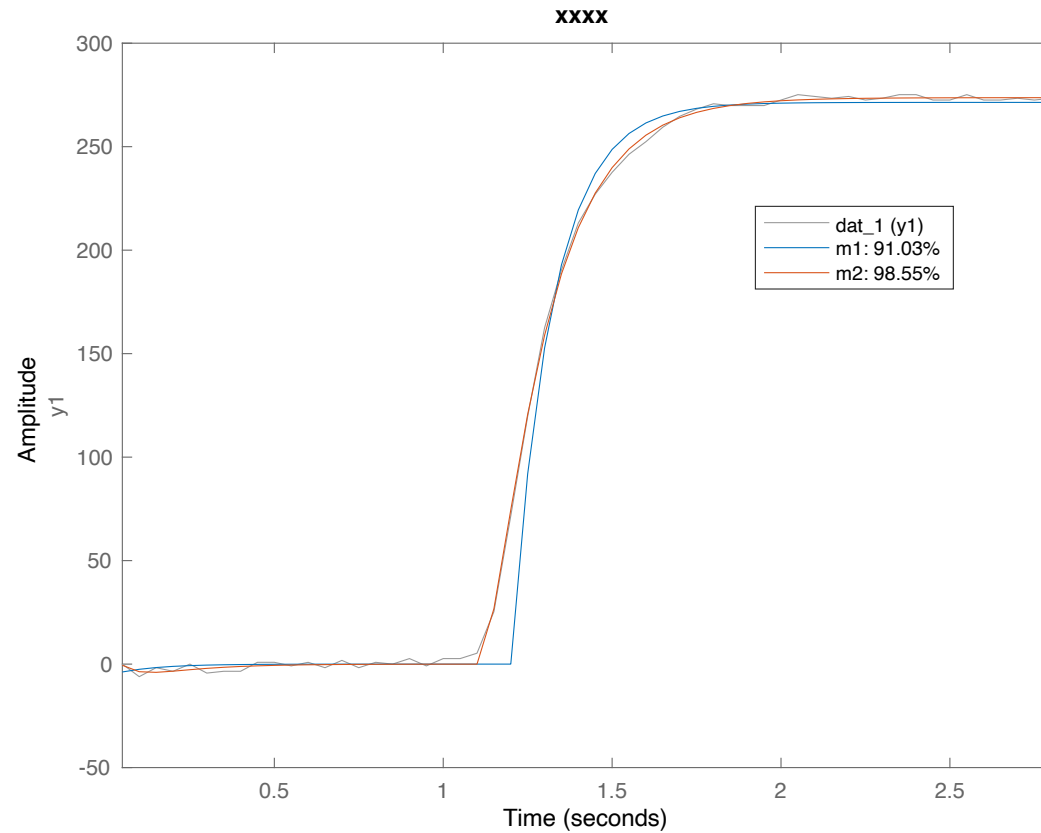


Se o motor for operar próximo a 2000 rpm, o degrau ao lado deve ser aplicado.

Um degrau de amplitude 12 é aplicado quando o motor está em 1925rpm, que faz a rotação ir para 2200 rpm.

Fig4

Modelo do motor CC



Um modelo de primeira ordem e outro de segunda ordem são estimados

Subtrai-se o valor inicial dos vetores de dados para facilitar estimar os parâmetros do modelo

Fig5

Especificação da resposta no tempo

Pode ser obtida a partir da resposta em malha aberta e considerando os objetivos do controlador, que neste caso é fazer com que a velocidade siga uma referência em degrau.

- $UP=0$; Sobrelevação
- $t_s=0$; Tempo de estabelecimento
- $t_r=0$; Tempo de subida
- $Er=0$; Erro em regime

A partir da sobrelevação UP , pode-se obter o amortecimento zeta correspondente.

Como $t_s = \frac{4}{\zeta \omega_n}$, pode-se obter ω_n e o protótipo de segunda ordem que fornece a resposta em malha fechada desejada, $M(s) = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}$

Especificação da resposta no tempo

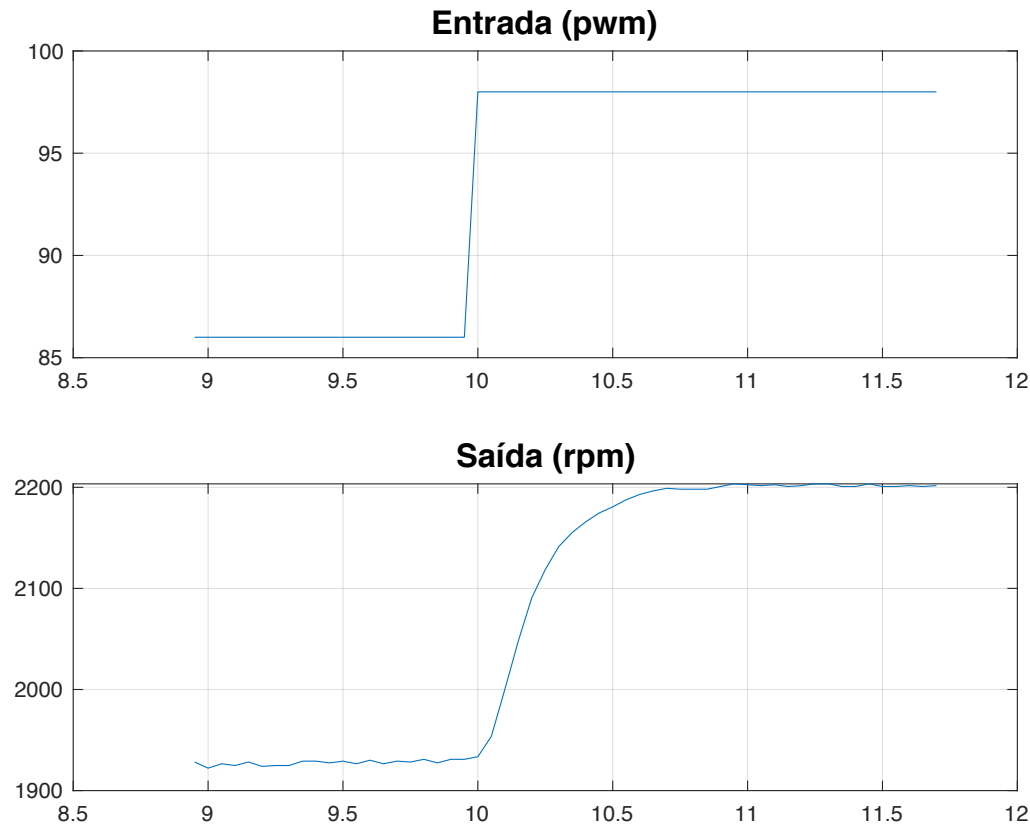
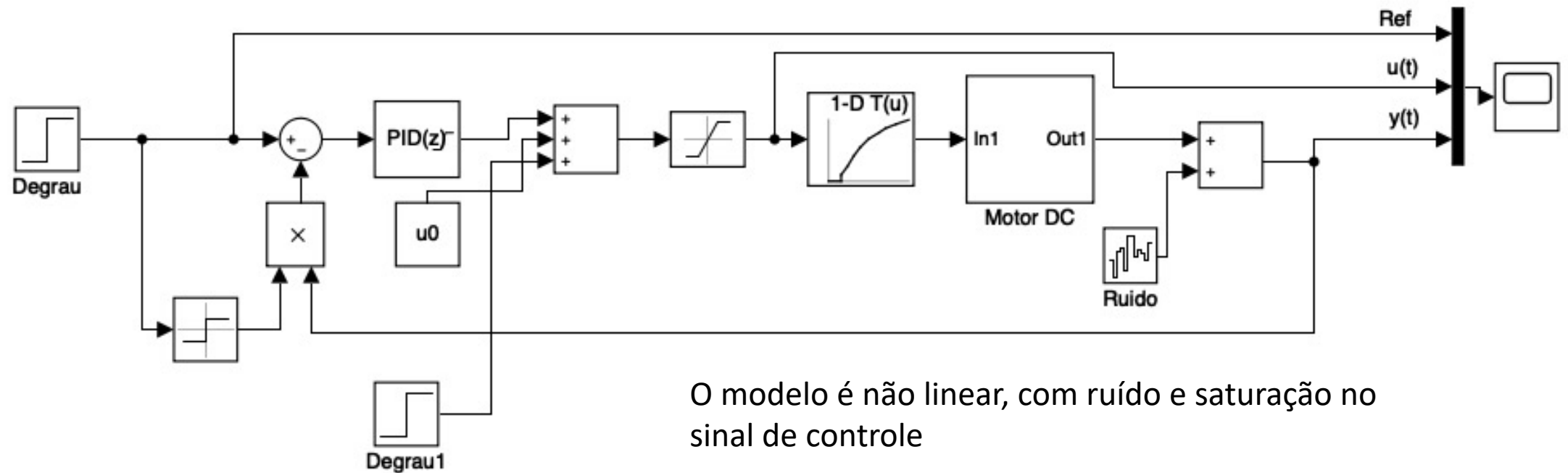


Fig6

Observa-se que a velocidade tende ao valor final em 0.8s, com uma constante de tempo de 0.2s.

Espera-se que em malha fechada seja tão ou mais rápido.

Modelo do Simulink utilizado nos testes



Sintonia de controladores PID

Primeiramente métodos de sintonia serão utilizados

Tabela 7. Sintonia do PID segundo Rivera, Morari e Skogestad para processos com tempo morto

Controlador	K_P	T_I	T_D	Sugestão para o Desempenho
PID	$\frac{2\tau + \theta}{K \times (2\lambda + \theta)}$	$\tau + \left(\frac{\theta}{2}\right)$	$\frac{\tau \times \theta}{(2\tau + \theta)}$	$\frac{\lambda}{\theta} > 0.8$
PI	$\frac{(2\tau + \theta)}{K \times 2\lambda}$	$\tau + \left(\frac{\theta}{2}\right)$	-	$\frac{\lambda}{\theta} > 1.7$

`[C, IAE]=sintonia(g1,'PID','lam',lambda)`

Sintonia de controladores PID

Primeiramente métodos de sintonia serão utilizados

Tabela 6. Sintonia do PID segundo Rivera, Morari e Skogestad

Modelo do Processo	K_P	T_I	T_D
$\frac{K}{\tau^2 s^2 + 2\xi\tau s + 1}$	$\frac{2\xi\tau}{K \times \lambda}$	$2\xi\tau$	$\frac{\tau}{2\xi}$

`[C, IAE]=sintonia(g2,'PID','ordem2', lambda)`

Escolha de λ

λ é a constante de tempo de malha fechada que se deseja!

Na simulação, λ pode ser escolhido tão pequeno quanto se queira, e vai funcionar.

No sistema real não! λ pequeno gera ganhos grandes, que geram sinais de controle de grande amplitude,

Ocorre que a amplitude dos sinais de controle do mundo real são limitados.

No caso do motor CC usado, o sinal de controle é o PWM que satura em 255.

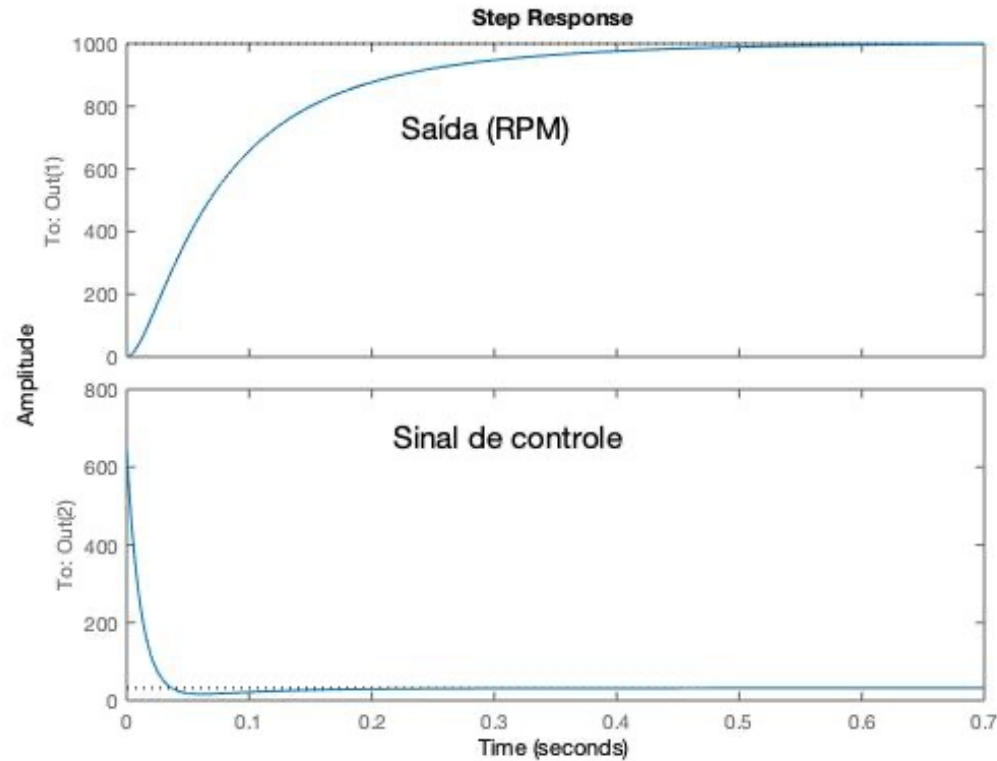
Escolha de lambda

```
c2=sintonia(g2,'PID', 'ordem2',0.1);  
c2.Tf=0.01;  
G=[feedback(c2*g2,1);feedback(c2,g2)];  
step(1000*G);
```

Estas linhas de código geram a resposta em malha fechada mostrada.

Observa-se que o sinal de controle chega quase em 800 na simulação.

No sistema real, não passa de 255!



Qual dos controladores é o melhor no método lambda?

Os dois modelos g_1 e g_2 serão usados para o projeto do controlador.

O comando `C=sintonia(g1,'PI','lam',lambda);` gera um controlador PI a partir de g_1 .

O comando `C=sintonia(g1,'PID','lam',lambda);` gera um controlador PID a partir de g_1 .

O comando `C=sintonia(g1,[],'ordem2',lambda);` gera um controlador PID a partir de g_2 (O PI não é possível neste caso).

Diferentes critérios podem ser usados para escolher o melhor controlador, sendo que em todos os casos ele será aplicado no motor CC.