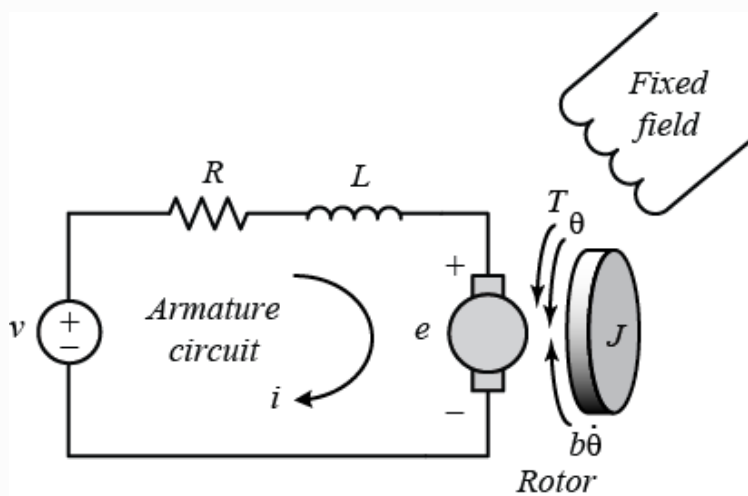


# Modelagem de Motor CC

Ref.: DC Motor Speed: Simulink Modeling, Control Tutorials for Matlab & Simulink, URL: <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&ion=SimulinkModeling> (acessado em 30/09/2020).

## Modelagem

Um atuador comum em sistemas de controle é o motor CC. Ele fornece movimento rotativo diretamente e, juntamente com rodas ou tambores e cabos, pode fornecer movimento translacional. O circuito elétrico da armadura e o diagrama de corpo livre do rotor são mostrados na figura a seguir:



Para este exemplo, iremos supor que a entrada do sistema é a fonte de tensão ( $V$ ) aplicada à armadura do motor, enquanto a saída é a velocidade de rotação do eixo  $\dot{\theta}$  (ou velocidade angular,  $\omega$ ). O rotor e o eixo são considerados rígidos. Assumimos ainda um modelo de atrito viscoso, ou seja, o torque de atrito é proporcional à velocidade angular do eixo.

Os parâmetros físicos para nosso exemplo são:

Parâmetro	Descrição	Valor (Unidades)
$J$	Momento de inércia d motor	0,01 ( $Kg \cdot m^2$ )
$b$	Constante de atrito viscoso do motor	0,1 ( $N \cdot m \cdot s$ )
$K_e$	Constante de força eletromititia	0,01 ( $V/rad/s$ )
$K_t$	Constante de torque do motor	0,01 ( $N \cdot m/A$ )
$R$	Resistência elétrica	1 ( $\Omega$ )
$L$	Indutância elétrica	0,5 ( $H$ )

Em geral, o torque gerado por um motor C é proporcional à corrente da armadura e à força do campo magnético. Neste exemplo, vamos assumir que o campo magnético é constante e, portanto, que o torque do motor é proporcional apenas à corrente da armadura  $i$  por um fator constante  $K_t$  conforme mostrado na equação abaixo. Isso é conhecido como motor controlado por armadura:

$$T = K_T \cdot i$$

A força contra-eletromotriz (f.e.m.),  $e$ , é proporcional à velocidade angular do eixo por um fator constante  $K_e$ :

$$e = K_e \cdot \dot{\theta}$$

Em unidades SI, o torque do motor e a constante f.e.m. são iguais, ou seja,  $K_t = K_e$ ; portanto, usaremos  $K$  para representar a constante de torque do motor e a constante de fem posterior.

This system will be modeled by summing the torques acting on the rotor inertia and integrating the acceleration to give velocity. Also, Kirchoff's laws will be applied to the armature circuit. First, we will model the integrals of the rotational acceleration and of the rate of change of the armature current.

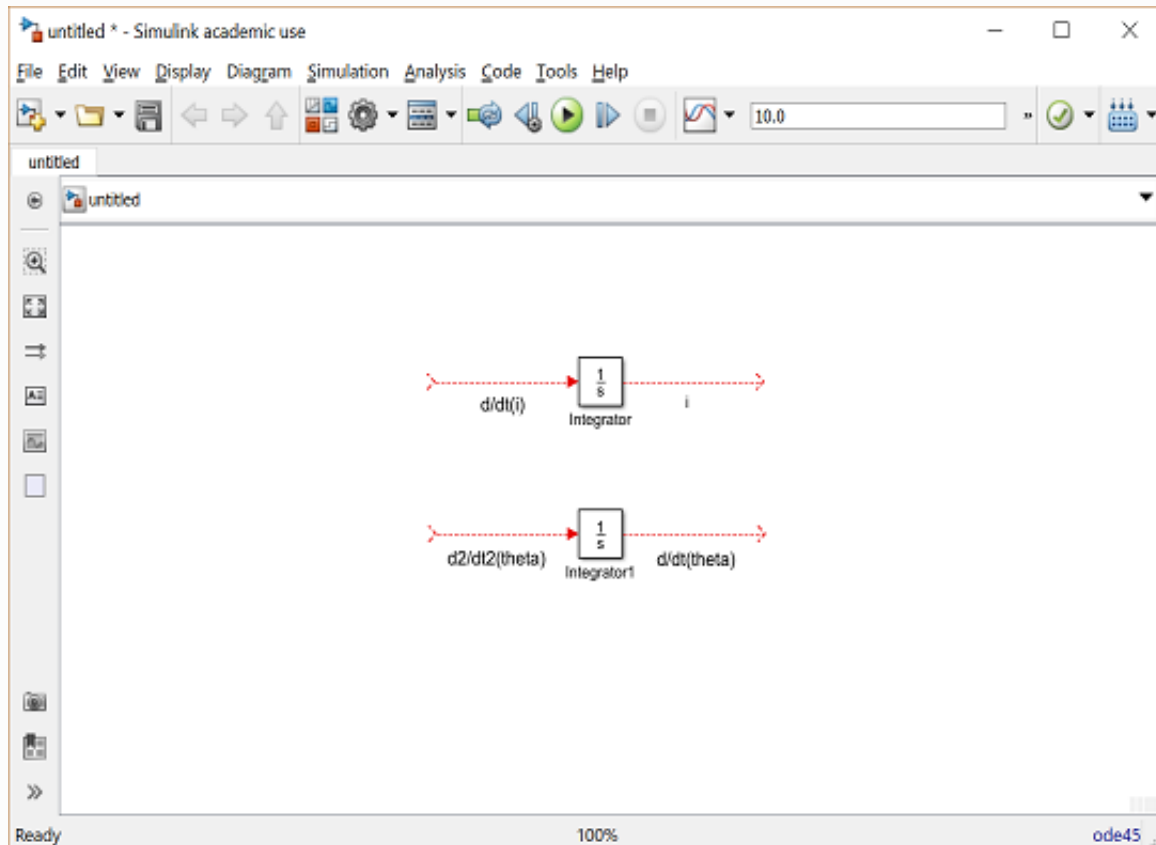
Este sistema será modelado somando os torques que atuam na inércia do rotor e **integrando a aceleração para obter a velocidade**. Além disso, as leis de Kirchoff serão aplicadas ao circuito de armadura. Primeiro, vamos modelar as integrais da aceleração rotacional e da taxa de variação da corrente da armadura:

$$\int \underbrace{\frac{d^2 \theta}{dt^2}}_{\text{aceleração}} dt = \underbrace{\frac{d\theta}{dt}}_{\text{velocidade}}$$

$$\int \frac{di}{dt} dt = i$$

Para construir o modelo de simulação, abra o **Simulink** e abra uma nova janela de modelo. Em seguida, siga as etapas listadas abaixo.

- Insira um bloco Integrador da biblioteca Simulink >> Continuous e desenhe linhas de e para seus terminais de entrada e saída.
- Rotule a linha de entrada " $d^2/dt^2$  (theta)" e a linha de saída " $d/dt$  (theta)" conforme mostrado abaixo. Para adicionar esse rótulo, clique duas vezes no espaço vazio logo abaixo da linha.
- Insira outro bloco integrador acima do anterior e desenhe linhas de e para seus terminais de entrada e saída.
- Rotule a linha de entrada " $d/dt$  (i)" e a linha de saída " $i$ ".



A seguir, vamos aplicar a lei de Newton e a lei de Kirchhoff ao sistema do motor para gerar as seguintes equações:

$$J \frac{d^2 \theta}{dt^2} = T - b \frac{d\theta}{dt}$$

$$\frac{d^2 \theta}{dt^2} = \frac{1}{J} \left( K_t i - b \frac{d\theta}{dt} \right)$$

e:

$$L \frac{di}{dt} = -Ri + V - e$$

$$\frac{di}{dt} = \frac{1}{L} \left( -Ri + V - K_e \frac{d\theta}{dt} \right)$$

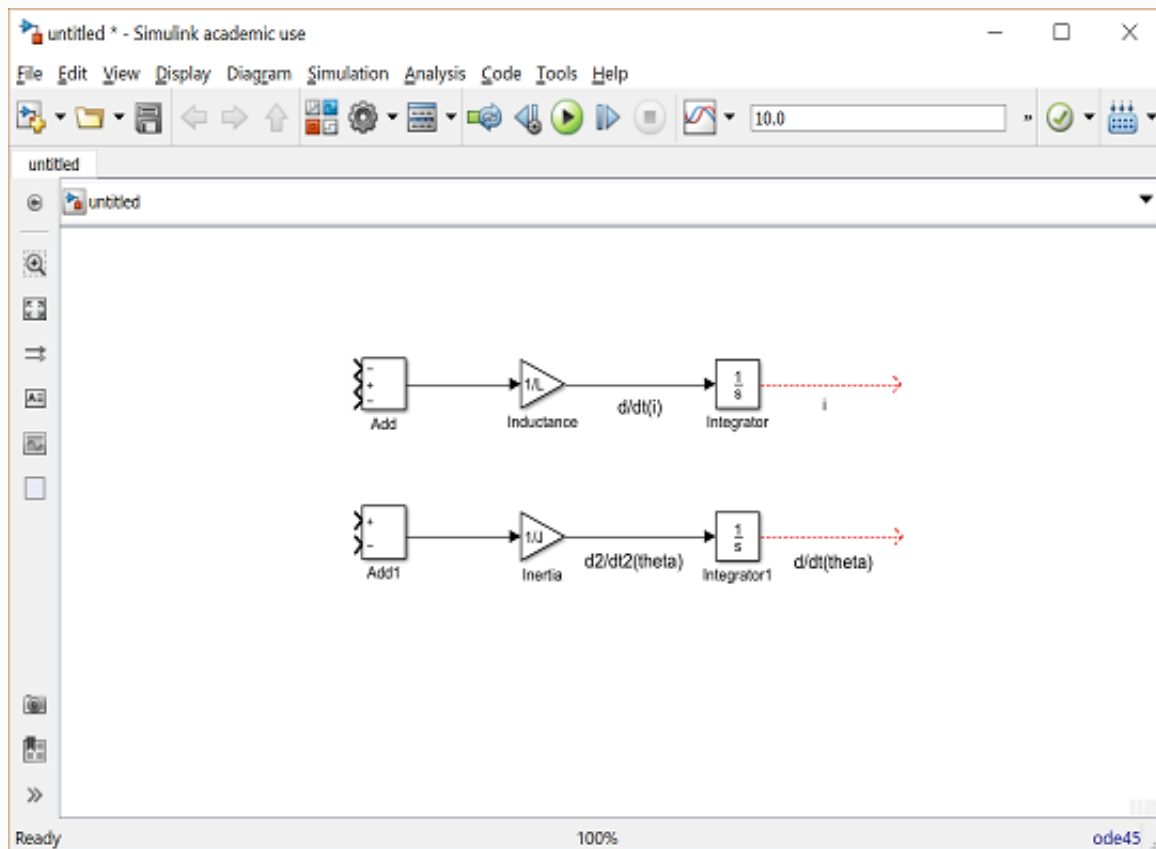
A aceleração angular é igual a  $1/J$  multiplicado pela soma de dois termos (um positivo, um negativo). Da mesma forma, a derivada da corrente é igual a  $1/L$  multiplicado pela soma de três termos (um positivo, dois negativos). Continuando a modelar essas equações no Simulink, siga as etapas fornecidas abaixo:

- Insira dois blocos de ganho da biblioteca *Simulink >> Math Operations*, um associada a cada um dos integradores.
- Edite o bloco *Gain* correspondente à aceleração angular clicando duas vezes nele e alterando seu valor para  $1/J$  (propriedades).
- Mude o rótulo deste bloco de ganho para "Inércia" clicando na palavra "Gain" abaixo do bloco.
- Da mesma forma, edite o outro valor de Ganho para  $1/L$  e seu rótulo para "Indutância".
- Insira dois blocos *Add* da biblioteca *Simulink >> Math Operations*, cada um associado com uma linha a

cada um dos blocos *Gain*.

- Edite os sinais do bloco *Add* correspondente à rotação para  $+$   $-$ , uma vez que um termo é positivo e outro é negativo (importante não confundir sequencia dos sinais).

Edite os sinais do outro *Add* bloco a  $-$   $+$   $-$  para representar os sinais dos termos na equação elétrica

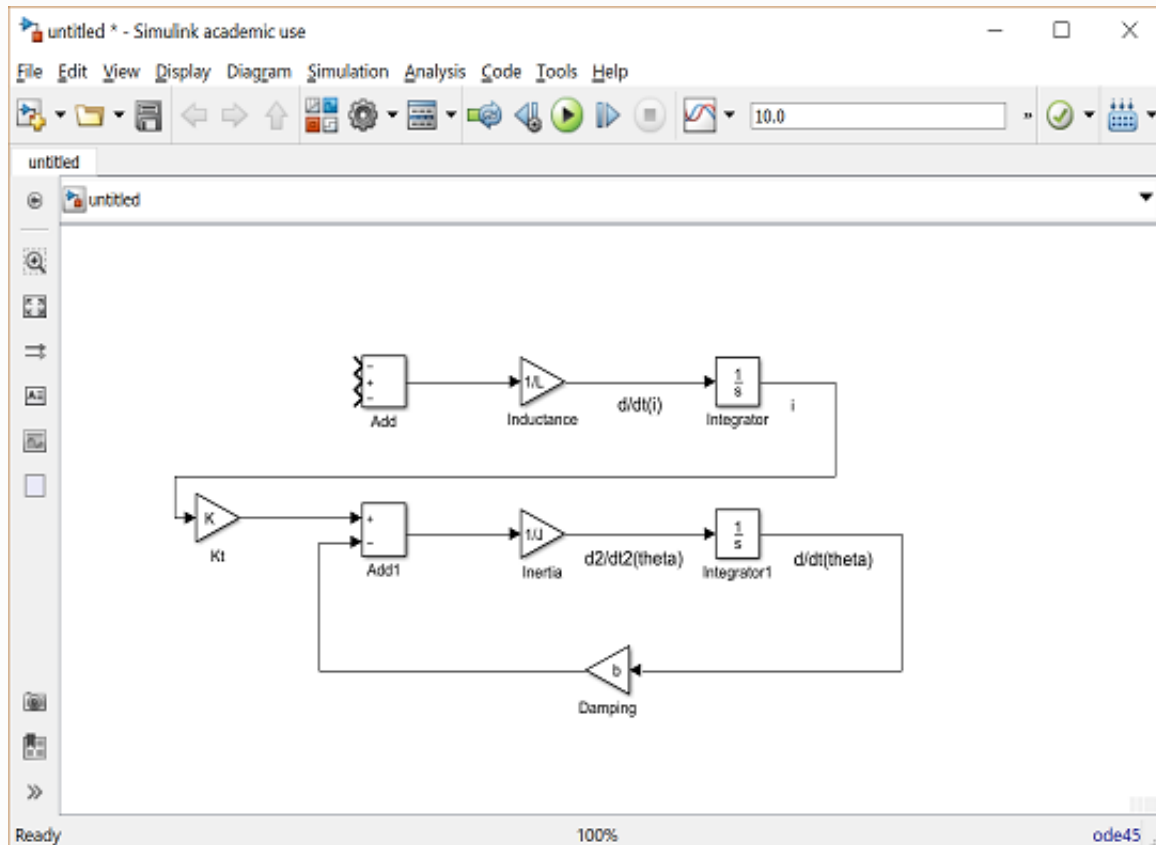


Agora, vamos adicionar os torques que são representados na equação de rotação. Primeiro, vamos adicionar o torque de amortecimento:

- Insira um bloco de *Gain* abaixo do bloco "Inércia". Em seguida, clique com o botão direito do mouse no bloco e selecione *Rotate & Flip > Flip Block* (Girar e Inverter > Inverter Bloco) no menu resultante para inverter o bloco da esquerda para a direita. Você também pode inverter um bloco selecionado mantendo pressionado **Ctrl-I**.
- Defina o valor de ganho para **b** e renomeie este bloco para "Amortecimento".
- Toque em uma linha (segure **Ctrl** enquanto desenha ou clique com o botão direito na linha) fora da saída do integrador rotacional e conecte-a à entrada do bloco "Amortecimento".
- Desenhe uma linha da saída do bloco "Amortecimento" até a entrada negativa do bloco *Add* rotacional.

A seguir, adicionaremos o torque da armadura.

- Insira um bloco *Gain* anexado à entrada positiva do bloco *Add* rotacional com uma linha.
- Edite seu valor para **Kt** para representar a constante do motor e rotule-o como "Kt".
- Continue desenhando a linha que sai do Integrador atual e conecte-o ao bloco "Kt".



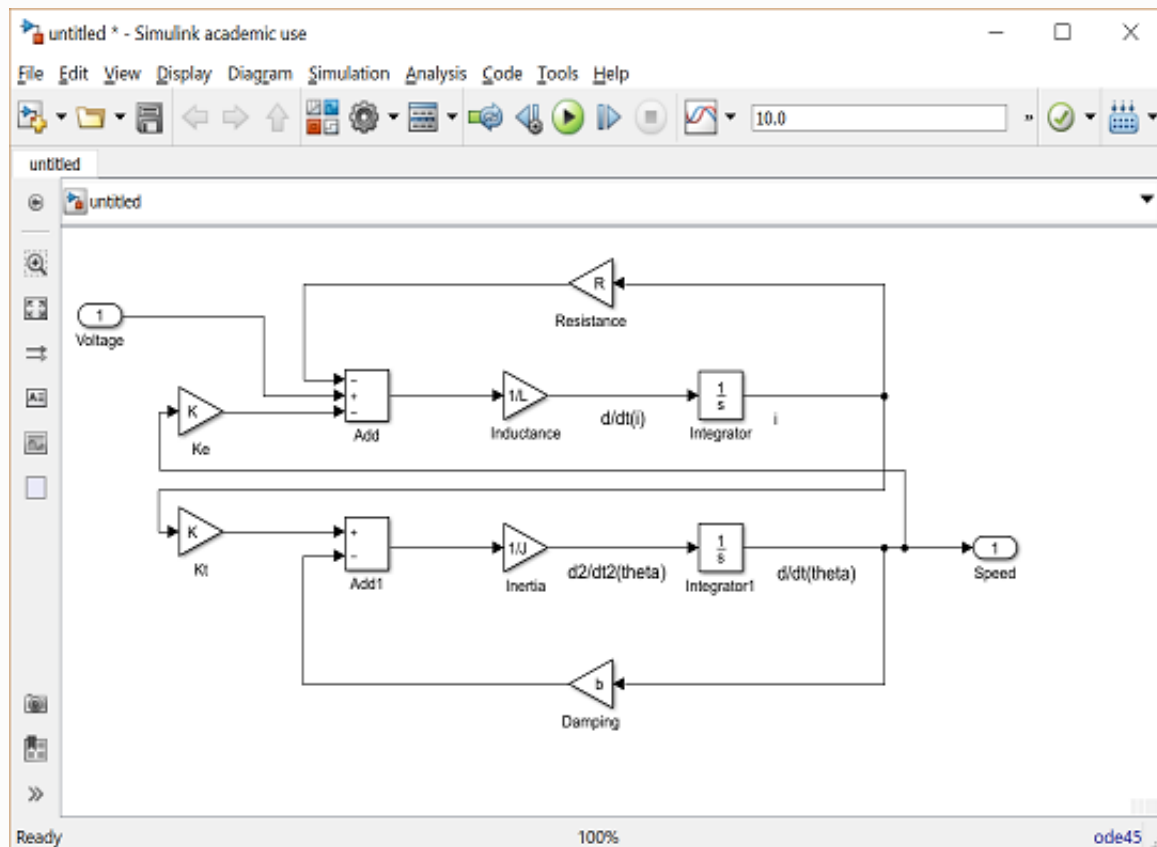
Agora, vamos adicionar os termos de tensão que são representados na equação elétrica. Primeiro, adicionaremos a queda de tensão na resistência da armadura:

- Insira um bloco de *Gain* acima do bloco de "Indutância" e gire-o da esquerda para a direita (*Flip*).
- Defina o valor de ganho para **R** e renomeie este bloco para "Resistência".
- Toque em uma linha da saída do integrador atual e conecte-a à entrada do bloco "Resistência".
- Desenhe uma linha da saída do bloco "Resistência" para a entrada negativa superior do bloco *Add* da equação atual.

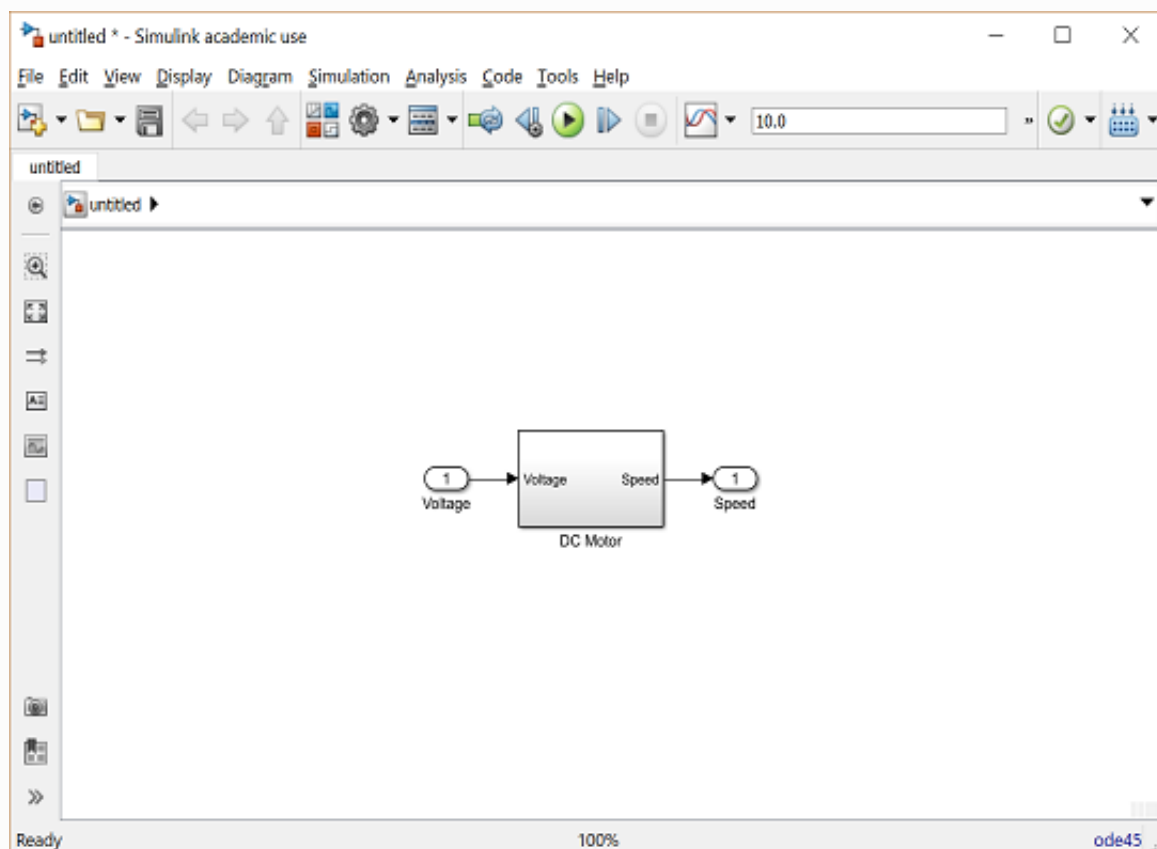
Em seguida, adicionaremos a fem traseira do motor.

- Insira um bloco *Gain* anexado à outra entrada negativa do bloco *Add* atual com uma linha.
- Edite seu valor para **Ke** para representar a constante emf de retorno do motor e rotule-o como "Ke".
- Toque em uma linha da saída do integrador rotacional e conecte-a ao bloco "Ke".
- Adicione blocos *In1* e *Out1* da biblioteca *Simulink >> Ports & Subsystems* e, respectivamente, rotule-os de "Voltagem" e "Velocidade".

O design final deve ser semelhante ao exemplo mostrado na figura abaixo:

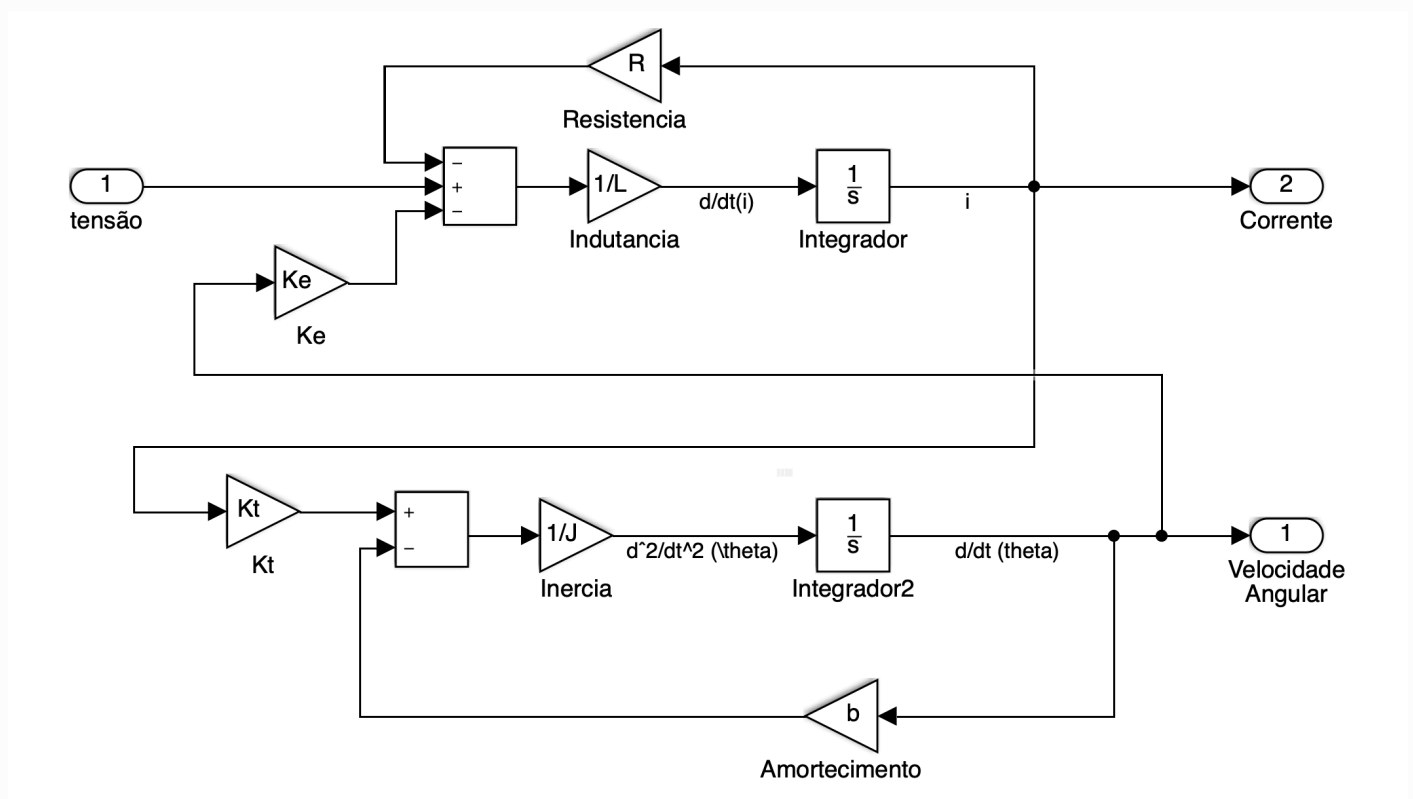


Para salvar todos esses componentes como um **único bloco de subsistema**, primeiro **selecione todos os blocos** e, em seguida, selecione **Criar Subsistema** a partir da Seleção após clicar com o botão direito do mouse na parte selecionada. Nomeie o subsistema "Motor DC" e salve o modelo. Seu modelo deve ter a seguinte aparência.

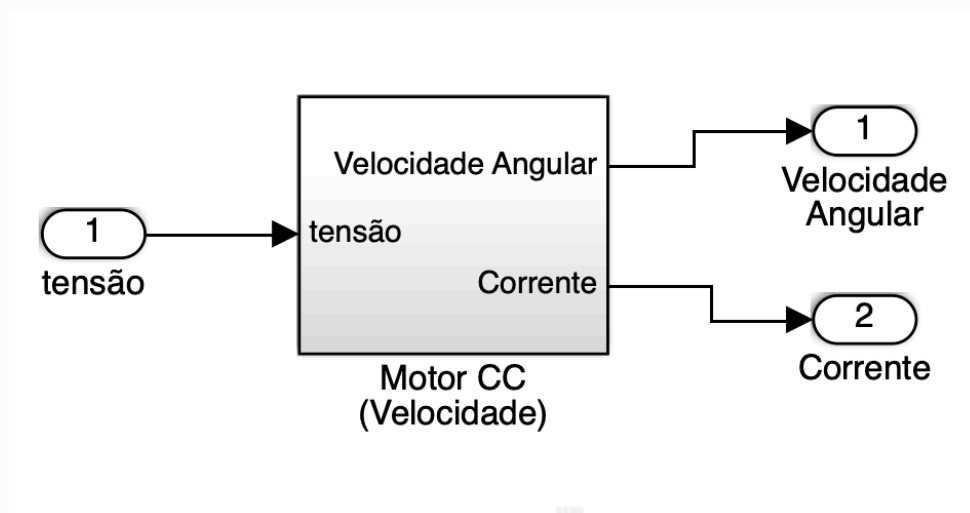


Você também pode baixar o arquivo para este sistema clicando com o botão direito aqui e selecionando o link Salvar como .... Usamos este modelo na seção Velocidade do Motor DC: Projeto do Controlador Simulink ([DC Motor Speed: Simulink Controller Design](#)).

Ou o diagrama **antes** de ser agrupado pode ser alterado para incluir a corrente exigida pelo motor:



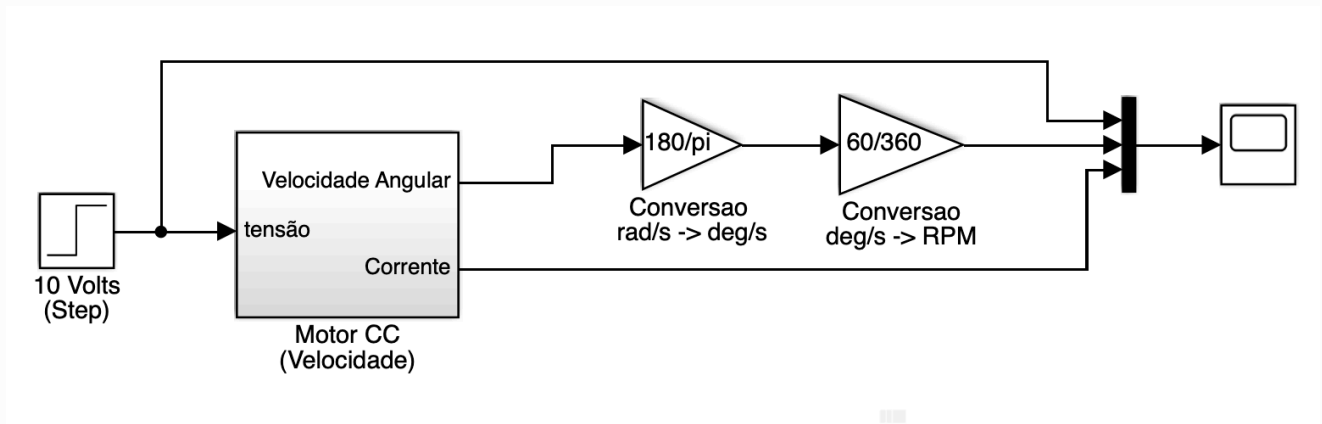
Que transformado num sub-sistema resulta em (arquivo [motor\\_cc\\_velocidade.slx](#)):



Este subsistema pode ser usado nas próximas simulações usando-se a opção **Save As...**

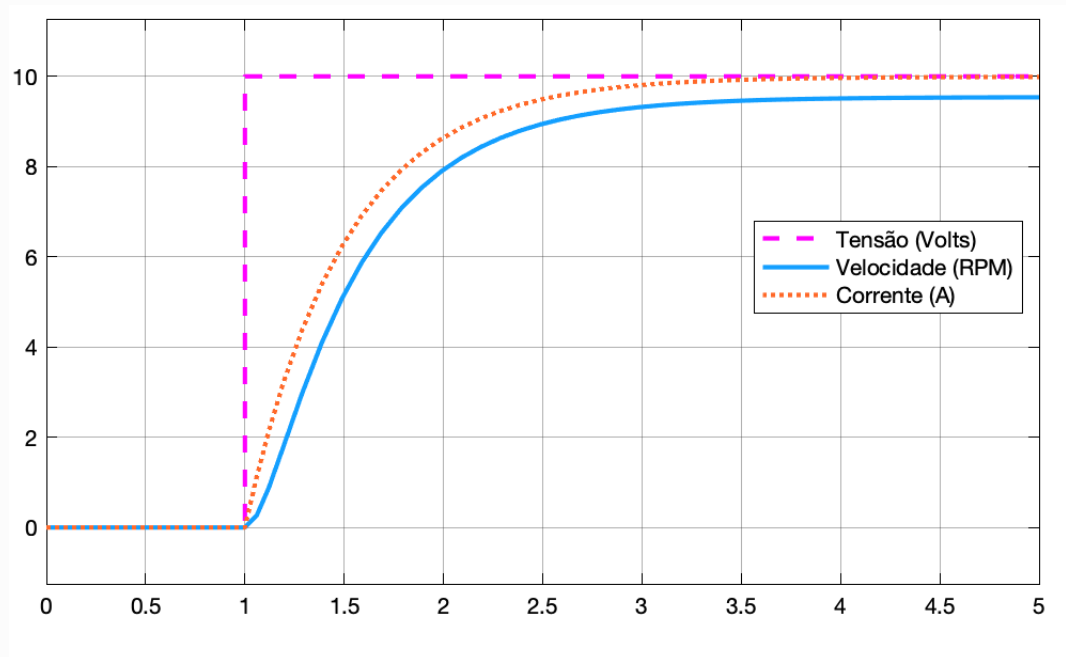
# Exemplo de Ações de Controle de Velocidade de Motor CC

- Ex<sub>1</sub>: Teste de controle em Malha aberta (arquivo: [motor\\_cc\\_velocidade\\_teste\\_MA.slx](#))



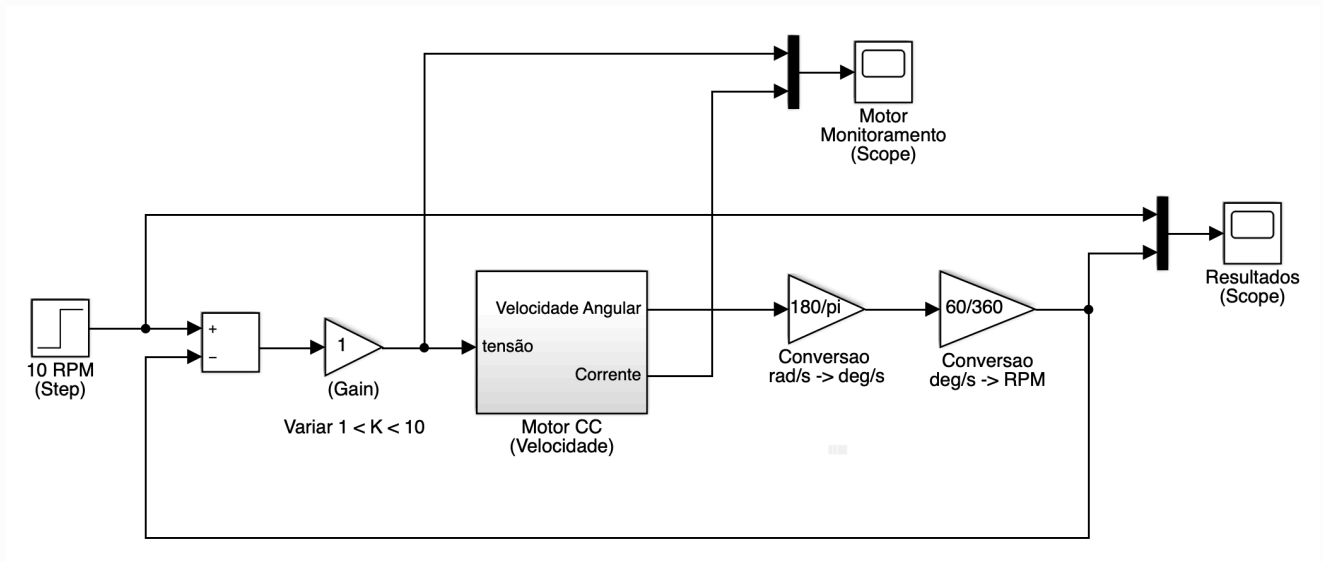
Note que neste caso, estamos alimentando o motor com 10 Volts (degrau) e monitorando a velocidade de saída, não em  $rad/s$  mas em RPM (por isto os blocos *Gain* de conversão).

Resultados:

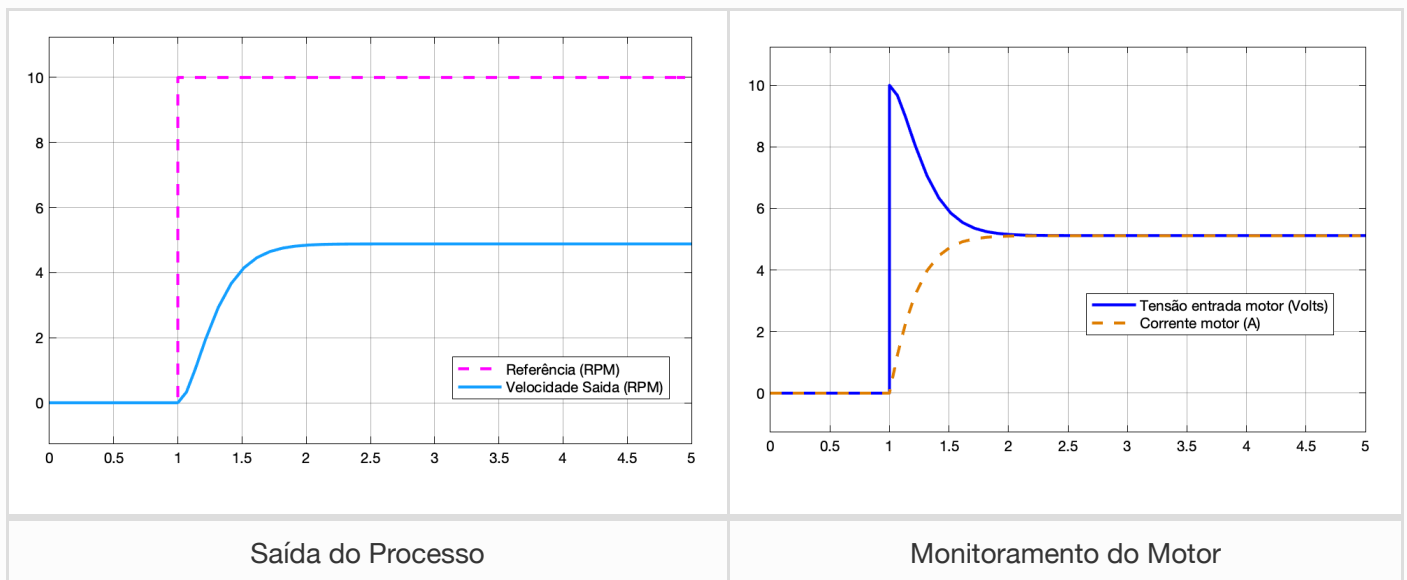


- Ex<sub>2</sub>: Fechando uma malha de **controle Proporcional** de velocidade para este motor, para tentar manter velocidade em 10 RPM (arquivo: [motor\\_cc\\_velocidade\\_cont\\_Prop.slx](#)):

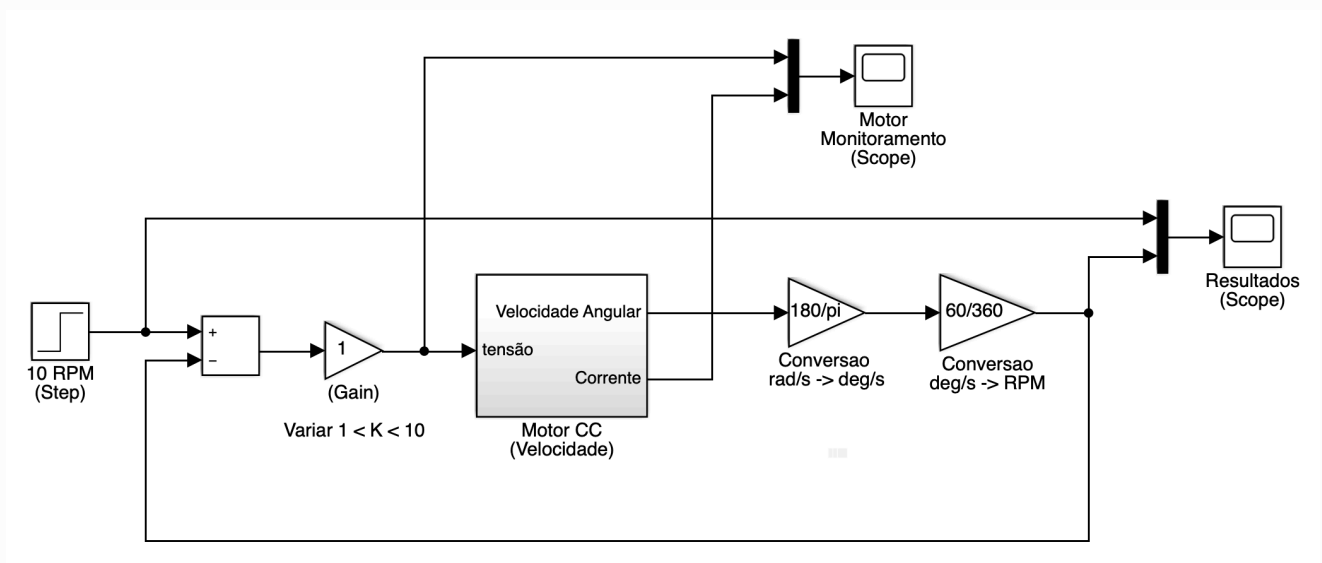




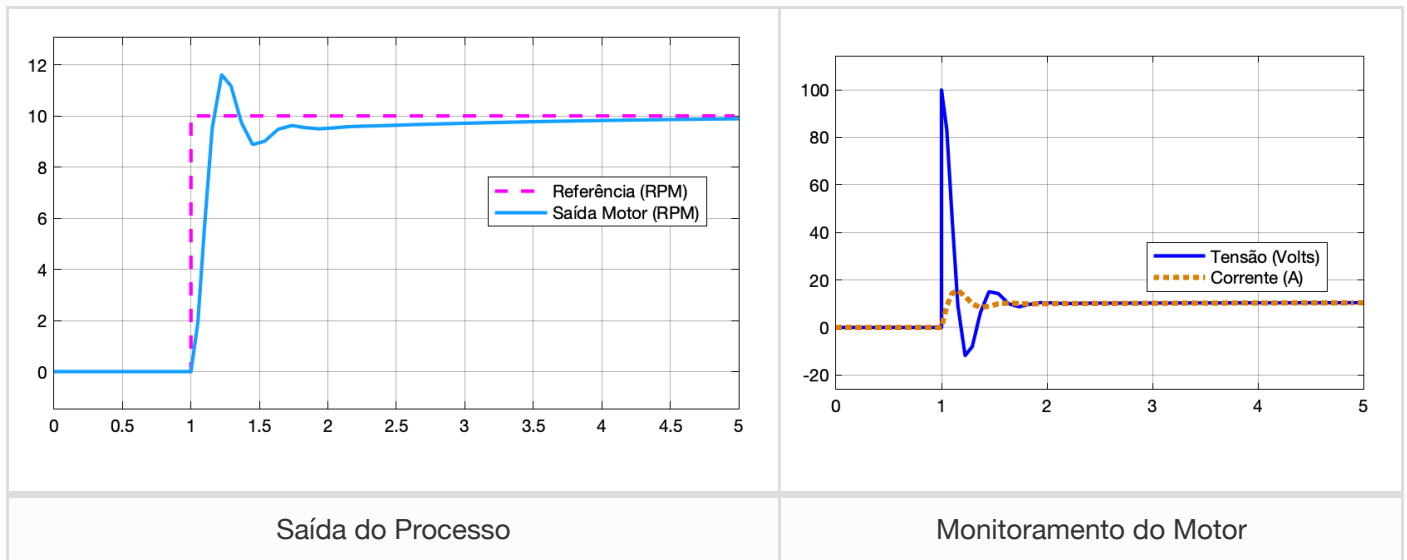
Que resulta em:



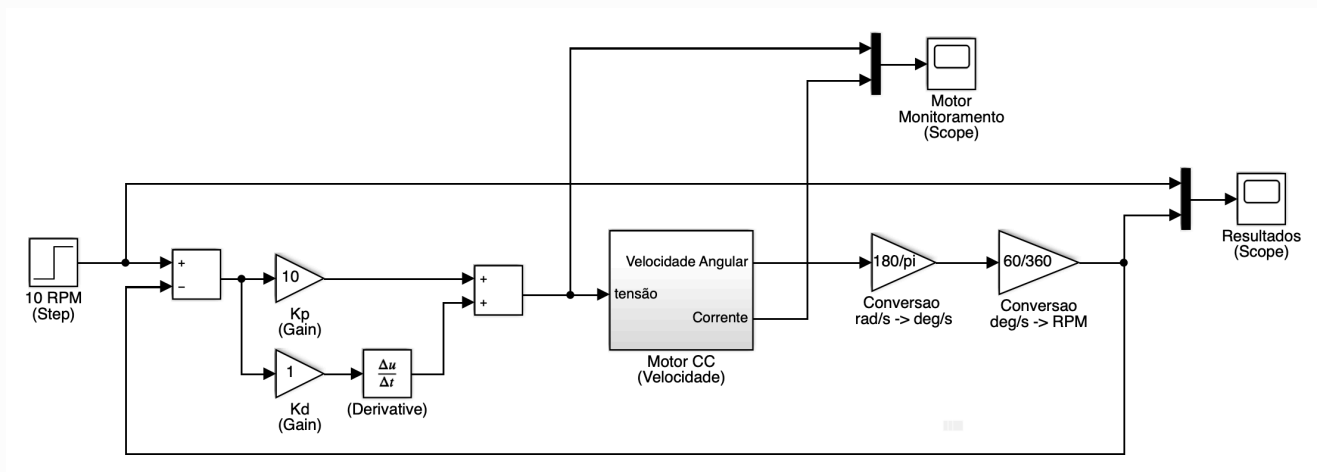
- Ex<sub>3</sub>: Fechando uma malha de **controle PI** (Proporcional-Integral) de velocidade para este motor, para tentar manter velocidade em 10 RPM (Arquivo: [motor\\_cc\\_velocidade\\_cont\\_PI.slx](#)):



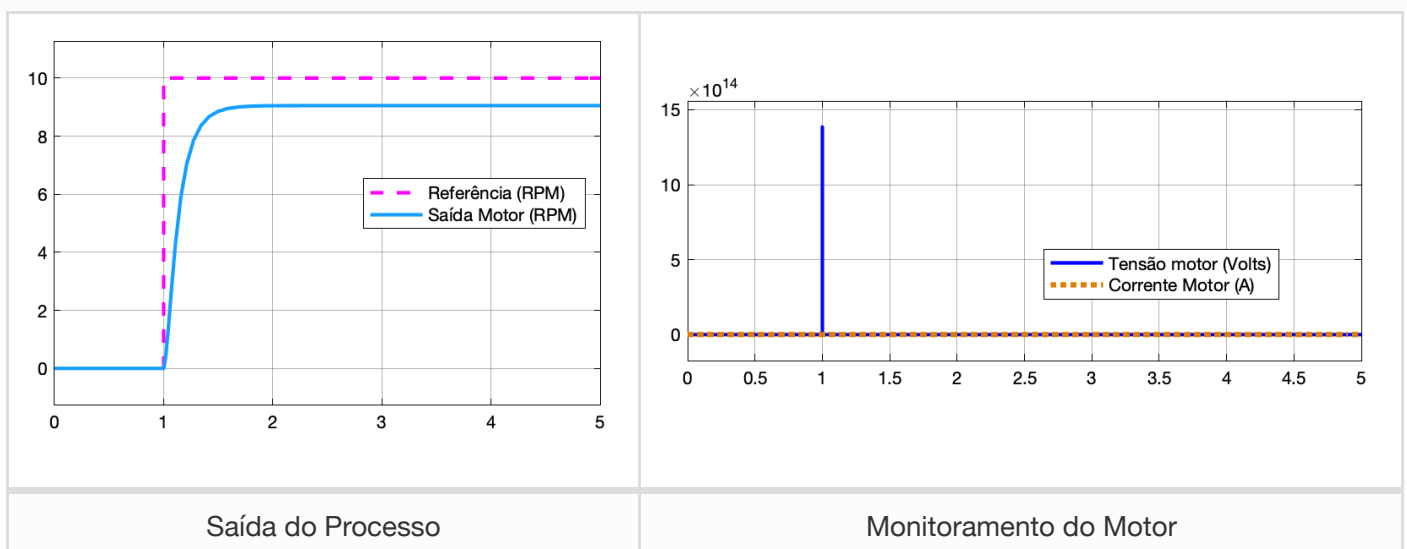
Que resulta em:



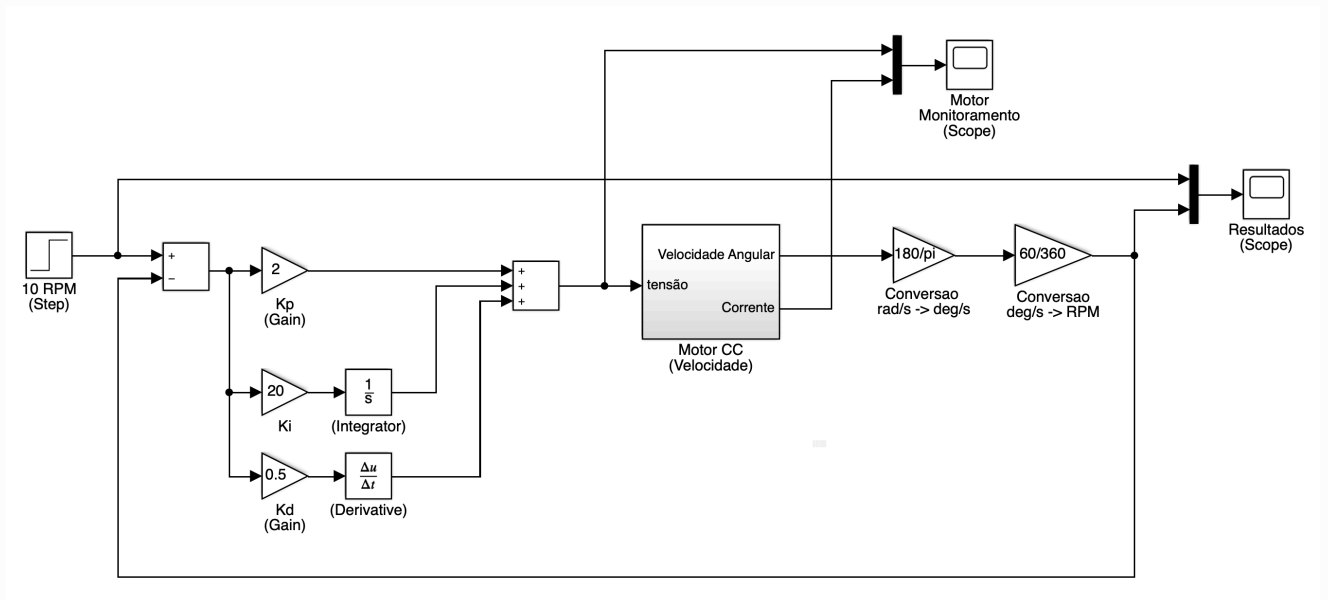
- Ex<sub>4</sub>: Fechando uma malha de **controle PD** (Proporcional-Derivativo) de velocidade para este motor, para tentar manter velocidade em 10 RPM (Arquivo: [motor\\_cc\\_velocidade\\_cont\\_PD.slx](#)):



Resultados:



- Ex<sub>4</sub>: Fechando uma malha de **controle PID** (Proporcional-Integrativo-Derivativo) de velocidade para este motor, para tentar manter velocidade em 10 RPM (Arquivo: [motor\\_cc\\_velocidade\\_cont\\_PID.slx](#)):



Resultados:

