2016 BRAZIL STUDY ABROAD PROGRAM

TEXAS A&M UNIVERSITY- UNIVERSITY OF SAO PAULO

KENNY ANDERSON QUEIROZ CALDAS

MAURÍCIO EIJI NAKAI

ELMER ALEXIS GAMBOA PEÑALOZA

RODOLPHO VILELA ALVES NEVES

MICHEL BESSANI

RAFAEL FERNANDO QUIRINO MAGOSSI

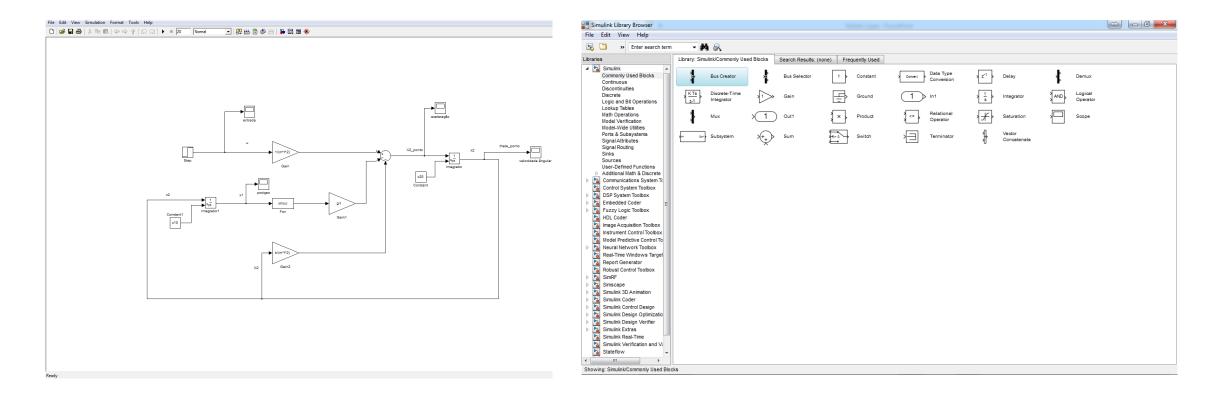
DEPARTAMENTO DE ENGENHARIA ELÉTRICA USP - SÃO CARLOS

Matlab has a tool called Simulink. Simulink allows you to create block diagrams, system modelling, analysis and so on.

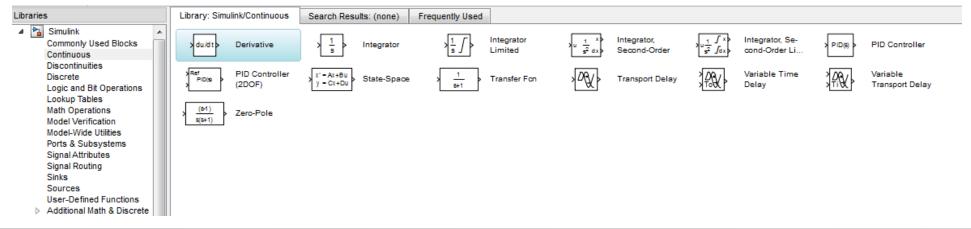
To access Simulink, type:

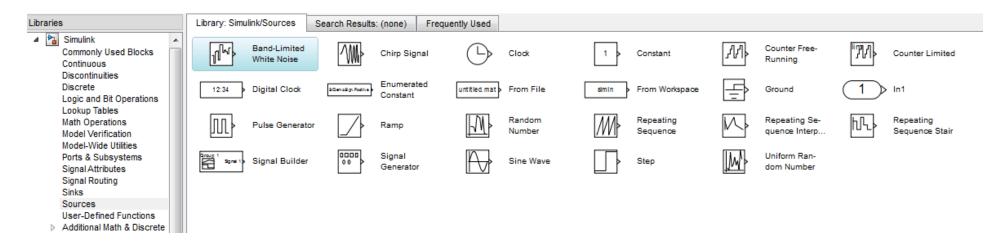
>> simulink

Simulink interface



Simulink interface





Simulation using gain, sum and integrator

Commonly Used Blocks Continuous Discontinuities Discrete Logic and Bit Operations Lokey Tables Math Operations Model Verification Model Virie Utilities Ports & Subsystems Signal Routing Sinks Sources User-Defined Functions Additional Math & Discrete Control System Toolbox Fuzzy Logic Toolbox Simulant Coder Simulink 20 Animation Simulink Coder Simulink Extras Stateflow System Identification Toolbo	Simulink Library Browser		
Libraries Libraries Library: Simulink Commonly Used Blocks Continuous Discontinuities Discrete Logic an Bit Operations Lockup Tables Math Operations Model Vification Model Vification Model Vification Model Vification Signal Attributes Signal Att	File Edit View Help		
 Simulink Commonly Used Blocks Continuouss Discortinuities Discrete Logic and Bto Operations Lookup Tables Model - Wide Utilities Ports & Subsystems Signal Attributes Signal Routing Sinks Sources User-Defined Functions Additional Math & Discrete Control System Toolbox Fuzy Logic Toolbox Image Acquisition Toolbox Simulink Coder Simulink Coder System Identification Toolbo System Identification Toolbo System Identification Toolbo System Identification Toolbo 	🔁 🗀 » gain	- M &	
Commonly Used Blocks Continuous Discrete Logic and Bit Operations Lookup Bables Math Operations Model Verification Model Verification Model Verification Signal Routing Sinks Sources User-Defined Functions Additional Math & Discrete Control System Toolbox Meural Network Toolbox Simulink 20 Animation Simulink Coder Simulink Extras Stateflow System Identification Toolb	Libraries	Library: Simulink/Commonly Used Blocks Found: 'gain' Frequently Used	
	Ibraries Ibraries Simulink Commonly Used Blocks Continuous Discontinuities Discrete Logic and Bit Operations Lookup Tables Moth Operations Model Verification Model-Wide Utilities Ports & Subsystems Signal Attributes Signal Attributes Signal Attributes Signal Routing Sinks Sources User-Defined Functions Additional Math & Discrete Control System Toolbox Fuzzy Logic Toolbox Image Acquisition Toolbox Mary Logic Toolbox Simulink Coder Simulink 2D Animation Simulink 2D Animation Simulink Coder Simulink Extras Simulink Extras Simulink Extras	Library: Simulink/Commonly Used Blocks Found: 'gain' Frequently Used Simulink Simulink Simscape DDD PS Gain	3 (2)
Matches for 'gain' 2 blocksets 0 subsystems 4 blocks			

This is the Simulink library. You can search for the block you want using the search box or the options on the left side. For example:

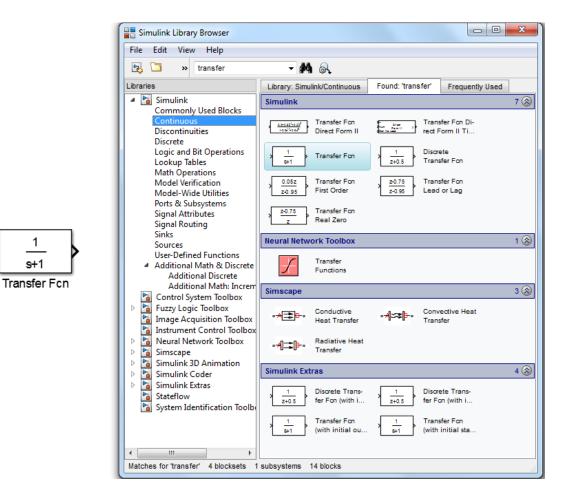
Gain

Sum

Integrator

Transfer function

To create a transfer function on Simulink, you can use the 'Transfer Fcn' block.



1

s+1

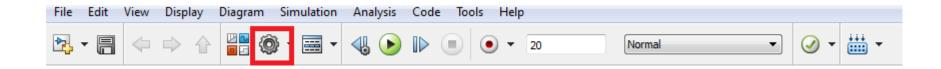
First order simple example

0.001s s+1 Transfer Fcn

Function Block Parameters: Transfer Fcn				
Transfer Fcn				
The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.				
Parameters				
Numerator coefficients:				
[0.001 0]				
Denominator coefficients:				
[1 1]				
Absolute tolerance:				
auto				
State Name: (e.g., 'position')				
1				
OK Cancel Help Apply				

Simulation parameters

Before you run your simulation, it's necessary to set the configurations:



Simulation parameters

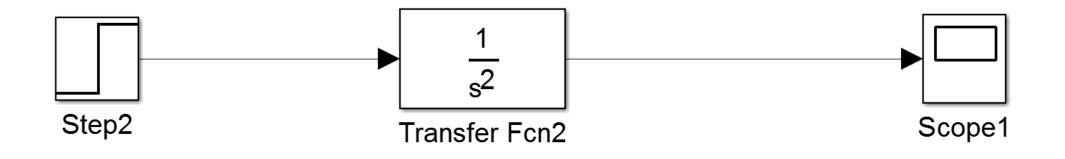
	Simulation time	
Solver Data Import/Export	Start time: 0.0	Stop time: 20
 Optimization Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation HDL Code Generation 	Solver options Type: Fixed-step Fixed-step size (fundamental sample time): Tasking and sample time options Periodic sample time constraint: Tasking mode for periodic sample times: Automatically handle rate transition for data transfer Higher priority value indicates higher task priority	 Solver: ode3 (Bogacki-Shampine) 0.001 Unconstrained Auto

First order system example

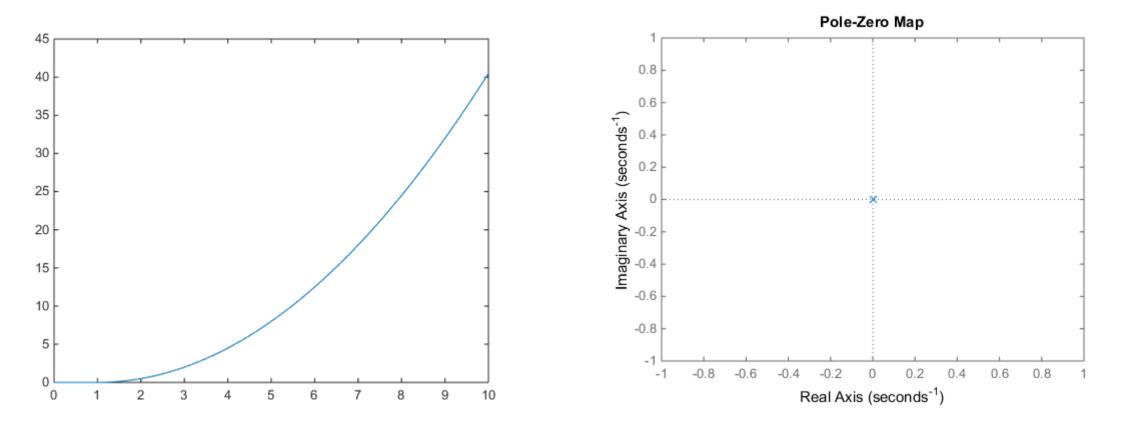
To check the response of a transfer function, use the 'Step' and 'Scope' on Simulink. Run and double click on 'Scope' to see the response.



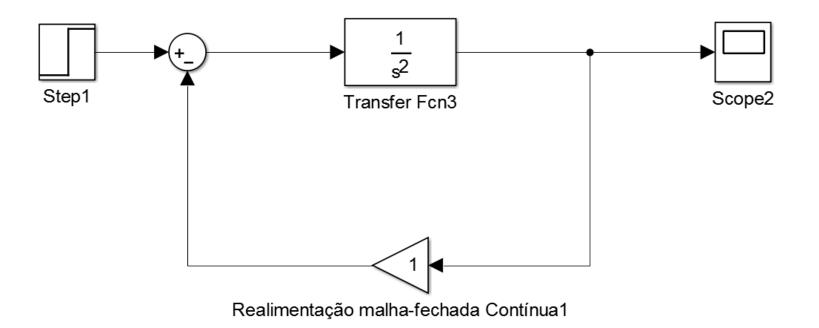
Lets try this transfer function and see what is the step response using Simulink



Forced step response



Lets try the same transfer function used in open loop, but now in a closed loop and see what happen using a step response.

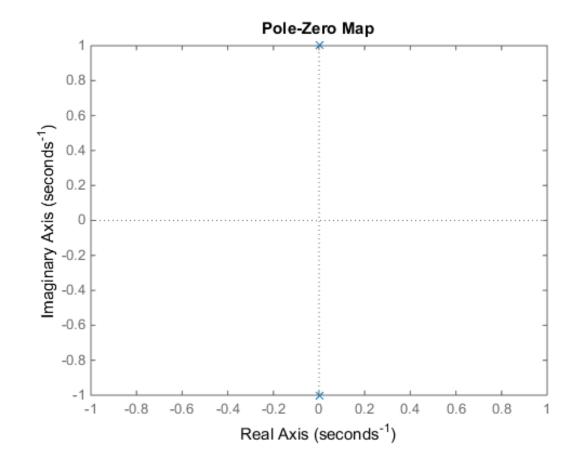


Before run the Simulink lets get the transfer function in a closed loop via Matlab console typing

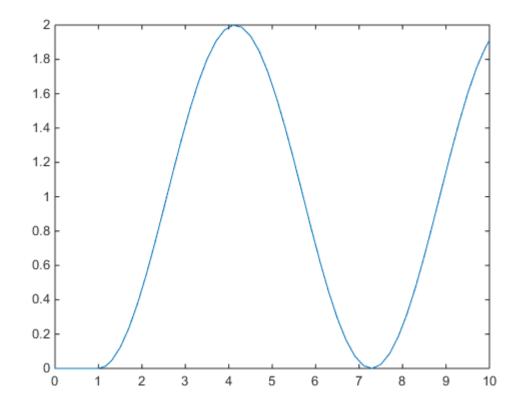
s = tf('s'); G = 1/s^2; T = feedback(G,1) pzmap(T)

Now the transfer function is given by:

$$G_{cl} = \frac{1}{s^2 + 1}$$

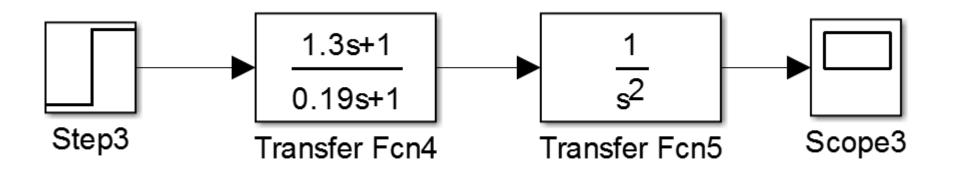


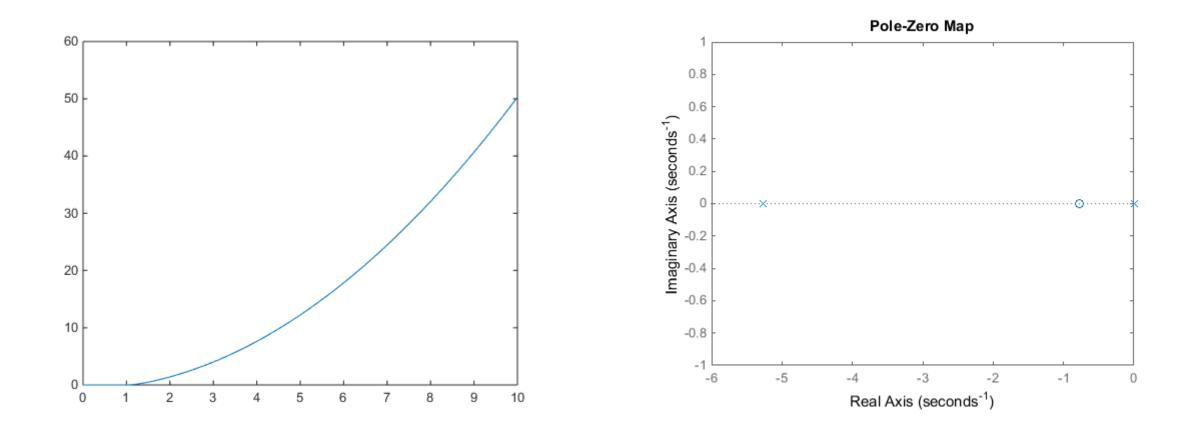
The new closed loop response is:



Now lets try to do the same to this system:

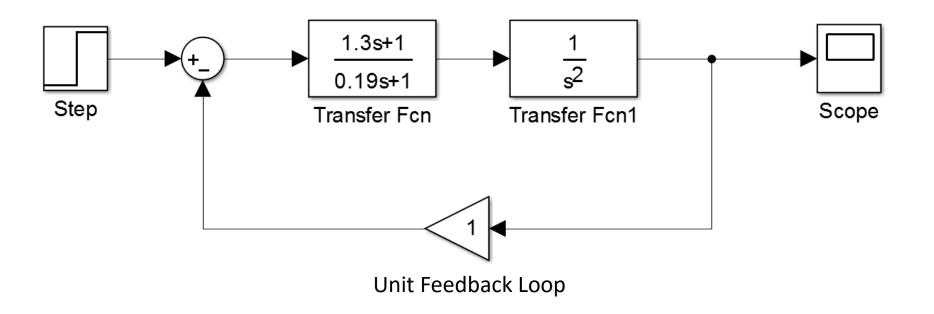
- Plot the root locus using of this system "pzmap()"
- Plot the step forced response to the system

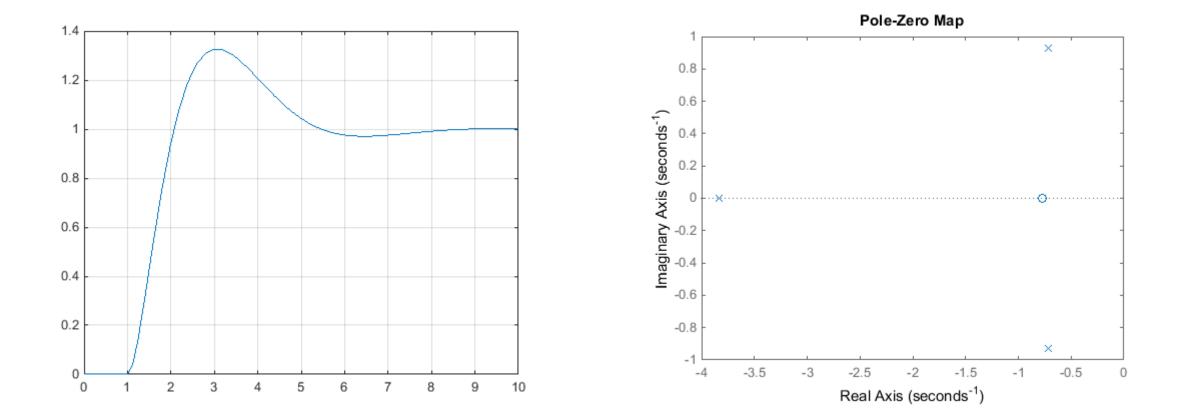




Analyze the same system before but with a unit feedback loop:

- Plot the root locus using of this system "pzmap()"
- Plot the step forced response to the system





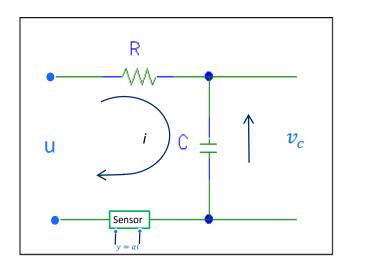
Hands on!

Built the transfer function bellow in Simulink and compare the responses to a Step input, with a Proportional, Integral and Derivative controllers:

$$G(s) = \frac{6(s+2)}{(s+1)(s+3)}$$

You'll have to use the 'Transfer Fcn', 'PID Controller', 'Scope', 'Step', 'Sum' and 'Bus Creator' blocks for this simulation.

Simple RC circuit example

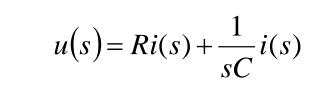


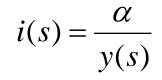
If
$$v_c = \frac{1}{C} \int i dt$$
 then $\frac{dv_c}{dt} = \frac{i}{C}$
 $i = \frac{u - v_c}{R}$
 $\frac{dv_c}{dt} = \frac{u - v_c}{RC}$ First order differential equation.

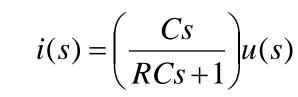
RC circuit transfer function

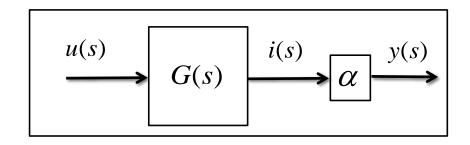
 v_c

Kirchhoff









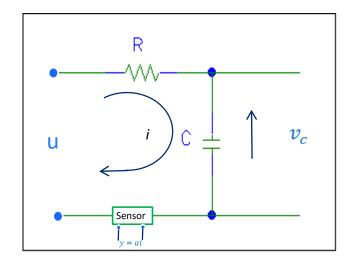
R

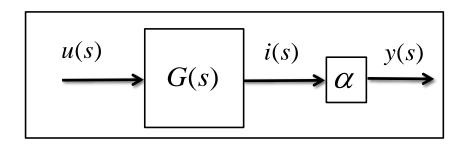
Sensor

u

$$G(s) = \frac{Cs}{RCs+1}$$

RC circuit transfer function





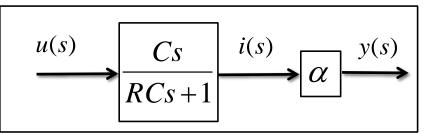
$$K(s) = \frac{y(s)}{u(s)} = \alpha G(s) = \frac{\alpha Cs}{RCs + 1}$$

$$y(s) = \left(\frac{\alpha Cs}{RCs+1}\right)u(s)$$

Hands on!

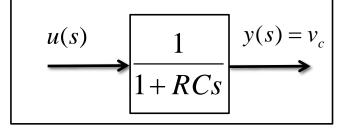
1. Built the closed loop system of the RC circuit using Simulink:

a)



% constants can be defined at the Command Window or in a file .m >>R = 1000; >>C = 1000e-6; >>alpha = 1000;

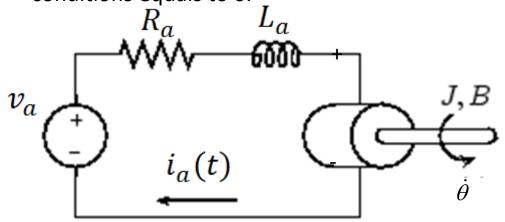
b)



2. Simulate the step responses .



1. Find the transfer function representation of a DC motor with input V_a and output θ . Consider the initial conditions equals to 0.



2. Build the closed loop system using Simulink.

3. Simulate the step response.

Constants:

J=3.2284E-6; B=3.5077E-6; Kt=Ke=0.0274; Ra=4; La=2.75E-6;

Newton $K: (x) \to \mathbb{R}$

$$K_t i_a(t) = B\dot{\theta} + J\ddot{\theta}$$

Kirchhoff

$$L_a \frac{di_a}{dt} + R_a i_a = v_a - K_e \dot{\theta}$$



[1] Matlab Product Help.

 [2] Matlab Demystified. A Self-Teaching Guide, David McMahon, McGraw Hill.
 [3] Matlab: An Introduction with Applications, Amos Gilat, Fourth Edition, JOHN WILEY & SONS.