Introduction to Simulink

Linguaggio Programmazione Matlab-Simulink (2018/2019)

Introduction to Simulink

- Simulink is a commercial tool for modeling, simulating and analyzing dynamic systems.
- Its primary interface is a graphical block diagramming tool and a customizable set of block libraries.
- It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it.
- Simulink is widely used in control theory and digital signal processing for simulation and design.
- A dynamic system may be given as a differential equation

Example: electric circuit

For the circuit of Figure B.1, the initial conditions are $i_L(0^-) = 0$, and $v_c(0^-) = 0.5\ V$. We will compute $v_c(t)$.

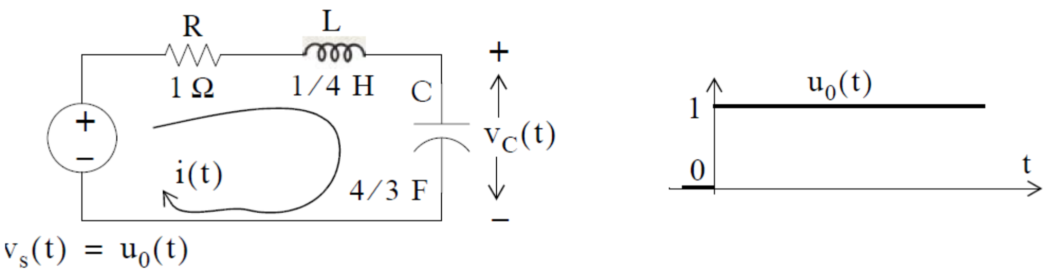


Figure B.1. Circuit for Example B.1

For this example,

$$i = i_L = i_C = C\frac{dv_C}{dt} \tag{B.1}$$

and by Kirchoff's voltage law (KVL),

$$Ri_L + L\frac{di_L}{dt} + v_C = u_0(t) \tag{B.2}$$

Substitution of (B.1) into (B.2) yields

Example: electric circuit

$$RC\frac{dv_C}{dt} + LC\frac{d^2v_C}{dt^2} + v_C = u_0(t) \tag{B.3}$$

Substituting the values of the circuit constants and rearranging we obtain:

$$\frac{1}{3}\frac{d^2v_C}{dt^2} + \frac{4}{3}\frac{dv_C}{dt} + v_C = u_0(t)$$

$$\frac{d^2v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3u_0(t) \tag{B.4}$$

$$\frac{d^2v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3 \qquad t > 0 \tag{B.5}$$

Example: electric circuit

- Characteristic equation:

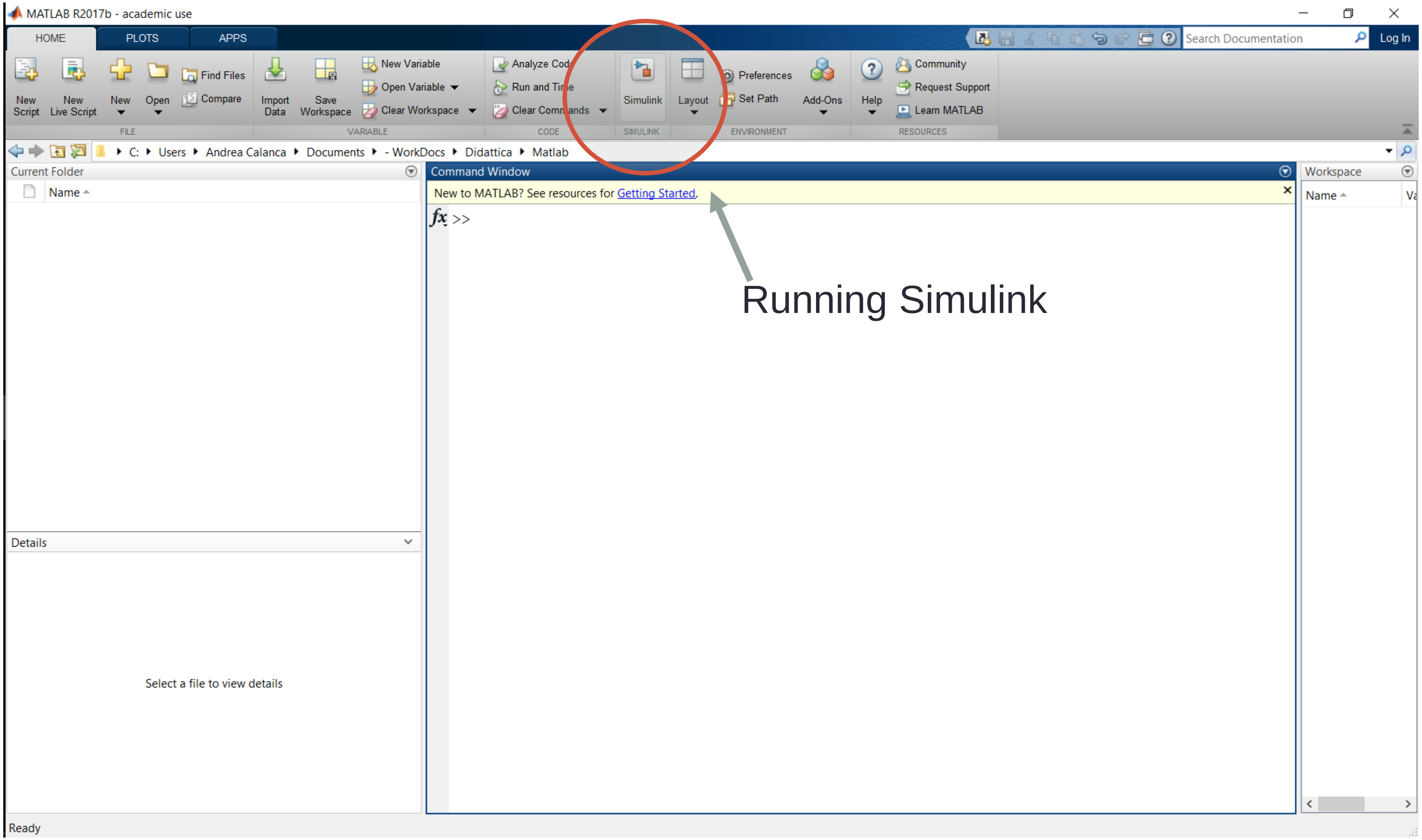
$$s^2 + 4s + 3 = 0 \qquad \text{(B.6)}$$

- The transfer function:

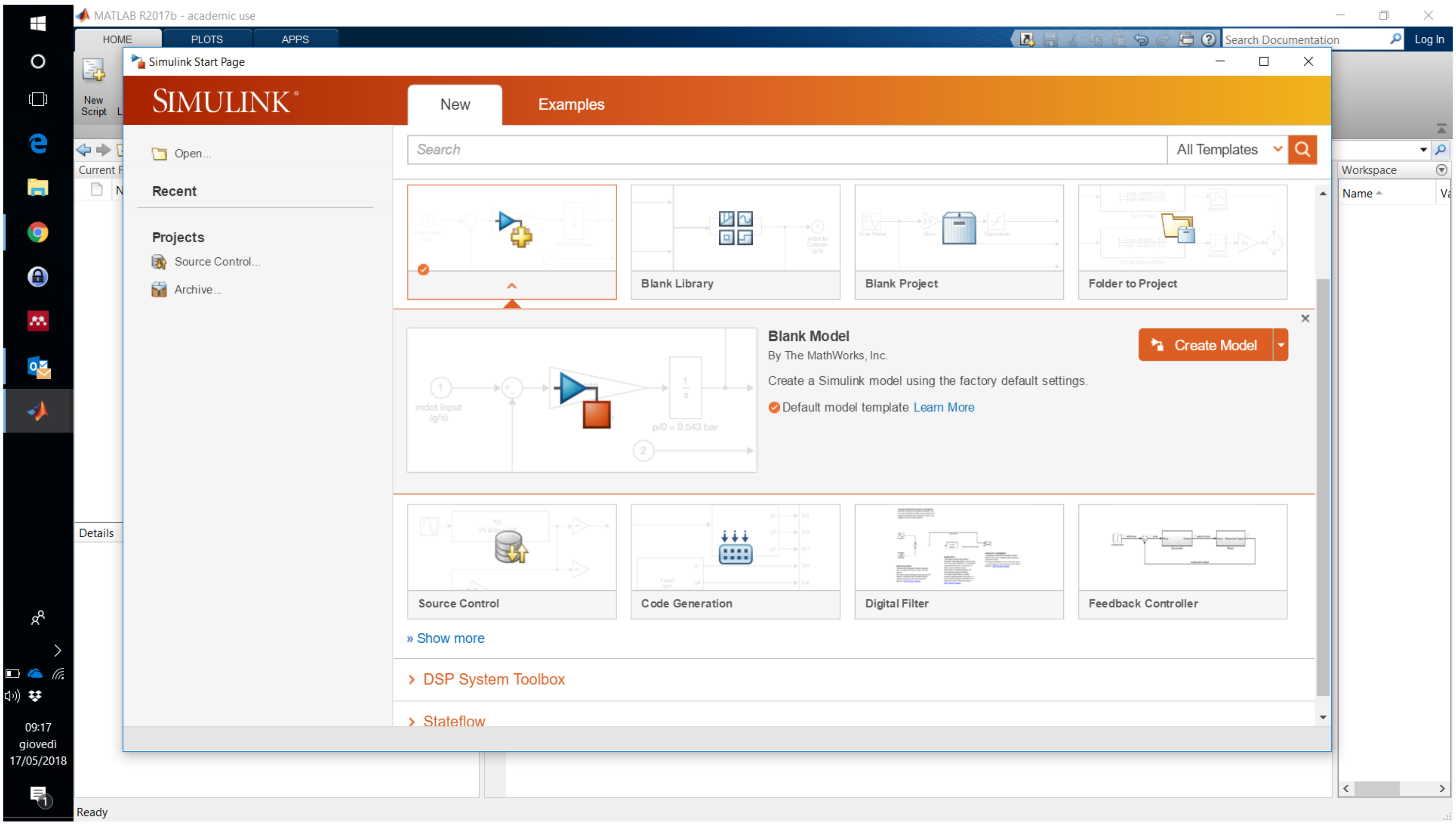
$$\mathrm{H}(s) = \frac{3}{s^2+4s+3} \qquad \text{(B.7)}$$

- We can plug this directly in Simulink

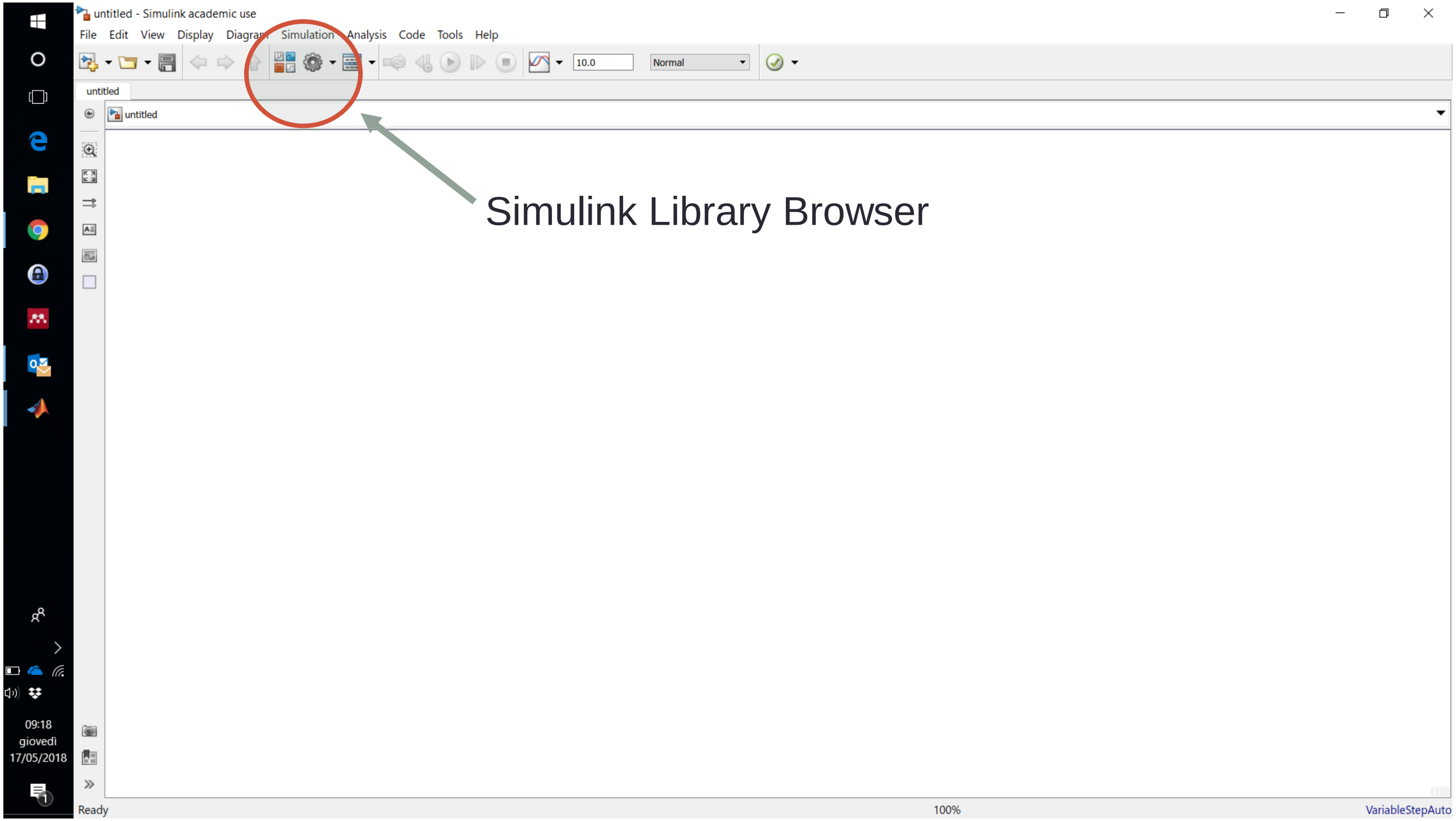
Implementation in Simulink



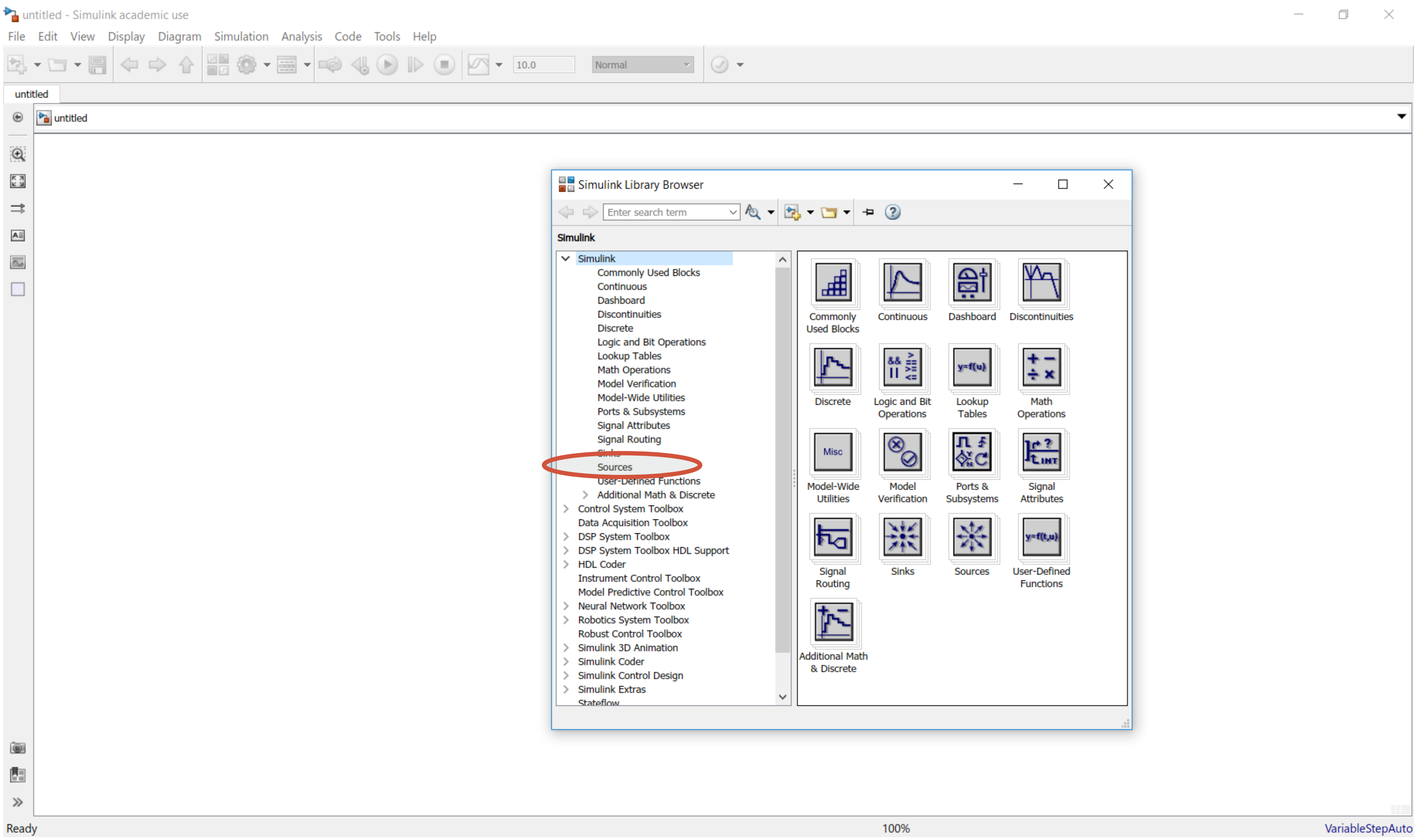
Implementation in Simulink



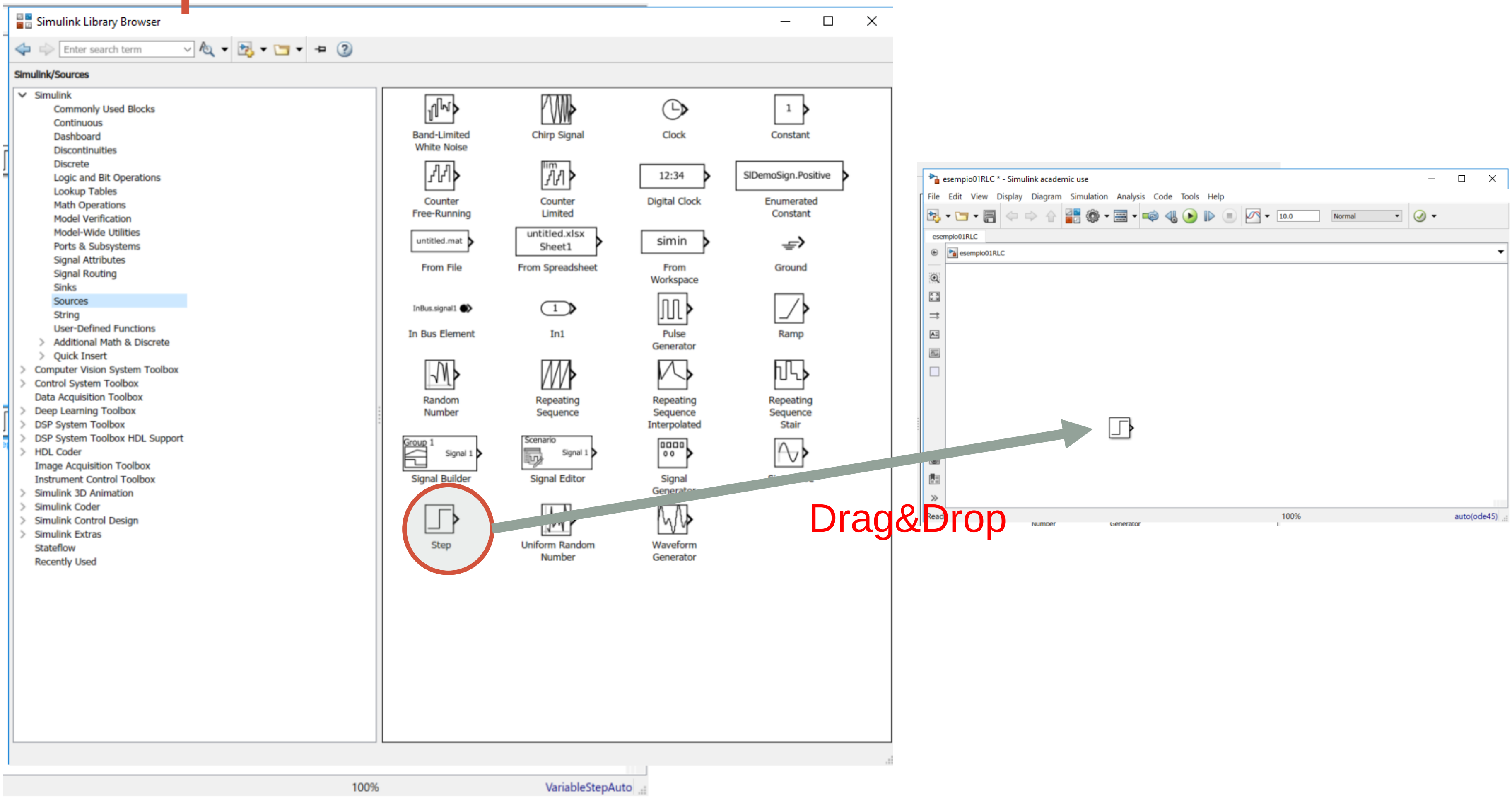
Implementation in Simulink



Implementation in Simulink



Implementation in Simulink

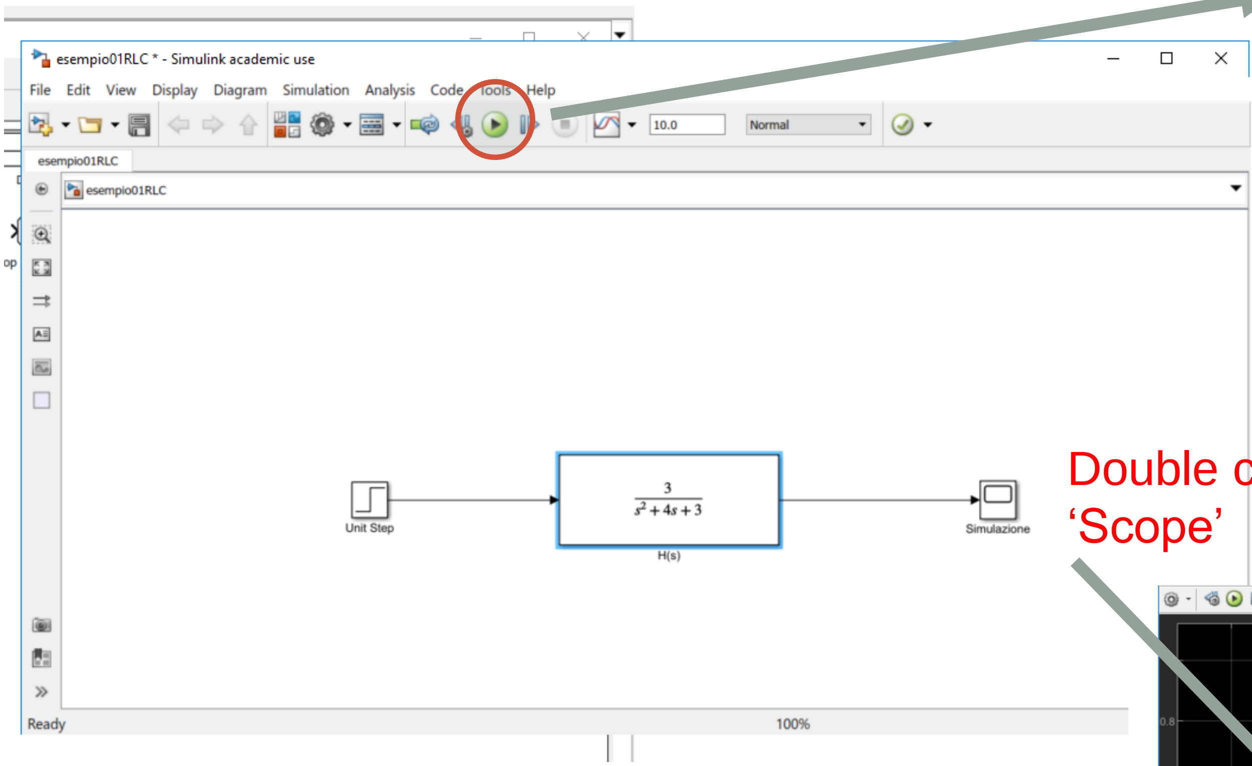


Implementation in Simulink

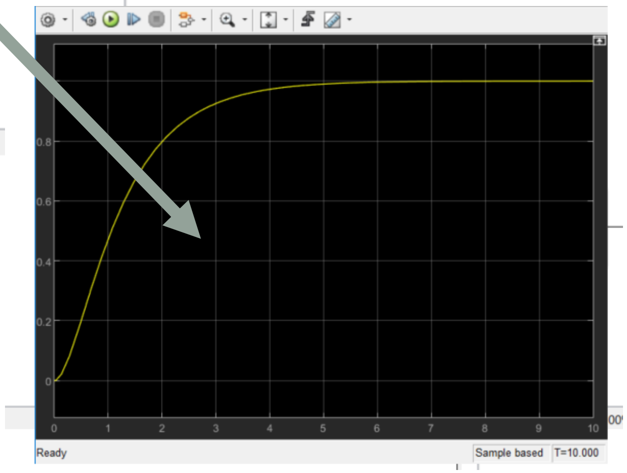
- From the ‘Continous’ module choose ‘Transfer Fcn’ (drag & drop)
- From the ‘Commonly Used Blocks’ choose ‘Scope’
- Merge the blocks with arrows
- Change the ‘Transfer Fcn’ (through double click)
- Optional: change the name of the blocks (double click on names)

Implementation in Simulink

Run simulation

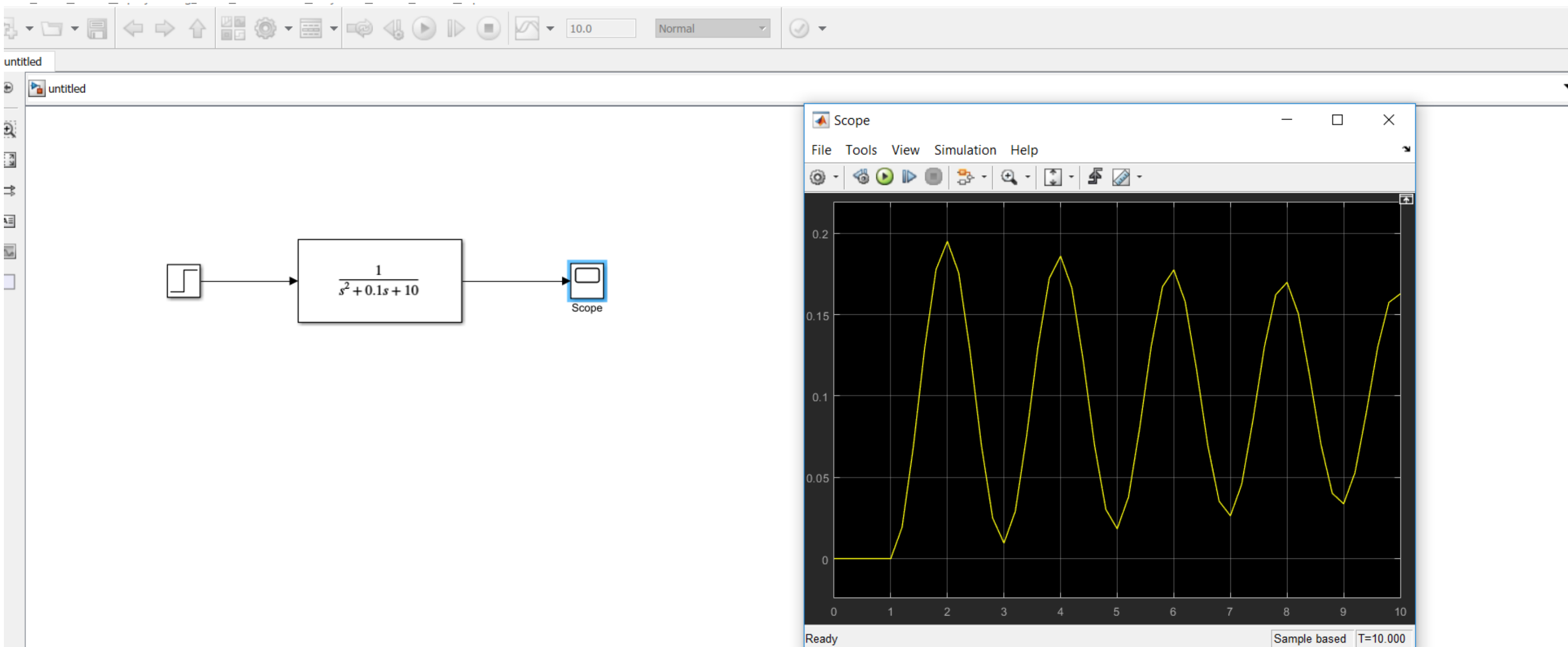


Double click on 'Scope'



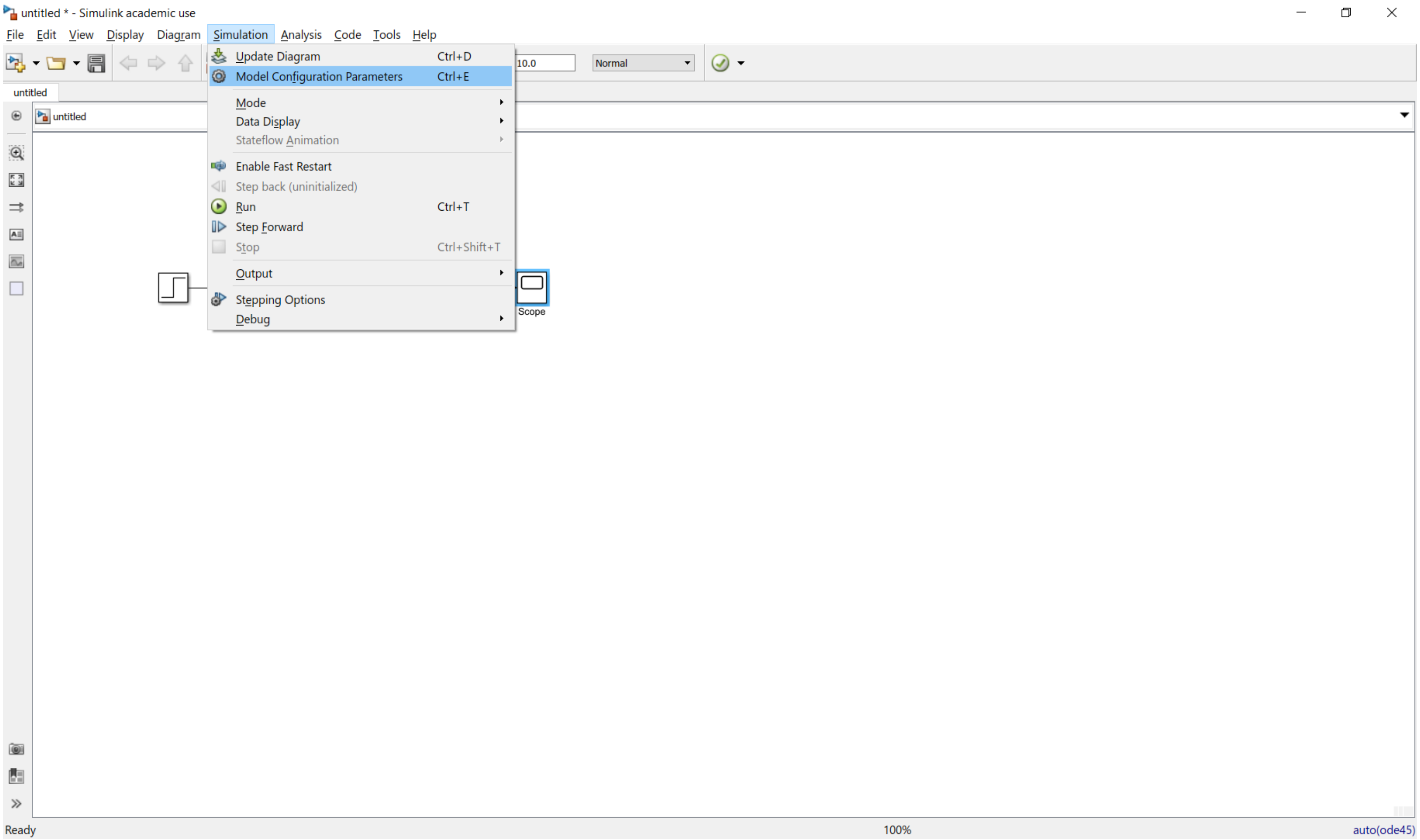
Implementation in Simulink

- Change the ‘Transfer Fcn’ to $s^2 + 0.1s + 10 = 0$ and run simulation

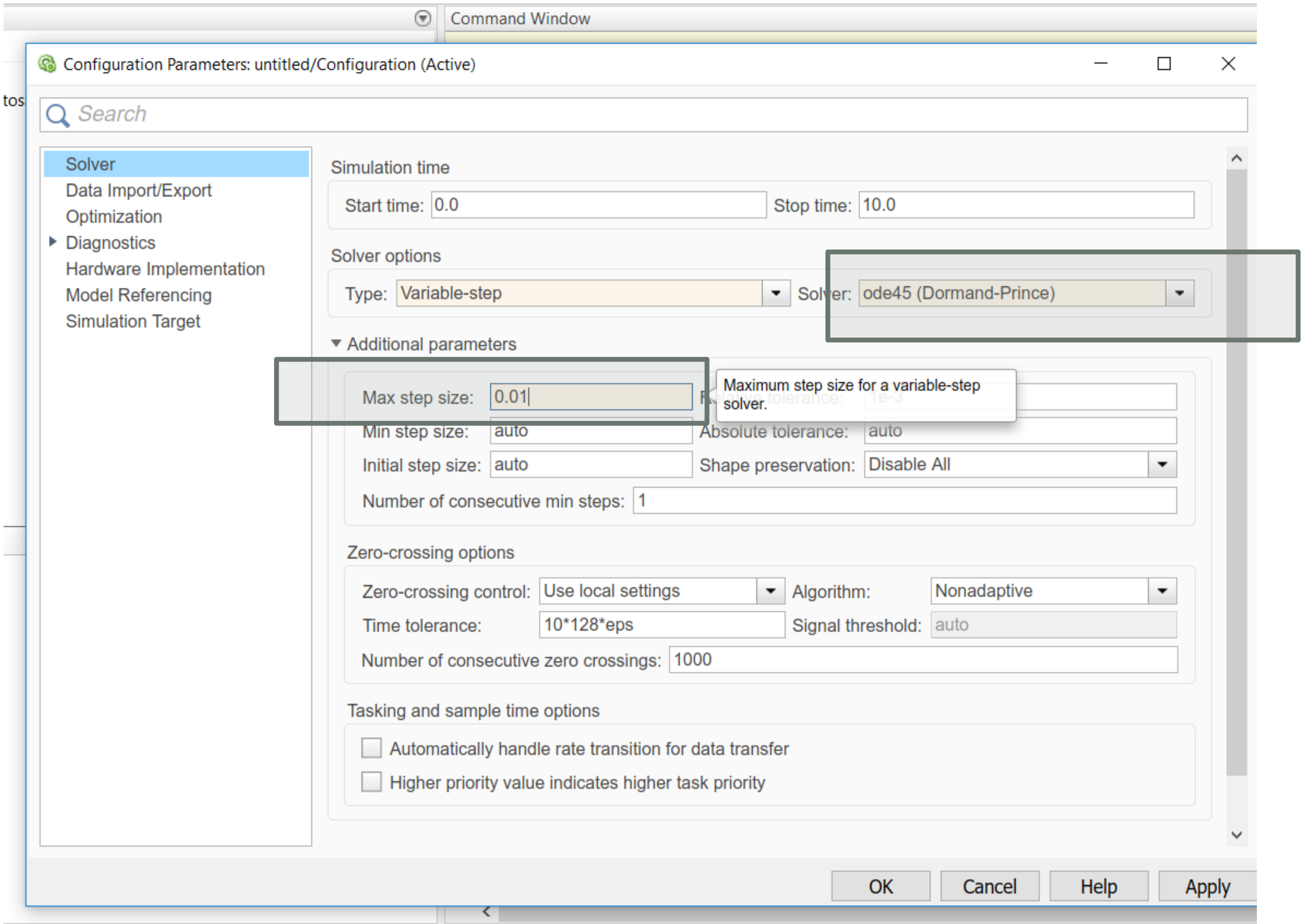


Accuracy of the solution depends on
- Solver (numerical integration algorithm)
- Step size

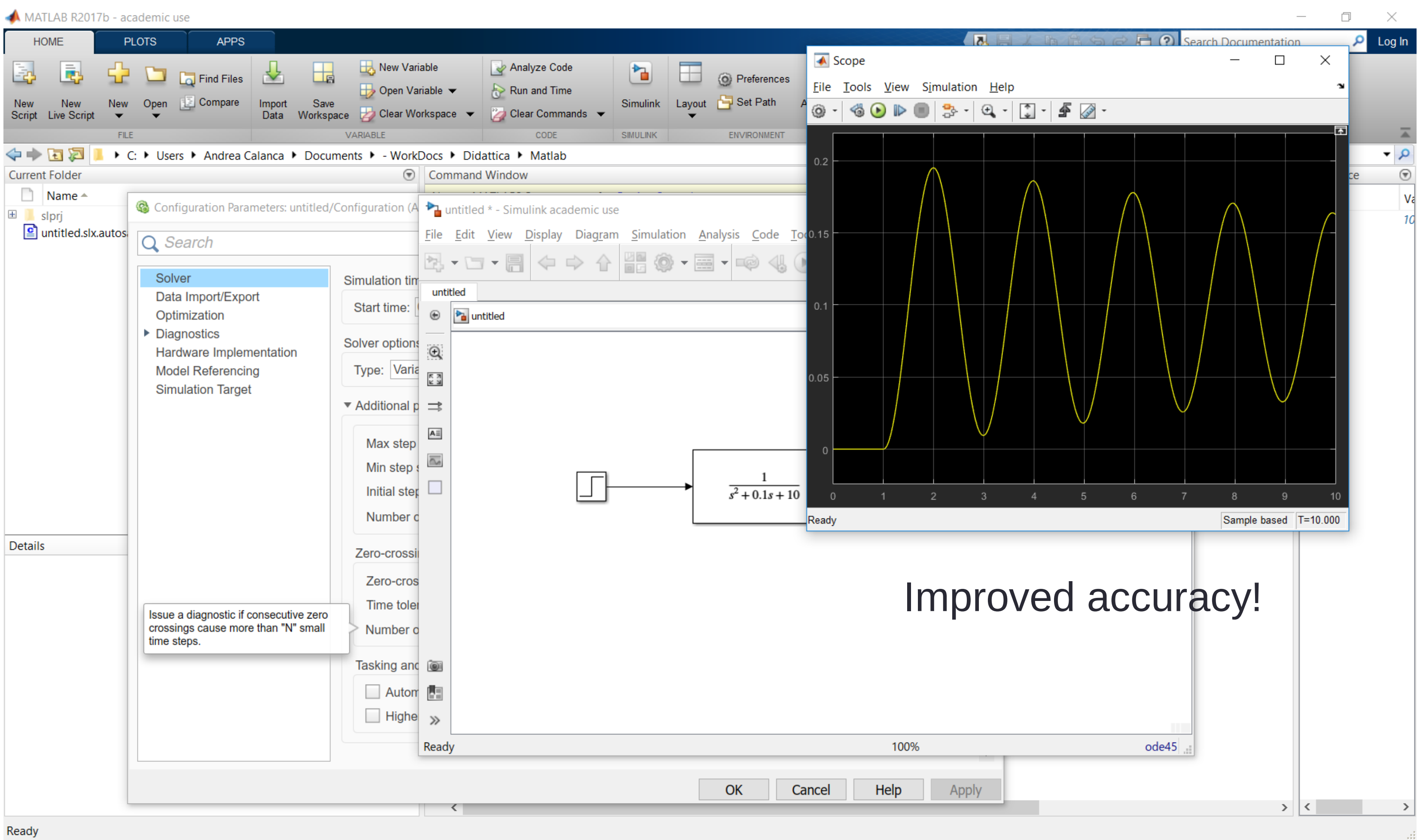
Implementation in Simulink



Implementation in Simulink



Implementation in Simulink



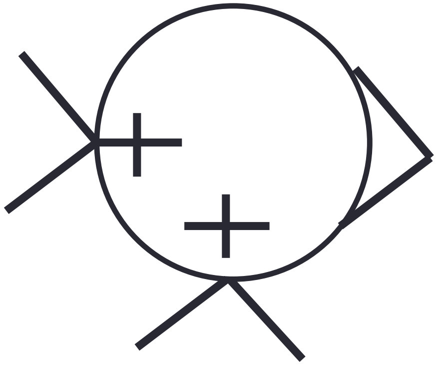
Improved accuracy!

Implementation: second method

- Simulate the same system using integrator blocks and sum by rewriting (B.4):

$$\frac{d^2v_c}{dt^2} = -4\frac{dv_c}{dt} - 3v_c - 3\,u_0(t) \qquad \text{(B.7)}$$

- The right side of (B.7) is the sum that yields $\frac{d^2v_c}{dt^2}$!
- Insert a ‘Sum’ block from the ‘Math Operations’ library (put 3 inputs)

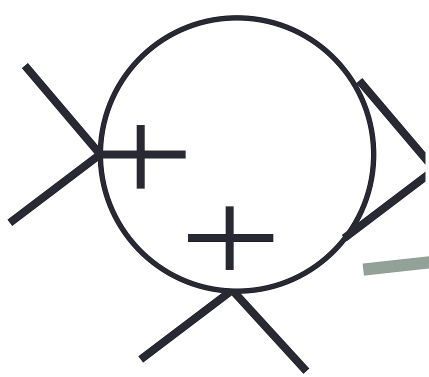


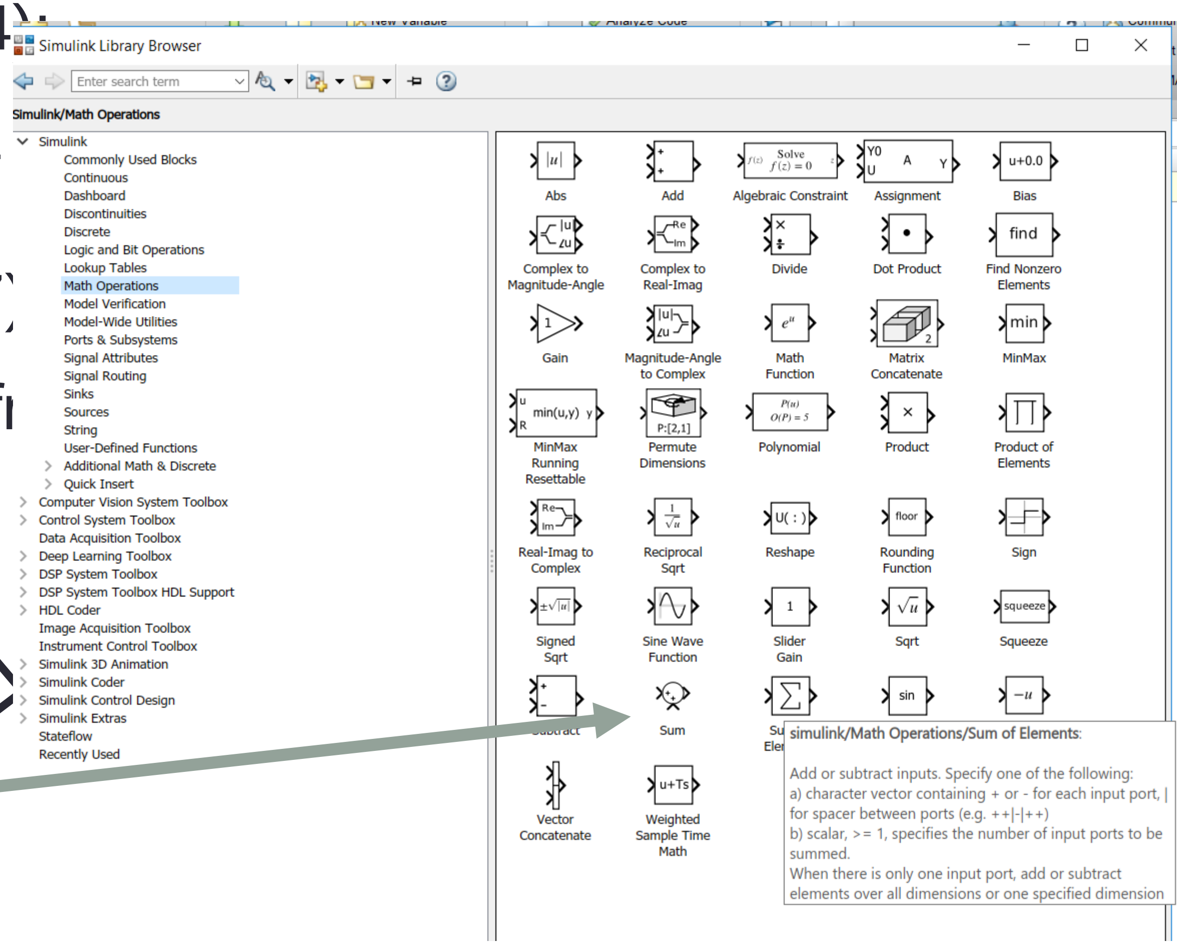
Implementation: second method

- Simulate the same system using integrator blocks and sum by rewriting (B.4

$$\frac{d^2 v_c}{dt^2} = -4\frac{dv_c}{dt}$$

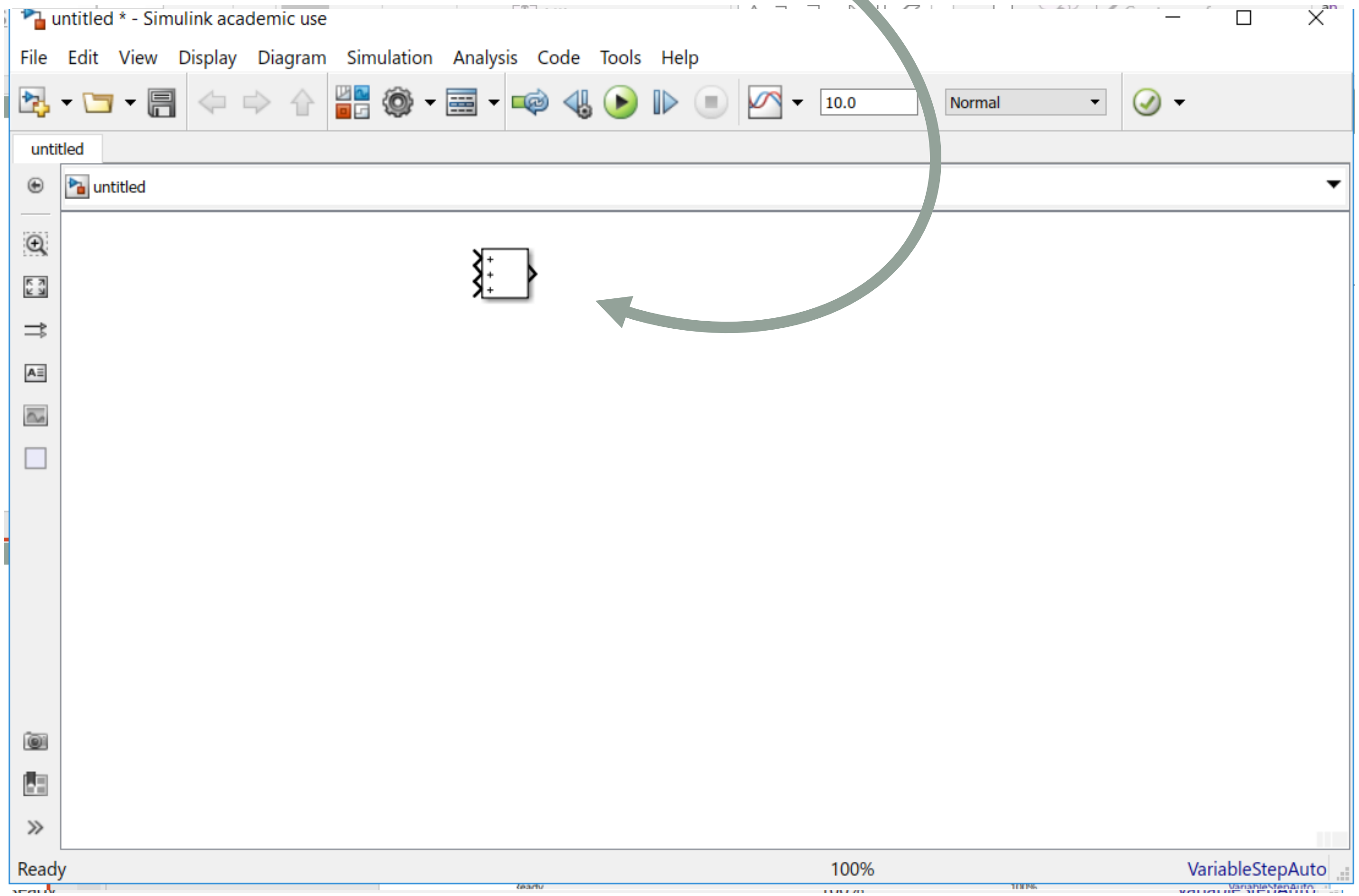
- The right side of (B.7
- Insert a ‘Sum’ block fr
3 inputs)





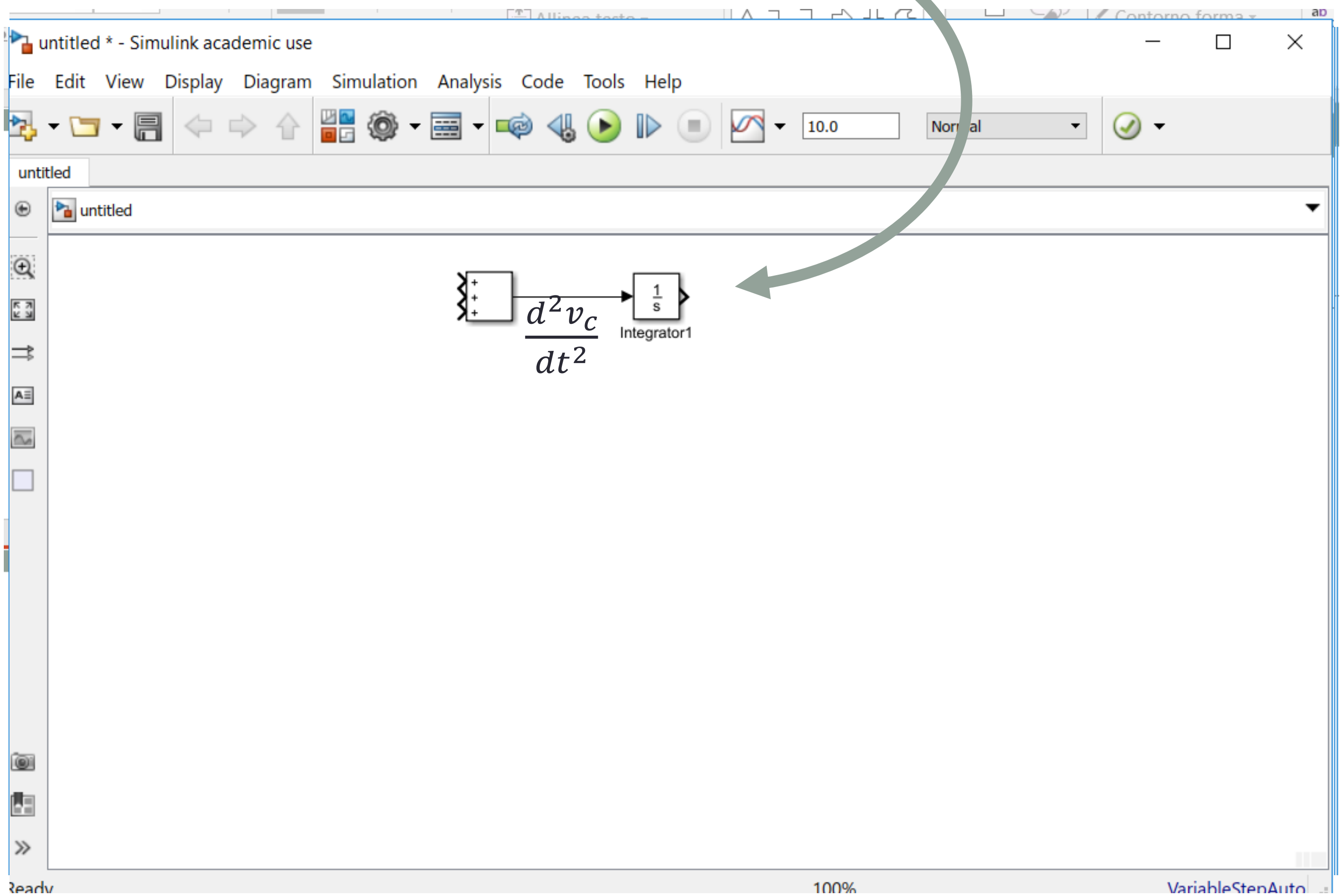
Implementation: second method

- Rember: the sum outputs $\frac{d^2 v_C}{dt^2}$



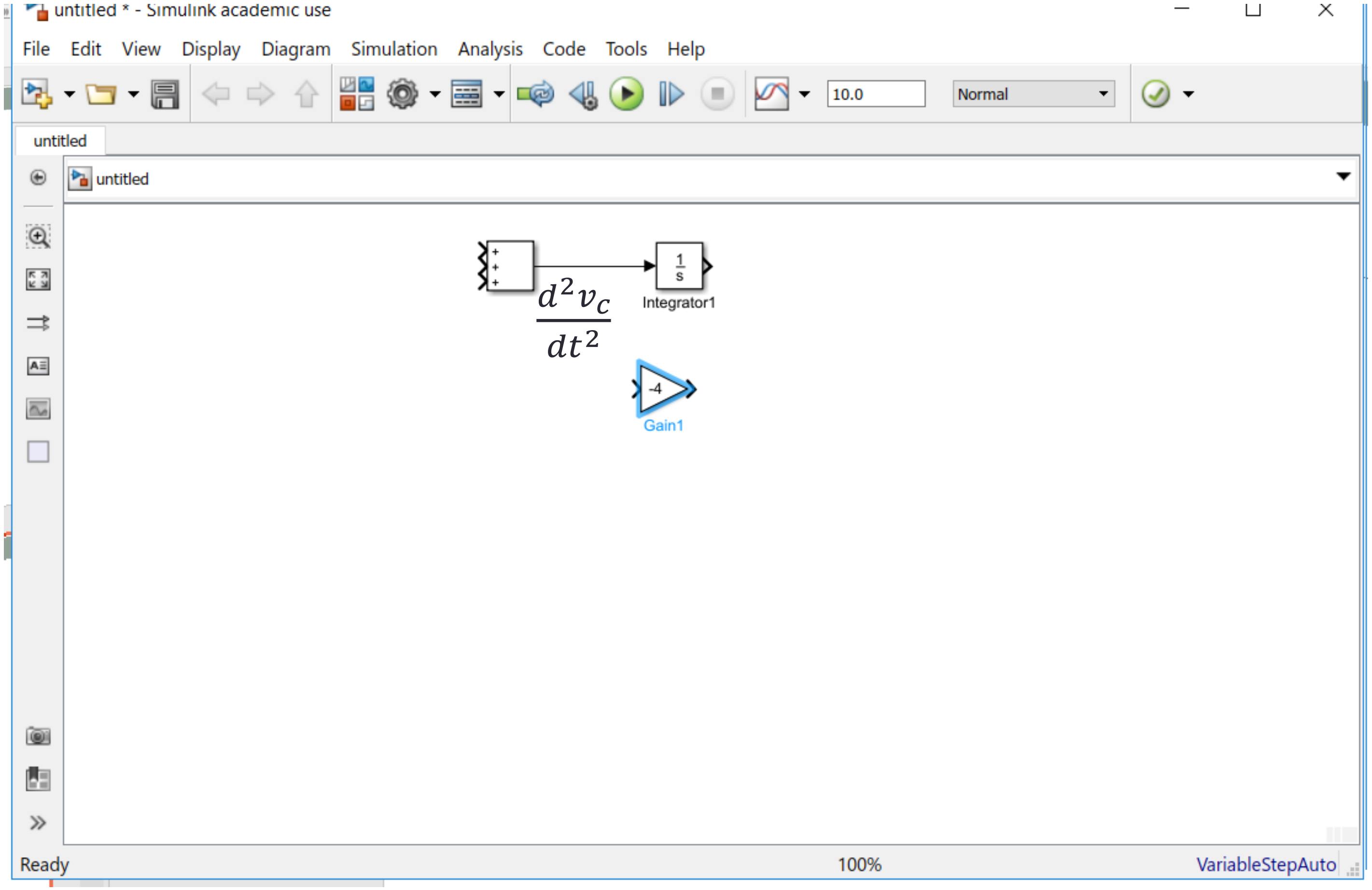
Implementation: second method

- Integrating $\frac{d^2 v_C}{dt^2}$ we obtain $\frac{dv_C}{dt}$



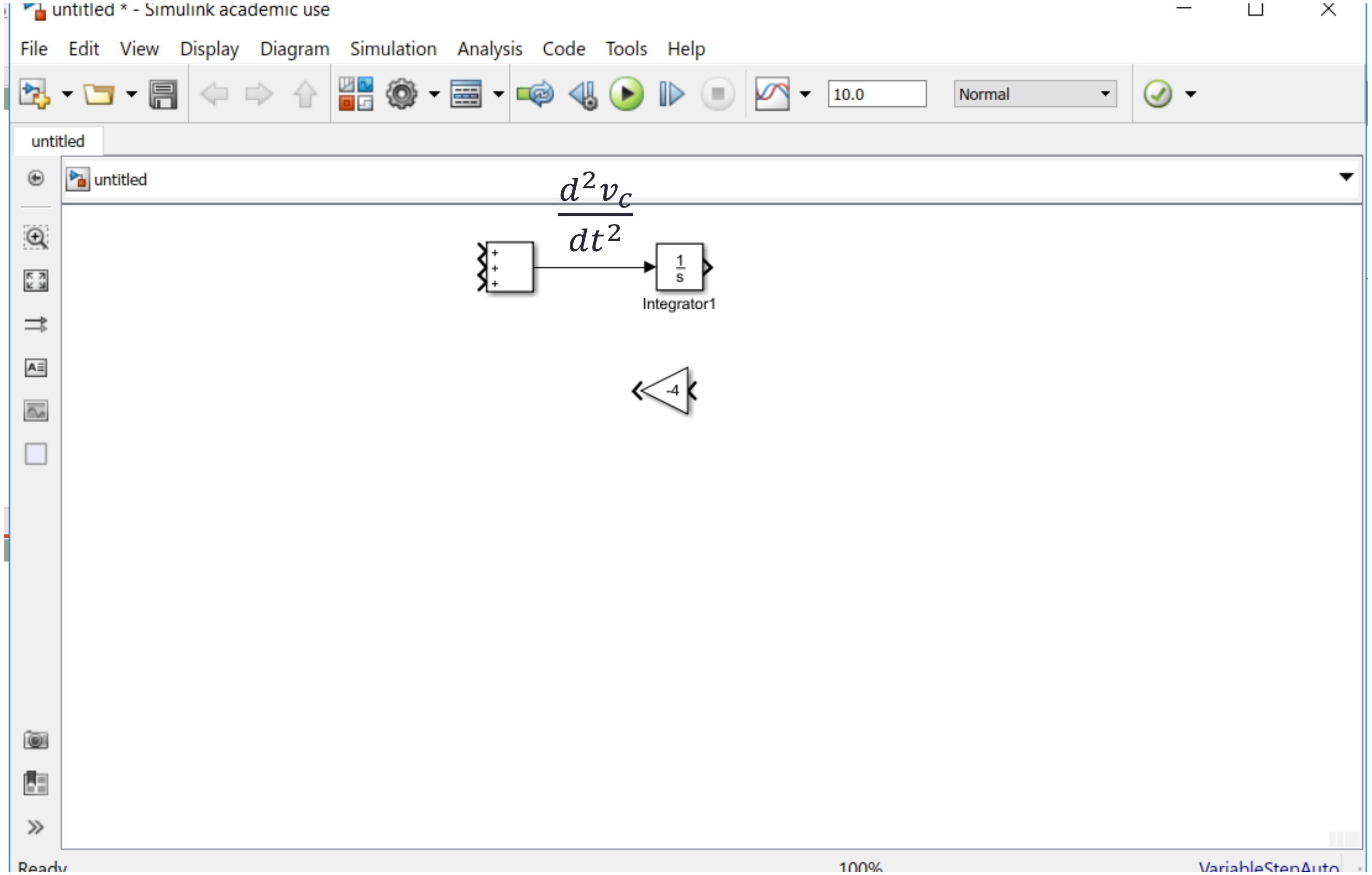
Implementation: second method

- Add a multiply block ‘Gain’ and set it to -4



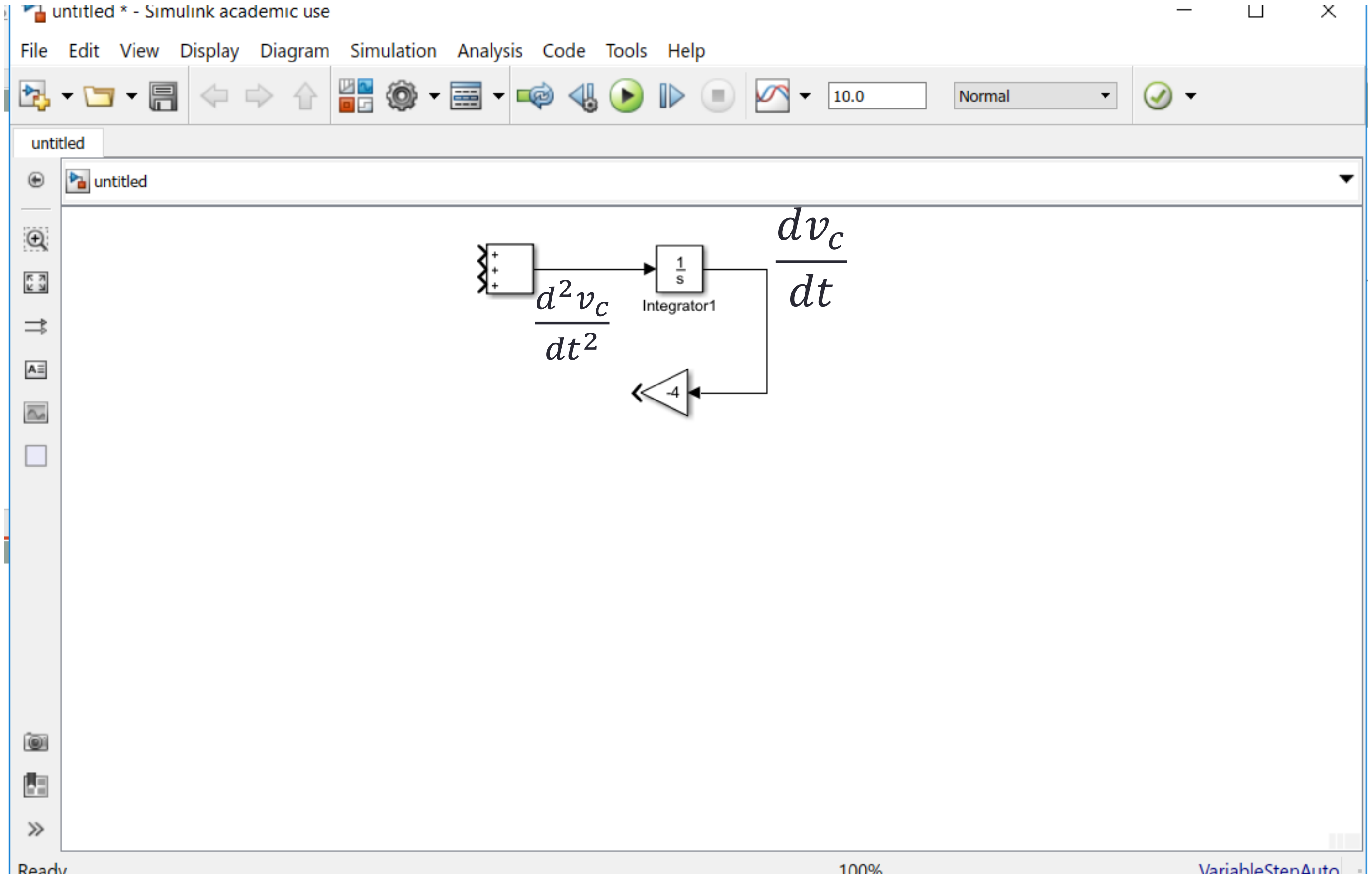
Implementation: second method

- Invert the 'Gain' (right click on 'Gain', then 'Flip Block' or CTRL-I)



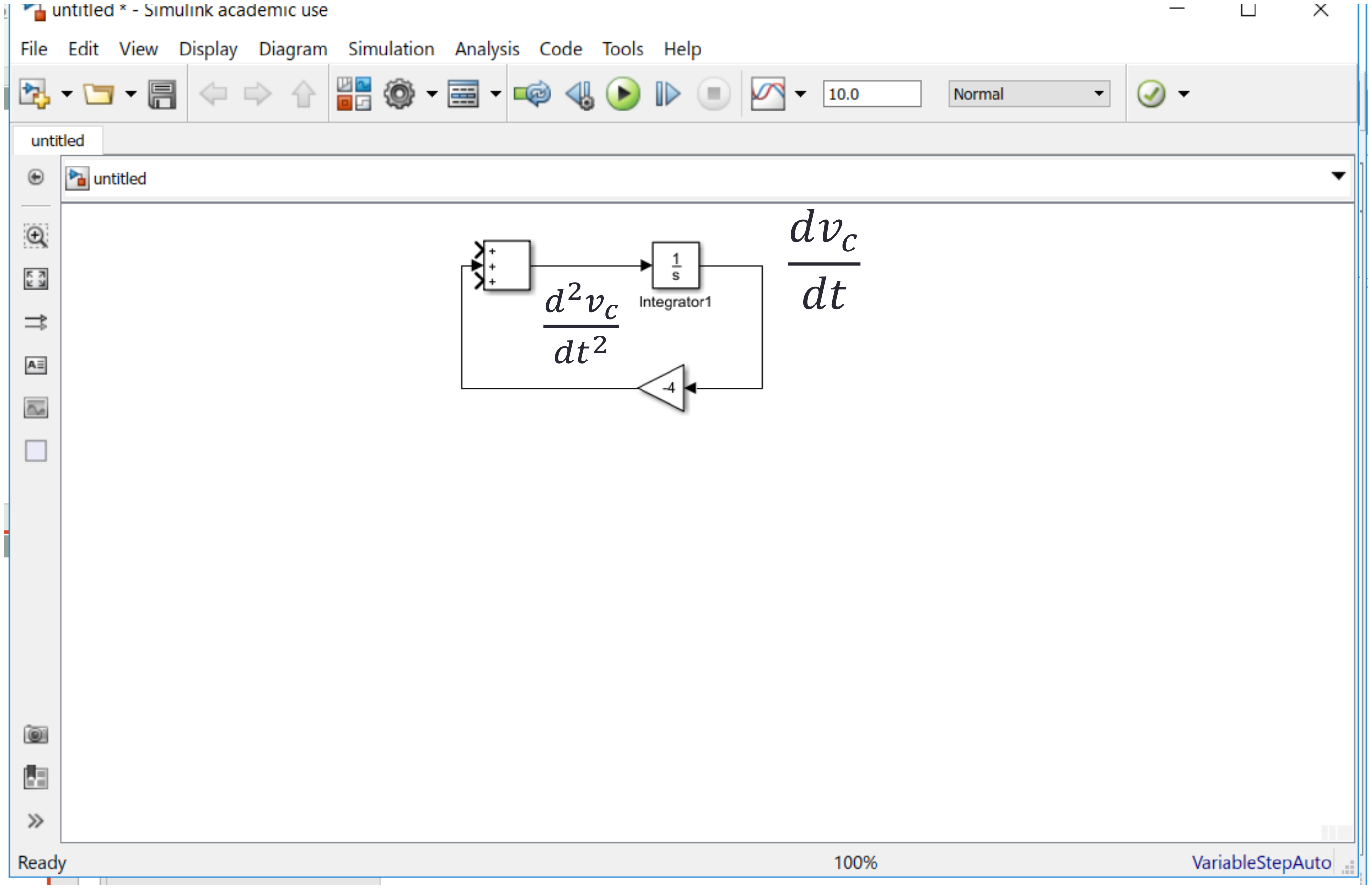
Implementation: second method

- Connect with the output of the integrator



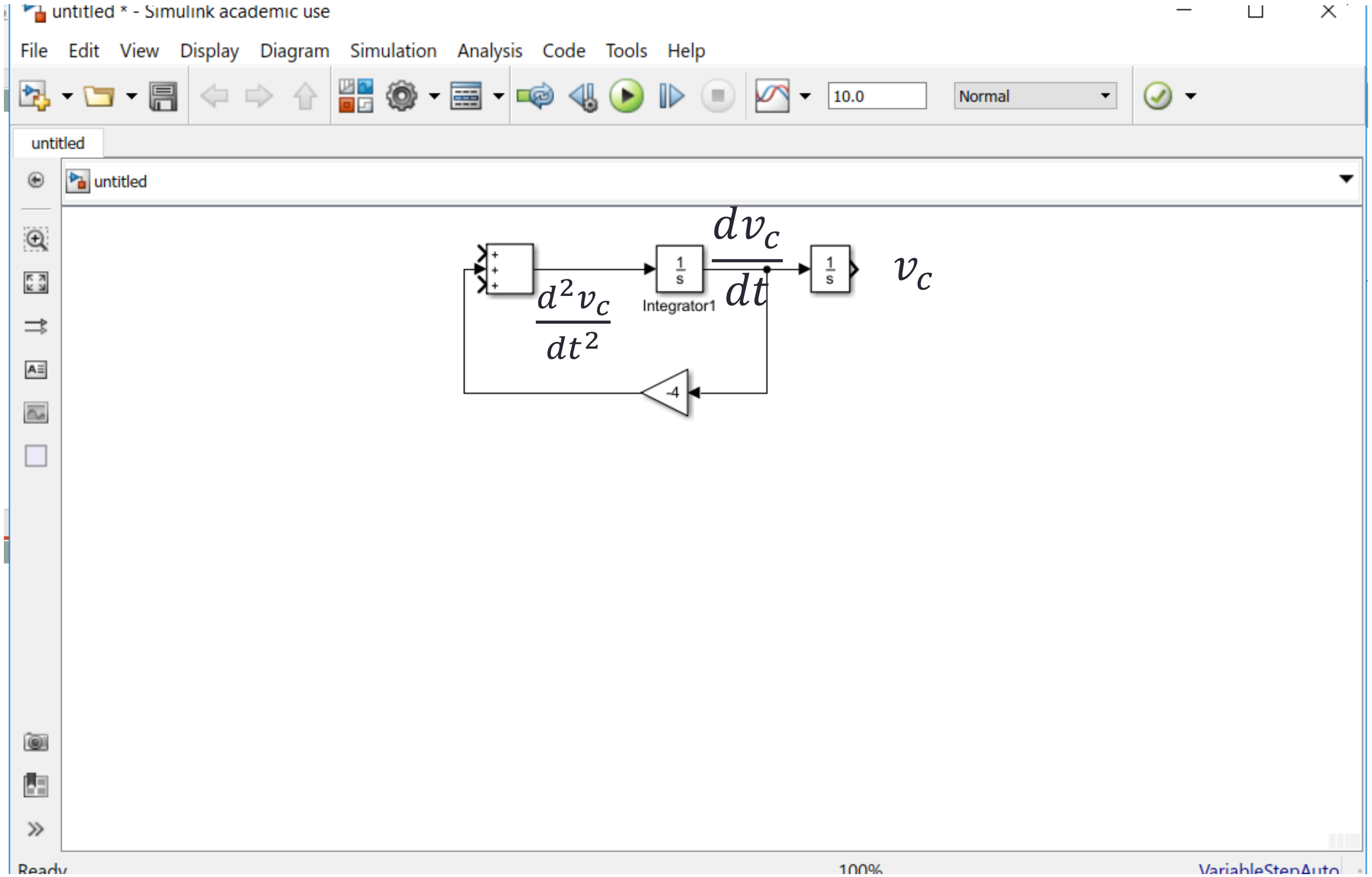
Implementation: second method

- Connect to the sum



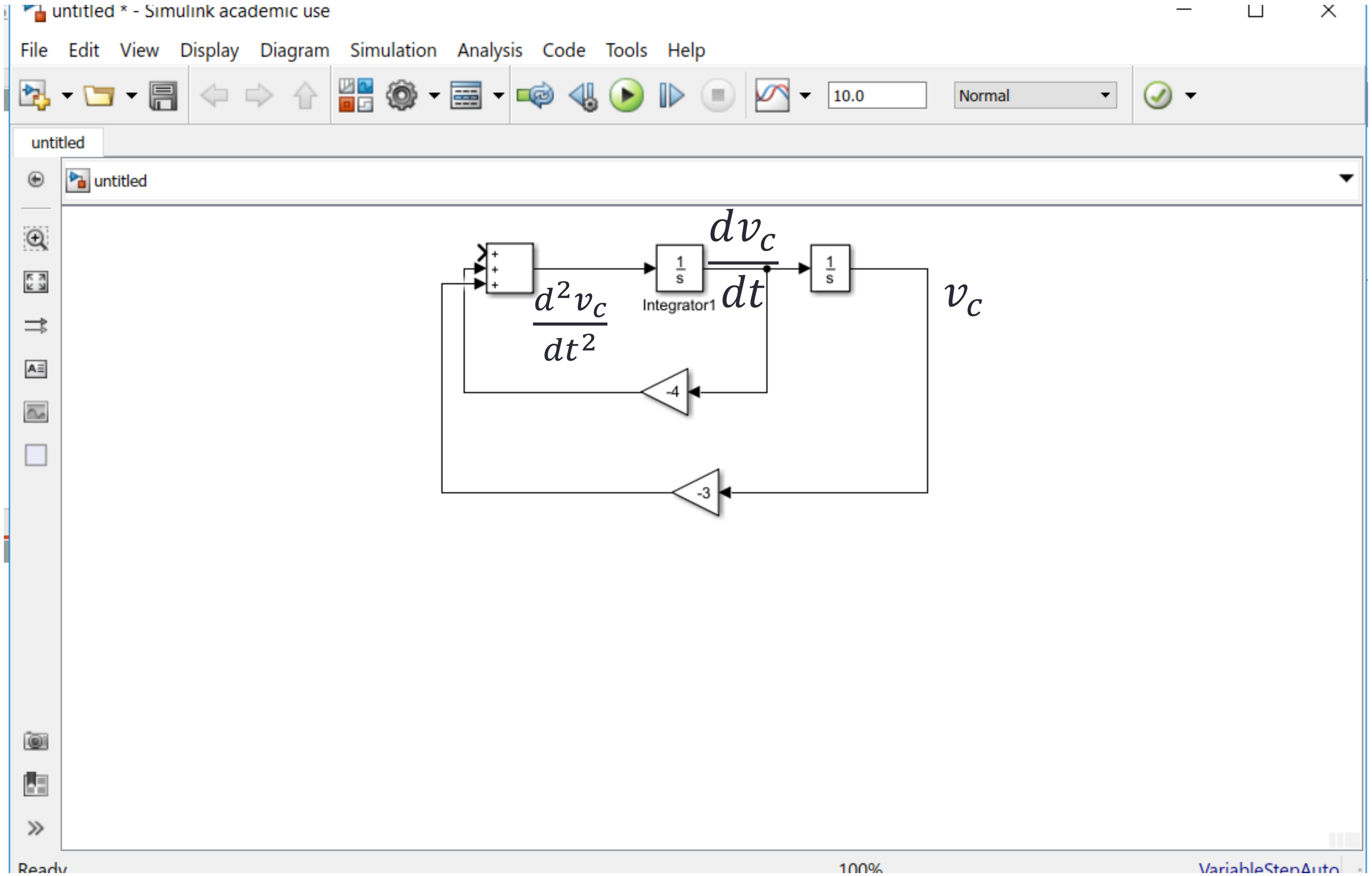
Implementation: second method

- Integrate $\frac{dv_c}{dt}$ to compute v_c



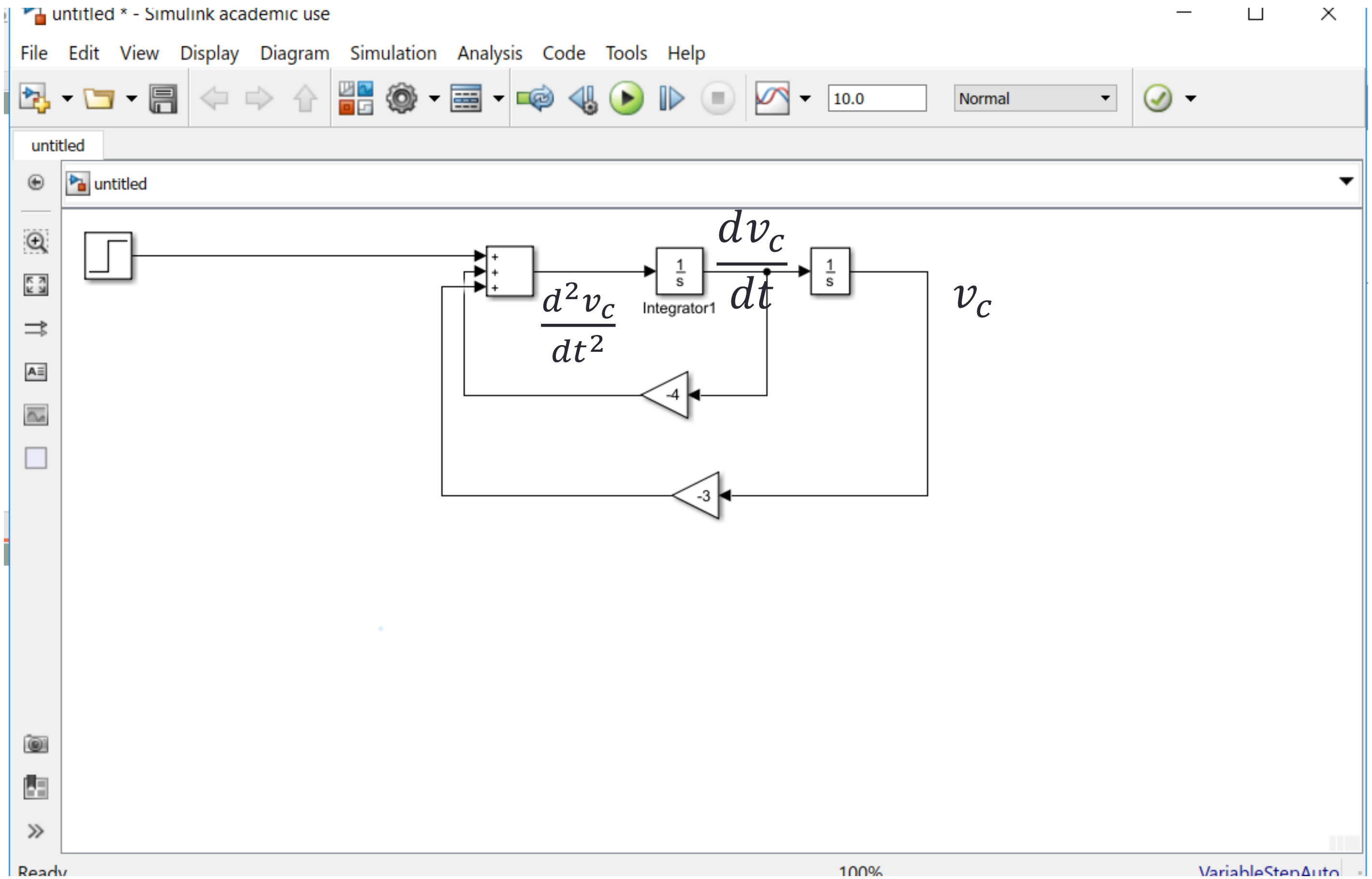
Implementation: second method

- Multiply v_c by -3 and insert into the sum



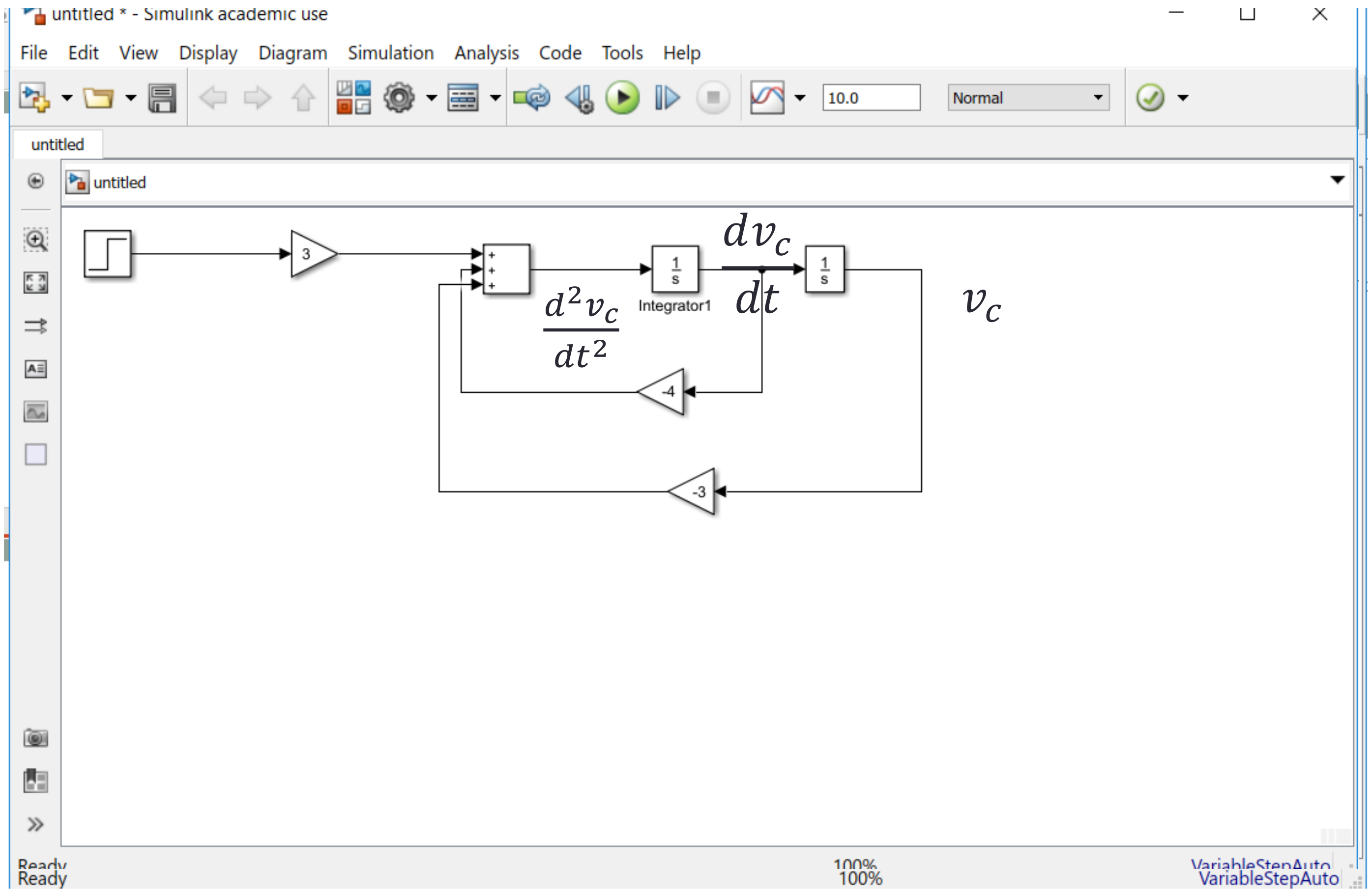
Implementation: second method

- Insert the input (step function)



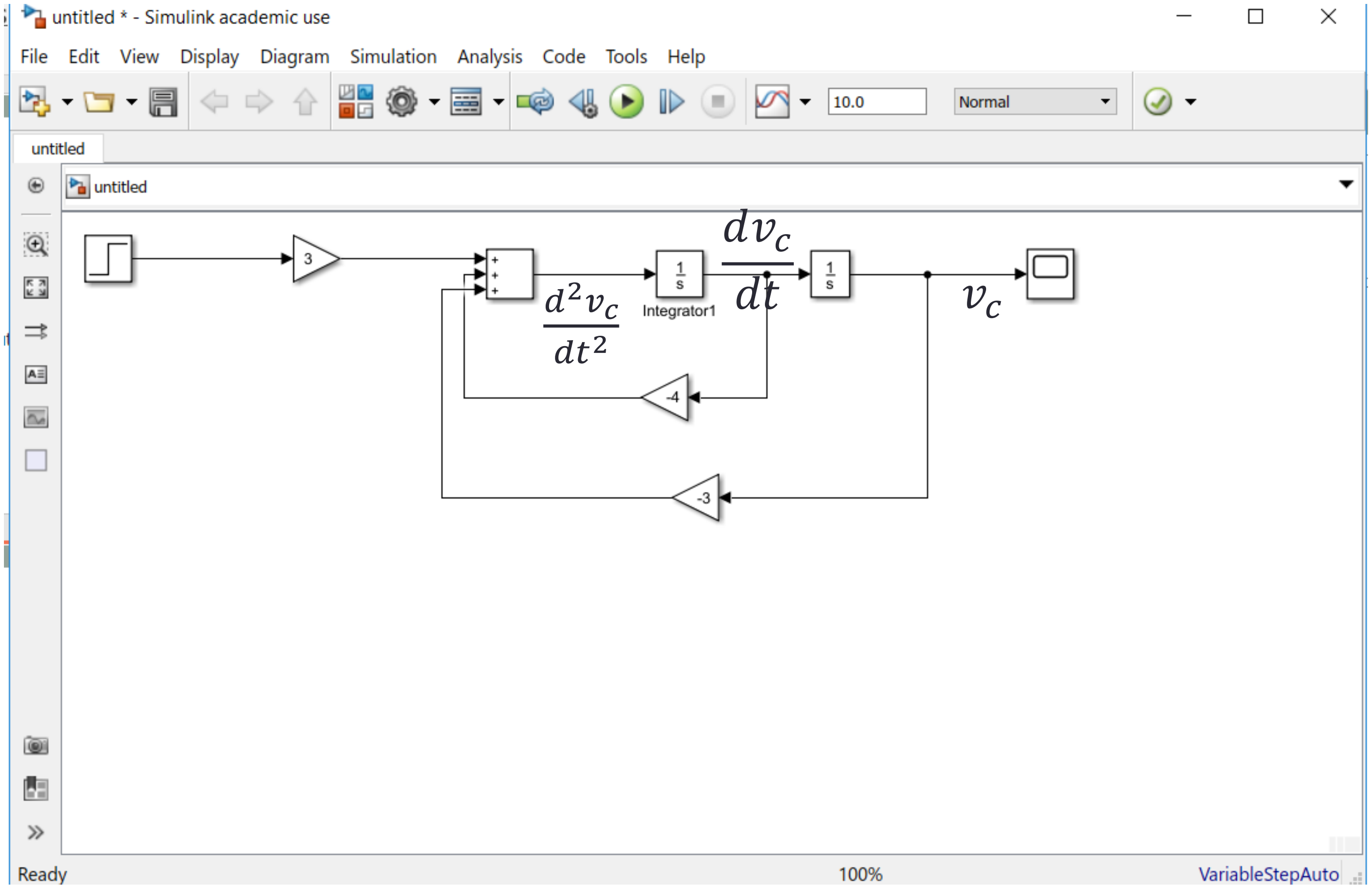
Implementation: second method

- Multiply the input by 3



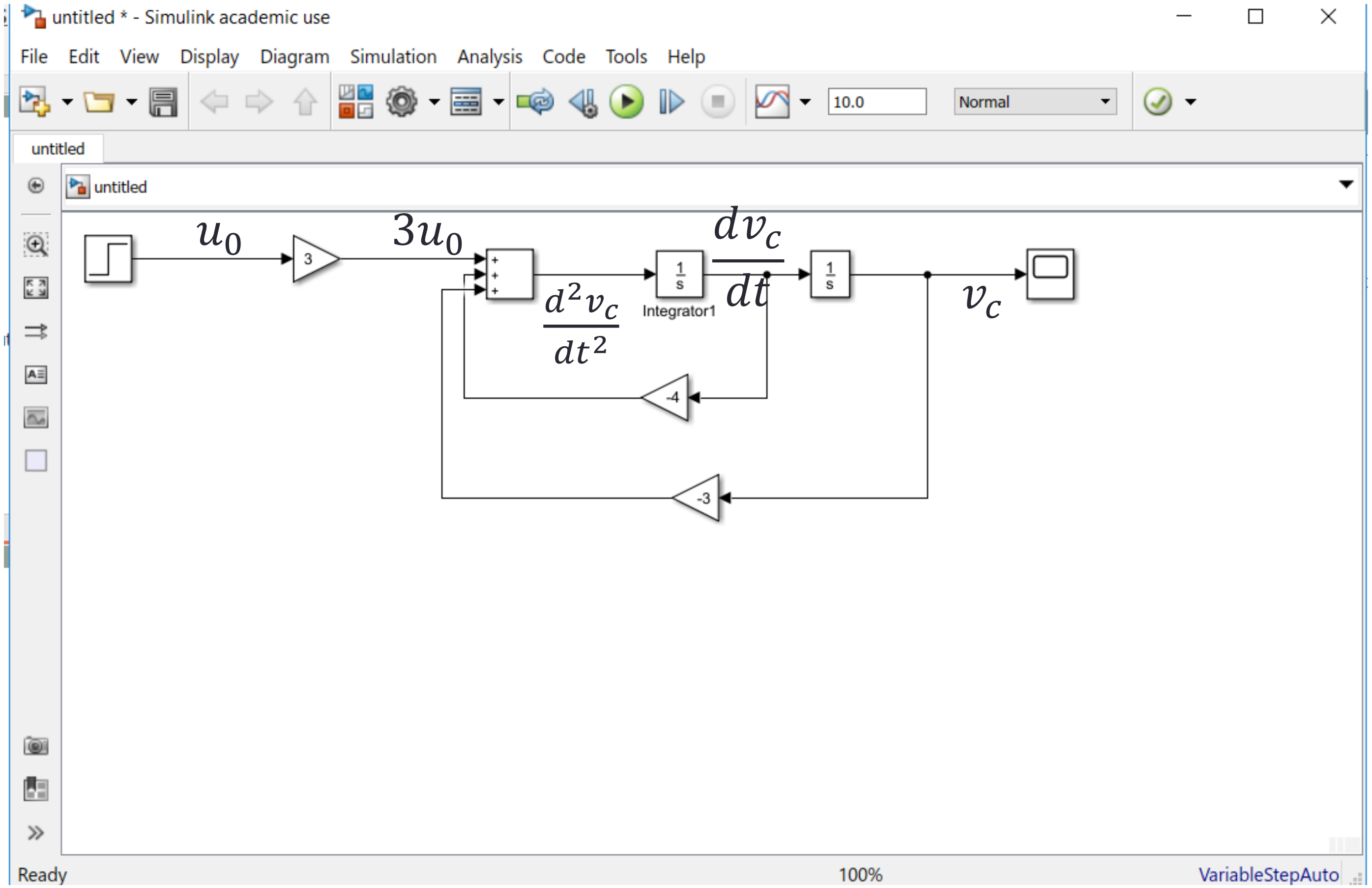
Implementation: second method

- Insert the 'Scope' block



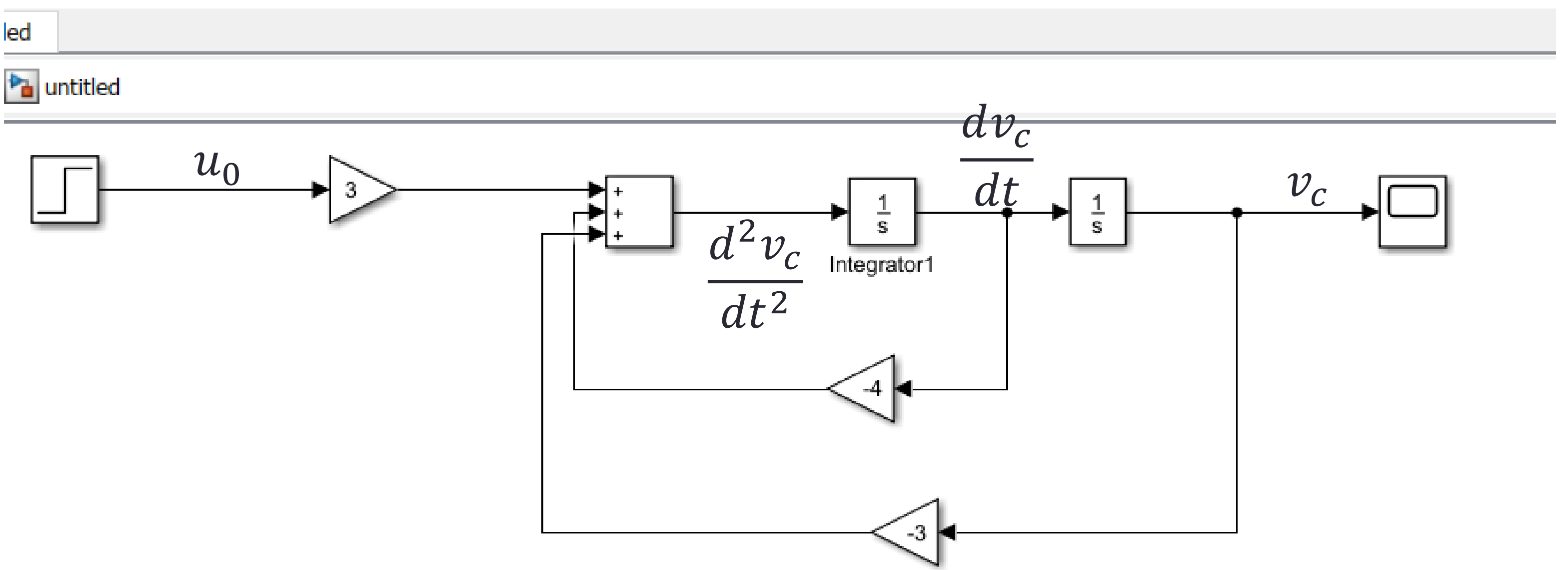
Implementation: second method

- Now all the signals are inserted in the block diagram:



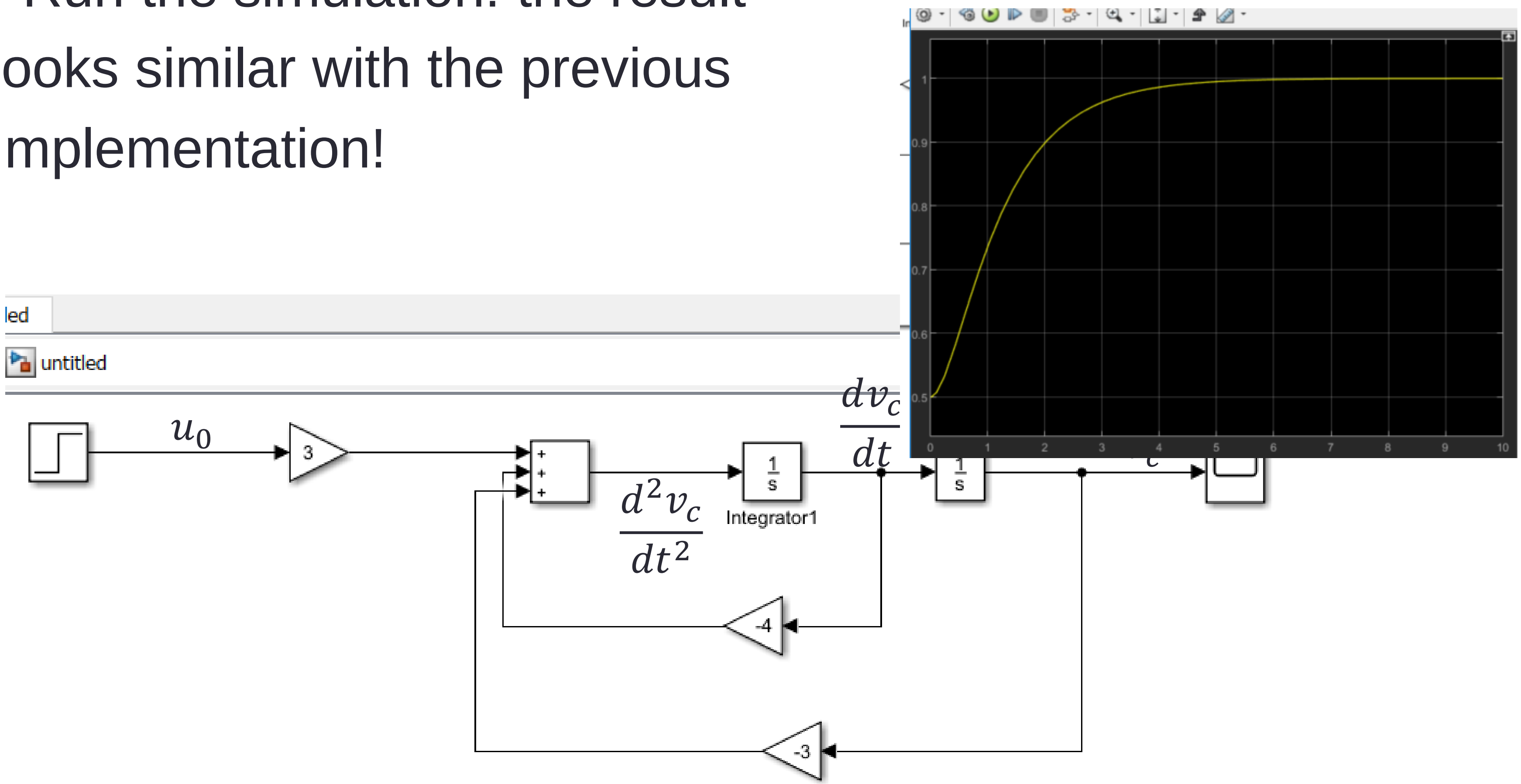
Implementation: second method

- The initial conditions are inserted by double clicking the Integrator blocks and entering the values 0 for the first integrator, and 0.5 for the second integrator.



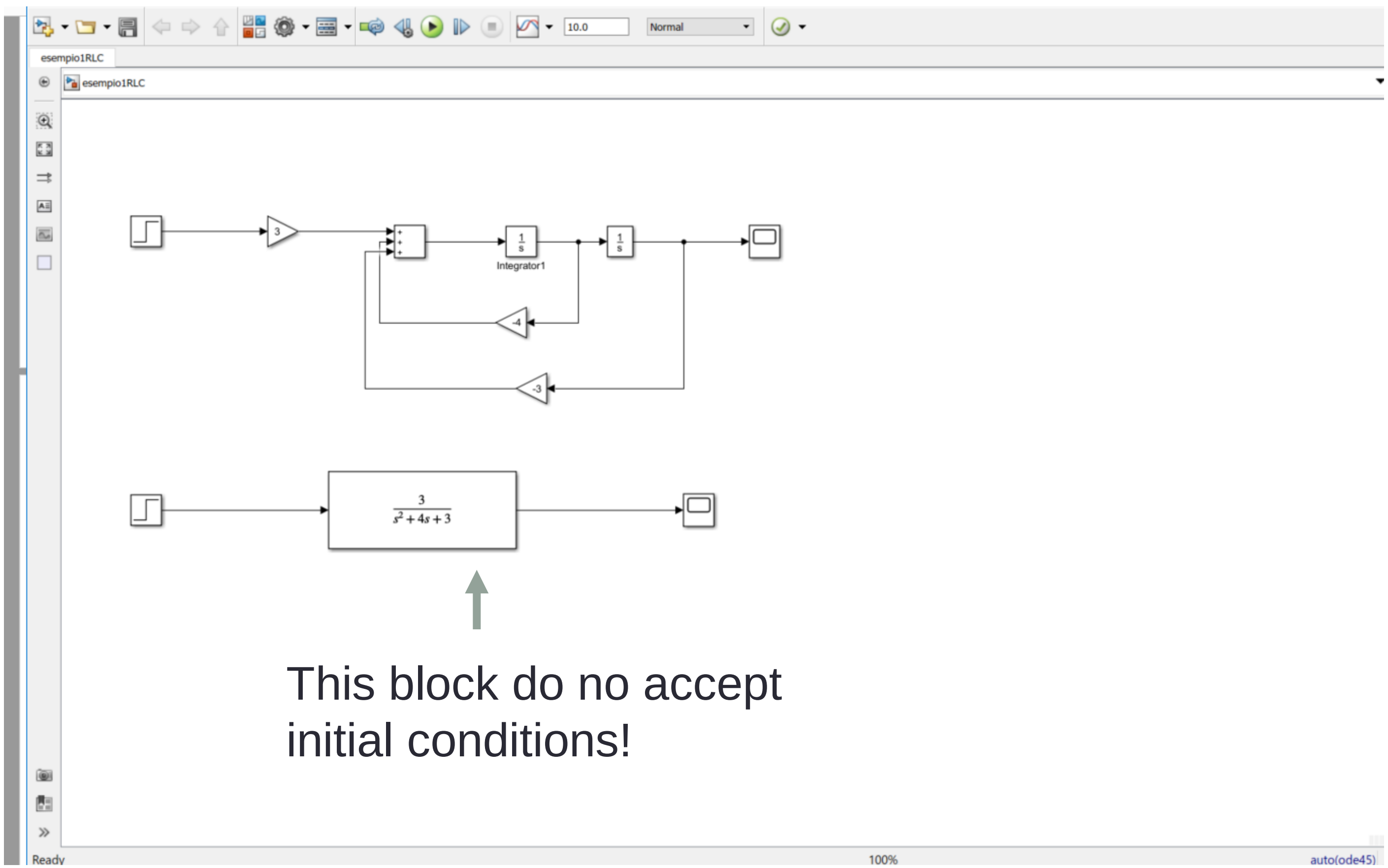
Implementation: second method

- Run the simulation: the result looks similar with the previous Implementation!



Implementation: putting all together

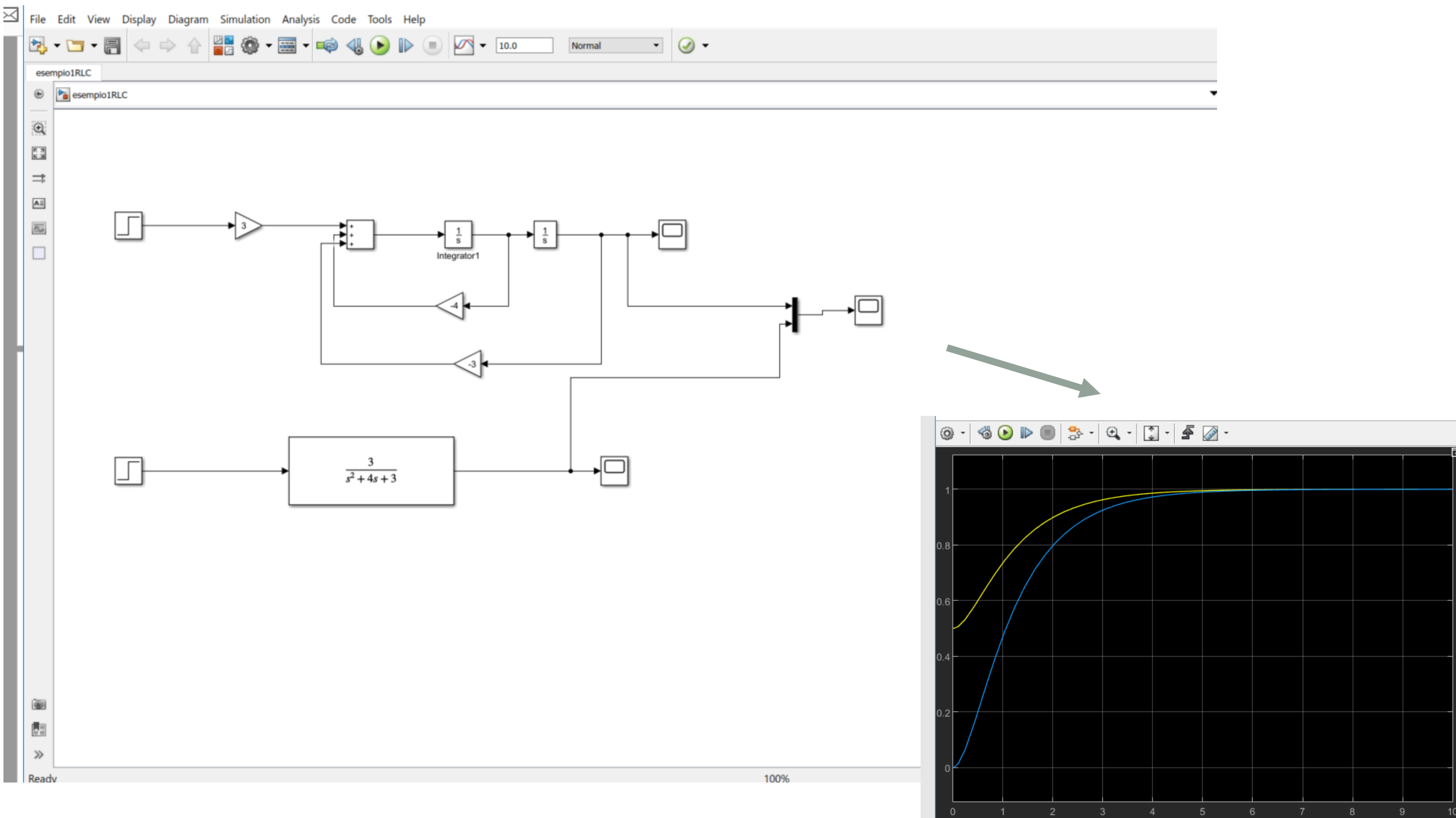
- Copy and paste the first diagram in the same window:



This block do no accept initial conditions!

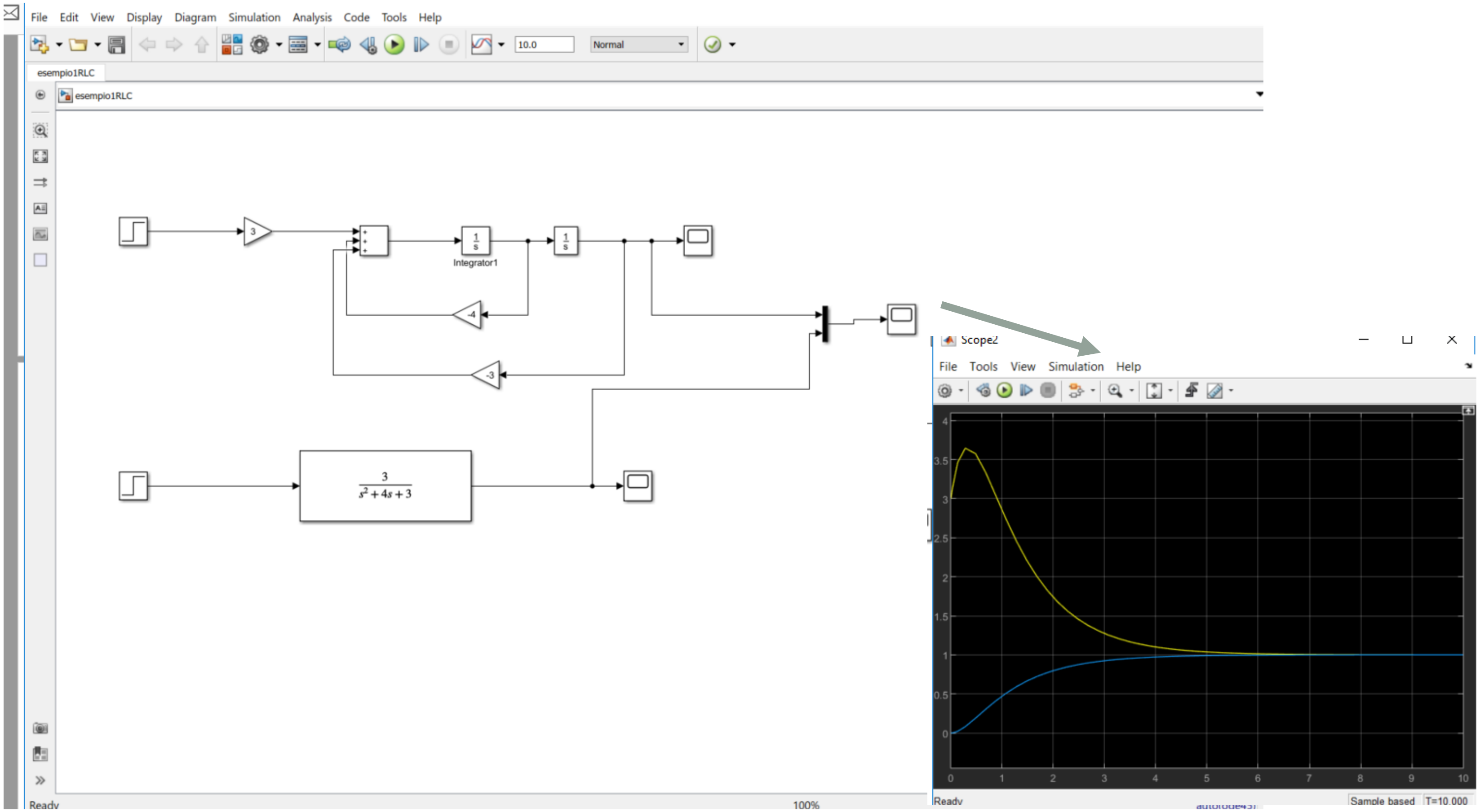
Implementation: putting all together

- Add a ‘Mux’ with two inputs and a ‘Scope’.



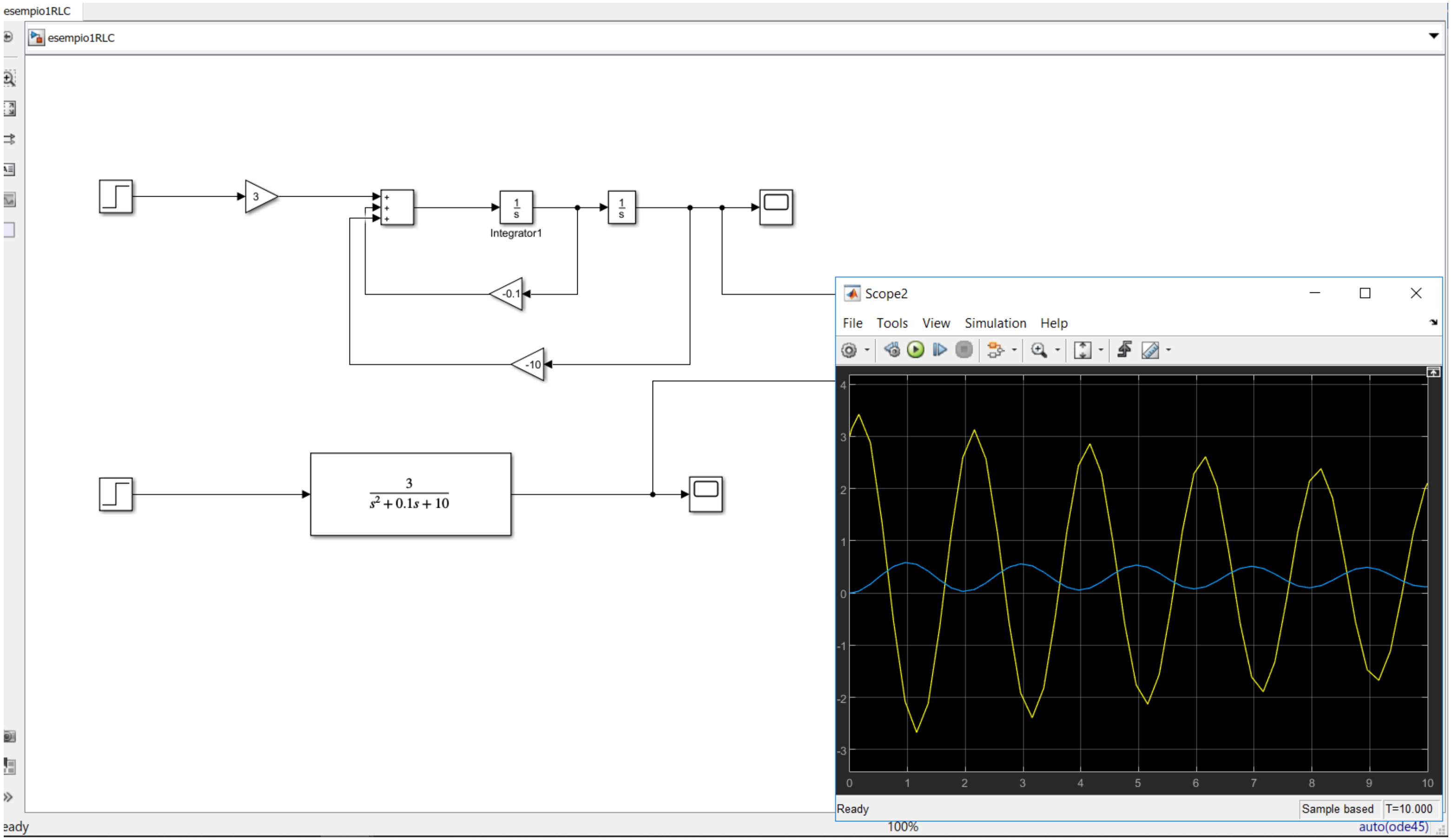
Implementation: putting all together

- Change the initial conditions and see what happens. How do you explain it?



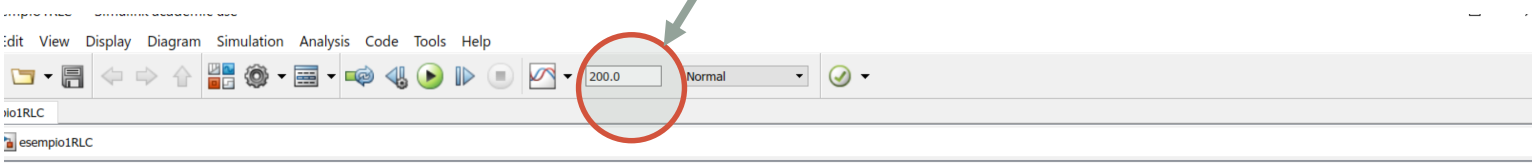
Implementation: putting all together

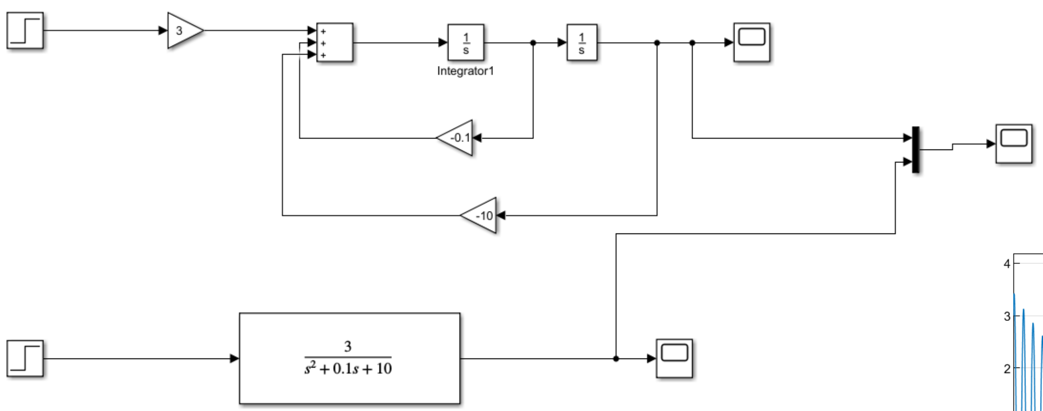
- Change the equations, the solver and the step size: analyze the results.

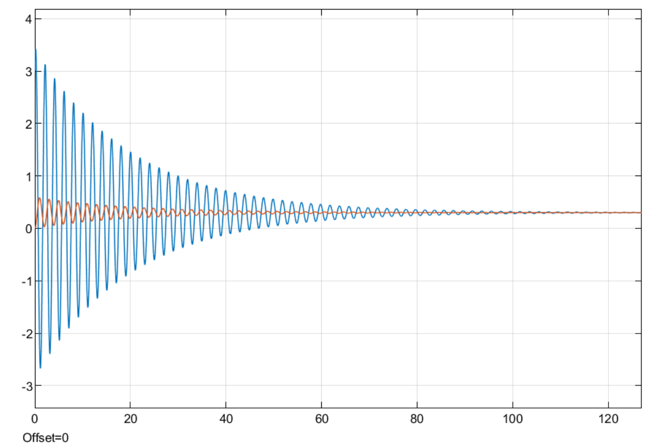


Implementation: putting all together

- Change the simulation time…







Exercise 1

- Following the same steps, simulate the following dynamic system:

$$\frac{dx}{dt} + 2x(t) = u(t)$$

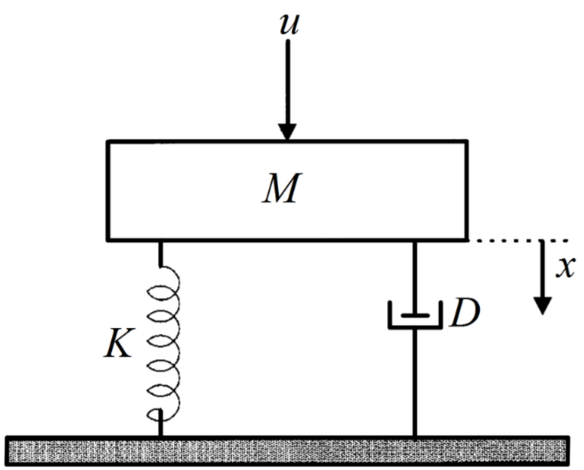
where $u(t)$ is a square wave with an amplitude of 1 and a frequency of 1 rad/sec and $x(t)$ is the output of the system.

To generate a Square Wave use a 'Signal Generator' block and select the Square Wave form but change the default units to radiants/sec.

- Method 1: use a single block for the transfer function
- Method 2: use an 'Integrator' block, a 'Gain' block and a 'Sum' block.
- What do you note in the absence of initial conditions?
- Hint: type 'Model a Continuous System' in the Help search of Matlab.

Exercise 2

- For the following mass-spring-damper system, the inputs are:



- M = 1Kg (mass)
- u = the gravity force Mg (g = 9.8)
- K = 5.0 N/m (spring constant)
- D = 0.05 Ns/m (damping coefficient)
- x(0) = 10 (initial position)
- x'(0) = 0 (initial velocity)

- Find the equation of motion x(t) and perform the simulation with Simulink.
- Optional: use a Matlab file parameters.m to set (all) the input parameters.
- Hints:
  1. for the input u(t) use the step function and multiply it by 9.8.
  2. The equation is:

$$\frac{d^2x}{dt^2} = -\frac{D}{M}\frac{dx}{dt} - \frac{K}{M}x + g$$