

OPEN SOURCE ENGINEERING



A SCILAB PROFESSIONAL PARTNER



MODELING IN SCILAB: PAY ATTENTION TO THE RIGHT APPROACH – PART 2

In this tutorial we show how to model a physical system described by ODE using Xcos environment. The same model solution is also described in Scilab and Xcos + Modelica in two other tutorials.

Level



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.



www.openeering.com

Step 1: The purpose of this tutorial

In Scilab there are *three different approaches* (see figure) for modeling a physical system which is described by Ordinary Differential Equations (ODE).

For showing all these capabilities we selected a *common physical system*, the *LHY model* for drug abuse. This model is used in our tutorials as a common problem to show the main features of each strategy. We are going to recurrently refer to this problem to allow the reader to better focus on the Scilab approach rather than on mathematical details.

In this second tutorial we show, step by step, how the LHY model problem can be implemented in the *Xcos environment*. The sample code can be downloaded from the Openeering web site.

1

Standard Scilab Programming

2

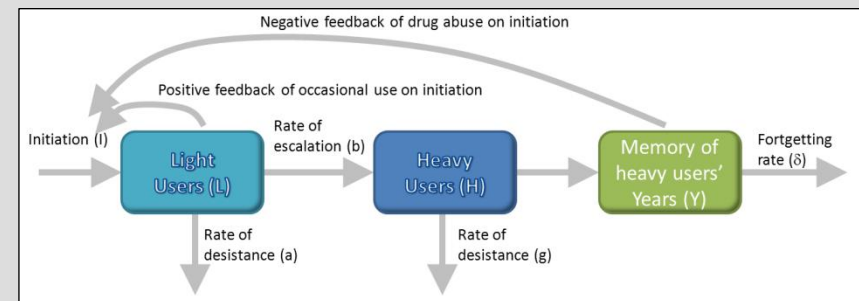
Xcos Programming

3

Xcos + Modelica

Step 2: Model description

The considered model is the LHY model used in the study of drug abuse. This model is a continuous-time dynamical system of drug demand for two different classes of users: *light users* (denoted by $L(t)$) and *heavy users* (denoted by $H(t)$) which are functions of time t . There is another state in the model that represents the decaying *memory of heavy users in the years* (denoted by $Y(t)$) that acts as a deterrent for new light users. In other words the increase of the deterrent power of memory of drug abuse reduces the contagious aspect of initiation. This approach presents a positive feedback which corresponds to the fact that light users promote initiation of new users and, moreover, it presents a negative feedback which corresponds to the fact that heavy users have a negative impact on initiation. Light users become heavy users at the *rate of escalation* b and leave this state at the *rate of desistance* a . The heavy users leave this state at the *rate of desistance* g .



Step 3: Mathematical model

The mathematical model is a system of **ODE (Ordinary Differential Equation)** in the **unknowns**:

- $L(t)$, **number of light users**;
- $H(t)$, **number of heavy users**;
- $Y(t)$, **decaying of heavy user years**.

The **initiation function** contains a “spontaneous” initiation τ and a memory effect modeled with a negative exponential as a function of the memory of year of drug abuse relative to the number of current light users.

The problem is completed with the specification of the **initial conditions** at the time t_0 .

The LHY equations system (omitting time variable t for sake of simplicity) is

$$\begin{cases} \dot{L} &= I(L, Y) - (a + b)L \\ \dot{H} &= bL - gH \\ \dot{Y} &= H - \delta Y \end{cases}$$

where the initiation function is

$$I(L, Y) = \tau + L \cdot \max \left\{ s_{\max}, s \cdot e^{-q \frac{Y}{L}} \right\}$$

The LHY initial conditions are

$$\begin{cases} L(t_0) &= L_0 \\ H(t_0) &= H_0 \\ Y(t_0) &= Y_0 \end{cases}$$

Step 4: Problem data

(Model data)

a : the annual rate at which light users quit

b : the annual rate at which light users escalate to heavy use

g : the annual rate at which heavy users quit

δ : the forgetting rate

(Initiation function)

τ : the number of innovators per year

s : the annual rate at which light users attract non-users

q : the constant which measures the deterrent effect of heavy use

s_{\max} : the maximum rate of generation for initiation

(Initial conditions)

t_0 : the initial simulation time;

L_0 : Light users at the initial time;

H_0 : Heavy users at the initial time;

Y_0 : Decaying heavy users at the initial time.

Model data

$$a = 0.163$$

$$b = 0.024$$

$$g = 0.062$$

$$\delta = 0.291$$

Initiation function

$$\tau = 50000$$

$$s = 0.610$$

$$q = 3.443$$

$$s_{\max} = 0.1$$

Initial conditions

$$t_0 = 1970$$

$$L_0 = 1.4 \times 10^6$$

$$H_0 = 0.13 \times 10^6$$

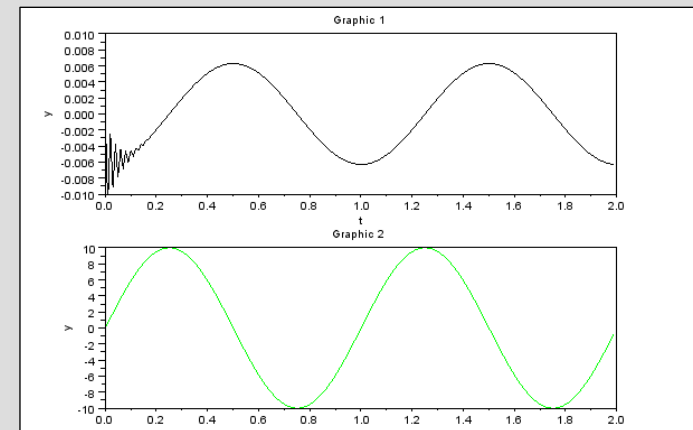
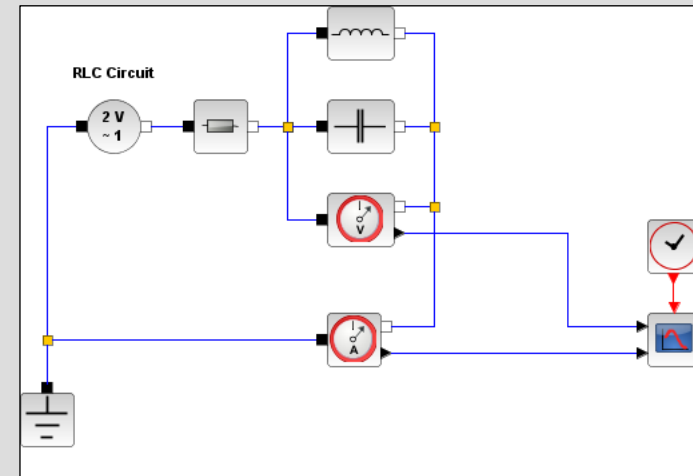
$$Y_0 = 0.11 \times 10^6$$

Step 5: Xcos programming – introduction

Xcos is a free graphical editor and simulator based on Scilab that helps people to model physical systems (electrical, mechanical, automotive, hydraulics, ...) using a graphical user interface based on a **block diagram approach**. It includes explicit and implicit dynamical systems and both continuous and discrete sub-systems.

This toolbox is particularly useful in control theory, digital and signal processing and model-based design for multidomain simulation, especially when continuous time and discrete time components are interconnected.

As an example of a Xcos diagram, we show on the right a Xcos model of a RLC electric circuit with its graphical output. The first output is relative to the voltage across the capacitor element, while the second one is relative to the current through the voltage generator.



Step 6: Xcos programming – getting started

Xcos environment can be started from Scilab Console typing

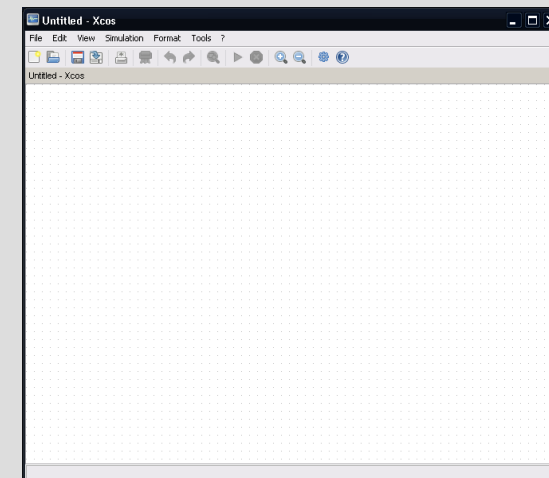
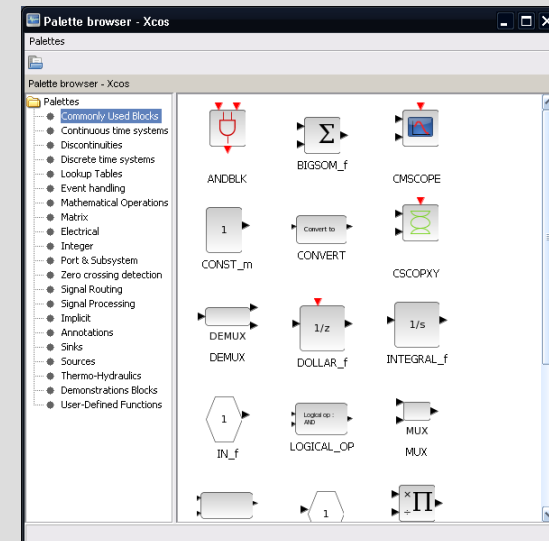
```
--> xcOS
```

or clicking on the button  in the Scilab menu bar.

The command starts two windows:

- the **palette browser** that contains all Xcos available blocks grouped by categories;
- an **editor window** where the user can drag blocks from the palette browser for composing new schemes.

All Xcos files end with extension “.xcos”. In previous versions of Scilab all Xcos files end with extension “.xcos”.



Step 7: Xcos programming – block types

In Xcos the main object is a **block** that can be used in different models and projects.

A Xcos block is an element characterized by the following features: input/output ports; input/output activations ports; continuous/discrete time states; ...

Xcos blocks contain several type of links:

- **Regular links** that transmit signals through the blocks ports (black triangle);
- **Activation links** that transmit activation timing information through the block ports (red triangle);
- **Implicit links**, see tutorial Xcos + Modelica (black square).

The user should connect only ports of the same type.

Block configuration can be specified from the **input mask** by double-clicking on the block.



Step 8: Roadmap

In this tutorial we describe how to construct the LHY model and simulate it in Xcos. We :

- provide a description of all the basic blocks used for the LHY model;
- provide a description of the simulation menu;
- provide a description of how to edit a model;
- construct the LHY scheme;
- test the program and visualize the results.

Descriptions	Steps
Basic blocks	9-16
Simulation menu	17-19
Editing models	20-22
Scheme construction	23-27
Test and visualize	28

Step 9: The integral block

Palette: Continuous time systems / INTEGRAL_m

Purpose: The output of the block $y(t)$ is the integral of the input $u(t)$ at the current time step t .

In our simulation we use this block to recover the variable $L(t)$ starting from its derivative $L'(t)$. The initial condition can be specified in the input mask.

Hint: Numerically it is always more robust using the integral block instead of the derivative block.

The integral block

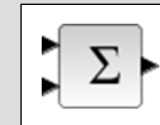


Step 10: The sum block

Palette: Math operations / BIGSOM_f

Purpose: The output of the block $y(t)$ is the sum with sign of the input signals. The sign of the sum can be specified from the input mask with "+1" for "+" and "-1" for "-".

The sum block



Step 11: The gain block

Palette: Math operations / GAINBLK_f

Purpose: The output of the block $y(t)$ is the input signal $u(t)$ multiplied by the gain factor. The value of the gain constant can be specified from the input mask.

The gain block



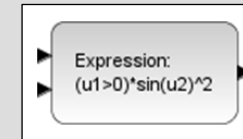
Step 12: The expression block

Palette: User defined functions / EXPRESSION

Purpose: The output of the block $y(t)$ is a mathematical combination of the input signals u_1, u_2, \dots, u_N (max 8). The name, u , followed by a number, is mandatory. More precisely, u_1 represents the first input port signal, u_2 represents the second input port signal, and so on.

Note that constants that appear in the expression must first be defined in the “context menu” before their use.

The expression block



Step 13: The clock block

Palette: Sources / CLOCK_c

Purpose: This block generates a regular sequence of time events with a specified period and starting at a given initialization time. We use this block to activate the scope block (see next step) with the desired frequency.

The clock block



Step 14: The scope block

Palette: Sinks / CSCCOPE

Purpose: This block is used to display the input signal (also vector of signals) with respect to the simulation time. For a better visualization it may be necessary to specify the scope parameters like y_{min} , y_{max} values.

The scope block

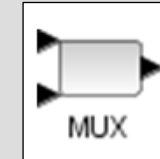


Step 15: The multiplexer block

Palette: Signal routing / MUX

Purpose: This block merges the input signals (maximum 8) into a unique vector output signal. We use this block for plotting more signals in the same windows. The number of input ports can be specified from the input mask.

The multiplexer block



Step 16: The annotation block

Palette: Annotations palette / TEXT_f

Purpose: This block permits to add comments to the scheme. Comments can also be in LaTeX coding. This block can be also called by a double click of the mouse on the scheme.

Examples of LaTeX code are L for generating L and \dot{L} for generating $L'(t)$.

The annotation block



Step 17: The simulation starting time

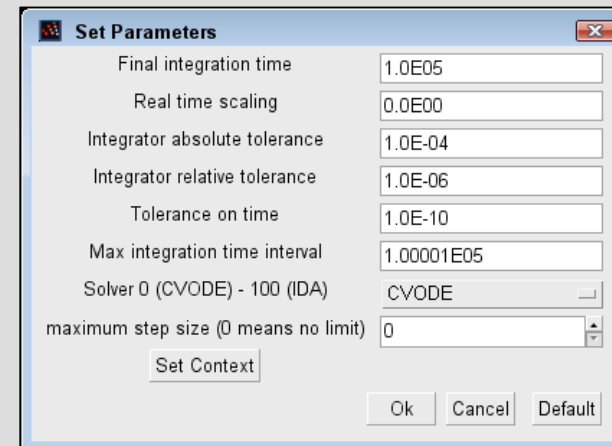
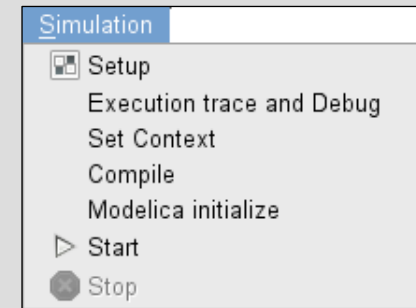
Each Xcos simulation starts from the **initial time 0** and ends at a specified final time.

The ending simulation time should be specified in the "Simulation/Setup" menu in the "Final integration time" field.

Simulation starting time is 0 !

Step 18: Set simulation parameters

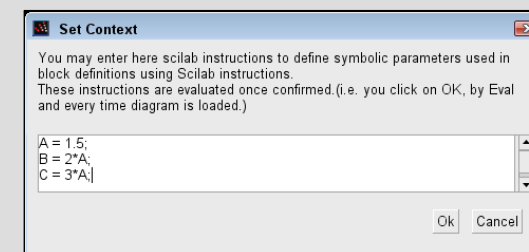
The simulation parameters such as the “final integration time” and solver tolerances can be specified from the “set parameters” dialog in the "Simulation/Setup" menu.



Step 19: Set Context

The model constants used in the block definitions can be entered in the “simulation set context”. This menu is available from the "Simulation/Set Context" menu.

In our simulation we set here all the model constants and the initial conditions, such that it is easier to change the model value for a new simulation since all the constants are available in an unique place.



Step 20: Editing the model : Align blocks

Here we report some hints to improve the visual quality of the connections between blocks. The first step is to drag some elements in the diagram and align them.

To make a selection:

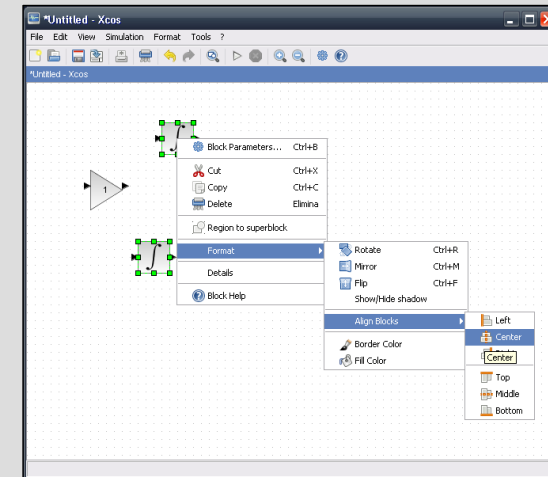
- Left-click where you want to start your selection;
- Hold down your left mouse button and drag the mouse until you have highlighted the area you want;

or

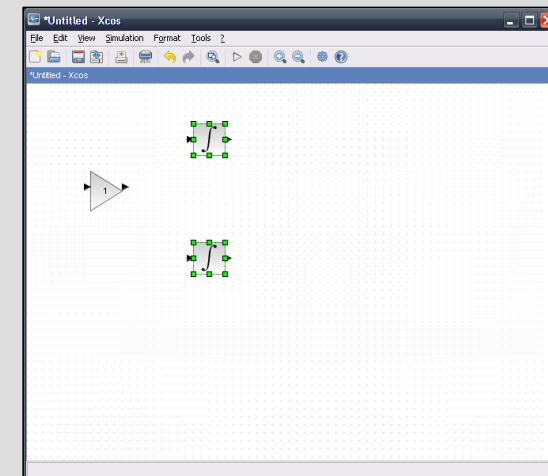
- Left-click of the mouse over the element you want;
- Ctrl + Left-click on elements that you want to select;

To align elements:

- Right-click of the mouse and select: Format -> Align Blocks -> Center.



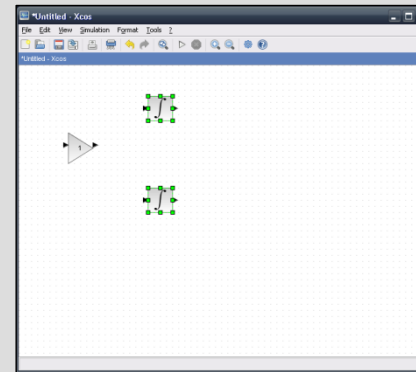
(before)



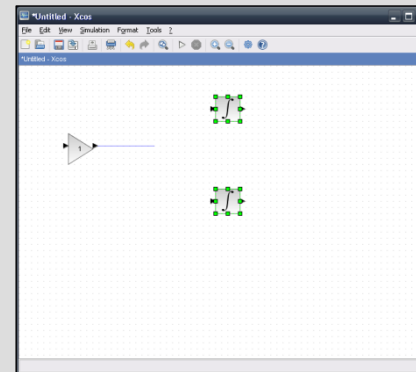
(after)

Step 21: Editing the model : intermediate points

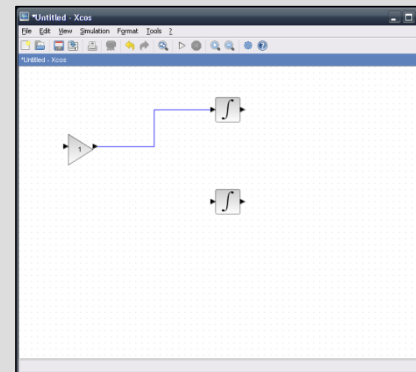
You can link elements specifying the path using a mouse click on intermediate points.



Starting configuration



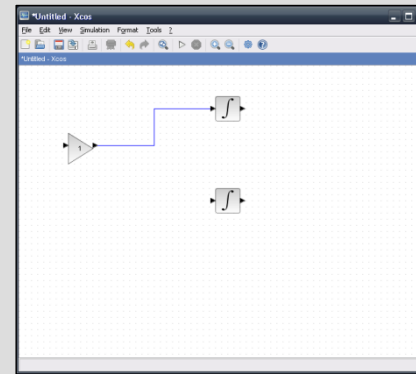
Intermediate path



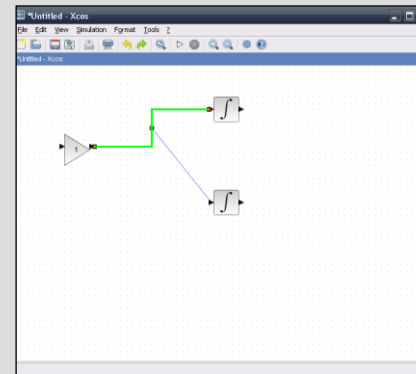
Final path

Step 22: Editing the model

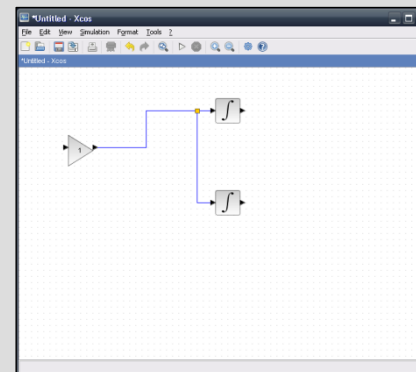
The same if you want to link elements to an already existing path (start from the element port and draw the path you want). Click with the mouse at the right position although the visualization is not correct.



Starting configuration



Intermediate path

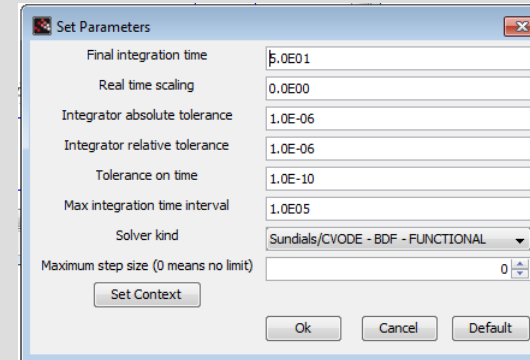


Final path

Step 23: Make the LHY scheme

Set the "Final integration time" to 50 in the "Simulation/Setup" menu;

In our simulation we modify the value of the "Final integration time" to 50 because the initial time in Xcos is 0, which corresponds to the year 1970 of our model. This means that 50 corresponds to the year 2020 of our model.

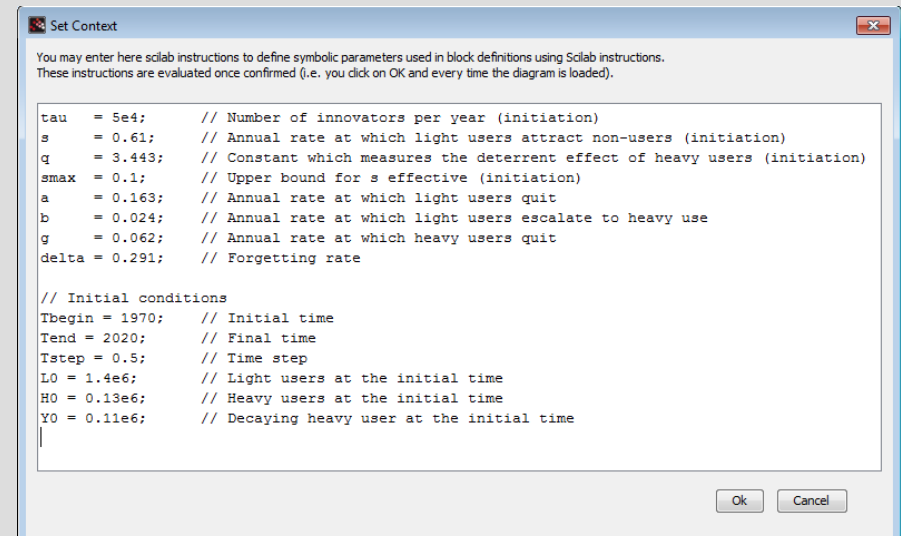


Step 24: Make the LHY scheme

Set the model constants in the "Simulation/Set Context" menu.;

```
tau = 5e4; // Number of innovators per year
(initiation)
s = 0.61; // Annual rate at which light users
attract non-users (initiation)
q = 3.443; // Constant which measures the deterrent
effect of heavy users (initiation)
smax = 0.1; // Upper bound for s effective
(initiation)
a = 0.163; // Annual rate at which light users quit
b = 0.024; // Annual rate at which light users
escalate to heavy use
g = 0.062; // Annual rate at which heavy users quit
delta = 0.291; // Forgetting rate

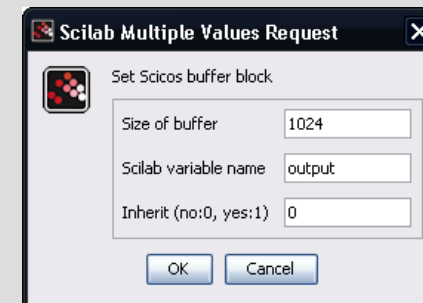
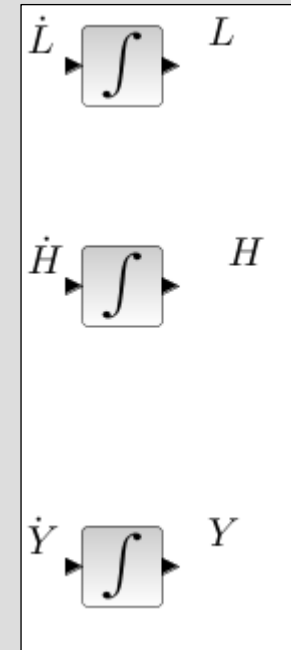
// Initial conditions
Tbegin = 1970; // Initial time
Tend = 2020; // Final time
Tstep = 0.5; // Time step
L0 = 1.4e6; // Light users at the initial time
H0 = 0.13e6; // Heavy users at the initial time
Y0 = 0.11e6; // Decaying heavy user at the initial
time
```



Step 25: Make the LHY scheme

Drag integration blocks and specify for each blocks the appropriate initial conditions.

Add also some annotations.

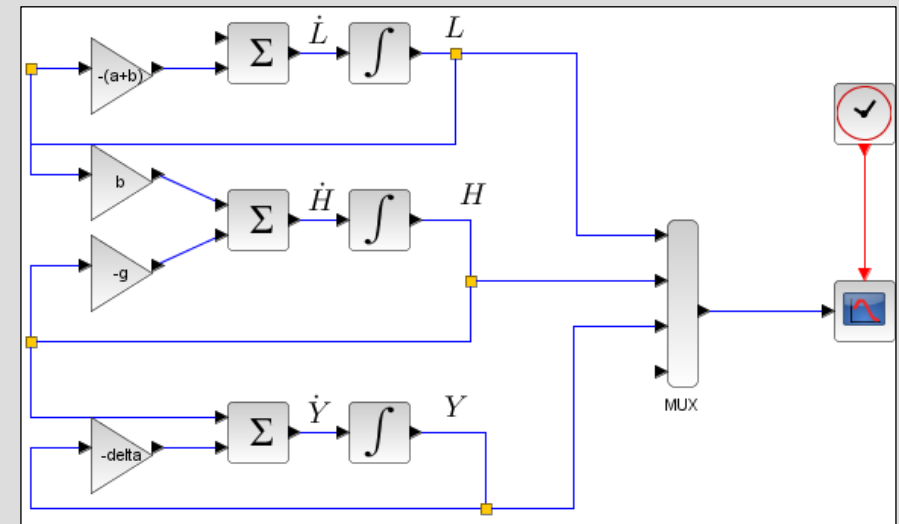


Step 26: Make the LHY scheme

Add:

- Three sum blocks;
- Three integral blocks;
- Four gain blocks with the appropriate constants;
- A multiplexer block (4 ports: L, H, Y, I);
- A scope block ($y_{min} = 0, y_{max} = 9e+6, \text{Refresh period} = 50$);
- A clock block ($\text{Period}=0.5, \text{Initiation} = 0$).

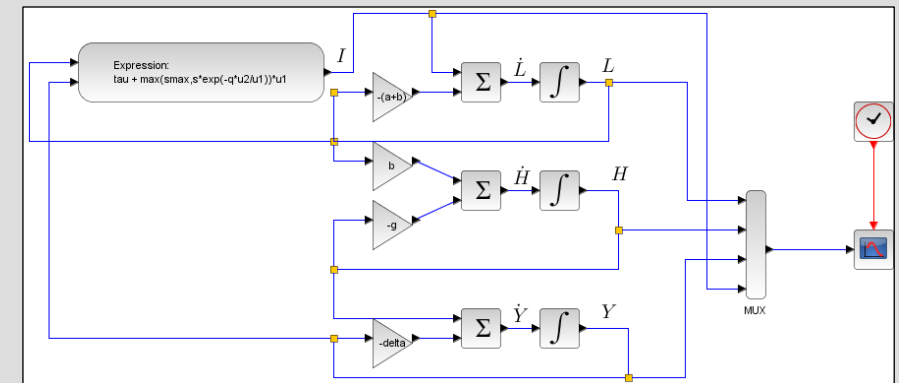
Connect the blocks as shown in the figure. For a more readable diagram it is better to comment blocks and connections using annotation blocks as reported in the figure.




Step 27: Make the LHY scheme

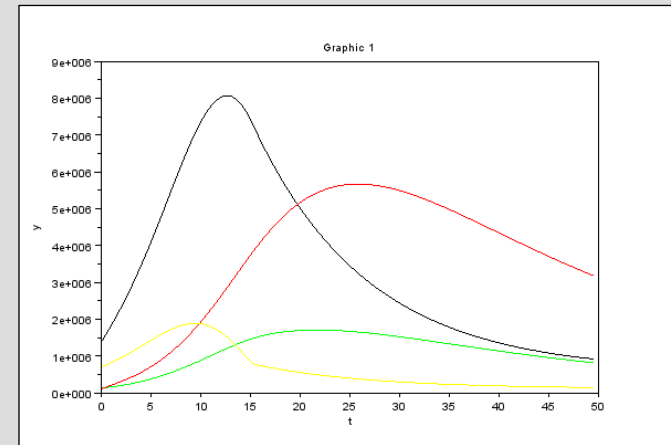
Add:

- One expression block (with two inputs and the Scilab expression $\text{tau} + \max(\text{smax}, s \cdot \exp(-q \cdot u_2 / u_1)) \cdot u_1$);



Step 28: Running and testing

Click on the menu button  .

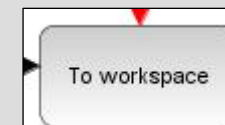


Step 13: Exercise #1

Modify the main program file such that it is possible to write data to Scilab environment and plot data from Scilab.

Hint:

Use the block in "Sinks" "To workspace" as reported on the right.



Step 14: Concluding remarks and References

In this tutorial we have shown how the LHY model can be implemented in Scilab/Xcos

On the right-hand column you may find a list of references for further studies.

1. Scilab Web Page: Available: www.scilab.org.
2. Openeering: www.openeering.com.
3. D. Winkler, J. P. Caulkins, D. A. Behrens and G. Tragler, "Estimating the relative efficiency of various forms of prevention at different stages of a drug epidemic," Heinz Research, 2002. <http://repository.cmu.edu/heinzworks/211/>.

Step 15: Software content

To report a bug or suggest some improvement please contact Openeering team at the web site www.openeering.com.

Thank you for your attention,

Manolo Venturin

```
-----  
LHY MODEL IN SCILAB/XCOS  
-----  
  
-----  
Main directory  
-----  
ex1.xcos           : Solution of the exercise  
LHY Tutorial Xcos.xcos : Main xcos program  
license.txt       : The license file
```