Documentation Sysschematic

J. Zwartjes, G.E.G. Notkamp

September 2020

1 How to use the program

1.1 Installing the program

1. To use the program first download python in which this program is used. Go to https://www.python.org/downloads/release/python-370/ and download the version which is correct for your system.

2. Second is to download the .zip file from the Github repository or clone the repository on your PC if you want to be able to get frequent updates.

   (a) For the first method go to https://github.com/joller134/Sysschematic and select the upper green button on the right to download the code as a .zip file. Extract this zip in your directory of preference and your done. Go to the next step.

   (b) for the second method go to https://git-scm.com/downloads and download and install the correct version for your operating system. Navigate to your directory of preference. Press here shift and right mouse click and select the option to open a terminal there (on windows it says open powershell window). Perform the following function

   ```
   git clone https://github.com/joller134/Sysschematic
   ```

3. next on is to install all the dependencies which are used by syschematic to run. To do this open a terminal here (if you used option A in the previous step look at option B how to open a terminal) and execute the following command

   ```
   pip install -r requirements.txt
   ```

   This should install everything necessary

4. last thing to do is run the actual program. This is done by executing the following in your terminal.

   ```
   python DeliveryDemo.py
   ```

the program will now open its user graphical interface note however that it still runs in the terminal so dont close the terminal. All debugging information is send here. If an error occurs please make a screenshot of this or copy the text and sent it to the developers.

1.2 Global Usage

the interface is divided in 3 parts.

1. the upper left menu has all global options
2. the main canvas is where all graphics are shown
3. the right menu shows node specific options.

Nodes can be added by pressing the upper add node button or a graph can already be preloaded from the examples folder. To perform an action node selection is needed. This can be done by hovering your mouse over a node and selecting one. A range of options will show up which are totalled to be

- add Node which when selected will put a node on the next mouse click
- connect Transfer/module will connect 2 selected nodes
- remove transfer removes a selected transfer
- Perform test identifiability calls harms toolbox. requires matlab to be properly configured with your python installation
- find shortest path shows the shortest path between 2 selected nodes
- find disjoint path shows the disjoint path and set between 2 groups.
- Create minimum tree will show the minimum pseudo tree
- merge maximum tree will perform merging until the largest pseudo tree is achieved.
- Immersion performs said action

the following options only show when a node is selected.

- Toggle transfer known peforms said action
- Toggle transfer pms performs said action
- remove node deletes the selected node
- Make group sets all selected nodes to one group
- Remove group will remove all made groups

- add external excitation will add said aspect to the selected node
- remove external excitation speaks for itself
- add noise to a specific selected node
- remove noise from a specific selected node
- make unkown speaks for itself
- the final checkboxes are such that you can assign variables to a node.

these options can be selected to perform operation on the current graph. The more global operations are assigned above and are as follows

- load .mat file will load in a .mat which contains information in the harms toolbox structure.
- export .mat file will save the current graph and ask for a filename after it will be save under the data directory.
- change node view will switch between abstract and detailed.
- Options will in the future contain preference view for the program like changing settings and choosing default saving directory.
- Clear window resets all the values and removes all graphics on the screen.
- load noise view will switch the view to the noise overlay
- load transfer view will reverse back from noise overlay to the normal overlay
- change line view will remove in noise view all the normal transfer lines.
- PMS peforms the PMS operation
- last but not least to dropdown boxes to select certain operation methods for example the algorithm with which to load in the matrix.

for further information for each specific functions you can take a look at their description below.

2 The functions

2.1 Manual functions

2.1.1 PMS

In all of the Predictor Model Selection cases, the user needs to select a target module. This is done by selecting a module and pressing the "toggle transfer

pms" button. When done correctly, the module will turn blue. The USC function also needs an extra input, a set of accessible nodes. Nodes are added to this set by selecting them, after which they turn pink.
Next, the selection of the PMS functions needs to be made. This is done in the scroll-bar under the "PMS" button. When the selection is made, the "PMS" button will start the function.

2.1.2 Shortest Path

Before pressing the find shortest path button in the menu, first two nodes need to be selected. This can be done by one mouse click on the nodes, turning them pink. If the nodes that are selected have no path between them, the function will give an error. If there is a path, the function will find the shortest path between the nodes and highlight the edges of the path and make them thicker.

2.1.3 Disjoint Path

The first step to finding the disjoint paths is creating two sets of nodes. This can be done by selecting all the nodes in a set and pushing the "make group" button. After pushing the "Find disjoint path" button, the function will search for a disjoint path from the first group to the second group.

2.2 Perform test identifiable

For this function matlab should be well configured. For this the internal files have to be edited which is not recommended. In the future this will be elaborated and the program will automatically configure matlab.

2.3 create minimum tree

when this is selected each node and all their outgoing edges are added to be one tree and given a color.

2.4 merge maximum tree

When selected a merge matrix is printed in the terminal and it will try to merge as many times as possible to the smallest amount of maximum trees

2.5 Global Structure

Before discussing each function first a global summary of the program is given
When the program is launched 2 important thinks happen

1. the menu is initialized in the variable Master

   (a) mainMenu contains the upper left menu with the global options

(b) subMenu contains the right-aligned menu which shows options per node

(c) canvas contains the grid in which we will draw our graphics

2. next in the canvas grid a canvas widget called draw will be launched

Master and draw both contain all the information we need to store for the visual interface. This will be passed along a lot to functions. In subMenu are all the buttons located. When a button is pressed it executes the primary function for that purpose which calls any necessary function to perform the operation. All the information is stored in in global variables in the program so that every function can read them but only edit the necessary variables.

3 Matlabhelp format of functions

3.1 USC

[D,Y,A,B] = USC(target,accessible) User Selection Case
target The target module
accessible Set of indexes of nodes that are accessible
D Set of indexes of nodes that appear in the vector of predictor inputs for predicted outputs
Y Set of indexes of nodes that appear in the vector of predicted outputs
A Set of indexes of nodes that only appear as predictor input, that do not have any confounding variable effect
B Set of indexes of nodes that only appear as predictor input

3.2 MIC

[D,Y,A,Q] = MIC(target) Minimum Input Case
target The target module
D Set of indexes of nodes that appear in the vector of predictor inputs for predicted outputs
Y Set of indexes of nodes that appear in the vector of predicted outputs
A Set of indexes of nodes that only appear as predictor input, that do not have any confounding variable effect
Q Set of indexes of nodes that appear both in the predicted output, and in the predictor input

3.3 FIC_call

[D,Y,A,B] = FIC_call(target) Full Input Case caller
target The target module
D Set of indexes of nodes that appear in the vector of predictor inputs for predicted outputs
Y Set of indexes of nodes that appear in the vector of predicted outputs

A Set of indexes of nodes that only appear as predictor input, that do not have any confounding variable effect
B Set of indexes of nodes that only appear as predictor input

3.4 Immersion

[G,B,R] = Immersion(NG,NH,NR,Unknownnodes)
G Immersed network matrix with modules
B Immersed network noise module
R Immersed network excitation module
NG Network matrix with modules
NH Network noise module
NR Network excitation module
Unkownnodes Set of indexes of nodes that are unknown