

User Guide for the MATLAB toolbox:
MTIDS
MATLAB Toolbox for Interconnected Dynamical
Systems

Chapter 1

Introduction to MTIDS

The MATLAB toolbox MTIDS is a tool for modelling and simulating systems that consist of several subsystems which are connected to each other either physically or via communication paths. These systems are also known as large-scale interconnected systems, networked dynamical/control systems, multi-agent systems or distributed dynamical systems. There is no other software tool that combines graph theory with simulation. Using MTIDS many systems or networks from very different fields can be modeled. Some technical examples are the power system, including transformers, loads and generators, the traffic system which tries to avoid jams and the internet which connects millions of computers. Even in biology some interconnected systems can be found, like neuronal networks for brain studies, animal populations or little ecosystems. Another point of research is the flocking behavior of birds or fish. Other practical applications are economic or social networks. In the next chapter some basic knowledge about the graph theory is stated, because MTIDS is based on graph theory.

Chapter 2

Graph theory

For modeling these interconnected systems a graph G is defined by a set of vertices V (also called nodes) and a set of edges E , [1], [3], [4]. Two vertices i and j that are connected by an edge can be written in brackets like (i,j) .

There are two types of graphs, directed or undirected. In undirected graphs the nodes that are connected with an edge communicate in both directions which is shown by a line. The nodes in directed graphs are connected with arcs that means the information flow is not bidirectional. If two nodes communicate in both directions, they are connected via two antiparallel arcs.

Three special undirected graphs can be found:

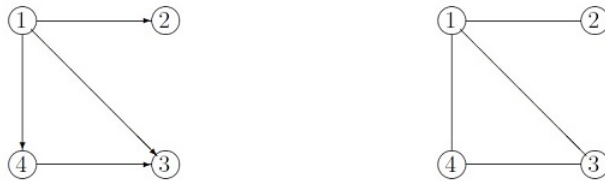


Figure 2.1: Directed and undirected graph

- The tree, where every node is connected by exactly one edge. It contains no cycles.
- The cycle in which the nodes are joined in a closed chain.
- The complete graph, in which every node is connected to all the other nodes

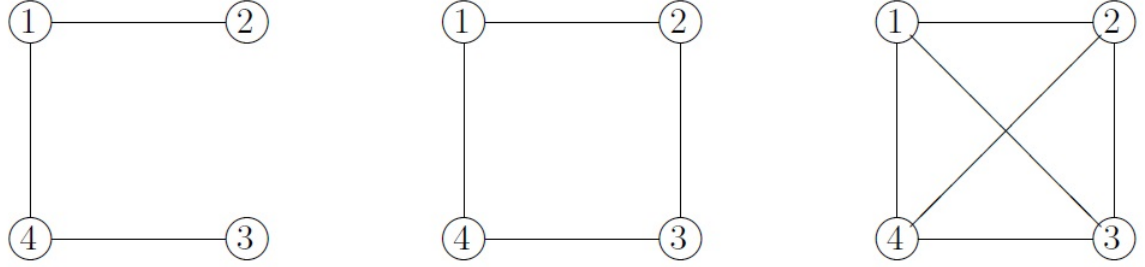


Figure 2.2: Special graphs: tree, cycle and complete graph

For the possibility of making some computations three important matrices are defined. The entries of the *incidence matrix* of a directed graph are defined as follows:

$$C = (c_{ij}) = \begin{cases} 1 & \text{if edge } j \text{ enters node } i \\ -1 & \text{if edge } j \text{ leaves node } i \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

The *incidence matrix* exists only for simple graphs that means, for graphs without self-loops.

The entries of *adjacency matrix* are computed like this:

$$A = (a_{ij}) = \begin{cases} 1 & \text{if } (i, j) \text{ is a member of the graph} \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

The matrix A is the binary directed or undirected connection matrix. Normally A has no diagonal elements, because in most cases no edge starts and ends in the same node.

The definition of the entries of the *degree matrix* is:

$$D = (d_{ij}) = \begin{cases} \deg(i) = \sum_{k=1}^N a_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

D is a diagonal matrix. Its elements are the degrees of the nodes.

The *laplacian matrix* L is difference between the *adjacency matrix* A and the *degree matrix* D .

$$L = D - A \quad (2.4)$$

The *laplacian matrix* is symmetric, positive semi definite and diagonally dominant.

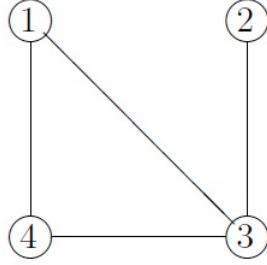


Figure 2.3: Undirected example graph 1

The matrices for this graph look like follows:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} L = \begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix} \quad (2.5)$$

If the graph is connected the eigenvalues of L increase starting with zero.

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_N \quad (2.6)$$

If the eigenvalue $\lambda_2 = 0$, which is called algebraic connectivity or fiedler value, the graph is not connected. It is a measure for connectivity. The greater λ_2 is, the better is the robustness of the graph.

Chapter 3

Working with MTIDS

MTIDS is a MATLAB toolbox with an open-source license (GNU GPL v2). To download and install the toolbox the link in [2] is needed. In Downloads the actual version can be downloaded with a *.zip folder and extracted on the computer.

Before starting the program, MATLAB has to be opened. After selecting the right folder the toolbox starts by running the file mtids.m. At first the following window appears:



Figure 3.1: Start window

Here the user can choose whether to work with former settings or to begin a new graph. Then the main user panel opens.

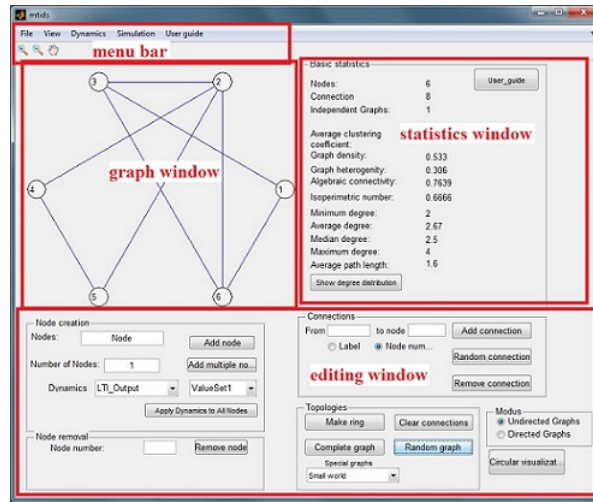


Figure 3.2: User panel

3.1 User panel

The picture above shows the user panel. It contains four main areas:

- the menu bar on top
- the graph window on the left
- the statistics window on the right
- the editing window on the bottom

3.2 Menu bar

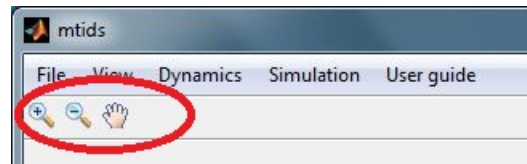


Figure 3.3: Menu bar

On top of the graph there are tools to manipulate the graph (red). After clicking on the loupe with the plus sign the user can zoom in. To zoom out of the graph the user has to select the loupe with the minus sign. If the hand-button is activated the whole graph can be moved. When a tool is selected the button stays pressed. To deactivate a tool the pressed button has to be clicked again.

3.2.1 File

Under the item File there are six elements:

- New graph
- Load...
- Save as...
- Export to workspace...
- Export graph
- Exit

With **File**→**New graph** the existing graph is deleted. So the user can start to build a new one.

With **File**→**Load...** a stored graph can be loaded.

With **File**→**Save as...** the actual graph can be saved with a certain name in a *.mat-file.

With **File**→**Export to workspace...** a new window where the three important graph matrices can be stored to the workspace opens:

Here the workspace name and the variable name can be changed. With the radio buttons, one of the three matrices can be selected:

- Laplacian Matrix
- Adjacency Matrix
- Edge List

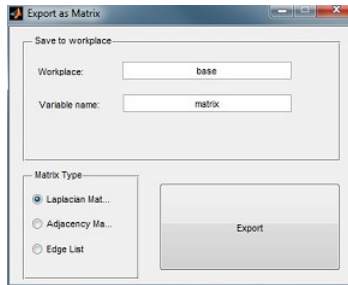


Figure 3.4: Export graph matrix window

More informations about the matrices is written in the chapter 2.
 With **File**→**Export graph** the actual graph can be exported in the opened window.

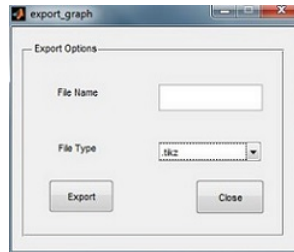


Figure 3.5: Export graph window

To save the graph as a *.tikz-, *.eps-, *.jpg-, *.pdf-, *.png- or *.tiff-file with a certain name, click the 'Export'-button. Using the 'Close'-button, the window shuts.

With **File**→**Exit** the user can quit the program.

3.2.2 View

In the menu point 'View' the appearance of the nodes can be changed in three possible ways:

- Label view
- Number view
- Blank view

Additionally the node color can be set in an external window by selecting **View→Set node color**.

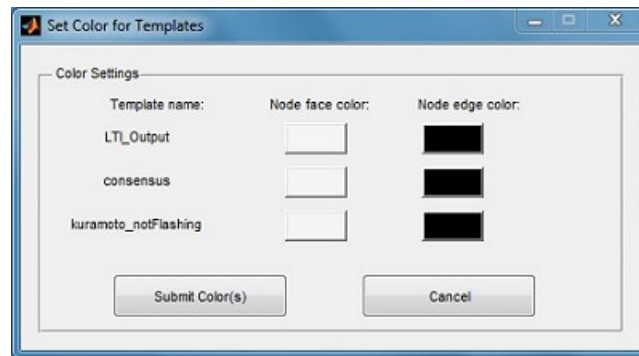


Figure 3.6: Set node color window

3.2.3 Dynamics

In the segment 'Dynamics' in the menu bar the dynamics of the nodes can be handled under the following topics:

- Create Dynamics Template
- Manage Parameter Sets
- LTI Analysis

The first thing to do when building an interconnected system is to create the subsystem dynamics which can be applied to the nodes of the graph. Therefore a SIMULINK model with a assigned frame of SIMULINK blocks called 'template' is provided. To use the templates, they have to be loaded in the parameter set manager (PSM) where the parameter sets of the template can be edited and saved. How to build templates and apply dynamics to a node is described in a more accurate way in the chapter 3.6.

Create Dynamics Template

To create dynamics of all kind, an empty template as SIMULINK model is provided. By clicking **Dynamics**→**Create Dynamics Template** the SIMULINK browser and a SIMULINK model called 'newtemplate' opens.

In this model some SIMULINK blocks are already integrated. Please do not change them! This structure is important for exporting or importing the SIMULINK models to MTIDS.

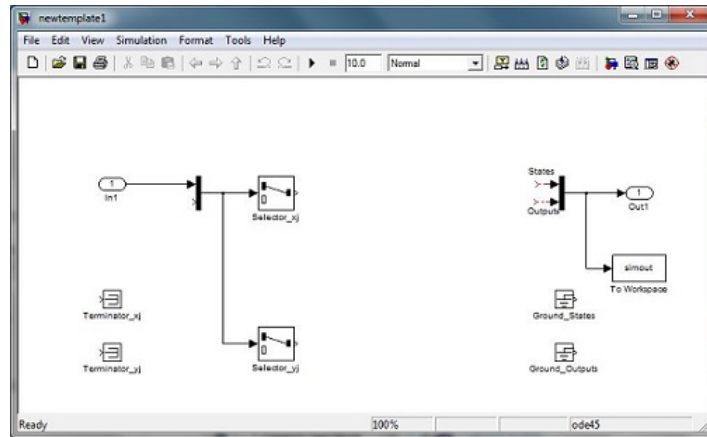


Figure 3.7: SIMULINK model of an empty template

In the free space which is marked red in the figure above any SIMULINK block can be used to create the desired dynamical system.

Manage Parameter Sets

Under the menu point **Dynamics**→**Manage Parameter Sets** the parameter settings window opens. Here, compared to the parameter setting window in the node settings after double-clicking on a node, the parameter can be managed in a more detailed manner.

In the 'Managing mode' (red) it can be selected whether the parameter set of an old template is edited or a new template is imported to MTIDS and a new parameter set is made. The 'Managing modes' only differ in the upper part of the PSM (blue).

In the 'New template import'-mode a template can be loaded by clicking the 'Load template'-button and choosing the new template. New blocks can be created with the 'Add Block'-button. In the appearing window the name of the wanted block has to be inserted, for example 'Gain'. With the 'Add Parameter'-button further parameters for new blocks can be created.

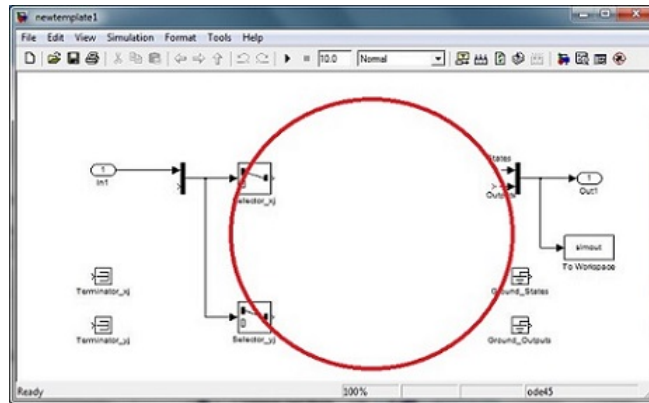


Figure 3.8: The space for putting in the system dynamics

The 'Test Value Set'-button tests the edited parameter set for compliance with the model to prevent unnecessary SIMULINK errors. With the 'Finish import'-button the import of the template is completed and the template with its parameter set is saved.

In the 'Edit parameter sets'-mode an already imported template and an existing parameter set can be selected. Three standard templates are given and can be selected in the 'Choose parameter set' field.

In this mode the parameter set can also be tested or deleted or saved in a new file. The 'Set is active'-button (green) controls whether the chosen parameter set can be used in the editing window to apply it to a node.

The chosen parameter set can be seen in the 'template parameter value set'. There, it is also possible to change the parameters manually.

The 'Input Specifications' show linear depending state or output parameters and how much of the are internal inputs. The 'Close'-button quits the parameter set manager.

In the 'Random Parameters Setup' a random value for the in the scroll bar with its name selected parameter can be set either in a uniform or a normal manner. For a normal distributed random value the mean and the standard deviation can be defined. The minimum and the maximum value are properties of the uniform distribution. According to the number of inputs and outputs the matrix dimensions have to be entered in the right way.

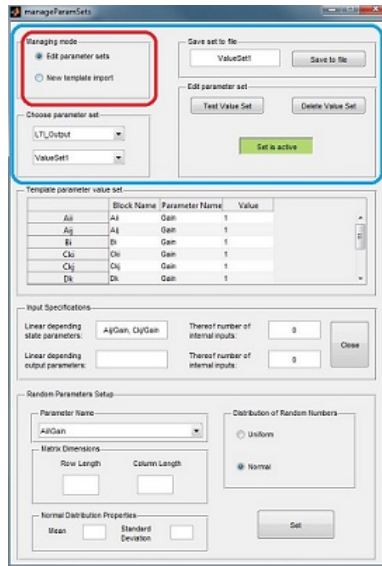


Figure 3.9: Parameter set manager

LTI Analysis

With a click on **Dynamics**→**LTI Analysis** test window appears. The LTI analysis is only possible, if the every node of the graph has LTI dynamics. The dynamics of a node can be chosen in the 'editing window' before clicking the 'Add node'-button or in the node settings.

A press on the 'Test'-button checks whether the interconnected dynamical system is stable, observable and controllable. If not a pop-up window gives instructions to solve the LTI problems.

With the 'Set'-button the program itself creates a stable states matrix for a node.

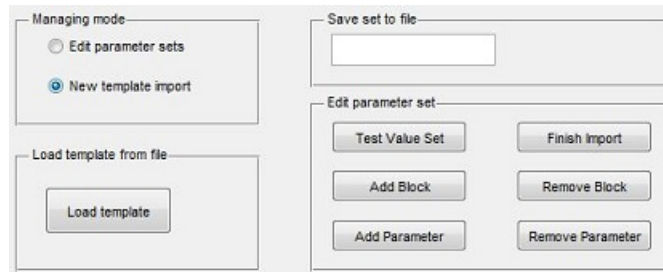


Figure 3.10: Parameter set manager when 'New template import' is selected

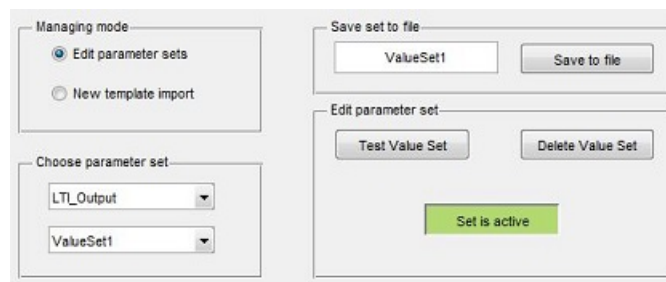
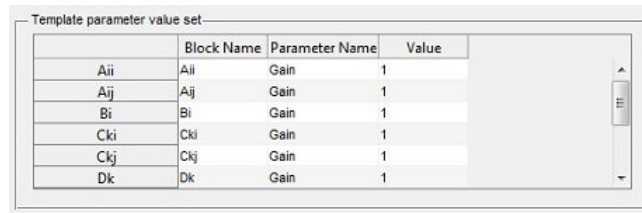


Figure 3.11: Parameter set manager when 'Edit parameter sets' is selected

3.2.4 Simulation

The element 'Simulation' in the menu bar serves the functions to run the interconnected systems made in MTIDS. In the chapter 3.6 is explained how to run simulations.

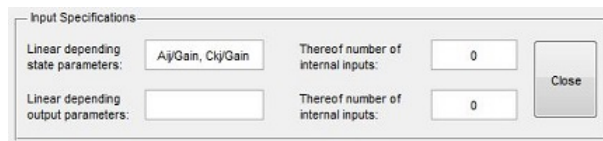
- Export to Simulink
- Disable Simulink Warnings
- Run Simulation & Plots
- Plot all output
- Show Simulink Model
- Input parameter adaption
- Integrator initial condition



Template parameter value set

	Block Name	Parameter Name	Value
Aii	Aii	Gain	1
Aij	Aij	Gain	1
Bi	Bi	Gain	1
Cki	Cki	Gain	1
Ckj	Ckj	Gain	1
Dk	Dk	Gain	1

Figure 3.12: Template parameter value set



Input Specifications

Linear depending state parameters: Thereof number of internal inputs:

Linear depending output parameters: Thereof number of internal inputs:

Figure 3.13: Input specifications

- Set simulation parameters

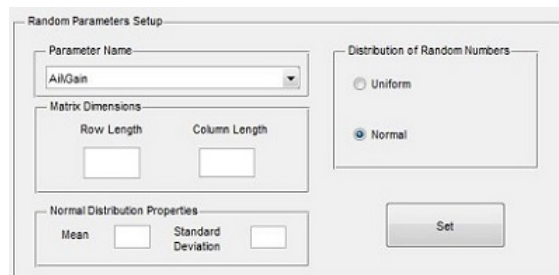
With **Simulation**→**Export to Simulink** the whole graph is exported to a SIMULINK model for reasons of simulation.

With **Simulation**→**Disable Simulink Warnings** the user turns off all the warning messages generated by SIMULINK.

With **Simulation**→**Run Simulation & Plots** the simulation starts in MTIDS with the user defined settings which can be changed in the edit node window after double-clicking on a node. If the plotting options are selected, MTIDS displays the plots after the simulation.

If **Simulation**→**Plot all output** is marked every output of the graph is plotted after the simulation.

If **Simulation**→**Show Simulink Model** is marked after starting the sim-



Random Parameters Setup

Parameter Name:

Matrix Dimensions: Row Length Column Length

Distribution of Random Numbers:
☐ Uniform
☒ Normal

Normal Distribution Properties: Mean Standard Deviation

Figure 3.14: Random parameter setup

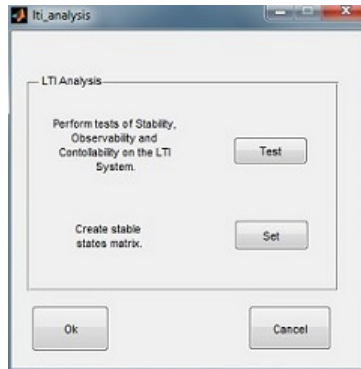


Figure 3.15: LTI analysis

ulation the SIMULINK model is opened.

When the user adds new edges to the graph, the input parameter matrix A_{ij} of a template does not have the correct size anymore. This problem can be fixed automatically using **Simulation**→**Input parameter adaption**.

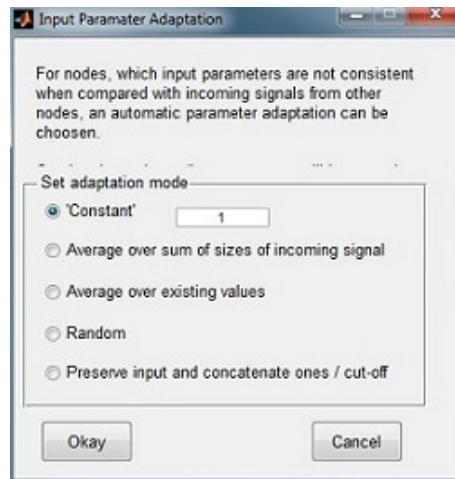


Figure 3.16: Input parameter adaption

In the appearing window there are five different options how MTIDS should set these size-depending parameters.

- Fill with a specific constant value the user can define.

- Set for every entry $1/n$, where n is the number of incoming signals.
- Average over the existing values if the parameter was set before.
- Fill them with a random matrix.
- Preserve the matrix if it was set already but concatenate with ones, or trim to the required size.

So there is no need to conform the parameters manually and further proceedings like simulations may follow quickly.

With **Simulation**→**Integrator initial condition** the initial conditions for all integrators in the entire system, which means for every SIMULINK integrator blocks of the used templates, can be set.

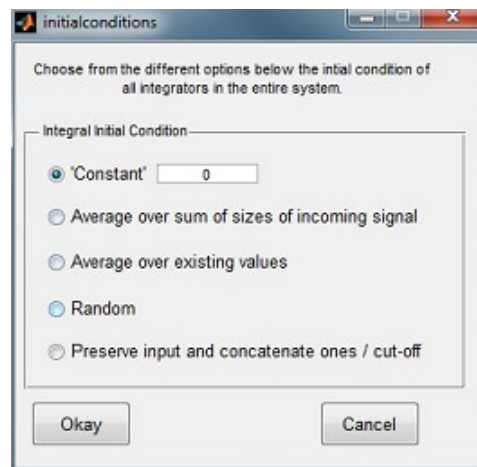


Figure 3.17: Initial conditions for integrators

The options are:

- Choose a certain constant.
- Average over the sum of the sizes of incoming signals.
- Average over the existing values.
- Take random values.

- Preserve the matrix if it was set already but concatenate with ones, or trim to the required size.

With **Simulation**→**Set simulation parameters** the simulation parameters window starts.

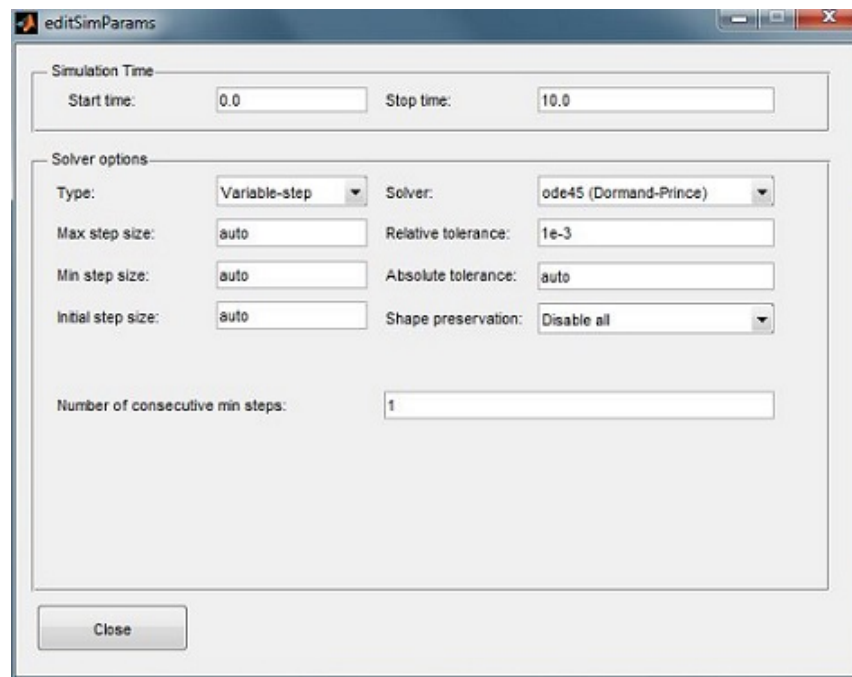


Figure 3.18: Edit simulation parameters

In this window the most important SIMULINK simulation options can be set. They can also be found in the SIMULINK browser under **Simulation**→**Configuration Parameters...**

In the 'Simulation Time' element the start time and the stop time can be insert. The solver type and certain solver can be selected in the 'Solver options'. It is feasible to set the maximal, the minimal and the initial step size. Other options are the relative tolerance, the absolute tolerance, the shape preservation and the number of consecutive min steps.

3.2.5 User guide

The User guide can be opened by clicking on the button in the statistics window or in the menu bar under the item **User guide**→**Start User guide**.

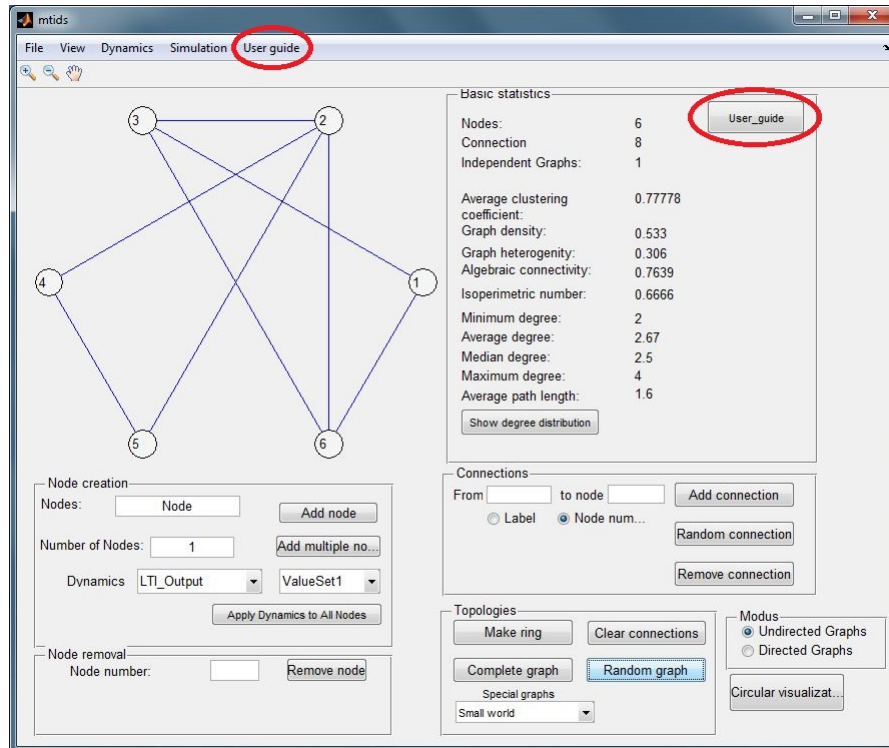


Figure 3.19: User panel with marked user guide buttons

The user guide provides information to the graph theory and the statistics and it also describes the toolbox MTIDS and explains how to use it.

3.3 Graph window

In the graph window the actual graph is presented. The nodes of the graph can be moved by a click and hold with the left mouse button. A node can be added by <Shift>-<Left-click> with the mouse. To build a connection a

`<ctrl>-<Left-click>` has to be made on both nodes. If there is an existing connection, this connection will be deleted. After double-clicking on a node the 'edit node'-window appears.

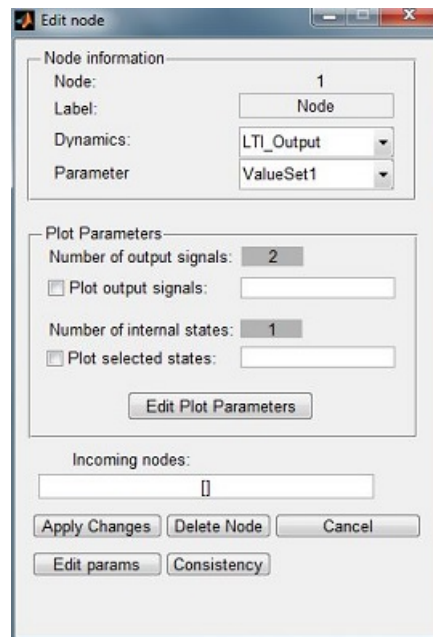


Figure 3.20: Edit node window

On top of the window is the 'Node information' element.

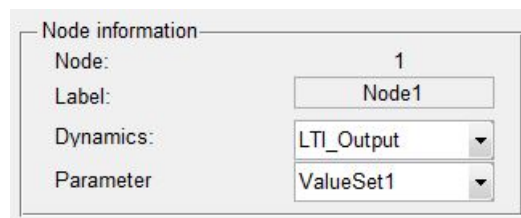


Figure 3.21: Node information

In this element the node number, the node label, the applied dynamics and the chosen parameter set can be seen. The name of the node can be changed by clicking in the labeling field. New dynamics and parameter sets can be selected by using the control bars.

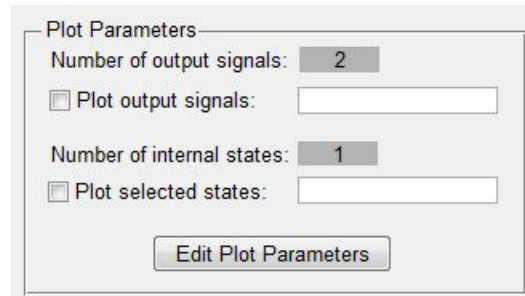


Figure 3.22: Plot parameters

To edit the plot parameters the window shown in the picture above is important. By selecting 'Plot output signals' or 'Plot selected states' the program plots the defined outputs or states after a simulation is run. A click on the 'Edit Plot Parameters'-button leads to following window, where the Line Width, the Line Style, the Line Color, the Marker Edge Color and the Marker Face Color can be varied.

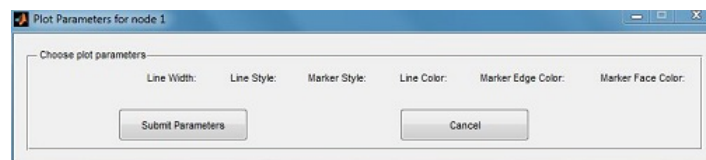


Figure 3.23: Color selection for the plots

The 'Submit Parameters'-button saves the parameters before closing the window, the 'Cancel'-button does not save them.

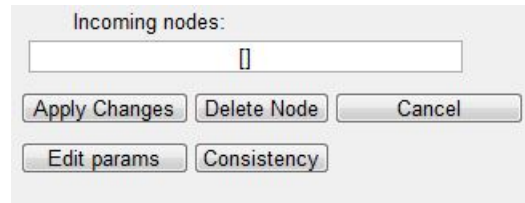


Figure 3.24: Incoming nodes and window buttons

In the element 'Incoming nodes' connections can be generated by writing the starting nodes in the brackets. Putting in a node number which does not exist, causes fatal errors.

The changes can be confirmed with the 'Apply Changes'-button. The node can be deleted with the 'Delete Node'-button. For closing the window without applying the changes the 'Cancel'-button can be used.

In order to check the consistency of the applied parameter values the 'Consistency'-button can be used. The 'Edit params'-button opens a window where the parameter values of the dynamics can be changed.

The 'Parameter Values'-window shows parameters of the dynamics in the selected template. A template is a SIMULINK model of the dynamics of a node. It can be built under the point '**Dynamics**→**Create Dynamics Template**' in the menu bar. There is also a frame where the name of the parameter set can be edited.

In the 'Input Specifications' the Linear depending state parameters and there internal inputs are listed and also the Linear depending outputs with the number of internal outputs.

To edit the parameter values the parameter set manager can be opened with

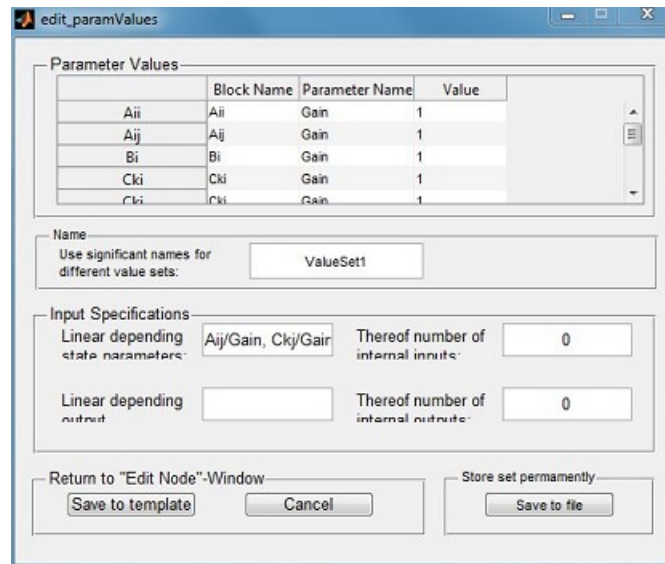


Figure 3.25: Edit parameter values

'Dynamics→Manage Parameter Sets' The settings are stored to an extra file with the 'Save to file'-button or saved to the actual template with the 'Save to template'-button. To leave the window without saving, click on the 'Cancel'-button.

3.4 Statistics window

The statistics window shows the basic statistics to the actual graph. To get informations about the statistics the user guide can be consulted by clicking the button. Also the chapter 4 provides detailed informations for the user.

3.5 Editing window

In the editing window the graph, which is displayed in the graph window, is generated.

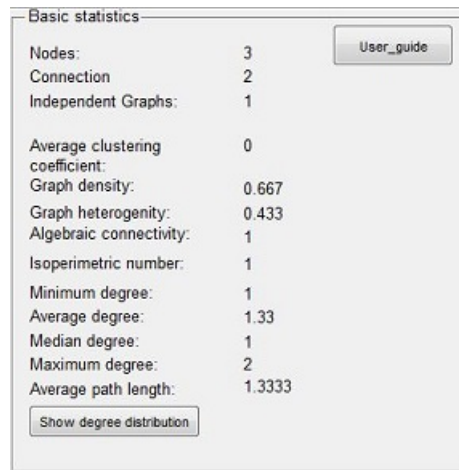


Figure 3.26: Basic statistics

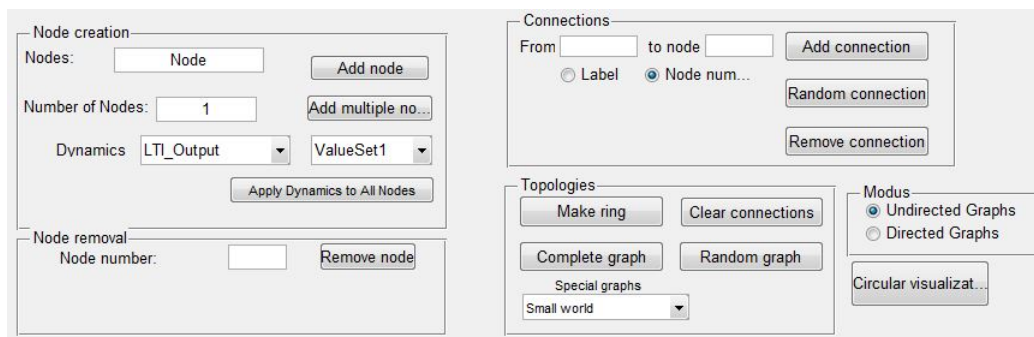
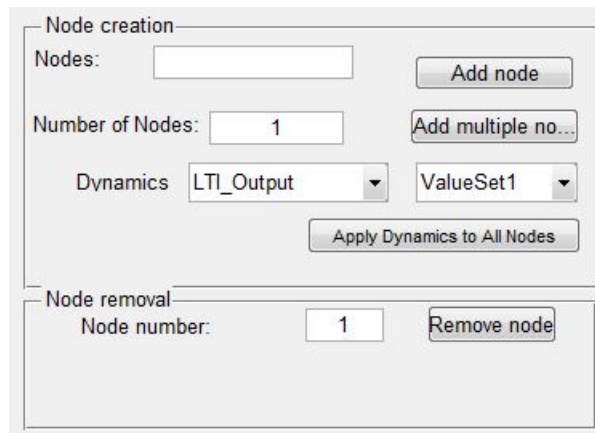


Figure 3.27: Editing window

Nodes

A node is created by clicking on the 'Add node'-button or by **<Shift>-<Left-click>** with the mouse. In the white window next to the button the node can be labeled. The labels are only displayed, when the 'Label view' under the menu point 'View' is active. It is also possible to add a user defined number of nodes with the 'Add multiple nodes'-button.



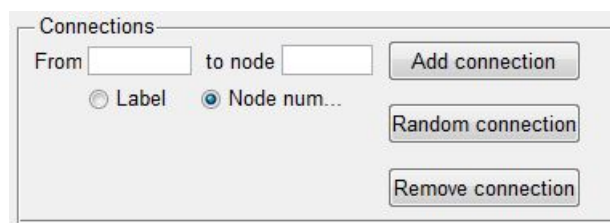
The 'Node creation' dialog box contains the following elements:

- Nodes:** A text input field followed by an 'Add node' button.
- Number of Nodes:** A text input field containing the value '1', followed by an 'Add multiple no...' button.
- Dynamics:** Two dropdown menus. The first is set to 'LTI_Output' and the second is set to 'ValueSet1'.
- Apply Dynamics to All Nodes:** A button located below the dynamics dropdowns.
- Node removal:** A section at the bottom containing a 'Node number:' label, a text input field with the value '1', and a 'Remove node' button.

Figure 3.28: Node creation

In the Dynamics-window between three dynamics and some value sets can be chosen. More explanations for the dynamics follow in chapter 3.6. With the 'Apply Dynamics to All Nodes'-button the selected dynamics are applied to every node. The 'Remove node'-button deletes the node with the user given number and all the connections starting or ending in the node.

Connections



The 'Connections' dialog box contains the following elements:

- From:** A text input field.
- to node:** A text input field.
- Radio buttons:** Two radio buttons labeled 'Label' and 'Node num...'. The 'Node num...' button is selected.
- Buttons:** Three buttons are located on the right side: 'Add connection', 'Random connection', and 'Remove connection'.

Figure 3.29: Connections creation

In the writing fields after 'From' and 'to node' the nodes that will be connected can be specified either with their label or their node number. The 'Add connection'-button generates the defined connection. If the user puts in a node number, which does not exist, a fatal error occurs. The 'Remove

connection'-button deletes the connection and the 'Random connection'-button adds a random connection anywhere in the graph. Another way to link two nodes, is a <Ctrl>-<Left-click> on both. If there is an existing connection, this connection will be deleted.

Topologies



Figure 3.30: Topologies

The 'topologies'-window provides the ability to create some standard or random graphs.

- **'Make ring'-button:** generates a circular graph
- **'Complete graph'-button:** generates a complete graph
- **'Clear connections'-button:** deletes all connections
- **'Random graph'-button:** generates a random graph

In the control bar with the heading 'Special graphs' between following random graph generating methods can be chosen:

- **Small world**
- **Scale-free**
- **Klemm-Eguiluz**
- **Adjustable scale free**

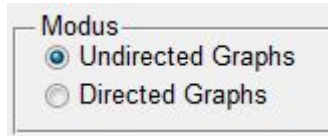


Figure 3.31: Modus selection

Modus

In this segment the modus of the graph can be selected with a radio button. The difference between an undirected graph and a directed graph is that the direction of the information flow in a directed graph is shown by an arrow connection. An edge in an undirected graph has no arrows, because the information is always exchanged in both directions.

The 'Circular visualization'-button, which is located under the modus choice, reorders the actual graph so that the nodes are arranged on a circle. The neighboring nodes have an equal distance.

3.6 Work flow

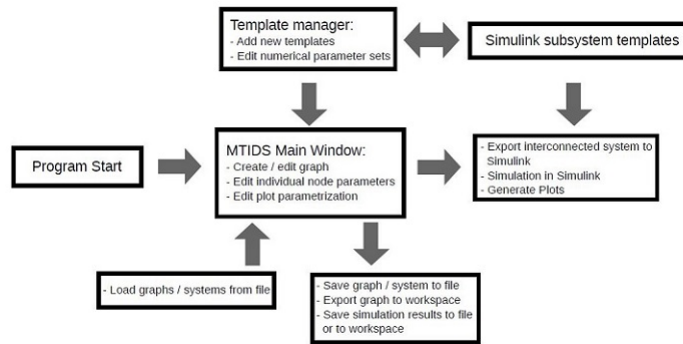


Figure 3.32: Work flow in MTIDS

The previous figure shows the general steps to work with MTIDS. After starting MTIDS the user panel which is the main interface to build intercon-

nected systems and their graph structure appears.

3.6.1 Creation of the subsystem dynamics

The first thing to do is to build the subsystem dynamics for every subsystem that will be part of the whole interconnected system. Therefore either an empty template has to be filled with the desired dynamics or one of the three standard templates can be used. A template is a SIMULINK model with a predefined frame structure (blue) for reasons of converting it to MTIDS and a region where the user can implement his own dynamics (red). This frame structure should not be changed! The following picture presents an example template with LTI dynamics. The terminator blocks are needed to avoid SIMULINK errors if the signal is not used.

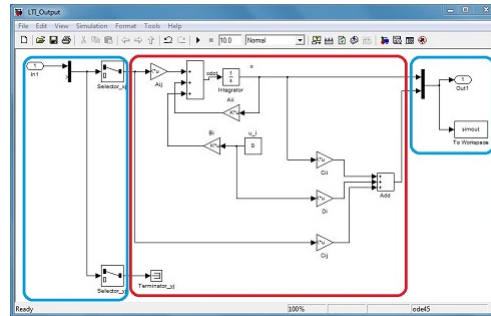


Figure 3.33: Template for a LTI system

Create a new template

With the command **Dynamics**→**Create Dynamics Template** in the menu bar the SIMULINK browser starts and an empty template turns up. In the free, red marked space the user can put in any desired dynamics by using every SIMULINK block. So there are endless possibilities for new templates and new dynamics.

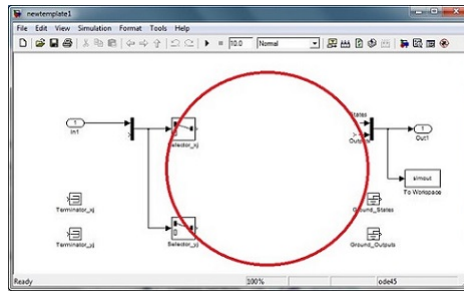


Figure 3.34: Empty template with red marked space for system dynamics

Use a provided template

Three standard dynamics are provided:

- LTI (linear time invariant) system
- Consensus problem
- Kuramoto oscillator

The corresponding templates are shown in the figures. At first the LTI system template:

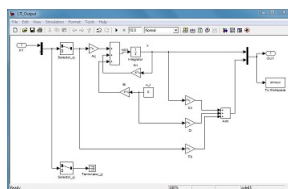


Figure 3.35: Template for a LTI system

The input of the LTI system template is constant zero, but can be replaced by any controller that uses state information from local or even external systems. Below the Consensus problem template is shown:

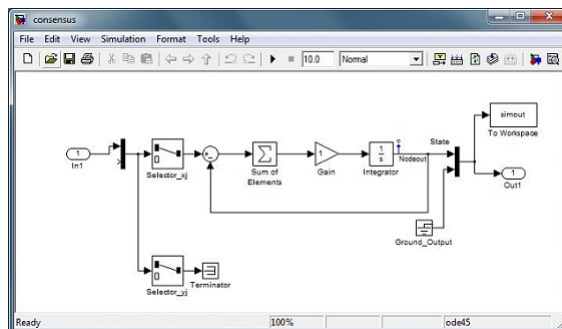


Figure 3.36: Template for a consensus problem

The Figure 3.37 displays the kuramoto oscillator template:
When the creation of the template is finished, in the next step the template

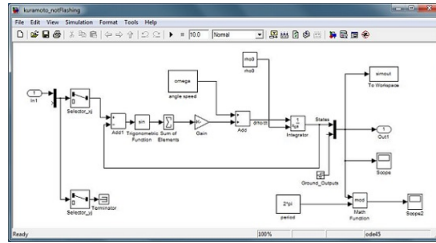


Figure 3.37: Template for a kuramoto oscillator

has to be imported to MTIDS and the parameters can be chosen in the parameter set manager (PSM). To open the PSM click **Dynamics→Manage Parameter Sets** in the menu bar. How to use the PSM is described in the chapter 3.2.3.

3.6.2 Building an interconnected graph

The building of a graph is straight forward. At first the desired Dynamics with preferred parameter set have to be chosen in the editing window of the MTIDS user panel. After that the node can be created with a mouse-click or the right key combination. Easy like this, connections for directed or undirected graphs can be added. It is also possible to build certain topologies, a complete graph, a circular graph or a random graph. When new edges are generated the parameters of a template do not fit together anymore. This problem can be solved by using **Simulation→Input parameter adaption**. The graph statistics are updated automatically. To get more informations look in the chapter 3.2.4.

3.6.3 Editing parameter settings

After double-clicking on a node the 'edit node'-window opens, where the name, the dynamics and the parameter set can be specified in the 'Node information'. In the 'Plot Parameters' it is possible to select whether the output and the states should be plotted after a simulation. More details are

explained in the chapter 3.3.

3.6.4 Simulation and post processing

When the graph with its dynamics and parameter settings is ready, the user can work with the system in MTIDS or export it to SIMULINK for a simulation. For the export **Simulation→Export to Simulink** builds a SIMULINK model. The simulation parameters can be determined in MTIDS under **Simulation→Set simulation parameters**. The simulation data and results can be saved in *.mat files. Under **Simulation→Run Simulation & Plots** the simulation starts and predefined plots are made. With **File→...** the graph can be saved or exported. More details in the chapter 3.2.

Chapter 4

Toolbox statistics

This chapter includes the explanations for the statistics of the toolbox located in the window next to the picture of the actual graph. For the two types of edges, either directed or undirected, there are two different versions of the graph statistics window.

4.1 Undirected graph

Nodes

The number of nodes that are given in the actual graph.

Connections

The number of connections (edges) between the nodes of the actual graph. If a node has no edges, it can be considered as a subgraph.

Independent Graphs

The number of independent subgraphs that are not connected to each other.

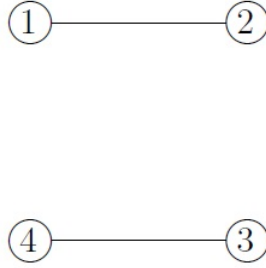


Figure 4.1: Undirected example graph 2

In this example the nodes one and two form a subgraph and the nodes three and four form a subgraph. So there are two independent subgraphs.

Average clustering coefficient

The average clustering coefficient is the number of all existing triangles divided by the number of all possible triangles. A triangle is a circle with three nodes. In directed graphs, three nodes generate up to eight triangles ($2 \times 2 \times 2$ edges), because every node be connected to another node in two directions.

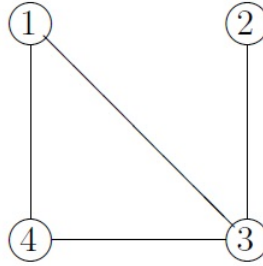


Figure 4.2: Undirected example graph 1

To get the clustering vector the number of triangles are computed for every node and put in a column vector. This means vector for the graph above is $[8, 0, 8, 8]^T$, because the node two is not an element of a triangle. For calculating the maximal possible number of triangles for every node, the number of all input and output edges are summed up in a degree vector $[4, 2, 6, 4]^T$. Since the node two is not element of a triangle the second entry is set to ∞ .

After computing the maximal number of triangles for all pairs of nodes and subtracting wrong pairs the resulting vector is $[8, \infty, 24, 8]^T$. So the vector of the clustering coefficients looks like $[1, 0, 0.333, 0]^T$. The average clustering coefficient is the mean value of the vector. In this example the average clustering coefficient is 0.583.

Graph density

The graph density is computed as follows:

$$\text{graph density} = \text{mean}(\text{degree vector}) / (\dim(L) - 1) \quad (4.1)$$

The graph density is the mean of the degree of every node divided by the number of nodes minus one. The degree of a node is the number of edges that are connected to the node. The Degree Matrix D contains the degrees of the nodes.

Graph heterogeneity

$$\text{graph heterogeneity} = \frac{\sqrt{\text{var}(\text{degree vector})}}{\text{mean}(\text{degree vector})} \quad (4.2)$$

The graph heterogeneity is the standard deviation divided by the mean of the degree vector. It says how much the number of connections differs from node to node. The degree vector is extracted from the Degree Matrix D mentioned in the chapter one.

Algebraic connectivity

The algebraic connectivity of a graph G is the second-smallest eigenvalue of the Laplacian matrix of G . This eigenvalue is greater than zero if and only if G is a connected graph. This is a corollary to the fact that the number of times zero appears as an eigenvalue in the Laplacian matrix is the number of connected components in the graph. The magnitude of this value reflects how well connected the overall graph is, and has been used in analyzing the robustness and the ability of synchronization of networks.

Rooted Spanning Tree

This field says 'Yes' if the graph can be built as a rooted spanning tree. Otherwise it says 'No'. As you can see in the figure every node is connected to the root node by one or more lines. There are no circles.

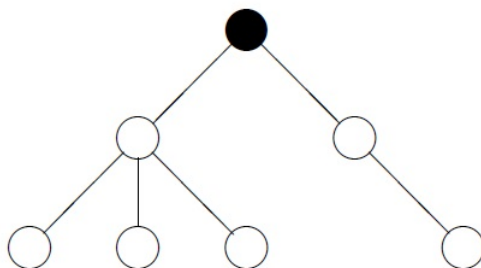


Figure 4.3: Rooted spanning tree

Minimum degree

$$\text{degree vector} = \text{diag}(L) \quad (4.3)$$

The degree vector contains the diagonal elements of the *laplacian matrix* or the *degree matrix*. These elements are the degrees of the nodes, which means the number of edges which leave or arrive in one node.

The minimum degree shows the smallest degree of all nodes.

Average degree

The average degree sums up the degrees of all nodes and divides it by the number of nodes. The degree of a node is the number of edges, which are connected to the node. It stands in the *degree matrix*.

Median degree

The median degree puts every degree of a node in an increasing row. If the number of the degrees is odd, it shows the value in the middle. If the number of the degrees is even, it shows the mean of two values in the middle.

The degree of a node can be extracted from the *degree matrix*.

Maximum degree

The maximum degree is the highest value of all degrees of the nodes. This means the biggest number of edges one node in the graph is connected to.

Average path length

The average path length is the sum of all path length divided by the number of nodes times the number of nodes minus one. The path length between two nodes is the smallest number of connections between the two nodes. If the graph contains not connected subgraphs the average path length is ∞ .

$$\text{average path length} = \frac{\sum_{i \in \text{nodes}} (\text{path length})_i}{(N(N-1))} \quad (4.4)$$

Isoperimetric number

This number gives knowledge about the robustness and the connectivity of the graph. S and S^c are subgraphs and $|S|$ and $|S^c|$ are the number of nodes in the subgraphs. $|\partial S|$ is the number of edges connecting the two subgraphs.

$$h_G = \min_{|S| \leq N/2} h_G(S) = \min_{|S| \leq N/2} \frac{|\partial S|}{\min\{|S|, |S^c|\}} \quad (4.5)$$

A small isoperimetric number means that big subgraphs are connected via a small number of edges. In the case a connection gets damaged it has great influence on the system. So the graph is not very robust.

Show degree distribution

The degree distribution shows the probability for a node in the graph to have a certain degree. The degrees of the nodes are known from the *degree matrix*. The occurrence of a certain degree is compared to the others. A big number of a certain degree leads to a big probability for a node to have these certain degree.

4.2 Directed graph

Nodes

The number of nodes that are given in the actual graph.

Weakly connected components

This field shows the number of weak connected subgraphs or weak connected components (WCC). A weak connected component consists of nodes that are connected with one or more edges. A node which has no starting or ending edge is also a WCC. For a strong connected subgraph the nodes must be connected in both directions.

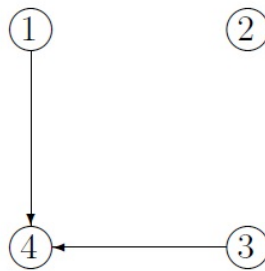


Figure 4.4: Directed example graph 1

In this example node two is a weak connected subgraph and then nodes one, three and four create a weak connected subgraph. Each node is a strong connected component itself.

Strong connected components

The field shows the number of strong connected subgraphs or strong connected components (SCC). The nodes in a strong connected subgraph are connected to each other in both directions. If a node is not strong connected to any other node, it is counted as SCC itself. The smaller the shown number, the less strong connected subgraph exist and the better the whole graph is connected. A complete graph shows the number one. If there are less sets of SCC as nodes in the graph, then there must be at least one cycle.

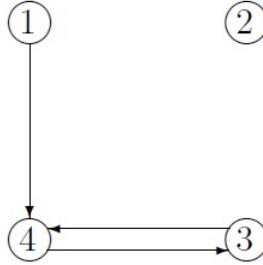


Figure 4.5: Directed example graph 2

Here the node number one and the node number two are strong connected subgraphs and the nodes three and four form together a strong connected subgraph. Two weak connected subgraphs can be seen.

Average clustering coefficient

The average clustering coefficient is the number of all existing triangles divided by the number of all possible triangles. A triangle is a circle with three nodes. In directed graphs, three nodes generate up to eight triangles ($2 \times 2 \times 2$ edges), because every node be connected to another node in two directions.

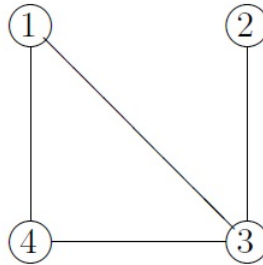


Figure 4.6: Undirected example graph 1

To get the clustering vector the number of triangles are computed for every node and put in a column vector. This means vector for the graph above is $[8, 0, 8, 8]^T$, because the node two is not an element of a triangle. For calculating the maximal possible number of triangles for every node, the number

of all input and output edges are summed up in a degree vector $[4, 2, 6, 4]^T$. Since the node two is not element of a triangle the second entry is set to ∞ . After computing the maximal number of triangles for all pairs of nodes and subtracting wrong pairs the resulting vector is $[8, \infty, 24, 8]^T$. So the vector of the clustering coefficients looks like $[1, 0, 0.333, 0]^T$. The average clustering coefficient is the mean value of the vector. In this example the average clustering coefficient is 0.583.

Average degree

The average degree is the mean of the In-degrees of all nodes in the graph.

Graph is balanced

A graph is balanced, if every node has the same in-degree and the same out-degree, this means the number of starting and arriving edges. This property is fulfilled by a ring or a complete graph. Otherwise the answer is 'No'.

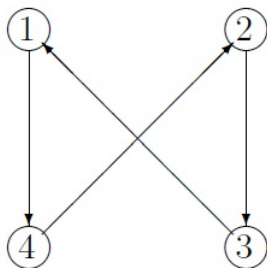


Figure 4.7: Directed example graph 3

The example graph above is balanced.

Has cycles

The answer is 'Yes', if the graph contains cycles. Otherwise it is negated. When you start in a node and travel along a path in the direction of the arrows until you return in the starting node, then you got a cycle. The graph below shows a cycle with the nodes two, three and four.

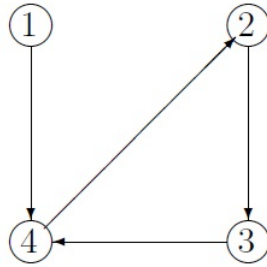


Figure 4.8: Directed example graph 4

Rooted spanning tree

This field says 'Yes' if the graph can be built as a rooted spanning tree. Otherwise it says 'No'. As you can see in the figure every node is connected to the root node by one or more lines. There are no circle.

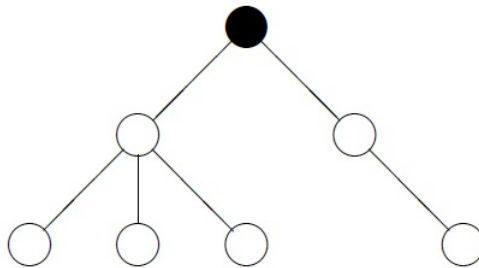


Figure 4.9: Rooted spanning tree

Minimum In-degree

The term in-degree refers to the number of edges that end in a node. The minimum in-degree gives back the smallest number of edges in the graph arriving in one node.

Maximum In-degree

The term in-degree refers to the number of edges that end in a node. The maximum in-degree gives back the highest number of edges in the graph ar-

iving in one node.

Minimum

Minimum stands for the smallest number of edges that start in a node of the graph. The full name is minimum outdegree.

Maximum

Maximum stands for the biggest number of edges that start in a node of the graph. The full name is maximum out-degree.

Average path length

The average path length is the sum of all path length divided by the number of nodes times the number of nodes minus one. The path length between two nodes is the smallest number of connections between the two nodes. If the graph contains not connected subgraphs the average path length is ∞ .

$$\text{average path length} = \frac{\sum_{i \in \text{nodes}} (\text{path length})_i}{(N(N - 1))} \quad (4.6)$$

References

- [1] F. Deroo and S. Hirche; A MATLAB Toolbox for Large-scale Networked Systems; at- Automatisierungstechnik, July 2013.
- [2] URL: <https://code.google.com/p/mtids/>
- [3] Christopher D. Godsil and Gordon Royle; Algebraic graph theory. Graduate texts in mathematics. Springer, 2001.
- [4] Dragoslav D. Siljak. Large-Scale Dynamic Systems: Stability and Structure. Dover Publications, 2007.