Numerical Test Rig for Large-Scale and Interconnected Dynamical Systems

submitted
Project Laboratory
Networked and Cooperative Control
by

cand. ing. Francisco Llobet

born on July 9, 1985
resident:
Briennerstr. 39
80333 München

cand. ing. Jose Rivera

born on December 18, 1986
resident:
Amalienstr. 87
80799 München

Institute of
Automatic Control Engineering
Technische Universität München

Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
Univ.-Prof. Dr.-Ing. Sandra Hirche

Supervisor:	F. Deroo, S. Erhart, A. Gusrialdi, H. Mangesius
Beginn:	09.05.2011
Submitted	04.07.2011

Abstract

The goal of this project was to develop a test rig for large-scale and interconnected dynamical systems. The result is MTIDS or Matlab Toolbox for Interconnected Dynamical Systems, which is a mash-up that wraps different toolboxes used for graph analysis and dynamic systems simulation together. MTIDS allows the definition and analysis of graphs, where each node has a specific dynamic assign to it. The template based design of nodes' dynamics allows great flexibility for the creation of complex interconnected dynamical systems with the possibility of implementing clusters/layers. MTIDS is an open-source project under the GNU GPL v2 license. This document presents a general desciption of MTIDS and intructions for its use.

Contents

Chapter 1

Introduction

In this first Chapter the motivation behind the MTIDS project is explained and the project's goal and framework is presented.

1.1 Motivation

Large-scale interconnected dynamical system are everywhere: biological systems, power and water systems, the brain neurons, social interaction networks, economic markets, etc. In a cononical form all of this systems can be thought as a bunch of nodes with local dynamics that interact with each other, e.g. a graph. Different topologies of the graph, may lead to different behavior. An example of various large scale interconnected systems can be seen in Figure 1.1.

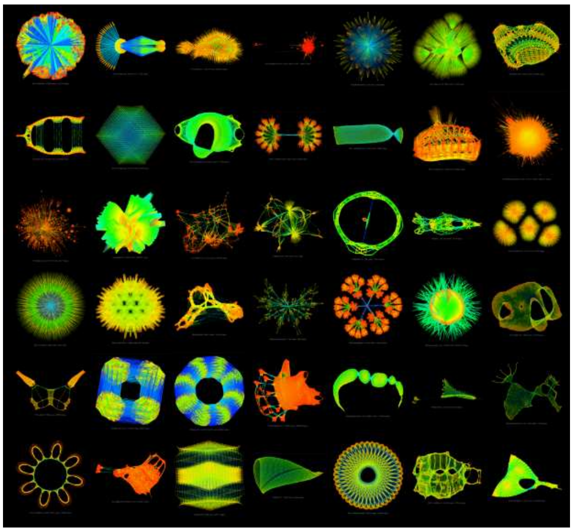

Figure 1.1: Visualization of various large scale systems using the sfdp algorithm © Dr. Yifan Hu of AT&T Labs

There are many tools available for the analysis of interconnected dynamical systems, for example in power systems you have PSSE and Power Factory. However, this simulation programs are normally very system specific and in most cases it takes a long time to learn how to use them correctly. The difficulties are specially noticed while testing control concepts, where small changes on the topology of the grid or control concept could lead to a painful redesign of your simulation set up. You may actually end up spending the most of your time in the implementation of a simulation. A more general and easy to use solution for the simulation of interconnected dynamical systems is needed.

1.2 Idea and Goal

MTIDS (Matlab Toolbox for Interconnected Dynamical Systems) is a project that aims to design an easy to use and flexible toolbox to make the simulation of large scale dynamical systems easier for students and researchers. The **goal** is to produce a mash-up that wraps different toolboxes used for graph analysis and dynamic systems simulation together into a framework.

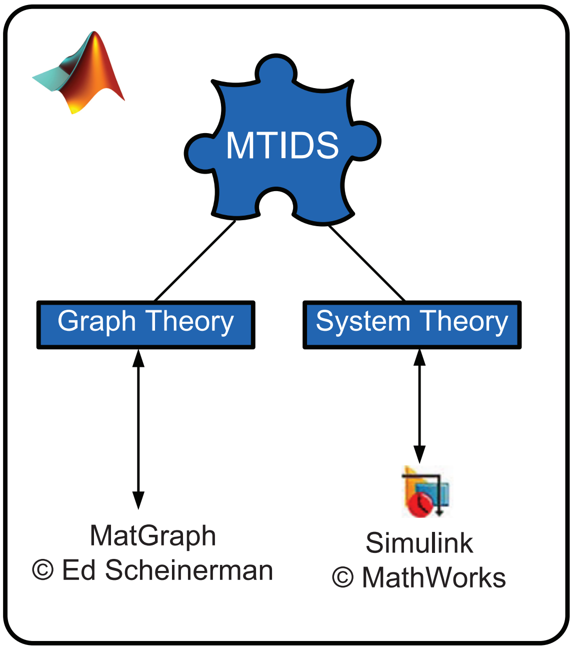

Figure 1.2: MTIDS: Matlab Toolbox for Interconnected Dynamical Systems

As we can see in Figure 1.2 MTIDS runs in the MATLAB enviroment and is basically a GUI that allows the interaction of tools used in graph theory and control theory. For graph theory we use Matgraph a toolbox design by Prof. Scheinerman of the John Hopkins University and for dynamical simulations we use Simulink.

1.3 Framework

The current framework of MTID is made out of three basic composnents. A GUI (**mtids.m**) an export to simulink function (**exportSimulink.m**) and an inport from Simulink function (**importSimulink.m**).

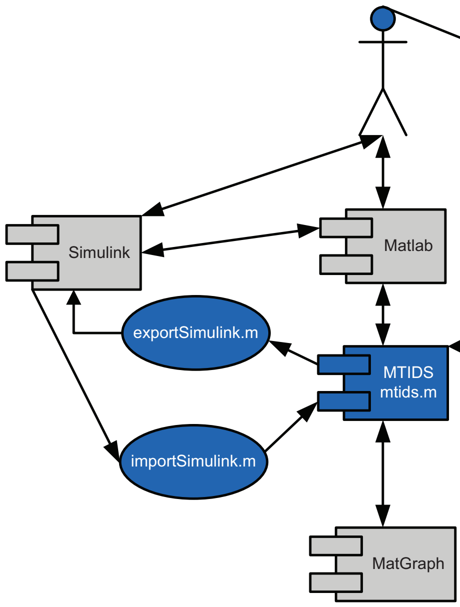

Figure 1.3: MTIDS: Components diagram

In Figure 1.3 we can see that the most important component is the user, specially its head. The better you are at producing templates and interacting with matlab and simulation the more functional MTIDS is going to be for you. In a nutshell MTIDS works as fallows:

- GUI (mtids.m) runs inside Matlab
- GUI interacts with Matgraph: create, modify and visualize graphs
- System Inteconnector (SI): exportSimulink.m and importSimulink.m called from GUI to interact with Simulink
- Templates done by User in Matlab/Simulink.
- Simulations done in Simulink

Chapter 2

Graph Theory

Chapter 3

System Theory

Chapter 4

MTIDS User Manual

4.1 MTIDS Concepts

4.1.1 Nodes and connections

Node:

Each node represents a subsystem . Each node has to have a unique label, Removing or adding a node changes the numbering of a node.

Conections:

Any connection made inside MTIDS is bidirectional and unweighted. Removing a node removes all connections coming from and to it.

4.1.2 Node dynamics and mdl-templates

Node dynamics are implemented as Simulink model files, this allows for implementation of systems which are not only limited to the traditional linear state-space models, but also to nonlinear and switching dynamics.

The MTIDS-Format for templates

4.1.3 Layering and clustering

4.2 User Interface

Opening MTIDS

Inside Matlab, change the current directory to the one which contains the *mtids.m* script Run the *mtids.m* script

4.2.1 Menu overview

File

View

Dynamics

Simulation

About

4.2.2 Drag-and-Drop operations

Adding nodes

Nodes are added using `<Shift>-<Left click>`. This adds a new node with label and dynamics as selected on the "node creation" section.

Drag nodes

Nodes can be dragged and repositioned on the screen by holding down the `<Left click>` buttom on the mouse.

Add/remove connections

Connections are added or removed using `<Ctrl>-<Left click>` on both nodes. If a connection doesn't exist, it will be created, otherwise it will remove this connection.

4.2.3 Access node properties

The node properties window is accessed by a `<Double Left click>` on the corresponding node.

4.2.4 Add mdl-templates

4.2.5 Adding and removing nodes

4.2.6 Adding and removing connections

4.2.7 Editting node properties

4.3 Interacting with Simulink

4.3.1 Exporting to Simulink

4.3.2 Creating and modifying templates

4.3.3 Layering

4.3.4 Simulation

4.4 Interacting with Matlab

4.4.1 Linear Algebra

4.4.2 Eigenvector and eigenvalue extraction of matrices

Ranks of matrices and nullspaces

4.4.3 Control Toolbox functions

Linearizing and testing for controlability

Chapter 5

Conclusion and Future Development

5.1 Conclusion

5.2 Future Development

List of Figures