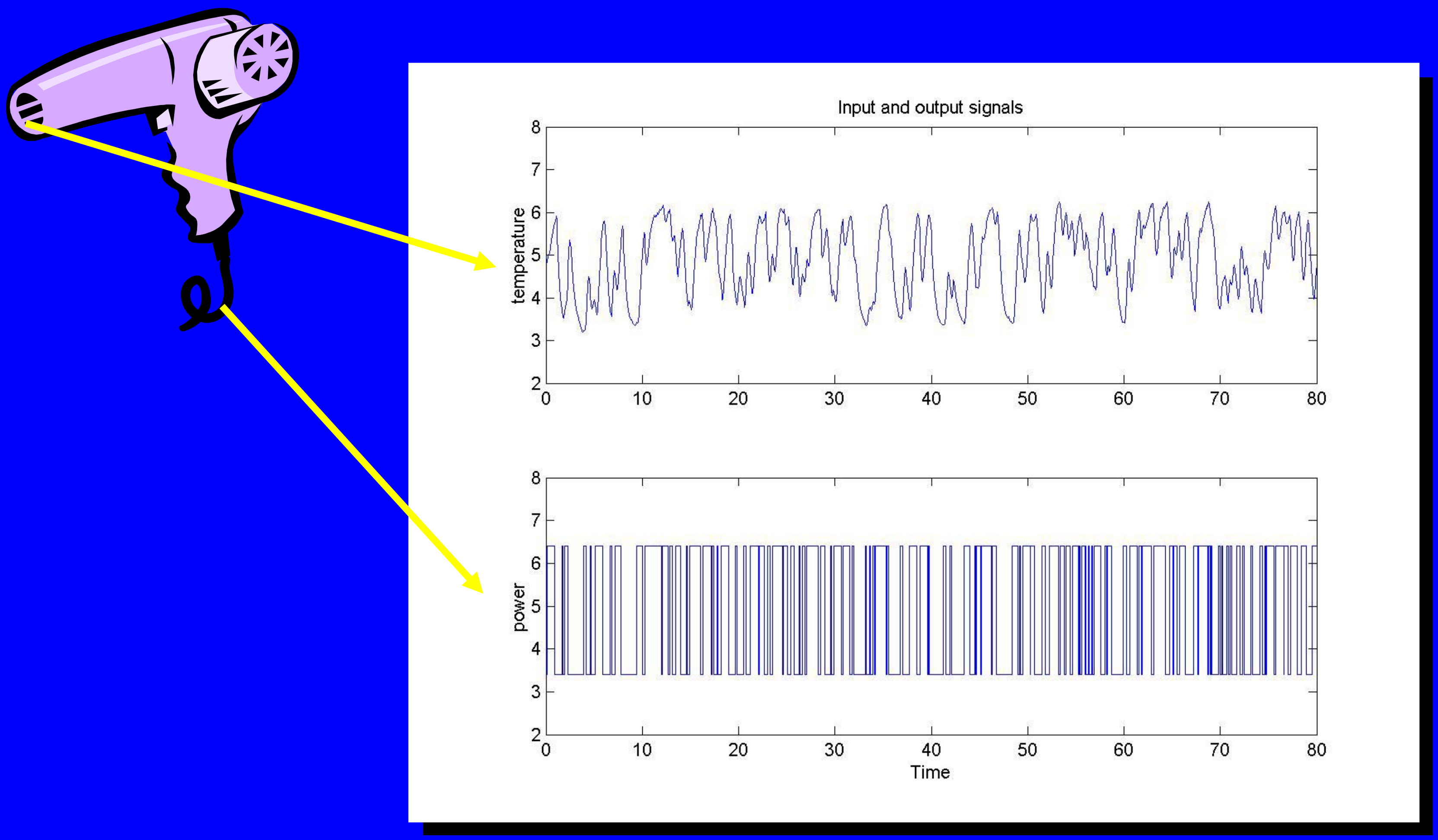
Project
T-61.6050

Amaury Lendasse, Jaakko Hollmén and Jarkko Tikka

Time	Lecturer	Subject
14.9.	Amaury Lendasse	Presentation of the course
21.9.	Elia Liitiäinen Esa Seuranen	Introduction to ANN and System Identification p.1-18 + paper 1
28.9.	Vibhor Kumar Sven Laur	Model Structure Selection p.18-37 + paper 1
5.10	Yoan Miché Kei Takahashi	Experiments and Determination of Weights p.38-84
12.10	Rami Rautkorpi Ali Pekcan	Validation Procedure and Summary p.85-119
19.10	-	Project
26.10	-	Project (no lecture)
2.11	-	Project
9.11	Mikael Pohjola Janne Pylkkönen	Introduction + Direct Inverse Control + IMC p.121-142
16.11	Antti Yli-Krekola Eemeli Aro	Feedback Linearization + Feedforward Control + Optimal Control + CBIL p. 143-175
23.11	Antti Sorjamaa Ji Yongnan Nima Reyhani	Predictive Control + Recapitulation + Case Study p. 178-233
30.11		Presentations of projects (20 minutes per group)

Deadline for the project: November 9th at 2pm.

System to Identify and to control



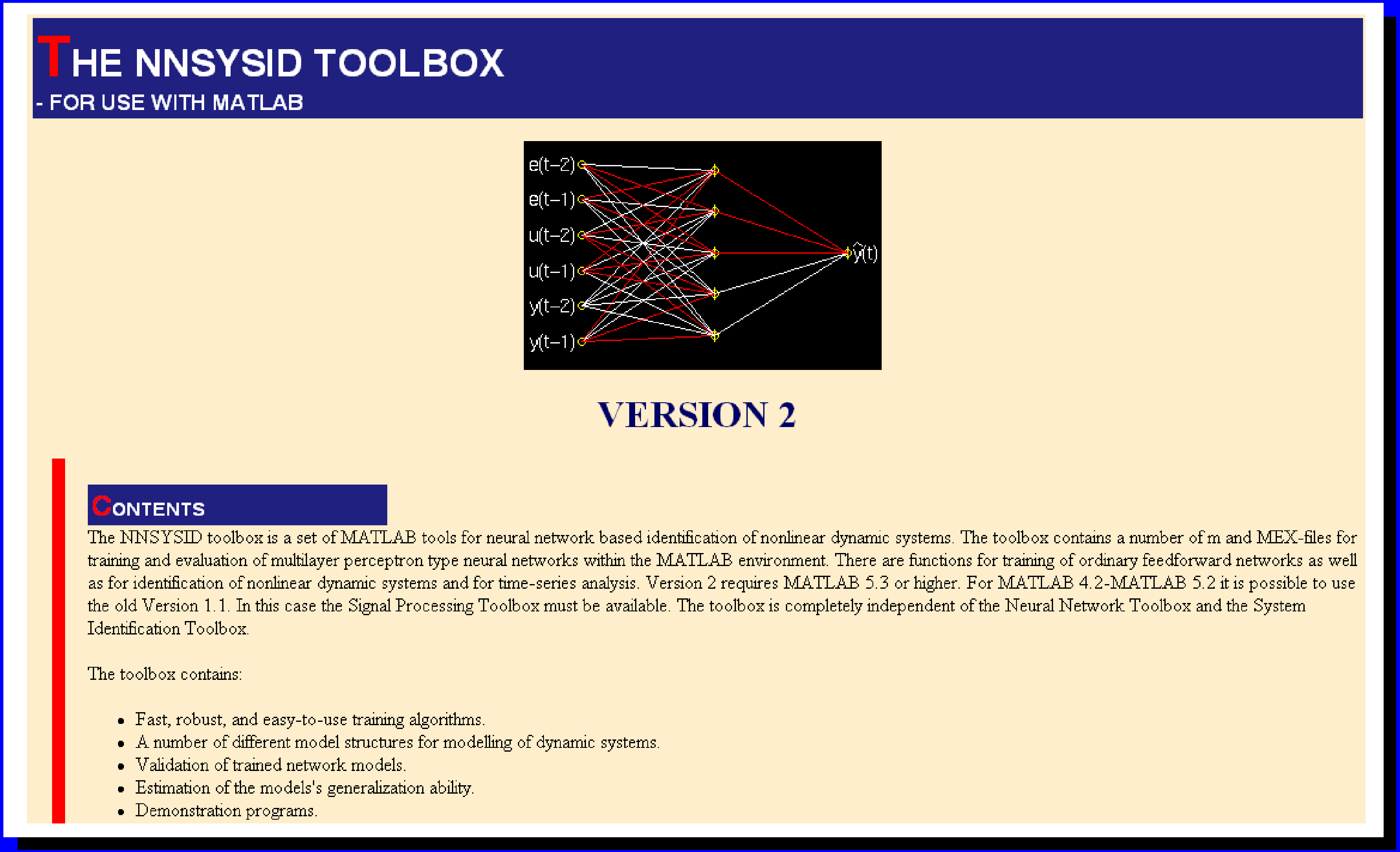
System to Identify and to control

Date available on:
http://www.cis.hut.fi/Opinnot/T-61.6050/dryer2.mat

You have to:

1) Identify the system (using NNSYSID Toolbox).
2) Build at least 3 controllers for the system (using NNCTRL Toolbox). Use the Neural Network to simulate the model.

Toolboxes: http://kalman.iau.dtu.dk/research/control/nnsysid.html



THE NNSYSID TOOLBOX

- FOR USE WITH MATLAB

VERSION 2

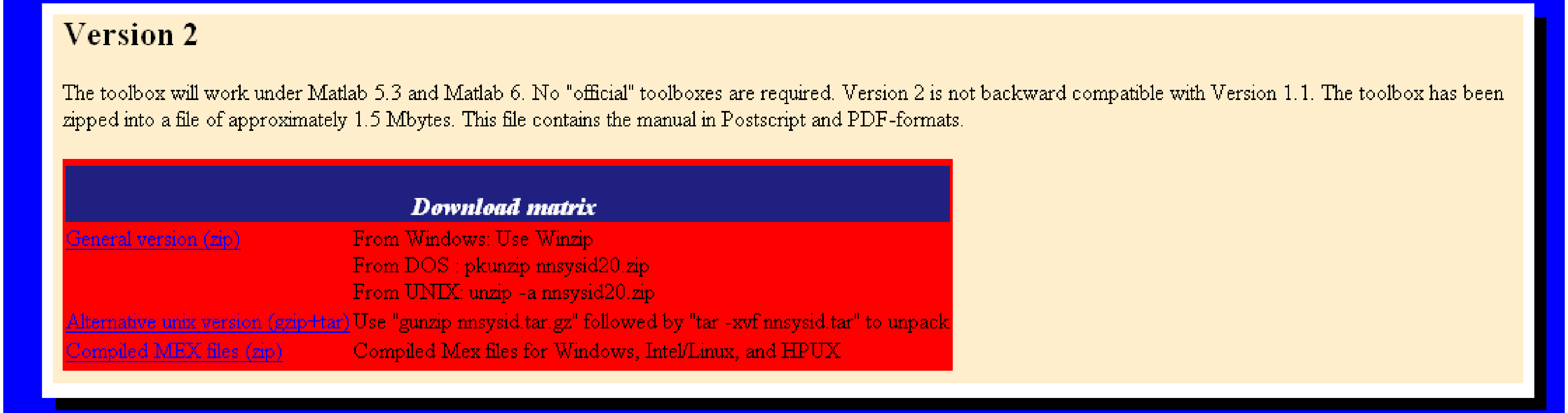
CONTENTS

The NNSYSID toolbox is a set of MATLAB tools for neural network based identification of nonlinear dynamic systems. The toolbox contains a number of m and MEX-files for training and evaluation of multilayer perceptron type neural networks within the MATLAB environment. There are functions for training of ordinary feedforward networks as well as for identification of nonlinear dynamic systems and for time-series analysis. Version 2 requires MATLAB 5.3 or higher. For MATLAB 4.2-MATLAB 5.2 it is possible to use the old Version 1.1. In this case the Signal Processing Toolbox must be available. The toolbox is completely independent of the Neural Network Toolbox and the System Identification Toolbox.

The toolbox contains:

- Fast, robust, and easy-to-use training algorithms.
- A number of different model structures for modelling of dynamic systems.
- Validation of trained network models.
- Estimation of the models's generalization ability.
- Demonstration programs.

Toolboxes: http://kalman.iau.dtu.dk/research/control/nnsysid.html

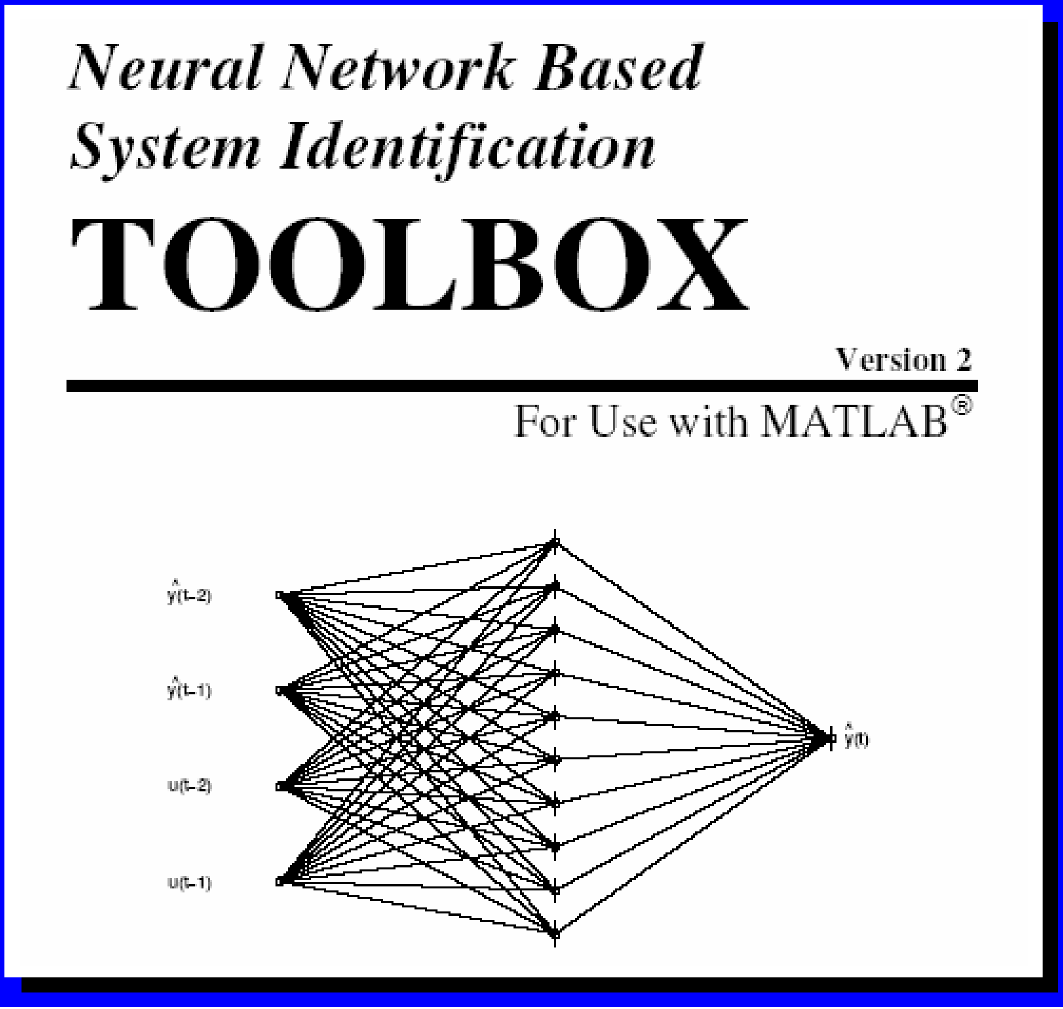


Version 2

The toolbox will work under Matlab 5.3 and Matlab 6. No "official" toolboxes are required. Version 2 is not backward compatible with Version 1.1. The toolbox has been zipped into a file of approximately 1.5 Mbytes. This file contains the manual in Postscript and PDF-formats.

Download matrix	
General version (zip)	From Windows: Use Winzip From DOS : pkunzip nnsysid20.zip From UNIX: unzip -a nnsysid20.zip
Alternative unix version (gzip+tar)	Use "gunzip nnsysid.tar.gz" followed by "tar -xvf nnsysid.tar" to unpack
Compiled MEX files (zip)	Compiled Mex files for Windows, Intel/Linux, and HPUX

NNSYSID: please read the manual1.pdf



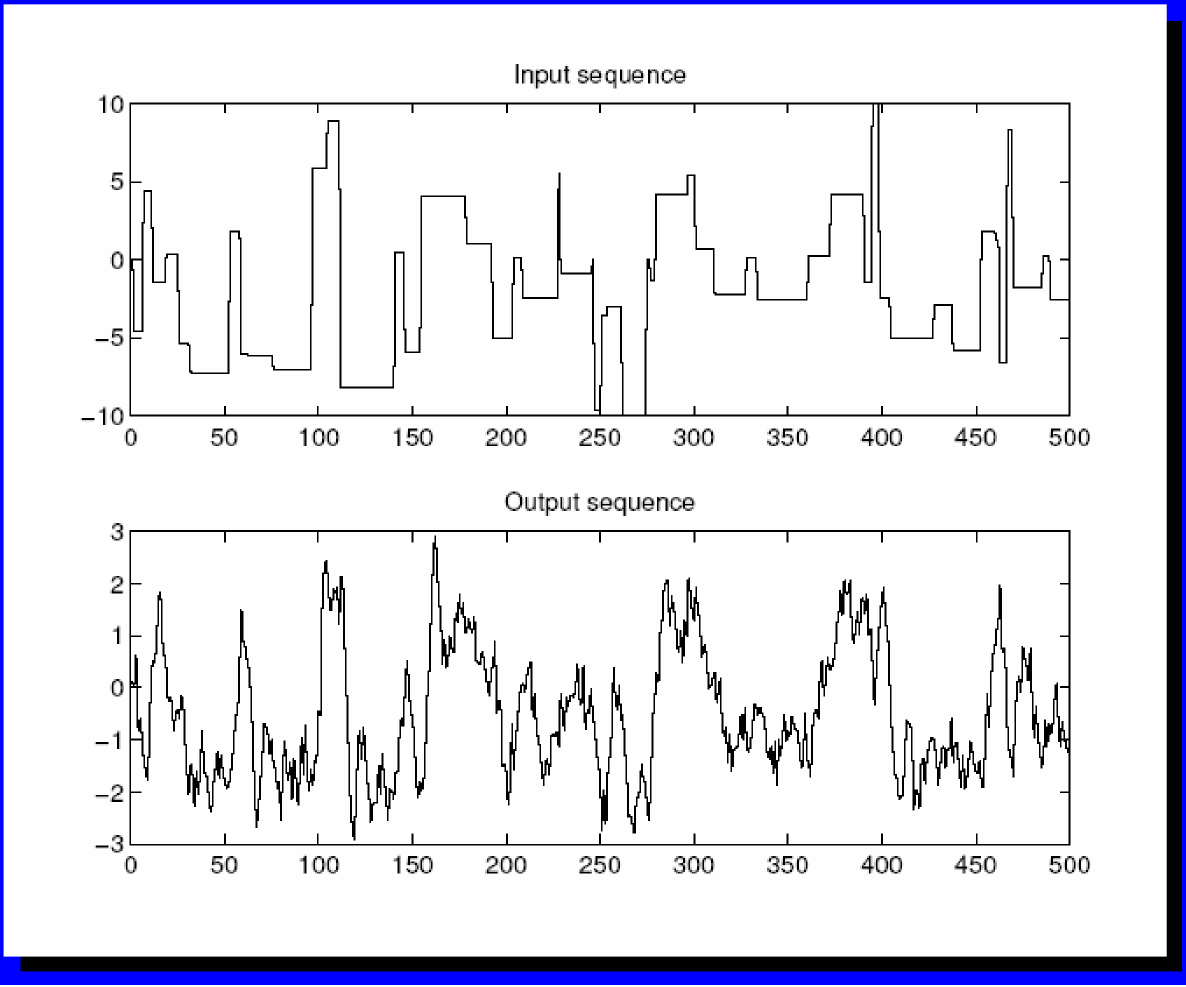
Neural Network Based System Identification

TOOLBOX

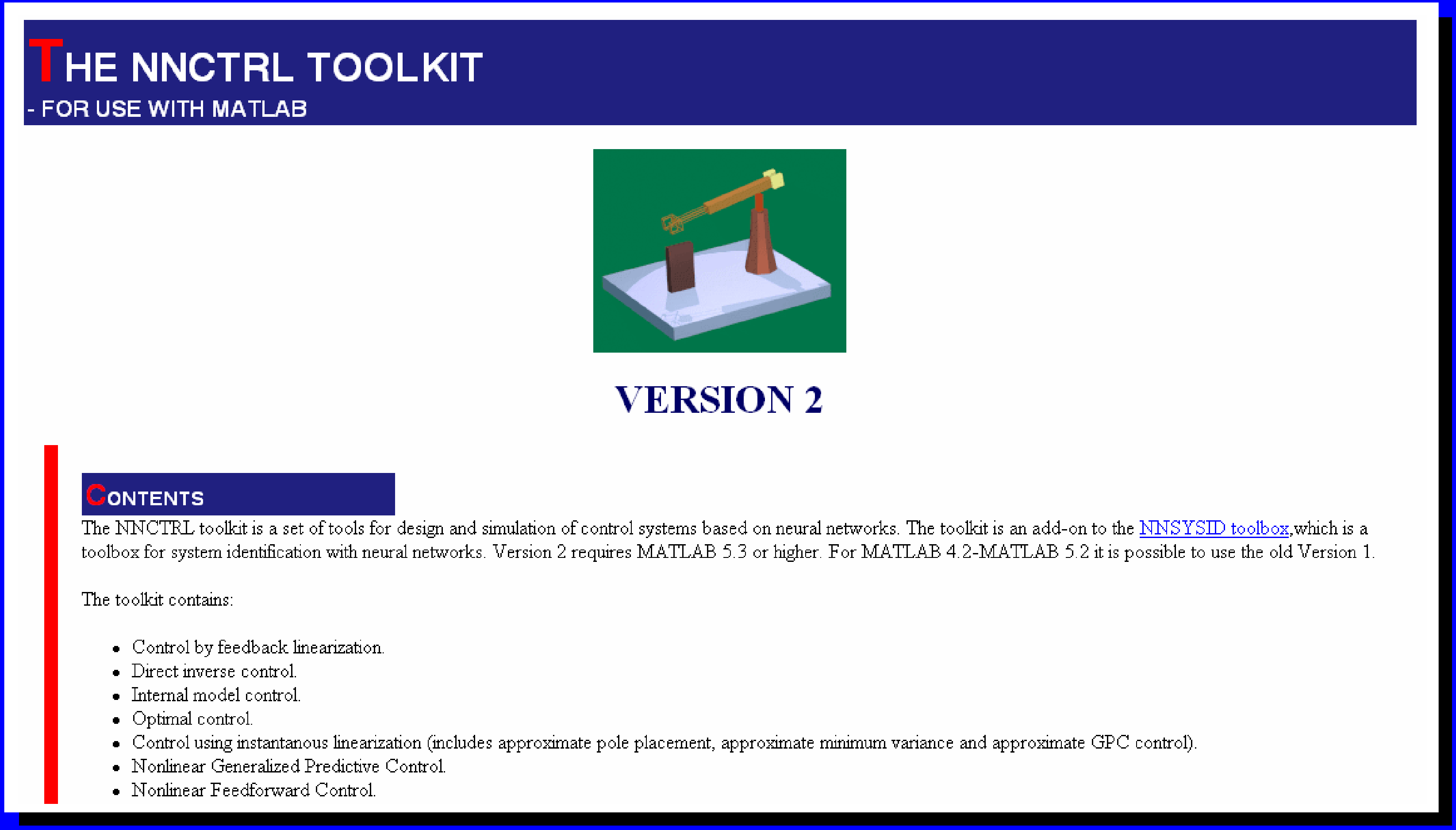
Version 2

For Use with MATLAB®

NNSYSID: and read and redo the Example



Toolboxes: http://www.iau.dtu.dk/research/control/nnctrl.html



THE NNCTRL TOOLKIT

- FOR USE WITH MATLAB

VERSION 2

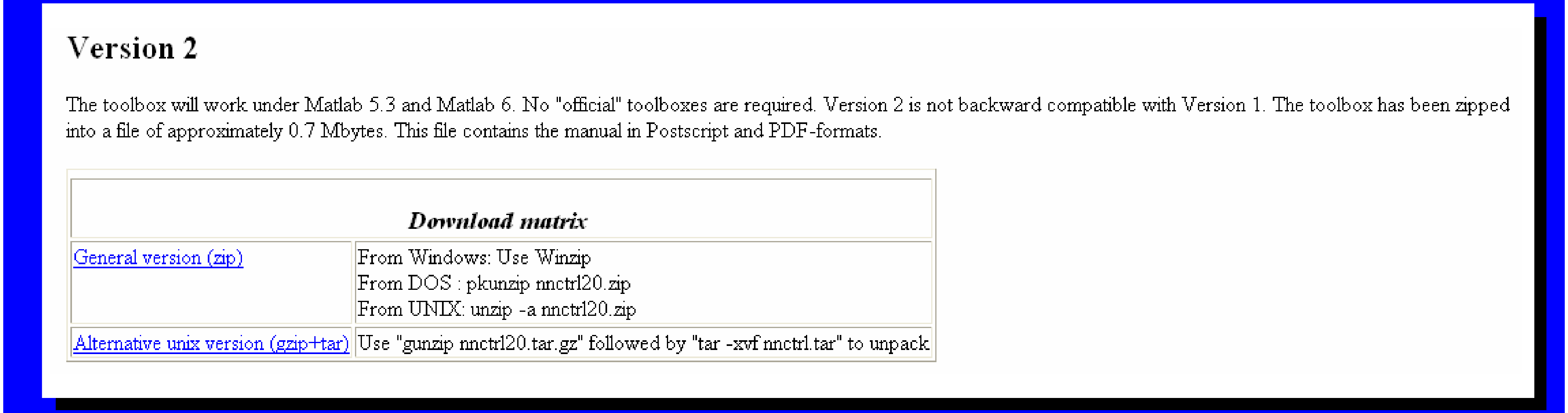
CONTENTS

The NNCTRL toolkit is a set of tools for design and simulation of control systems based on neural networks. The toolkit is an add-on to the NNSYSID toolbox,which is a toolbox for system identification with neural networks. Version 2 requires MATLAB 5.3 or higher. For MATLAB 4.2-MATLAB 5.2 it is possible to use the old Version 1.

The toolkit contains:

- Control by feedback linearization.
- Direct inverse control.
- Internal model control.
- Optimal control.
- Control using instantanous linearization (includes approximate pole placement, approximate minimum variance and approximate GPC control).
- Nonlinear Generalized Predictive Control.
- Nonlinear Feedforward Control.

Toolboxes: http://www.iau.dtu.dk/research/control/nnctrl.html

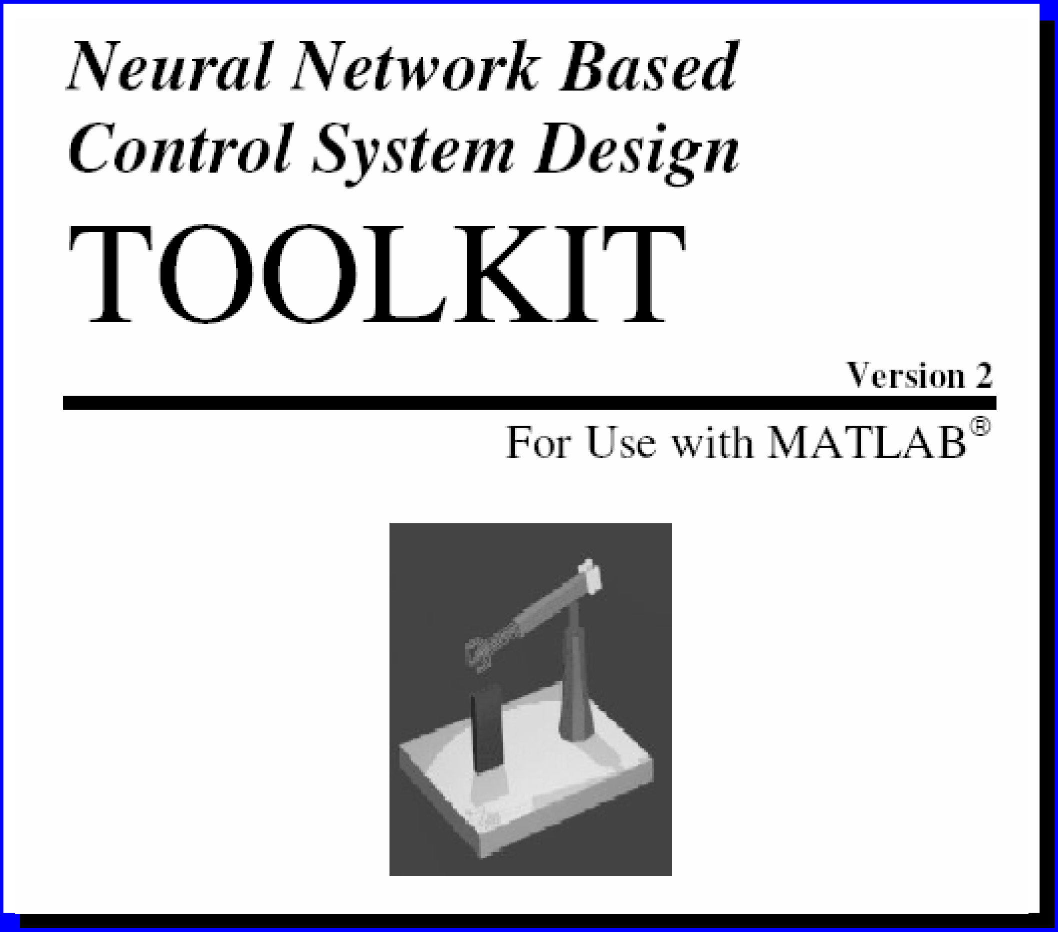


Version 2

The toolbox will work under Matlab 5.3 and Matlab 6. No "official" toolboxes are required. Version 2 is not backward compatible with Version 1. The toolbox has been zipped into a file of approximately 0.7 Mbytes. This file contains the manual in Postscript and PDF-formats.

Download matrix	
General version (zip)	From Windows: Use Winzip From DOS : pkunzip nnctrl20.zip From UNIX: unzip -a nnctrl20.zip
Alternative unix version (gzip+tar)	Use "gunzip nnctrl20.tar.gz" followed by "tar -xvf nnctrl.tar" to unpack

NNSYSID: please read the manual.pdf



Neural Network Based
Control System Design

TOOLKIT

Version 2

For Use with MATLAB®

NNSYSID: please read the manual.pdf

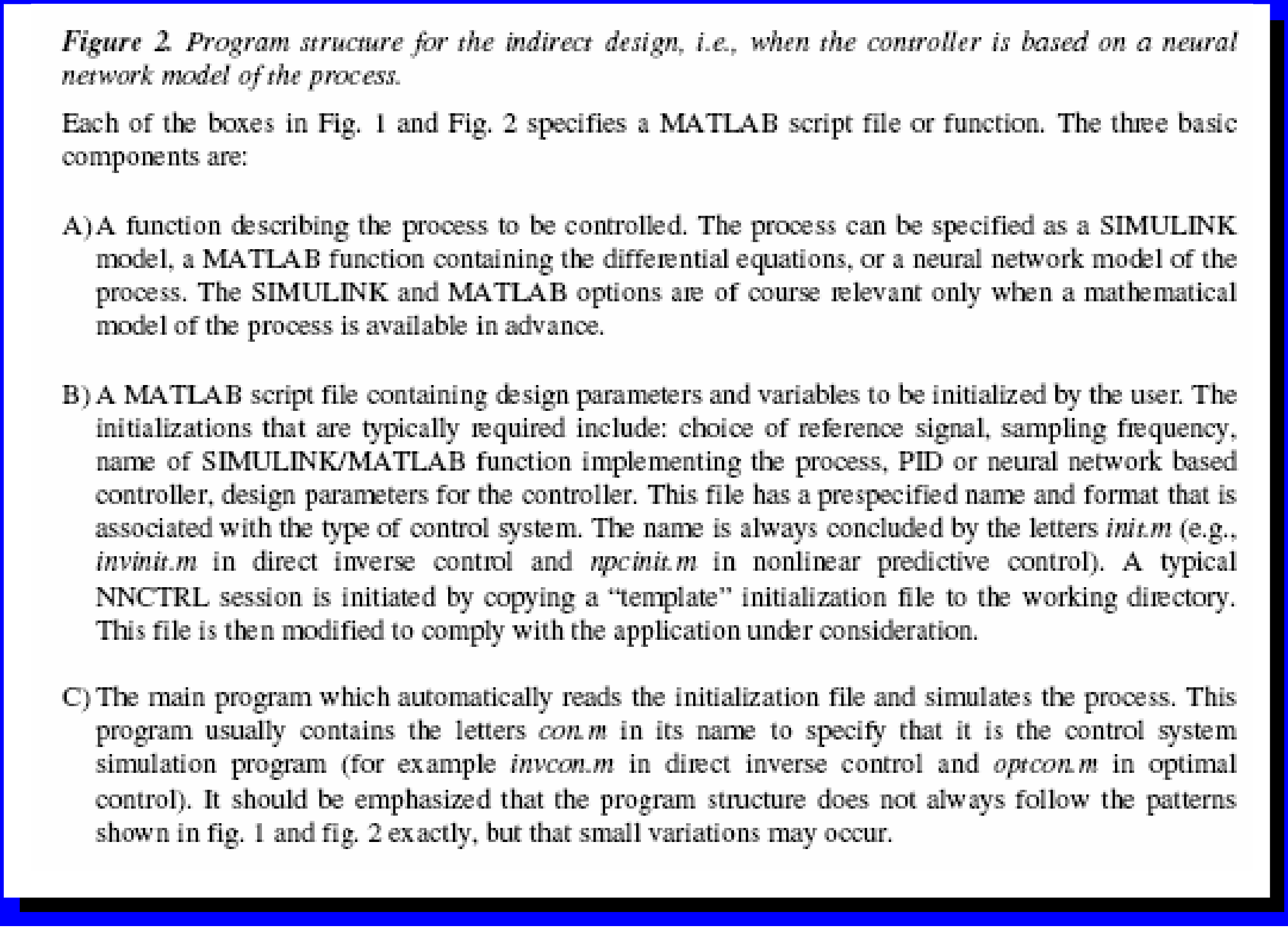


Figure 2. Program structure for the indirect design, i.e., when the controller is based on a neural network model of the process.

Each of the boxes in Fig. 1 and Fig. 2 specifies a MATLAB script file or function. The three basic components are:

A) A function describing the process to be controlled. The process can be specified as a SIMULINK model, a MATLAB function containing the differential equations, or a neural network model of the process. The SIMULINK and MATLAB options are of course relevant only when a mathematical model of the process is available in advance.

B) A MATLAB script file containing design parameters and variables to be initialized by the user. The initializations that are typically required include: choice of reference signal, sampling frequency, name of SIMULINK/MATLAB function implementing the process, PID or neural network based controller, design parameters for the controller. This file has a prespecified name and format that is associated with the type of control system. The name is always concluded by the letters *init.m* (e.g., *invinit.m* in direct inverse control and *npcinit.m* in nonlinear predictive control). A typical NNCTRL session is initiated by copying a "template" initialization file to the working directory. This file is then modified to comply with the application under consideration.

C) The main program which automatically reads the initialization file and simulates the process. This program usually contains the letters *con.m* in its name to specify that it is the control system simulation program (for example *invcon.m* in direct inverse control and *optcon.m* in optimal control). It should be emphasized that the program structure does not always follow the patterns shown in fig. 1 and fig. 2 exactly, but that small variations may occur.

Project

- Deadline for the project: November 9th at 2pm.
- Same groups than for the presentations.
- Use NNSYSID to identify the system (quite easy)
  Use NNCTRL to control the system (more difficult)
- For the report:
  - Format of a scientific paper
    (Abstract, Introduction, Methods, Experiments, Conclusion, References, …)
  - 12 pages maximum
  - Use this Latex style:
    http://www.dice.ucl.ac.be/esann/index.php?pg=cfp#latexstylefile
  - Show that you understand the theory presented in the book.