AMS

Auditory Modelling System

Dr. Lowel P O'Mard

lpo21@cam.ac.uk

AMS: Auditory Modelling System
by Dr. Lowel P O'Mard

Published 8th January 2007

Revision History

Revision 1.2-1 8th Jan 2007 Revised by: LPO
Revised.

Dedication

I dedicate this work to my wife Rachel, for her love and support and to our beautiful twin girls, Anesu and Shamiso who provide us with such joy.

Acknowledgements

I would first like to acknowledge God as the creator of all things and without whose support none of this would be possible. I would also like to thank all those who have contributed to the making of this document.

Table of Contents

Preface

The AMS application was produced using the Development System for Auditory Modelling (DSAM). It's facilities and interface has many features in common with other applications produced using DSAM.

Chapter 1. Getting started with AMS

Introduction

The auditory model simulator (AMS) allows users to design, run and evaluate auditory models. It is backed up by the extensive collection of published and new models supplied by the development system for auditory modelling (DSAM), formerly known as LUTEar.

Installation

The AMS is available in executable form for PC (Windows(tm) 95/98 NT) and RPM Linux platforms (at the time of writing). These packages will allow you to install AMS and start using it straight away. The source code for DSAM and AMS is also available and can be used for compiling AMSon other systems, including Mac OS X. The executable installations and source code are available from the the web site: www.dsam.org.uk.

The Linux binary RPM is dependent upon other pacakges being installed in addition to the DSAM RPM which is also available from the web site. These libraries are available as a standard part of most linux distributions, and are also readily available as source code packages.

Package	Source	Description
DSAM	www.dsam.org.uk	Development System for Auditory Modelling library.
wxWidgets	www.wxwidgets.org	wxWidgets lets developers create applications for Win32, Mac OS X, GTK+, X11, Motif, WinCE, and more using one codebase.
Libsndfile	www.mega-nerd.com/libsndfile/	This is a C library which provides support for reading and writing files containingsampled sound.
PortAudio	www.portaudio.com	PortAudio is a free, cross platform, audio I/O library.

Quick start example simulation

Introduction

This is a good starting point for those users who wish to get a quick feel for AMS and how useful it will be. This example takes the user through loading an auditory nerve simulation then running it to produce output, providing displays of the simulation process at different stages. The user is then shown how to change parameters in the simulation which will be reflected by a change in the output. It also serves as a introduction to the AMS interface which is described in more detail later in this document.

Starting AMS

Starting the AMS application will be different on different platforms. On Windows installations, the application can be found by navigating from *start* on the task bar: *start->All Programs->DSAM Applications*. On unix-like systems, AMS can be started from the command line using the command *ams*. Once started the main AMS window will appear as show in Figure 1-1.

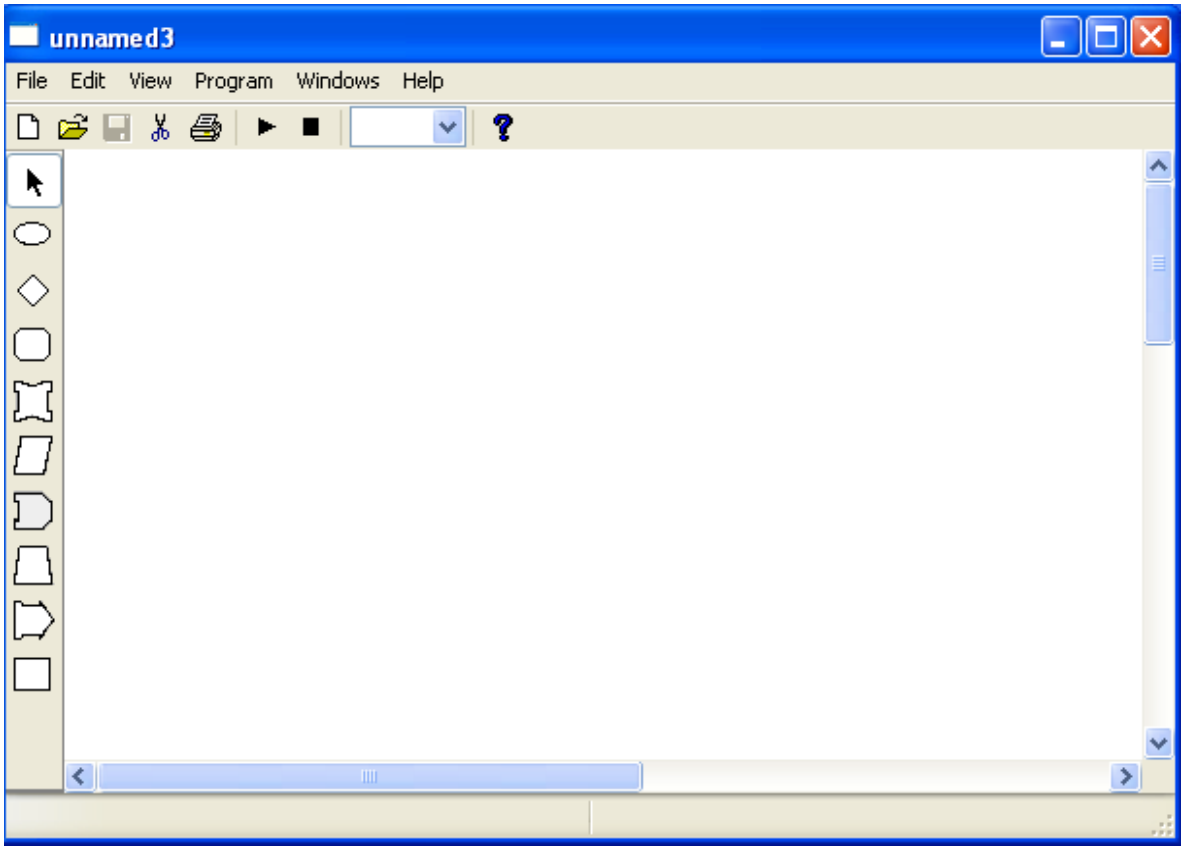

Figure 1-1. The main AMS window

Loading an example simulation into AMS

The first step is to load the example auditory nerve simulation from the file "auditoryNerve.spf" found in the "Tutorials/AuditoryNerve" directory. From the *File* menu on the menu bar select the *Open* option. This will open the *Select a file* dialog window as shown in Figure 1-2. Ensure that the "Simulation Parameter File (*.spf)" option is selected for the "Files of type", then open the "auditoryNerve" file (the ".spf" extension is not normally shown in Windows installations).

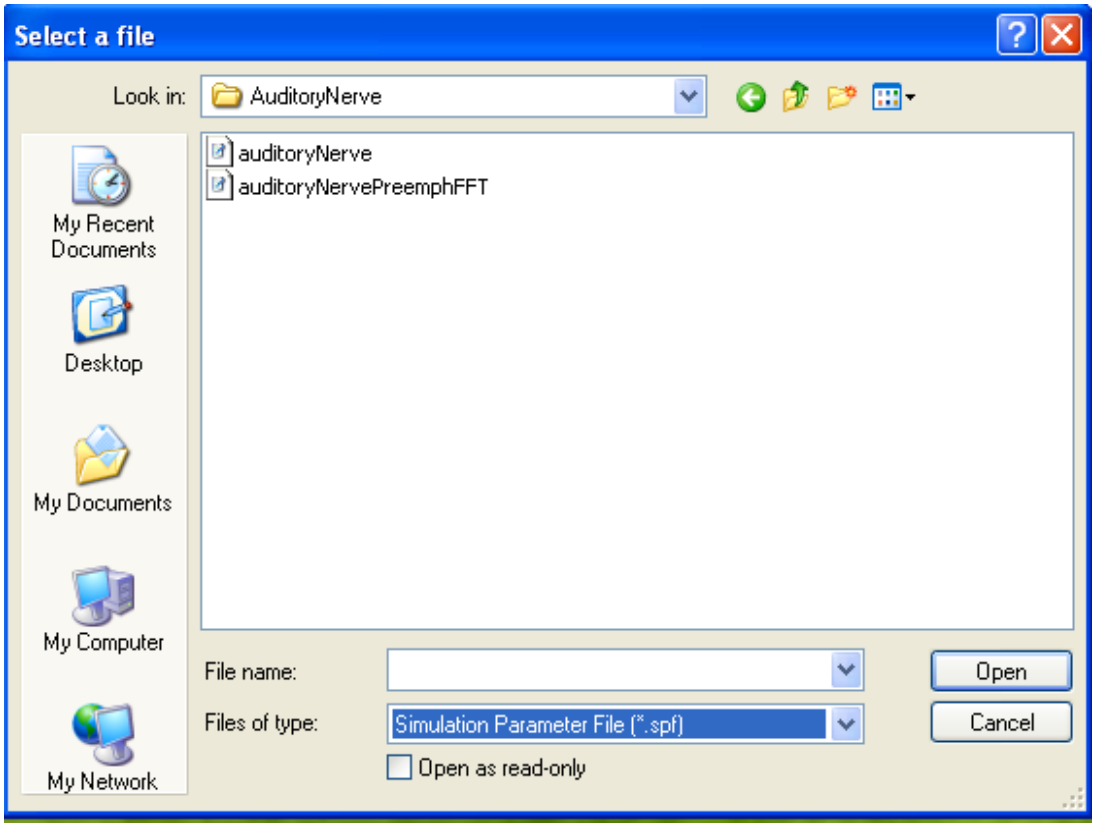

Figure 1-2. Select a file dialog window.

Once loaded the AMS main window will look as show in Figure 1-3. Each shape in the simulation design represents a process and the execution/data direction is shown by the connection arrows. The shape denotes the class of process and the central label gives the process' specific name. For example, the first process is the "Stim_Harmonic" stimulus generation module which is in the I/O class. The process also has an additional label ("p0") which will be explained more fully later in this document.

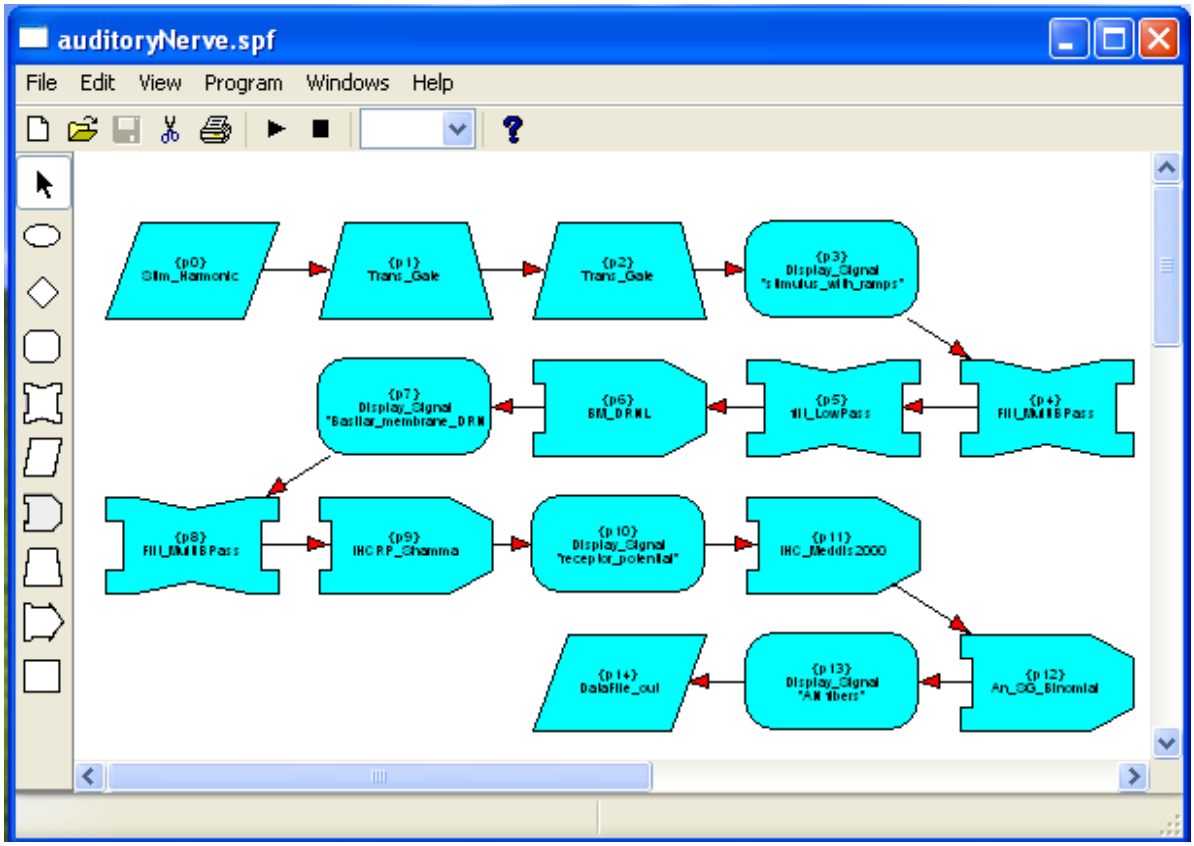

Figure 1-3. Example auditory nerve simulation.

Running the example simulation

Run the simulation by processing the media-player style "play" button as indicated in Figure 1-4. When run four displays will be produced as shown in Figure 1-5. These displays correspond to the display processes in the simulation, labelled, "p3", "p7", "p9" and "p13". The process shapes also show the display window labels.

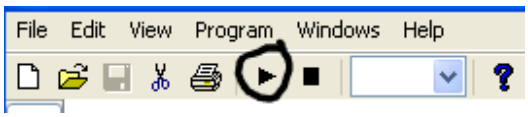

Figure 1-4. Play button to run the simulation

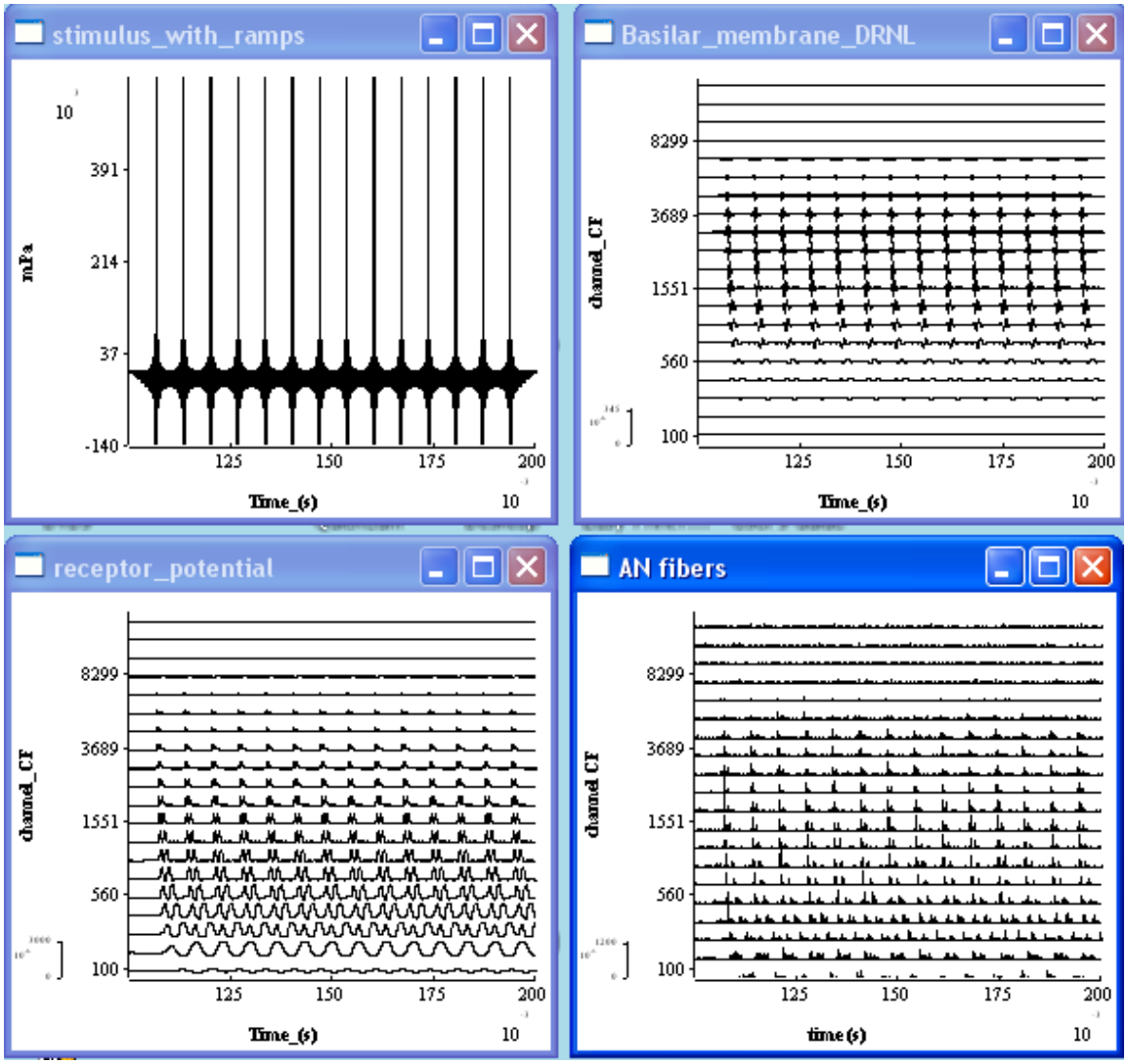

Figure 1-5. Auditory nerve example simulation displays

What the example simulation does

The example simulation describes a non-linear auditory nerve model. The first four processes create, gate then display the stimulus. The stimulus display is the top left display in Figure 1-5. The next two processes are filter-class processes which are used to model the outer-/middle-ear. The following "BM_DRNL" process is a non-linear model of basilar membrane processing. This models the frequency selectivity of the basilar membrane, and as shown in Figure 1-5 separates processing into multiple channels. Each channel represents the response at a particular characteristic frequency. The next two stages in the simulation model the response of the inner hair-cell, which is the point at which mechanical to neural transduction takes place. The output from this process is shown in the bottom left display of Figure 1-5. The bottom right display shows spike output of the auditory-nerve process which takes as its input the spike probabilities from the previous process.The final process outputs the auditory-nerve spike information to a file.

Changing parameters

It is possible to examine and change the parameters of processes shown in the *SimulationDesign Window*. Double-click on the "BM_DRNL" process shape. This will open the process' properties dialog as show in Figure 1-6. Hover over a par-

ticular parameter to produce an additional description for the parameter. Parameters that cannot be changed for a particular mode choice will be greyed-out. Change the "channels" parameter value from 20 to 10 channels then run the simulation. There is no need to close the properties dialog before running the simulation again. The displays will change reflecting the reduction in the properties dialog "channels" parameter.

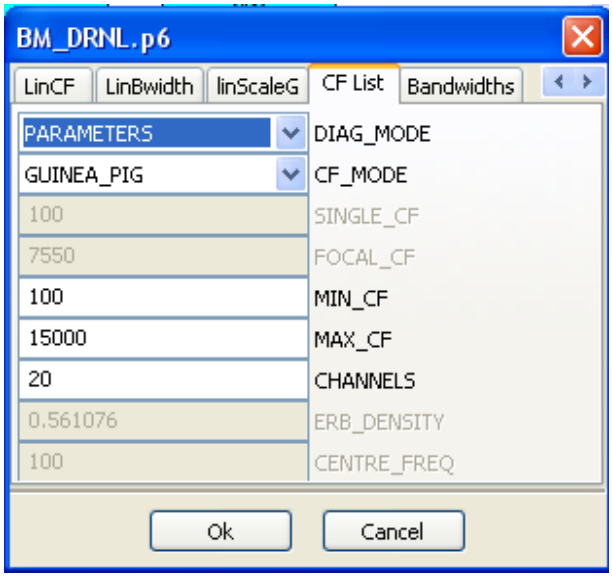

Figure 1-6. BM_DRNL process properties dialog

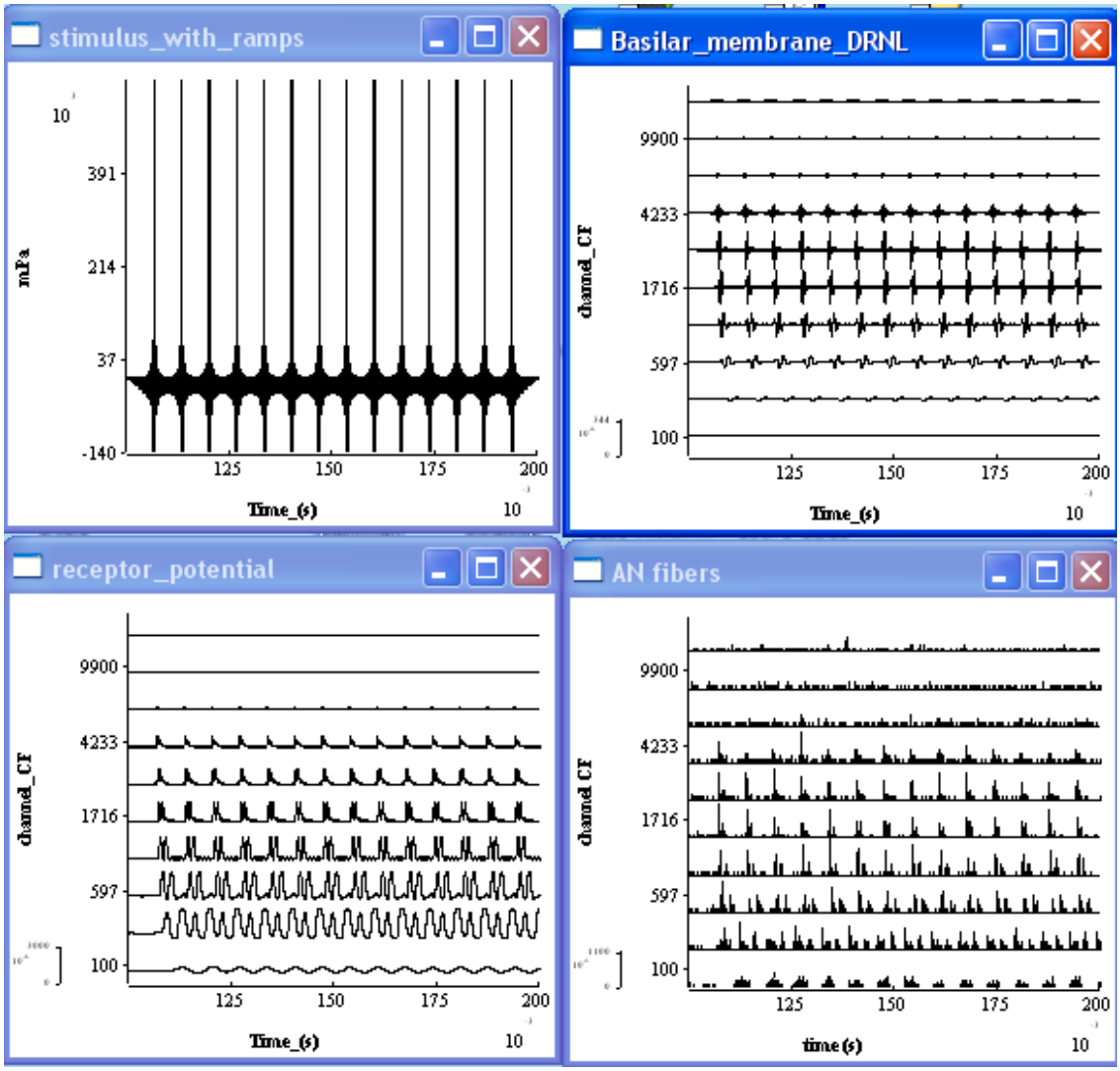

Figure 1-7. Example simulation displays after reducing the number of channels

Quick start simulation creation example

Introduction

In this example the user is carried through the process of creating a simulation from scratch using the Simulation Design Interface (SDI). Building a simulation from scratch requires an understanding of the auditory system, to provide the user with a clear idea of what they trying to achieve. A good method is to start with an established model and develop the simulation using that model as the basis. This is the approach that will be used in this example, where we will build a simulation using the Meddis auditory-nerve model ([Meddis86].

Chapter 2. Main Application Window

Application Menus

File Menu

Edit Menu

View Menu

Pogram Menu

Windows Menu

Help Menu

Tool Bar

Simulation Design Window

Process Palette

Bibliography

Meddis R., (*1988*) "Simulation of mechanical to neural transduction in the auditory receptor" *J. Acoust. Soc. Am.* [79], 702-710.

Index