

Demoplayer Interface

25. January 2010

Software, terminology

The *demoplayer software* consists of:

- demoplayer.p
- demoplayercb.p
- demoplayerget.p
- demoplayerset.p
- demoplayer.mat

Additionally, for any special demo (e.g. the „Signal Space“ or „Eye Pattern“, see demoplayer help) there is a *demo software*. While the *demoplayer software* is used as the framework for a demo, the *demo software* is special and depends on the topic to demonstrate. So a *demo* is realized by a *demo software* and the *demoplayer* is realized by the *demoplayer software*.

Additional routines from the matoffice package are needed to run demoplayer, e.g. *uniplayer routines*. All routines can be found in their corresponding matoffice directory. There is no need to take care of individual routines if the matoffice zip package is installed, see demoplayerHelp.pdf.

Graphical user interface

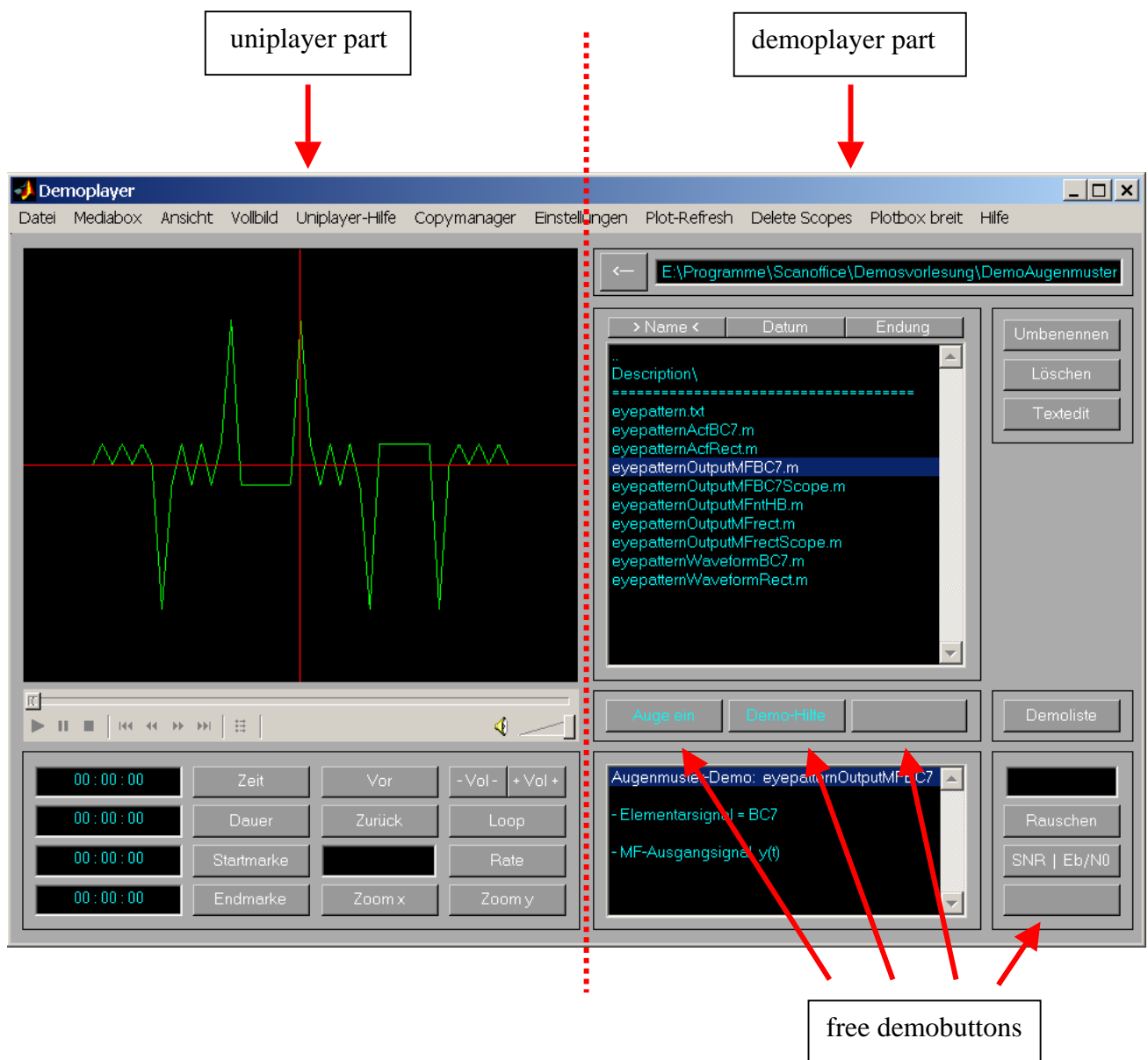
The demoplayer.p routine starts the demoplayer graphical user interface (GUI). It consists of the following parts:

- a) the demoplayer figure window,
- b) demoplayer graphical objects (GO),
- c) uniplayer graphical objects.

See image on next page. The uniplayer GOs are on the left, the demoplayer GOs on the right.

The uniplayer part is identical with uniplayer itself, with the difference to uniplayer stand-alone, that the GOs are within the demoplayer figure window. So, after starting „uniplayer“ with Matlab a GUI occurs, which is only the left hand part of the demoplayer GUI. See uniplayer help for more details.

Both GUIs (demoplayer and uniplayer) have their own callback routines (demoplayercb and uniplayercb) which are called when clicking with a mouse button on a GO, e.g. clicking on a push button. A speciality of demoplayercb is, that inside first the routine uniplayercb is called to serve callbacks from the uniplayer part at the left. If the callback is caused by a GO from the demoplayer part, the uniplayercb call within demoplayercb has no effect.



Interface between demo software and demoplayer software

The software for a special demo (e.g. „Signal Space“) uses the demoplayer GUI. All graphics, animations and video should run within the big window at the left (the *demo window*). Audio and video are played with the integrated Windows media player (mplayer2), see GUI part below the demo window.

The interface between a special demo software and the demoplayer software is realized by „set routines“ and „get routines“: demoplayer/set/get belong to the GOs at the right, uniplayer/set/get to the GOs at the left. The two uniscope routines (uniscopeset/get) are special, there is no link to GO within the demoplayer window. The GUI routine

... \ Matoffice\Tools\too300svchelp.m

lists all possible calls of demoplayer/get/set and uniplayer/set/get and a short description.

How to program a demo

The basic structure of a demo m-file is as follows:

1. Ini and free demobutton callback section
2. demo routine
 - 2.1 plotbox ini
 - 2.2 SNR ini, if needed
 - 2.3 demo ini
 - 2.4 plot/animation loop ini
 - 2.5 plot/animation loop
 - 2.6 plot/animation loop finish
3. local routines

There is a sample Demo (“Test/External Use of Demoplayer”) which demonstrates how to program an animated demo. The key strings for `uniplayerset/get`, `uniscipeset/get` and `demoplayerset/get` are described in the appendix.

How to include a demo in the demolist

The *demolist* contains all those details which demoplayer needs to know for running a special demo. It is stored in the mat file *demoplayerparam.mat* and can be checked or modified with the demoplayer menu „Einstellungen – Demoliste bearbeiten“. A text editor is started for this purpose and the demolist is shown. It can be modified and after saving the modifications with the text editor the new demolist will be saved again in *demoplayerparam.mat*. The text editor is started in modal mode, i.e. Matlab waits until the text editor window is closed.

Beside a block of explanation comments (preceded by „%“) at the beginning, the demolist contains sections with demo specific information. The following example is for the Viterbi algorithm demo:

```
%-----Viterbi Algorithm:      TB1 for "In steps"
%                               PB2 for "Demo help"
%                               PB3 for "Step"
%                               PB4 for "0/1/random tx"
%
NAME=Viterbi-Algorithmus
TEXTFILE=viterbialgorithm.txt
INI=dem700viterbialgorithm('INI')
CALLBACK=dem700viterbialgorithm('CALLBACK-TB1')
CALLBACK=dem700viterbialgorithm('CALLBACK-PB2')
CALLBACK=dem700viterbialgorithm('CALLBACK-PB3')
CALLBACK=dem700viterbialgorithm('CALLBACK-PB4')
%KEYPRESSFCN=dem700viterbialgorithm('KEYPRESSFCN')%-----not used
%CONTEXTMENU1=dem700viterbialgorithm('CONTEXTMENU1')%--not used
```

The strings with capital letters at the left of „,=“ are mandatory, als the „,=“ itself. On the right the names / text strings can be defined appropriately with the exception of the capital letter key strings. They are also mandatory. TB1 stands for „toggle button 1“ which is the first free demo button. PB2 to PB4 are the free demo push buttons, see above. The key strings itself, e.g. 'CALLBACK-TB1', are used as argument in a call of the special demo software, i.e. a call of the m-file `dem700viterbialgorithm.m` in the example. In the NAME line Viterbi-Algorithmus is the demoname which appears in the demolist when pushing the demolist button in the demoplayer Gui, see above.

Within the demo routine, i.e. `dem700viterbialgorithm.m` in this example, there must be sections which are addressed by the key strings in the argument of a call (`'INI'` , `'CALLBACK-TB1'` , . . . , `'CALLBACK-PB4'`).

Remark:

Demoplayer is part of the more general Matlab SW called “Matoffice” which was created over many years. Starting with version 1.3 of the demoplayer the whole Matoffice package may also be used by others. For more details see `help/MatofficeReadme.pdf`