The Adaptive Signal Processing Toolbox For Matlab

Installation and Getting Started Guide.

This document contains information to help you install and configure the Adaptive Signal Processing Toolbox for use with Matlab. It also contains contact information, should you need support in installing the package.

CONTENTS

License Note

Note: This software is copyrighted and is protected by copyright laws and international treaty provisions. Make sure that you have read and understood the license terms and conditions included in the file 'asptlicense.pdf' before copying, installing, or using this software.

What is ASPT?

The Adaptive Signal Processing Toolbox (ASPT) is a software package for use with C/C++ and Matlab. ASPT contains a large collection of adaptive signal processing algorithms and many practical applications. ASPT therefore makes the simulation and implementation of adaptive signal processing applications much easier and faster, even for the most novice to the subject. ASPT isolates the user from the details of the algorithms' implementations and allows using the adaptive algorithms as reliable and well tested black box functions. ASPT contains adaptive algorithms for transversal, lattice, recursive, and non-linear adaptive filters, with implementations in the time and frequency domains, as well as specialized algorithms for applications such as active noise and vibration control, and beam forming. ASPT natively supports single and multi-channel systems for both real and complex signals and filter applications.

ASPT comes with simulation examples for applications of adaptive filters including echo cancellers, single channel and multichannel active noise and vibration control, beam forming, channel equalization, adaptive line enhancers, system identification, and linear prediction. The following is a list of the functions in the current ASPT release.

Transversal and Linear Combiner adaptive filters	
asptarlmsnewt	AR modeling implementation of LMS Newton method.
asptbfdaf	Block Frequency Domain Adaptive Filter.

asptblms	Block Least Mean Squares.
asptbnlms	Block Normalized Least Mean Squares.
asptdrlms	Data Reusing Least Mean Squares
asptdrnlms	Data Reusing Normalized Least Mean Squares
asptftrls	Fast Transversal Recursive Least Squares.
asptleakynlms	Leaky Normalized Least Mean Squares.
asptlclms	Linearly Constrained LMS.
asptlms	Least Mean Squares (LMS) and its variants.
asptmvsslms	Reduced complexity Variable Step Size LMS
asptnlms	Normalized LMS.
asptpbfdaf	Partitioned Block Frequency Domain.
asptrcpbfdaf	Reduced Complexity Partitioned Block Frequency Domain.
asptrdrlms	Recent Data Reusing Least Mean Squares
asptrdrnlms	Recent Data Reusing Normalized Least Mean Squares
asptrls	Recursive Least Squares.
aspttdftaf	Transform domain Fault Tolerant Adaptive Filter.
aspttdlms	Transform domain LMS.
asptvffrls	Variable Forgetting Factor Recursive Least Squares.
asptvsslms	Variable Step Size LMS.

Initialization of Transversal and Linear Combiner filters :	
init_arlmsnewt	AR modeling implementation of LMS Newton method.
init_bfdaf	Block Frequency Domain Adaptive Filter.
init_blms	Block Least Mean Squares.
init_bnlms	Block Normalized Least Mean Squares.
init_drlms	Data Reusing Least Mean Squares
init_drnlms	Data Reusing Normalized Least Mean Squares
init_ftrls	Fast Transversal Recursive Least Squares.
init_leakynlms	Leaky Normalized Least Mean Squares.
init_lclms	Linearly Constrained LMS.
init_lms	Least Mean Squares (LMS) and its variants.
init_mvsslms	Reduced complexity Variable Step Size LMS
init_nlms	Normalized LMS.
init_pbfdaf	Partitioned Block Frequency Domain.
init_rcpbfdaf	Reduced Complexity Partitioned Block Frequency Domain.
init_rdrlms	Recent Data Reusing Least Mean Squares
init_rdrnlms	Recent Data Reusing Normalized Least Mean Squares
init_rls	Recursive Least Squares.
init_tdftaf	Transform domain Fault Tolerant Adaptive Filter.
init_tdlms	Transform domain LMS.
init_vffrls	Variable Forgetting Factor Recursive Least Squares.
init_vsslms	Variable Step Size LMS.

Lattice adaptive filters :	
asptftrls	Fast Transversal Recursive Least Squares.
asptlbpef	Adaptive LMS Lattice Backward Prediction Error Filter.
asptlfpef	Adaptive LMS Lattice Forward Prediction Error Filter.
asptlmslattice	LMS Lattice Joint Process Estimator.
asptrlslbpef	Adaptive RLS Lattice Backward Prediction Error Filter.
asptrlslfpef	Adaptive RLS Lattice Forward Prediction Error Filter.
asptrlslattice	RLS Lattice joint process estimator using a posteriori estimation errors.
asptrlslattice2	RLS Lattice joint process estimator using a priori estimation errors with error feedback.

Initialization of Lattice filters :	
init_ftrls	Fast Transversal Recursive Least Squares.
init_lbpef	Adaptive LMS Lattice Backward Prediction Error Filter.

init_lfpef	Adaptive LMS Lattice Forward Prediction Error Filter.
init_lmslattice	LMS Lattice Joint Process Estimator.
init_rlslbpef	Adaptive RLS Lattice Backward Prediction Error Filter.
init_rlslfpef	Adaptive RLS Lattice Forward Prediction Error Filter.
init_rlslattice	RLS Lattice joint process estimator using a posteriori estimation errors.
init_rlslattice2	RLS Lattice joint process estimator using a priori estimation errors with error feedback.

Recursive adaptive filters :	
asptcsoiir2	Cascaded Second Order IIR adaptive filter.
aspteqerr	Equation Error IIR adaptive filter.
asptouterr	Output Error IIR.
asptsharf	Simple Hyperstable Adaptive Recursive Filter.
asptsoiir1	Second Order IIR adaptive algorithm type 1.
asptsoiir2	Second Order IIR adaptive algorithm type 2.

Initialization of Recursive filters :	
init_csoiir2	Cascaded Second Order IIR adaptive filter.
init_eqerr	Equation Error IIR adaptive filter.
init_outerr	Output Error IIR.
init_sharf	Simple Hyperstable Adaptive Recursive Filter.
init_soiir1	Second Order IIR adaptive algorithm type 1.
init_soiir2	Second Order IIR adaptive algorithm type 2.

Active Noise & Vibration Control filters :	
asptadjlms	Adjoint LMS.
asptfdadj	Frequency Domain Adjoint LMS.
asptfdfxlms	Frequency Domain Filtered-x LMS.
asptfxlms	Filtered-x LMS.
asptmcadjlms	Multichannel Adjoint LMS.
asptmcfdadjlms	Multichannel Frequency Domain Adjoint LMS.
asptmcfdfxlms	Multichannel Frequency Domain Filtered-x LMS.
asptmcfxlms	Multichannel Filtered-x LMS.

Initialization of Active Noise & Vibration Control filters :	
init_adjlms	Adjoint LMS.
init_fdadj	Frequency Domain Adjoint LMS.
init_fdfxlms	Frequency Domain Filtered-x LMS.
init_fxlms	Filtered-x LMS.
init_mcadjlms	Multichannel Adjoint LMS.
init_mcfdadjlms	Multichannel Frequency Domain Adjoint LMS.
init_mcfdfxlms	Multichannel Frequency Domain Filtered-x LMS.
init_mcfxlms	Multichannel Filtered-x LMS.

Nonlinear adaptive filters	
asptsovlms	Second Order Volterra LMS and several of its variants.
asptsovnlms	Second Order Volterra Normalized LMS algorithm.
asptsovrls	Second Order Volterra RLS algorithm.
asptsovtdlms	Second Order Volterra Transform domain LMS algorithm.
asptsovvsslms	Second Order Volterra Variable Step Size LMS algorithm.

Initialization of nonlinear filters	
init_sovlms	Second Order Volterra LMS and several of its variants.
init_sovnlms	Second Order Volterra Normalized LMS algorithm.
init_sovrls	Second Order Volterra RLS algorithm.
init_sovtdlms	Second Order Volterra Transform domain LMS algorithm.

init_sovvsslms	Second Order Volterra Variable Step Size LMS algorithm.

Visualization and helper functions	
init_ipwin	Initializes iteration progress GUI window.
ipwin	Builds the iteration progress GUI window.
getStop	Returns the condition of the stop button in the IPWIN.
guifb	Handles the GUI feedback functions of the IPWIN.
mcmixr	Calculates the response of N speakers at M microphones.
osfilter	Fast FIR filter using overlap-save.
plot_ale	Generates plots for the Adaptive Line Enhancer problems.
plot_anvc	Generates plots for Active Noise and Vibration Control problems.
plot_beam	Generates plots for beam forming problems.
plot_echo	Generates plots for echo cancellers applications.
plot_model	Generates plots for modeling problems.
plot_predict	Generates plots for linear prediction problems.
sovfilt	Second Order Volterra filter.
update_ipwin	Updates the iteration progress GUI window.

Examples and applications.	
ale_csoiir2	Adaptive Line Enhancer using CSOIIR2.
ale_soiir1	Adaptive Line Enhancer using SOIIR1.
ale_soiir2	Adaptive Line Enhancer using SOIIR2.
anvc_adjlms	Active noise and vibration control using ADJLMS.
anvc_fdadjlms	Active noise and vibration control using FDADJLMS.
anvc_fdfxlms	Active noise and vibration control using FDFXLMS.
anvc_fxlms	Active noise and vibration control using FXLMS.
anvc_mcadjlms	Active noise and vibration control using MCADJLMS.
anvc_mcfdadjlms	Active noise and vibration control using MCFDADJLMS.
anvc_mcfdfxlms	Active noise and vibration control using MCFDFXLMS.
anvc_mcfxlms	Active noise and vibration control using MCFXLMS.
beambb_lclms	Beam former at base band frequency using LCLMS.
beamrf_lms	Beam former at RF frequency using LMS.
echo_bfdaf	Echo canceller using BFDAF.
echo_leakynlms	Echo canceller using LEAKYNLMS.
echo_nlms	Echo canceller using NLMS.
echo_pbfdaf	Echo canceller using PBFDAF.
echo_rcpbfdaf	Echo canceller using RCPBFDAF.
equalizer_nlms	Inverse modeling using NLMS.
equalizer_rls	Inverse modeling using RLS.
model_arlmsnewt	Modeling using LMSNEWTON.
model_eqerr	IIR modeling using EQER.
model_lmslattice	Modeling using LMSLATTICE.
model_mvsslms	FIR modeling using MVSSLMS.
model_outerr	IIR modeling using OUTERR.
model_rls	FIR modeling using RLS.
model_rlslattice	Modeling using RLSLATTICE.
model_sharf	IIR modeling using SHARF.
model_tdlms	FIR modeling using TDLMS.
model_vsslms	FIR modeling using VSSLMS.
powerline	interference cancellation using LMS.
predict_lbpef	lattice prediction using LBPEF.
predict_lfpef	lattice prediction using LFPEF.
predict_rlslbpef	lattice prediction using RLSLBPEF.
predict_rlslfpef	lattice prediction using RLSLFPEF.

System Requirements

ASPT will run on any system running Matlab version 5.3 or better. Older versions of Matlab are not supported.

Installing ASPT

To install the Adaptive Signal Processing Toolbox do the following.

1. Unpack the file you received where you want to install ASPT. Below I assume that you are using Windows OS and you unpacked the package at "D:\", if you are using other OS, Unix or Linux for instance, replace "D:\" with your home directory or where Matlab is installed. Unpacking will create a directory "D:\dspalgorithms\asptxyz", where xyz is the ASPT version number. After unpacking you should find the following on your disk drive:

- Implementations of all adaptive algorithms, initialization functions, and helper functions.
- Matlab help files in the help directory.
- ASPT for Matlab documentation in PDF format in the docs directory.
- Short test files for each algorithm in the test directory.
- A larger application for each algorithm in the apps directory.
- Data files used in the applications in the data directory.
- Wave audio files used as input for the applications in the wavin directory.
- Wave audio files produced by running the applications in the wavout directory.
- ASPT License Agreement.

2. Carefully read the "asptlicense.pdf" or "asptlicense.txt" file located in the doc directory of ASPT distribution. If you agree to the license agreement, you may continue further with the installation and configuration, otherwise contact DSP algorithms sales (sales@dspalgorithms.com) for a full refund, subject to the conditions mentioned in the License Agreement. Note that it is NOT possible to obtain refund AFTER obtaining your full licensed package. This means if you have provided DSP Algorithms with your ASPT ID CODE and received a fully licensed software, no refund will be applicable.

3. Start your Matlab software and change the current working directory inside Matlab to "D:\dspalgorithms\asptxyz", and then run the ASPT installation program from there. You can do this by typing the following at the Matlab prompt and hitting the [Enter] key after each instruction

```
>> cd D:\dspalgorithms\asptxyz
>> asptinstall
```

   To successfully install ASPT you need write permission to some Matlab files. The installation program will return an error if you have no write permission to those files. In this case you will need to run asptinstall each time you start Matlab, since the path information will be forgotten when you quit Matlab.

4. Now you should be ready to start using ASPT. To test your installation and get started immediately, run any of the short test files located in the test directory or one of the applications located in the apps directory.

```
>> testfxlms
```

This should execute without any errors.

5. If you have already provided DSP Algorithms with your ASPT ID CODE and received your licensed ASPT package, test your license by running the "testlicense" program from within Matlab,

   ```
   >> testlicense
   ```

   This should show a dialog box with the text "License installed correctly". Any text otherwise means that you have the wrong license and you should contact us immediately to correct any error.

Uninstalling ASPT

To uninstall ASPT, type the following at the Matlab prompt

```
>> asptuninstall
```

and press the [Enter] key. This will remove the ASPT directories from Matlab path but will not delete any files from your disk drive. To successfully change the Matlab path you need write permission to the file "pathdef.m". You can then manually delete the ASPT directory tree from your disk drive.

Getting Started With ASPT

ASPT comes "out-of-the-box" with many components that help making your ASPT learning curve as short as possible. You can get started by doing one or all of the following:

1. Type "help aspt" at Matlab prompt to retrieve a list of ASPT functions. If you need more help on, say, the fxlms algorithm, then simply type "help asptfxlms".

2. Each algorithm has two or more examples. A short example located in the "test" directory and a longer example located in the "apps" directory. ASPT users have indicated that those examples have been their best starting point. You can start learning ASPT features by browsing through those examples and editing the examples to experiment with the parameters of each ASPT algorithm. Make sure, however, to keep a copy of the original file for future use.

3. The ASPT documentation is available in PDF format as well as in HTML format. Please refer to this documents for more in depth information on each algorithm.

4. If you can not find the information you need, we will be glad to help you. Send your question directly to aspt@dspalgorithms.com and we will try to answer your questions to our best knowledge as soon as we can.

ASPT Updates and Other Related products

Your ASPT license includes free-of-charge updates for the period of SIX MONTHS from the delivery date. From time to time we will send you ASPT update information to keep you informed with new ASPT releases, new features, and new products related to ASPT. If you prefer NOT to receive such update information, send an email to aspt@dspalgorithms.com at any time with "NO UPDATES" in the subject.

Customer Support

For technical support, license problems, technical questions, suggestions, feedback and remarks please write to aspt@dspalgorithms.com. For licensing, payments, prices, and marketing information, and marketing cooperation, please contact sales@dspalgorithms.com.

Note: Matlab is a registered trademark of the Mathworks Corporation.