

Java-DSP: Recent Developments and Extensions

Karthikeyan Natesan Ramamurthy,
Jayaraman J. Thiagarajan and Andreas Spanias

J-DSP Cyprus Workshop

June 24, 2012



Agenda



- i-JDSP : Interactive iPhone/iPad based tool based on Java-DSP.
- Java-DSP Quiz Version : Interface to enhance student learning.
- Interfacing Java-DSP to MATLAB and LabVIEW.
- User-defined Java Code.
- Java-DSP and Sensor Motes.
- Java-DSP Earth Systems Edition.

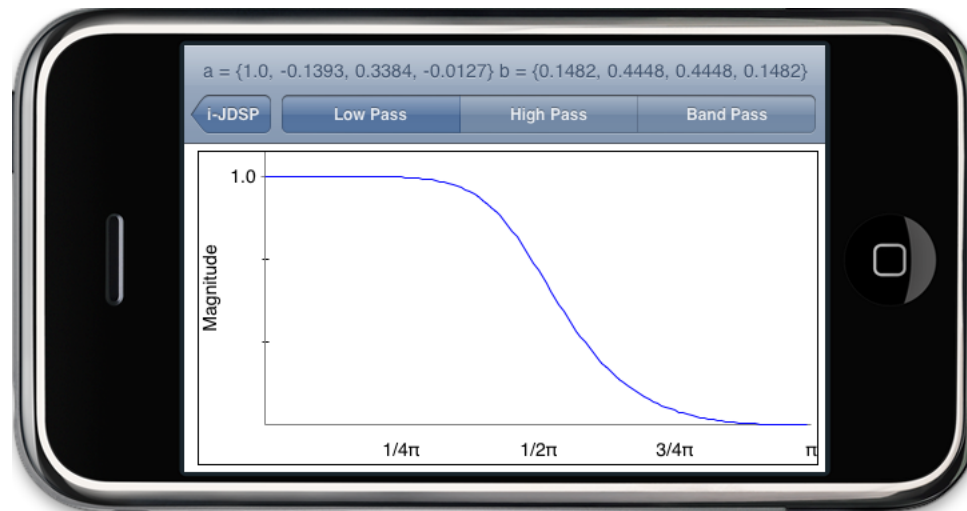


i-JDSP : Interactive Signal Processing Tool for iPhone/iPad



i-JDSP

- An interactive iPhone/iPad based signal processing tool based on Java-DSP.
- Implemented in Objective-C and C as a native Cocoa Touch application that can be run on any iOS device.
- Offers basic signal processing simulation functions on the new compact and convenient iPhone/iPad graphical user interface (GUI).
- Provides a very compelling multi-touch programming experience.

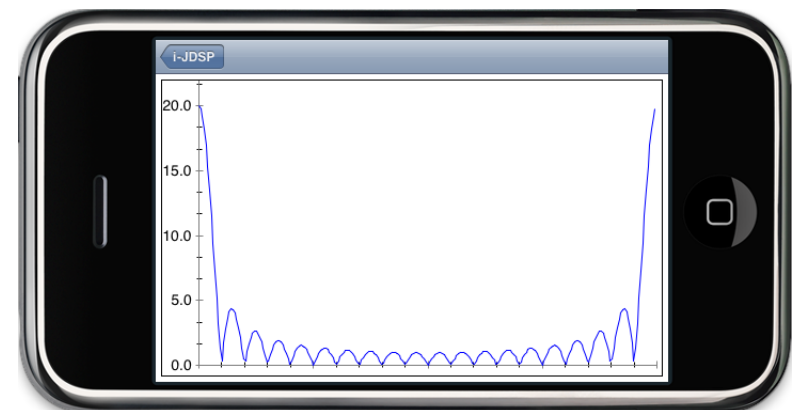
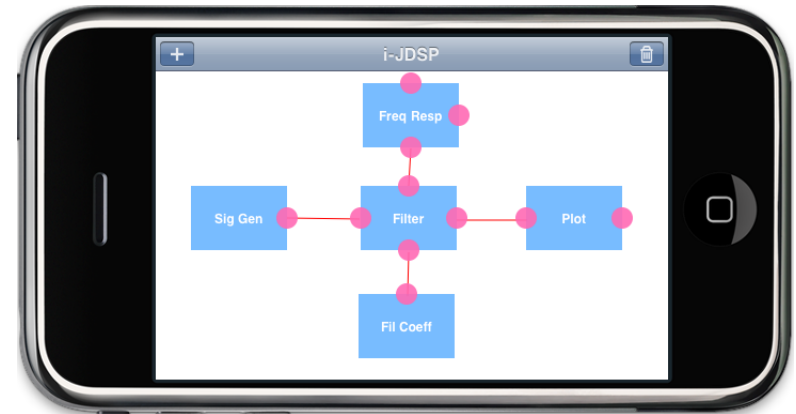


Using the Tool

- All simulations can be visually established by forming interactive block diagrams through multi-touch and drag-and-drop.

- List of existing function blocks:

- a. signal generator
- b. Filter
- c. Filter coefficient
- d. Frequency response
- e. FFT
- f. Plot
- g. Frequency response demo



Key Features



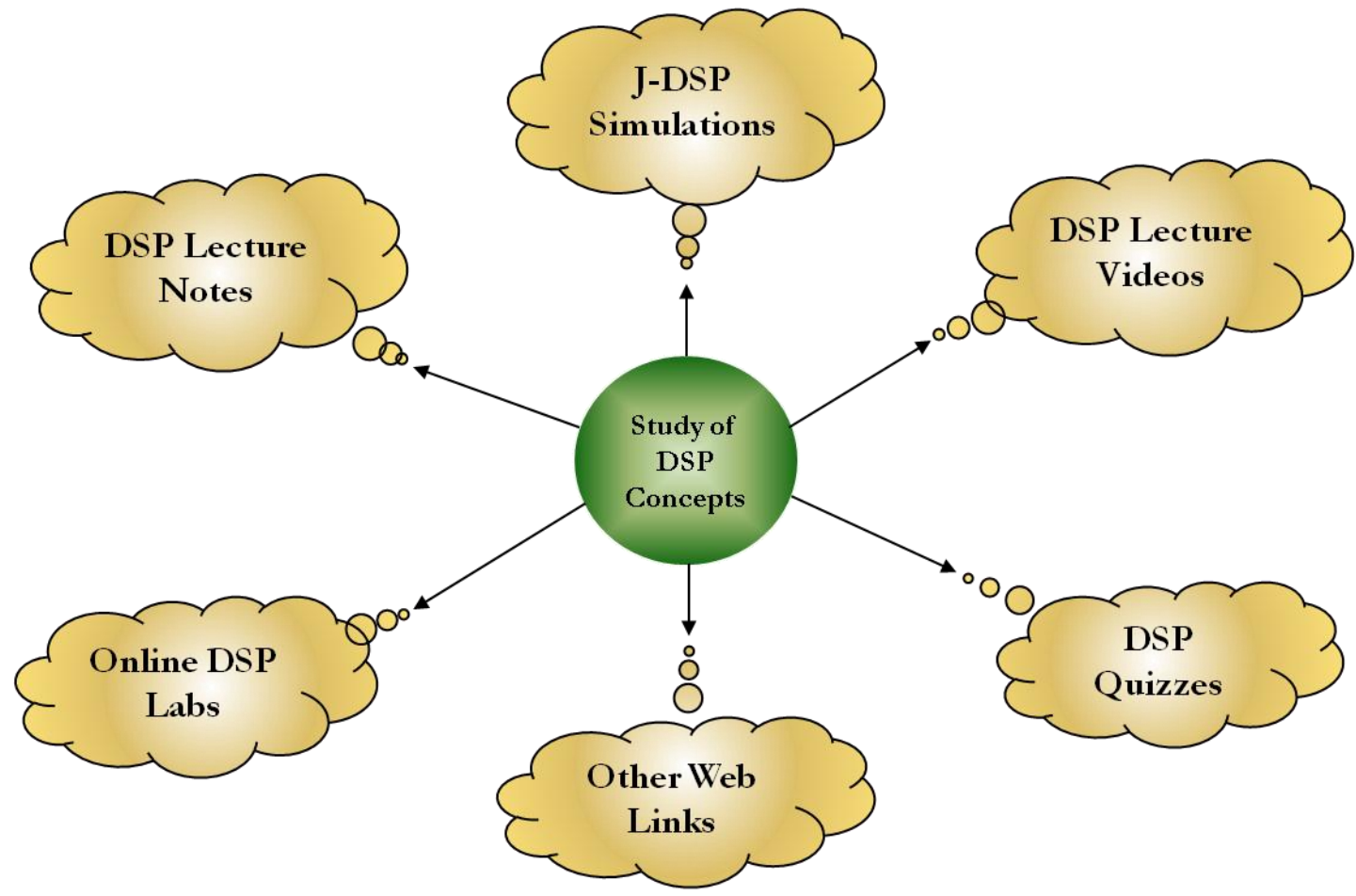
- Graphical programming experience.
- Easy to use (multi-touch and drag-and-drop).
- Portability and accessibility through online app store.
- Small footprint/efficiency.
- Scalability.
- **Future Work**
 - i. Adding more function blocks.
 - ii. Extension to other mobile platforms.
 - iii. Release to online app store.
 - iv. Bug fixing and Testing.



Java-DSP Quiz Version for Enhanced Student Learning

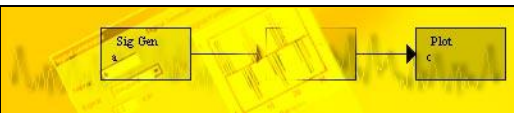
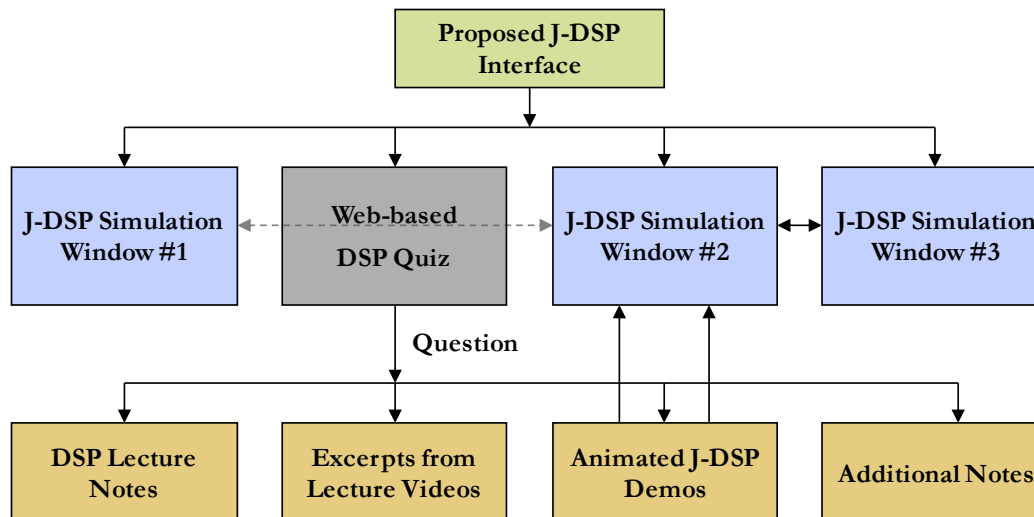


Motivation



Proposed Interface

- Existing Java-DSP framework – Rich palette of signal processing functions.
- Integration of several discrete learning components to Java-DSP.
- Access to three simulations in a single Java-DSP window.
- Allows implementation of large simulations that cannot be accommodated in a single window.



Web-Based Quiz



- Connecting link between DSP questions, J-DSP simulations and other learning components.

Chapter 4 Question 7


Question

The worse ripple characteristic when truncating the FS is produced by:

Answer (Select Only One Answer)

a) L-point rectangular window b) L-point triangular window

Correct Answer

Answer: **a, L-point rectangular window**  Comments: The rectangular window is associated with the sharpest transition in the time domain which causes the ripples in the frequency domain,


Evaluate **Return to Index** Switch to JDSP Editor Proceed

JDSP Demo Brief Explanation DSP Lecture Video



Links to Relevant Material


- Links to all relevant materials and demos are activated after the student attempts a question.
- Includes animated Java-DSP demos, lecture notes, lecture videos.



Ira A. Fulton
Schools of Engineering
ARIZONA STATE UNIVERSITY

Brought to you by
Global Outreach & Extended Education

Slide 1 of 3
powered by mediasite



Paused 00:07:03:00

DSP Quiz - Chapter 4 - Q12
Andreas Spanias

11/13/2009 10:20 AM PST Length: 00:03:00 More...

Min-Max and Parks-McClellan Optimum FIR Design

The Parks-McClellan design is based on Min-Max

Equiripple and linear phase design is possible

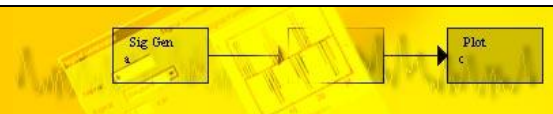
This class of methods involve minimizing the maximum error between the designed FIR filter frequency response and a prototype

$$\min_{\{h(i), i=0,1,\dots,L\}} \left\{ \max |E(e^{j\theta})| \right\}$$

where

$$E(e^{j\theta}) = W(e^{j\theta})(H_d(e^{j\theta}) - H(e^{j\theta}))$$

Copyright © Andreas Spanias Ch 4-51



Interfacing Java-DSP to MATLAB and LabVIEW



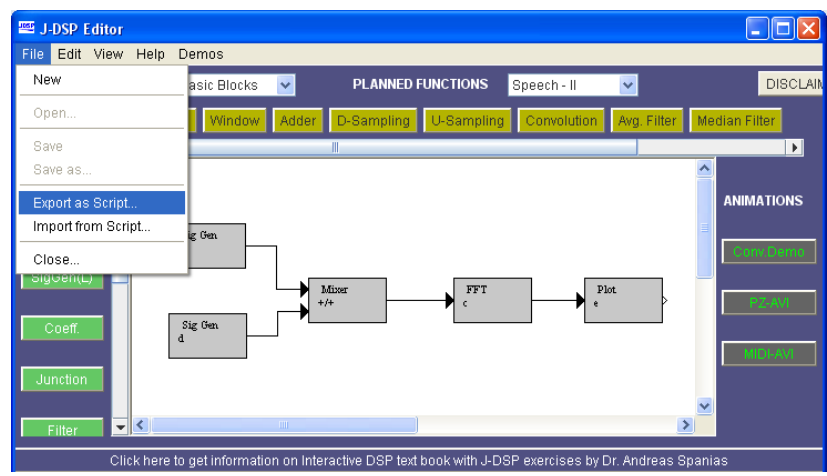
Overview



- Enables students and instructors to exchange data and perform simulations on both the platforms.
- Allows to translate the block diagram from Java-DSP into MATLAB code.
- Enables users to repeat, verify and expand simulations in the MATLAB environment.
- Conversely, MATLAB programs can mapped to flowchart like diagrams and run in Java-DSP.



MATLAB Interface



Script Export

Copy and paste this code: MATLAB(TM) Code

- Applet Code Only
- Applet in HTML Code
- MATLAB(TM) Code**

```

% in a manner consistent with the program
% working on software updates that will a

clear all; close all;

%Signal Generator (a,1)
t=[0:1:255];y=3.0 * sin(pi*t*0.2);
VAR1= [zeros(1,0) y];

%Signal Generator (d,4)
t=[0:1:255];y=2.0 * sin(pi*t*0.4);
VAR2= [zeros(1,0) y];

% Mixer (+/,2)
if (length(VAR1)>length(VAR2))
VAR2=[VAR2 zeros(1,(length(VAR1)-length(VAR2)))]];
else
VAR1=[VAR1 zeros(1,(length(VAR2)-length(VAR1)))]];
end
t=[0:1:max(length(VAR1),length(VAR2))-1];
VAR3= VAR1+VAR2;

% FFT (c,3)
VAR4= fft(VAR3,256);
% Plot (e,5)
figure(5);
eps = 1e-7;
    
```



Java-DSP LabVIEW Interface



J-DSP Editor RC_Filter.vi Block Diagram

File View Help Demos File Edit View Project Operate Tools Window Help

EXISTING FUNCTIONS Filter Blocks PLANNED FUNCTIONS Speech

PZ Placement PZ-Plot FIR Design IIR Design Kaiser Design Parks-McClellan

Sig. Gen. SigGen(L) Coeff. Junction Filter FFT Freq-Resp Plot Plot2

Block Diagram: Sig Gen a → Filter b → [Hz Plot] → [Scope] → [Scope] → [Plot] → magnitude response → phase response

PZ Placement Block: Add poles/zeros: Graphically. Imaginary vs Real plot. Buttons: Add Zero, Add Pole, Delete, Reset. Java Applet Window.

magnitude response plot: Magnitude vs Frequency [Hz].

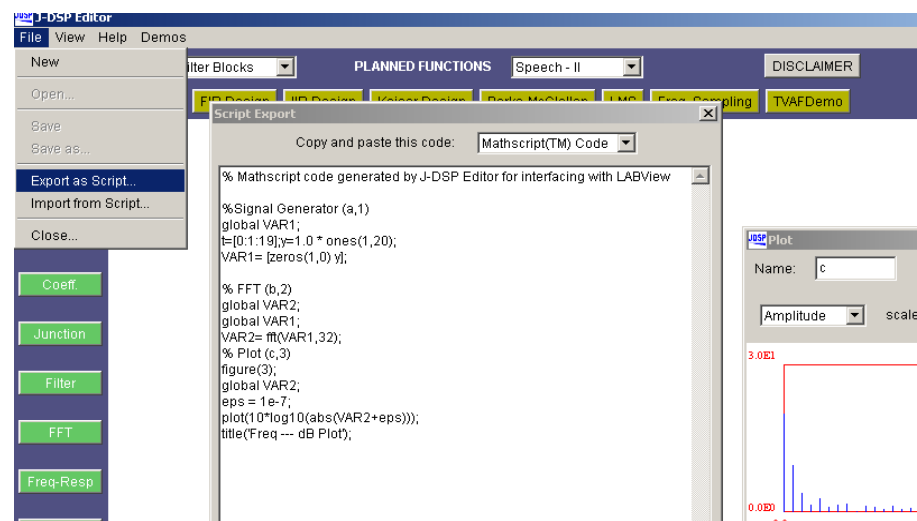
phase response plot: Phase vs Frequency [Hz].



Mathscript Code from Java-DSP

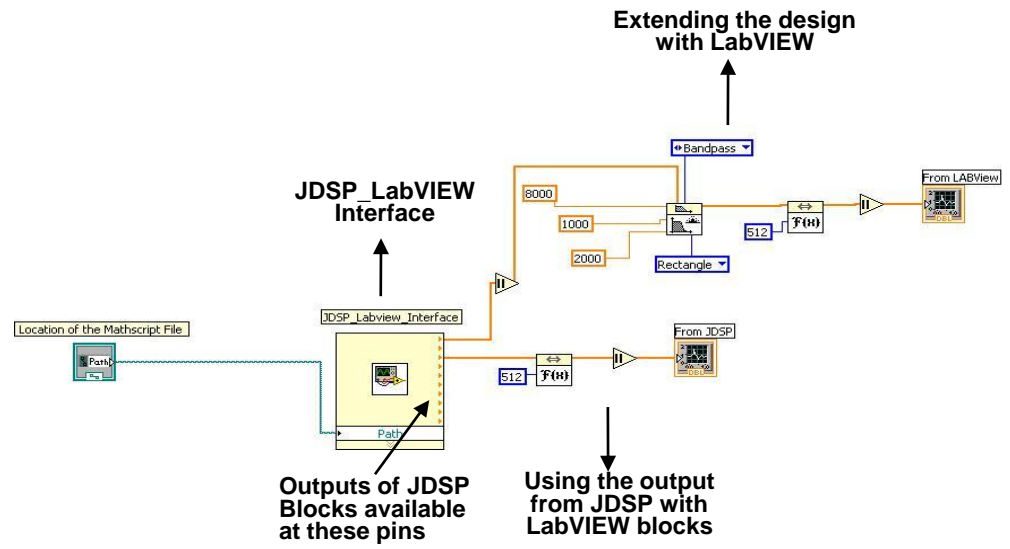
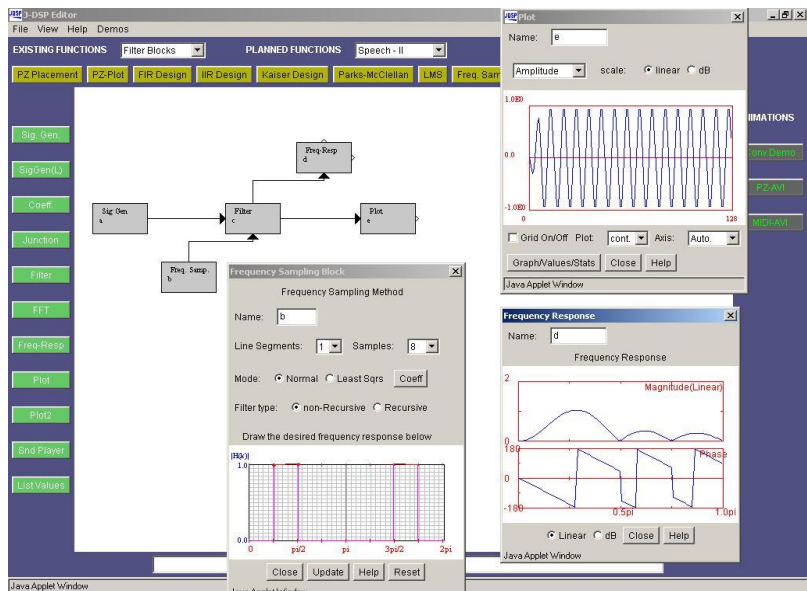


- Uses the script generation capabilities of J-DSP to generate Mathscript code.
- Most J-DSP simulations can be translated to Mathscript code.
- “Export Script” from file menu and select “Mathscript(TM) Code”.
- Copy paste code into m-file.
- Signed Applet – will eliminate copy-paste process.



Simulation Example

- Uses the script generation capabilities of J-DSP to generate Mathscript code.
- The design is extended in LabVIEW with native blocks

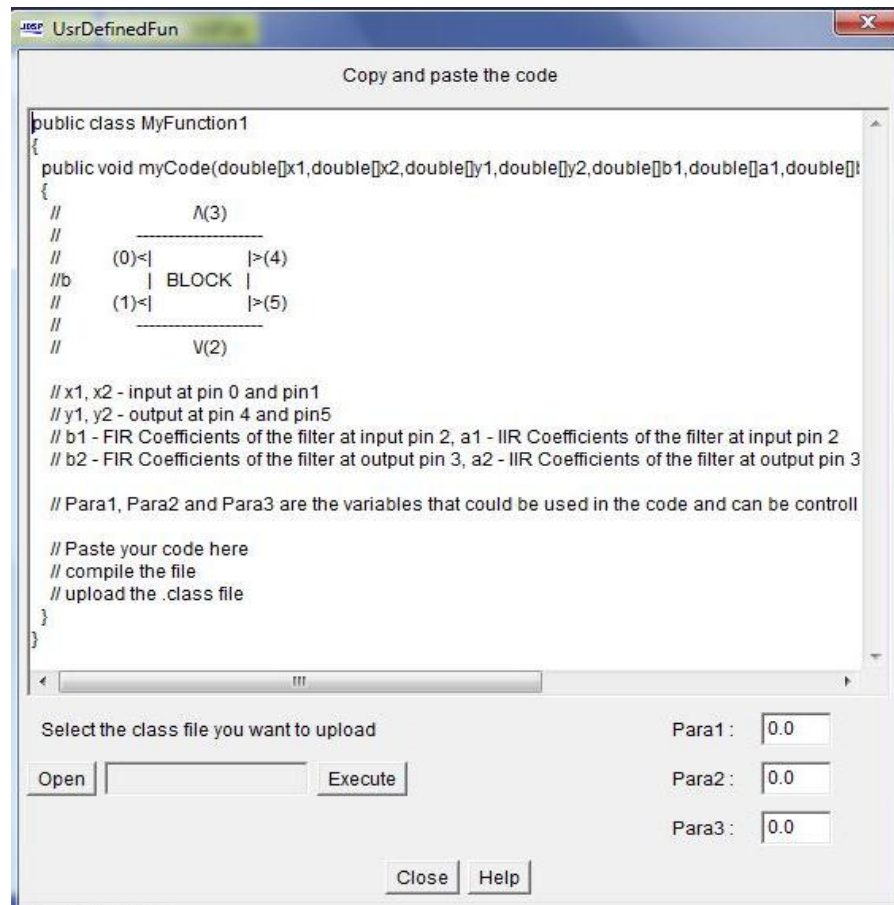


Including Custom Java Code



User-defined Block

- The user-defined block allows custom java code to be interfaced with other J-DSP blocks.



```
public class MyFunction1
{
    public void myCode(double[]x1,double[]x2,double[]y1,double[]y2,double[]b1,double[]a1,double[]
    {
        //
        //          N(3)
        //          -----
        //   (0)<|>(4)
        //b   | BLOCK |
        //   (1)<|>(5)
        //          -----
        //          V(2)

        // x1, x2 - input at pin 0 and pin1
        // y1, y2 - output at pin 4 and pin5
        // b1 - FIR Coefficients of the filter at input pin 2, a1 - IIR Coefficients of the filter at input pin 2
        // b2 - FIR Coefficients of the filter at output pin 3, a2 - IIR Coefficients of the filter at output pin 3

        // Para1, Para2 and Para3 are the variables that could be used in the code and can be controll

        // Paste your code here
        // compile the file
        // upload the .class file
    }
}
```

Select the class file you want to upload

Open Execute

Para1 :

Para2 :

Para3 :

Close Help



Example



Sample Java Code for the User Defined Block

```
public class MyFunction1
{
    public void myCode(double[] x1, double[] x2, double[] y1, double[] y2
    {
        /*
            ^ (3)
            (0) < |-----| > (4)
                | BLOCK | // BASIC BLOCK MODEL
            (1) < |-----| > (5)
                v (2)
        */

        // x1 - input at pin 0, exponential signal
        // check the top portion of the plot on the Left side

        // y1 - output at pin 4, absolute value of input signal
        // check the top portion of the plot on Right side

        for(int i=0; i<256; i++)
        {
            y1[i] = Math.abs(x1[i]);
        }

        // x2 - input at pin 1, triangular signal
        // check the bottom portion of the plot on the Left side

        // y2 - output at pin 5, input-Paral (Paral = 1)
        // check the bottom portion of the plot on Right side

        for(int i=0; i<256; i++)
        {
            y2[i] = x2[i] - paral;
        }
    }
}
```



Example

EXISTING FUNCTIONS New Blocks
 PLANNED FUNCTIONS Speech - II
 DISCLAIMER

WaveRead WaveWrite MP3 decoder UsrDefinedFun

Sig. Gen.

SigGen(L)

Coef.

Junction

Filter

FFT

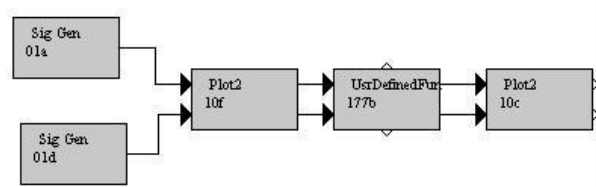
Freq-Resp

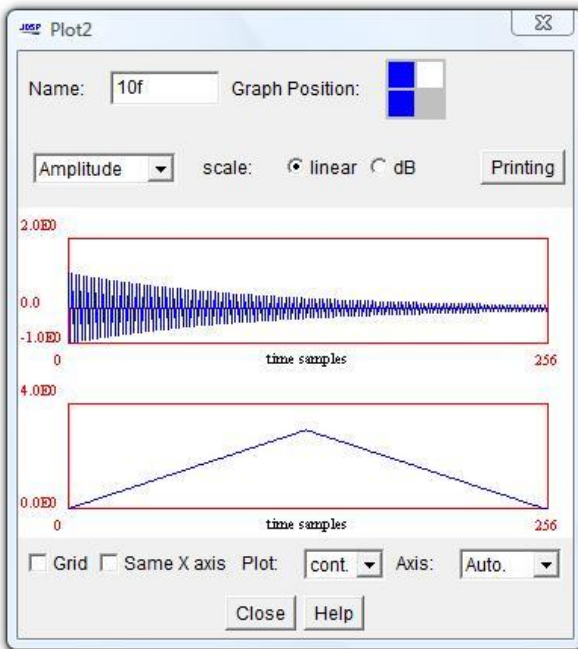
Plot

Plot2

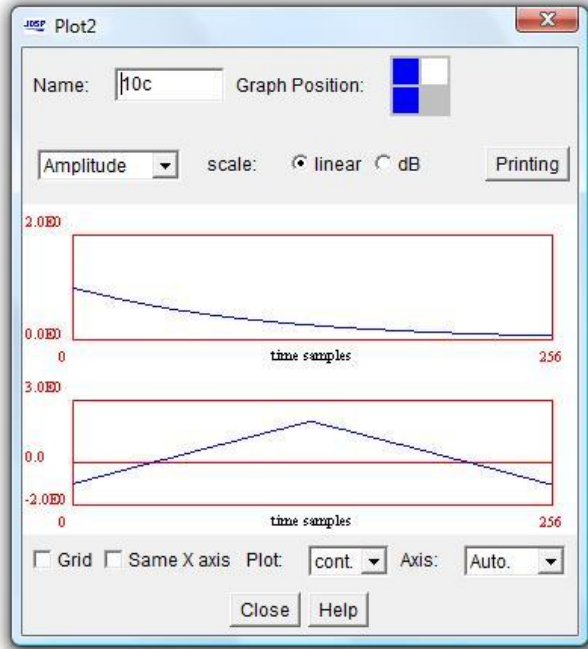
Snd Player

List Values





Name: 10f Graph Position:
 Amplitude: scale: linear dB Printing
 Grid Same X axis Plot: cont Axis: Auto.



Name: 10c Graph Position:
 Amplitude: scale: linear dB Printing
 Grid Same X axis Plot: cont Axis: Auto.



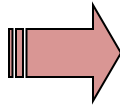
Java-DSP and Sensor Motes



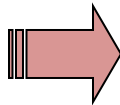
Overview



Java-DSP



Wireless
Sensor Motes

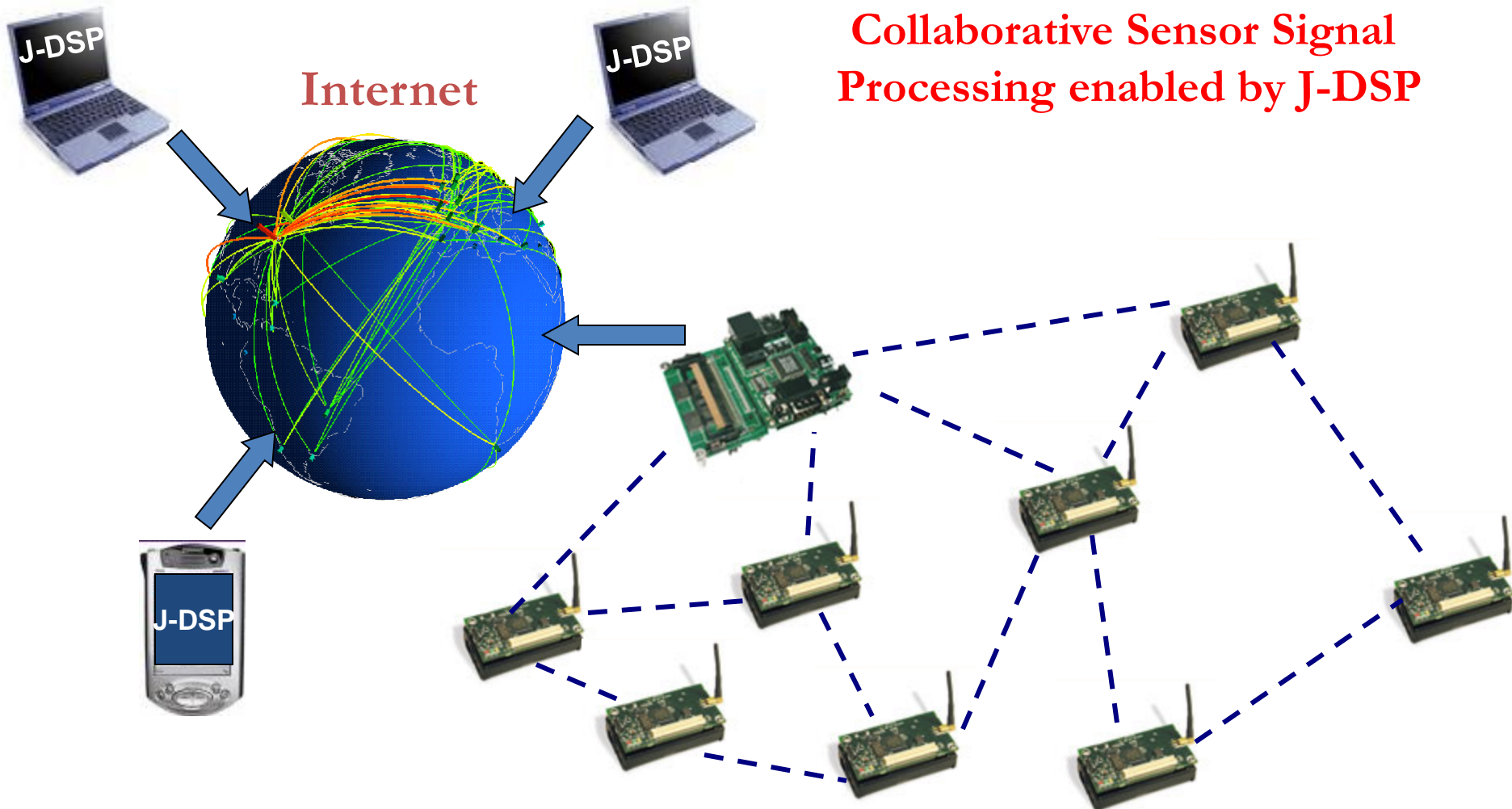


- A Web-based DSP Simulation Tool
- Universally accessible DSP functions
- Embeds Interactive Simulations in Web pages
- Seamlessly Integrates Animated Demos
- Integration enables real-time sensor signal analysis
- Java interface natural for remote sensing
- User-friendly GUI for computation/graphics using the J-DSP-Mote interface
- Hardware: *Mica2* from *Crossbow*



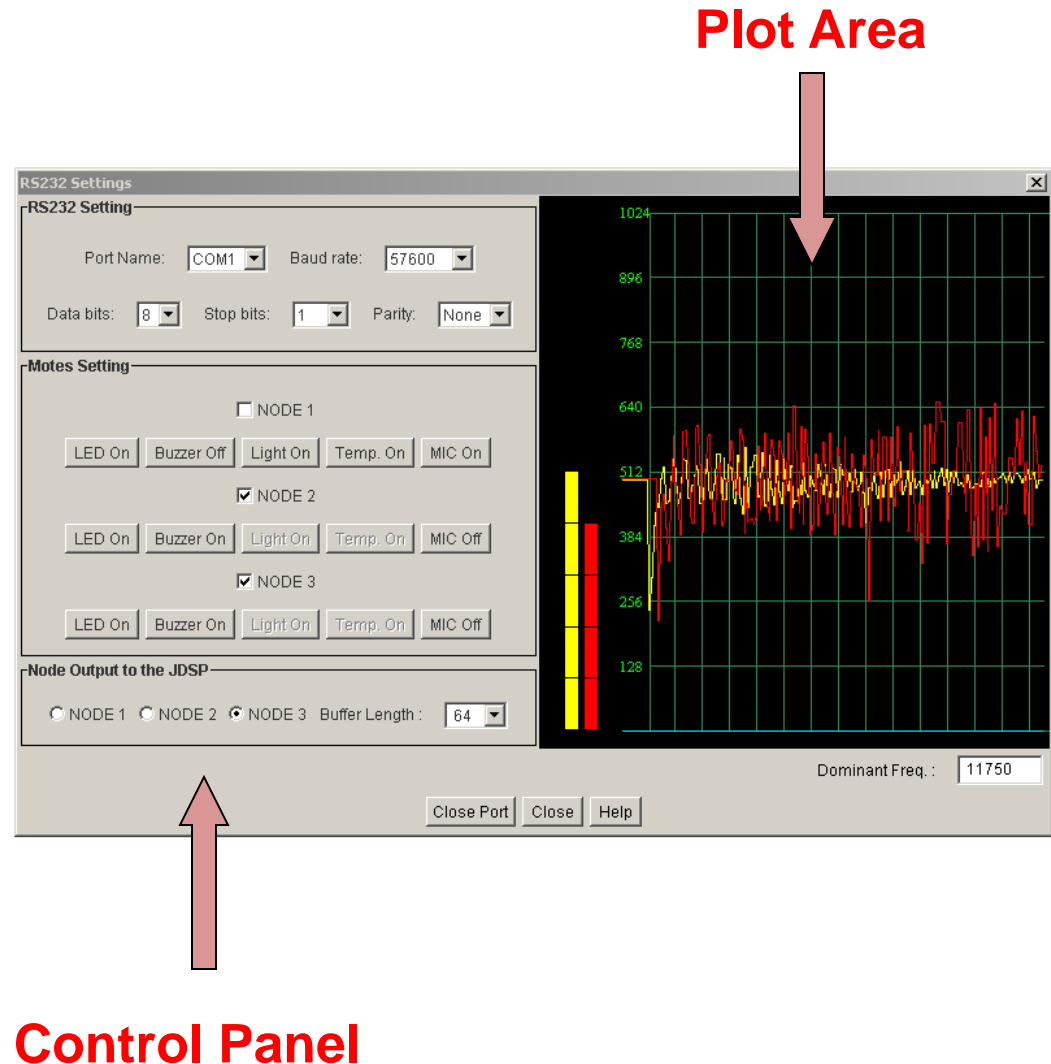
Java-DSP and the Motes

**Collaborative Sensor Signal
Processing enabled by J-DSP**



MOTE Block

- GUI for the motes
- Control panel is used to control the individual motes and the RS232 settings
- MOTE block in J-DSP allows users to control individual motes
- Real-time graph plots data as it comes

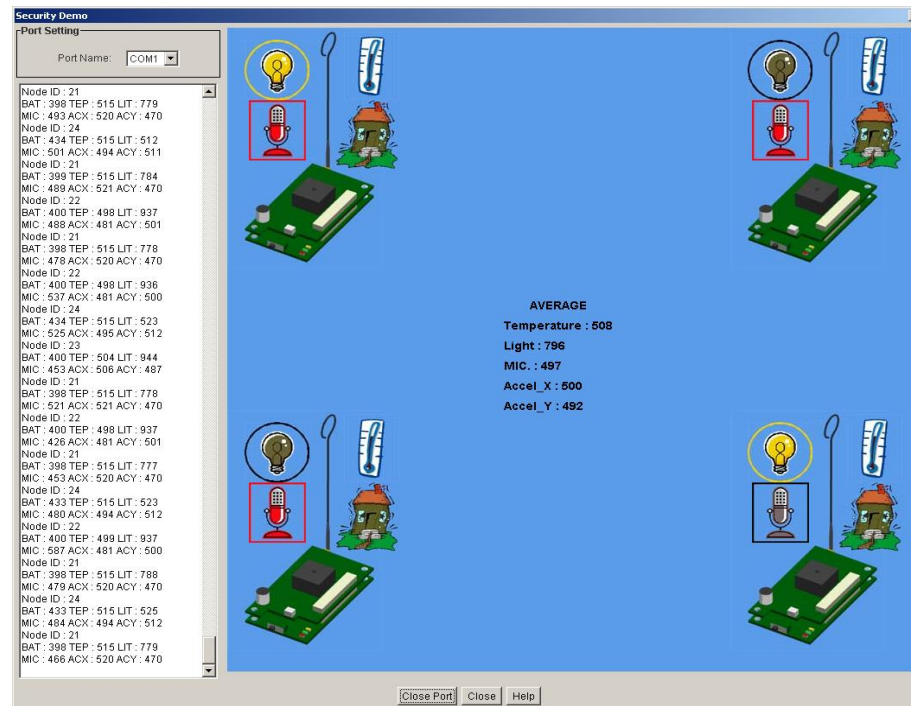


Remote Sensing Example

• Preliminary example shows possibilities for sensing and security applications.

• Display panel shows which sensors are active

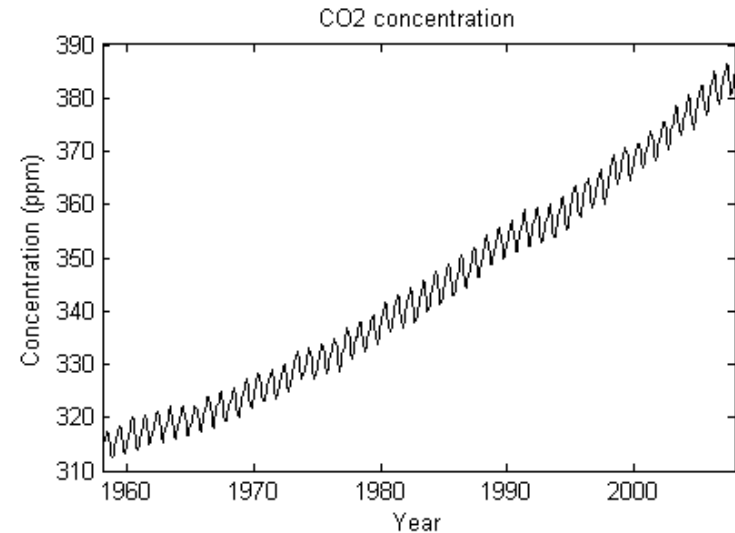
- Active Sensors:
- Light
- Sound
- Temperature
- Accelerometer



Java-DSP Earth Systems Edition

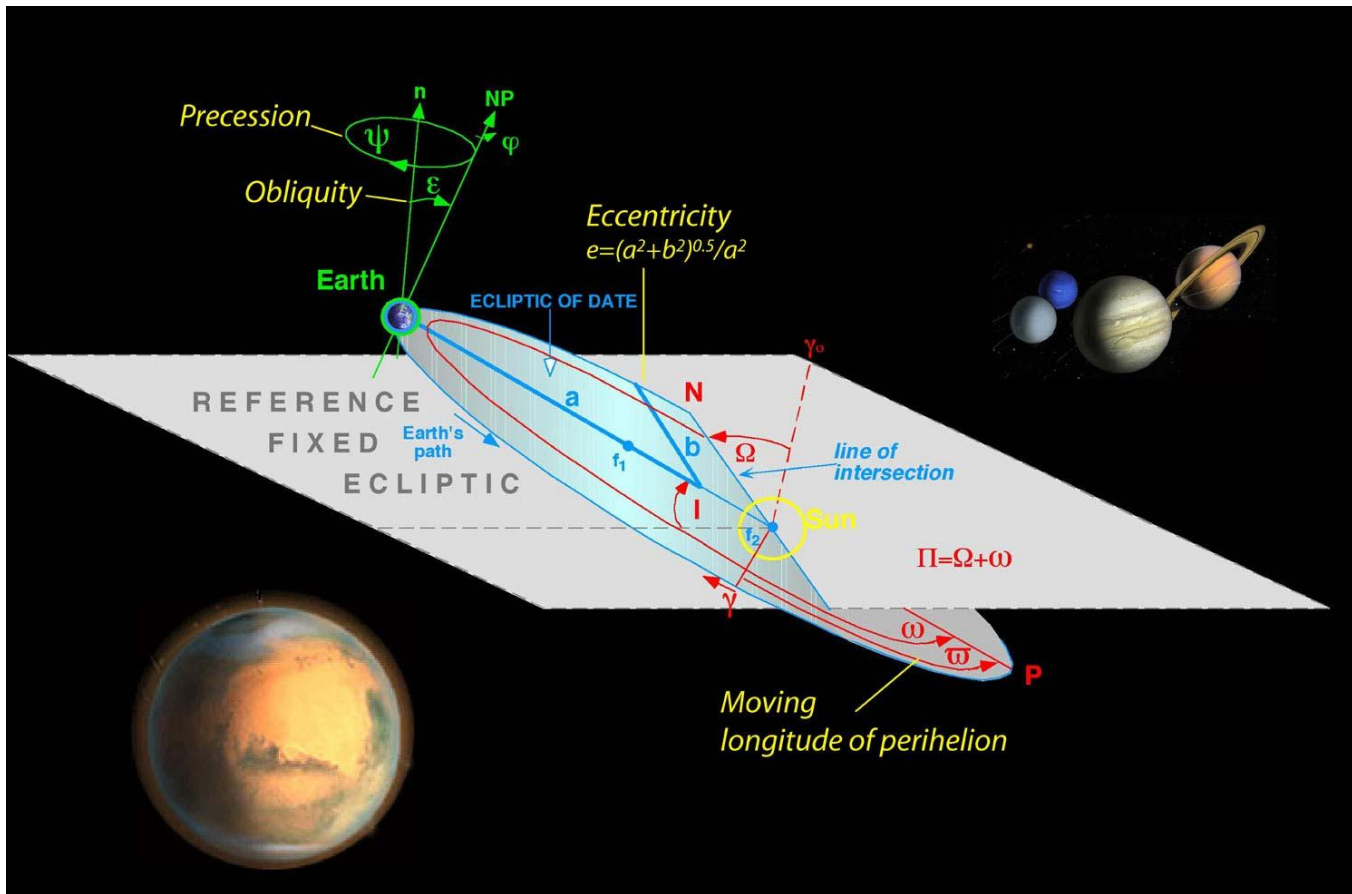


- **“Real-Time” monitoring**
 - Global temperature, Concentration of greenhouse gases, River flow, atmospheric pressure, earth orientation.
- **“Deep-Time” proxy data**
 - Proxy data that are representative of past Earth system behaviour.
 - Ice sheet isotopes (air temperatures), tree ring thicknesses (hydrology), magnetic intensity of ancient sediment (geomagnetic field).



Late Miocene (9.3 to 8.4 Ma) sapropel cycles from the Gibliscemi A section, south-central Sicily. Photo courtesy of F. Hilgen, University of Utrecht.

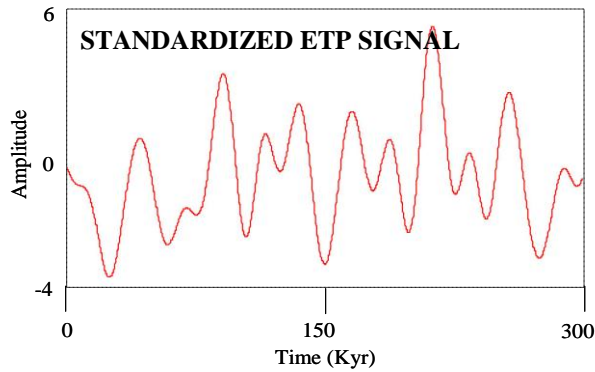
Expected Astronomical Frequencies



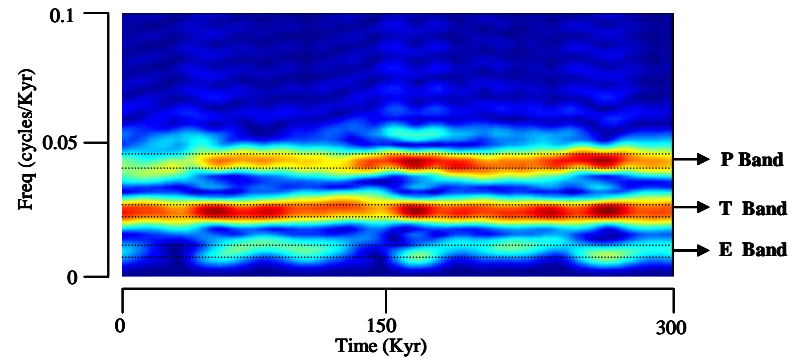
Earth's orbital parameters

E - periods 400,000 and 100,000 years, T - period 41,000 years,
P - periods of 23,000 and 19,000 years.

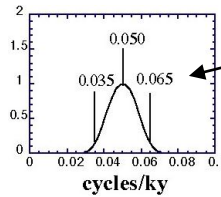
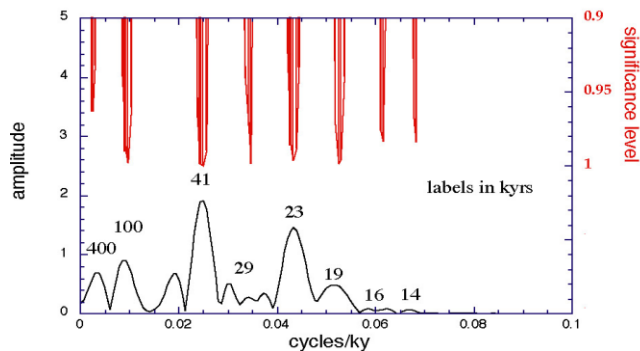
ETP Model Analysis



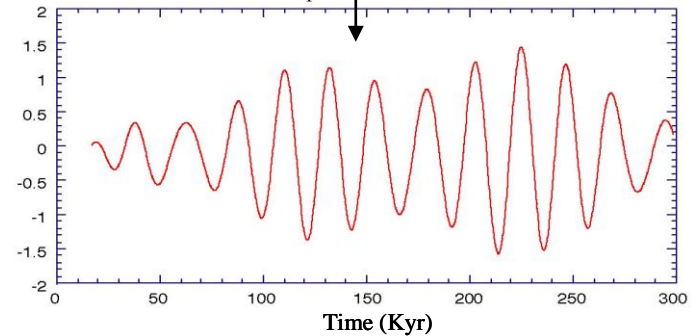
SPECTROGRAM

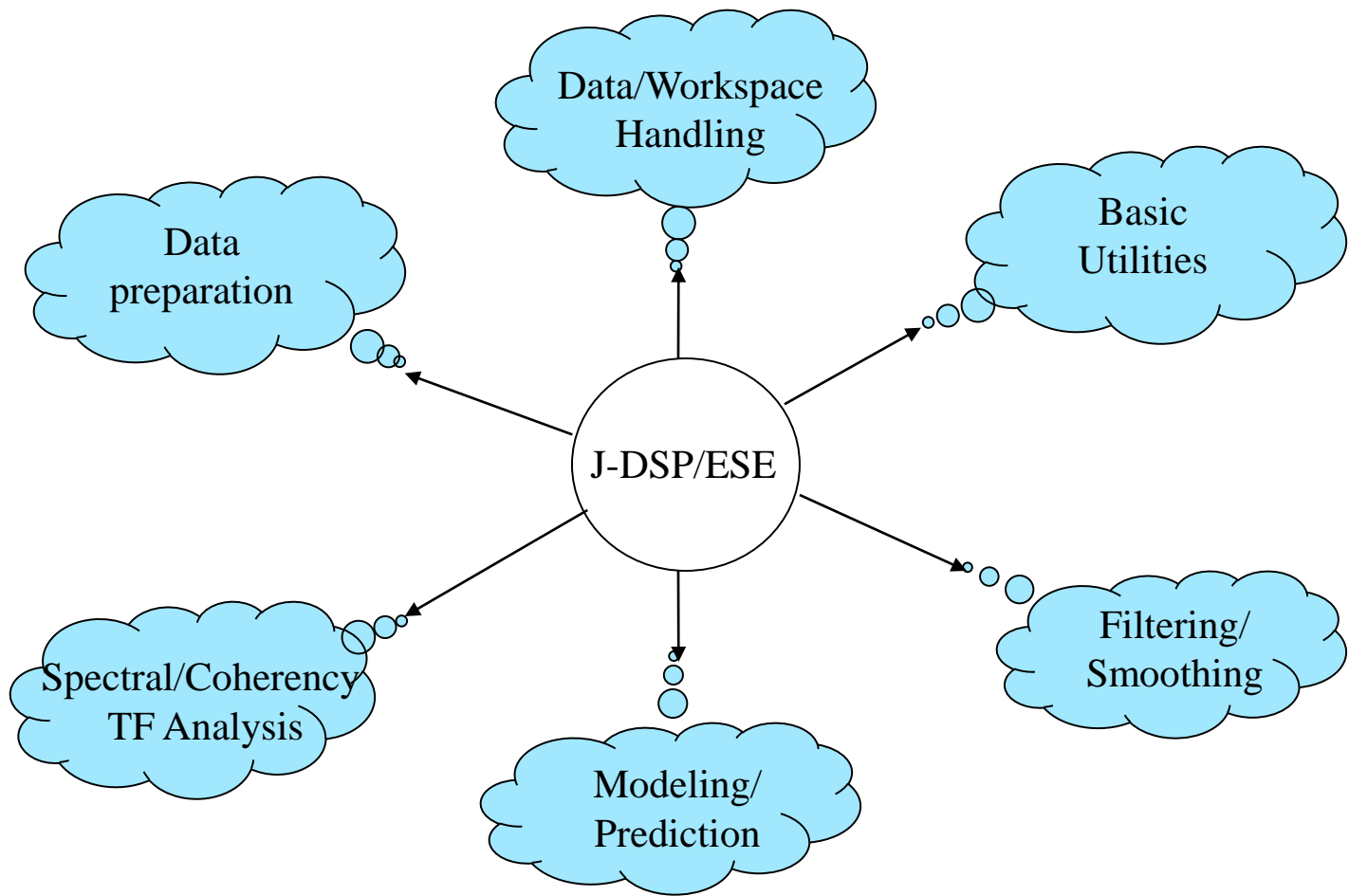


MULTIPLE PROLATE TAPER SPECTRUM



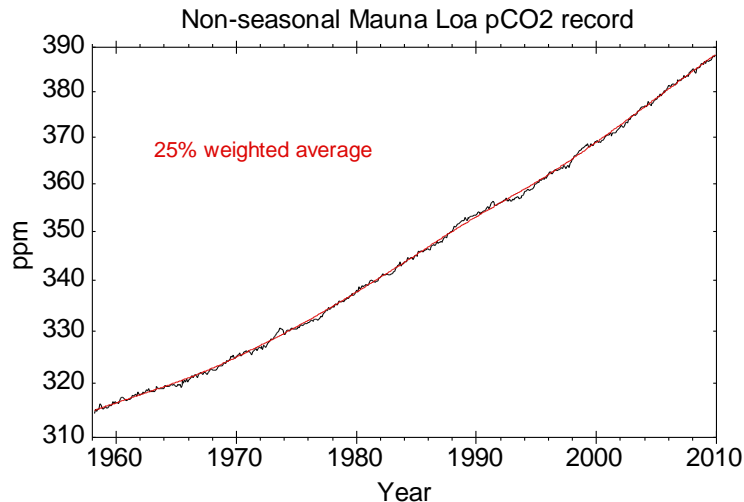
FILTERED SIGNAL P-BAND





Example – Global Warming in 20th Century

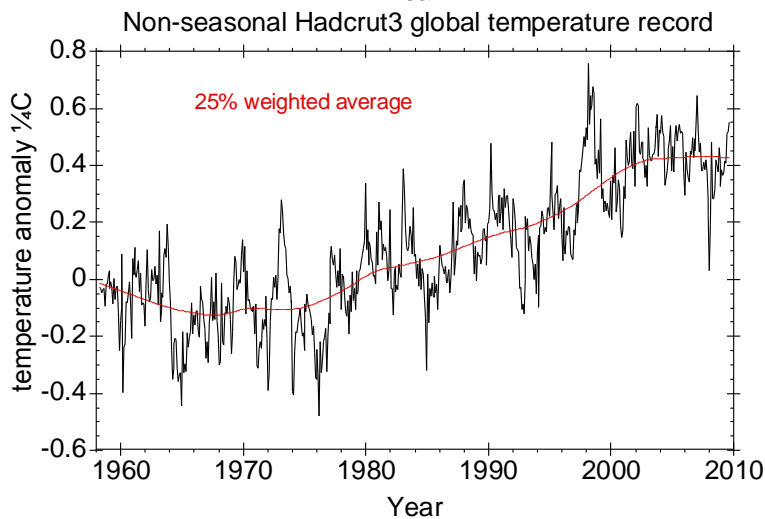
How is $p\text{CO}_2$ correlated with global climate change?



Mauna Loa atmospheric $p\text{CO}_2$ (ppm) mid-monthly values, with the seasonal cycle removed by subtracting “a 4-harmonic fit with a linear gain factor.”

<http://scrippsco2.ucsd.edu/>

Monthly global temperature record averaged over $5^\circ \times 5^\circ$ areal grids, from more than 3000 stations temperature time series, preprocessed to remove the seasonal cycle and biases from stations at different elevations and different averaging formulae.



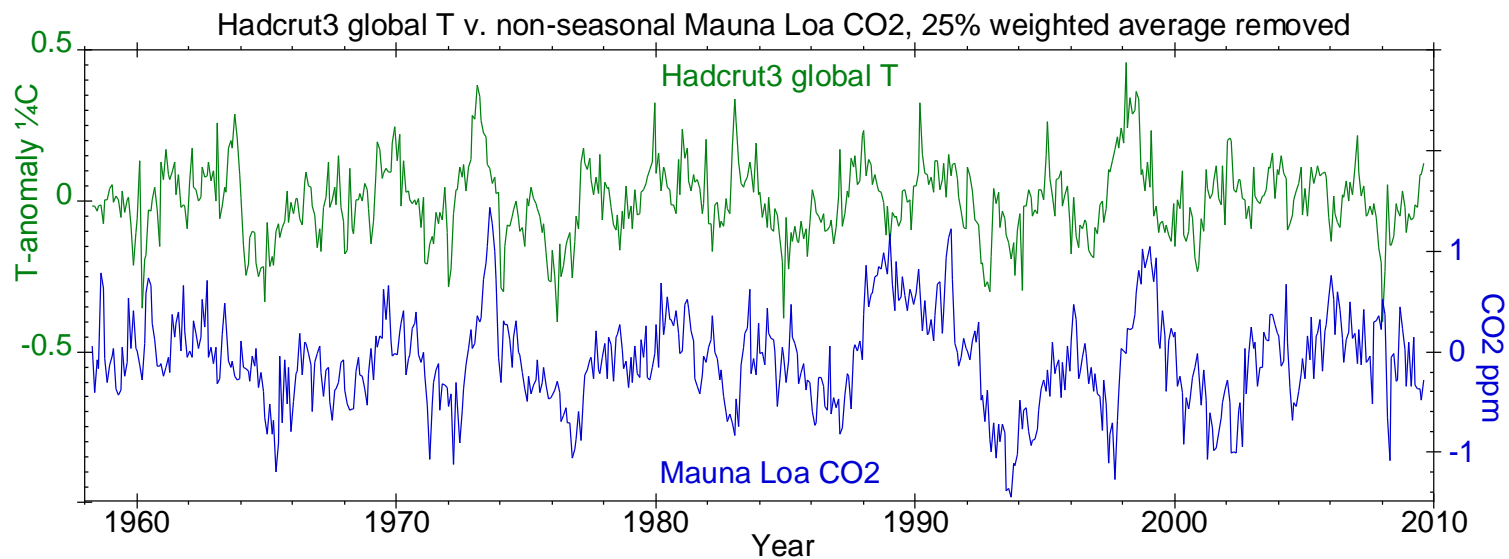
<http://www.cru.uea.ac.uk/cru/data/temperature>

Global Warming in 20th Century

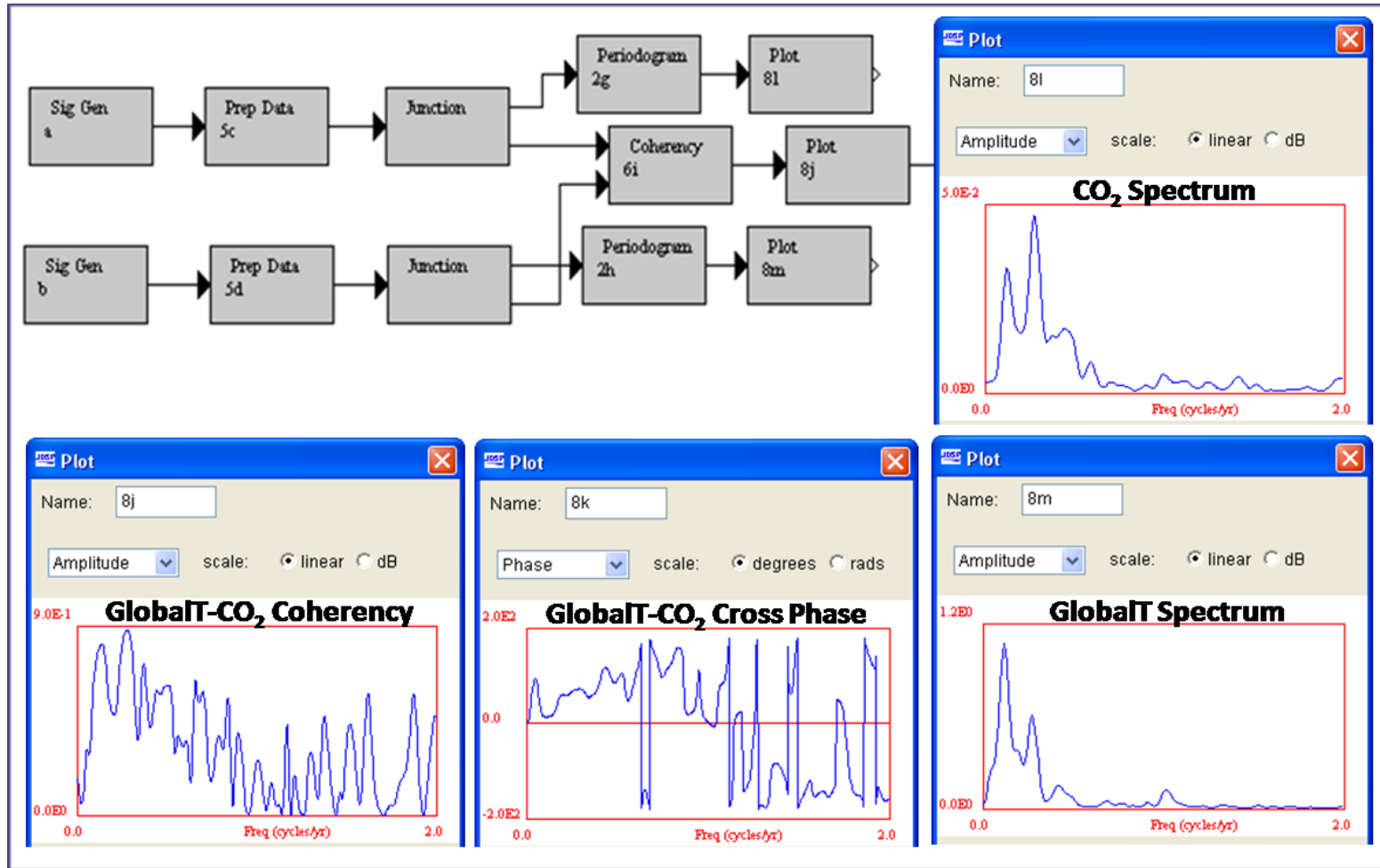
INTERANNUAL VARIATIONS ?

The “interannual” variations in global temperature and their relationship to those in $p\text{CO}_2$ provide additional important information, and are best assessed statistically, through signal processing and time series analysis.

The interannual variations of global temperature v. $p\text{CO}_2$ at Mauna Loa shown below (seasonal cycles and long-term trends removed) appear to share cyclic variations -- are these cycles significant and are they correlated?



Global Warming in 20th Century



- National Science Foundation
 - Grant award no. 0719714
 - Phase 3 Grant award no. 0817596



- SenSIP Center
School of ECEE
Arizona State University

