MATLAB Seminar

Επεξεργασία σήματος, εικόνας και βίντεο με το γραφικό περιβάλλον Simulink

Πανεπιστήμιο Αθηνών
Υπολογιστικό Κέντρο
Τρίτη 6 Μαρτίου 2012

Ομιλητής: Ζαχαρίας Γκέτσης

MENTOR Hellas
Scientific Engineering Software

Agenda

Επεξεργασία σήματος, εικόνας και βίντεο με το Simulink

- **Εισαγωγή στο Simulink (~20min)**
  Ανάπτυξη μοντέλων και προσομοίωση δυναμικών συστημάτων με μπλοκ διαγράμματα του Simulink.

- **Συστήματα επεξεργασίας σήματος, εικόνας και βίντεο (~2h)**
  Βιβλιοθήκες με μπλοκ γενικών δυναμικών συστημάτων καθώς και αλγορίθμων και συστημάτων επεξεργασίας σήματος, εικόνας και βίντεο. Εργαλεία σχεδιασμού και προσομοίωσης φίλτρων.

- **Παραγωγή κώδικα και υλοποίηση σε Hardware (~30min)**
  Εργαλεία αυτόματης παραγωγής κώδικα ANSI/ISO C/C++ και HDL. Δυνατότητες υλοποίησης σε DSPs, FPGAs και Real-Time Operating Systems.

MATLAB & Simulink in research and industry

- *Technical computing* (fft, linear algebra, mathematics)
- Application specific algorithms (image processing, bioinformatics, …)
- Algorithm development
- Test & Measurement
- System simulation
- Embedded design

Technical Computing

Image Processing

Test & Measurement

Computational Biology

Communication Systems

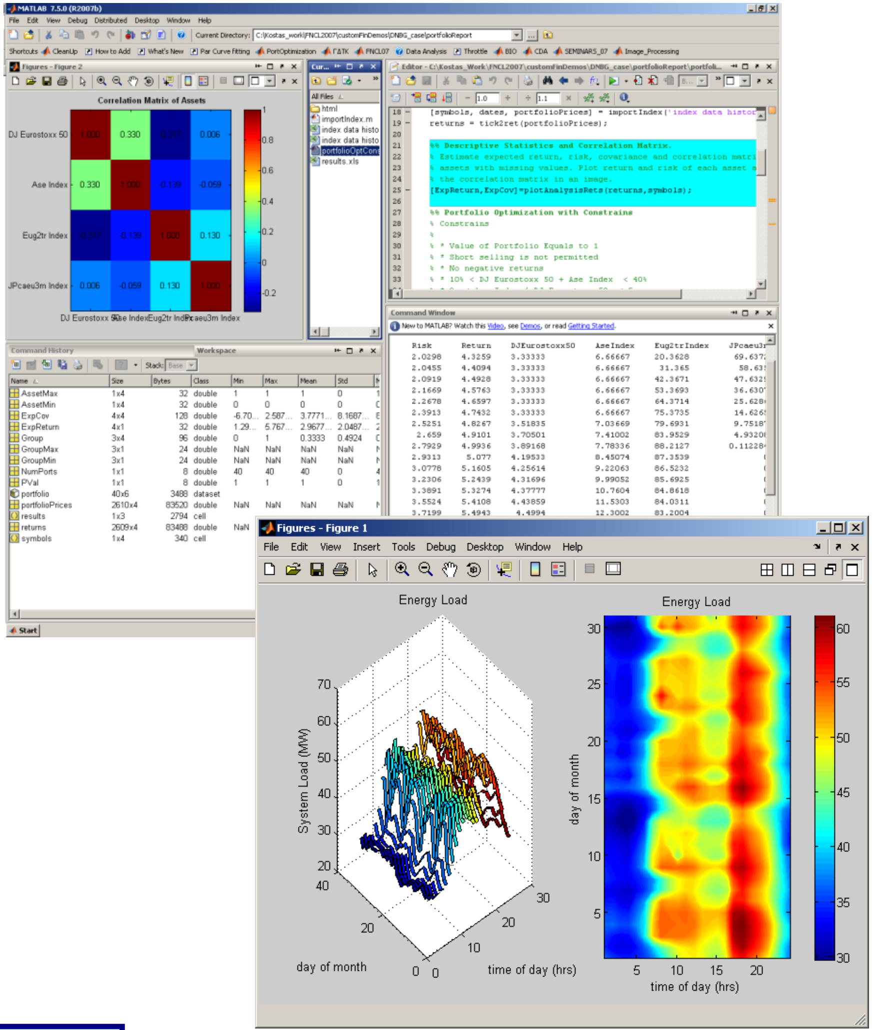
Control Systems

Computational Finance

MATLAB®

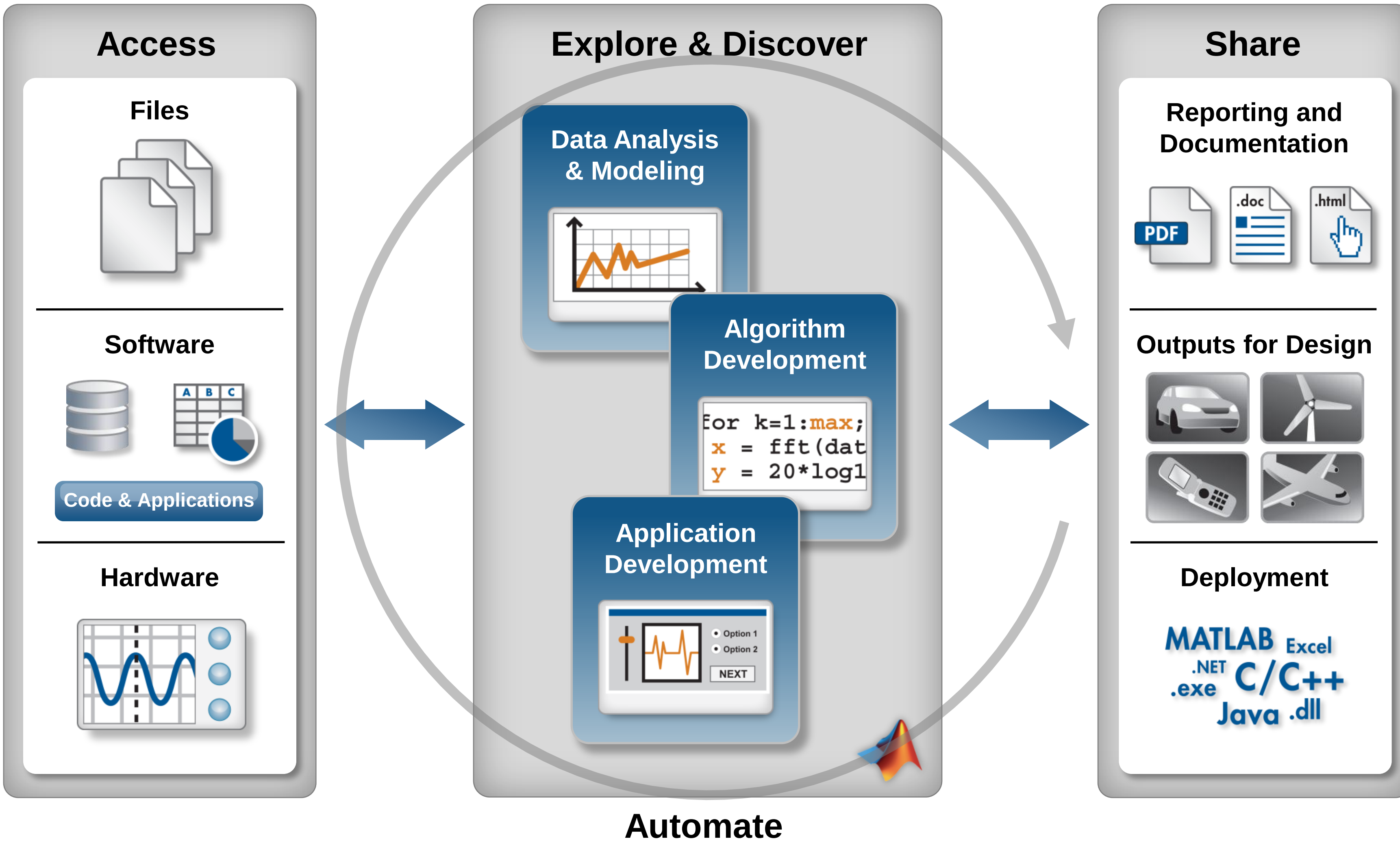
The leading environment for technical computing

- Quick Prototyping Environment
  - 1000’s of viewable-source functions
  - Powerful visualization engine
  - Powerful tools for application development
  - Optimized for matrix operations
- Integration with Existing Systems
  - Work with various data sources
  - Easily deploy for use with other programs
  - Leverage existing software



- > 1,000,000 Installations
- Third Party Software Companies > 300
- More than 1000 books

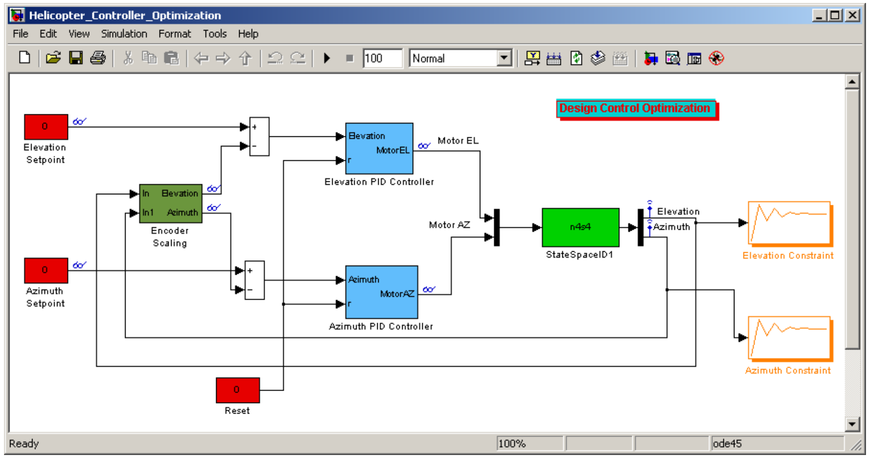
Technical Computing Workflow



Simulink

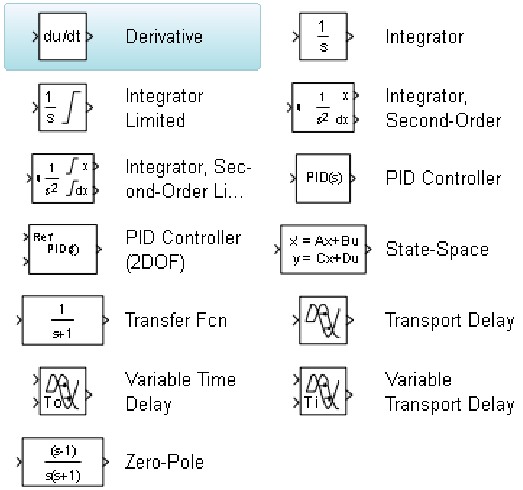
Environment for multidomain simulation and Model-Based Design for dynamic and embedded systems

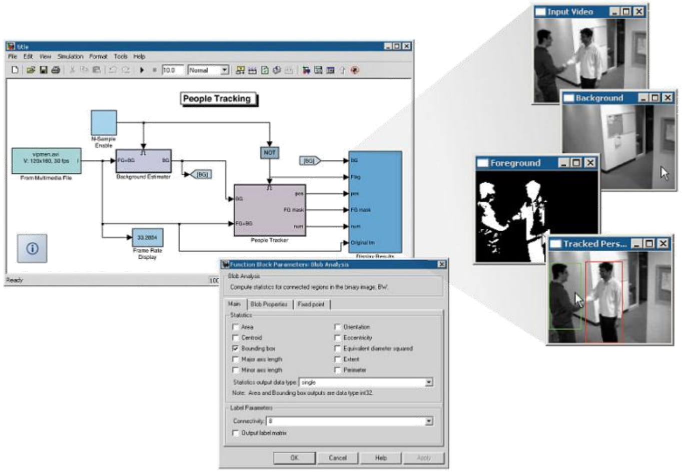
- GUI-based block diagram environment on top of MATLAB
- Simulation engine for modeling, designing and solving dynamic systems
- Advanced tools for
  - Automatic code generation
  - Continuous Test and Verification



Block Diagrams

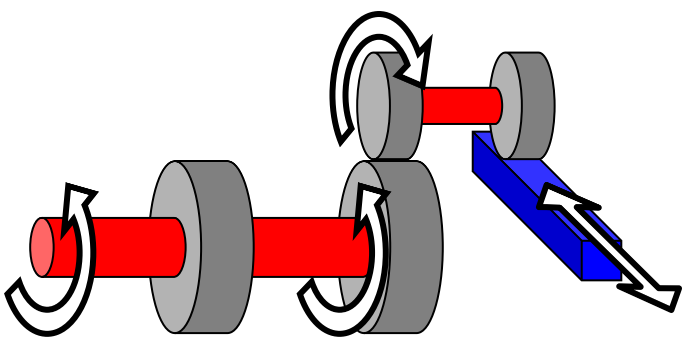
- Differential equations
  - Integrators, gains
  - Transfer functions
  - State space models
  - Lookup Tables
  - Core Simulink libraries
- Application-specific Blocksets
- External Legacy code (C/C++, ADA, Fortran)
- Embedded MATLAB Function Block

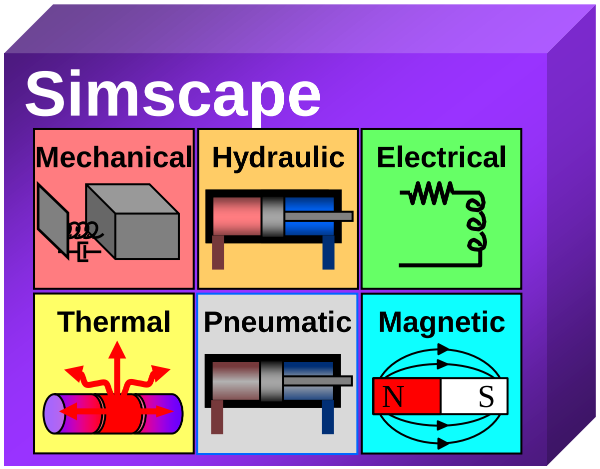


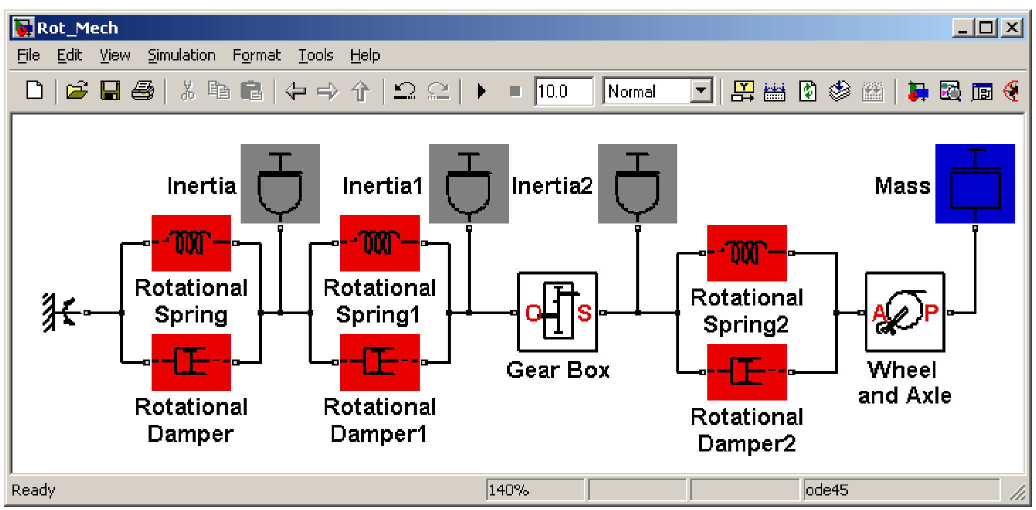


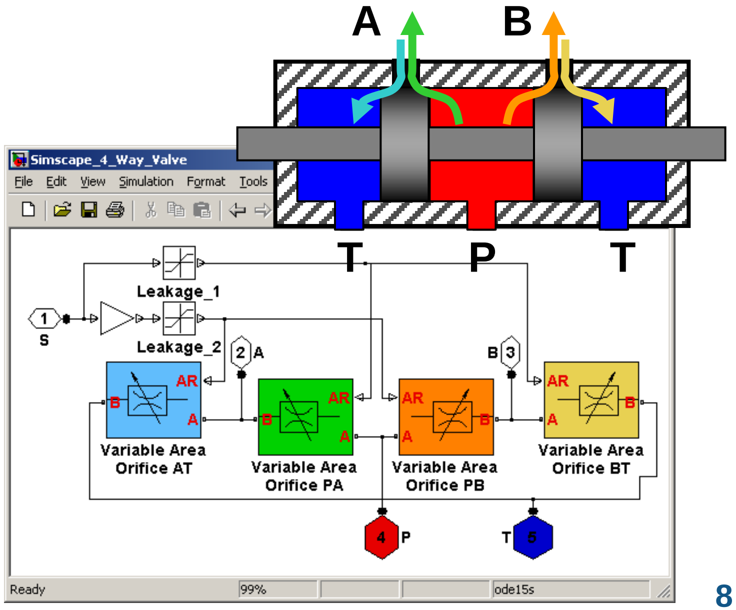
Physical Diagrams

- Library of physical elements
  - Mechanical, hydraulic, electrical…





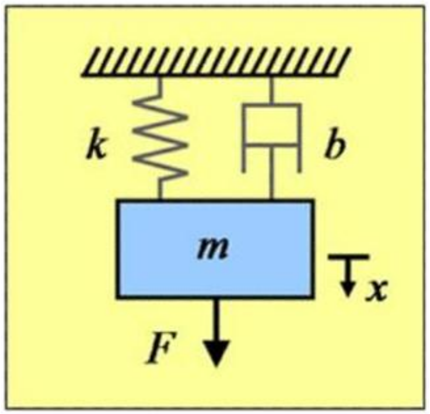




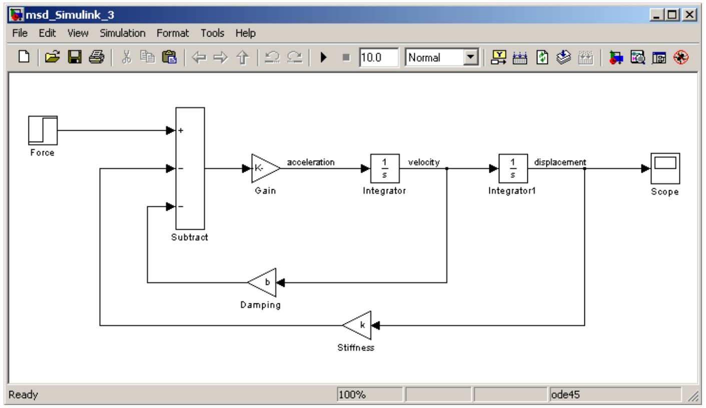
1 DOF mass-spring-damper system

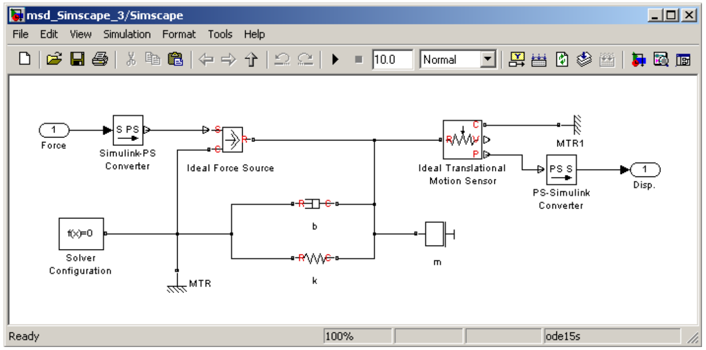
Demo

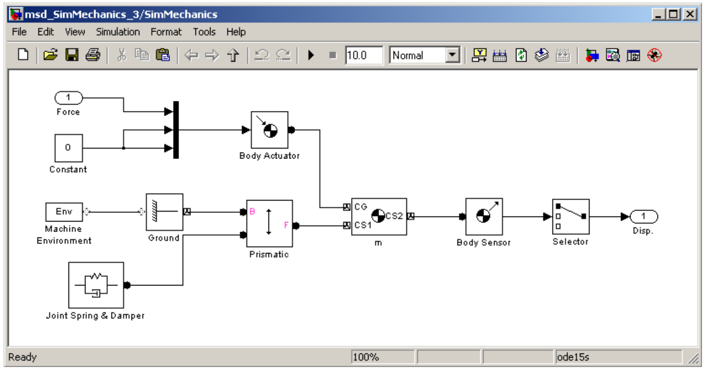
Model:



Problem: Model a 1 DOF mass-spring-damper system within the Simulink environment from first principles and using physical modeling tools.

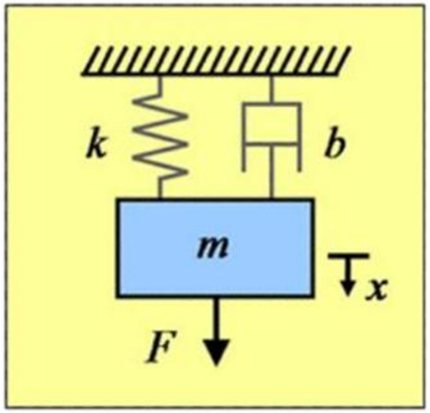






Derive the equations of motions of the system

- Use Newton's second law to derive EOM
- Take Laplace Transform to get transfer function



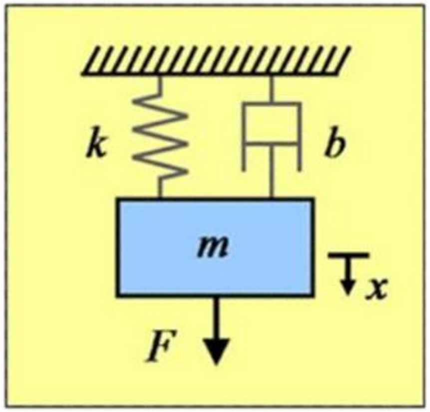
$$\begin{cases} m\ddot{x} = F(t) - kx - b\dot{x} \\ \ddot{x} = \dfrac{1}{m}\left[F(t) - kx - b\dot{x}\right] \end{cases}$$

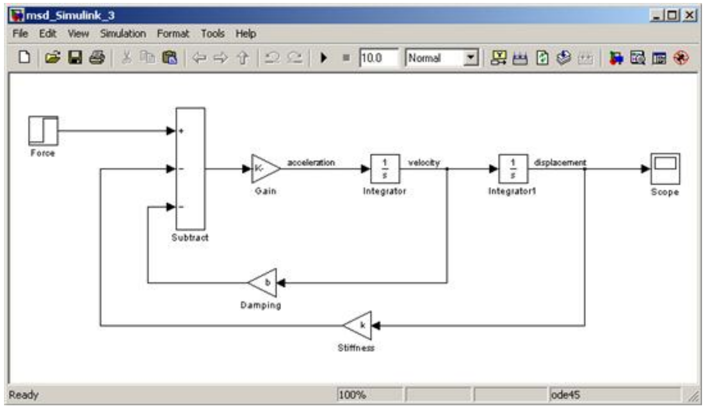
$$\begin{cases} ms^2\overline{X}(s) = \overline{F}(s) - k\overline{X}(s) - bs\overline{X}(s) \\ \dfrac{\overline{X}(s)}{\overline{F}(s)} = \dfrac{1}{ms^2 + bs + k} \end{cases}$$

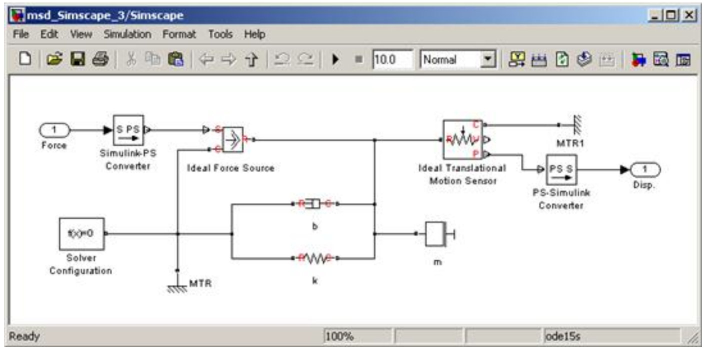
Model the mass-spring-damper system

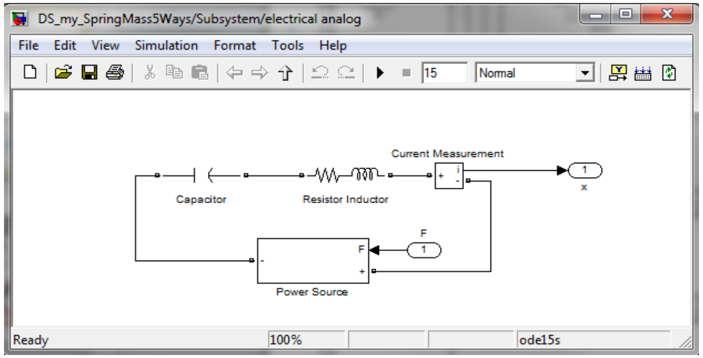
Using:

- First principles (EOM)
- Transfer function
- Physical models (mass, spring, damper)
- Electrical equivalent of the system
  (capacitor, resistor, power source)







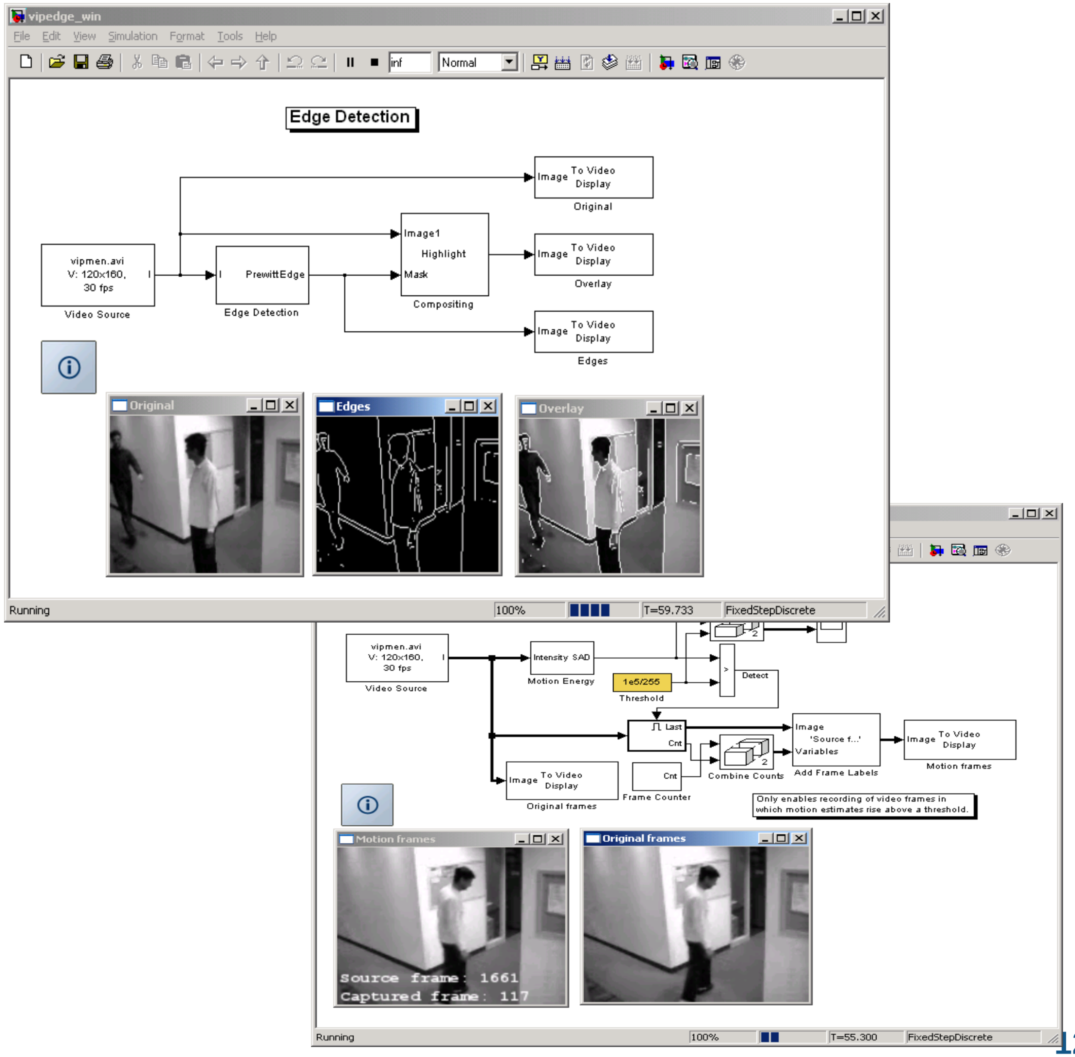


Edge Detection / Motion Detection

Demo

- Acquire live image/video data from a camera
- Find the edges of objects in the video input.

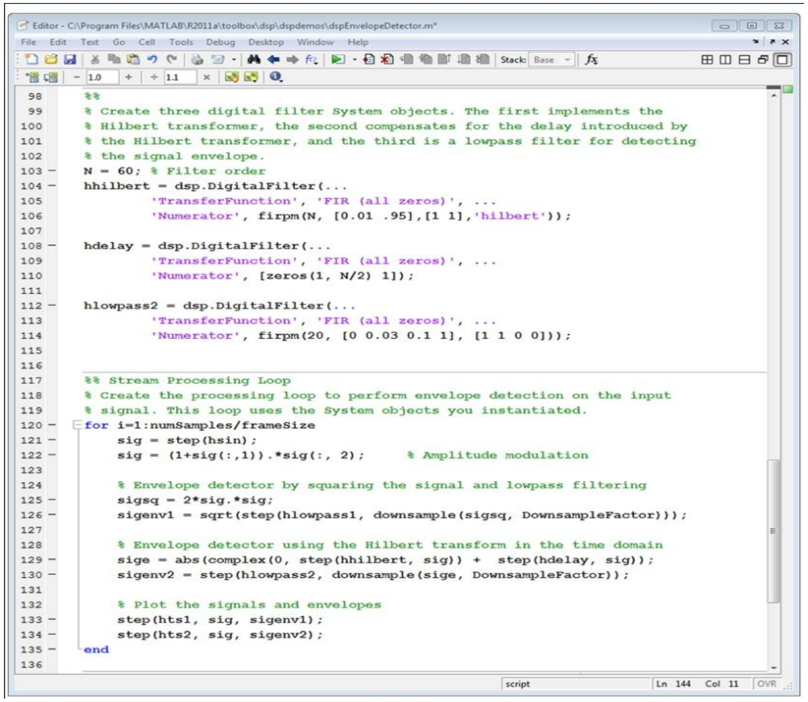
- Live motion detection using the sum of absolute differences (SAD) method
- Capture only the "interesting" video frames.



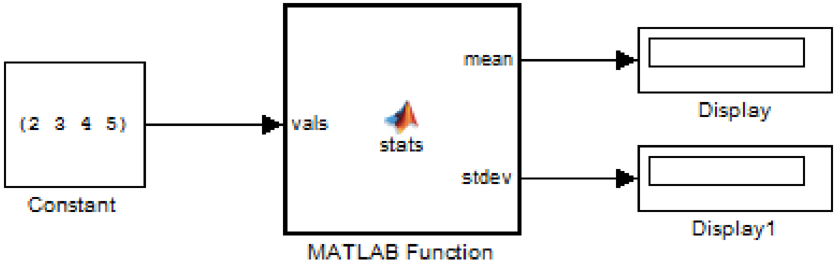
Using System Toolboxes with Simulink

Access algorithm libraries from either MATLAB or Simulink

- Use the algorithm as a MATLAB System object or as a Simulink block with identical implementations
- Use the MATLAB Function block to execute MATLAB code (including System objects) within Simulink models
- Reuse MATLAB-based IP in Simulink



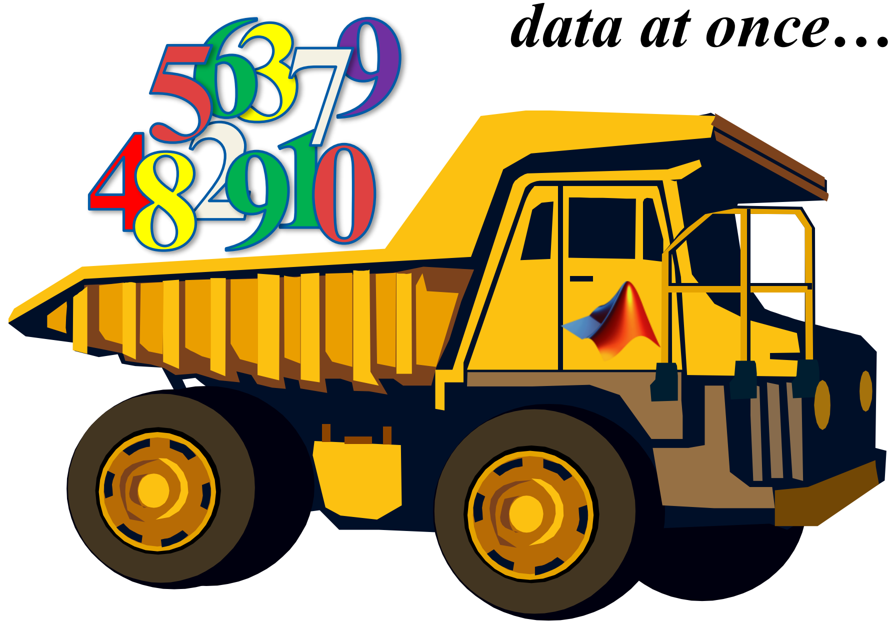

```
%%
% Create three digital filter System objects. The first implements the
% Hilbert transformer, the second compensates for the delay introduced by
% the Hilbert transformer, and the third is a lowpass filter for detecting
% the signal envelope.
N = 60; % Filter order
hhilbert = dsp.DigitalFilter(...
        'TransferFunction', 'FIR (all zeros)', ...
        'Numerator', firpm(N, [0.01 .95],[1 1],'hilbert'));

hdelay = dsp.DigitalFilter(...
        'TransferFunction', 'FIR (all zeros)', ...
        'Numerator', [zeros(1, N/2) 1]);

hlowpass2 = dsp.DigitalFilter(...
        'TransferFunction', 'FIR (all zeros)', ...
        'Numerator', firpm(20, [0 0.03 0.1 1], [1 1 0 0]));


%% Stream Processing Loop
% Create the processing loop to perform envelope detection on the input
% signal. This loop uses the System objects you instantiated.
for i=1:numSamples/frameSize
    sig = step(hsin);
    sig = (1+sig(:,1)).*sig(:, 2);      % Amplitude modulation

    % Envelope detector by squaring the signal and lowpass filtering
    sigsq = 2*sig.*sig;
    sigenv1 = sqrt(step(hlowpass1, downsample(sigsq, DownsampleFactor)));

    % Envelope detector using the Hilbert transform in the time domain
    sige = abs(complex(0, step(hhilbert, sig)) +  step(hdelay, sig));
    sigenv2 = step(hlowpass2, downsample(sige, DownsampleFactor));

    % Plot the signals and envelopes
    step(hts1, sig, sigenv1);
    step(hts2, sig, sigenv2);
end
```



Batch Processing

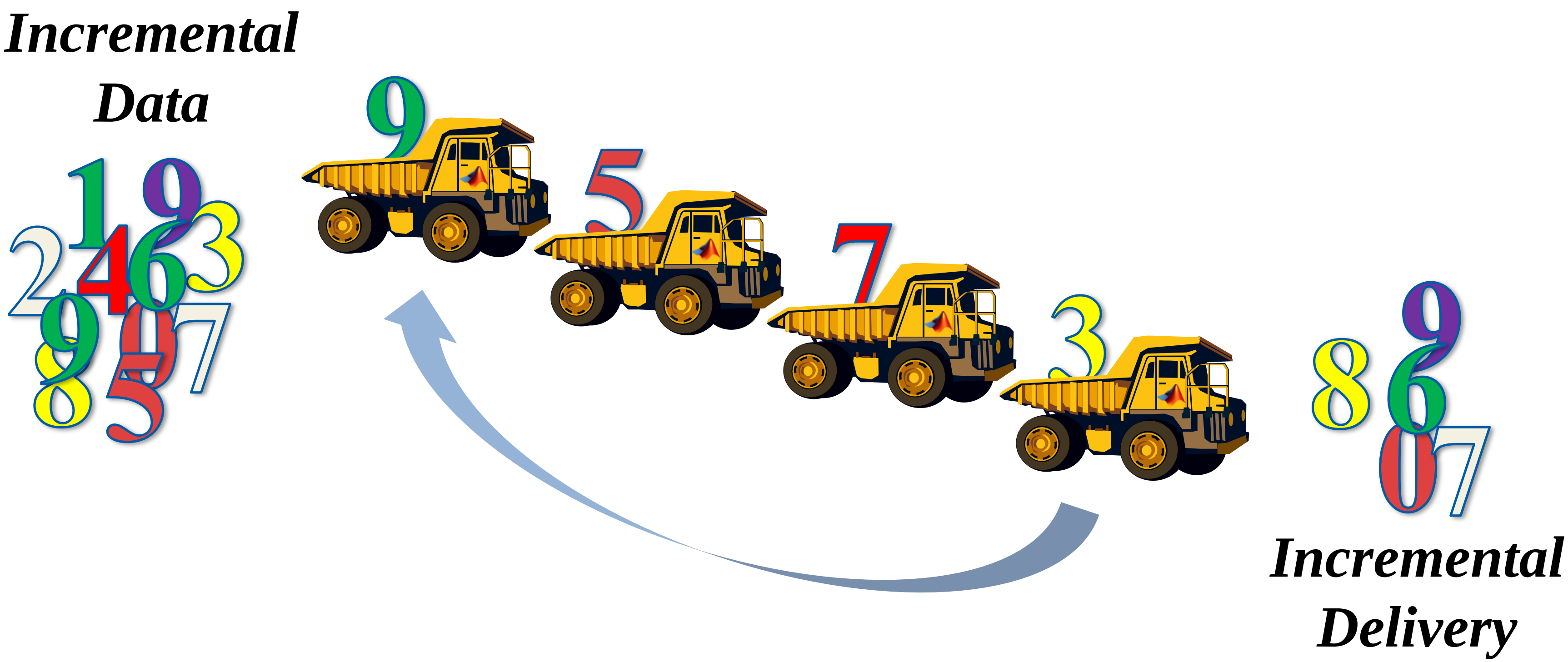
All the data

Work on all the data at once…



Deliver all at once

Stream Processing



- Data streams are potentially infinite in length
- Need to maintain time-based state information
- Need to be efficient with memory and performance

15

Batch Processing in MATLAB is Simple

Loads entire dataset into workspace

```
filename = 'dspafxf_8000.wav';
[audio Fs] = wavread(filename);
filt = fir1(40, 0.8, 'high');
audiofilt = filter(filt,1,audio);
wavplay(audiofilt,Fs);
```

Requires entire dataset to play audio

“audio” data uses more space than needed (double vs. uint16)

16

Stream Processing in MATLAB is Hard

```
%% Streaming the MATLAB way
% set up initializations
filename = 'dspafxf_8000.wav';
Fs = 8000;
info = mmfileinfo(filename);
num_samples = info.Duration*Fs;
frame_size = 40;
bLP = fir1(40, 0.8,'high');
zLP = zeros(1,numel(bLP)-1);
output = zeros(1,num_samples);

%% Processing in the loop
index= 1;
while index < (num_samples-frame_size+1)
    data = wavread(filename,[index index+frame_size-1]);
    [datafilt, zLP] = filter(bLP,1,data,zLP);
    output(index:index+frame_size-1) = datafilt;
    index = index + frame_size;
end
wavplay(output,Fs);
```

More code than in batch processing example

Explicit state management

Explicit indexing

Need to maintain output buffer

17

System Objects Make It Easier

New Feature

```
% set up initializations
filename = 'dspafxf_8000.wav';
hFilter = dsp.DigitalFilter;
hFilter.TransferFunction = 'FIR (all zeros)';
hFilter.Numerator = fir1(40, 0.8,'high');
hAudioSource = dsp.MultimediaFileReader(filename,  ...
        'SamplesPerAudioFrame',40,'AudioOutputDataType','double');
hAudioOut = dsp.AudioPlayer('SampleRate', 8000);

%% Processing in the loop
while ~isDone(hAudioSource)
    data     = step(hAudioSource);
    datafilt = step(hFilter, data);
    step(hAudioOut, datafilt);
end
```

Initialize objects

“In-the-loop” code is much simpler

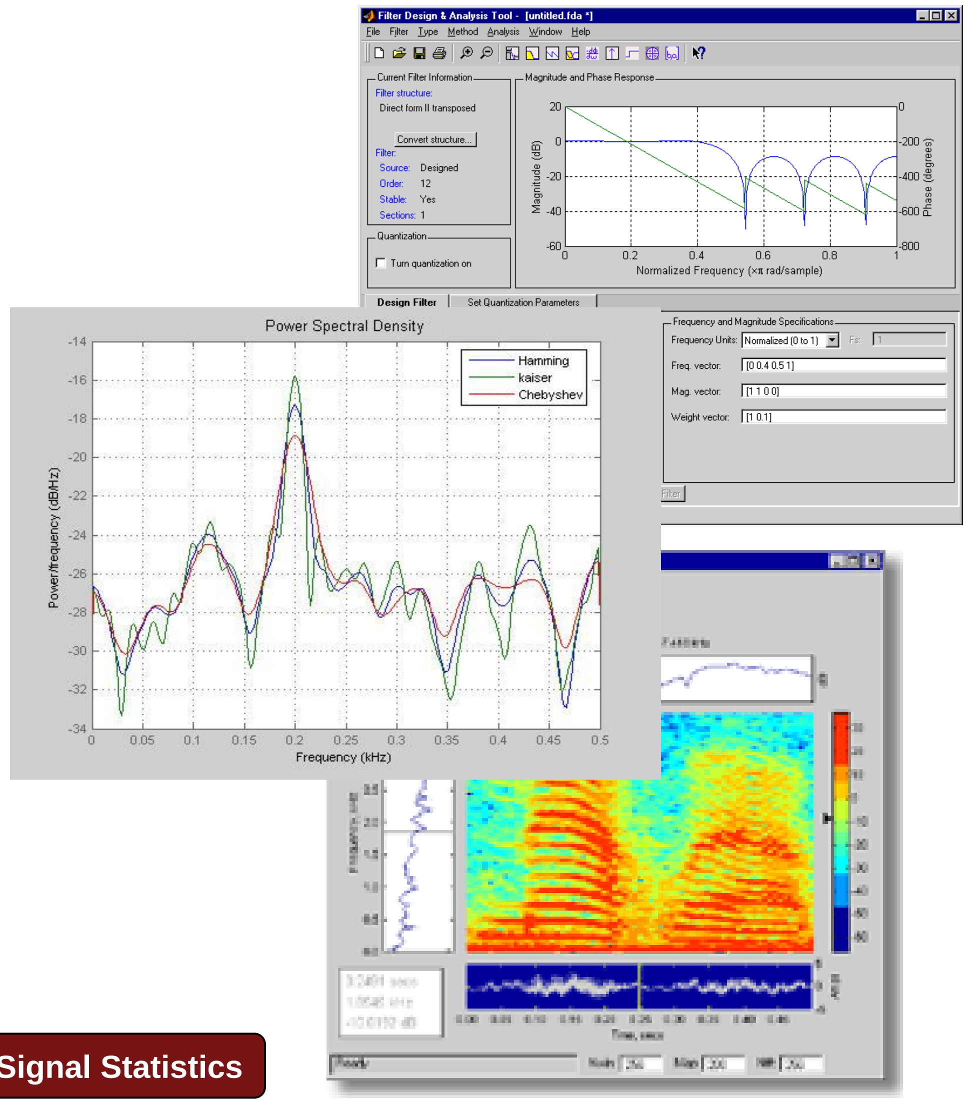
Implicit states and indexing

Audio player runs in-the-loop

DSP System Toolbox

Design and simulate signal processing systems

- Signal generators
- Filter design, analysis and implementation
- Transforms
- Statistical signal processing
- Spectral analysis
- Parametric time-series modeling

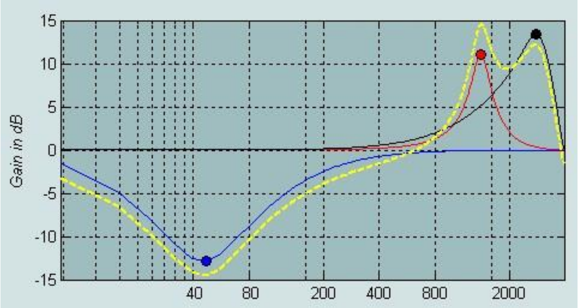


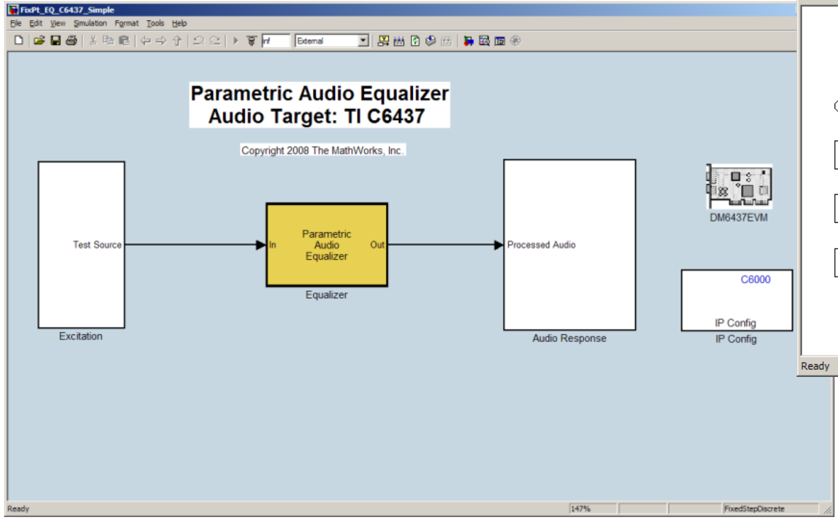
Demo: Signal Statistics

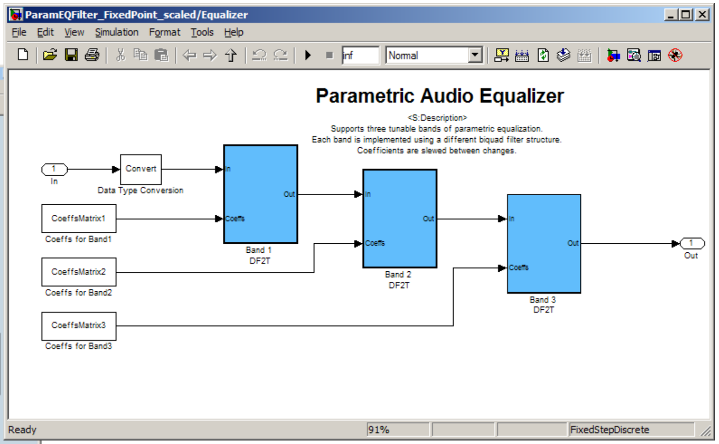
Parametric Audio Equalizer

Demo

- Three band fixed point audio equalizer whose response can be tuned to a desired audio characteristic
- Equalizer Range: -8 to +8 dB
  - Bass: 45 to 1200 Hz
  - Midrange: 2400 to 4800 Hz
  - Treble: 6 to 12 kHz
- MATLAB GUI is used to control and monitor the equalizer as it runs on the target hardware

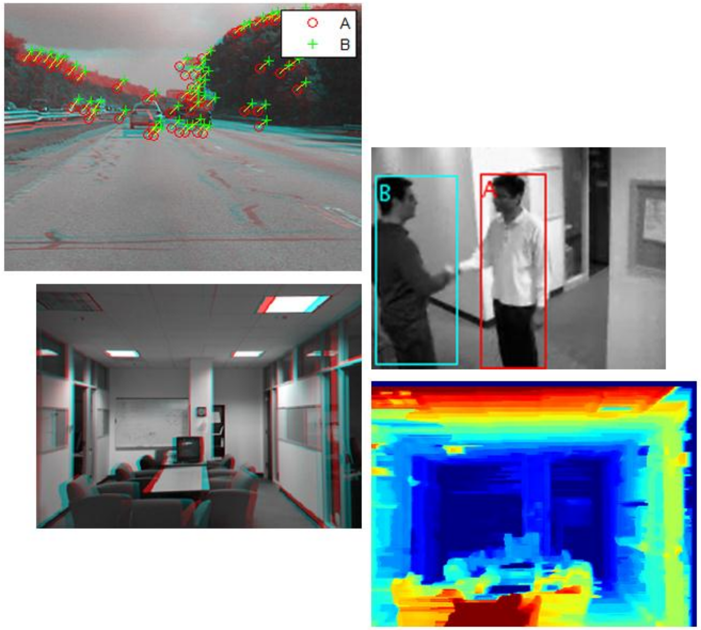






Computer Vision System Toolbox

Design and simulate computer vision and video processing systems

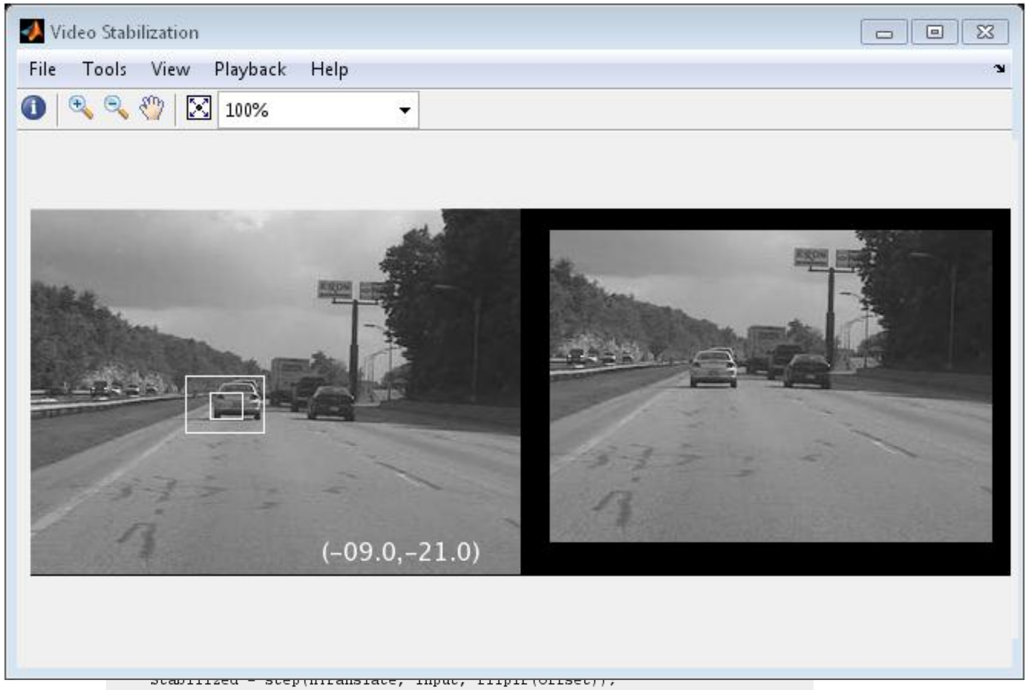


- Feature detection
- Feature extraction and matching
- Feature-based registration
- Motion estimation and tracking
- Stereo vision
- Video processing
- Video file I/O, display, and graphics

Video Stabilization using System Objects and Blocks

Demo

- Defines the target to track.
- Determines how much the target has moved relative to the previous frame.
- Remove unwanted translational camera motions and generate a stabilized video.

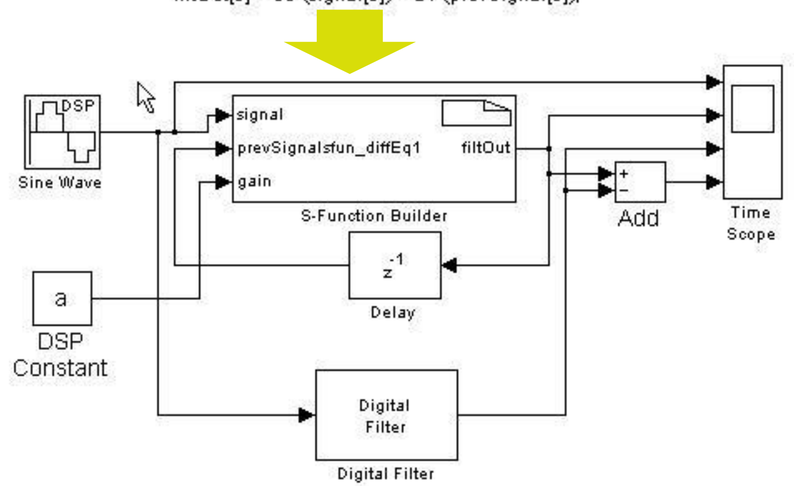



```
    Target = Stabilized(TargetRowIndices, TargetColIndices);

    % Add black border for display
    Stabilized(:, BorderCols) = 0;
    Stabilized(BorderRows, :) = 0;

    TargetRect = [pos.template_orig-Offset, pos.template_size];
    SearchRegionRect = [SearchRegion, pos.template_size + 2*pos.search_border];

    % Draw rectangles on input to show target and search region
    input = step(hShapeInserter, input, [TargetRect; SearchRegionRect]);

    % Display the offset values on the input image
    input = step(hTextInserter, input, Offset);

    % Display video
    step(hVideoOut, [input Stabilized]);
end
```

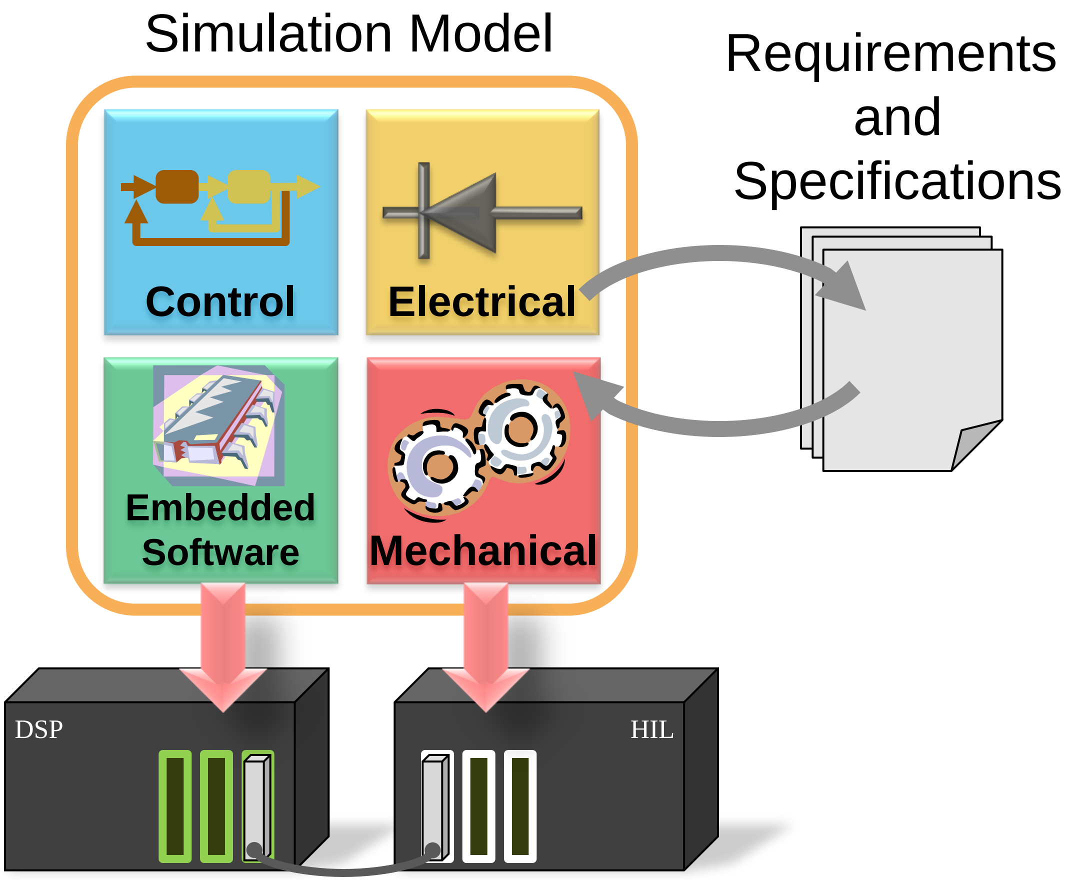
Integrating Existing Legacy Code into Simulink

- Εισαγωγή κώδικα C μέσα σε μοντέλα Simulink τόσο για διαδικασίες προσομοίωσης όσο και για διαδικασίες αυτόματης παραγωγής C κώδικα του συνολικού μοντέλου (αρχικού C-κώδικα και blocks του Simulink):
  - Hand Crafted S-Function
  - S-Function Builder
  - Legacy Code Tool

```
double a1,b0;

/* Limit gain */
if (gain[0] > 1)
   a1 = 1;
else if (gain[0] < 0)
   a1 = 0;
else
   a1 = gain[0];

/* Calculate filter const */
b0 = 1-a1;

/* Calculate difference equation */
filtOut[0] = b0*(signal[0]) + a1*(prevSignal[0]);
```



Model-Based Design Process

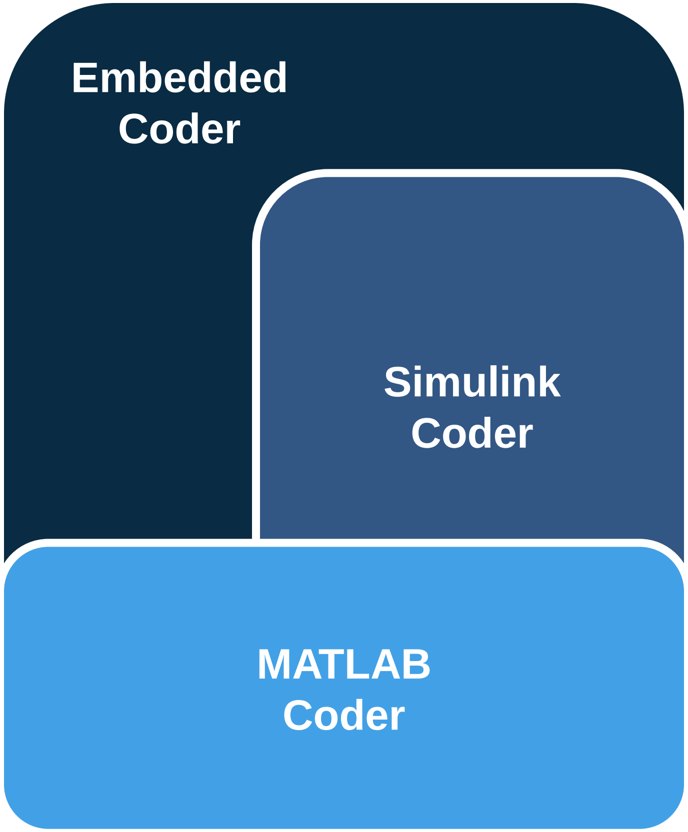


Save time by developing in a single simulation environment

Produce better designs by continuously comparing design and specification

Lower costs by using HIL tests and fewer hardware prototypes

Code Generation Technologies



Embedded Coder™
Automatically generate C and C++ optimized for embedded systems comparable to the efficiency of handwritten code

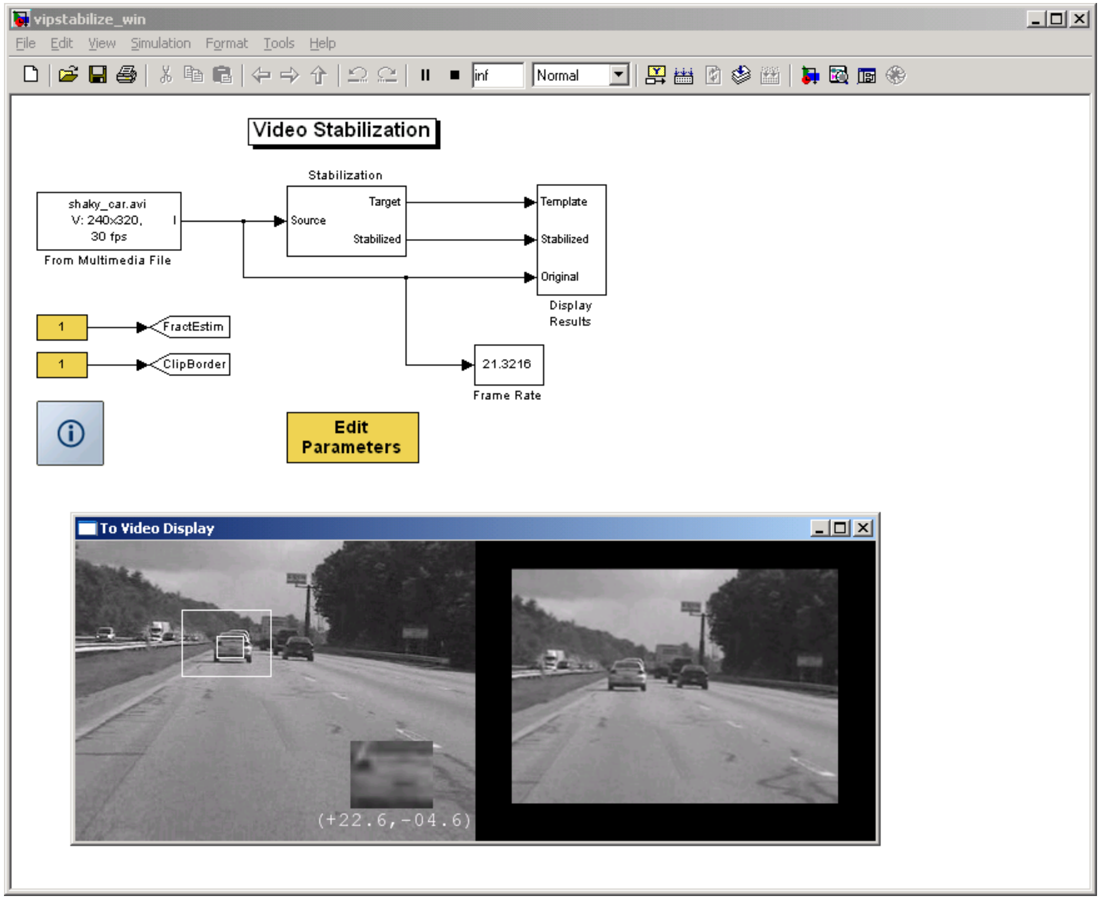
Simulink® Coder™
Automatically generate C and C++ from Simulink models and Stateflow charts for Rapid Prototyping and Hardware-in-the-Loop

MATLAB® Coder™
Automatically generate C and C++ from the suitable MATLAB subset

Video Stabilization & Implementation on DSP TI C6000

Demo

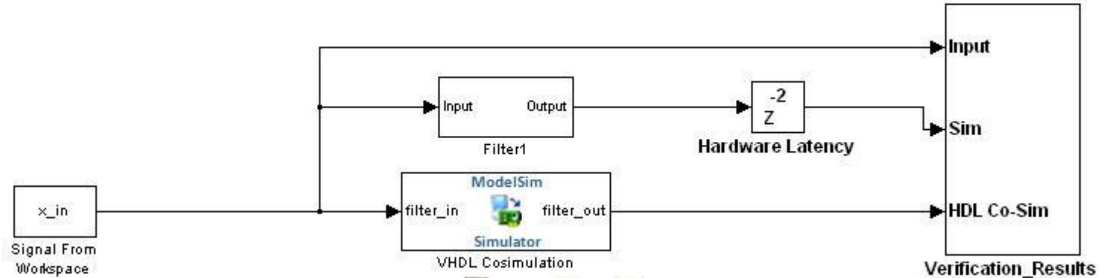
- Defines the target to track.
- Determines how much the target has moved relative to the previous frame.
- Remove unwanted translational camera motions and generate a stabilized video.
- Implement on DSP TI C6000

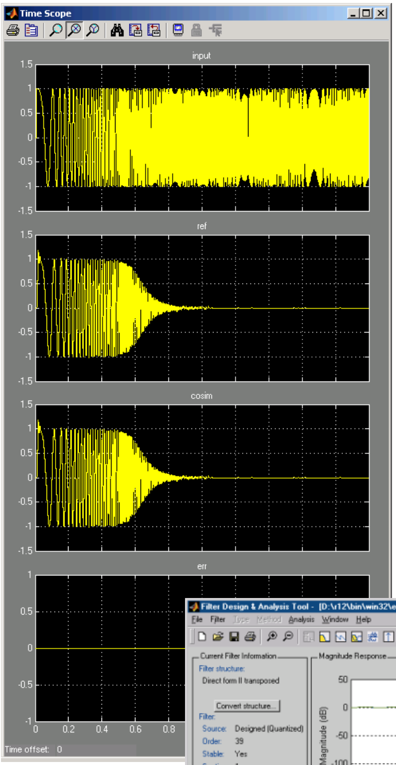


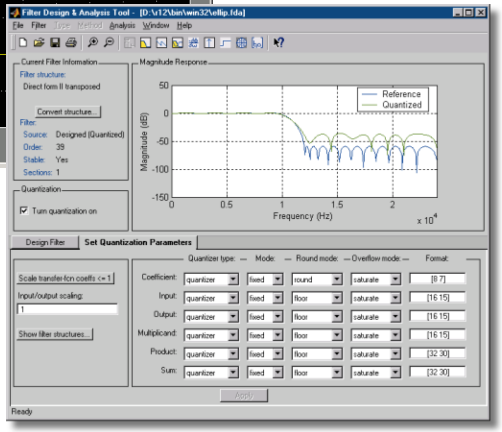
Design and Simulate Filters

Demo

- Design filters (IIR,FIR, …)
- Fixed Point & Floating Point
- Generate HDL Code
- Co-simulate HDL Code
- Implement on FPGAs

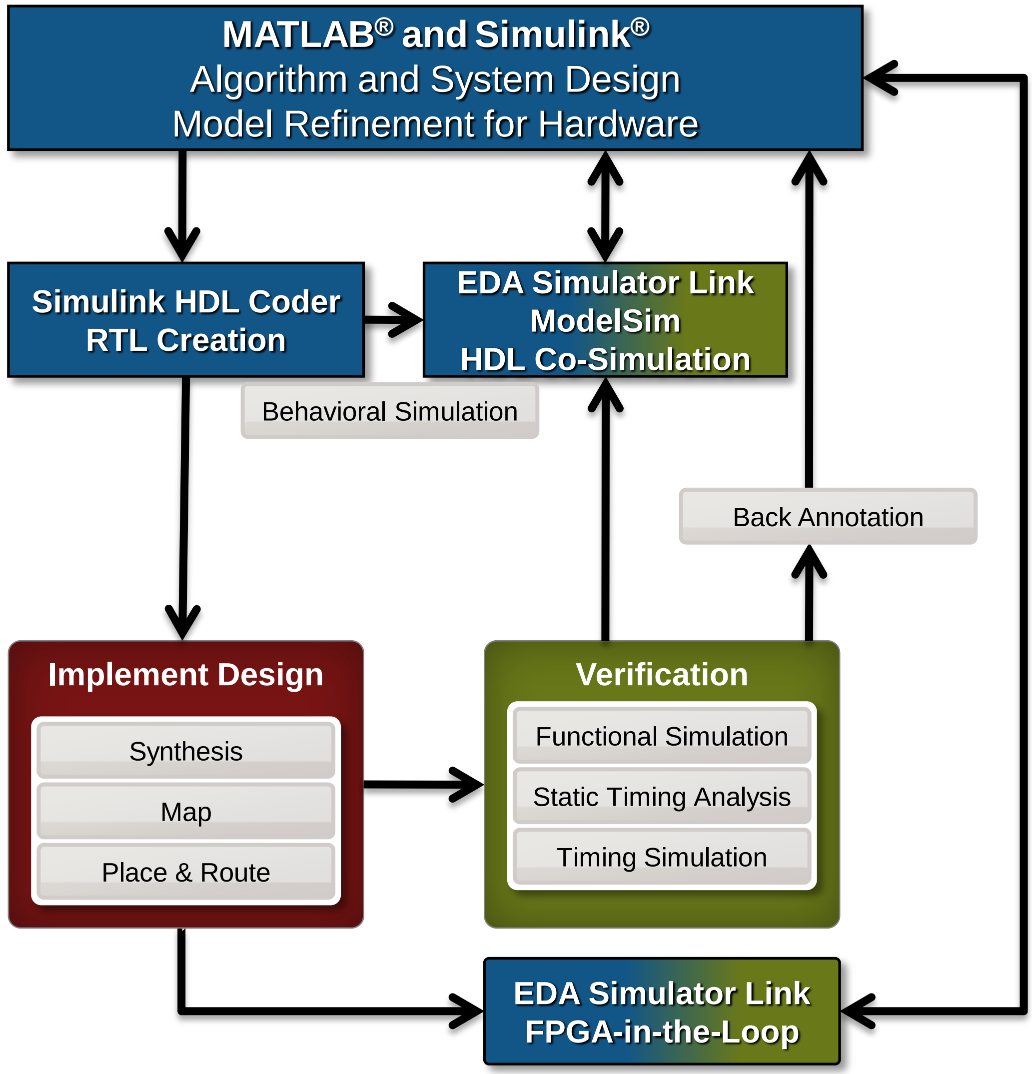






From Algorithm to FPGA Implementation

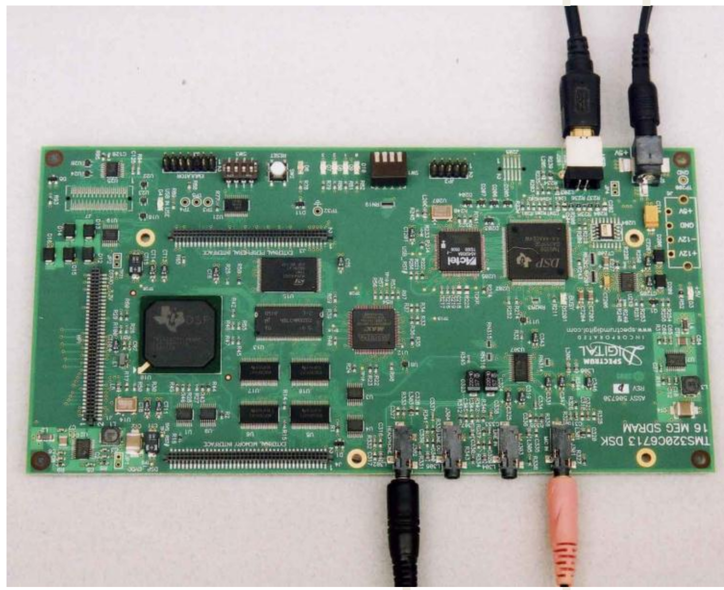
Demo: Symmetric FIR



Signal Processing Laboratory

USB to PC

to +5V



Headphones

Microphone

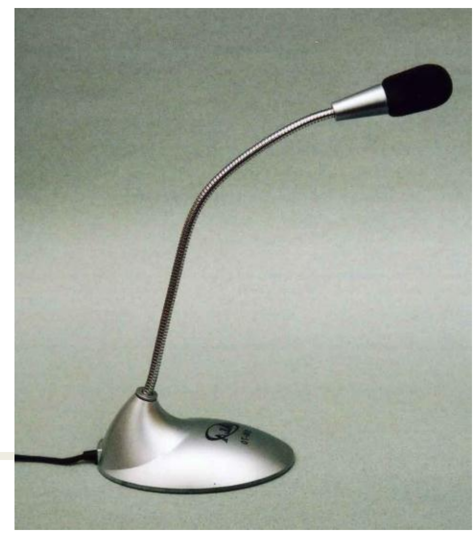
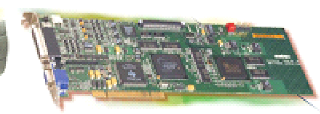


Image and Video Processing Laboratory

- Image and Video Acquisition
  - Frame grabbers
  - Digital cameras / DCAM-compatible cameras
  - Windows video devices / Third-party adaptors

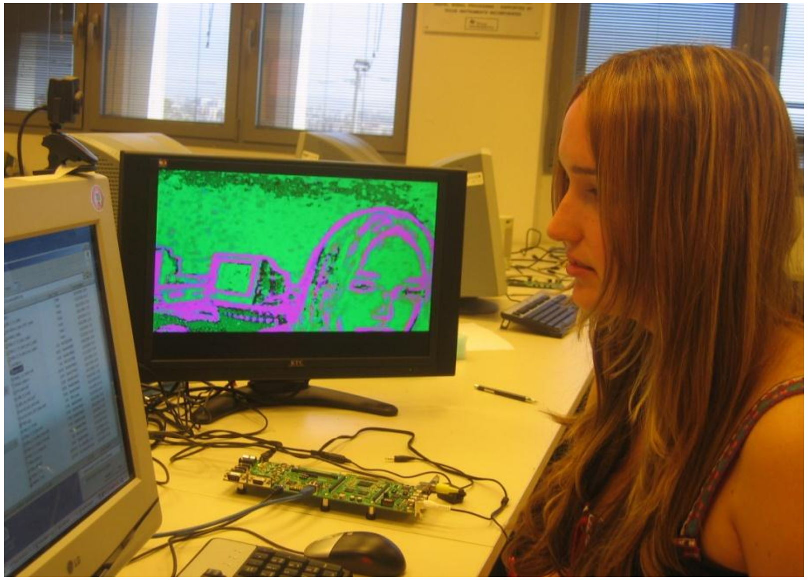
Cameras and Video Equipment


- Real-Time Processing
  - Implement on DSPs
  - Implement on FPGAs

- Educational models

Educational Models



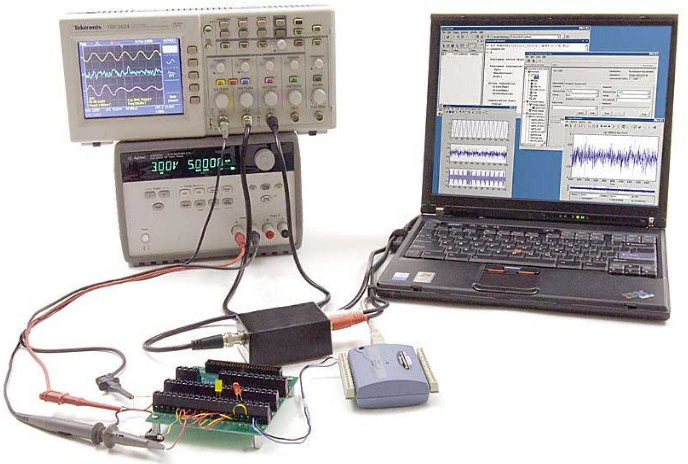
Texas Instruments   ANALOG DEVICES

ALTERA   XILINX

Control Systems Laboratory

- Data Acquisition Boards
- Real-Time Control Systems
  - microcontrollers
  - Target PCs
- Educational Models
  - Helicopter Model
  - Ball and Plate Model
  - Magnetic Levitation Model
  - 3 DOF Gyroscope
  - Lego NXT

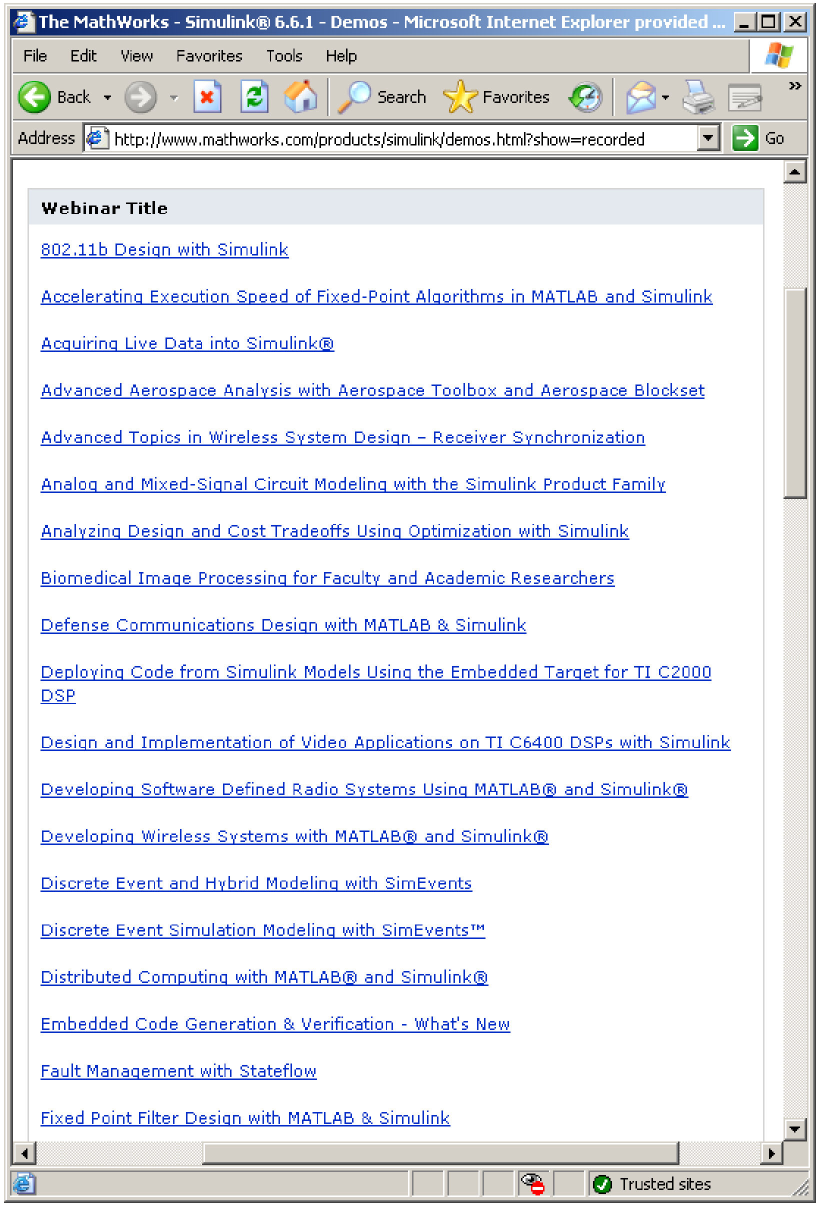
Plug-in data acquisition boards



Educational Models

Επόμενα βήματα

- Ψάξτε για “recorded webinars” στο site www.mathworks.com . Θα βρείτε περισσότερα από 75 webinars για το Simulink, μεταξύ άλλων τα:
  - **Εισαγωγή στο Simulink**
  - **Introduction to Simulink for Control Design**
  - **Introduction to Simulink for Signal Processing and Communications**
  - **Image and Video Processing with Simulink**



Επικοινωνία & πληροφορίες

MENTOR Hellas
Scientific Engineering Software

Τηλ: 210 – 60 31 121
Fax: 210 – 60 31 024
e-mail: info@mentorhellas.com

Μάθετε τα νέα μας στο διαδίκτυο:

επισκευθείτε το web site μας : www.mentorhellas.com

επικοινωνήστε μαζί μας: www.mentorhellas.com/contact

Follow us on twitter **:** twitter.com/mentorhellas

Join our group in LinkedIn **:**

MATLAB seminars for professional engineers in Greece

www.linkedin.com/groups/MATLAB-seminars-professional-engineers-in-3128153