



Model-Based Design of Video Applications for TI DSPs

Dick Benson

Consultant Application Engineer
dick.benson@mathworks.com



Agenda

- Introduction to Model-Based Design (15 min)
- Basics of Simulink® (15 min)
- Design and Implementation of Video Applications (35 min)
 - *Edge detection example*
- Advanced Video Applications (15 min)
 - *Video stabilization example*
- Next Steps and Discussion (10 min)



System Design Challenges

- Increasing system complexity and computation demands
- Embedded system resource constraints
 - Real-time requirements
- Designing for a target processor
 - Micro-controller, GPP, DSP, FPGA
- End-product price, power, size
 - Roadmap for adding features, performance
- Testing and validating results



Problems with Traditional Development

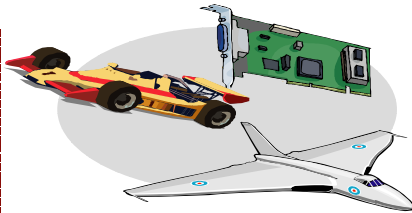
Requirements and Specifications



Text-based

- Prevents rapid iteration

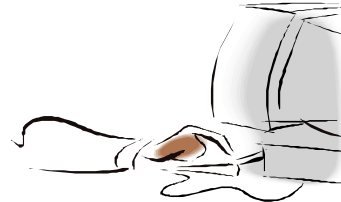
Design



Physical prototypes

- Incomplete and expensive

Implementation



Manual coding

- Introduces human error

Test and Verification



Traditional testing

- Errors found too late in the process



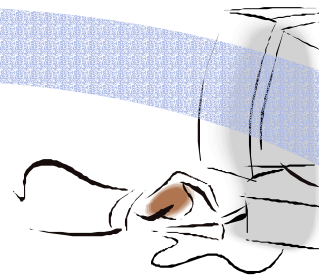
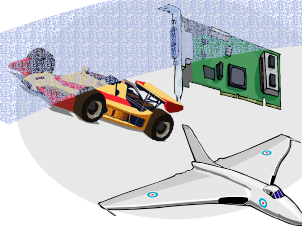
Advantages of Model-Based Design

Requirements and Specifications

Design

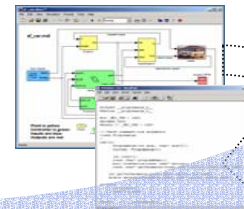
Implementation

Test and Verification



Continuous Verification

Model Elaboration



Executable models

- Unambiguous
- Only "one truth"

Simulation

- Reduces "real" prototypes
- Systematic "what-if" analysis

Automatic code generation

- Minimizes coding errors

Test with Design

- Detects errors earlier



TI TECH DAYS

Real Results Across Industries

TOYOTA

Standard for Powertrain Controls
Production Code Development

LOCKHEED MARTIN



JSF Flight Control System



W-CDMA Baseband Processors



TEXAS INSTRUMENTS

Specialty Chipsets for DSP
Customers

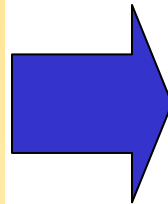
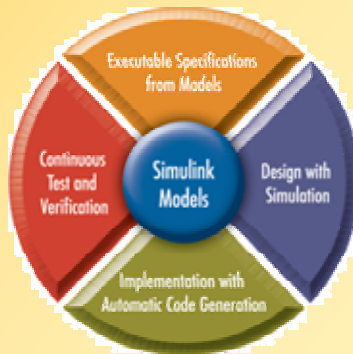




The Value of Model-Based Design

Model-Based Design

- **Executable specification**
- **Design with simulation**
- **Implementation through code generation**
- **Continuous test and verification**



Innovation

- **Rapid design iterations**
- **“What-if” studies**
- **Unique features and differentiators**

Quality

- **Reduce design errors**
- **Minimize hand coding errors**
- **Unambiguous communication internally and externally**

Cost

- **Reduce expensive physical prototypes**
- **Reduce re-work**
- **Reduce testing**

Time-to-market

- **Get it right the first time**

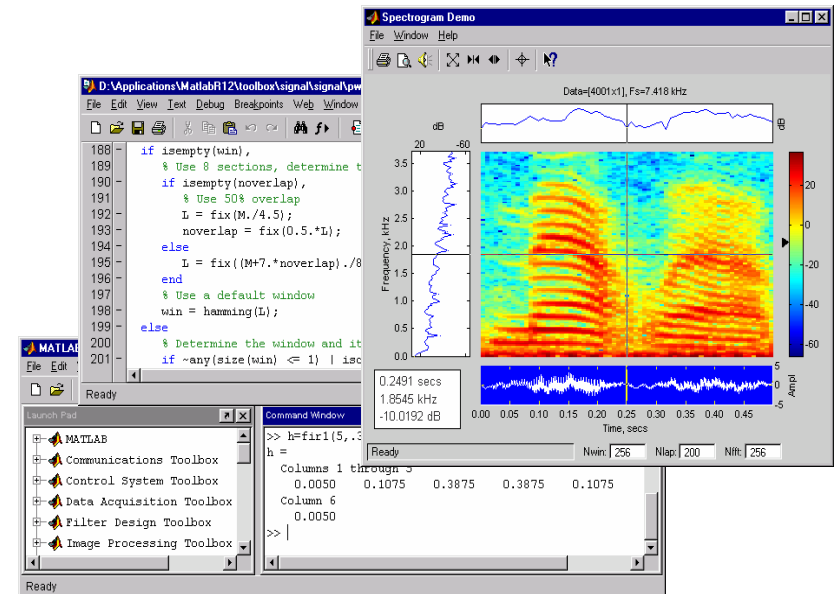


Technical Computing

MATLAB®

The leading environment for technical computing

- The *de facto* industry-standard, high-level programming language for algorithm development
- Toolboxes for control system design, signal and image processing, statistics, optimization, symbolic math, and other areas
- Foundation of the MathWorks product family



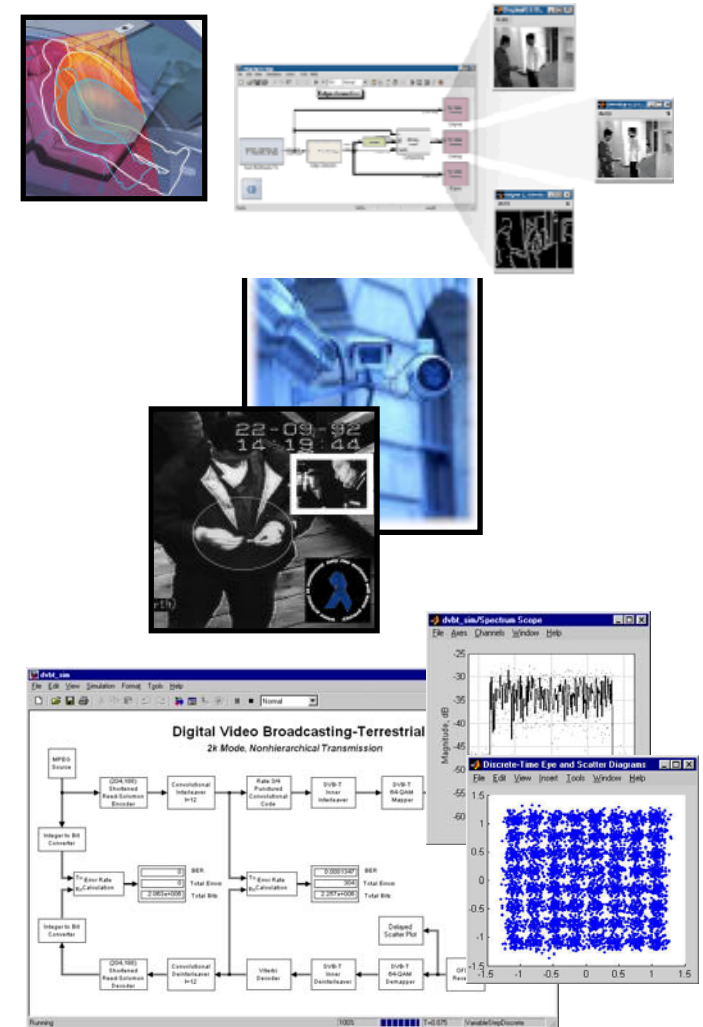


Model-Based Design

SIMULINK

The leading environment for modeling, simulating, and implementing dynamic and embedded systems

- Foundation for model-based design, including physical-domain modeling, automatic code generation, and verification and validation
- Open architecture for integrating models from other tools
- Applications in controls, signal processing, communications, and other system engineering areas





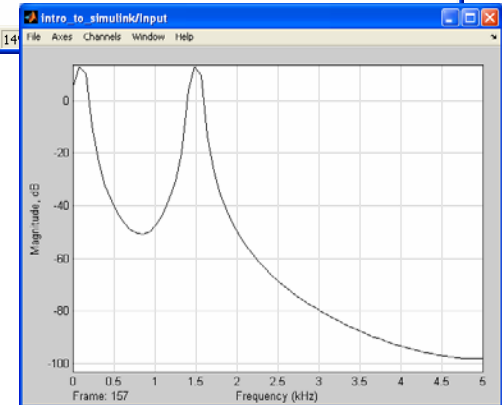
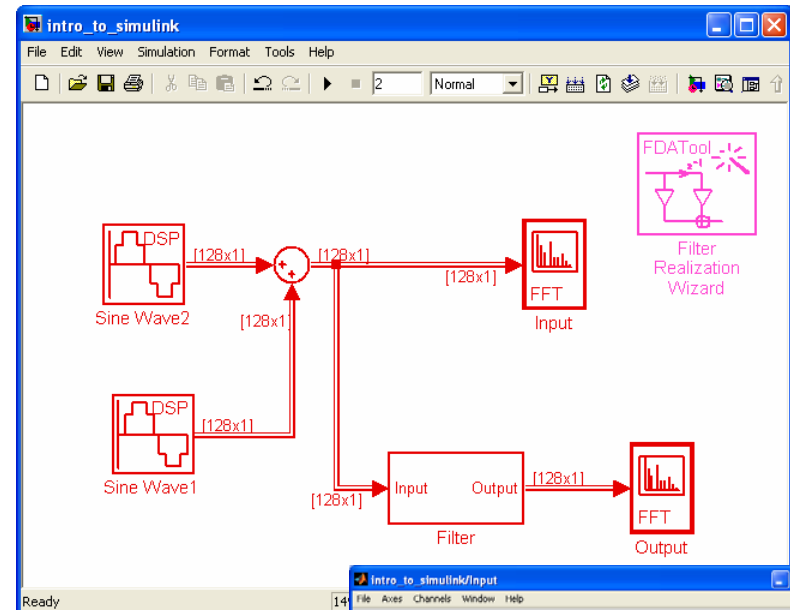
TI TECH DAYS

Basics of Simulink



Basics of Simulink

- Simple signal processing model
 - Signal Processing Blockset
 - SP Sources: Sine Wave
 - SP Sinks: Spectrum Scope
 - Filtering: Filter Designs: Filter Realization Wizard
 - Simulink
 - Math Operations: Sum

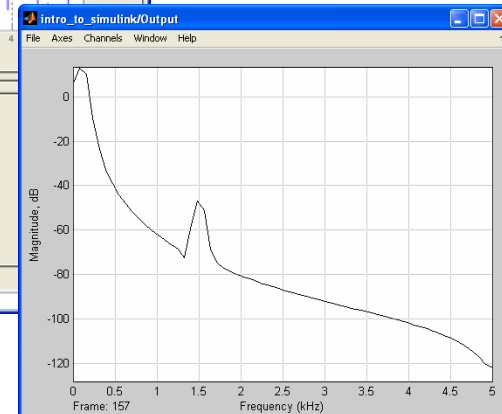
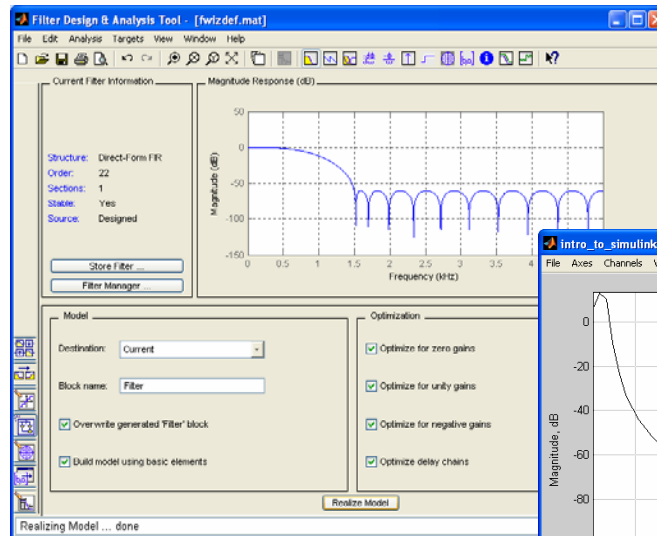


Demo



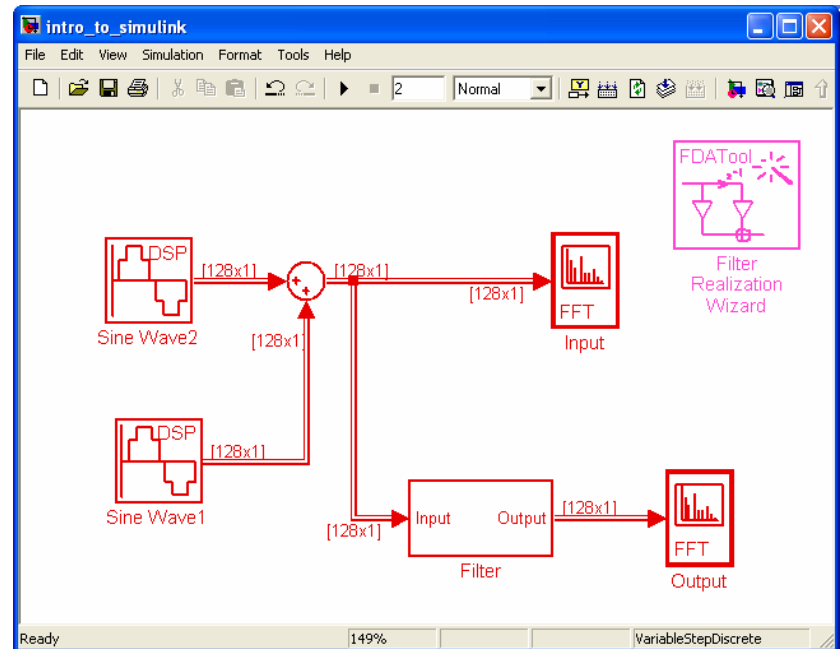
Basics of Simulink (continued)

- Build low-pass filter
- Filter out the high-frequency tone



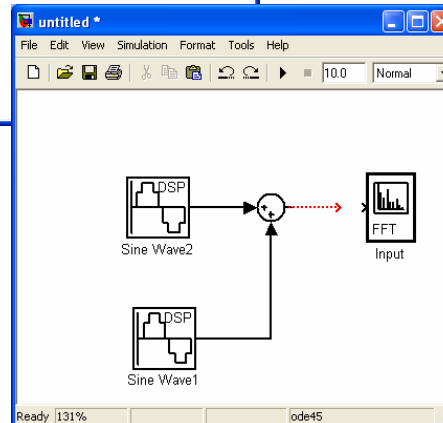
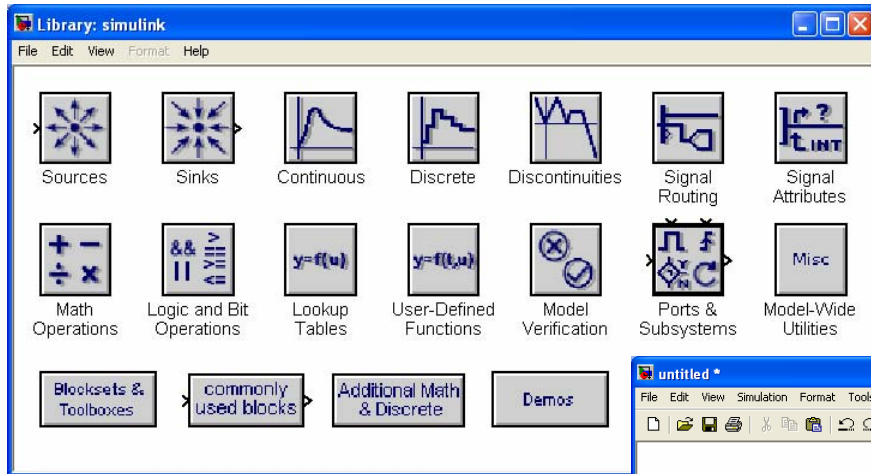
Basics of Simulink (continued)

- Design, simulate, test, and visualize with Simulink
- Frame-based processing
- Use M-code and filters designed in MATLAB





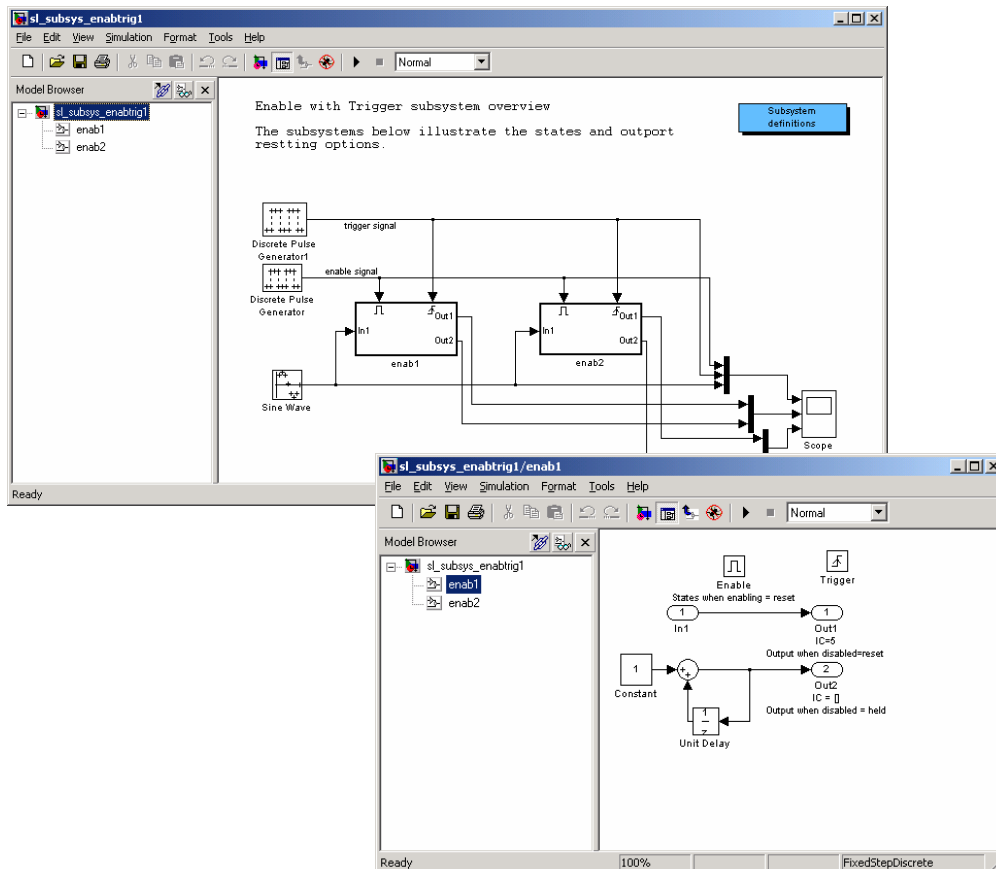
Model Construction



- Drag and drop
- Connect
- Digital
 - Fast frame-based simulation
- Analog
 - Variable-step numerical integration solvers
 - Zero-crossing detection



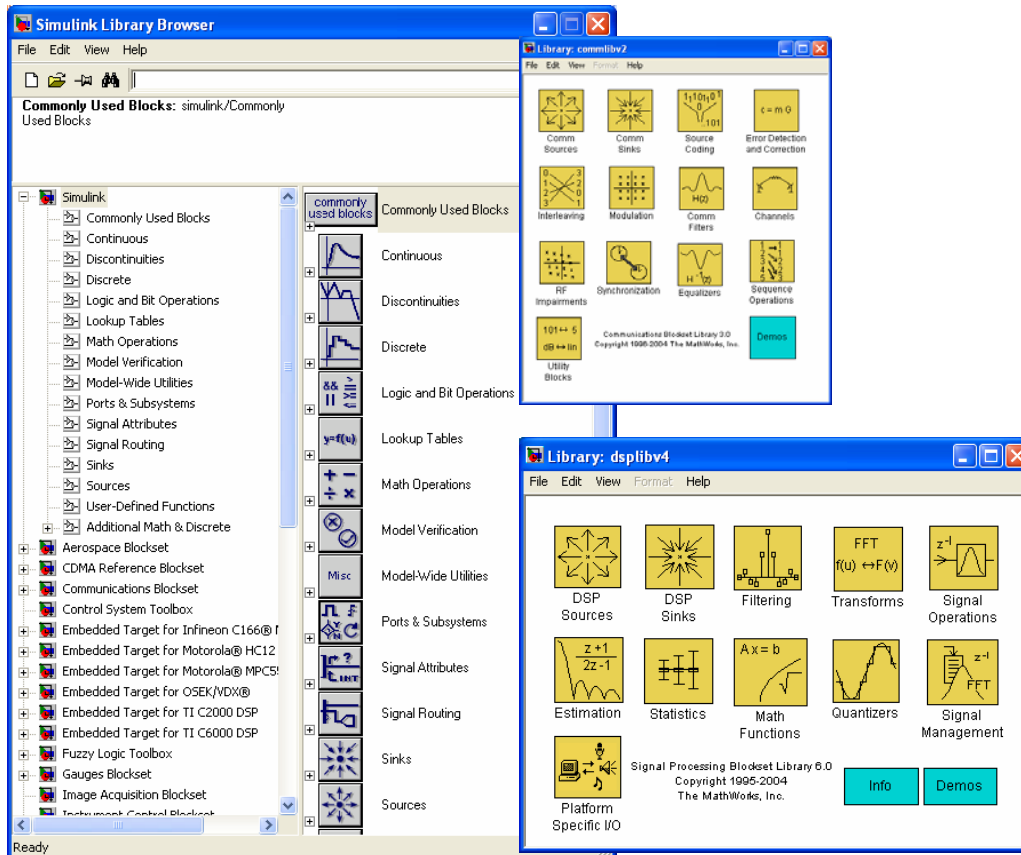
Sub-Systems and Hierarchy



- Group multiple blocks into subsystem to any level
- Model browser
- Conditionally executed subsystem
 - Enabled and triggered
 - If, while, for, switch
- Configurable subsystem
 - Swap model components easily



The Block Libraries

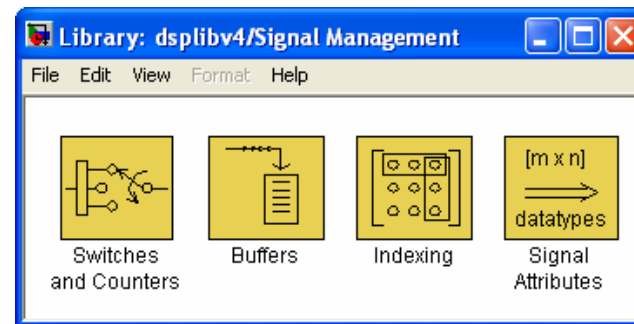
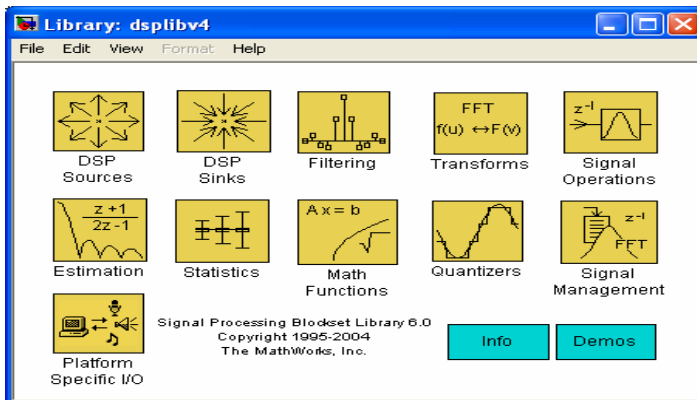
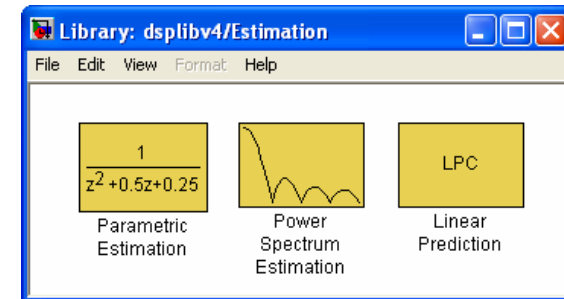
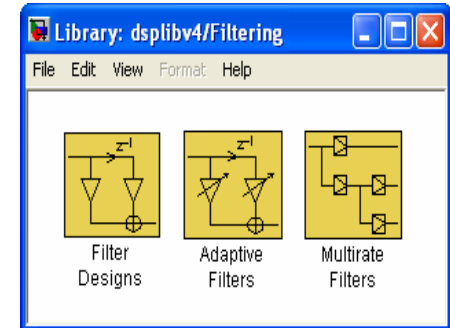


- Simulink
 - Sources
 - Sinks
 - Continuous
 - Discrete
 - Nonlinear
 - Math
- Simulink Fixed Point
- **Signal Processing Blockset**
- **Video and Image Processing Blockset**
- Communications Blockset
- RF Blockset
- Others



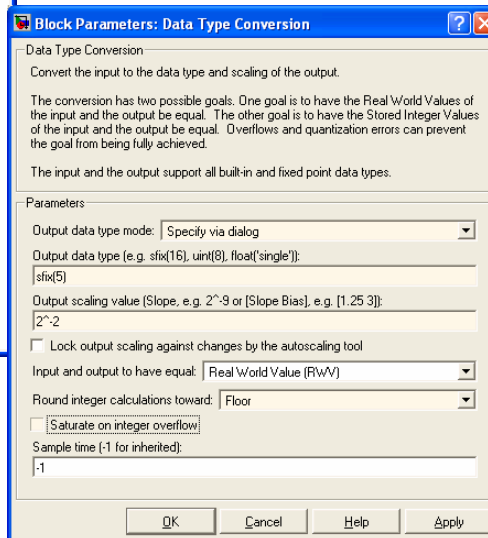
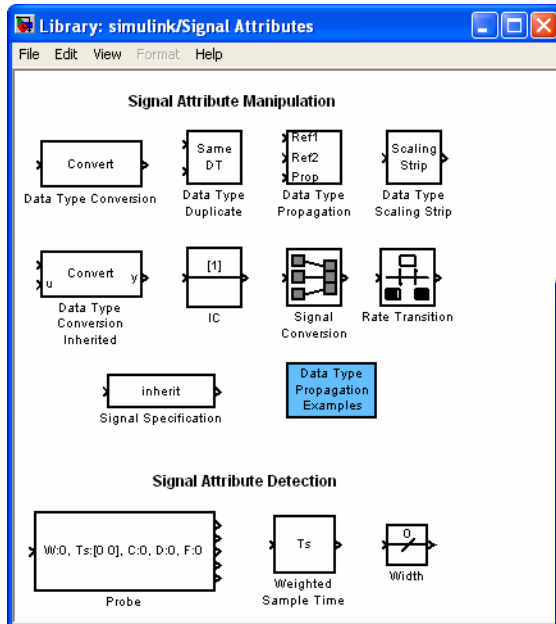
Signal Processing Blockset

- Streaming data
- Multi-rate systems
- Transforms, filters, estimators
- Enables frames in Simulink
- Fixed- and floating-point support





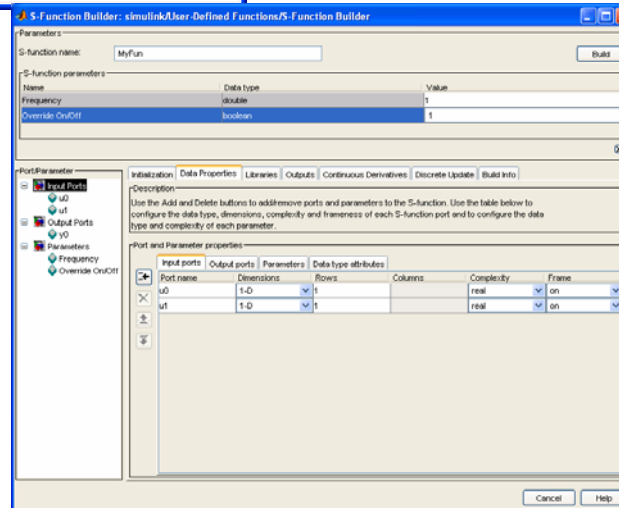
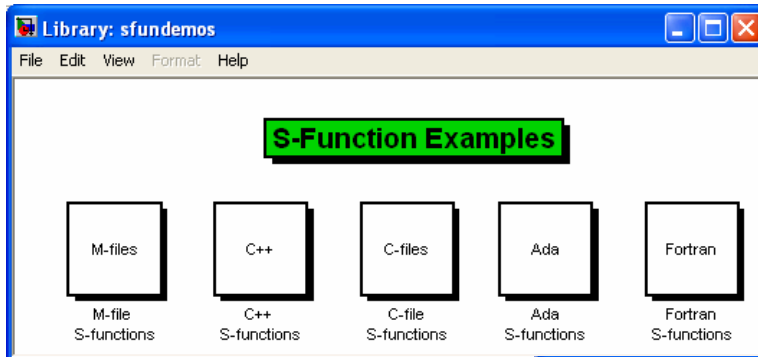
Data Types



- Default double
- C data types in Simulink
- Simulink Fixed Point
 - Specify word length
 - Integer, fixed, fractional, and custom float types
 - Trap overflow and saturation
 - Auto-scaling
 - Round-off options
 - Include own bit-true code



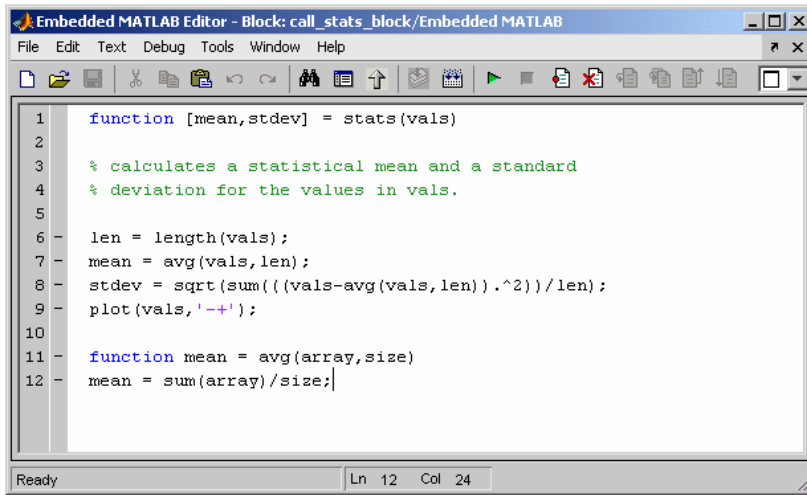
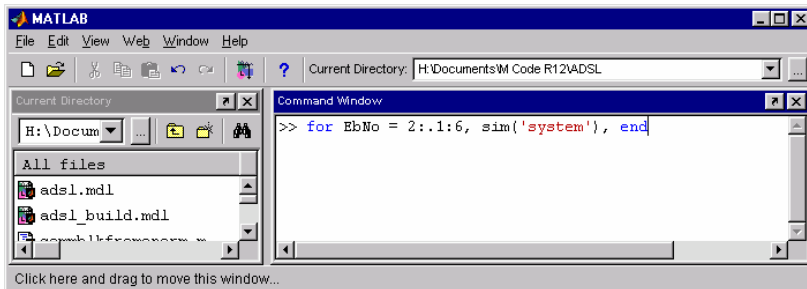
User Defined Blocks and Libraries



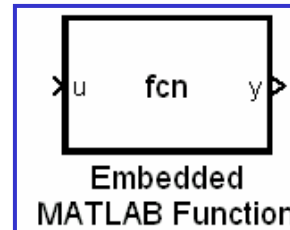
- Created from
 - Other blocks
 - C Code
 - MATLAB Code
- User-defined:
 - Parameter GUIs “masks”
 - Icons
 - Libraries



Co-Develop with MATLAB

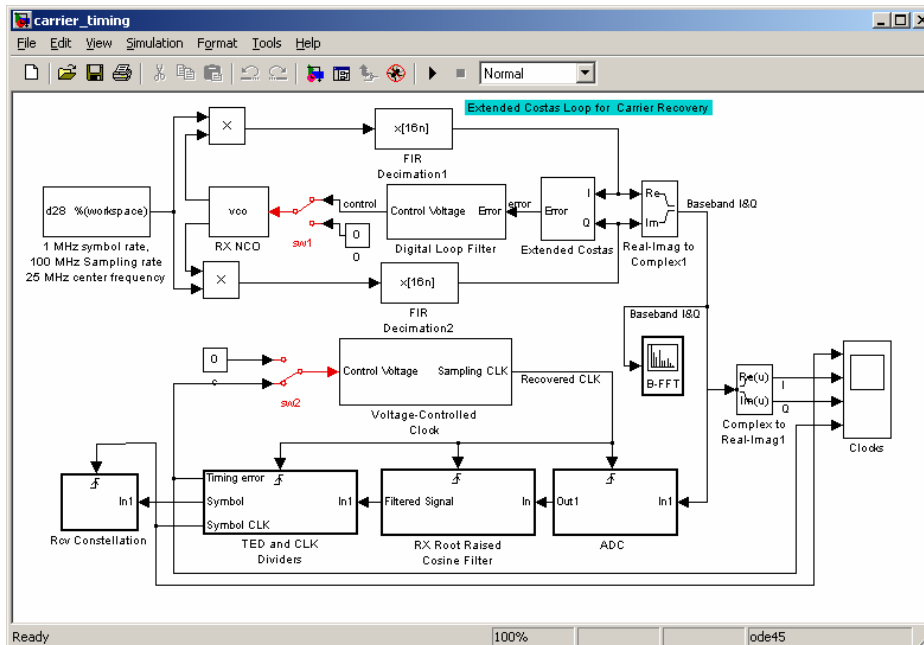


- Change parameters and run Simulink simulations from MATLAB
- **>> for EbNo = 2:1:6, sim('system'), end**
- MATLAB S-functions
- Embedded MATLAB Function
 - **Integration of Embedded MATLAB Functions in Simulink**





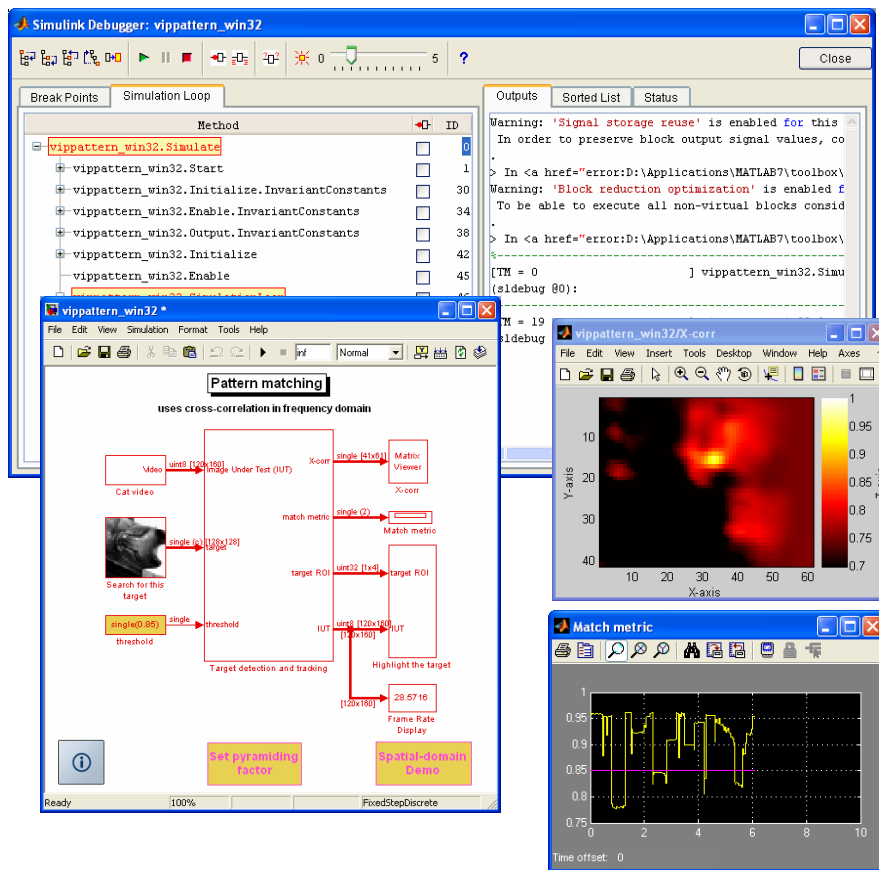
Complex Timing and Concurrency



- Complex timing
 - Feedback
 - Asynchronous edge triggered blocks
 - Multirate digital with arbitrary sample rates
- Concurrency
 - True expression of parallelism
 - Important for whole system or hardware subsystem design
 - Not possible with programming language such as C



Debug, Profile, and Accelerate Models



- Debug
 - Single step blocks and look at inputs, state, and outputs
 - Stop on block or at specific time
- Profile
 - Generate report
 - Show elapsed time on every block
 - Optimize model simulation time
- Accelerate
 - Compile to C Code and run on host



TI TECH DAYS

Design and Implementation of Video Applications



Modeling Video Applications

- Video and Image Processing Blockset
- Provides over 50 components and 100's of algorithms **focused on implementation of embedded systems**



Streaming
video in/out

Detection,
Thresholding

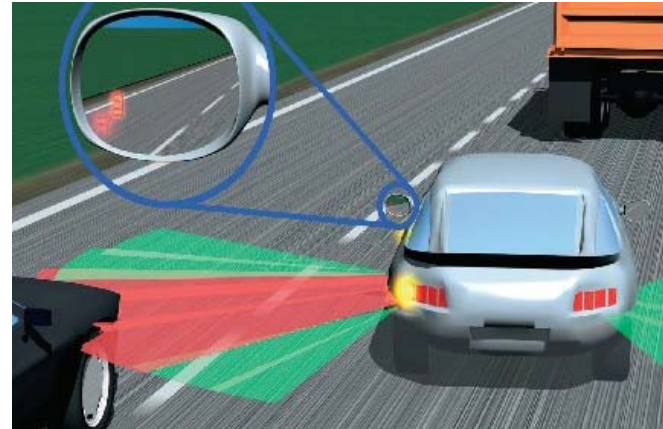
Tracking,
Counting

Background
Estimation



Video and Image Processing Blockset Libraries

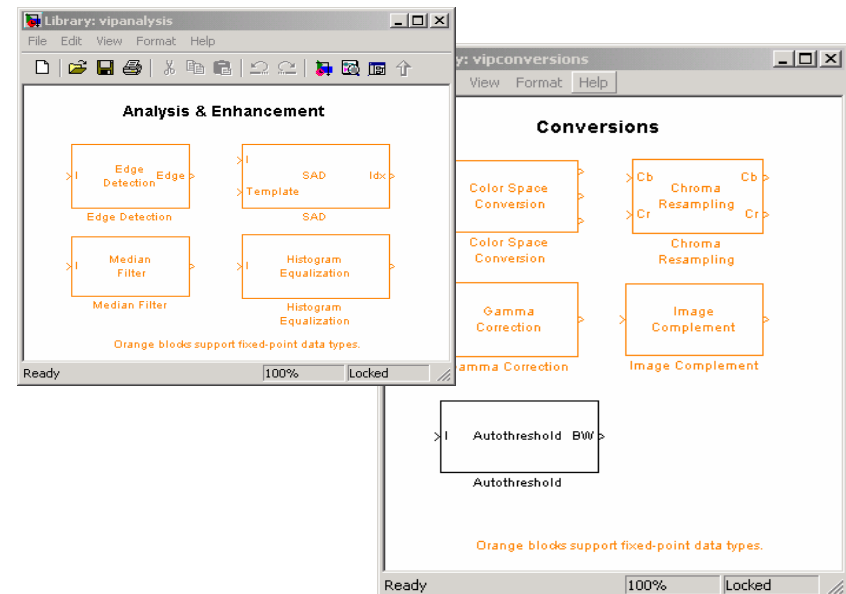
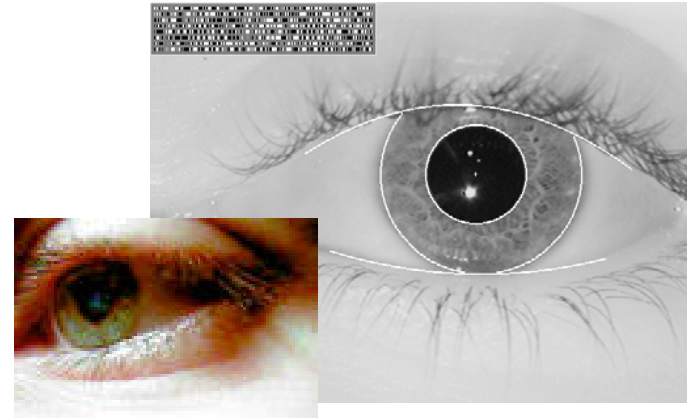
- Basic primitives
 - Padding, correlation, statistics, thresholding
 - Block processing
 - 2-D filtering, 2-D transforms
- Geometric transformations
 - Rotation, translation, resize, shear
 - Interpolation: nearest neighbor, bi-linear, bi-cubic
- Edge detection
 - Sobel, Prewitt, Roberts
- Morphological operations
 - Erode, dilate, open, close
 - Labeling of connected-components





Libraries (continued)

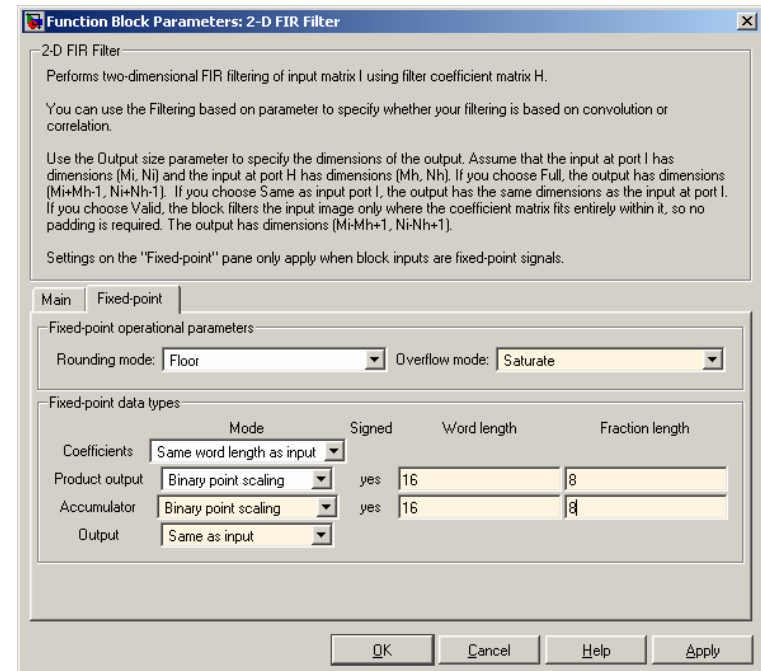
- Analysis and Enhancement
 - Edge detection, median filtering, motion vector estimation (SAD)
- Superimposing images and graphics
 - On-screen text overlays, Picture-in-picture
- Conversions
 - Color-space conversions (RGB, YCbCr, etc)
 - Chrominance re-sampling (4:2:2, 4:2:0, etc)





Simulating Video Applications

- Simulink, Signal Processing Blockset, Video and Image Processing Blockset*
- Fixed-Point considerations
- Avoid inaccurate results due to finite word effects
- Built in tools for scaling and modeling finite word effects
- Easy to change parameters to simulate impact of rounding, overflow, etc.



*Requires Simulink Fixed-Point for integer and fixed point data types



TI TECH DAYS

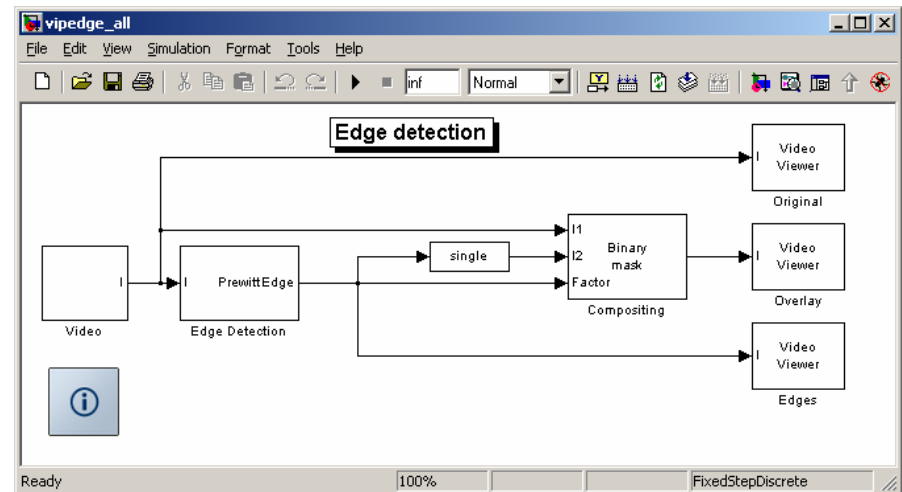
Design an Edge Detection System



Edge Detection

- Fundamental component of many applications
 - Object tracking and recognition
 - Biomedical signal processing
 - Unmanned vehicle technology
 - Segmentation for video compression

Demo



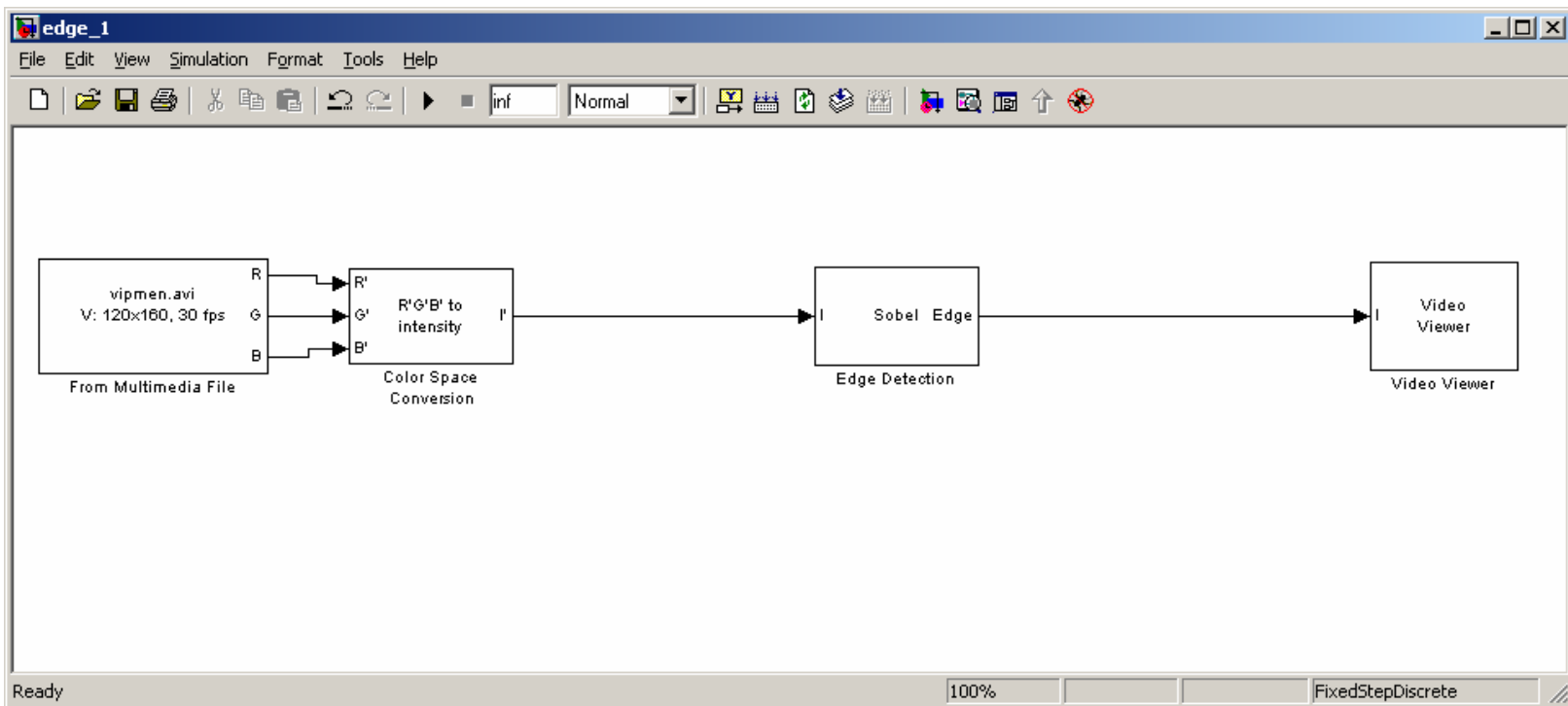


Building Edge Detection Model

- **What you will do...**
 - Find edges
 - Overlay it onto original input
 - Convert the model to fixed-point
- **What you will learn about ...**
 - Video sources and sinks
 - Data type and interpretation of color and intensity
 - Integer processing as a special case of fixed-point
 - Accelerator mode and fixed-point models



Edge Detection and Video Compositing





Configuring Parameters

The screenshot shows a MATLAB/Simulink environment with a Simulink model and a parameter dialog box. The Simulink model contains two blocks: 'Sobel Edge' (labeled 'Edge Detection') and 'Video Viewer'. A yellow arrow points from the 'Sobel Edge' block to the 'Function Block Parameters: Edge Detection' dialog box. The dialog box has the following settings:

- Kernel: Sobel
- Output type: Binary image
- User-defined threshold
- Threshold scale factor: 4
- Edge thinning

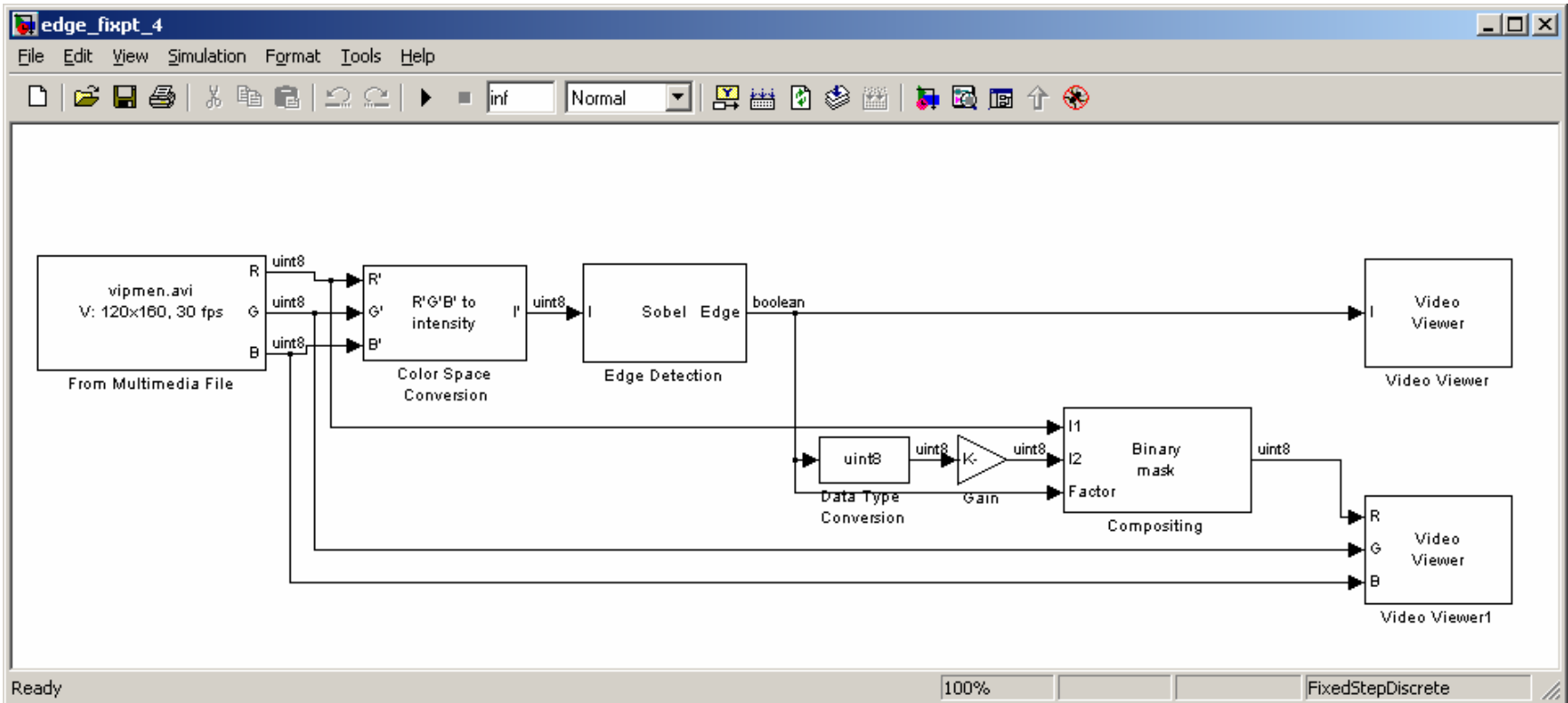
Buttons at the bottom of the dialog include OK, Cancel, Help, and Apply. A status bar at the bottom of the dialog shows 'Unapplied change' and 'Ready'. The Simulink window shows a zoom level of 100% and a 'FixedStepDiscrete' solver.

Choice of threshold computation

Turn on Edge thinning

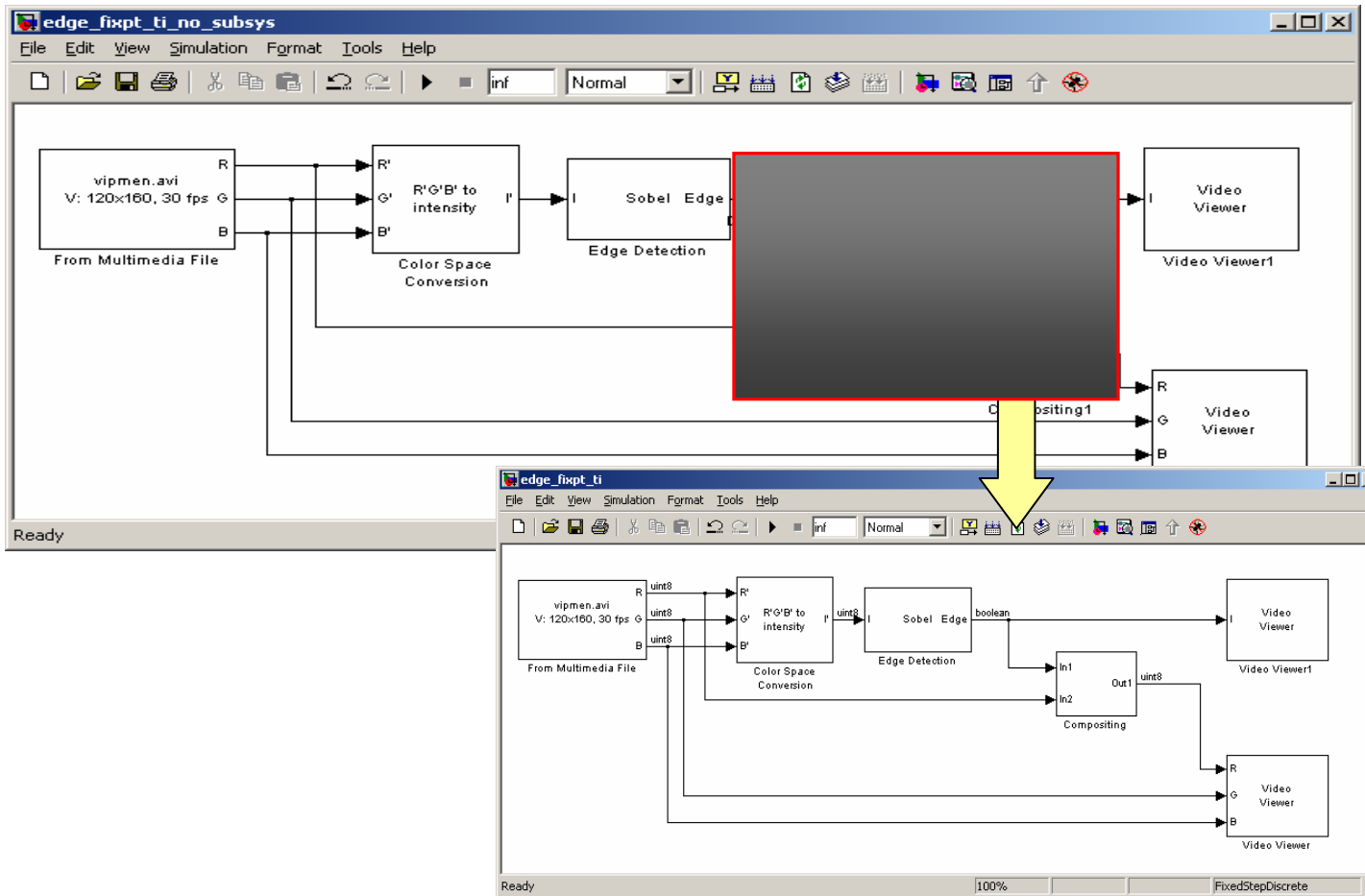


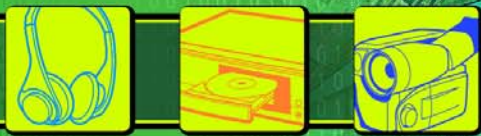
Overlay Original and Detected Edge





Integrating the Final System

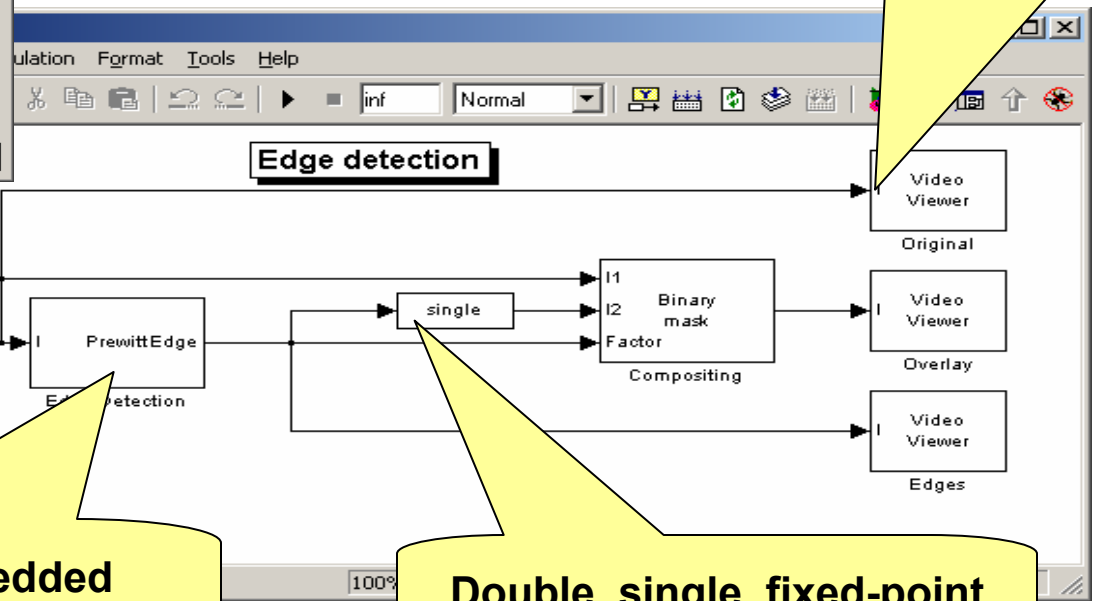
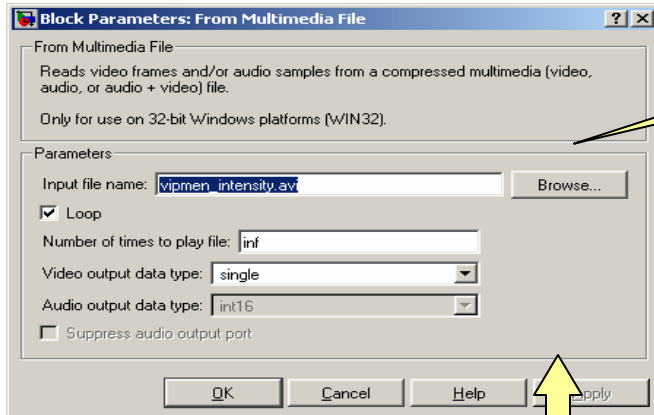




Model Summary

Easy to import streaming video into the simulation

Handy viewers for inspecting video at any point in the algorithm



Options important to embedded system designers

Double, single, fixed-point

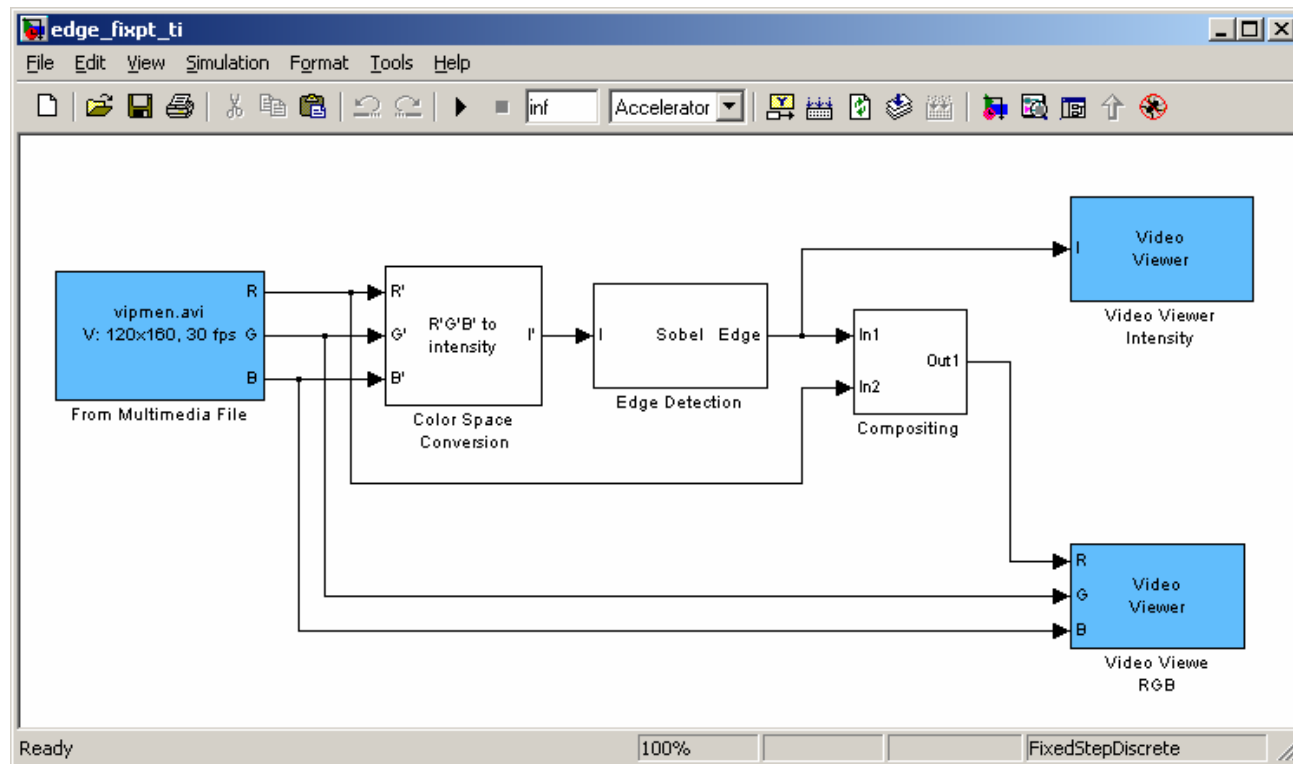


TI TECH DAYS

Implementation of the Edge Detection System on TI DSP



Generating Target-Independent Code



Demo

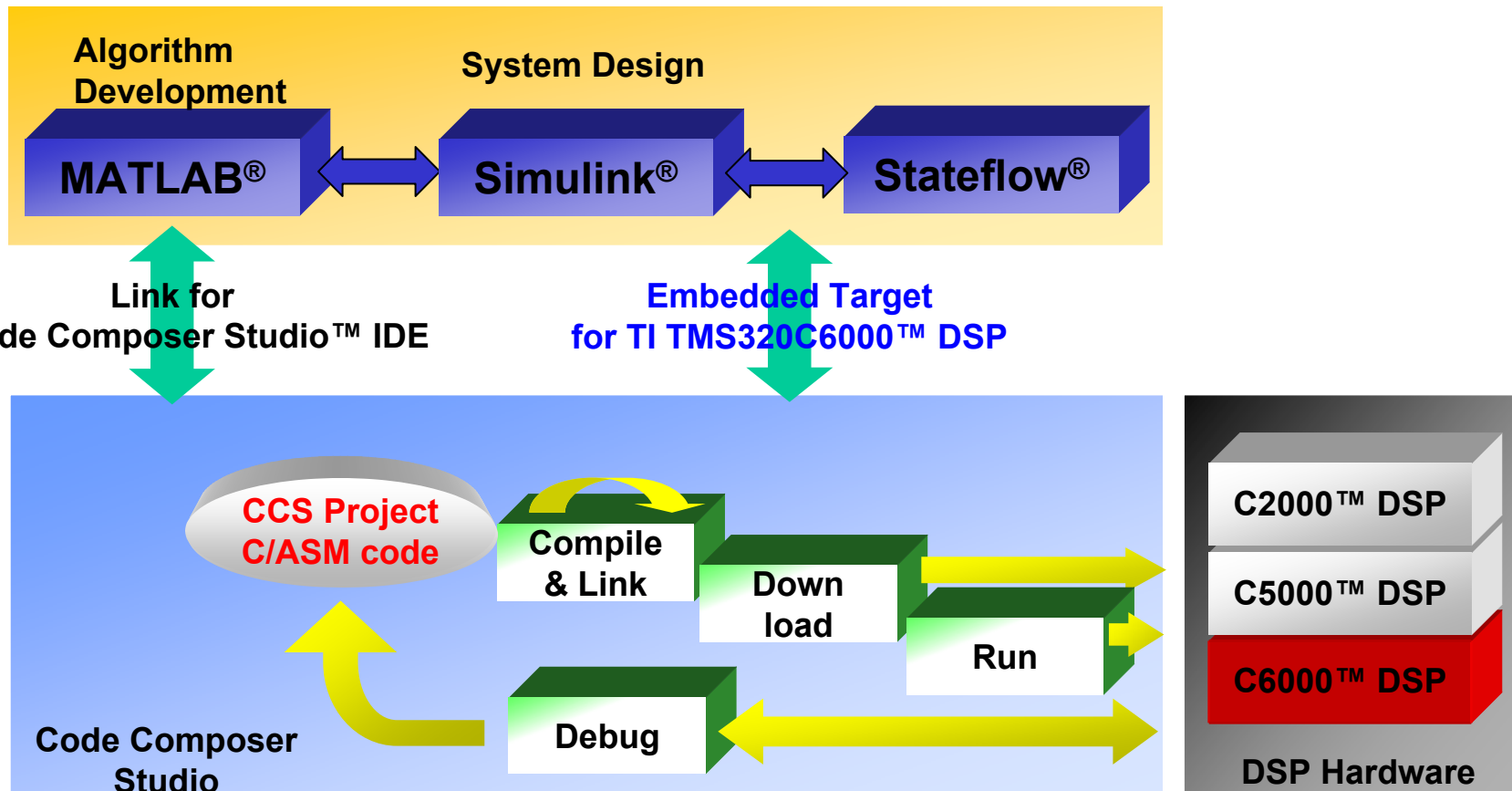


Target-Independent Code (continued)

```
175     order to achieve rounding */
176     acc = coeff[0]*inCC1[i] + coeff[1]*inCC2[i] +
177           coeff[2]*inCC3[i] + 32768;
178
179     acc >>= 16;                               /* scale back */
180     outCC1[i] = (uint8_T)acc;
181 }
182 }
183
184 {
185     /* Video Processing Blockset Edge Detection (svipedge)
186        - '<Root>/Edge Detection' */
187
188     int32_T accOne;
189     int32_T accTwo;
190     int32_T accThree;
191     int32_T accFour;
192
193     int32_T prod;
194
195     boolean_T *outImg = edge_fixpt_ti_B.EdgeDe
196
197     /* gradients for vertical and horizontal ed
198
199     /* offsets pointing to non-zero elements of the gradient kernels */
200     const int32_T *vOffsets = &edge_fixpt_ti_ConstP.EdgeDetection_VO RTP
201     const int32_T *hOffsets = &edge_fixpt_ti_ConstP.EdgeDetection_HO RTP
```

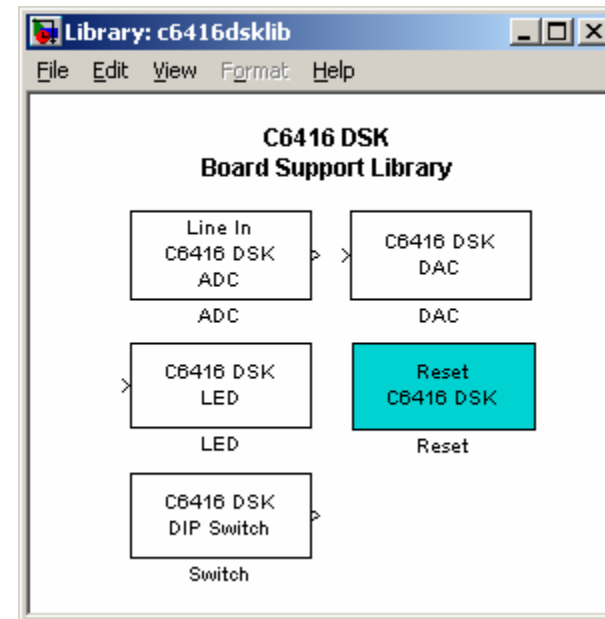
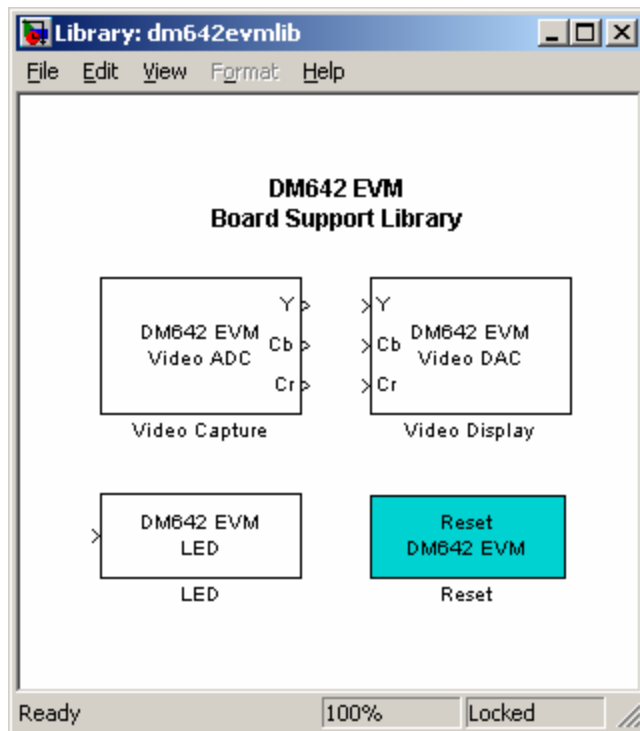


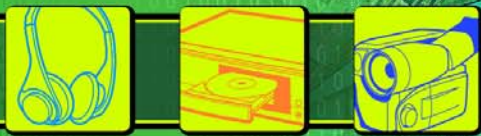

Embedded Target for TI C6000



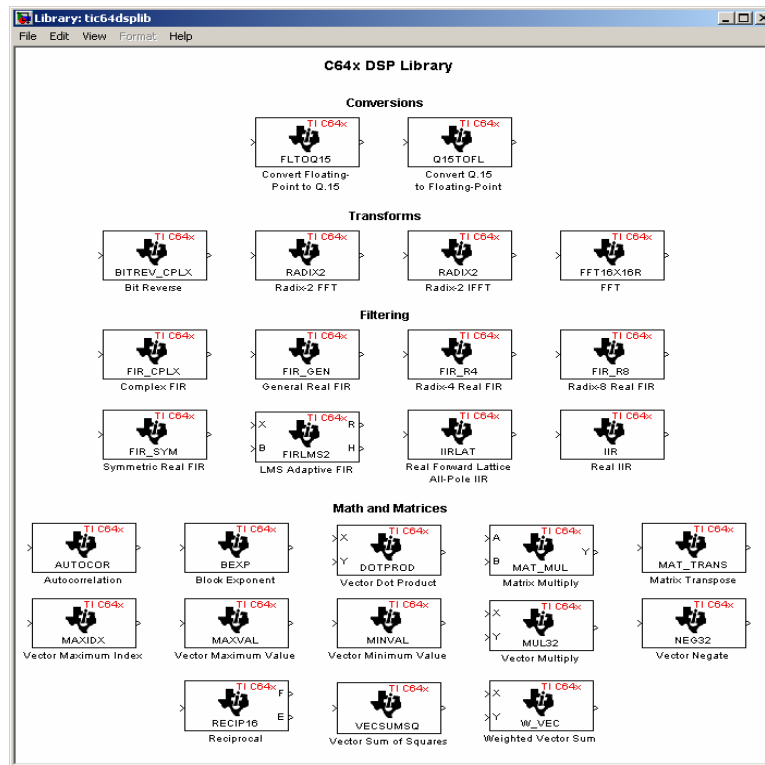


Tour of Device-Driver Libraries



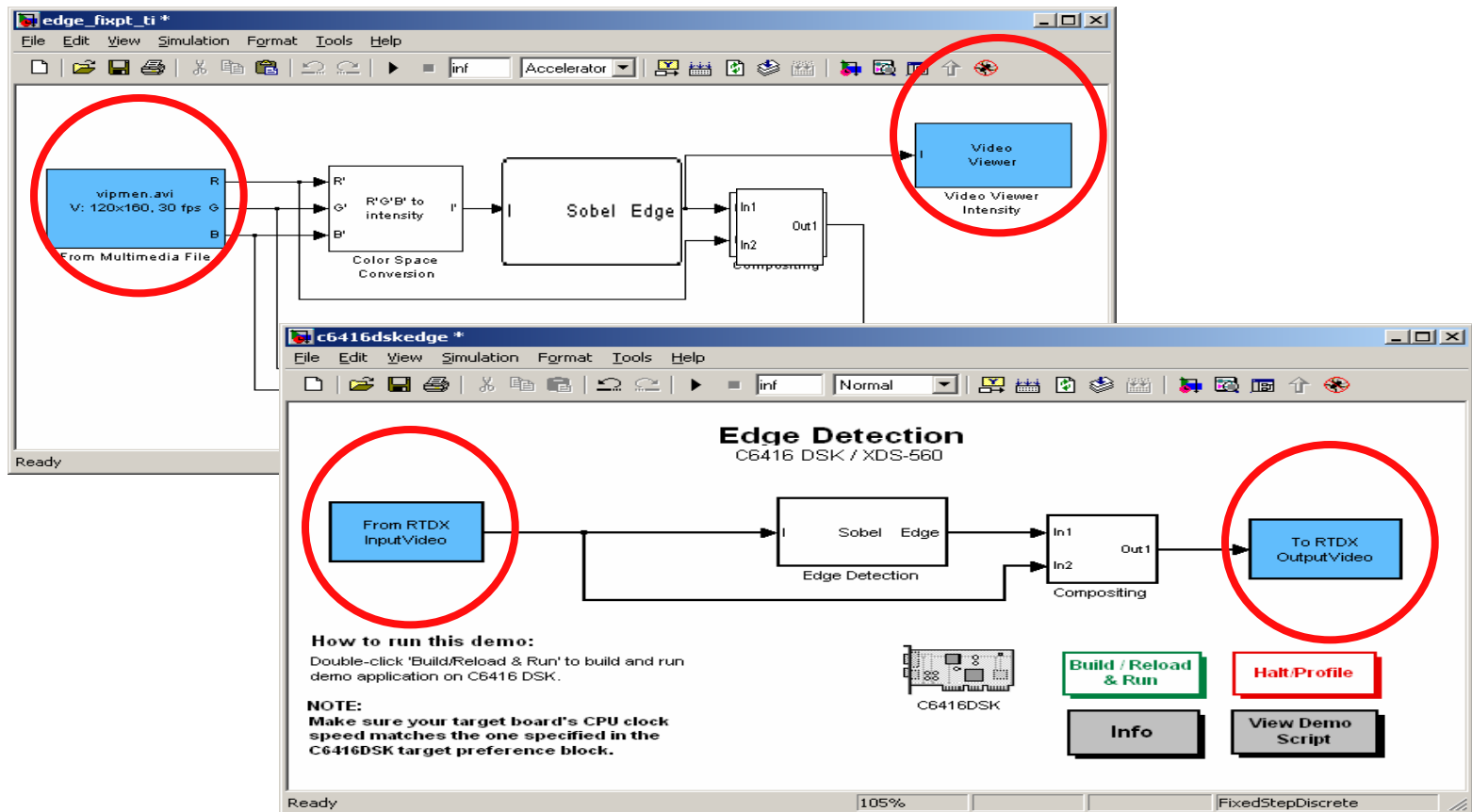


Optimized Block Libraries





Creating Target-Specific Model





Generating Code for Target

The screenshot shows the 'c6416dskedge' application window. The main window title is 'Edge Detection' and the subtitle is 'C6416 DSK / XDS-560'. The menu bar includes File, Edit, View, Simulation, Format, Tools, and Help. The toolbar contains various icons for file operations and simulation. The main workspace is currently empty.

Overlaid on the main window is the 'Configuration Parameters: c6416dskedge/Configuration' dialog box. The 'Select:' tree on the left has 'Real-Time Workshop' selected. The right pane of the dialog box contains the following sections:

- Target selection:** RTW system target file: `ti_c6000_ert.tlc` (with a 'Browse...' button). Description: Current system target file. Use Browse button at right to select a different target.
- Documentation:** Generate HTML report, Include hyperlinks to model, Launch report after code generation completes.
- Build process:** TLC options: [empty field], Make command: `make_rtw`, Template makefile: `ti_c6000_ert.tmf`.
- Custom storage class:** Ignore custom storage classes.
- Generate code only (with a 'Generate code' button).

At the bottom of the dialog box are 'OK', 'Cancel', 'Help', and 'Apply' buttons.

On the left side of the main window, there is a blue box with the text 'From RTDX InputVideo'. Below it, the text reads: 'How to run this demo: Double-click 'Build/Reload & Run' demo application on C6416 DSK. NOTE: Make sure your target board speed matches the one specified in the C6416DSK target preference'. The status bar at the bottom left of the main window shows 'Ready'.



Analyzing Generated Code

```
/* Model step function */
void c6416dskedge_step(void)
{
    STS_set(&stsSys0_OutputUpdate, CLK_gettime());

    /* S-Function Block: <Root>/From RTDX (rtdx_src) */
    if (!RTDX_channelBusy(&InputVideo)) {
        RTDX_readNB( &InputVideo, (void*) c6416dskedge_B.FromRTDX
            19200*sizeof(uint8_T));
    }

    /* Video Processing Blockset Edge Detection (svipege
    - '<Root>/Edge Detection' */

    int32_T accOne;
    int32_T accTwo;
    int32_T accThree;
    int32_T accFour;

    int32_T prod;

    boolean_T *outImg = c6416dskedge_B.EdgeDetection;

    /* gradients for vertical and horizontal edge responses */
    /* offsets pointing to non-zero elements of the gradient kernels */
```

Build Complete,
0 Errors, 0 Warnings, 0 Remarks.

Build /

HALTED For Help, press F1 Ln 115, Col 21

Code generated
for edge detection
block



Taking a Closer Look

Generated DSP/BIOS configuration file

Model base rate is tied to Timer 1 INT

ISR is assigned to HW_INT15 (Timer 1)

The screenshot shows the Code Composer Studio interface for a Blackhawk CPU. The File View on the left shows the project structure, including the generated DSP/BIOS configuration file 'c6416dsksedge.cdb'. The System tree in the center shows the hardware configuration, with HW_INT15 selected. The Properties window on the right shows the configuration for HW_INT15, including the interrupt source and function.

Property	Value
comment	defines the INT15 Inter...
interrupt source	Timer_1
Function	_Timer1_ISR
monitor	Nothing
addr	0x00000000
type	signed
operation	STS_add(*addr)
Use Dispatcher	True
Arg	0x00000001
Interrupt Mask	none
Interrupt Bit Mask	0x0000
Don't modify cache control	True
Program Cache Control ...	cache enable
Data Cache Control Mask	cache enable



Verifying Target Code

RTDX block generates target code for reading data from host

RTDX block generates target code for writing data to host

on Format Tools Help

inf Normal

Edge Detection

C6416 DSK / XDS-560

```

graph LR
    A[From RTDX InputVideo] --> B[Sobel Edge  
Edge Detection]
    B --> C[Compositing  
In1 In2 Out1]
    C --> D[To RTDX OutputVideo]
  
```

How to run this demo:
Double-click 'Build/Reload & Run' to build and run demo application on C6416 DSK.

NOTE:
Make sure your target board's CPU clock speed matches the one specified in the C6416DSK target preference block.

C6416DSK

Build / Reload & Run

Halt/Profile

Info

View Demo Script

Ready 105% FixedStepDiscrete



Running Target Code

Click to run previously built DSP application

Click here to halt demo and display profile report

How to run this demo:
Double-click 'Build/Reload & Run' to build and run demo application on C6416 DSK.

NOTE:
Make sure your target board's CPU clock speed matches the one specified in the C6416DSK target preference block.

Build / Reload & Run
Halt / Profile
Info
View Demo Script

C6416DSK

From RTDX InputVideo
Sobel Edge
Edge Detection
In1 In2 Out1
Compositing
To RTDX OutputVideo

Ready 105% FixedStepDiscrete



Profiling Real-time Execution

The screenshot shows a web browser window titled "Profile Report" with a menu bar (File, Edit, View, Go, Debug, Desktop, Window, Help) and a location bar. The main content area displays the following information:

Profile Report
Simulink model: [c6416dskedge.mdl](#)
Target: C6416DSK
Report of profile data from Code Composer Studio (tm)
07-Jan-2005 10:59:06

Timing constants

Base sample time	66.67 ms
CPU Clock speed ¹	720 MHz

Profiled Simulink Subsystem

System name	c6416dskedge
STS object	stsSys0_OutputUpdate
Max time spent in this subsystem per interrupt	6.27 ms
Max percent of base interval	9.41%
Number of iterations counted	203

STS Objects
Raw profile data reported by Code Composer Studio (tm)



Getting Further Insight

The screenshot displays the MATLAB/Simulink environment. The main window is titled "c6416dskedge *". A "Function Block Parameters: EdgeDetection" dialog box is open, showing various settings. A yellow callout bubble points to the "Treat as atomic unit" checkbox, which is checked. The background shows a Simulink model with blocks for "Sobel Edge", "Edge Detection", "Compositing", and "To RTDX OutputVideo". A "C6416DSK" hardware icon is also visible. At the bottom, there are buttons for "Build / Reload & Run", "Halt/Profile", "Info", and "View Demo Script".

Function Block Parameters: EdgeDetection

- Subsystem: Select the settings for the subsystem block.
- Parameters:
 - Show port labels
 - Read/Write permissions: ReadWrite
 - Name of error callback function:
 - Permit hierarchical resolution: All
 - Treat as atomic unit
 - Minimize algebraic loop occurrence
 - Sample time (-1 for inherited): -1
 - RTW system code: Auto
 - RTW function name options: Auto
 - RTW file name options: Auto

Make the subsystem atomic



Examining Profile Report

Execution statistics for overall model

Link to the profiled subsystem

The screenshot shows a web browser displaying a profile report. The main content is titled "Profiled Simulink Subsystems" and contains three tables of execution statistics. Each table lists the system name, STS object, maximum time spent per interrupt, maximum percent of base interval, and the number of iterations counted. The first table is for the overall model, and the subsequent two are for atomic subsystems. Below these tables is a section for "STS Objects" with a table of raw profile data.

System name	c6416dskedge
STS object	stsSys2_OutputUpdate
Max time spent in this subsystem per interrupt	6.254 ms
Max percent of base interval	9.38%
Number of iterations counted	466

System name	c6416dskedge/EdgeDetection
STS object	stsSys1_OutputUpdate
Max time spent in this subsystem per interrupt	5.22 ms
Max percent of base interval	7.83%
Number of iterations counted	466

System name	c6416dskedge/Compositing
STS object	stsSys0_OutputUpdate
Max time spent in this subsystem per interrupt	877.1 μs
Max percent of base interval	1.32%
Number of iterations counted	466

STS Objects
Raw profile data reported by Code Composer Studio (tm)

STS Object Name	count (measurements)	total	max	average
IDL_busyObj	51778899	1496341355	89916	28.8987

Execution statistics for atomic subsystems



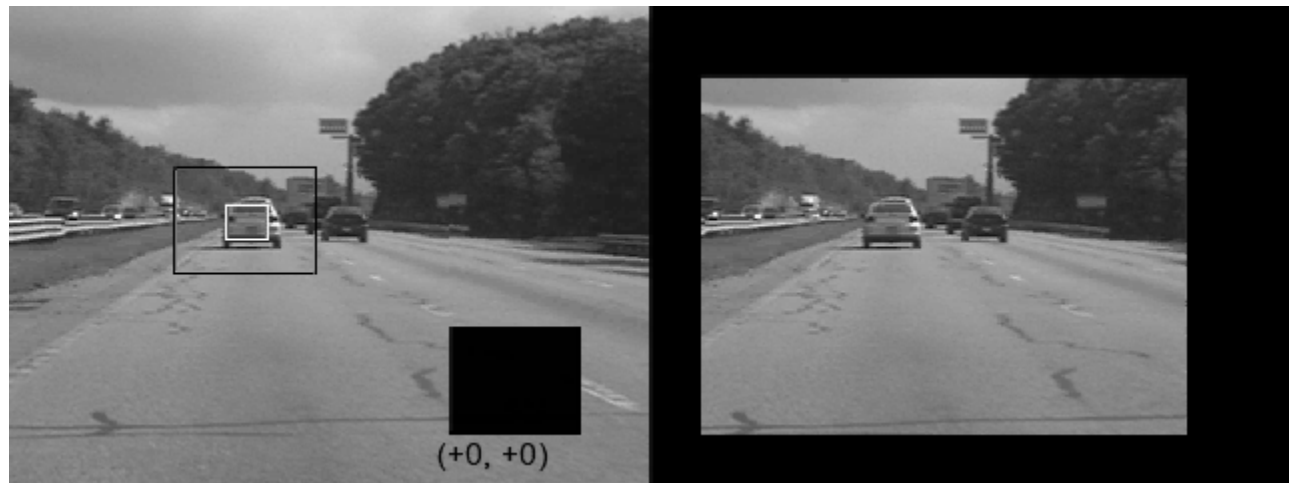
TI TECH DAYS

Advanced Video Applications



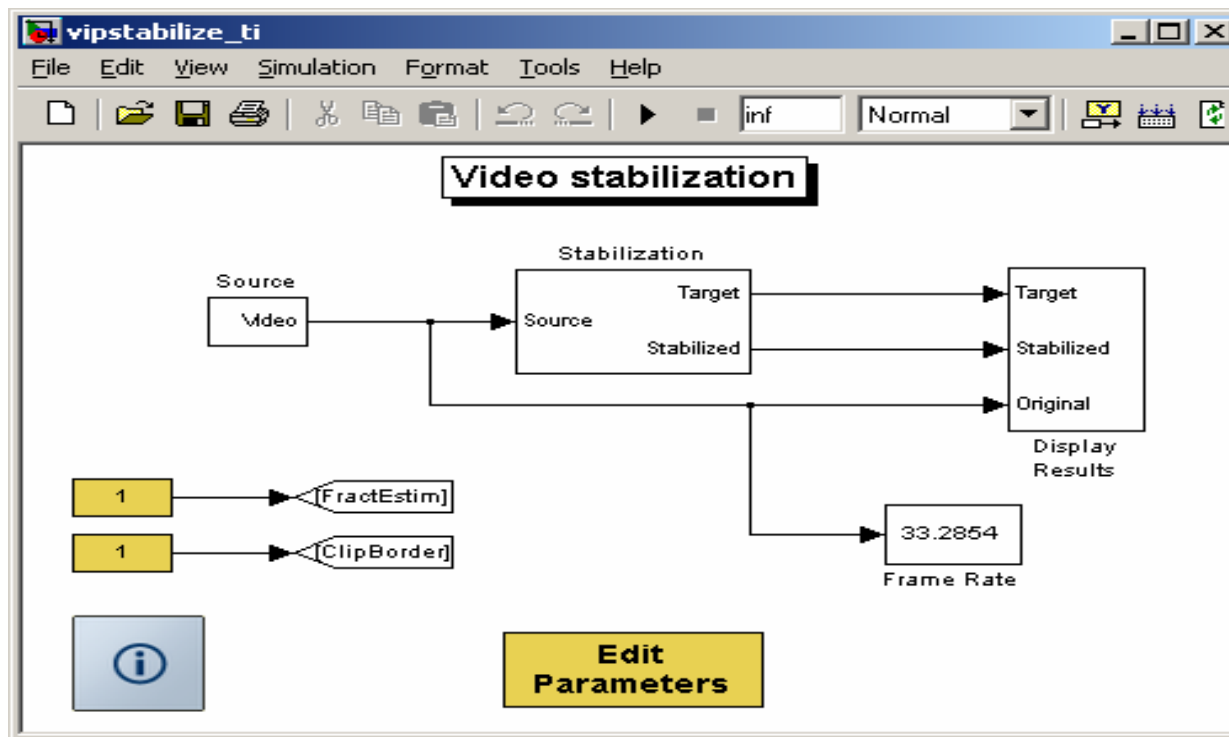
Video Stabilization

Track and remove
motion in a video
sequence





Video Stabilization (continued)

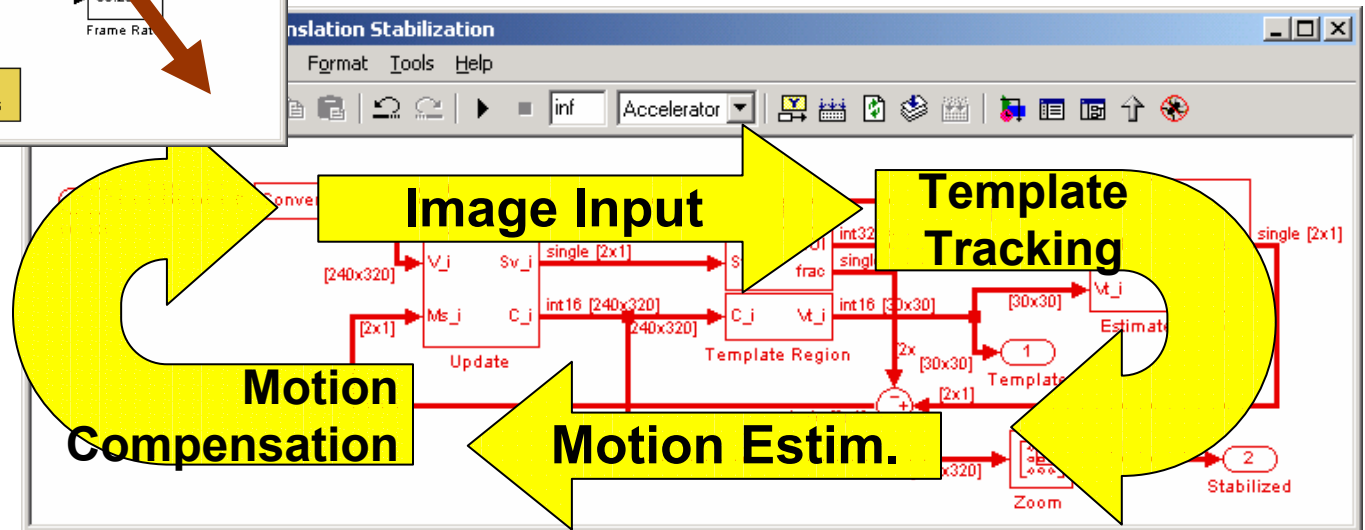
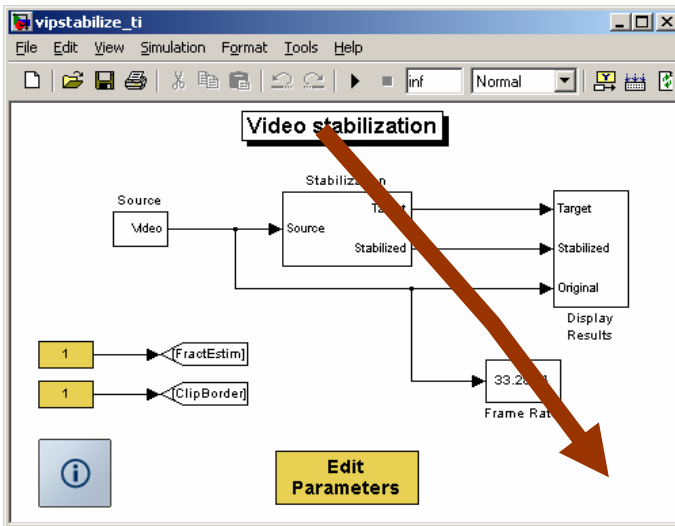


Demo



TI TECH DAYS

Model Overview





Algorithm Overview

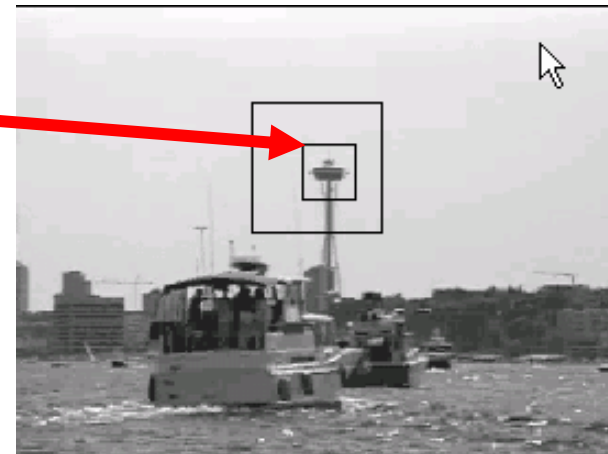
- **Steps to Stabilize Motion**

- Estimate target position from template
- Compute inter-frame motion
- Compensate motion
- Update matching template

Frame (n-1):
Origin=(100,100)



Frame n:
Origin=(80,80)



$\Delta x, \Delta y$



Estimate Target Position

(Computationally expensive)

$$E = \min_S \sum_T |V(x, y) - T(x - x_0, y - y_0)|$$

Video Image, V



Motion estimate
 (x_0, y_0)

Computational Cost
= $2 * N^2$ (over T : $N \times N$)
* L^2 (over S : $L \times L$)
* fps

Ex:

32x32 template
64x64 search
30 frames/sec
>250 million adds
per second





Search for Target Position

Sum of Absolute Differences (SAD)

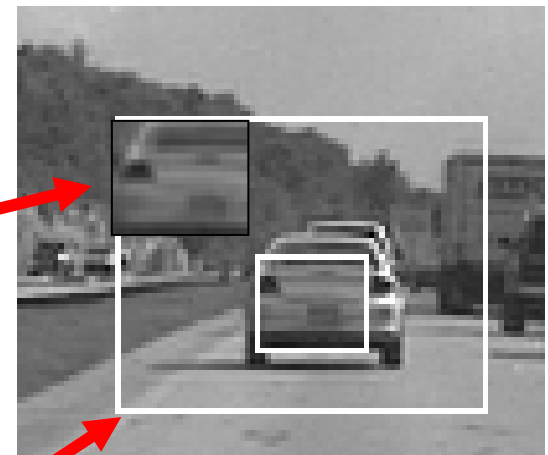
$$E = \min_S \sum_T |V(x, y) - T(x - x_o, y - y_o)|$$

> I	Idx >
> Template SAD	NVals >
> ROI	NValid >
SAD	

Video Image, V



Template Image, T



Search for best fit (minimum SAD)

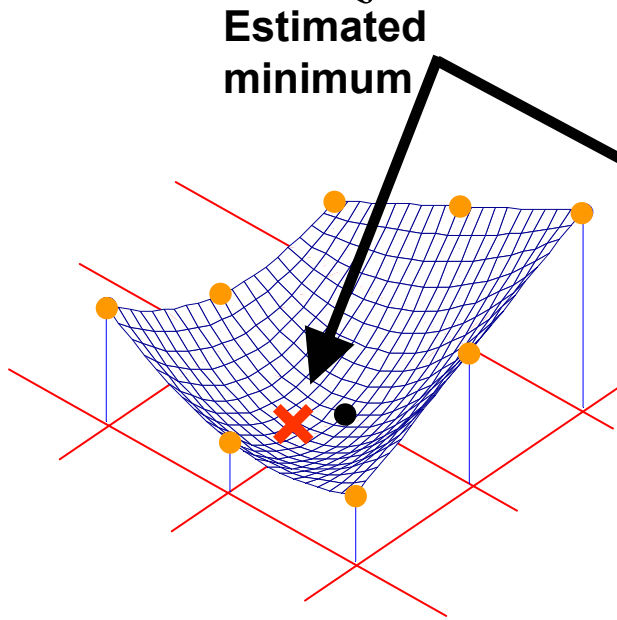


Sub-Pixel Estimation of Target Motion

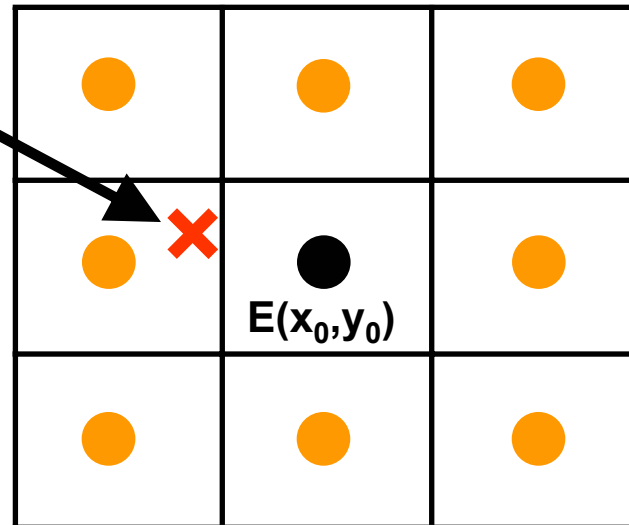
Refine coarse motion estimates

- Find minimum of a quadratic surface over 3x3 neighborhood

$$z = a + bx + cy + dxy + ex^2 - fy^2$$

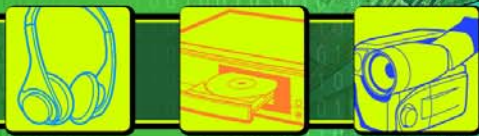


Estimated
minimum

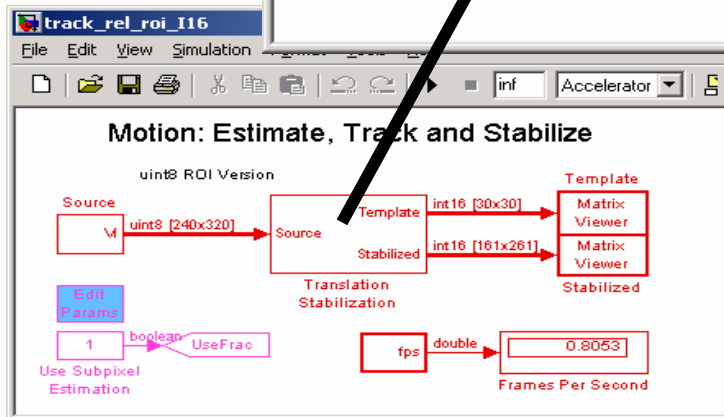
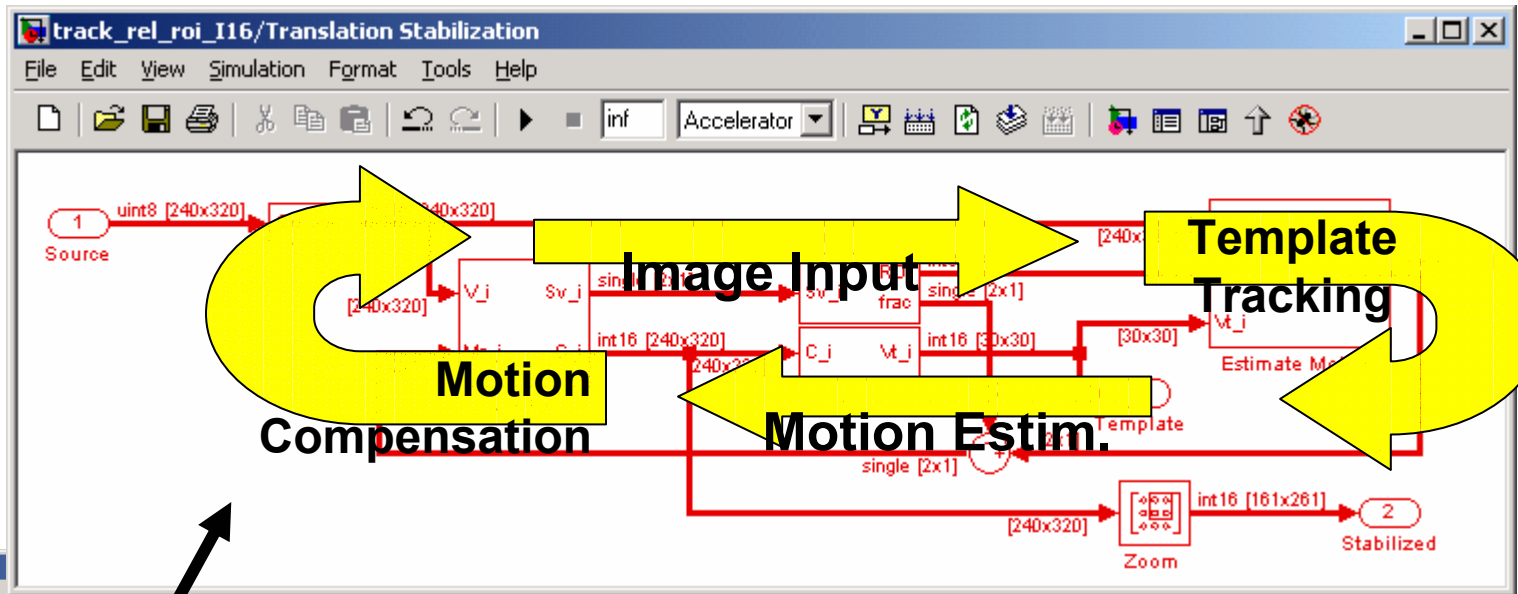


→ Solve $Ax=B$
for minimum

→ Sub-pixel
estimate



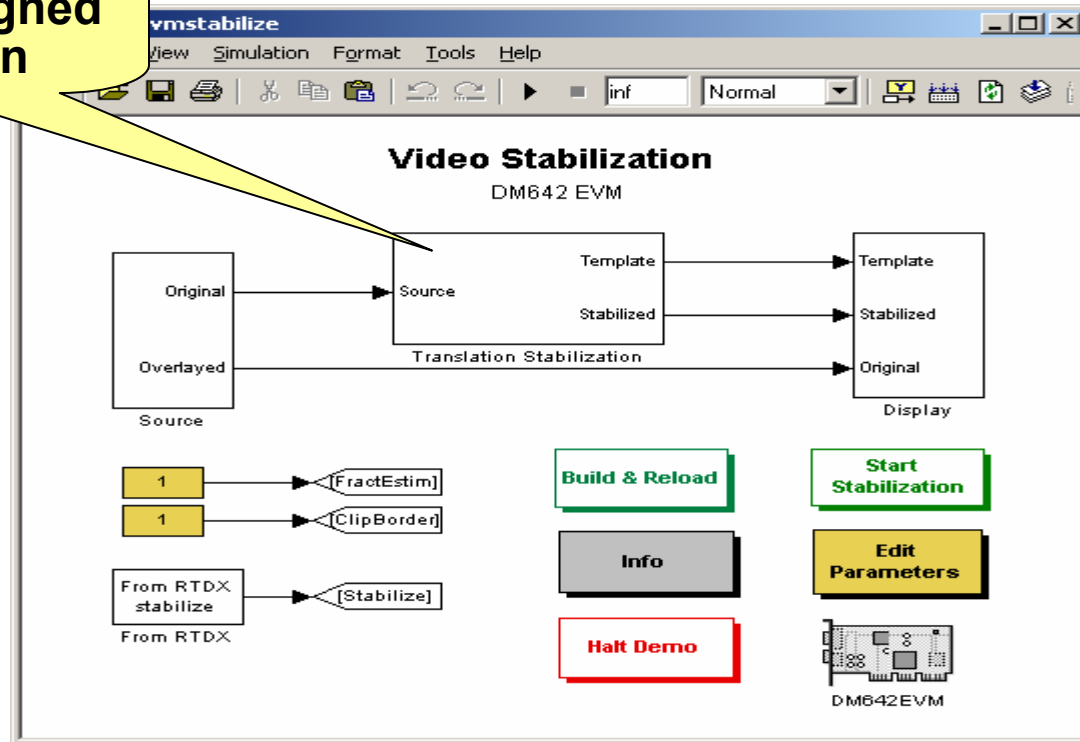
Integrating Video Stabilization System





Video Stabilization on TI DM642 EVM

Stabilization algorithm designed in simulation





TI TECH DAYS

Next Steps



More Information

- Products
 - www.mathworks.com/dsp
- Example models available for download on MATLAB Central
 - www.mathworks.com/matlabcentral/
 - Click on “File Exchange”
- Upcoming seminars, Webinars, and more...
 - www.mathworks.com/dsp_events
- View a recorded Webinar (more than 50 available)
 - www.mathworks.com/webinar



Next Steps

- **Arrange for a live online demo and discussion**
- **Arrange for an onsite visit by MathWorks Applications Engineer**
- **Request an evaluation license and try it out**
- **Attend a MathWorks training course**
- **Contact Rob Segal, your account manager:**
 - 508-647-7615
 - Robert.Segal@mathworks.com
- **Thank you for your interest!**