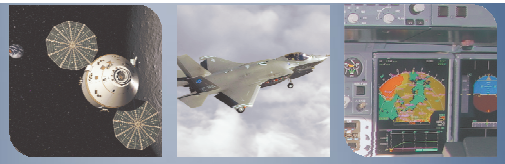
The MathWorks™ MATLAB®&SIMULINK®

Algorithm Design and Code Generation with Embedded MATLAB™

Richard Anderson
Consultant Production Code Generation

richard.anderson@mathworks.co.uk



MathWorks
Aerospace and Defence Conference '08

Agenda

- Contribution of MATLAB functionality to a Model-Based Design Workflow
- What is the Embedded MATLAB functionality?
- Demonstration
  - Converting a simple Median Filter to be C code generation compatible.
- Conclusion

But First a Quick Demonstration

Model-Based Design Workflow

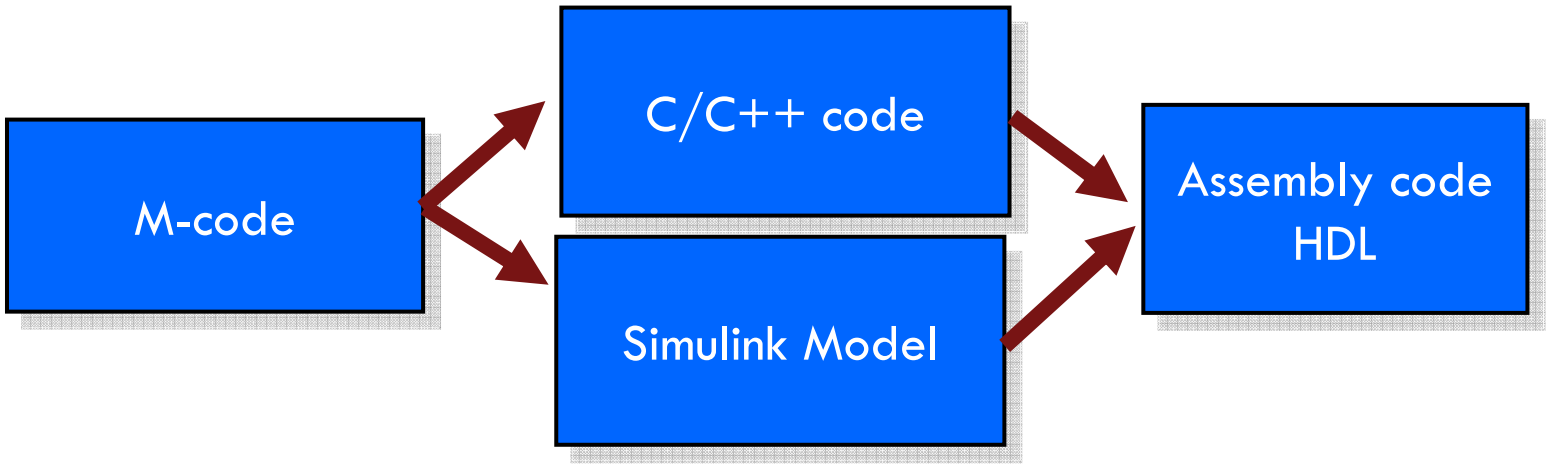
- Model algorithm using a high level modelling language
- Execute system to test response and prove design
- Generate code from block diagram
  - Either automatically
  - Or by hand
- Test target implementation
- If automatic code generation is used there is a single source which can be referenced from each stage

Where does the algorithm originate?

- Sometimes start with a blank sheet and develop it block by block
  - This fits traditional Model-Based Design
- Often originates from a prototype implementation written in the m language.
  - How do you incorporate this?
  - How do you maintain traceability to original?
  - How can you reduce need to re-write algorithm?

Traditional Concept to Implementation:

Re-implement as you go down the level of abstraction



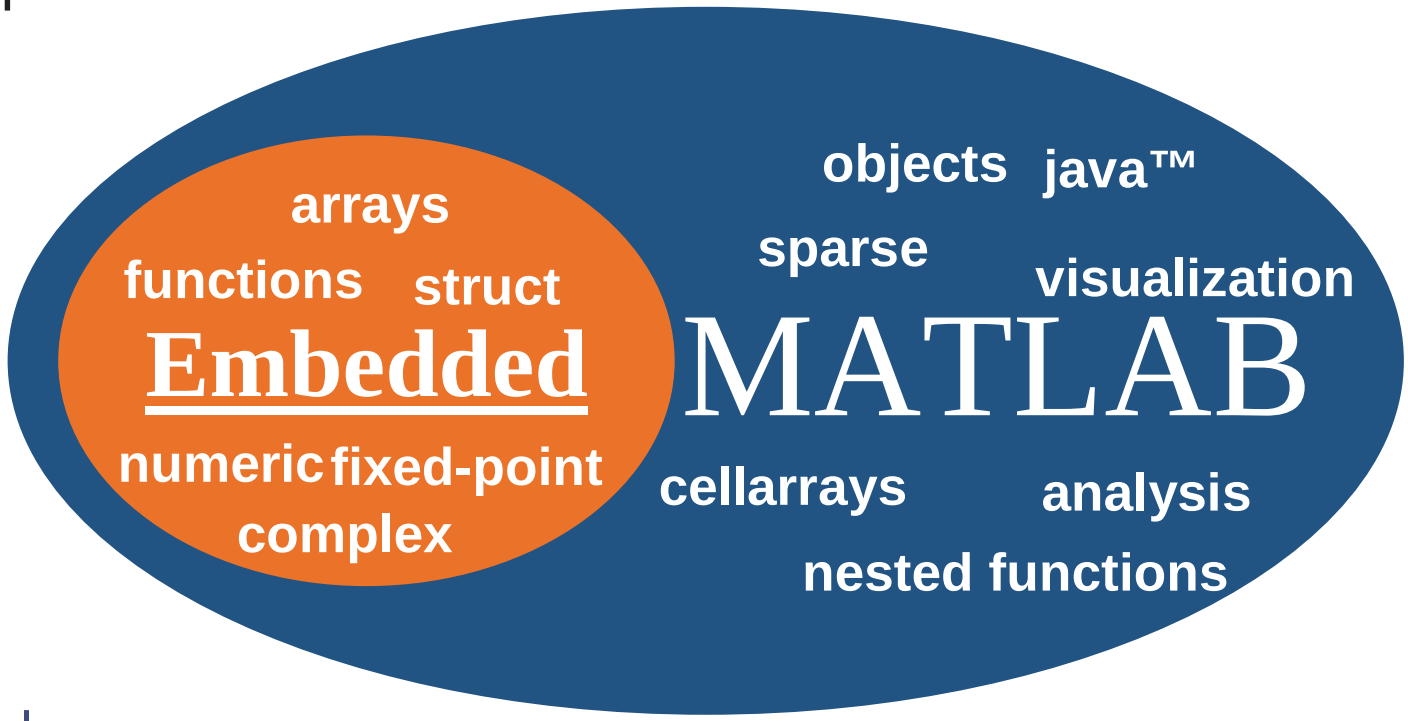
- Time-consuming, error-prone
- Maintain multiple copies of the same algorithm
- Need to fix and test across implementation boundaries

Better Concept to Implementation

- One language
  - No multiple copies of source code
  - Elaborate M-code with real-world design constraints
  - Familiar development environment
  - Reuse of original test cases
  - Integrated visualization, analysis & debugging
- Easy integration into Simulink environment
- Automatic code generation
  - Path to embedded software (M to C)

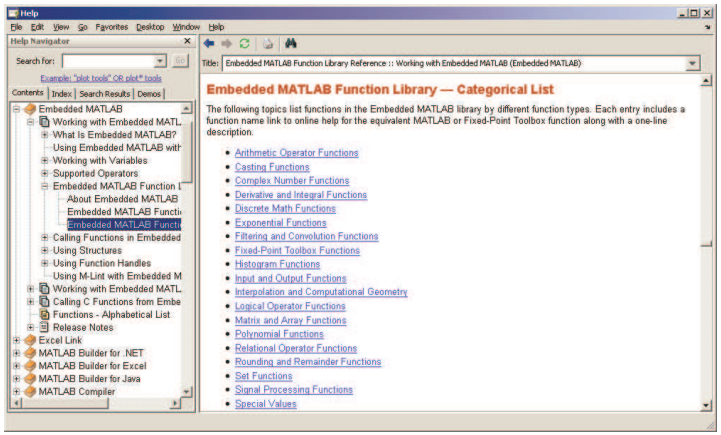
What is the Embedded MATLAB functionality?

- **Embedded MATLAB™** is the subset of the MATLAB® language that can be compiled into code for embedded implementation.



Features of Embedded MATLAB

- Extensive function library
  - 270 MATLAB operators and functions
  - 90 Fixed-Point Toolbox™ functions
- Can automatically generate C code
  - EMLMEX – Generate C-Mex functions from M-code
  - EMLC – Generate C-Code directly from M-code



- Reference M-files on the MATLAB path
  - Enables re-use of M-code in System Models
  - Enables partitioning of large Embedded MATLAB programs
- Call custom C code directly with eml.ceval
- Integration with Simulink and Stateflow®
  - Supports Frames, Structures, Data Type Override

What you will learn in this class

- Make your MATLAB code compliant with the Embedded MATLAB subset
- Generate prototype C code automatically from the MATLAB Desktop
- Integrate your Embedded MATLAB compliant code into Simulink models
- Elaborate code to develop towards a production solution
- Some tips for using Embedded MATLAB

Demonstration

Algorithm design with Embedded MATLAB

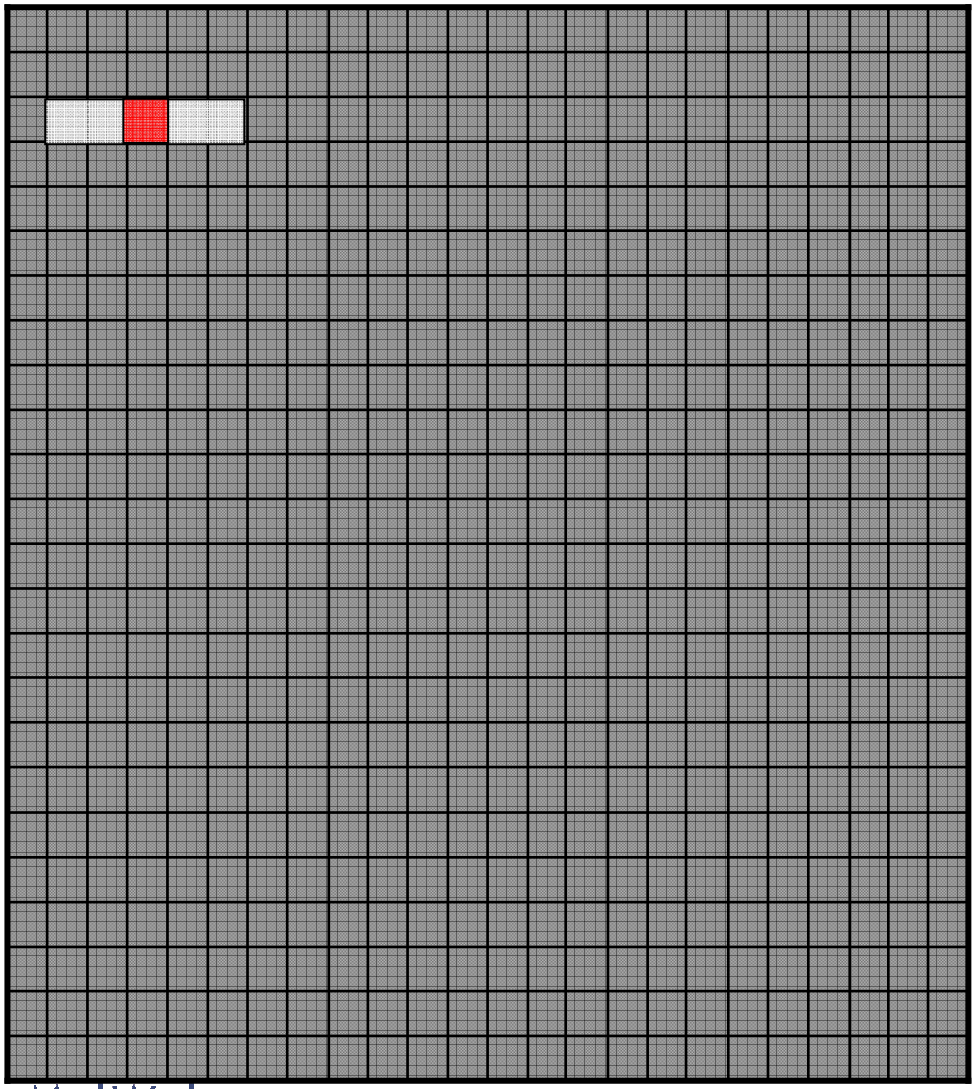
Noise removal and video enhancement with Median Filtering

Algorithm compliance with the Embedded MATLAB subset

Typical steps to make MATLAB algorithms compliant

1. Remove instances of dynamic memory allocation
2. Lock down data types and sizes of variables
   - Assignment in function body
   - Assignment at compile time
3. Replace unsupported toolbox function calls with Embedded MATLAB supported calls
4. Optimise algorithm for embedded implementation

Median Filter



- Calculate Block Statistics
- Replace Center Pixel or Keep Center Pixel
- Repeat for each Pixel

Median Filter

- For each pixel in the image calculate the median value in a surrounding neighborhood of the pixel.
- Replace the starting pixel with the median value, if it's value looks skewed
- Can affect all image pixels
- Only performs well on low noise densities
- Simple to implement
- Computationally intensive

Summary (1)

- Embedded MATLAB is the subset of MATLAB language for embedded implementation
  - Supports 270 core MATLAB and 90 Fixed-Point Toolbox operators and functions
- Narrows concept-to-implementation design gap
  - Reduces algorithm translation, maintaining traceability
- Embedded MATLAB Function block
  - Seamless integration of Embedded MATLAB compliant algorithm into Simulink
  - Simulate the effect of your algorithm in the context of the whole system
  - Automatically generate C code using Real-Time Workshop

Summary (2)

- Embedded MATLAB in Simulink models provides tools for better array manipulation
- EMLC command
  - Automatically generate C source code from your compliant MATLAB code directly from MATLAB desktop
- EMLMEX command
  - Automatically generate C-Mex functions from compliant m-code for algorithm acceleration

Products That Support Embedded MATLAB

Product	Functionality
Fixed-Point Toolbox	Fixed-point functions and operators emlmex command, which compiles M-files for accelerated execution
Simulink®	Embedded MATLAB Function block, which executes code in one or more M-files
Stateflow®	Embedded MATLAB functions that execute code in one or more M-files
SimEvents®	Attribute block that executes Embedded MATLAB code in one or more M-files
Real-Time Workshop®	C code generation from MATLAB algorithms included in Simulink models, Stateflow charts, or from the MATLAB command line (emlc command)
Simulink HDL Coder™	Verilog and VHDL code generation from MATLAB algorithms included in Simulink models and Stateflow charts
Simulink Verification and Validation™	Develop designs and test cases mapped to requirements and measure test coverage
Simulink Design Verifier™	Generate tests and prove model properties using formal methods

Bonus Tips for Efficient Code Generation

- Examine algorithm to see if you can extract some sections into Simulink.
  - Embedded MATLAB very good for array manipulation and conditional execution.
  - Simulink better for complex computation
- Examine use of colon (:) operator.
  - Remove it completely if operating on entire array/matrix
  - Unravel matrix operations and write out For loop you would expect to see.
- Inline functions were possible

Bonus Tips (Continued)

- Embed Embedded MATLAB functions within Stateflow and pass large data via ‘global’ chart level variables.
  - Reduces data copies
- Can “encourage” pass-by-reference for sub-functions if output is same variable as an input
  - Declare it as v = fcn(v) [Multiple inputs and outputs can be supplied]
  - Make sure call to function assigns output to input variable
- Control typing throughout your code to prevent unnecessary up-casting.
  - i.e. Type literals (uint8(1)) when used with typed variables

Bonus Tips (Continued)

- Turn off “Saturate on Integer Overflow” and build in protection explicitly
- Ensure %#eml is inserted in all m-files you expect to generate C code or C-Mex files from
  - Ensures Embedded MATLAB M-lint checks are turned on
  - Improves error messages generated by emlc and emlmex
- Use 1xN arrays rather than Nx1