



# MathWorks UK Conference 2008

## Simulink Code of Practice for MBDA

Tony Gailor

Head of Simulation and Modelling for the Anti-Armour, Air-Air and Air Defence sector of MBDA in the UK and France.

Capability Leader for Simulation and Modelling across UK & France

Tyrone Catt

Technical Expert for Simulation and Modelling

Team Leader for Future Projects in the area of Simulation and Modelling within MBDA.

Ref.: - Page 1 -

This document and the information contained herein is proprietary information of MBDA and shall not be disclosed or reproduced without the prior authorisation of MBDA. © MBDA 2005.

**MBDA**  
MISSILE SYSTEMS

# Agenda

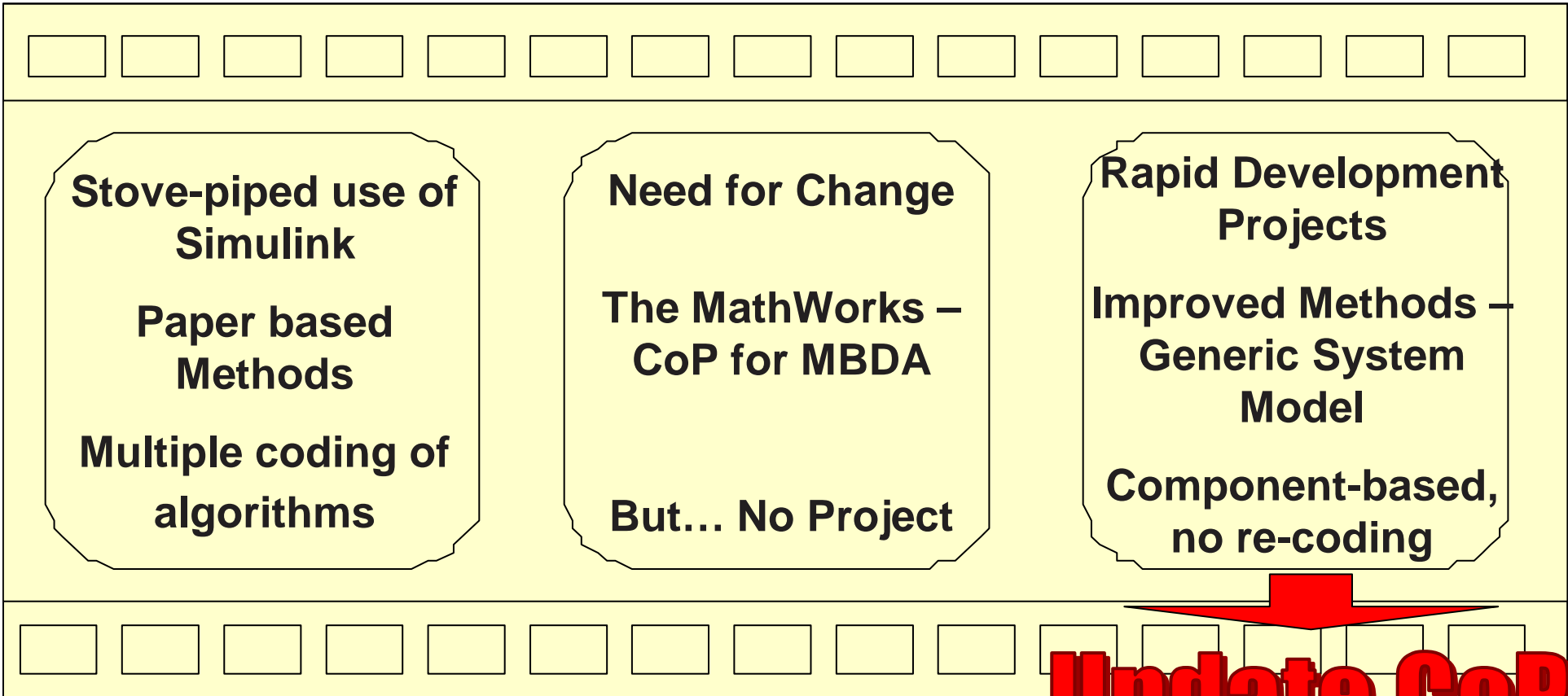
---

- Brief History & initial Simulink Code of Practice
- Key Principles for establishing a Code of Practice
- Mandatory Rules
- Summary

# A Little bit of history

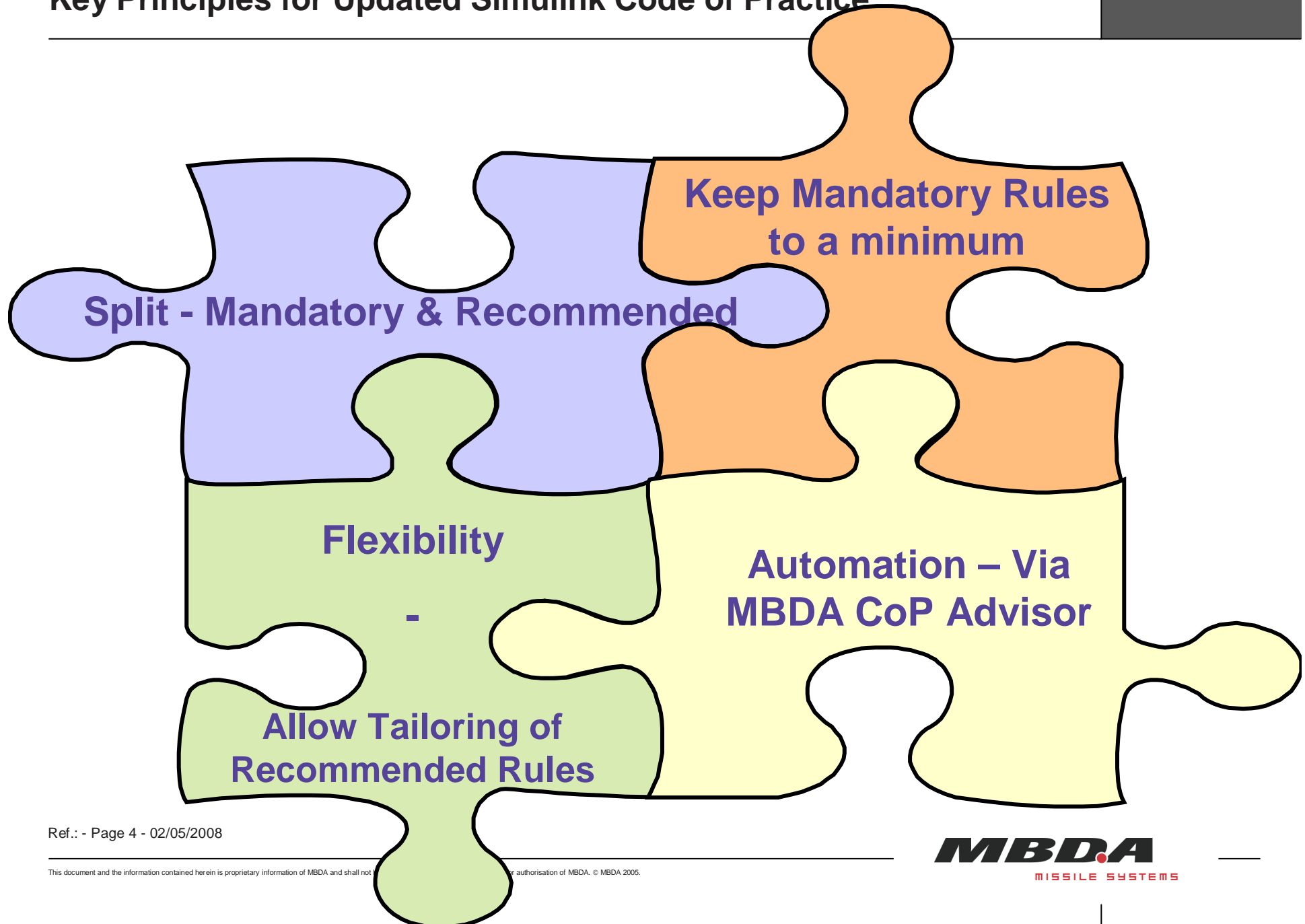


**Then . . . . . 2003 . . . . . Now**



**Update CoP**  
MISSILE SYSTEMS

# Key Principles for Updated Simulink Code of Practice

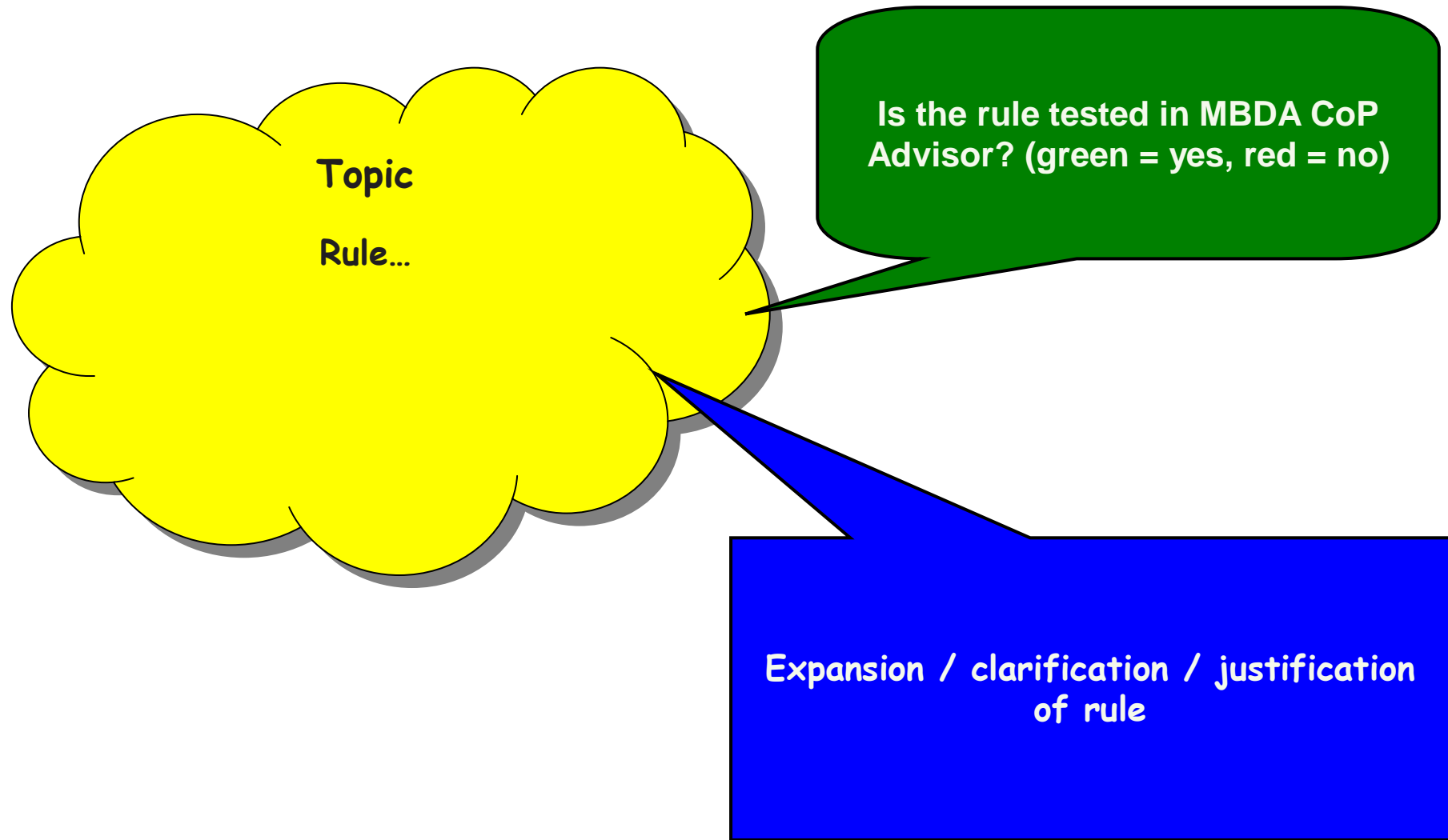




# The MBDA Code of Practice for Simulink

## Mandatory Rules

# Schematic – Rules listed in Priority Order





**Diagram Layout**

Ensure that no portion of a signal line is coincident (overlaid) with a portion of another as this can cause confusion tracing signal paths

Available in MBDA CoP Advisor

This may look neat - but it is confusing on print outs & on diagrams requires user manipulation to unravel which signal has come from/gone to - adding avoidable time and effort when debugging or reviewing.  
Better to use Databuses



## Diagram Layout

Draw signal lines vertically and horizontally, not heading off at many different angles

Tested in MBDA CoP Advisor

Lines running in parallel can be very neatly drawn side by side round right angled bends. Avoid lines crossing over as far as possible.





## Diagram Layout

**Goto and From blocks - Never use across module boundaries**

**Tested in MBDA CoP Advisor**

**Using these block breaks visibility of I/O for the module, which may result in interfacing mistakes.**



**Diagram Content**

Avoid algebraic loops as they slow the simulation down significantly. Therefore set the "allow algebraic loop" flag to "error".

**Tested in MBDA CoP Advisor**

With most single-body systems, it should be possible to avoid most algebraic loops, particularly if sensor and actuator dynamics are modelled.



**Diagram Content**

**Memory block - Never use with variable step solvers.**

**If the system is discrete, use the unit delay block instead.**

**Tested in MBDA CoP Advisor**

**Avoid using to fix algebraic loops.**



**Diagram Content**

Transport delay and variable transport delay - Don't use these blocks if at all possible; Try to use integration step dependent discrete time delays instead.

**Tested in MBDA CoP Advisor**

Any delays in a missile system would typically be due to processing, and as such should be modelled using discrete time delays. This is computationally much more efficient.



**Diagram Content**  
**Derivative block - Never use this block.**

**Tested in MBDA CoP Advisor**

**It is numerically inefficient, and should not be required for modelling most continuous systems. If a derivative is required in an algorithm, use a discrete time unit delay to calculate the derivative.**



**Diagram Content**

Set Mux block dimension parameters to -1, to force Simulink to determine what the dimensions are before running the simulation.

Tested in MBDA CoP Advisor

**Under Review**

However...for cases where the signal dimensions are fixed regardless of application, then it is strongly recommended to set the dimensions accordingly. This not only helps the block/model being misused/misinterpreted by someone else, but also reduces model initialisation time.

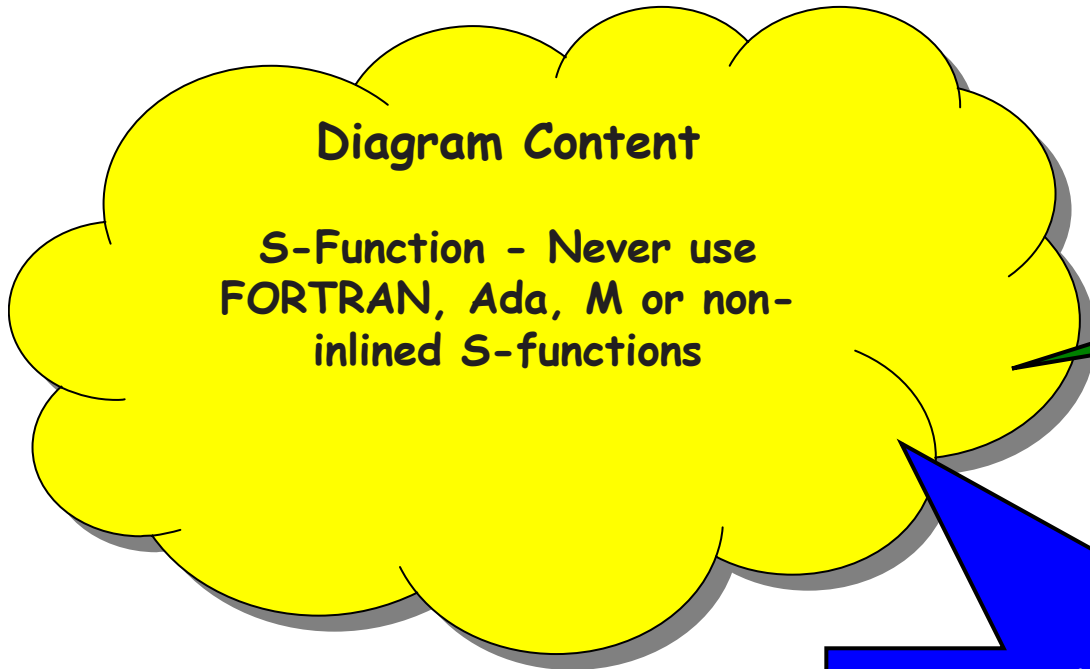


**Diagram Content**  
**MATLAB Fcn - Never use this block**

**Tested in MBDA CoP Advisor**

**Incompatible with Real-Time Workshop, hence making use with SME Target impossible (thus reducing the ability to re-use models).**

**Generally less efficient than implementing using native Simulink blocks.**



**Tested in MBDA CoP Advisor**

**Incompatible with Real-Time Workshop, hence making use with SME Target impossible (thus reducing the ability to re-use models).**

**Embedded MATLAB and C S-functions are permissible.**





## Module Information

Indicate the name, version, status and date of the model with text in the highest level of the block diagram.

Tested in MBDA CoP Advisor

This is good for configuration and control. The Model Info block can be used for this and can be configured to extract relevant information from your configuration management system. E.g. from Dimensions - Highly Recommended

## Module Information

Ensure that the DOC block employs an OpenFcn of the correct form to allow linking to HTML in the same directory as the model

Tested in MBDA CoP Advisor

**Under Review**

The steps are:

- Copy a DOC block into your model.
- Right click, and select Block Properties. Select the Callbacks tab, and then the OpenFcn callback.
- Replace the docblock callback with web('htmlfilename.htm'). If the documentation is stored in a different directory, then add the path to the filename.



**Module Information**

A defined set of units shall always be used at interfaces between modules. Where non-SI units are used, then the relevant signal and port names shall include units in their names.

Tested in MBDA CoP Advisor

**Benefit:** interfacing mistakes are much less likely to be made.  
For example, an angle-of-attack signal in degrees could be given the name 'alpha\_deg', whereas its counterpart in radians could simply be 'alpha'.  
M file contains set of acceptable units

# Recommendation 1

## Zero-Crossing

For simulation efficiency, turn-off zero-crossing events if they are not required check for selected solver & which blocks have zero-crossing turned on

Not Tested in MBDA CoP Advisor

Each of the following blocks which use zero-crossing, has a tick-box in the mask to facilitate this. Turning off all zero-crossing detection from the Simulation parameters drop-down menu is not recommended as other model users may miss that it has been turned off.

Abs, Backlash, Dead Zone, Hit Crossing, Integrator, MinMax, Relay, Relational Operator, Saturation, Sign, Step, Subsystem, Switch

For Details of Zero Crossing in these blocks [Click Here](#)

# Blocks which use Zero-Crossing

Block	Number of zero crossings and description
Abs	One: to detect when the input signal crosses zero in either the rising or falling direction.
Backlash	Two: one to detect when the upper threshold is engaged, and one to detect when the lower threshold is engaged.
Dead Zone	Two: one to detect when the dead zone is entered (the input signal minus the lower limit), and one to detect when the dead zone is exited (the input signal minus the upper limit).
Hit Crossing	One: to detect when the input crosses the threshold. These zero crossings are not affected by the <b>Disable zero crossing detection check box in the Simulation Parameters dialog box.</b>
Integrator	If the reset port is present, to detect when a reset occurs. If the output is limited, there are three zero crossings: one to detect when the upper saturation limit is reached, one to detect when the lower saturation limit is reached, and one to detect when saturation is left.
MinMax	One: for each element of the output vector, to detect when an input signal is the new minimum or maximum
Relay	One: if the relay is off, to detect the switch on point. If the relay is on, to detect the switch off point.
Relational Operator	One: to detect when the output changes.
Saturation	Two: one to detect when the upper limit is reached or left, and one to detect when the lower limit is reached or left.
Sign	One: to detect when the input crosses through zero.
Step	One: to detect the step time.
Subsystem	For conditionally executed subsystems: one for the enable port if present, and one for the trigger port, if present.
Switch	One: to detect when the switch condition occurs.





**Diagram Content**

**Band-limited white noise block - sample time should be at least 100 times smaller than the smallest time constant of interest**

**Not Tested in MBDA CoP Advisor**

**Use this block with caution. The dialog box entry labelled "Noise Power" is in fact the height of the power spectral density. Care must be taken to set the sample time to a suitably small value. In preference, use the White Noise source block plus an appropriate first order filter. Reduces risk if incorrect noise level implemented.**

## Recommendation 3

**Repeated sub-models**  
If the simulation model contains several copies of a sub-model this should be made into a 'library' block. This block is then copied and repeated, and all parameter information relating to a particular instance of the library block should be contained in a mask.

Not Tested in MBDA CoP Advisor

Warning rather than error (identical in terms of functional layout not necessarily signals)



## Defensive Programming

M-file name: Never choose filenames which clash with existing MATLAB toolbox filenames.

Not Tested in MBDA CoP Advisor

In the event of two identical filenames on the MATLAB path, MATLAB picks the first one it finds. This will result in different results depending on the MATLAB path ordering on any given machine.



## Recommendation 5

### Portability

Never use upper or lower case letters as the only distinguishing feature of different m-function names.

Not Tested in MBDA CoP Advisor

This is to ensure portability between Windows and UNIX based systems. UNIX does distinguish between upper and lower case names, whereas the Windows operating system does not.

# Recommendation 6



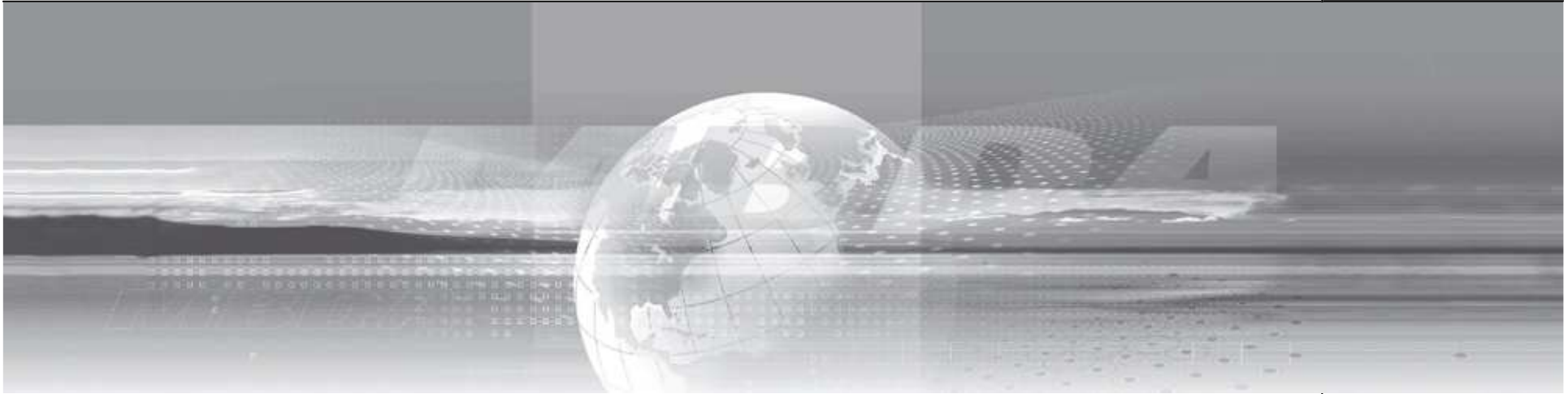
## Diagram Clarity

Use grouped sub-systems. Simulink makes it easy to divide the model up into sub-systems containing related parts, with the higher level screen showing the sub-systems as single blocks.

Not Tested in MBDA CoP Advisor

Try to avoid the diagram becoming larger than the maximum window size.  
Benefit: aids readability and understanding of your model by others  
Where possible, use atomic sub-systems to assist autocoding.

- Key Benefits
  - Improvement in Simulink model standard
  - Improved efficiency during Simulink model exchange
  - Improved diagram readability → improved methods across Organisation
- Code of Practice is a Living Document
  - Still learning and updating rules
  - Balance of improved efficiency vs implementation cost
- Important to stick to Key Principles
  - Split into Mandatory and Recommended
  - Keep Mandatory Rules to a minimum
  - Flexibility – allow tailoring of Recommended Rules
  - Automation – use of Model Advisor technology



# Questions?