

Filter Design, C code generation and Deployment

Key Takeaways

- ❑ Design and test filter algorithms quickly in MATLAB
- ❑ Benefit from automatic C code generation out of the algorithms
- ❑ Integrate generated C code easily into external C projects from other IDEs
- ❑ Deploy the entire design as a standalone application using the MATLAB Runtime

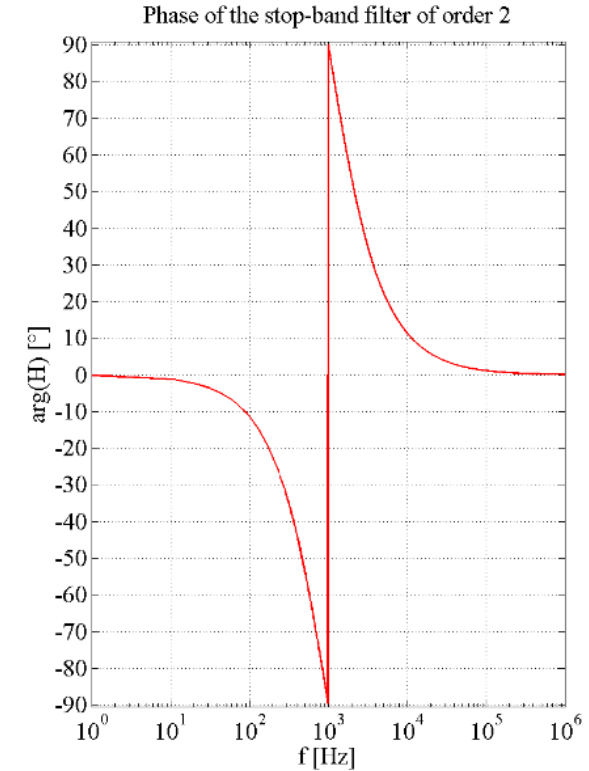
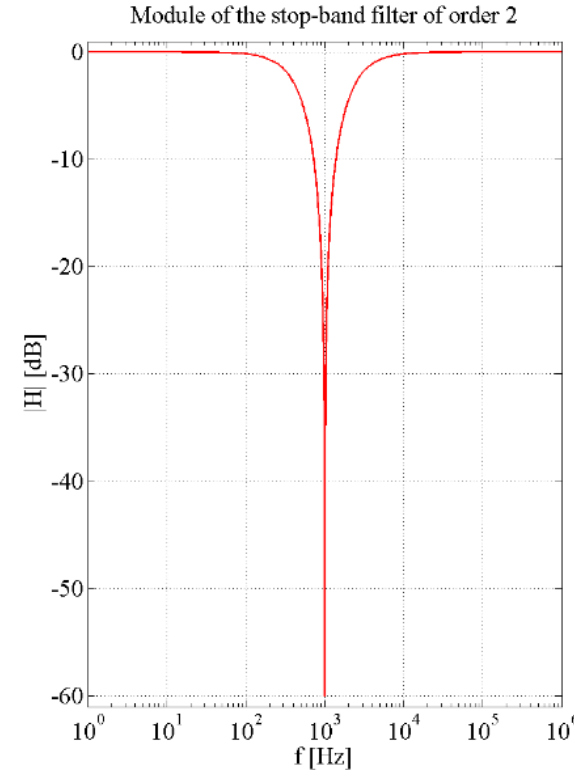
Filter's Bodes Representation in MATLAB

Design and test of the algorithm

- frequency range
- type and order of filter
- cutoff frequency
- quality factor



```
% Pulsation's vectors
w  = 2*pi*frequency;
w_0 = 2*pi*f_0;
% Transfer function
H = 1./ (1+(1i/w_0)*w);
H = (1i/w_0)*w.*H;
```



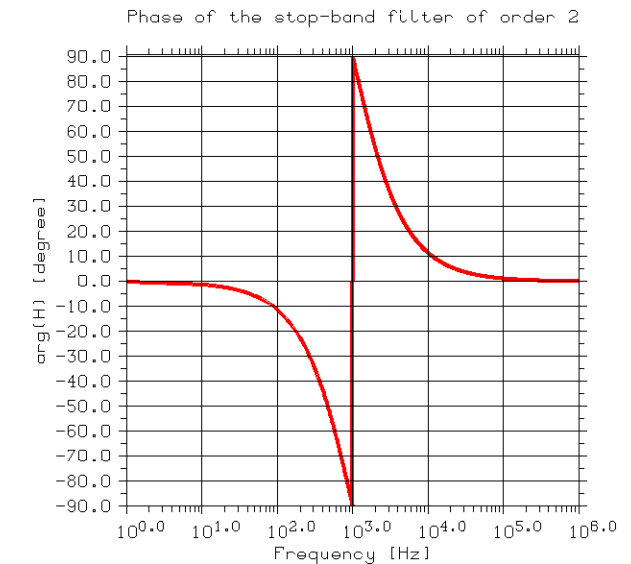
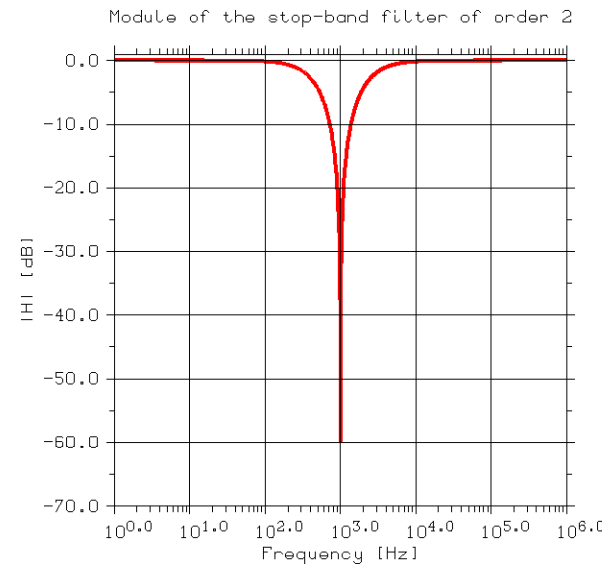
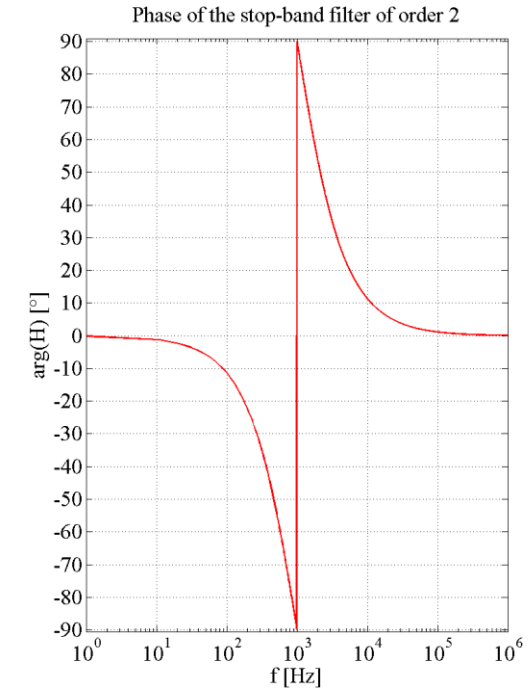
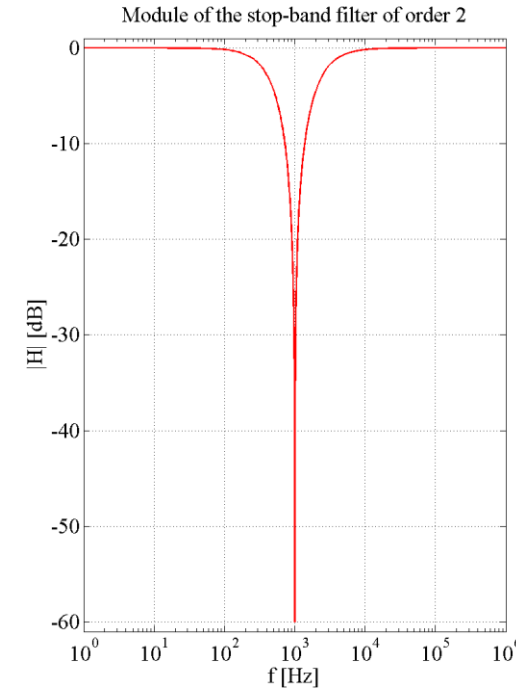
From MATLAB to C

Code Generation

Matlab's user
filter's Bodes
visualization

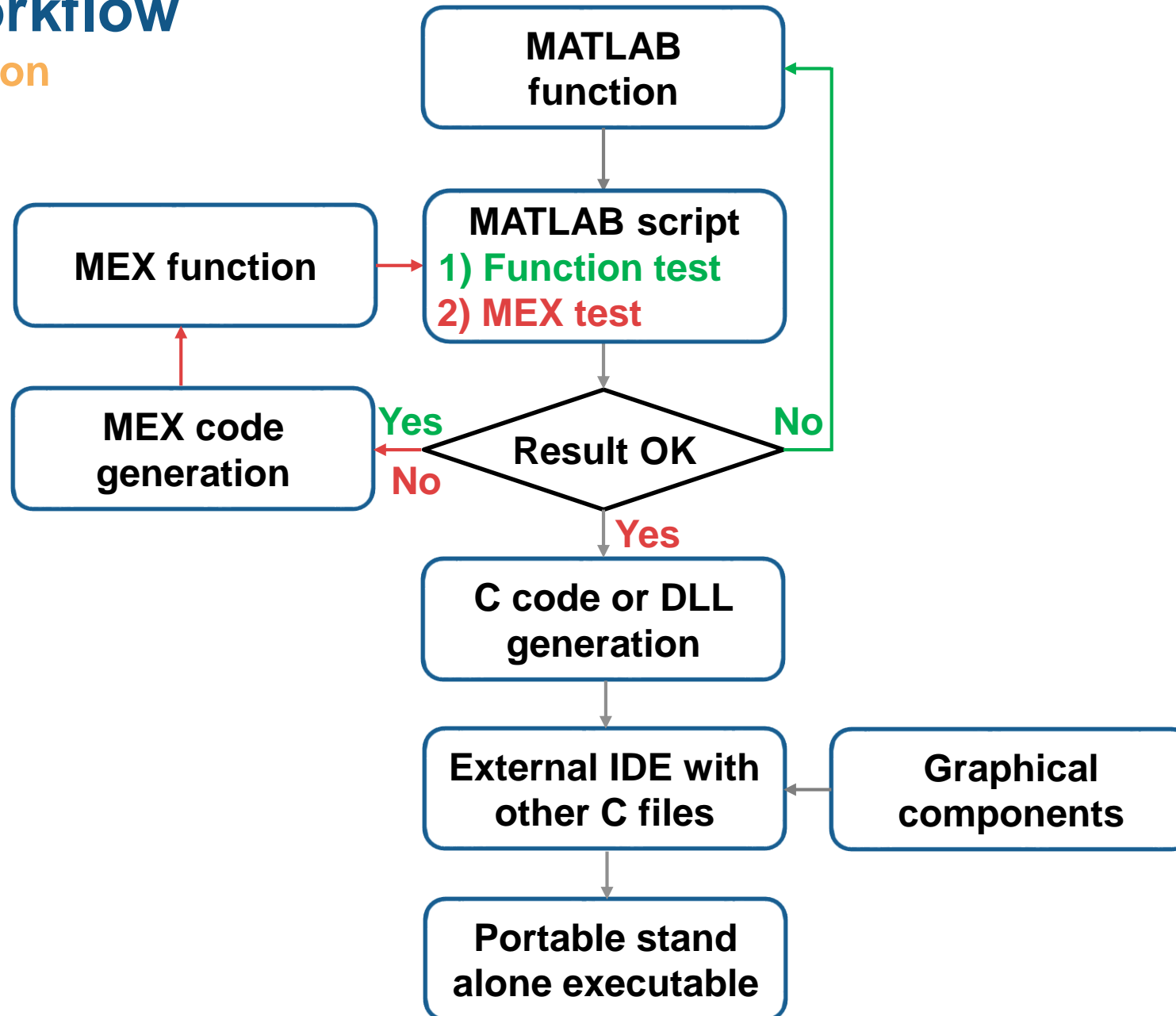
Automatic C code
generation

Full portable
ANSI C version
of the project




Design workflow

Code Generation



Application Deployment

App Designer to C#


UI Figure

Inputs

Outputs

Filter's parameters

Type:

Lowpass

Cutoff Frequency:

1

Order:

5

6

7

8

9

10

1

2

3

4

3

Quality Factor:

1

Linear Gain:

1

Compute Filter's Data


Create Bodes

Status

View Bodes

Log Message:

The Bodes computation finished successfully


UI Figure

Inputs

Outputs

Filter's parameters

Type:

Lowpass

Cutoff Frequency:

1

Order:

5

6

7

8

9

10

1

2

3

4

3

Quality Factor:

1

Linear Gain:

1

Compute Filter's Data

Create Bodes

Status

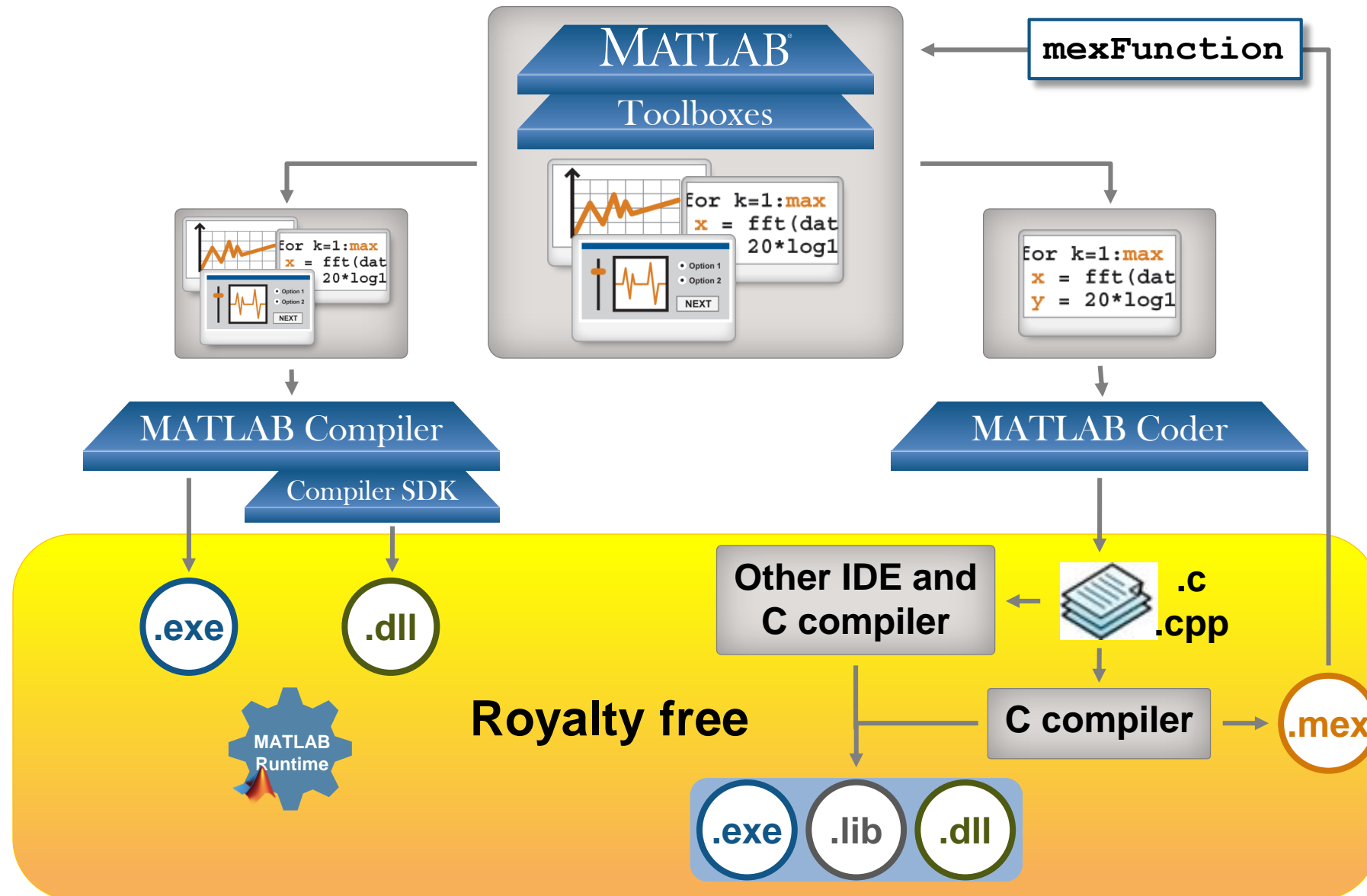
View Bodes

Log Message:

The Bode plots have been computed successfully.

6

MATLAB Application Deployment



Key Takeaways

- ❑ Design and test filter algorithms quickly in MATLAB
- ❑ Benefit from automatic C code generation out of the algorithms
- ❑ Integrate generated C code easily into external C projects from other IDEs
- ❑ Deploy the entire design as a standalone application using the MATLAB Runtime