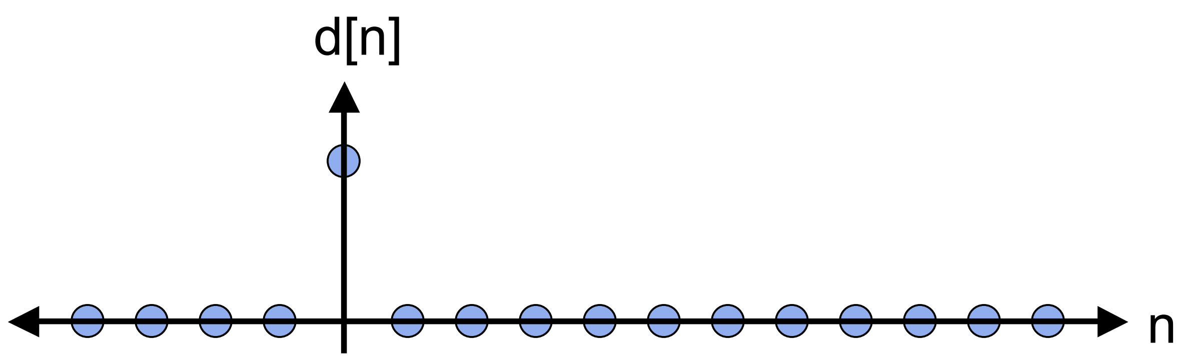
ECE4703: Implementation of Real-Time FIR Filters

D. Richard Brown III
Associate Professor
Worcester Polytechnic Institute
Electrical and Computer Engineering Department
drb@ece.wpi.edu

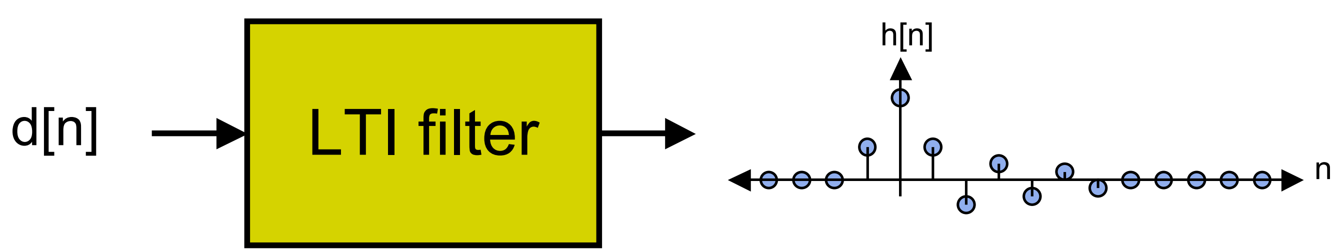
Lecture 5: 31-Oct-2006

Signals Review

- **Definition**: A discrete time impulse function, d[n], is defined as: d[n] = 1 if n=0, d[n] = 0 otherwise.



- **Definition**: The “impulse response” of a linear time invariant filter is the output that occurs if the input is d[n].



Finite Impulse Response (FIR) Filtering – Basics

- **Definition**: A filter is FIR if there exists $N<\infty$ such that the filter's impulse response $h[n]=0$ for all $n>N$.
- FIR filters are frequently used in real-time DSP systems
  - Simple to implement
  - Guaranteed to be stable
  - Can have nice properties like linear phase
- Input/output relationship

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$

x=input, ***y***=output, ***h***=impulse response (aka "filter coefficients")
M=# of filter coefficients

Finite Impulse Response (FIR) Filtering – More Basics

- Transfer function

$$H(z) = \sum_{m=0}^{M-1} z^{-m} h[m]$$

- Frequency response

$$H(e^{j\omega}) = \sum_{m=0}^{M-1} e^{-j\omega m} h[m]$$

Implementation of FIR Filters

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$

- If everything is infinite precision, then there is nothing to worry about.
- Finite precision raises some issues:
  - Precision:
    - How is the input signal quantized?
    - How is the output signal quantized?
    - How are the filter coefficients quantized?
    - How are intermediate results (products, sums) quantized/stored?
  - "Realization Structure"
    - In what order should we do the calculations?

Actual performance can be significantly affected by these choices.

FIR filtering is usually less sensitive to these choices than IIR filtering for reasons we will see later.

Typical Procedure for Designing and Implementing FIR Filters

Matlab

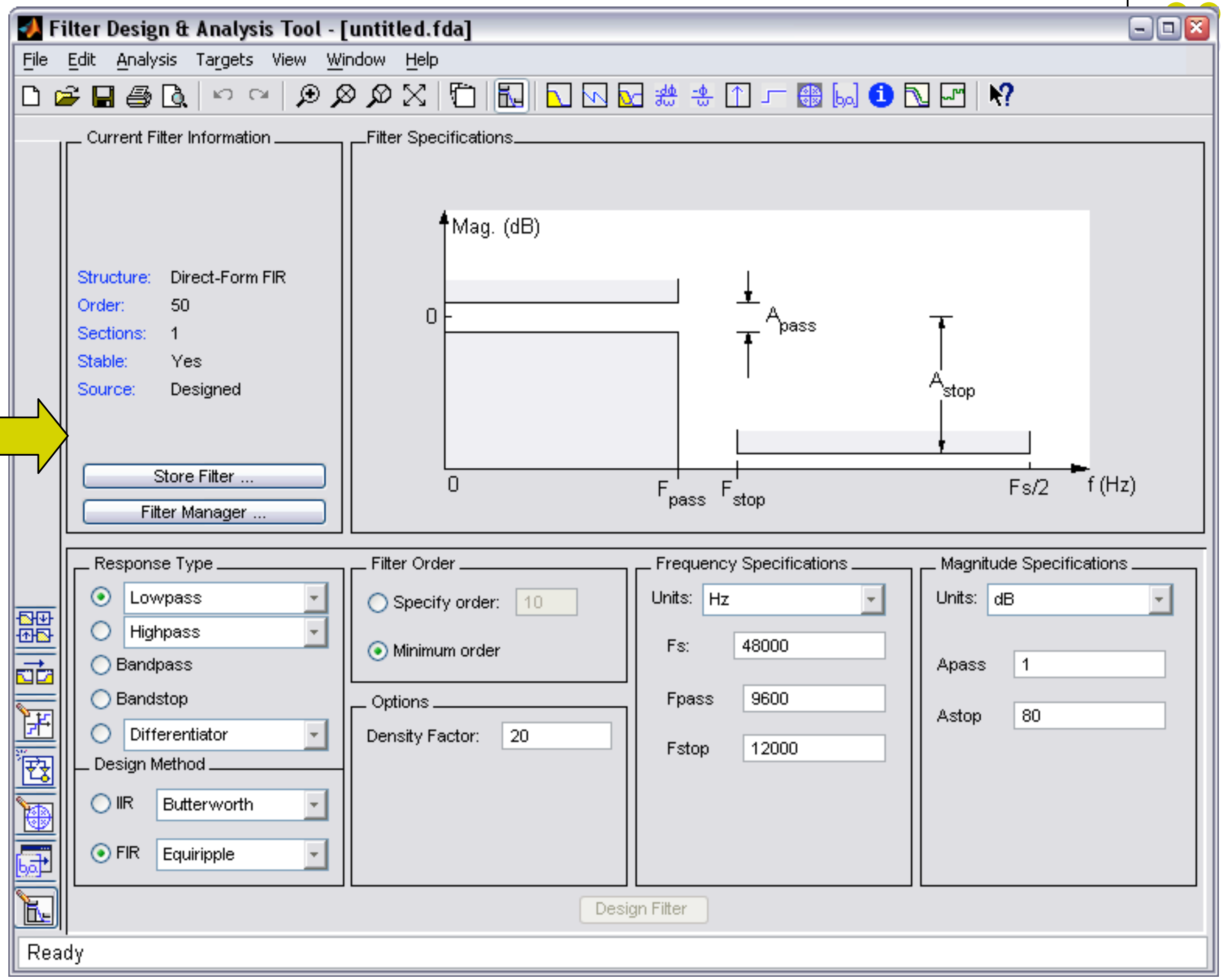
1. Design filter
   - Type: low pass, high pass, band pass, band stop, ...
   - Filter order M
   - Desired frequency response
2. Decide on a realization structure
3. Decide how coefficients will be quantized.
4. Compute coefficients

CCS

5. Decide how everything else will be quantized (input samples, output samples, products, and sums)
6. Write code to realize filter (based on step 2)
7. Test filter and compare to theoretical expectations

Tools for Designing FIR Filters

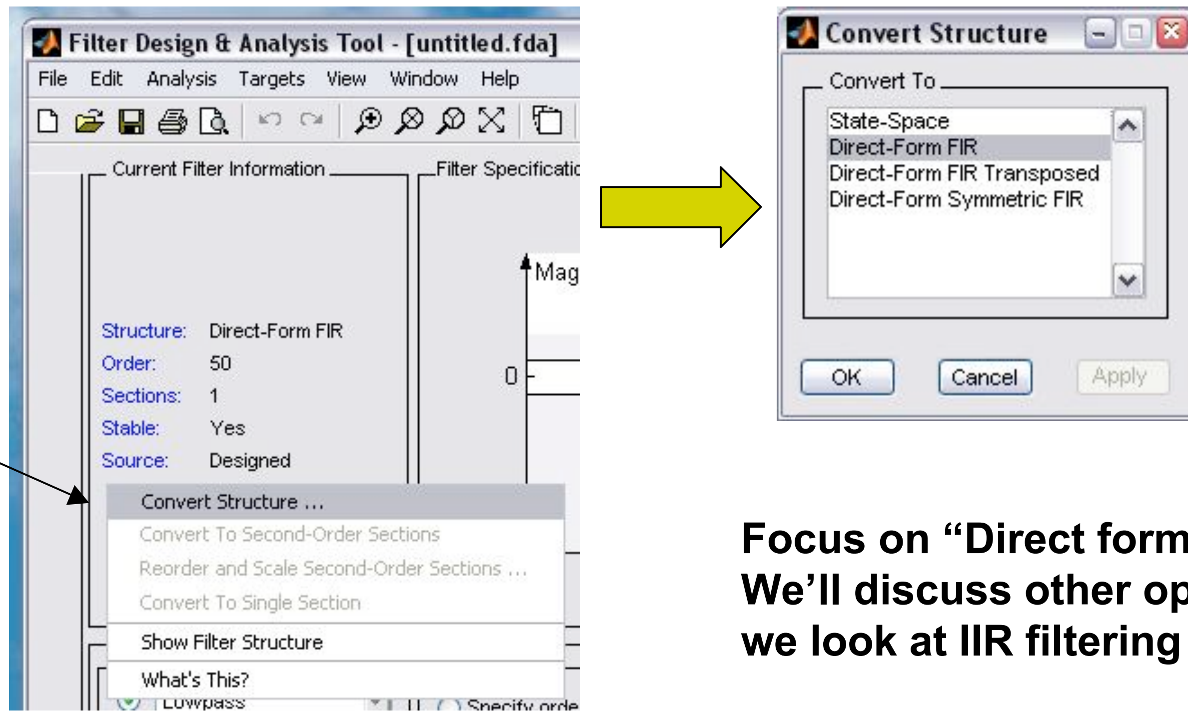
>> fdatool



Filter Realization Structures

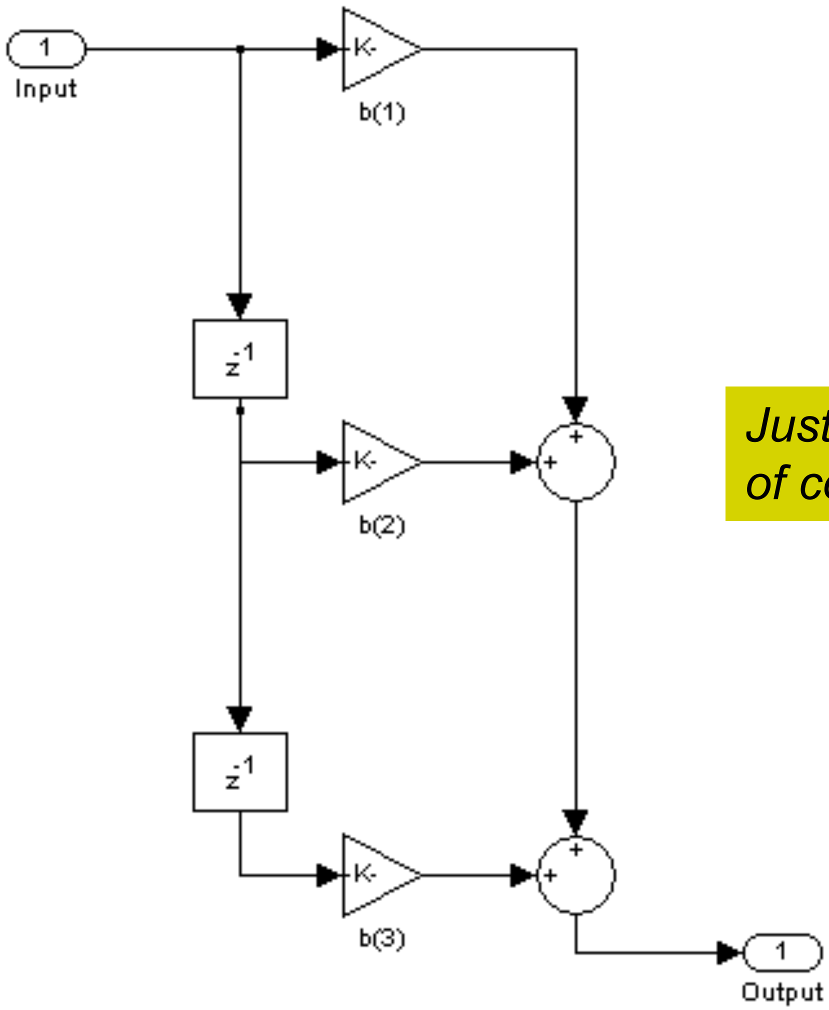
- Filter realization structure specifies **<u>how past calculations are stored and the order in which calculations are performed.</u>**
- Lots of different structures available
  - Direct form I, direct form II, transposed forms, cascade, parallel, lattice, …
  - Choice of structure affects computational complexity and how quantization errors are manifested through the filter

right click
in this pane



Focus on “Direct form” for now. We’ll discuss other options when we look at IIR filtering later.

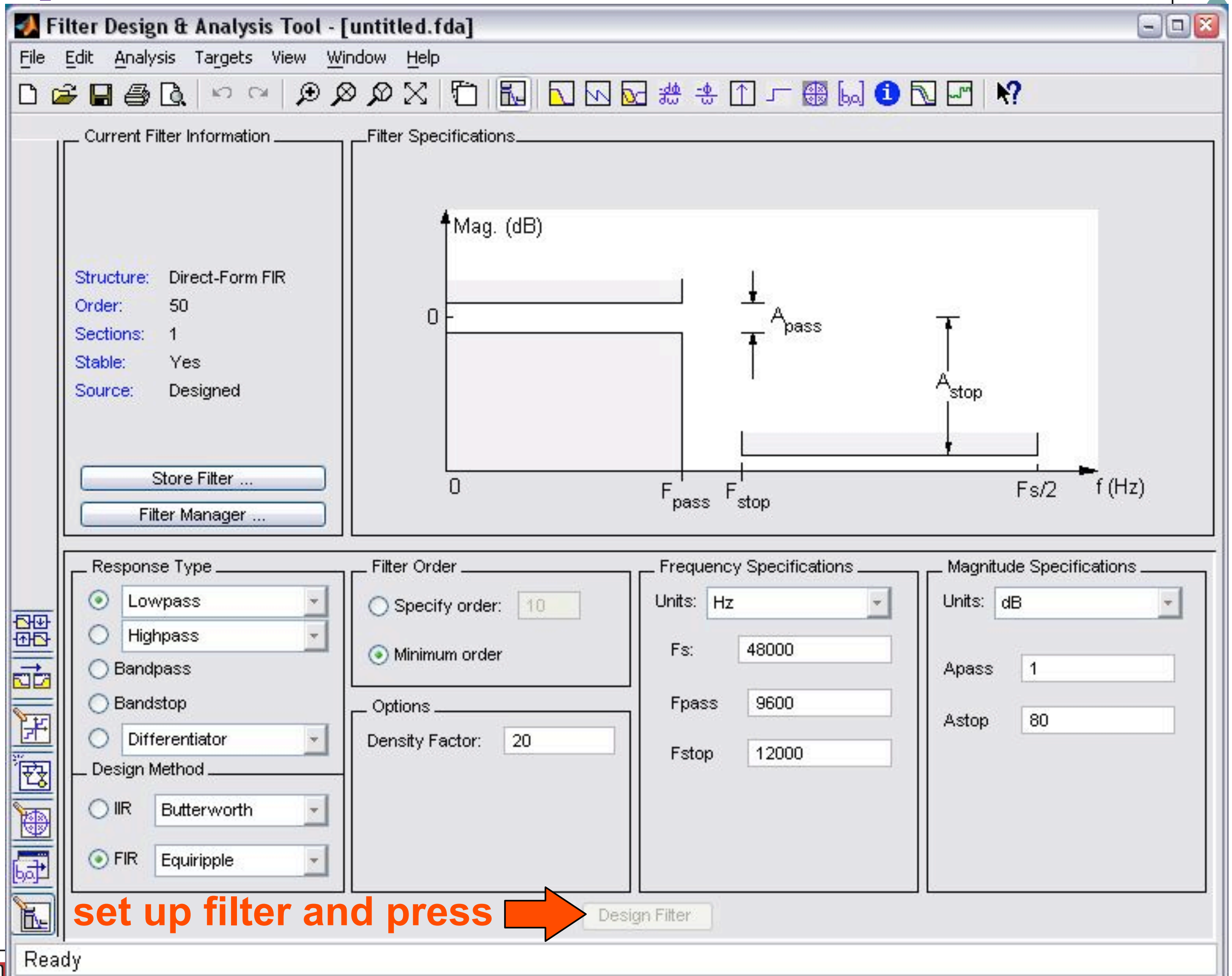
Direct Form I Filter Structure



Just a pictorial depiction of convolution.

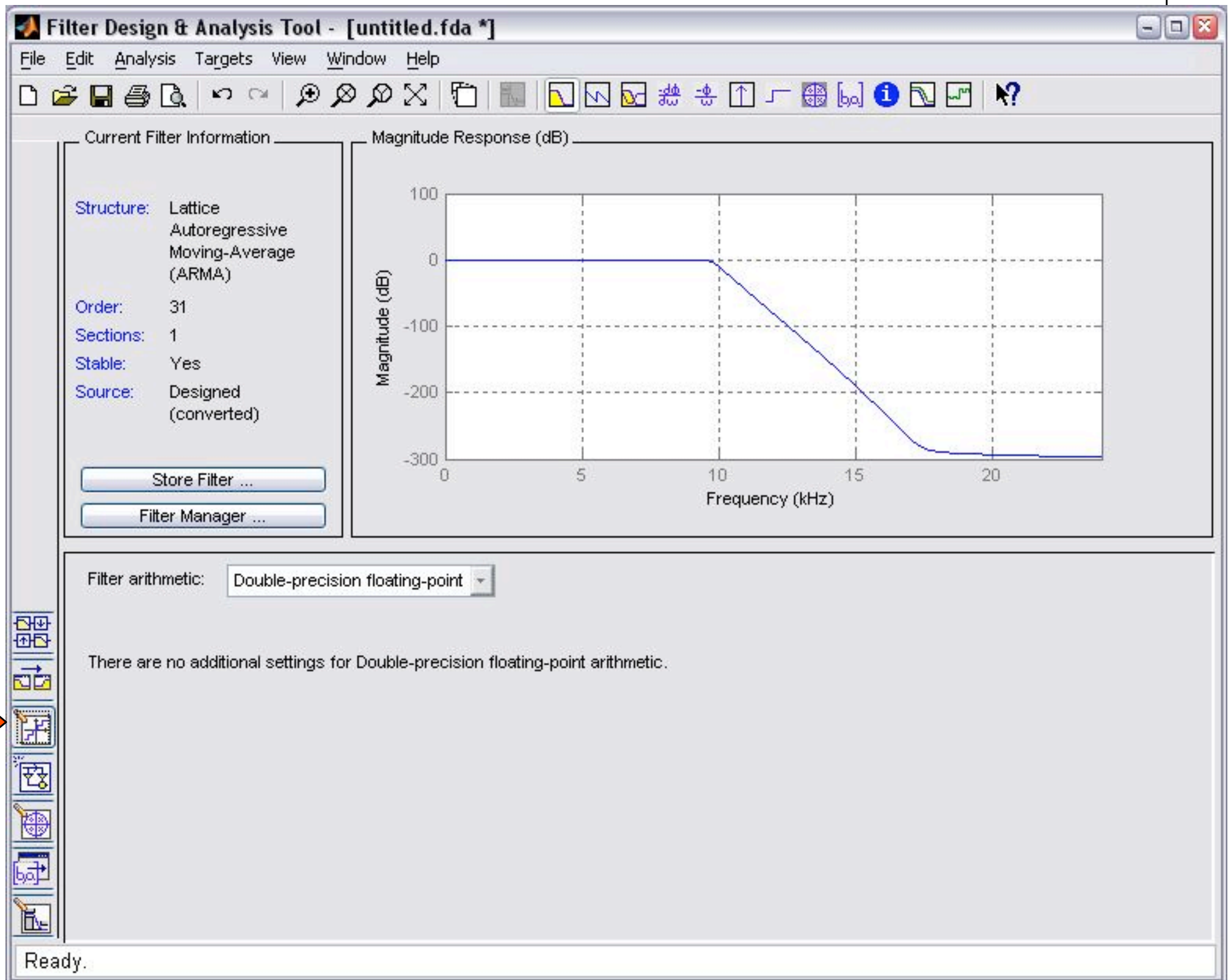
(picture from Matlab's help system)

Compute FIR Filter Coefficients

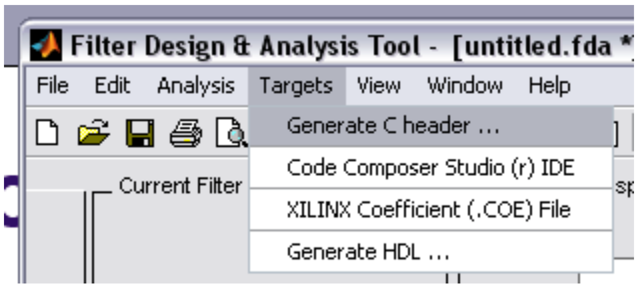


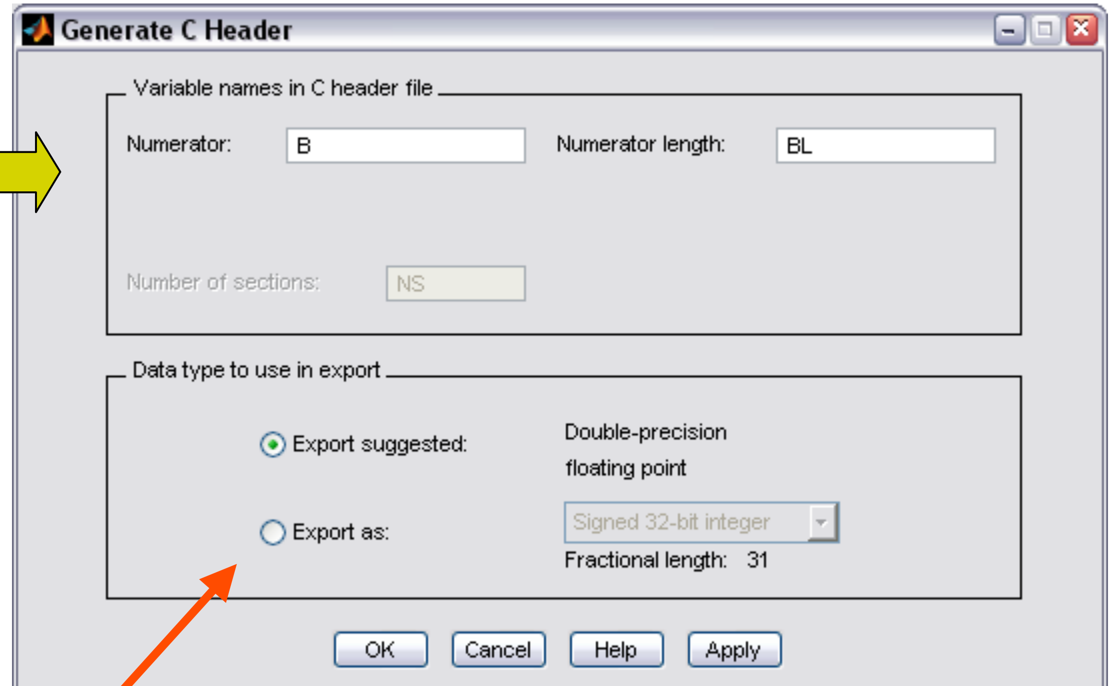
Determining How Coefficient Quantization Will Affect Your Filter

set quantization parameters



Make Coefficient File For CCS





Here you can change the coefficient data type to match your desired quantization.

Example DP-FP Coefficient File

```
/*
 * Filter Coefficients (C Source) generated by the Filter Design and Analysis Tool
 *
 * Generated by MATLAB(R) 7.0 and the
 *
 * Generated on: 19-Aug-2005 13:04:09
 *
 */

/*
 * Discrete-Time FIR Filter (real)
 * -------------------------------
 * Filter Structure  : Direct-Form FIR
 * Filter Order      : 8
 * Stable            : Yes
 * Linear Phase      : Yes (Type 1)
 */

/* General type conversion for MATLAB generated C-code  */
#include "tmwtypes.h"
/*
 * Expected path to tmwtypes.h
 * C:\MATLAB7\extern\include\tmwtypes.h
 */
const int BL = 9;
const real64_T B[9] = {
    0.02588139692752,  0.08678803067191,   0.1518399865268,   0.2017873498839,
     0.2205226777929,   0.2017873498839,   0.1518399865268,  0.08678803067191,
    0.02588139692752
};
```

Note this new header file needed for CCS to understand Matlab's strange data types.

Add this header file to your project (in the Matlab directory tree) or edit the datatypes.

FIR Filter Coefficient Quantization Considerations

- Key choice: **floating point** vs. **fixed point**
- Advantages of floating point math:
  - Less quantization error
  - Don't have to worry about keeping track of scaling factors
  - Less likelihood of overflow/underflow
  - Much easier to code
- Disadvantages of floating point math:
  - Executes slower than fixed point
  - Requires you to use a floating-point DSP ($$$, power, heat,…)
- C code allows you to "cast" variables into any datatype

WORCESTER POLYTECHNIC INSTITUTE 1865 WPI

TMS320C6000 C/C++Data Types

Type	Size	Representation	Range Minimum	Range Maximum
char, signed char	8 bits	ASCII	-128	127
unsigned char	8 bits	ASCII	0	255
short	16 bits	2s complement	-32768	32767
unsigned short	16 bits	Binary	0	65535
int, signed int	32 bits	2s complement	-2147483648	214783647
unsigned int	32 bits	Binary	0	4294967295
long, signed long	40 bits	2s complement	-549755813888	549755813887
unsigned long	40 bits	Binary	0	1099511627775
enum	32 bits	2s complement	-2147483648	214783647
float	32 bits	IEEE 32-bit	1.175494e-38†	3.40282346e+38
double	64 bits	IEEE 64-bit	2.22507385e-308†	1.79769313e+308
long double	64 bits	IEEE 32-bit	2.22507385e-308†	1.79769313e+308

Write Code to Realize FIR Filter

- Direct form I implies direct realization of the convolution equation (multiply and accumulate)

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$

- Some practical considerations:
  - Allocate buffer of length M for input samples.
  - Move input buffer pointer as new data comes in or move data?
- See Kehtarnavaz Lab 4 examples.

Verifying your real-time filter works correctly

- Method 1: Sinusoids (easy but labor intensive)
  - Generate input sinusoid at frequency *f* with amplitude a_{in}.
  - LTI filter output will also be at frequency f but with amplitude a_{out}.
  - Magnitude response of the filter is $20\log_{10}(a_{out}/a_{in})$
  - Compare actual magnitude response to the predicted response from Matlab
- Method 2: White noise (more complicated but less work)
  - Generate at least 10 seconds of a white noise input signal (matlab command randn)
  - Record filter output to a file
  - Use Matlab command pwelch to estimate power spectral density of the output