EE482/682: DSP APPLICATIONS

CH1 INTRODUCTION TO REAL-TIME DSP

OUTLINE

- Intro to DSP
- Real-time DSP system components
- Matlab primer

SIGNALS

- Continuous-time (CT or analog)
  - Everyday signals from nature
  - Defined continuously in “time” – at all time instances
  - Infinite amplitude value resolution
  - Can be processed using analog electronics (active and passive circuit elements)
- Discrete-time (DT)
  - Only defined on particular set of “time” instances
  - Sequence of numbers with continuous value range
  - Used for theoretical study and mathematical convenience
- Digital
  - Both discrete “time” and discrete amplitude values
  - Processed with computers and DSP chips

WHAT IS DSP?

- Digital representation of signals (coding)
- Design and use of digital systems to
  - Analyze
  - Modify
  - Store
  - Transmit
  - Extract information

DSP ADVANTAGES

- Flexibility
  - Software implementation for upgrades, multiple tasks, etc.
- Reproducibility
  - Easier to repeat implementation, to store and transfer digital signals
- Reliability
  - DSP hardware design is quite robust due in modern computation age
- Complexity
  - Can implement sophisticated tasks on specialized hardware
- Cost
  - Moore's Law for semiconductors, software development cycle and powerful packages (e.g. Matlab, Python)

DSP DISADVANTAGES

- Unnatural
  - Our everyday signals come from analog processes
- Physical limitations
  - Bandwidth of DSP system limited by sampling rate, aliasing
- Numerical effects
  - Limited precision and dynamic range, quantization and arithmetic errors

REAL-TIME DSP SYSTEMS

- Non-real-time
  - Signals that are stored in digital form
  - Not necessarily for current or real-time data
- Real-time
  - Demands design to ensure tasks are completed within a given timeframe
  - Typically expect this to be related to the current time

- Emphasis on real-time in this class
  - Fun processing streaming data
  - See bandwidth processing time relationship in Section 1.3.4
    - Faster processing means less available bandwidth

REAL DSP SYSTEM

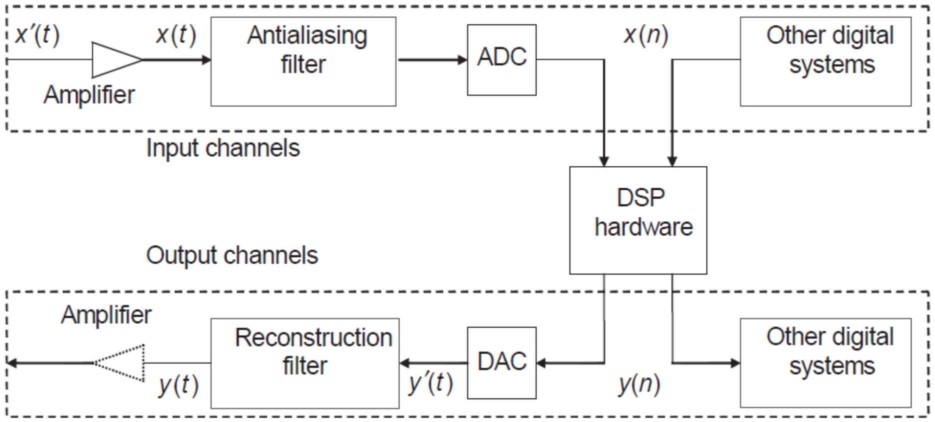


Figure 1.1 Basic functional block diagram of a real-time DSP system

- CT Analog signal
  - $x(t) \quad t \in \mathbb{R}$
- DT/digital signal
  - $x(n) \quad n \in \mathbb{Z}$
- ADC – analog to digital conversion
- DAC – digital to analog conversion

- Analog signals are converted to electrical by a transducer
  - E.g. Microphone
- Amplifier
  - Gain selected to match ADC
  - Often need auto gain control (e.g. white balance)
- Antialiasing filter
  - Deal with finite bandwidth of digital system
- Reconstruction filter
  - Interpolation between digital and analog signal

ADC – SAMPLING

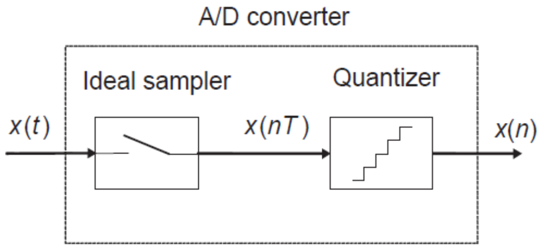


Figure 1.2 Block diagram of an ADC

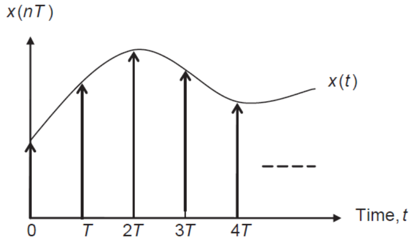
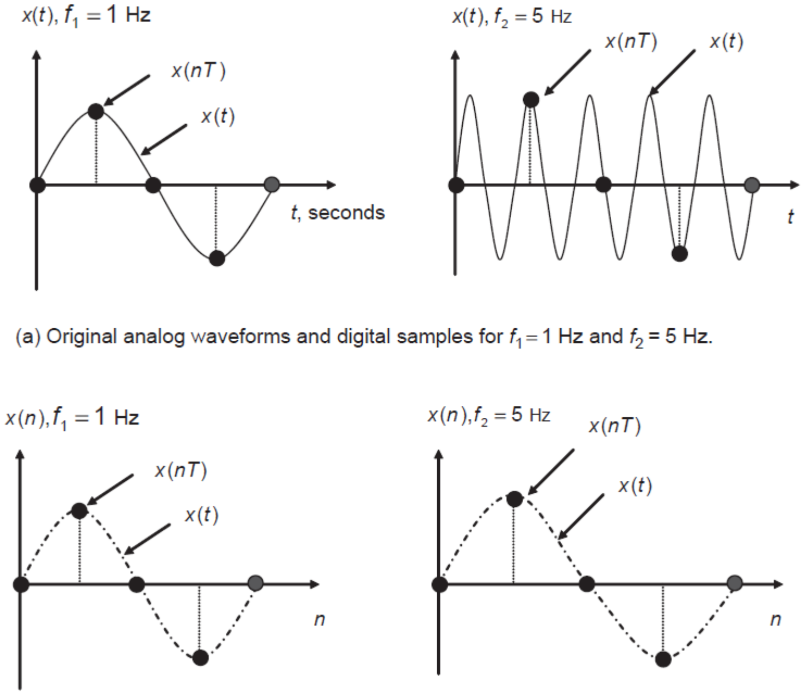


Figure 1.3 Sampling of analog signal $x(t)$ and the corresponding discrete-time signal $x(nT)$

- Sampling
  - $x[n] = x(nT)$
    - T – sampling period
  - Analog signal value extracted at fixed uniformly spaced times
- Shannon's sampling theorem
  - $f_s = \frac{1}{T} > 2f_M$
  - Sampling frequency must be twice the bandwidth to avoid aliasing
  - Nyquist rate - $f_n = 2f_M$



(a) Original analog waveforms and digital samples for $f_1 = 1$ Hz and $f_2 = 5$ Hz.

(b) Digital samples of $f_1 = 1$ Hz and $f_2 = 5$ Hz and the reconstructed waveforms.

Figure 1.4 Example of the aliasing phenomenon

ADC - QUANTIZATION

- Quantization
  - Amplitude value is represented by one of 2^B binary levels
  - Rounding – set value to closest quantization level
  - Truncation – replaces by value below it (chop bits)

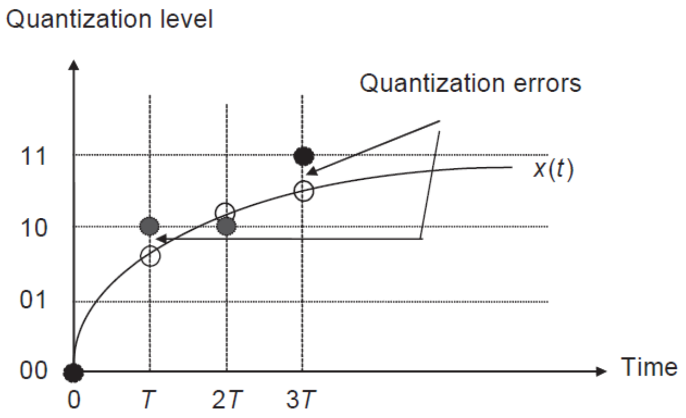


Figure 1.5 Digital samples using 2-bit quantizer

- Quantization error/noise
  - Difference between quantized value and original value
  - Appears as random noise at output of converter
  - Signal-to-quantization-noise ration(QNR)
    - $SQNR \approx 6B$ dB

SMOOTHING FILTERS

- DACs are zero-order-hold
  - Keep fixed sample value until next sample

- Smoothing with low pass (LP) filter is done to remove high frequency components of "staircase"
  - LP filter in reconstruction block

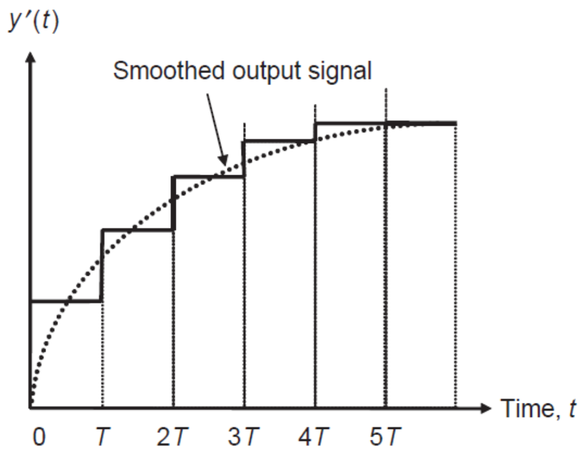
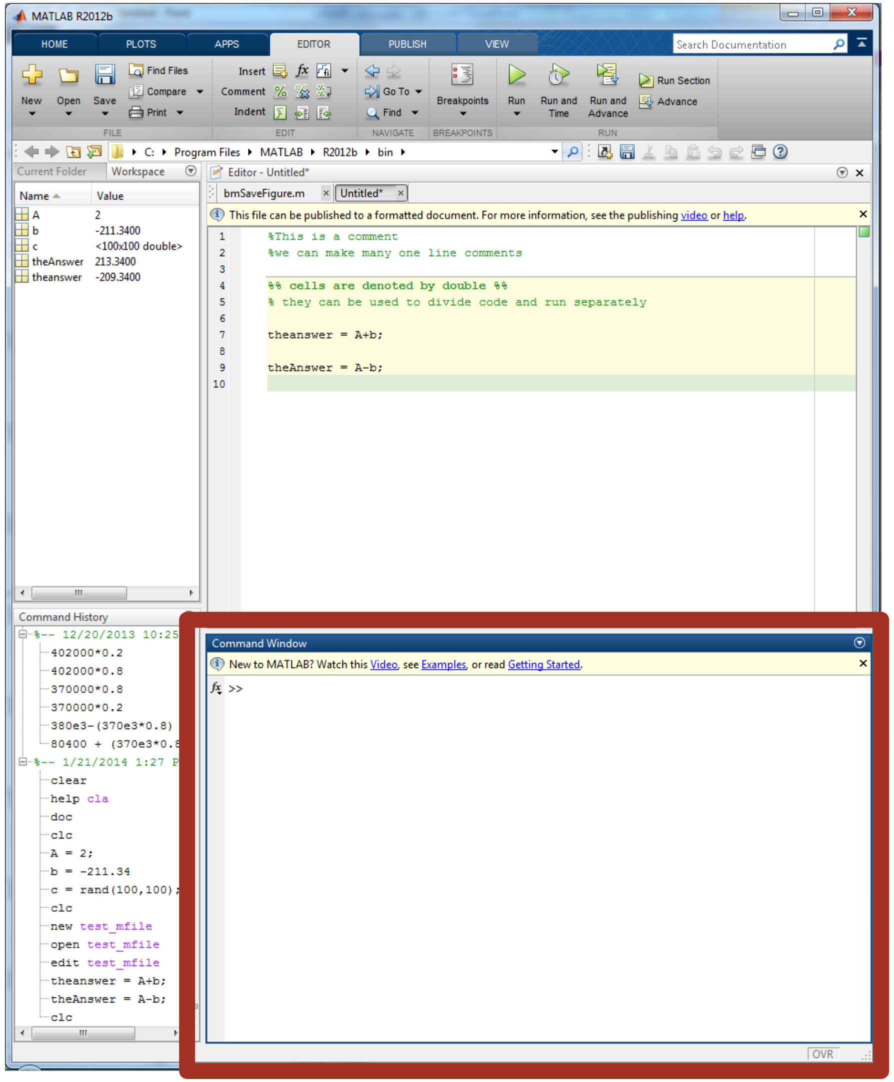


Figure 1.6 Staircase waveform generated by DAC and the smoothed signal

MATLAB PRIMER

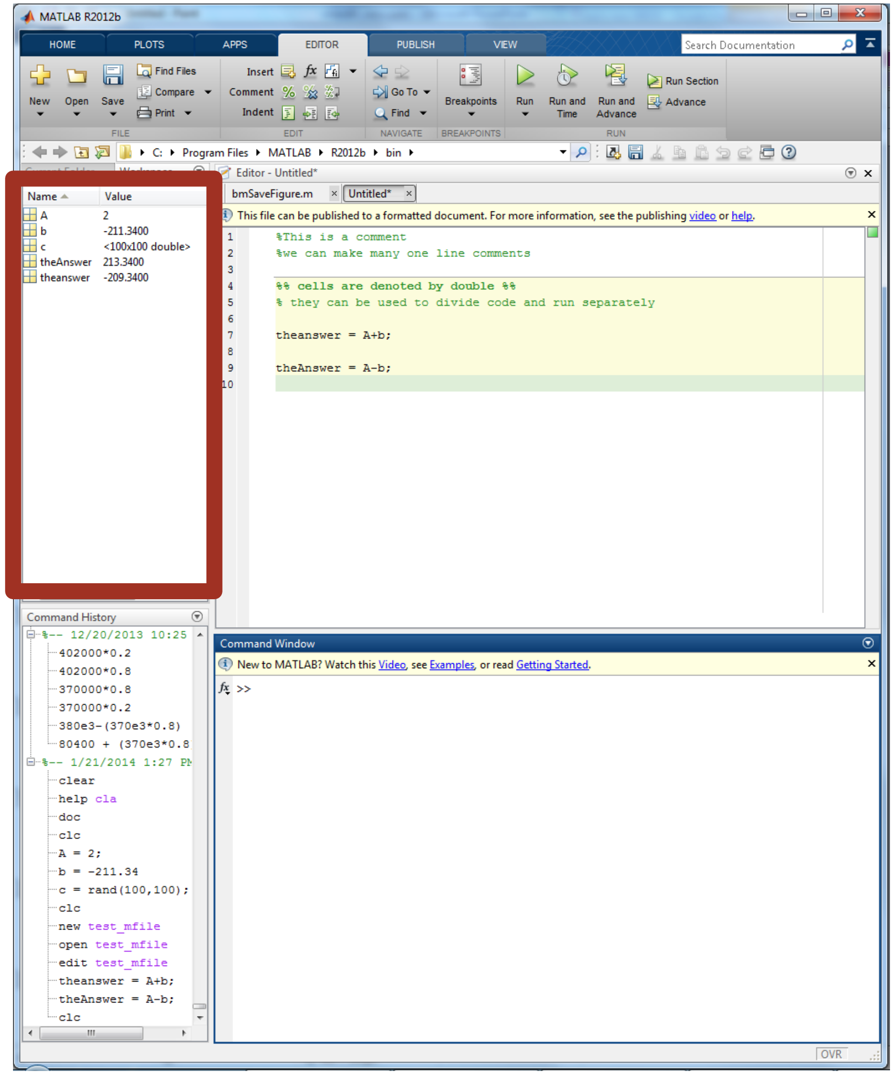
- See the web for many tutorials and help
  - https://matlabacademy.mathworks.com/
- Matlab has very good in program help
  - Use the help.m and doc.m commands
- Go through tutorials
  - Signal processing
  - Image processing

MATLAB PRIMER



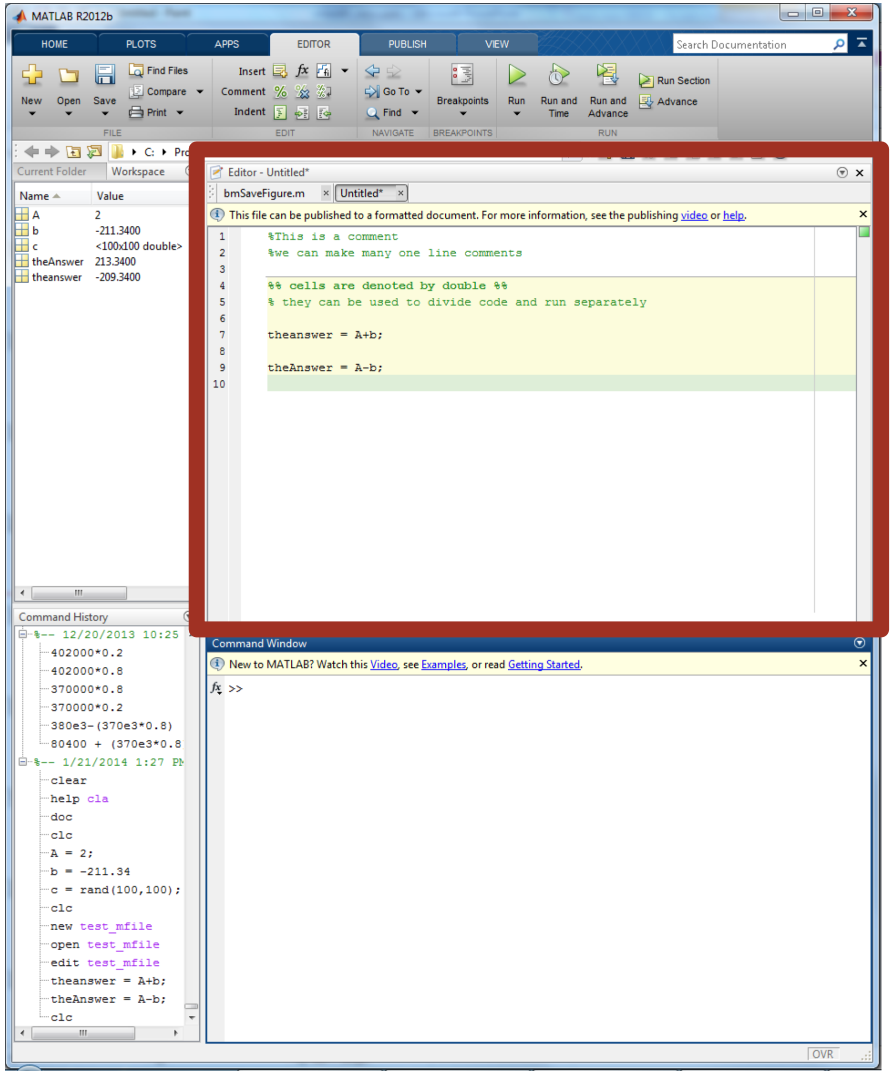
- Command Window
  - Interactive interpreted area
  - The calculator space

MATLAB PRIMER



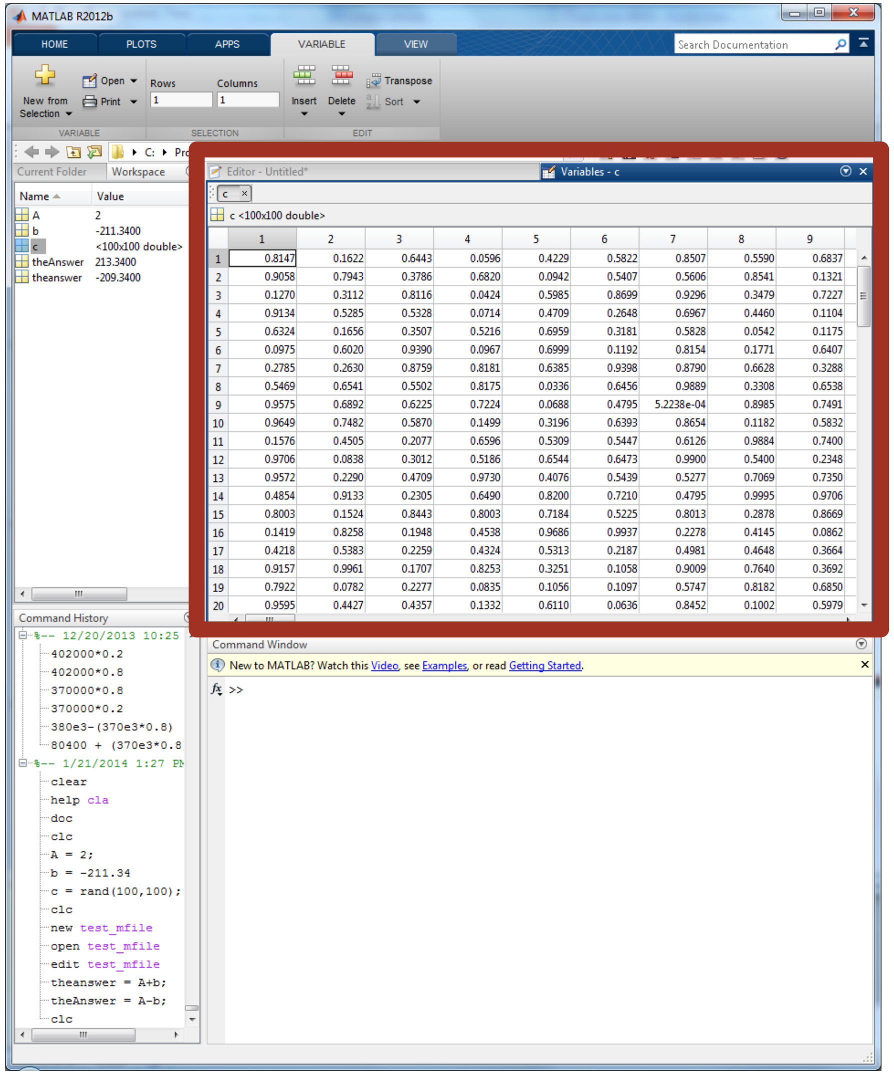
- Workspace
  - Lists all variables in memory
    - All Are currently available

MATLAB PRIMER



- Editor
  - Build script files (m-files)
  - What makes Matlab so much more than a calculator
- M-files
  - Learn to write these, it will make your life much easier
  - Provides ability to document and re-run code quickly
  - Must submit for class assignments
- Note:
  - ; suppresses command window output
  - % is comment character

MATLAB PRIMER



- Variables
  - Quick way to read contents of your workspace variables
- Useful for debugging
  - There is a debugger in Matlab!
    - Use the keyboard shortcuts (F5 run, shift-F5 stop, F10 step, F11 step-in)
  - Must write m-files to utilize this
- Ctrl+c to kill running code

PYTHON

- If you aren't familiar with Matlab (and even if you are), it is highly recommended to use Python
  - Free and lots of tutorials and support online
  - In heavy use for ML/AI
- We will not teach it in class so you'll have to look online for resources
  - Use Python virtual environments (venv) or Anaconda/Miniconda
  - Use a notebook like Juptyer or Google Colab to have an interactive interface