ELEC 599 Project:

Handset Algorithms & Architectures for Blind Channel Estimation & Detection

Frank Livingston

May 1, 2000

Project Objectives

- Investigate enhancements to existing algorithms & architectures to improve performance while maintaining BER.

- Investigate implementation on DSP's.
  - Study performance & real-time constraints.
  - Study fixed-point math issues = required dynamic range & precision.

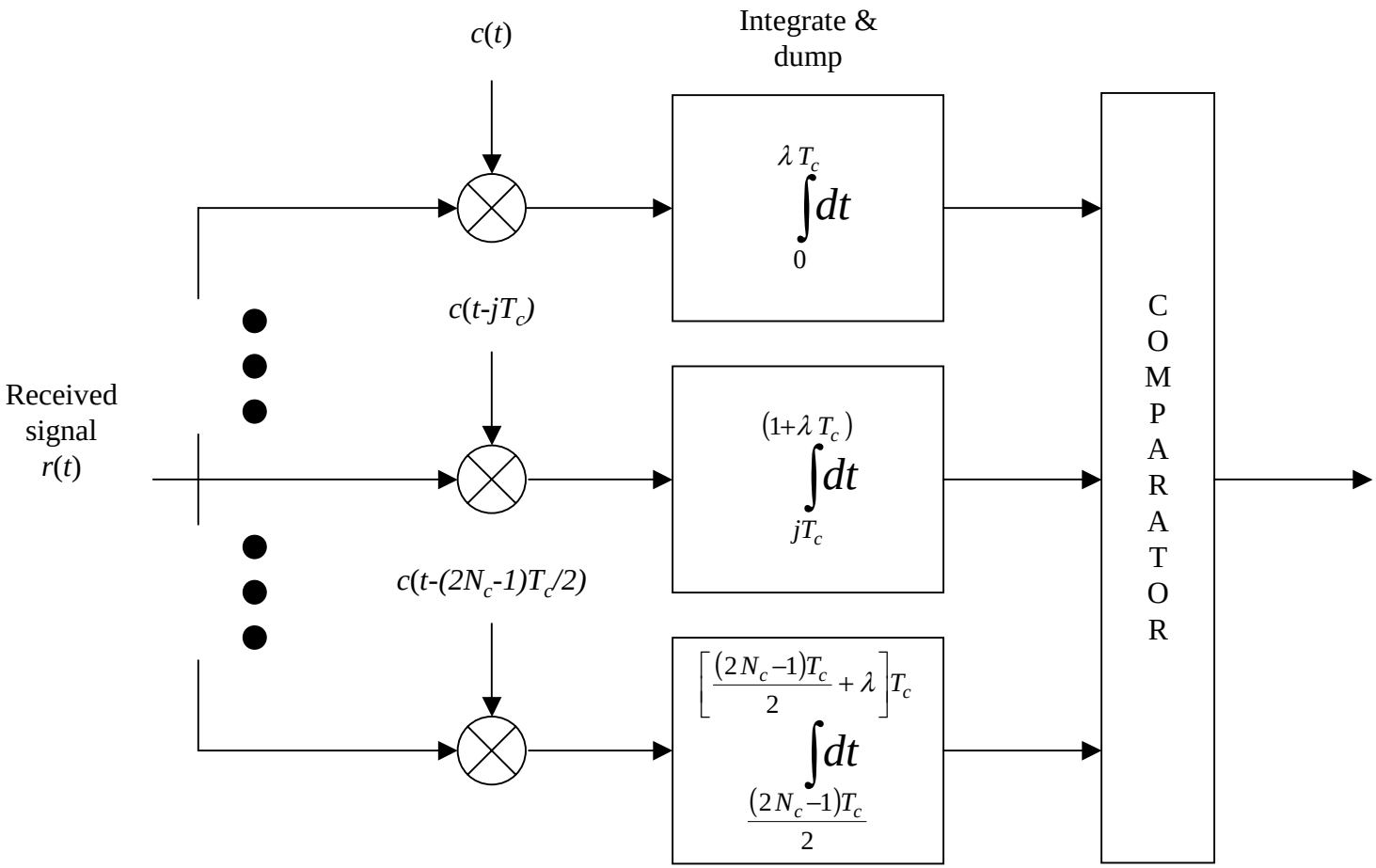
Algorithms

- Channel estimation = estimate attenuation & delay introduced by wireless channel.
  - Sliding correlator.
  - Maximum likelihood (ML).
- Detection = estimate information bits.
  - Code-matched filter.
  - Minimum mean squared error (MMSE).

Sliding Correlator

- Simplest option for blind channel estimation.
- Used in current systems.
- Received signal correlated with locally generated copies of spreading code spaced by fraction of chip.

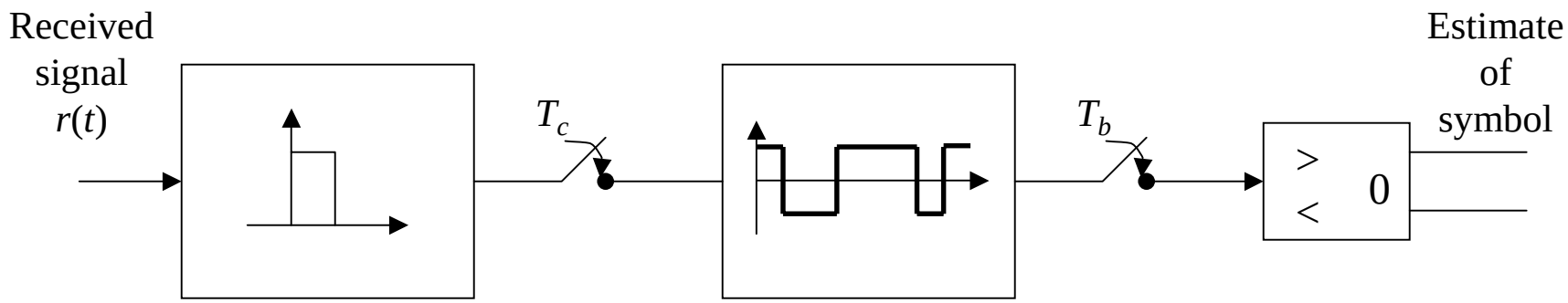
Sliding Correlator (cont.)



Code-Matched Filter

- Simplest way of estimating bits using minimum information/processing.
- Success hinges on orthogonality of codes.
- Chip-matched filter output correlated with locally generated copy of code.

Code-Matched Filter (cont.)



Implementation

- Implementing algorithms on TI ‘5410 DSP.
  - Hoping to gain insight into algorithm/architecture combination.
  - Using ‘5410 because realistic DSP for handset architecture: small size, low power, low cost.

Implementation: ‘5410 DSP

- Three data busses & one program bus.
- 40-bit ALU, 2 40-bit ACC’s & 40-bit barrel shifter.
- 17x17 MAC unit.
- 8 data address registers, 2 address arithmetic units.
- Exponent encoder.
- 64 K x 16-Bit On-chip RAM.
- On-chip software (e.g. FFT, companding, …).

Implementation: AMIDALA

- '5410 DSP.
- Two flash ports.
- LCD display.
- Keyboard input.
- High-quality stereo codec.
- JTAG interface.

Implementation: DSP Software Design Flow

1. Synthesize test data in Matlab/Simulink.
2. Implement algorithm in Matlab/Simulink.
3. Implement algorithm in floating-point C on host.
4. Implement algorithm in fixed-point C on host.
5. Compile floating/fixed-point C on target.
6. Implement fixed-point algorithm core in C-callable assembly language on target.

Implementation: Data Synthesis

- Matlab script generates handset receiver data.
  - Input command file; output data for DSP.
  - Synchronous & equal transmit powers.
  - Walsh spreading codes.
- Can band limit transmit signal
- FIR filter used for channel attenuation & delay.
- Can quantize receive signal to any word length.

Data Synthesis: Input/Output Files

```
#
#  Example command file.
#
1e3         # data rate
8           # chips/symbol
32e3        # samp. freq.
1.0         # xmit power of user k
1000        # num. bits to xmit
4           # num. of users
FALSE       # band limit xmit
1.0         # channel gain
7.8125e-5   # channel delay
0.1         # channel AWGN
TRUE        # scale/quantize
```

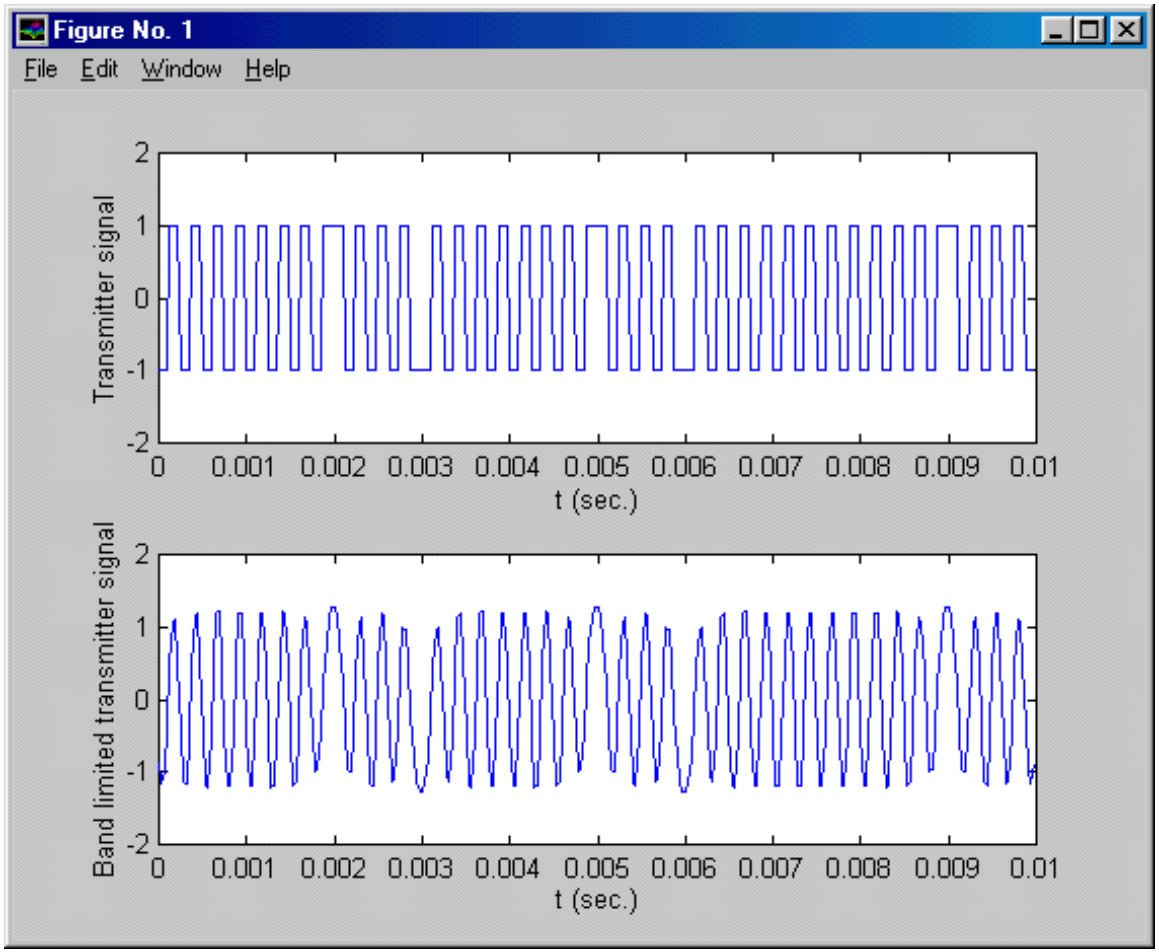
```
CODES:
-----
  1 -1  1 -1  1 -1  1 -1


BITS:
----
  1  0  1  0  1  1  0  0  1  0


TRANSMIT SIGNAL: (320 points)
---------------
 1.000000
 1.000000
 1.000000
-1.000000
```

Data Synthesis: Band limiting transmit signal

- $f_{cut} = f_{chip} + 0.1 f_N$
- 50^{th} order filter = remez()
- zero-pad input to remove delay (25 samples)

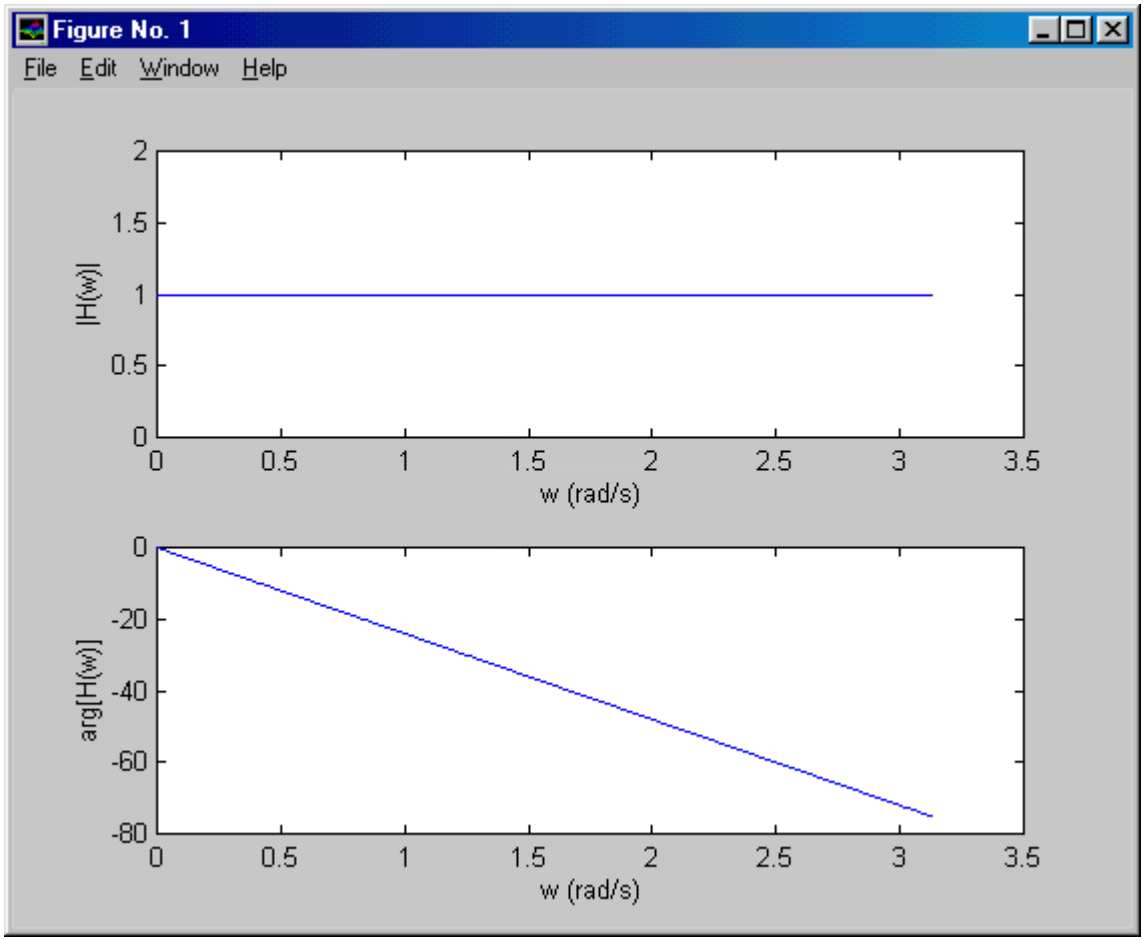


Data Synthesis: Channel Delay

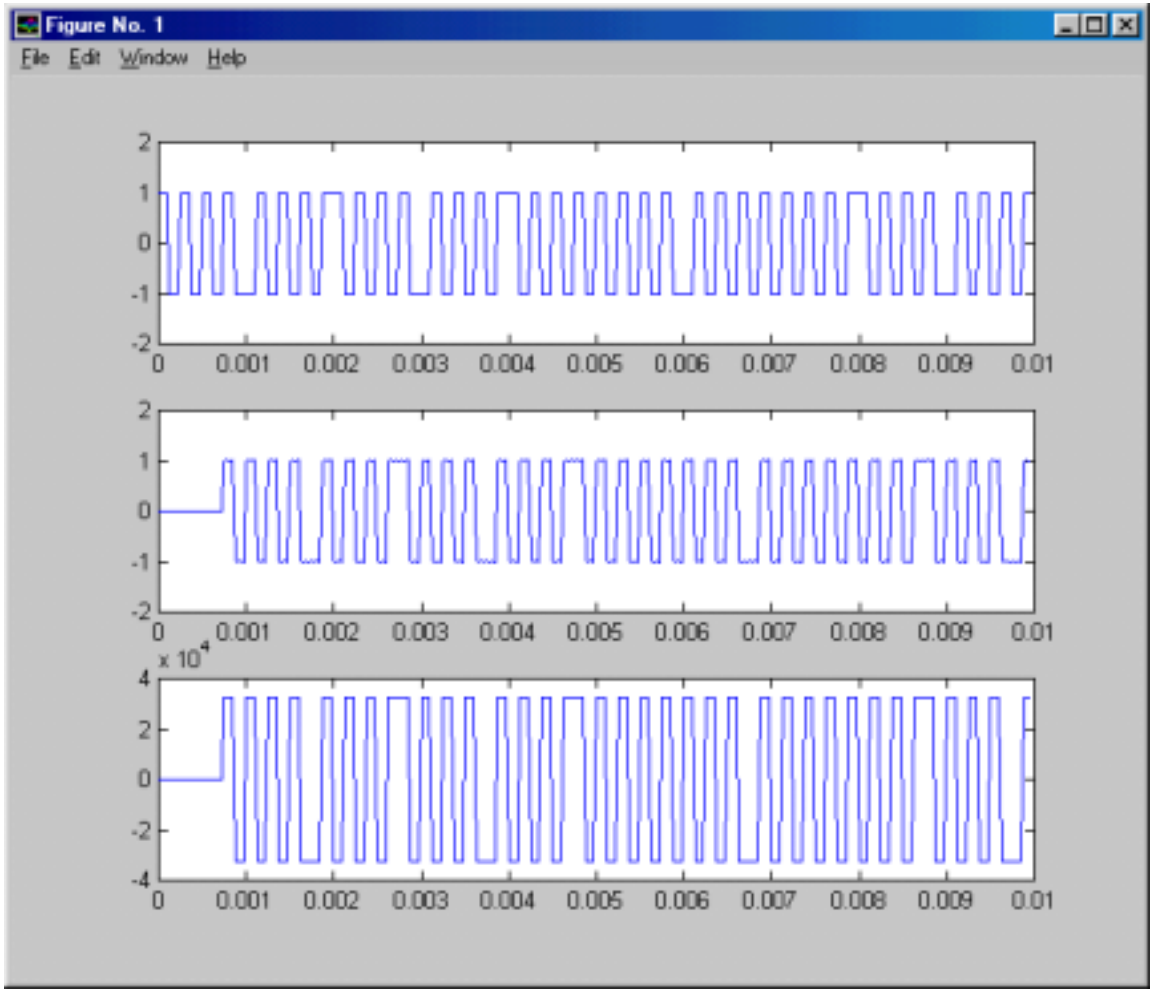
- Delay spec'd seconds in command file.
- Delay generated w/ FIR filter: generate desired frequency resp. & estimate coefs.
  - Unit magnitude response.
  - Linear phase w/ slope $-\tau$.

Data Synthesis: Channel Delay (cont.)

- $H(e^{j\omega}) = \left|H\left(e^{j\omega}\right)\right| e^{\Phi\left(e^{j\omega}\right)}$
- inv_freqz()
- 7.5×10^{-4} sec. $\Rightarrow$ 24 samples @ 32 kHz



Data Synthesis: Channel Delay (cont.)



Data Synthesis: Quantization

$$1 \times s = 2^{N-1} - 1$$

$$-1 \times s = -2^{N-1}$$

$$\Rightarrow \quad s = \frac{2^{N-1} - 1}{2^{N-1}} + \delta$$

& take floor

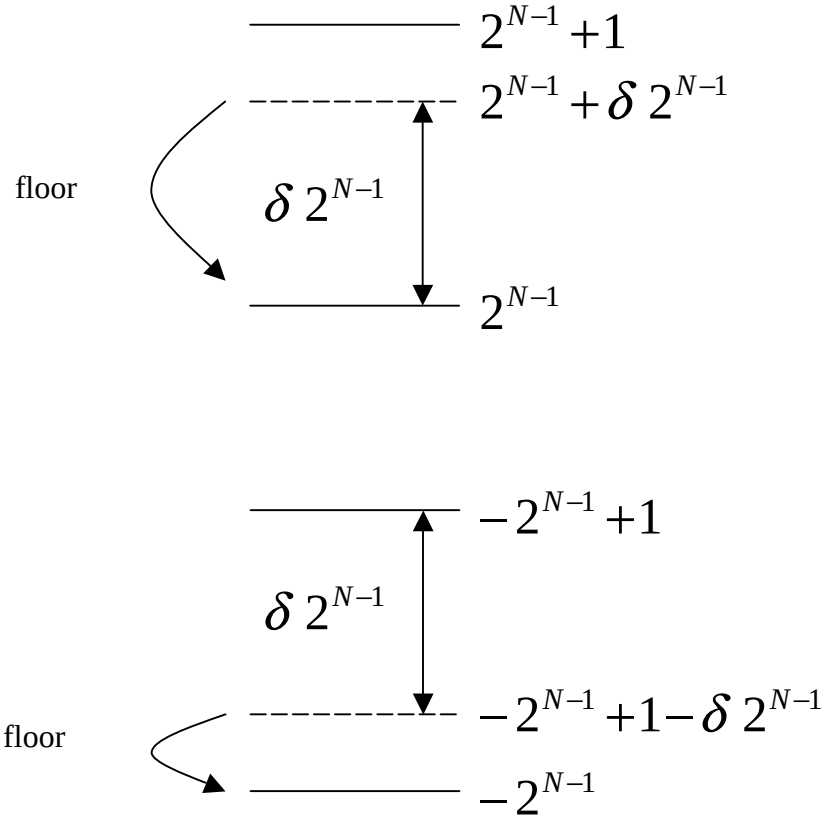
Data Synthesis: Quantization (cont.)

$$\delta = \frac{1}{2^N}$$

$$\alpha = \frac{\left(2^{N-1} - 1\right)}{2^{N-1}} + \delta$$

$$s_{tmp} = \frac{\alpha\, s}{\max\left(\left|s\right|\right)}$$

$$s_{fix} = \left\lfloor 2^{N-1} s_{tmp} \right\rfloor$$



N = word length

Implementation: Sliding Correlator

- Implemented in floating-point C on host.
  - Can accommodate any delay resolution to w/in single sample period.
- Have compiled on target: init. does not operate correctly.
  - Floating-point inaccuracies in ’54x RTS.
  - Solutions: move code to host, rewrite init. routine, rewrite RTS.

Implementation: Code-Matched Filter

- Implemented in floating-point C on host.

- Have compiled on target: operates correctly.
  - Host/target communications operate.
  - Init. routines operate.

Short-Term Goals

- Repair FltPt-SC on target.
- Implement algorithm cores in FxdPt C/assembly.
  - Take MIP's counts.
  - Investigate quantization.
- Add optional pulse shaping to data-generation script.

Long-Term Goals

- Understand/implement ML algorithm for channel estimation.
- Understand/implement MMSE algorithm for detection.
- Continue investigating real-time constraints & quantization error issues.