# ECE 5654 – Advanced Digital Communications

Instructor: R. McGwier

Synchronization

Spring 2014

# Learning Objectives

- After completing this lecture, the student should be able to
  - Define synchronization and list the parameters of a communications signal that must be estimated for proper data demodulation
  - Describe basic circuits for frequency, phase and timing estimation

# What is Synchronization?

- Synchronization is the alignment of local clocks and oscillators with the timing/frequency/phase of the incoming signal
- There are four basic synchronization aspects
  - Frame timing
  - Symbol timing
  - Carrier frequency
  - Carrier phase

# System Model

- Assume that the signal has been mixed down to baseband using a quadrature (i.e., sine and cosine) carriers at frequency $f_c$
- Assuming linear modulation (e.g., PSK or QAM) the received signal can modeled as

$$r(t) = e^{j(2\pi\nu t+\theta)} \sum_k s_k g\left(t - kT_s + \tau\right)$$

$\nu$ – Frequency offset
$\theta$ - Phase Offset
$\tau$ – Time offset

$s_k$ - data symbols
$g(t)$ - pulse shape

# Parameter Estimation

- We are interested in the data symbols $s_k$ (and ultimately the bits that define the symbols)
- However, in order to estimate the symbols we must also estimate the unknown timing parameters (often called nuisance parameters)

$$\mathbf{\Phi} = \begin{bmatrix} \nu \\ \theta \\ \tau \end{bmatrix}$$

- The MAP estimate of the parameter vector $\Phi$ is the estimate which maximizes

$$p\left(\mathbf{\Phi} \middle| \mathbf{r}\right) = \frac{p\left(\mathbf{r} \middle| \mathbf{\Phi}\right) p\left(\mathbf{\Phi}\right)}{p\left(\mathbf{r}\right)}$$

# Maximum Likelihood Estimation

- Typically the distribution of the unknown parameter is unknown and assumed to be uniform over some range
- In this case

$$p\left(\mathbf{\Phi}\right|\mathbf{r}) \sim p\left(\mathbf{r}\right|\mathbf{\Phi})$$

- Thus, we maximize

$$p\left(\mathbf{r}\right|\mathbf{\Phi})$$

- This known as the ML estimate

# ML Estimate

- Assuming Gaussian noise

$$p\left(\mathbf{r}\middle|\,\boldsymbol{\Phi}\right) = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^{K} \exp\left\{-\sum_{k=1}^{K}\frac{\left[r_k - s_k(\Phi)\right]^2}{2\sigma^2}\right\}$$

$$r_k = \int_{T_o} r(t)\phi_k(t)dt$$

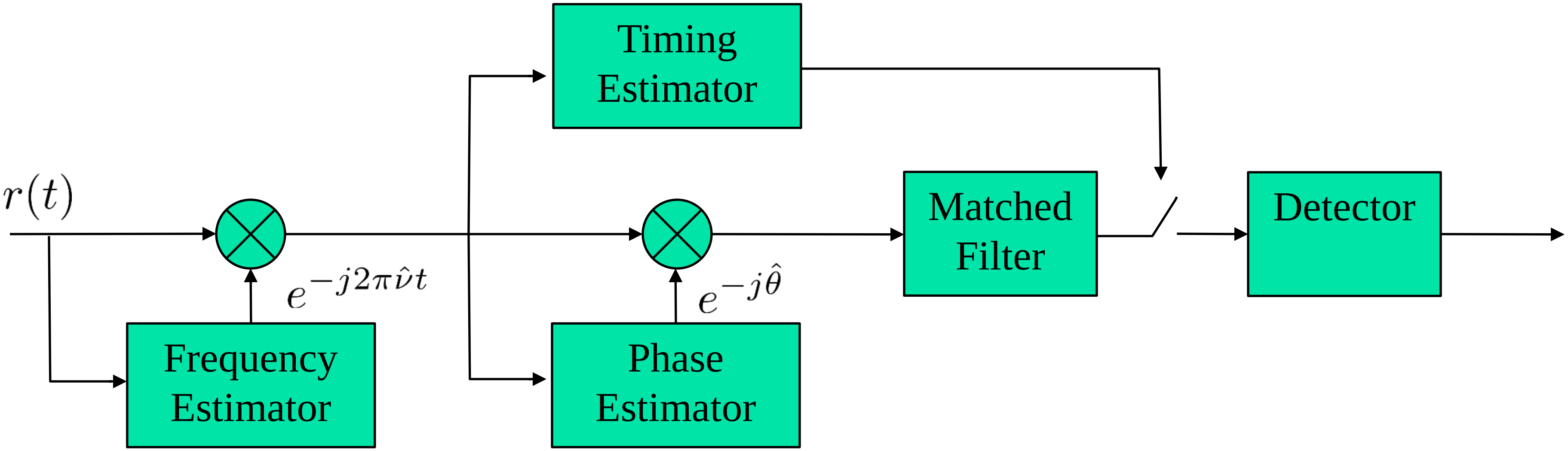
$$s_k = \int_{T_o} s(t,\Phi)\phi_k(t)dt$$

Appropriate basis functions

Observation interval

# Practical Estimator

- The ML estimate is typically too complex to implement
- However, if we separate the parameters and estimate them separately, we can approximate the ML approach

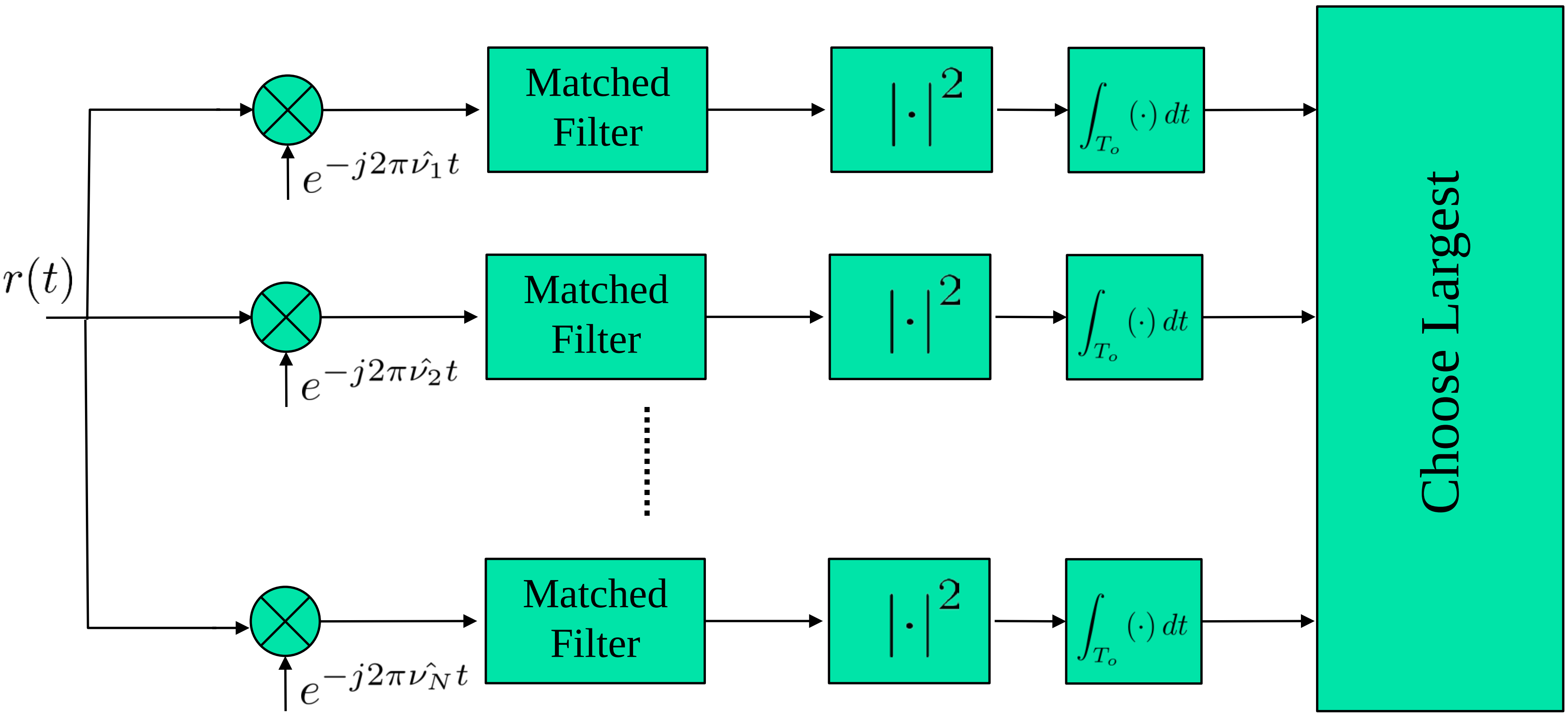


# Burst Mode vs. Continuous Mode

- Burst mode – a packet of symbols is received and processed at once. A single estimate of the unknown parameters is created (assumes that they are not time-varying over the burst)
  - You should assume this in your project
  - This is the approach we will examine here
- Continuous mode – a continuous stream of symbols is received. An initial estimate of the parameters is created, but then the values must be continuously updated (known as tracking)

# Frequency Estimator

- Assume that timing and phase are unknown



# Complexity Tradeoff

- Obviously the accuracy is limited by the number of candidate offsets examined
- The complexity is directly related to the number of offsets examined
- Thus, there is a trade-off between accuracy and complexity
- One approach is the find a coarse estimate and then progressively find finer estimates
- If timing and data are known (through pilot sequences), the FFT can be used to estimate frequency offset

# ML Phase Estimator

- Estimator with known timing and data

$$\hat{\theta} = \arg\left\{\sum_{k=0}^{L-1} s_k^* x(k)\right\}$$

- Where *L* is the number of symbols observed, $s_k$ are the known complex symbols and

$$x(t) = \left[r(t)e^{-j2\pi\nu t}\right] \otimes g(-t)$$

# Decision Directed Phase Estimator

- The previous phase estimator can be used with a pilot sequence.
- In order to use the entire data sequence to estimate the phase, we make decisions on the data
- Procedure:

1. Use the pilot sequence to estimate the phase
2. Use this estimate to create data estimates for the next *L* symbols
3. Using those decisions create a new phase estimate using the "known data" estimator

# Impact of residual frequency offset

- If frequency offset hasn't been completely removed, the phase offset will be time-varying
- The residual frequency offset will limit the number of symbols over which the phase offset can be estimated
- This creates a trade-off between robustness to noise (increases with $L$) and robustness to frequency offset (decreases with $L$ )

# Frame Timing Recovery

- In order to properly send and receive data, we put the data in packets.
- Within each packet there are different fields that need to be found (including the code word boundaries)
- The frame boundaries/fields can be found by including a known synch word at a known location

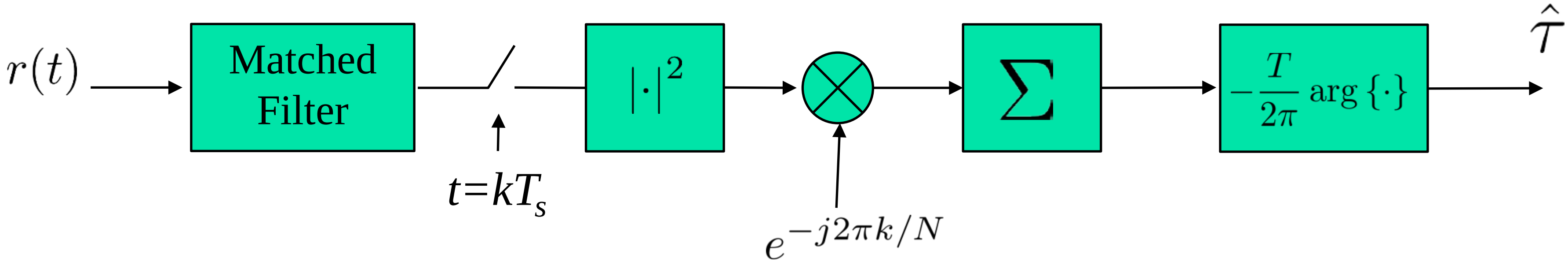
$$\left| \int_{T_{synch}} r(t) w_{synch}(t - \hat{\tau}) dt \right|^2$$

- The estimate of $\tau$ that maximizes this output indicates the correct synch sequence location where

$$w_{synch}(t) = \sum_{i=0}^{N_{synch}} s_i g(t - iT_s)$$

# Symbol Timing Estimation

- Assume that the data symbols and phase offset are unknown but that the frequency offset is known and removed
- The Oerder and Meyr algorithm estimates the delay by



# Summary

- In this lecture we have examined a few basic estimators for frequency, phase and timing
- The joint ML estimator is too difficult so we typically estimate each parameter separately and serially
- In general it is easiest to estimate and remove frequency first
- It is probably easiest to estimate the symbol timing second
- Phase estimation is typically last (but not always)