展頻通訊
(Spread Spectrum Communications)

國立清華大學電機系暨通訊工程研究所
蔡育仁
台達館 821 室
Tel: 62210
E-mail: yrtsai@ee.nthu.edu.tw

Chapter 5
Initial Synchronization

Contents

- Introduction
- Optimum Synchronizer
- Serial-Search Synchronization
- Generalized Analysis of Average Synchronization Time
- Synchronization Using Matched Filter

Introduction

- **Serial search** is a widely used technique for initial synchronization:
  - **Search serially** through **all** potential code phases until the correct phase is identified
  - Each reference phase is evaluated by attempting to **despread** the received signal
  - If the estimated code phase is **correct**, despreading will occur and **will be sensed**
  - If the estimated code phase is **incorrect**, the received signal will not be despread
    - Then, the reference waveform will be stepped to a **new phase** for evaluation

Introduction (Cont.)

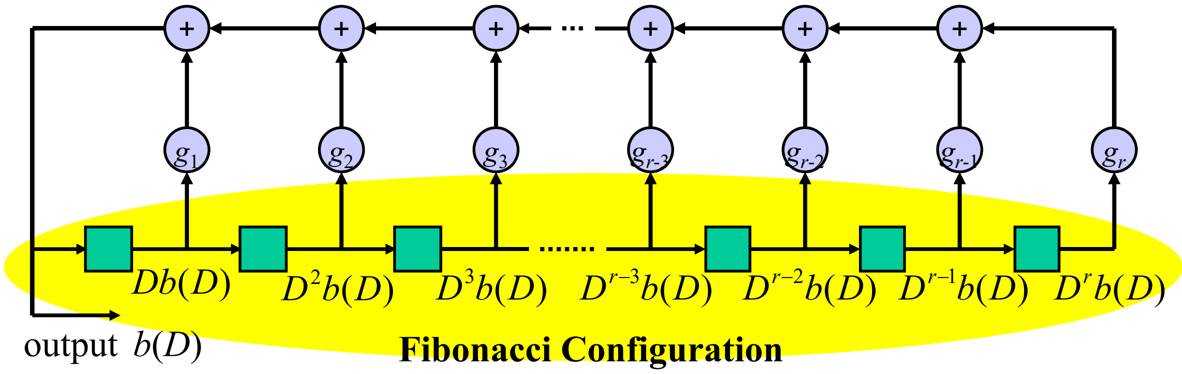
- The probability that the detector indicates that the trial phase is correct when, **in fact, it is not**
  - The probability of **false alarm** P_{fa}
- The probability that the detector indicates that the trial phase is correct when **it is indeed correct**
  - The probability of **correct detection** P_d
- P_{fa} and P_d are a function of **evaluation (integration) time** and **signal-to-noise ratio**

Serial Search／Matched Filter Detection

- **Serial-search** acquisition schemes are referred to as **low-decision-rate detectors**
  - A **large** number of spreading code symbols must be received to make a correct/incorrect decision
- **Matched filter detection** is a highly efficient method of initial synchronization
  - When **a particular sequence of code symbols** is received, the matched filter will output a pulse
  - Require matched filters with extremely **large time-bandwidth products**

Rapid Acquisition

- **Rapid acquisition** by sequential estimation (**RASE**):
  - The DSSS system should employ BPSK spreading
  - The spreading code is an *m*-sequence generated using the **generator of Fig. 3-6**
  - **Demodulate** the received symbol **(chip)** stream directly and **load** the shift register generator with this symbol stream



Fibonacci Configuration

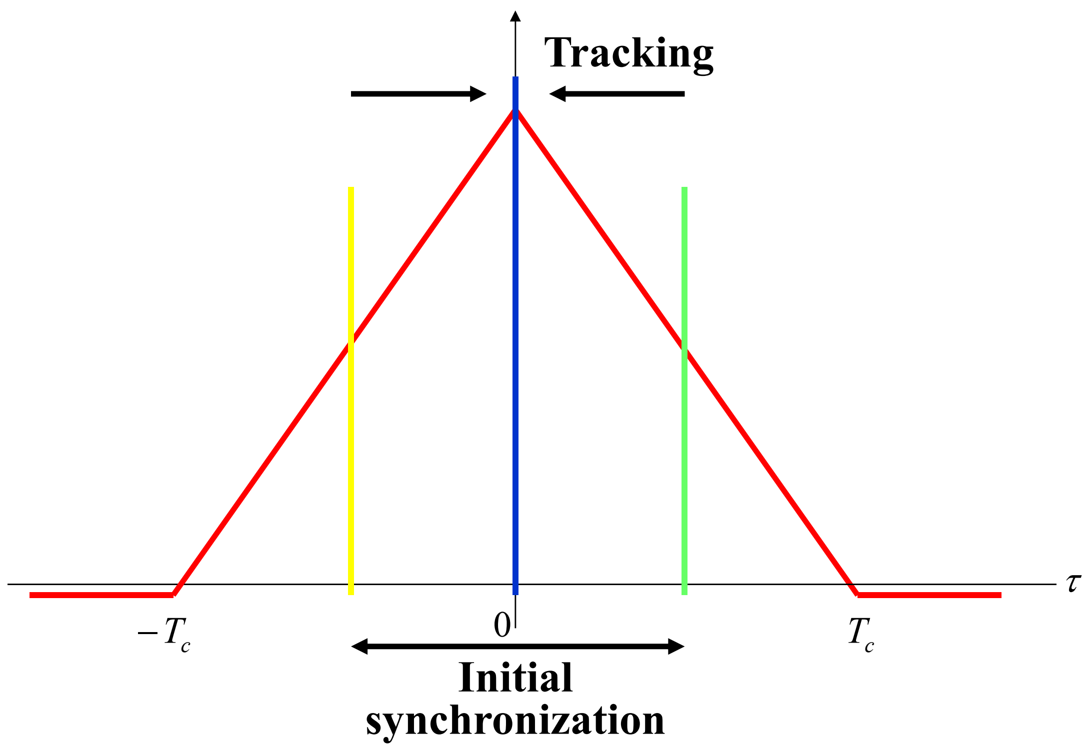
Rapid Acquisition (Cont.)

- **At very high SNR**: the demodulated symbols **(chips)** are correct
  - ⇒ The shift register initial condition is also correct
- **At reduced SNR**: the shift register load may not be correct
  - ⇒ Additional loads will have to be attempted

Initial Synchronization Complexity

- The number of **code phases** and **frequencies** that must be evaluated to obtain initial synchronization is proportional to:
  - **Propagation delay uncertainty**: expressed in spreading code chips
    - Induce **code phase** uncertainty
  - **Relative dynamics** of the transmitter and the receiver: Doppler effect
    - Induce **frequency** and **code phase** uncertainty
- The initial synchronization techniques are generally capable of determining the received spreading code phase to within an accuracy of ±½ **to ± ¼ of a chip**

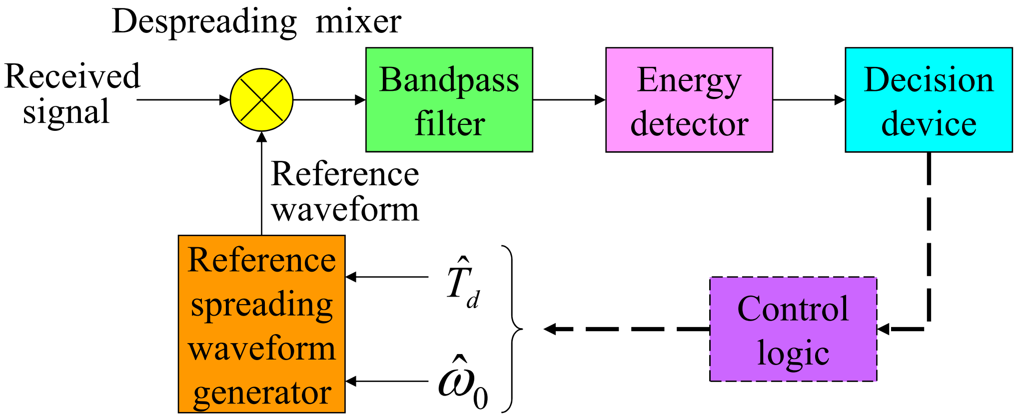
Initial Synchronization Complexity (Cont.)



Optimum Synchronizer

Functional Blocks of Synchronizer

- Assume that the **phase uncertainty** is bounded to a range of ΔT seconds and the **frequency uncertainty** is bounded to a range of $\Delta\Omega$ rad/sec
- A simplified conceptual block diagram to evaluate a **single** reference phase and frequency is



Functional Blocks of Synchronizer (Cont.)

- If the phase and frequency of the receiver-generated replica of the spreading waveform are correct:
  - The energy detector will detect the presence of the signal
- The **control logic** is used to select the values of $\hat{T}_d$ and $\hat{\omega}$ for evaluation

Synchronization Detection

- Suppose that the received signal $r(t)$ is a carrier which is BPSK modulated by an m-sequence waveform $c(t - T_d)$ plus AWGN

$$r(t)=c(t-T_d)\cos(\omega_0 t+\phi)+n(t)$$

  - where ω_0 is the carrier frequency and ϕ is the carrier phase
- The reference waveform is:

$$a(t)=c(t-\hat{T}_d)\cos[(\hat{\omega}_0+\omega_{IF})t]$$

- The despreader output (difference frequency term) is:

$$x(t)=c(t-T_d)c(t-\hat{T}_d)\cos\left[\omega_{IF}t+(\hat{\omega}_0-\omega_0)t-\varphi\right]+n'(t)$$

Synchronization Detection (Cont.)

- The power spectrum of $c(t-T_d)c(t-\hat{T}_d)$ is: $\varepsilon = T_d - \hat{T}_d$

$$S_b(f,\varepsilon) = \left[1-\left(1+\frac{1}{N}\right)\frac{|\varepsilon|}{T_c}\right]^2 \delta(f)$$
$$+\left(1+\frac{1}{N}\right)\left(\frac{|\varepsilon|}{T_c}\right)^2 \sum_{n=-\infty,n\neq 0}^{\infty} \operatorname{sinc}^2\left(nf_c|\varepsilon|\right)\delta\left(f-nf_c\right)$$
$$+\frac{N+1}{N^2}\left(\frac{|\varepsilon|}{T_c}\right)^2 \sum_{m=-\infty,m\neq 0}^{\infty} \operatorname{sinc}^2\left(\frac{mf_c}{N}|\varepsilon|\right)\delta\left(f-\frac{mf_c}{N}\right)$$

- For $\varepsilon = 0$: a single spectral line at **zero frequency**
- For $\varepsilon = T_c$: a phase-shifted replica of $c(t)$, so $S_b(f, \varepsilon) = S_c(f)$
- For $\varepsilon \neq 0$ or T_c: significantly **wider** than the spectrum of $c(t)$
  - $b(t,\varepsilon) = c(t-T_d)c(t-\hat{T}_d)$ **transits more rapidly** than $c(t)$

Synchronization Detection (Cont.)

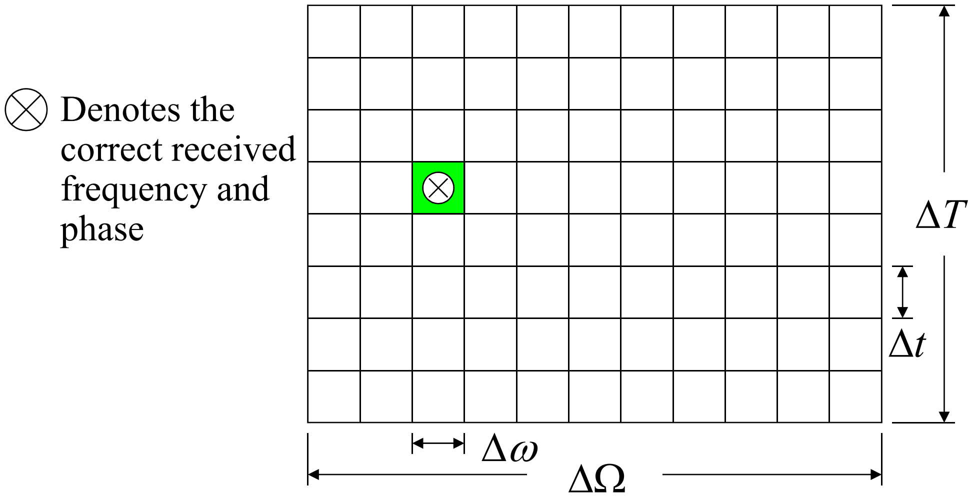
- The power spectrum of $c(t-T_d)c(t-\hat{T}_d)$ contains an impulse at zero frequency which corresponds to a single spectral line
  - at $\omega = \omega_{IF} + \hat{\omega}_0 - \omega_0$ and with power

$$R_c^2(\tau),\ \tau = \hat{T}_d - T_d$$

  - If τ is **sufficiently small** (large enough power) and $\hat{\omega}_0 - \omega_0$ is not larger than the bandwidth of the bandpass filter:
    - This component will be sensed by the energy detector
- The phase/frequency uncertainty region is assumed to be $\Delta\Omega\times\Delta T$
  - Subdivided the region into some smaller cells $\Delta\omega\times\Delta t$
  - A single test in each cell will be sufficient to determine the correct received phase and frequency

Optimum Synchronizer

- An **optimum** initial synchronization system (in the sense that it achieves synchronization **in the minimum possible time**):
  - It evaluates **all cells** within a possible range **simultaneously**



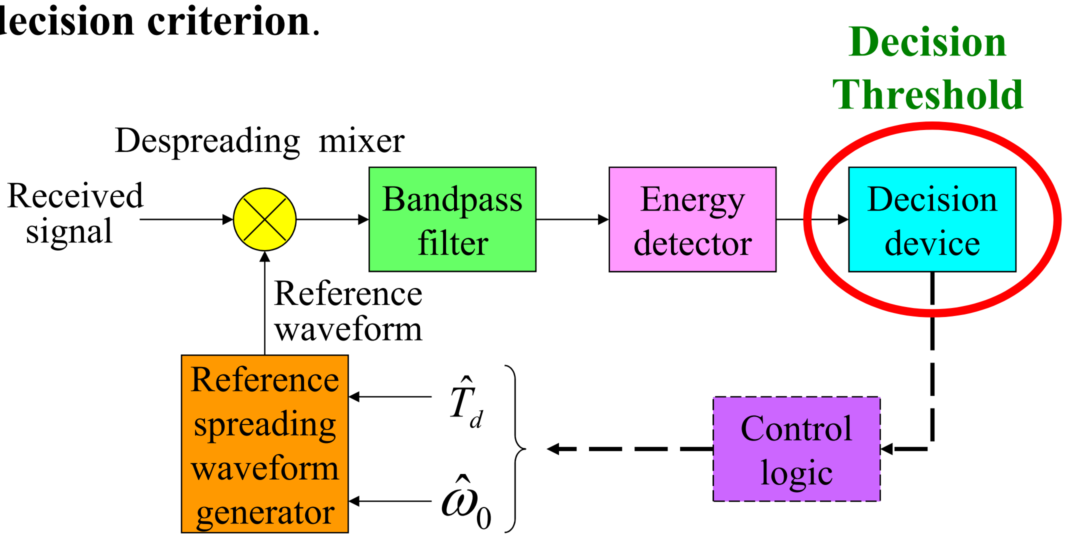
Optimum Synchronizer (Cont.)

- An **minimum-acquisition-time** system requires a evaluating subsystem for every phase/frequency cell
  - It is **not optimum** in a **minimum-hardware sense**
  - It is rarely implemented because of its hardware complexity
- An initial synchronization system is designed to achieve a **compromise** between acquisition time and reasonable hardware complexity
- Synchronization time is proportional to the **total number of cells** that must be evaluated
  - It makes no difference whether these cells were all generated from frequency uncertainty or all from phase uncertainty or a combination

Neyman–Pearson Hypothesis Testing Concept

Binary Hypothesis Testing Problem

- Binary hypothesis testing problem: Either one or the other hypothesis is true.
- The probabilities of the decision errors depend on the selected **decision criterion**.



Acquisition Problem

- For acquiring synchronization, the problem is whether the synchronization is achieved or not.
- Define hypothesis $\boldsymbol{H_1}$ as synchronization being achieved and $\boldsymbol{H_0}$ as synchronization being absent.
- Assumes that the process $r(t)$ is observed in some time interval.
  - It also assumes that the **conditional probability distributions** of the process $r(t)$ are **known** for the observations under the two hypotheses.
  - The distribution is due to the existence of **noise/interference**
- There are two kinds of **decision error probabilities**:
  - The probability to accept hypothesis $\boldsymbol{H_1}$ when $\boldsymbol{H_0}$ is true
  - The probability to accept hypothesis $\boldsymbol{H_0}$ when $\boldsymbol{H_1}$ is true

Binary Hypothesis Testing

- When the hypothesis $\boldsymbol{H_1}$ is desired (acquire synchronization)
  - The probability of accepting hypothesis $\boldsymbol{H_1}$ when $\boldsymbol{H_0}$ is true is called the **false alarm probability** P_{fa}.
  - The probability of accepting hypothesis $\boldsymbol{H_0}$ when $\boldsymbol{H_1}$ is true is called the **probability of a miss** P_{miss}.
- The **probability of detection** is denoted as $P_d = 1 - P_{miss}$.
- In general, we should like to make P_{fa} as **low** as possible and P_d as **high** as possible (P_{miss} as **low** as possible).
  - For most problems of practical importance these are **conflicting** objectives.
  - An obvious approach is to constrain one of the probabilities and to maximize (or to minimize) the other.

Neyman–Pearson Hypothesis Testing

- The **Neyman–Pearson** criterion maximizes P_d (or minimizes P_{miss}) under the constraint $P_{fa} \le \alpha$, where α is a pre-chosen constant.
- The solution is obtained easily by using **Lagrange multipliers**.
- We construct the function

$p_0(z)$: the pdf under $\boldsymbol{H_0}$
$p_1(z)$: the pdf under $\boldsymbol{H_1}$

$$\Phi = P_{miss} + \lambda\left(P_{fa} - \alpha\right)$$
$$= \int_{Z_0} p_1(z)\, dz + \lambda\left[\int_{Z_1} p_0(z)\, dz - \alpha\right]$$

- The total observation space Z is divided into two parts, Z_0 and Z_1 (The space is divided by the **decision threshold**)
  - When an observation falls into Z_0: we say $\boldsymbol{H_0}$
  - When an observation falls into Z_1: we say $\boldsymbol{H_1}$

Neyman–Pearson Hypothesis Testing (Cont.)

- If $P_{fa} = \alpha$, then minimizing Φ will minimize P_{miss}.
  - Because $\Phi = P_{miss} + \lambda(P_{fa} - \alpha)$
- Notice that $\Phi = \int_{Z_0} p_1(z)\, dz + \lambda\left[\int_{Z_1} p_0(z)\, dz - \alpha\right]$

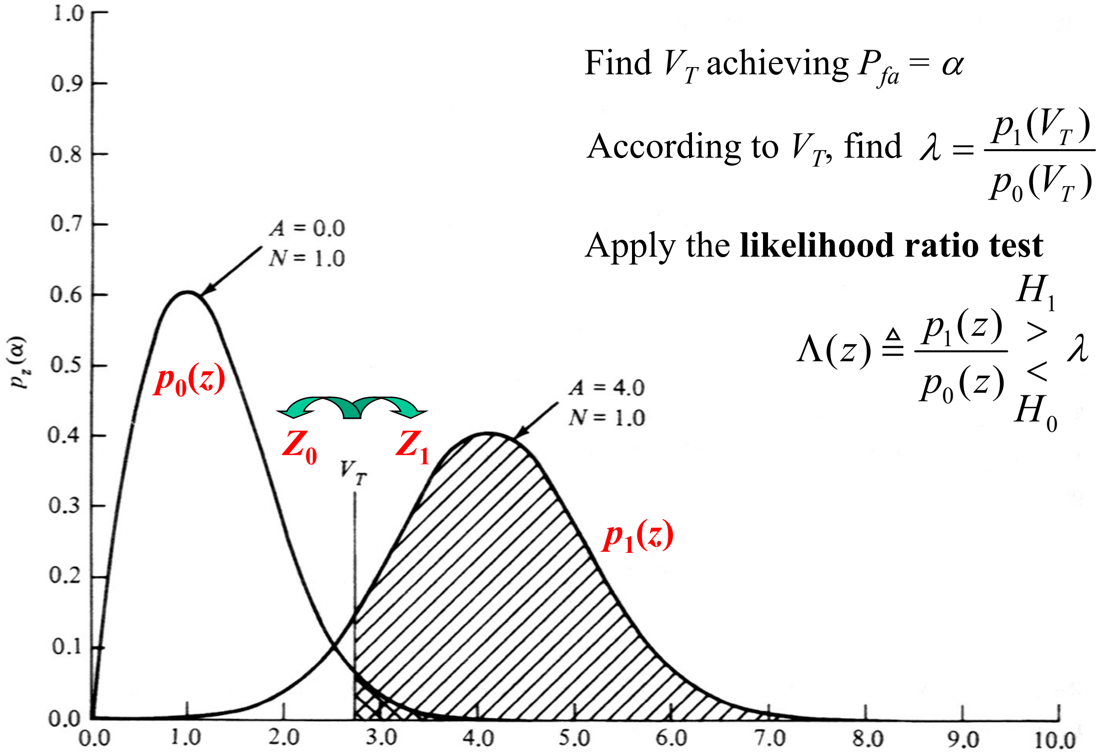
$$= \int_{Z_0} \underline{\left[p_1(z) - \lambda p_0(z)\right]}\, dz + \lambda\left(1-\alpha\right).$$

- For a positive λ, to minimize Φ, we assign a point (a received observation) z to Z_0 only when $p_1(z) - \lambda p_0(z)$ is negative.
  - This is equivalent to the **likelihood ratio test**

$$H_0 : \Lambda(z) \triangleq \frac{p_1(z)}{p_0(z)} < \lambda; \quad H_1 : \Lambda(z) \triangleq \frac{p_1(z)}{p_0(z)} \ge \lambda$$

- The threshold λ depends on the constraint on $P_{fa} = \alpha$.
  - Find V_T corresponding to λ such that $P_{fa} = \int_{V_T}^{\infty} p_0(z)\, dz = \alpha$

Neyman–Pearson Hypothesis Testing (Cont.)



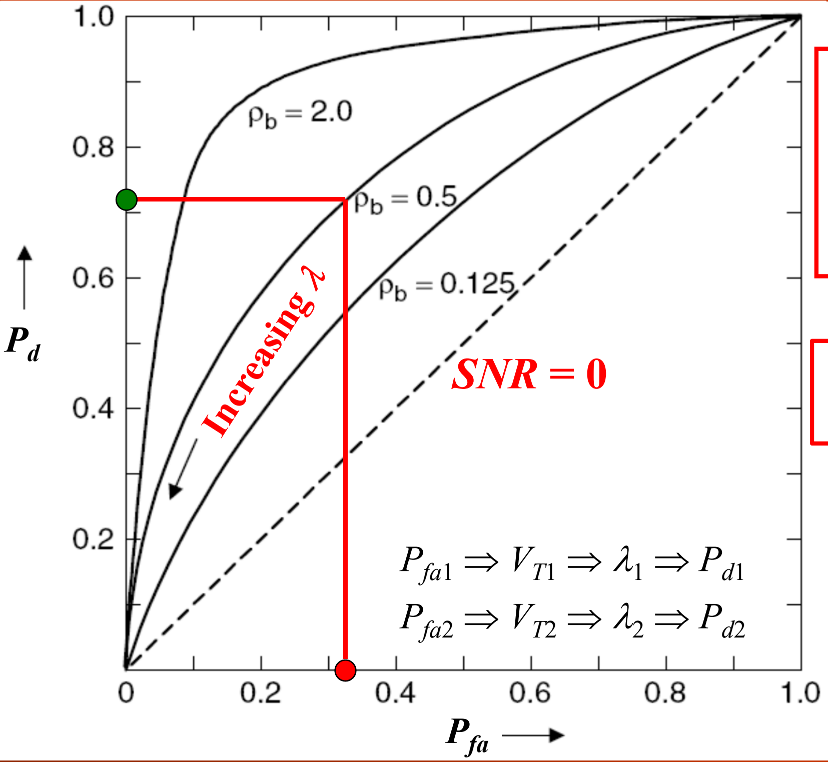
Find V_T achieving $P_{fa} = \alpha$

According to V_T, find $\lambda = \dfrac{p_1(V_T)}{p_0(V_T)}$

Apply the **likelihood ratio test**

$$\Lambda(z) \triangleq \frac{p_1(z)}{p_0(z)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

Neyman–Pearson Hypothesis Testing (Cont.)



When $P_{fa1} < P_{fa2}$
$\Rightarrow V_{T1} > V_{T2}$
$\Rightarrow \lambda_1 > \lambda_2$
$\Rightarrow P_{d1} < P_{d2}$

Receiver Operating Characteristic (ROC)

Source: *Theory of Code Division Multiple Access Communication*, by Kamil Sh. Zigangirov

Serial-Search Synchronization

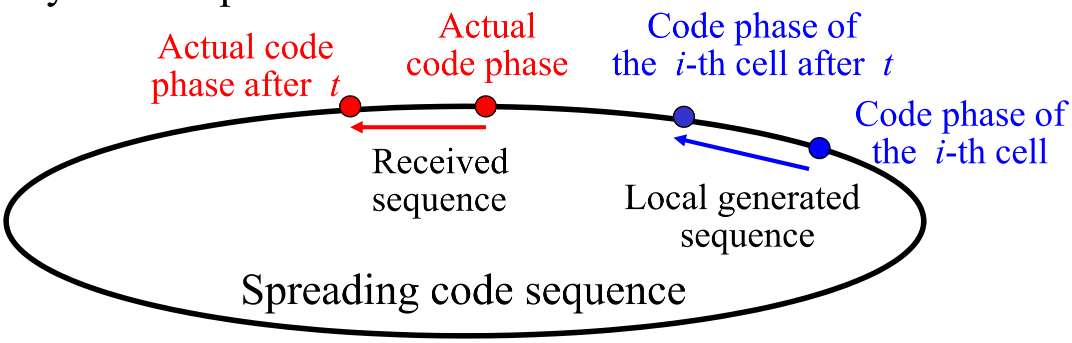
Serial-Search Synchronization

- Serial-search techniques are commonly used in SS synchronization
  - Evaluate the phase/frequency cells **serially** (one after another) until the correct cell is found
- Assume that **no frequency uncertainty** exists and the correct phase is uniformly distributed over the region ΔT
  - A phase step size of Δt seconds is chosen

  $\Rightarrow \Delta T/\Delta t = C$
  - The search will advance through **one cell at a time** until C cells have been evaluated

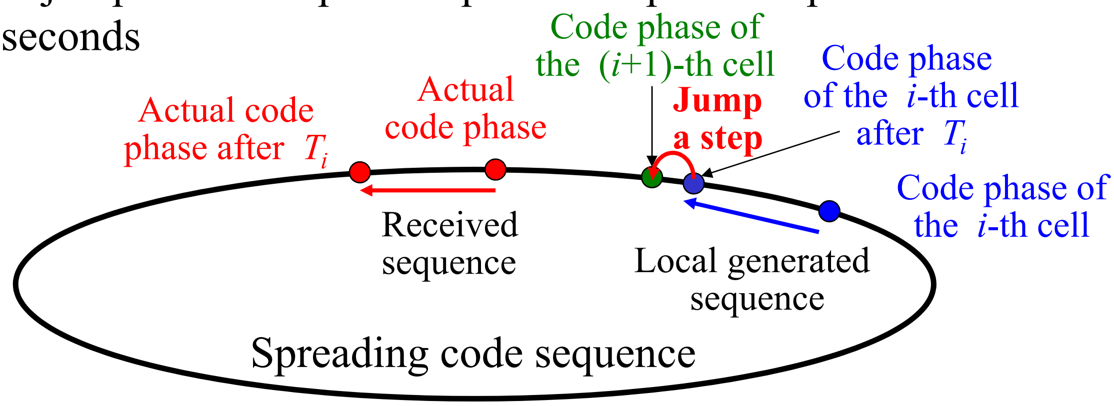
Serial-Search Synchronization (Cont.)

- The transmitted code sequence is generated with a constant rate
  - The phase of the received code sequence moves forward with a constant rate (i.e., the spreading code rate)
- The local generated code sequence is also generated with the same spreading code rate
- The code phase of a specific cell (e.g., the *i*-th cell) changes linearly with respect to time



Serial-Search Synchronization (Cont.)

- After the evaluation time duration of a cell (e.g., T_i) , the synchronizer advances to the next cell for generating the code sequence
  - The local sequence generator jumps to the code phase of the (i + 1)-th cell
  - A jump of a code phase equal to the phase step size of Δt seconds



Synchronization Time

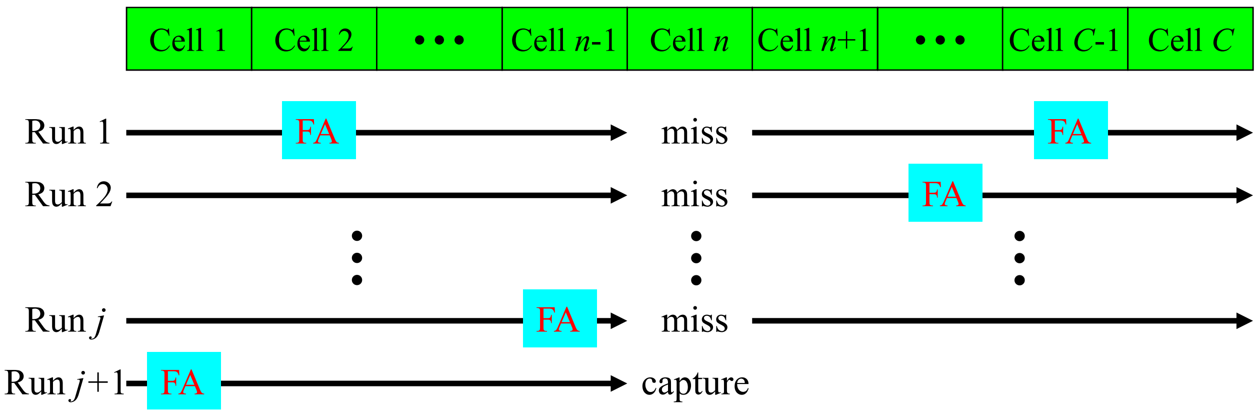
- Define
  - n: a particular location for the **correct** phase cell
  - j: the number of **missed detections** of the correct phase cell
  - k: the number of **false alarms** in all **incorrect** phase cells
- The total synchronization time for a particular event is

$$T(n, j, k) = nT_i + jCT_i + kT_{fa}$$

  - T_i is the (**fixed**) integration time for evaluation of each cell
  - T_{fa} is the time required to reject an incorrect cell when a false alarm occurs
    - T_{fa} may be many times **larger** than T_i

Synchronization Time (Cont.)

- The total number of cells evaluated for this event is $(n + j\,C)$
- The total number of **correct** cells is $(j + 1)$
- The total number of **incorrect** cells is $(n + jC - j - 1)$



Synchronization Time (Cont.)

- The probability of the correct cell being the *n*-th cell is

$$1/C$$

- The probability of j missed detections followed by a correct detection is

$$P_d(1-P_d)^j$$

- The *k* false alarms can occur in any order within the incorrect cells

$$(n+jC-j-1) \triangleq K$$

  – The probability of a particular ordering:

$$P_{fa}^k(1-P_{fa})^{K-k}$$

  – There are $\binom{K}{k}$ orderings of *k* false alarms in *K* cells

Synchronization Time (Cont.)

- The probability of the event (*n, j, k*) is

$$\Pr(n,j,k) = \frac{1}{C}P_d(1-P_d)^j\binom{K}{k}P_{fa}^k(1-P_{fa})^{K-k}$$

- The mean synchronization time is

$$\bar{T}_s = \sum_{n,j,k} T(n,j,k)\Pr(n,j,k)$$

  – The correct cell number *n* can range over (1*, C*)
  – The number of missed detections *j* can range from 0 to ∞
  – The number of false alarms *k* can range over (0*, K*)

Synchronization Time (Cont.)

- The mean synchronization time becomes

$$\bar{T}_s = \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\sum_{k=0}^{K}\left[(n+jC)T_i + kT_{fa}\right]\binom{K}{k}P_{fa}^k(1-P_{fa})^{K-k}P_d(1-P_d)^j$$

$$= \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}(n+jC)T_i\underline{\left[\sum_{k=0}^{K}\binom{K}{k}P_{fa}^k(1-P_{fa})^{K-k}\right]}P_d(1-P_d)^j$$

$$+\frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}T_{fa}\left[\sum_{k=0}^{K}\binom{K}{k}kP_{fa}^k(1-P_{fa})^{K-k}\right]P_d(1-P_d)^j$$

- The first summation over k equals unity:
  - By the identity $\sum_{k=0}^{K}\binom{K}{k}a^k b^{K-k} = (b+a)^K$

    with $a = P_{fa}$ and $b = 1 - P_{fa}$, $b+a = 1.0$

Synchronization Time (Cont.)

- The second summation over k equals KP_{fa}:
  - The **mean** of a discrete random variable which has a binomial distribution
- The mean synchronization time is simplified to

$$\bar{T}_s = \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\left[(n+jC)T_i + KT_{fa}P_{fa}\right]P_d(1-P_d)^j$$

- With $K = n + jC - j - 1$:

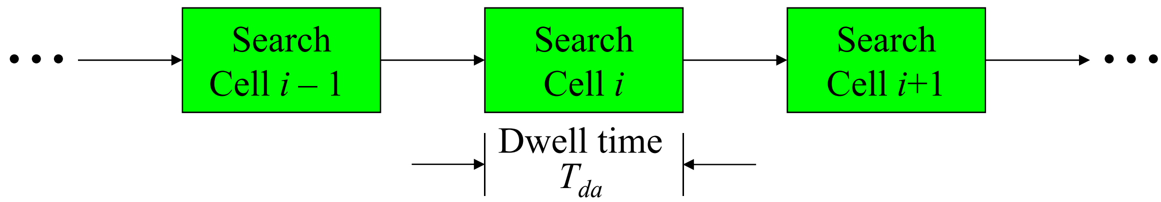
$$\bar{T}_s = \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\left[(n-1)T_{da} + (j+1)T_i + j(C-1)T_{da}\right]P_d(1-P_d)^j$$

  - where $T_{da} = T_i + T_{fa}P_{fa}$ is the average **dwell time** at a incorrect phase cell

Synchronization Time (Cont.)

- The average **dwell time** at a incorrect phase cell
  - When no false alarm occurs, it is T_i (the integration time for evaluation of each cell)
  - When a false alarm occurs, it is $T_i + T_{fa}$ (the time required to reject an incorrect cell)

$$T_{da} = (1 - P_{fa})T_i + P_{fa}(T_i + T_{fa}) = T_i + T_{fa}P_{fa}$$



Synchronization Time (Cont.)

- By using the identities:

$$\sum_{i=1}^{L} i = L\left(\frac{L+1}{2}\right)$$

$$\sum_{i=0}^{\infty} m^i = \left(\frac{1}{1-m}\right)$$

$$\sum_{i=0}^{\infty} i m^i = \frac{m}{(1-m)^2}$$

- The final result of the mean synchronization time is

$$\overline{T}_s = (C-1)T_{da}\left(\frac{2-P_d}{2P_d}\right) + \frac{T_i}{P_d}$$

  - A function of P_{fa} (through the definition of T_{da})

Synchronization Time (Cont.)

- The mean-square value is

$$\sigma_x^2 = E[(x-\bar{x})^2] = E[x^2] - E^2[x]$$

$$\overline{T_s^2} = \sum_{n,j,k} T^2(n,j,k)\Pr[n,j,k]$$

$$\overline{T_s^2} = \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\sum_{k=0}^{K}\left[(n+jC)T_i + kT_{fa}\right]^2 \binom{K}{k} P_{fa}^k (1-P_{fa})^{K-k} P_d (1-P_d)^j$$

- Expanding and grouping like powers of k, we have

$$\overline{T_s^2} = \frac{P_d}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\sum_{k=0}^{K}\left(A_2 k^2 + A_1 k + A_0\right)\binom{K}{k} P_{fa}^k (1-P_{fa})^{K-k} P_d (1-P_d)^j$$

  - where $A_2 = T_{fa}^2;\quad A_1 = 2(n+jC)T_i T_{fa};\quad A_0 = (n+jC)^2 T_i^2;$

Synchronization Time (Cont.)

- The sums over k are the moments of a binomially distributed random variable
  - The first sum is (the second moment):

$$\sum_{k=0}^{K} A_2 \underline{k^2} \binom{K}{k} P_{fa}^k (1-P_{fa})^{K-k} = A_2\left[K^2 P_{fa}^2 + K P_{fa}(1-P_{fa})\right]$$

- The mean-square value is simplified to

$$\overline{T_s^2} = \frac{P_d}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\left\{A_2\left[K^2 P_{fa}^2 + K P_{fa}(1-P_{fa})\right] + A_1\left[K P_{fa}\right] + A_0\right\}(1-P_d)^j$$

Synchronization Time (Cont.)

- With $K = n + jC - j - 1$ and by grouping like power of j, we have

$$\overline{T_s^2} = \frac{P_d}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}\left(B_2 j^2 + B_1 j + B_0\right)(1-P_d)^j$$

  – where

$$B_2 = C^2T_i^2 + 2C(C-1)T_iT_{fa}P_{fa} + (C-1)^2T_{fa}^2P_{fa}^2$$

$$B_1 = 2nCT_i^2 + 2(2Cn - n - C)T_iT_{fa}P_{fa} + 2(n-1)(C-1)T_{fa}^2P_{fa}^2 + (C-1)T_{fa}^2P_{fa}(1-P_{fa})$$

$$B_0 = n^2T_i^2 + 2n(n-1)T_iT_{fa}P_{fa} + (n-1)^2T_{fa}^2P_{fa}^2 + (n-1)T_{fa}^2P_{fa}(1-P_{fa})$$

Synchronization Time (Cont.)

- By using the identities:

$$\sum_{i=1}^{L} i = L\left(\frac{L+1}{2}\right)$$

$$\sum_{i=0}^{\infty} m^i = \left(\frac{1}{1-m}\right)$$

$$\sum_{i=0}^{\infty} i m^i = \frac{m}{(1-m)^2}$$

$$\sum_{i=0}^{\infty} i^2 m^i = \frac{m(1+m)}{(1-m)^3}$$

- The mean mean-square value is simplified to

$$\overline{T_s^2} = \frac{1}{C}\sum_{n=1}^{C}\left[B_2\frac{(1-P_d)(2-P_d)}{P_d^2} + B_1\frac{1-P_d}{P_d} + B_0\right]$$

Synchronization Time (Cont.)

- By grouping like power of n, we have

$$\overline{T_s^2} = \frac{1}{C}\sum_{n=1}^{C}\left(D_2 n^2 + D_1 n + D_0\right)$$

  – where

$$D_2 = \left(T_i + T_{fa}P_{fa}\right)^2$$

$$D_1 = -2T_iT_{fa}P_{fa} - 2T_{fa}^2P_{fa}^2 + T_{fa}^2P_{fa}(1-P_{fa}) + \left[2CT_i^2 + 2(2C-1)T_iT_{fa}P_{fa} + 2(C-1)T_{fa}^2P_{fa}^2\right](1-P_d)/P_d$$

$$D_0 = T_{fa}^2P_{fa}^2 - T_{fa}^2P_{fa}(1-P_{fa}) + \left[-2CT_iT_{fa}P_{fa} - 2(C-1)T_{fa}^2P_{fa}^2 + (C-1)T_{fa}^2P_{fa}^2(1-P_{fa})\right](1-P_d)/P_d + \left[C^2T_i^2\,2(2C-1)T_iT_{fa}P_{fa} + (C-1)^2T_{fa}^2P_{fa}^2\right](1-P_d)(2-P_d)/P_d^2$$

Synchronization Time (Cont.)

- By using the identities:

$$\sum_{n=1}^{C} n = \frac{C(C+1)}{2}$$

$$\sum_{n=1}^{C} n^2 = \frac{C(C+1)(2C+1)}{6}$$

- The final result of the variance is

$$\sigma_{T_s}^2 = \overline{T_s^2} - (\overline{T}_s)^2 = \left[\frac{C^2-1}{12} - \frac{(C-1)^2}{P_d} + \frac{(C-1)^2}{P_d^2}\right]T_{da}^2 + (2C-1)\frac{1-P_d}{P_d^2}T_i^2 + 2(C-1)\frac{1-P_d}{P_d^2}T_iT_{fa}P_{fa} - (C-1)\frac{2-P_d}{2P_d}T_{fa}^2P_{fa}^2 + (C-1)\frac{2-P_d}{2P_d}T_{fa}^2P_{fa}$$

Synchronization Time (Cont.)

- For $C >> 1$, $1 - P_d << 1$, and $P_{fa} << 1$, the variance is approximated by

$$\sigma_{T_s}^2 \approx T_{da}^2 C^2 \left(\frac{1}{12} - \frac{1}{P_d} + \frac{1}{P_d^2} \right)$$

- For $C >> 1$, $P_d = 1$, and $P_{fa} = 0$, the variance is

$$\sigma_{T_s}^2 = T_{da}^2 C^2 \left(\frac{1}{12} \right)$$

Minimizes Mean Synchronization Time

- The mean synchronization time is a function of P_d, P_{fa}, T_i, T_{fa}, and C
- The phase uncertainty is fixed, but it can be subdivided into **any number** of cells C
- The remaining four variables are **not independent** of one another:
  - **High** P_d together with **low** P_{fa} implies large T_i
  - ⇒There will be **an optimum set** of P_d, P_{fa}, T_i, T_{fa} which minimizes mean synchronization time
- It is **not correct** to assume that the minimum average synchronization time will always be achieved with $1 - P_d << 1$
  - A moderate P_d will result in a much lower T_i than a high P_d
  - ⇒Result in reduced average T_s

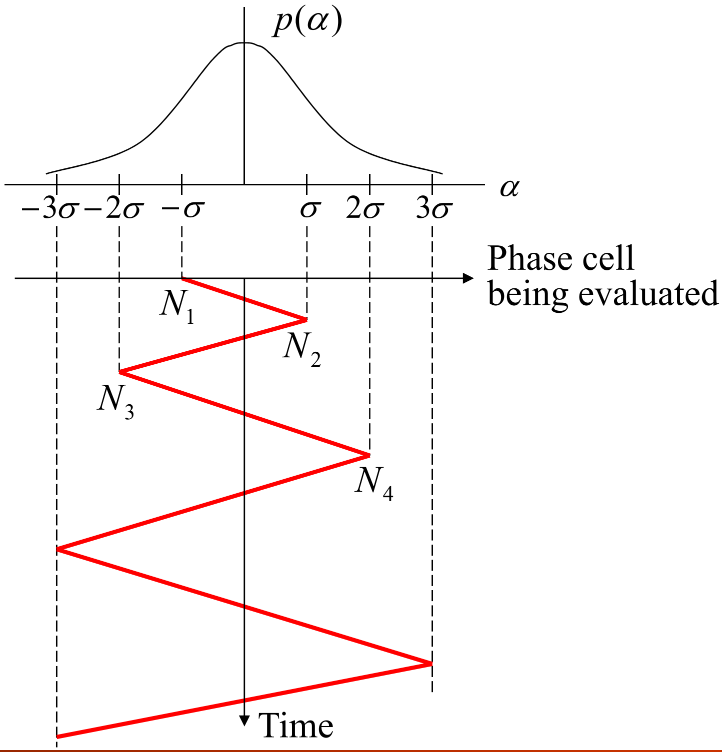
Modified Sweep Strategies

Modified Sweep Strategies

- Suppose that the received phase distribution is defined by $p(n)$ within the n-th phase cell
  - The sweep strategy should be modified to **search the most likely phase cells first** and then the less likely cells
- If the received phase distribution were **Gaussian**:
  - The discrete probability of the n-th phase cell being correct is
    $$p(n) = A\exp\left(-\frac{n^2}{2T^2}\right); \quad -\frac{1}{2}C \le n \le +\frac{1}{2}C$$
  - Search the cells within **one standard derivation** (T) of the most likely cell first
    $$T = (1/6)C$$

$$N_1 = -C/6; \quad N_2 = +C/6; \quad N_3 = -C/3; \quad N_4 = +C/3;$$

Modified Sweep Strategies (Cont.)



Modified Sweep Strategies (Cont.)

- The use of modified sweep strategies results in **reduced average synchronization time** when the distribution of the received phase is nonuniform
  - The saving time is a function of the **variance** of the received phase distribution

Continuous Linear Sweep

Continuous Linear Sweep

- Most current SS systems employ a sweep system which moves from one uncertainty cell to the next in **discrete steps**
- Another strategy: **offset the clock frequency** of the reference waveform generator (in the receiver) **slightly**

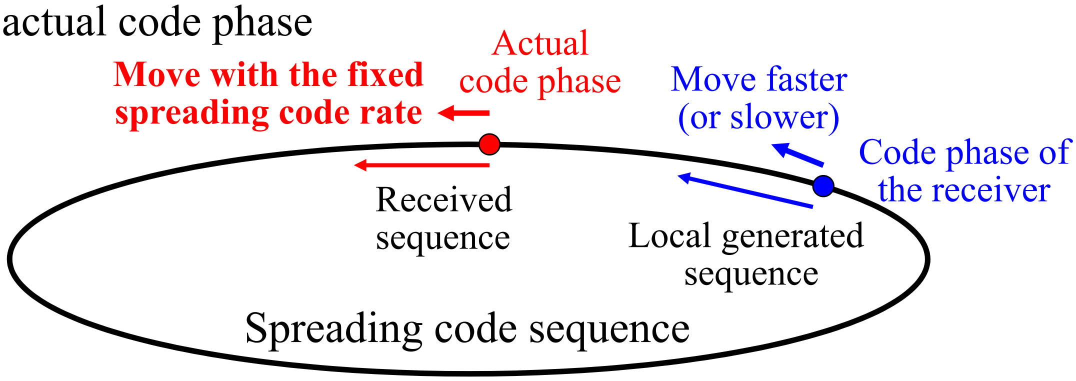
  ⇒The phase of the waveform **slips linearly**

  – The output of the despreading mixer is a **wideband signal** except when the received and reference waveform phases have **slipped sufficiently close** to one another ⇒ **Despread**

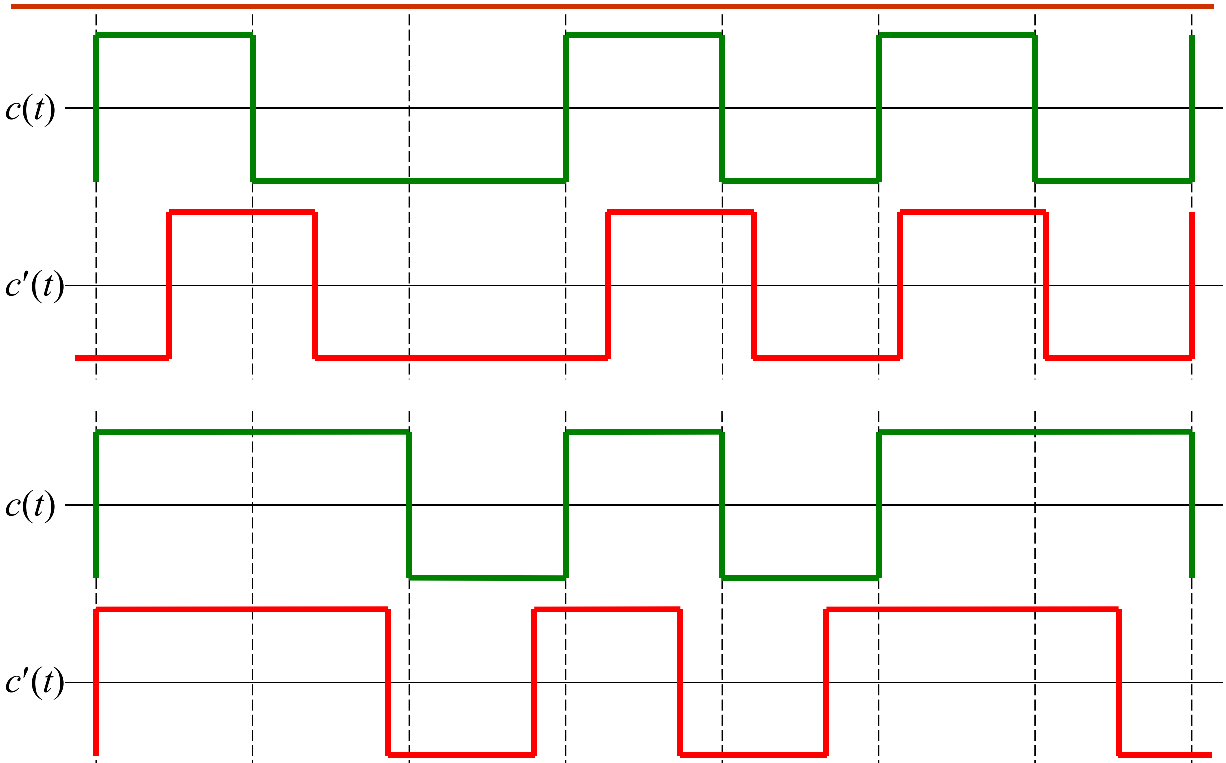
  – When despreading occurs, the energy can be detected and the sweep terminated

Continuous Linear Sweep (Cont.)

- The transmitted code sequence is generated with a constant rate
  - The phase of the received code sequence moves forward with a constant rate (i.e., the spreading code rate)
- The local generated code sequence is generated with a rate slightly **larger** (or **smaller**) than the spreading code rate
- Finally, the phase of the local sequence generated will capture the actual code phase

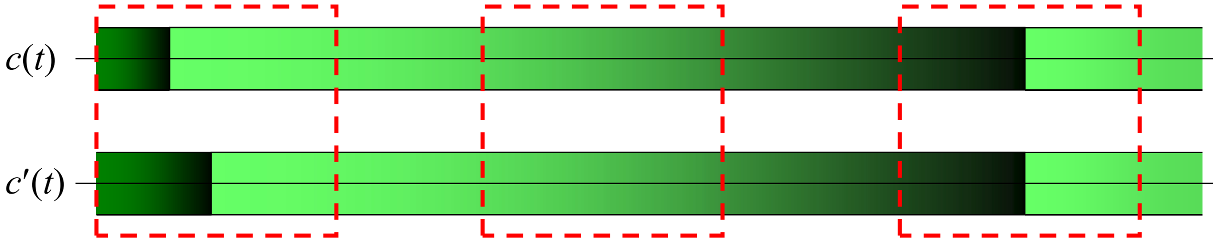


Continuous Linear Sweep (Cont.)

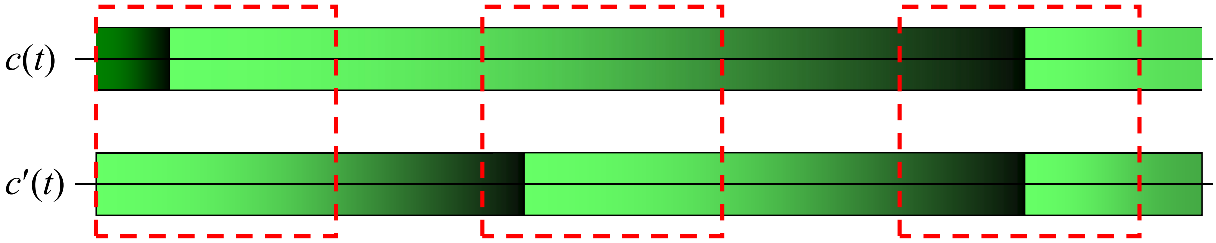


Continuous Linear Sweep (Cont.)

- Slow sweeping

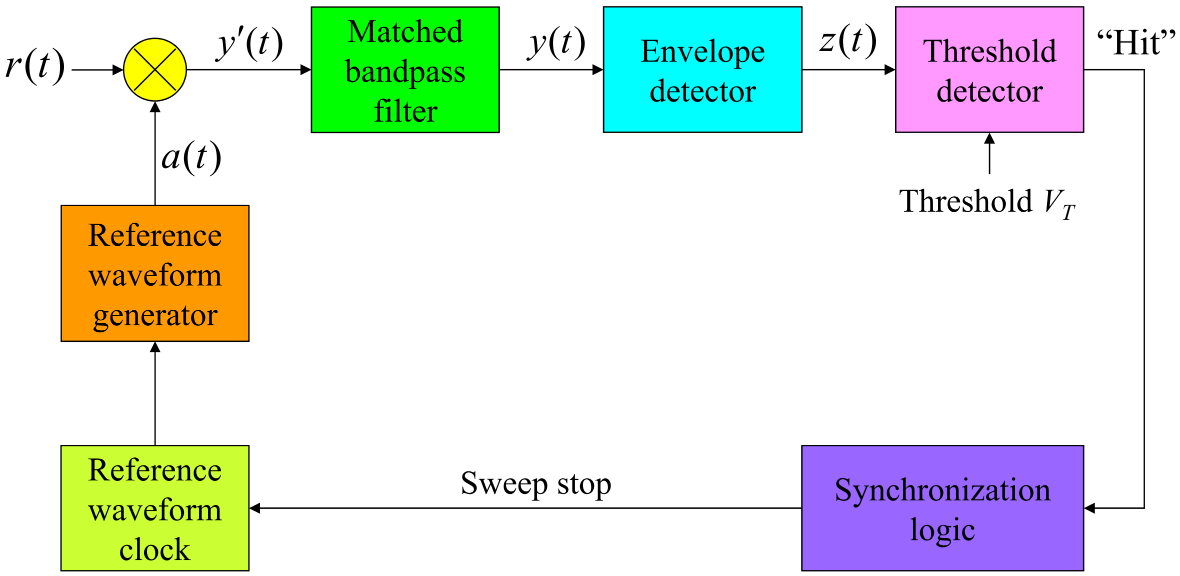


- Fast sweeping



Continuous Linear Sweep (Cont.)

- The receiver configuration for synchronization is



Continuous Linear Sweep (Cont.)

- Consider BPSK DS modulation, the received waveform is

$$r(t) = \sqrt{2P}c(t - T_d)\cos\omega_0 t + n(t)$$

- The despreading mixer output signal is

$$y'(t) = \sqrt{2P}c(t - T_d)c(t - \hat{T}_d)\cos(\omega_0 t + \theta) + n(t)c(t - \hat{T}_d)$$

  – where θ is an unknown random phase
- The propagation delay estimate is **varying linearly** with time

$$\hat{T}_d = \hat{T}_d(t) = \hat{T}_{do} + Kt$$

  – where $\hat{T}_{do}$ is an arbitrary fixed initial condition;
  – $K = T_c/T_{SS}$, and T_{SS} is the time required to **search one chip**
  – In general, $T_{SS} >> T_c \Rightarrow K << 1$
  – $K > 0$ ($K < 0$): reference clock frequency is lower (higher)

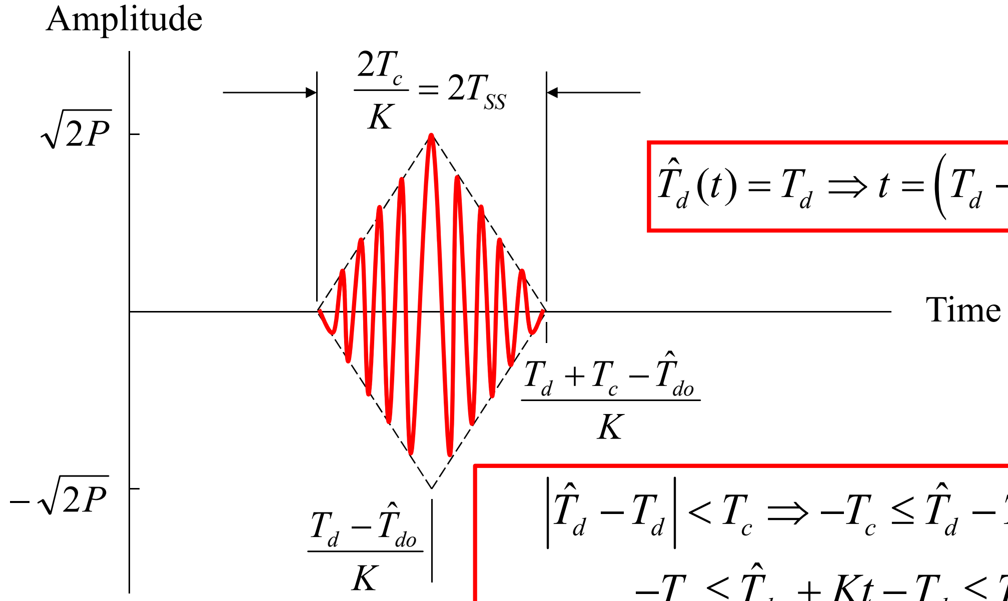
Continuous Linear Sweep (Cont.)

- From Eq. (5-31):
  – when $T_d - \hat{T}_d(t) > T_c$: a wideband signal
  – when $T_d - \hat{T}_d(t) = \tau < T_c$: the despreading mixer output is a **sinusoidal component** with frequency ω_0 and magnitude $\sqrt{2P}R_c(\tau)$
- The despreading mixer output becomes

$$x'(t) = \sqrt{2P}R_c(T_d - \hat{T}_{do} - Kt)\cos(\omega_0 t + \theta)$$

- The bandpass filter, envelope detector and threshold detector are used to detect the signal approximately when the **peak amplitude** is reached

Continuous Linear Sweep (Cont.)



$$\boxed{\hat{T}_d(t) = T_d \Rightarrow t = \left(T_d - \hat{T}_{do}\right)/K}$$

$$\left|\hat{T}_d - T_d\right| < T_c \Rightarrow -T_c \le \hat{T}_d - T_d \le T_c$$

$$-T_c \le \hat{T}_{do} + Kt - T_d \le T_c$$

$$\left(T_d - T_c - \hat{T}_{do}\right)/K \le t \le \left(T_d + T_c - \hat{T}_{do}\right)/K$$

Continuous Linear Sweep (Cont.)

- The **bandpass filter** is selected to have an impulse response which is **matched to** the signal $x'(t)$:

$$h(t) = bR_c(Kt - T_c)\cos\hat{\omega}_0 t$$

  – The frequency of the received signal is unknown during synchronization $\Rightarrow$ $\hat{\omega}_0$ is used (preset frequency)

- The matched filter output:

$$\cos\left(\hat{\omega}_0\alpha\right)\cos\left[\omega_0\left(t-\alpha\right)\right]$$
$$\cos(x)\cos(y) = \cos(x+y)/2 + \cos(x-y)/2$$

$$x(t) = \int_{-\infty}^{\infty} h(\alpha)x'(t-\alpha)\,d\alpha$$

$$= \sqrt{\frac{P}{2}}b\int_{-\infty}^{\infty} R_c\left(K\alpha - T_c\right)R_c\left(T_d - \hat{T}_{do} - Kt + K\alpha\right)\cos\left(\Delta\omega_0\alpha + \omega_0 t + \theta\right)d\alpha$$

$$+\sqrt{\frac{P}{2}}b\int_{-\infty}^{\infty} R_c\left(K\alpha - T_c\right)R_c\left(T_d - \hat{T}_{do} - Kt + K\alpha\right)\cos[\underline{(\omega_0 + \hat{\omega}_0)}\alpha - \omega_0 t - \theta]d\alpha$$

  – where $\Delta\omega_0 = \hat{\omega}_0 - \omega_0$

Continuous Linear Sweep (Cont.)

- Assume that the carrier frequency is **large** relative to the maximum rate of change of the autocorrelation product

  $\Rightarrow$ The second integral may be ignored
- Hence,

$$x(t) \cong \sqrt{\frac{P}{2}} b \int_{-\infty}^{\infty} R_c\left(K\alpha - T_c\right) R_c\left(T_d - \hat{T}_{do} - Kt + K\alpha\right) \underline{\cos\left(\Delta\omega_0\alpha + \omega_0 t + \theta\right)}\, d\alpha$$

$$= \sqrt{\frac{P}{2}} b \underline{\cos(\omega_0 t + \theta)} \int_{-\infty}^{\infty} R_c(K\alpha - T_c) R_c(T_d - \hat{T}_{do} - Kt + K\alpha) \times \underline{\cos(\Delta\omega_0\alpha)}\, d\alpha$$

$$- \sqrt{\frac{P}{2}} b \underline{\sin(\omega_0 t + \theta)} \int_{-\infty}^{\infty} R_c(K\alpha - T_c) R_c(T_d - \hat{T}_{do} - Kt + K\alpha) \times \underline{\sin(\Delta\omega_0\alpha)}\, d\alpha$$

Continuous Linear Sweep (Cont.)

- Assume that the frequency error $\Delta f_0 = \Delta\omega_0 / 2\pi$ is **small enough** $\left(\Delta\omega_0 = \hat{\omega}_0 - \omega_0\right)$

  $\Rightarrow$ The sine and cosine within integrals are approximately **constant** over the range of nonzero values of the autocorrelation products $\boxed{\cos(\Delta\omega_0\alpha); \sin(\Delta\omega_0\alpha)}$
- $h(\alpha)$ takes values only for $\underline{0 \le \alpha \le 2T_c / K} \Rightarrow$ set $\alpha = T_c / K$
- Set the arguments of the sine and cosine be $\Delta\omega_0\alpha = \Delta\omega_0 T_c / K$

$$x(t) \cong \sqrt{\frac{P}{2}} b \cos(\omega_0 t + \theta + \Delta\omega_0 \frac{T_c}{K}) \qquad \boxed{\begin{array}{l} \cos(\Delta\omega_0\alpha) \cong \cos(\Delta\omega_0 T_c / K) \\ \sin(\Delta\omega_0\alpha) \cong \sin(\Delta\omega_0 T_c / K) \end{array}}$$

$$\times \int_{-\infty}^{\infty} R_c(K\alpha - T_c) R_c(T_d - \hat{T}_{do} - Kt + K\alpha)\, d\alpha$$

- The maximum occurs at the **end** of the triangular pulse: (Fig. 5-5) $t = (T_d + T_c - \hat{T}_{do}) / K$

Continuous Linear Sweep (Cont.)

- The maximum filter output is

$$x_{\max}(t) = \sqrt{\frac{P}{2}}b\cos(\omega_0 t + \theta + \Delta\omega_0 \frac{T_c}{K})\int_{-\infty}^{\infty} R_c^2(K\alpha - T_c)\, d\alpha$$

$$= \frac{\sqrt{2P}}{3} bT_{ss}\cos(\omega_0 t + \theta + \Delta\omega_0 T_{ss}) \qquad (5.37)$$

- The noise power is calculated using the **baseband equivalent** filter transfer function:

$$h(t) = 2\mathrm{Re}\left[\tilde{h}(t)e^{j\hat{\omega}_0 t}\right]$$

$$\tilde{h}(t) = \frac{b}{2}R_c(Kt - T_c)$$

Continuous Linear Sweep (Cont.)

- The Fourier transform of $\tilde{h}(t)$ is

$$\tilde{H}(f) = \frac{b}{2}T_{ss}\,\mathrm{sinc}^2(fT_{ss})$$

- The equivalent baseband noise PSD is

$$S_{\tilde{n}}(f) \cong 2S_{n_I}(f) = \begin{cases} N_0 & |f| < B \\ 0 & \text{elsewhere} \end{cases}$$

- The noise power at the filter output is

$$N = \int_{-\infty}^{\infty} \left|\tilde{H}(f)\right|^2 S_{\tilde{n}}(f)df$$

$$= \frac{1}{4}b^2T_{SS}^2N_0\int_{-B}^{B}\mathrm{sinc}^4(fT_s)df \cong \frac{b^2}{6}T_{SS}N_0$$

Continuous Linear Sweep (Cont.)

- The maximum signal-to-noise power ratio at the matched filter output is

$$SNR_{\max} = \frac{E\left[x_{\max}^2(t)\right]}{N} = \frac{\left[(\sqrt{2P}/3)bT_{ss}\right]^2/2}{b^2T_{ss}N_0/6} = \frac{2}{3}\left(\frac{PT_{ss}}{N_0}\right)$$

  – The SNR increases with **increasing** $\boldsymbol{T_{ss}}$ or, equivalently, with **decreasing sweep rate**

- The function of the envelope detector and threshold comparator is used to
  – Detect the **presence** of the sinusoid (Eq. (5-37))

Continuous Linear Sweep (Cont.)

- The density function of the envelope detector output is **Ricean** pdf.
- Denote the envelope detector output by $z(t)$:

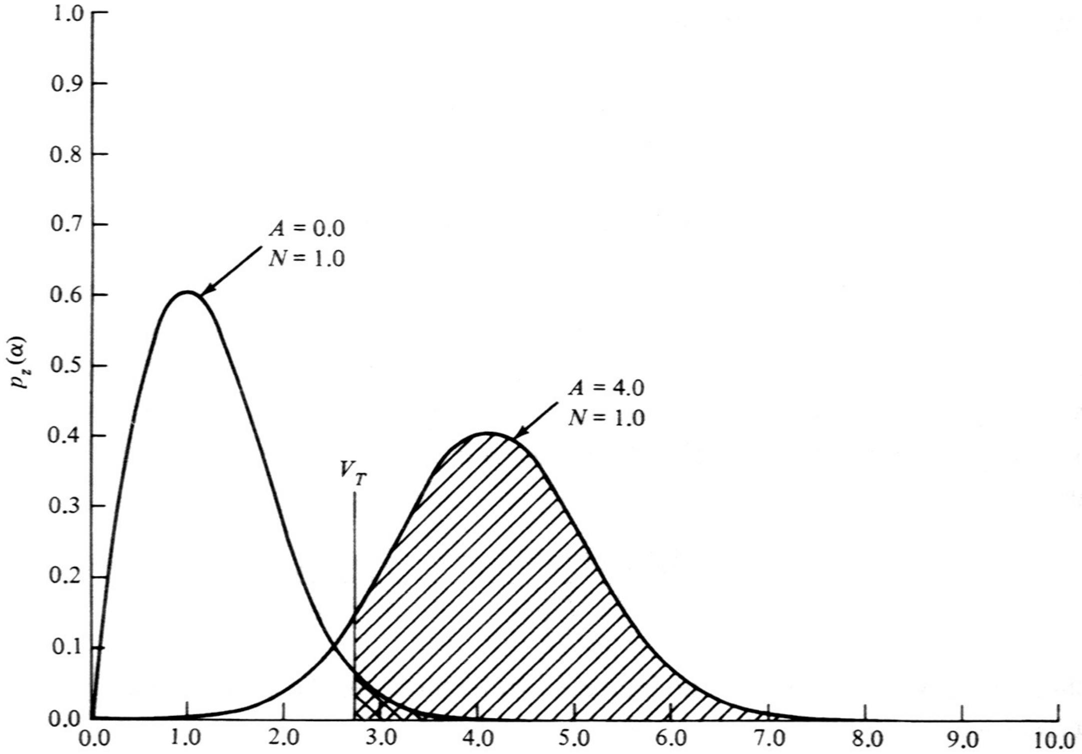
$$p_z(\alpha) = \begin{cases} \dfrac{\alpha}{N}\exp\left(-\dfrac{\alpha^2+A^2}{2N}\right)I_0\left(\dfrac{\alpha A}{N}\right), & \text{for } \alpha \ge 0 \\ 0, & \text{elsewhere} \end{cases}$$

  – where N: the noise power, A: the amplitude of the sinusoid, and $I_0(\cdot)$: the zeroth-order modified Bessel function

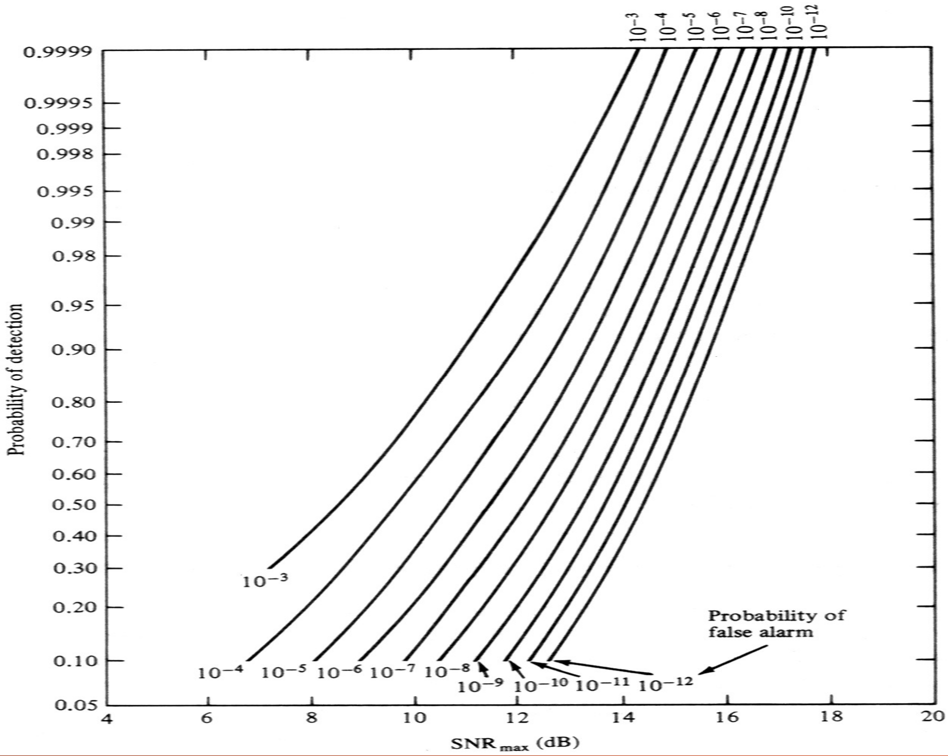
Continuous Linear Sweep (Cont.)

- For the probability of **detection**:
  - $A \neq 0$, signal and noise present $\Rightarrow$ **Ricean** pdf
  - Integration from V_T to infinity of the Ricean pdf evaluated at the **maximum signal-to-noise ratio**
- For the probability of **false alarm**:
  - $A = 0$, noise only $\Rightarrow$ **Rayleigh** pdf
  - Integration from V_T to infinity of the Rayleigh pdf evaluated at the **minimum signal-to-noise ratio** (for $A = 0$, noise only)
- Both P_d and P_{fa} are functions of SNR and V_T, and **cannot be independently selected**
  - For a specific SNR, the selection of P_{fa} implies a particular threshold V_T, which then sets P_d

Continuous Linear Sweep (Cont.)



Continuous Linear Sweep (Cont.)



Continuous Linear Sweep (Cont.)

- The calculation of points on the curves of Fig. 5-7:
  - A preset P_{fa} implies a particular threshold V_T
  - Integrate the Ricean density function over the limits of V_T to infinity using the desired SNR $\Rightarrow P_d$
  - Marcum Q-function:

$$Q(a,b) \triangleq \int_b^{\infty} \alpha \exp\left[-\frac{1}{2}(\alpha^2 + a^2)\right] I_0(a\alpha)\, d\alpha$$

    - SNR $\rightarrow a$
    - Threshold $V_T \rightarrow b$

Example

- The spreading code clock frequency is f_c = 3 MHz
- The received carrier power-to-noise power spectral density ratio is 46.25 dB-Hz = 10 log(P/N_0)
- The propagation delay uncertainty is ±1.2 ms
- If (P_d, P_{fa}) pairs are (0.9, 10^{-6}), (0.8, 10^{-6}), (0.9, 10^{-3}), (0.8, 10^{-3})
- From Fig. 5-7, the required signal-to-noise ratios for these four (P_d, P_{fa}) pairs are 13.4, 12.8, 11.0, and 10.3 dB

$$10\log\left(SNR_{\max}\right)=10\log\left(\frac{2}{3}\right)+10\log\left(\frac{P}{N_0}\right)+10\log\left(T_{ss}\right)$$

$$13.4=-1.761+46.25+10\log\left(T_{ss}\right)$$

$$10\log\left(T_{ss}\right)=-31.09$$

$$T_{ss}=778\ \mu s$$

Example (Cont.)

- Solving for T_{ss}:

$$T_{ss}=\begin{cases}778\ \mu s & \text{for } \left(0.9,10^{-6}\right)\\ 678\ \mu s & \text{for } \left(0.8,10^{-6}\right)\\ 448\ \mu s & \text{for } \left(0.9,10^{-3}\right)\\ 381\ \mu s & \text{for } \left(0.8,10^{-3}\right)\end{cases}$$

- If the false-alarm penalty is 100 T_i:

$$T_i=T_{ss}$$

$$T_{da}=T_{ss}+P_{fa}\left(100T_{ss}\right)=T_{ss}\left(1+100P_{fa}\right)$$

$$C=2\times\left(1.2\text{ms}\right)\times\left(3\times10^6\text{ chips/s}\right)=7.2\times10^3$$

$$\overline{T}_s=(C-1)T_{da}\left(\frac{2-P_d}{2P_d}\right)+\frac{T_i}{P_d}=7199T_{ss}\left(1+100P_{fa}\right)\left(\frac{2-P_d}{2P_d}\right)+\frac{T_{ss}}{P_d}$$

Example (Cont.)

- Solving for $\overline{T}_s$:

$$\overline{T}_s = \begin{cases} 3.42\, s & \text{for } \left(0.9, 10^{-6}\right) \\ 3.66\, s & \text{for } \left(0.8, 10^{-6}\right) \\ 2.17\, s & \text{for } \left(0.9, 10^{-3}\right) \\ 2.26\, s & \text{for } \left(0.8, 10^{-3}\right) \end{cases}$$

- Permitting P_{fa} to increase by three orders of magnitude can decrease synchronization time significantly

Detection of A Signal in AWGN

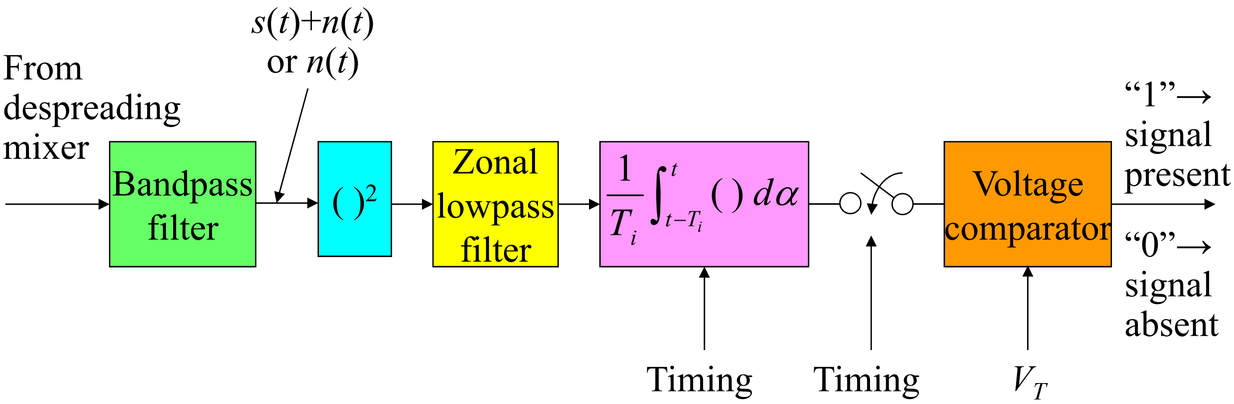
Detection of A Signal in AWGN

- All synchronization systems that employ a **discrete step serial search** evaluate a particular phase/frequency cell by estimating the presence of signal energy
- The mean and variance of the synchronization time, P_d and P_{fa} are a function of evaluation time T_i and the received SNR
- Three different methods of detecting signal energy are
  - **Fixed integration time detection**
  - **Multiple-dwell detection**
  - **Sequential detection**
- Each method results in a different relationship between P_d, P_{fa}, T_i, and SNR

Fixed Integration Time Detection

Fixed Integration Time Detection

- The **simplest method** of detecting the presence of signal energy at the output of a narrowband filter
- At the bandpass filter output:
  - $s(t)$ appears when the reference waveform **phase is correct**
  - $n(t)$ is the AWGN process that will be **always present**



Fixed Integration Time Detection (Cont.)

- The filter output is squared and then lowpass filtered (Zonal lowpass filter) to eliminate the **double-frequency terms** (which result from the squaring operation)
- The lowpass squared output is integrated for T_i seconds
  - From $t - T_i$ to t (present time)
- The output is compared to a fixed threshold V_T:
  - Above the threshold: the signal is declared **present**
  - Otherwise: the signal is declared **absent**

Multiple-Dwell Detection

Multiple-Dwell Detection

- Fixed integration time detection is limited in that only two parameters, T_i and V_T, can be varied to reduce the synchronization time
- Note that there is **only one** correct cell within the uncertainty region
  - Other cells evaluated by the energy detector are **noise alone**

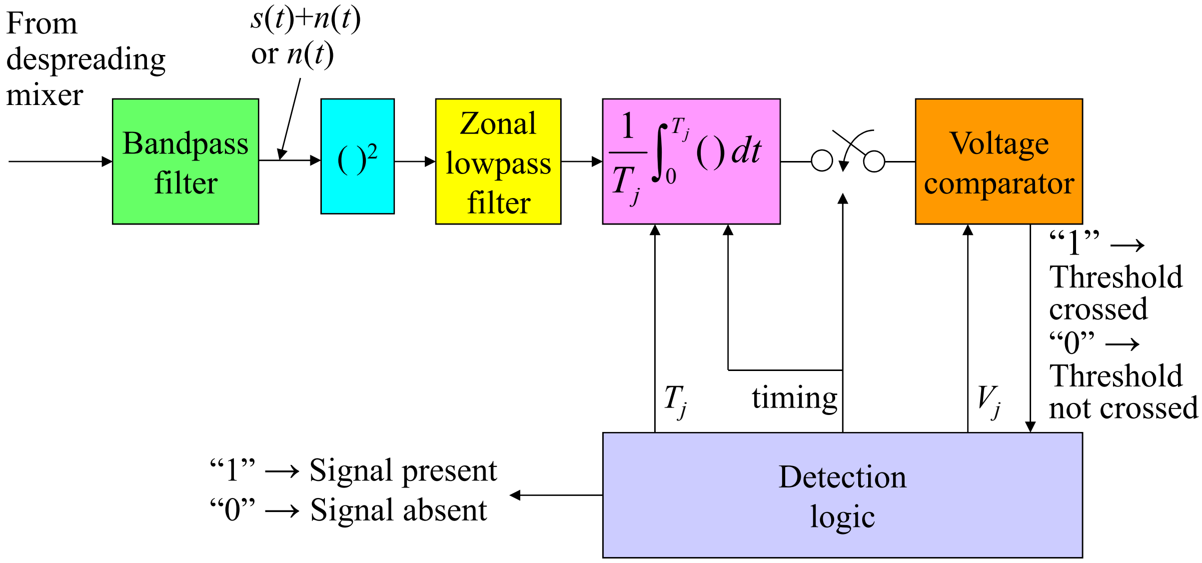
  $\Rightarrow$ An energy detection scheme that is capable of rejecting incorrect phase cells **rapidly** while not letting P_{fa} become so large is highly desirable

Multiple-Dwell Detection (Cont.)

- The multiple-dwell detection scheme uses **multiple evaluations** for each phase cell
  - The **first evaluation** is **very short** (short integration time)
    - Results in immediate rejection of many incorrect cells
    - Results in a high false-alarm probability
  - When a false alarm occurs on the first evaluation, a **second evaluation** (**longer** integration time) of the same cell begins
    - Reduces the false-alarm probability
  - The second evaluation may be followed by a third, fourth, or as many as desired to achieve a particular performance

Multiple-Dwell Detection – Block Diagram

- Fig. 5-13 is a simplified conceptual block diagram:
  - Selects integration times T_j and thresholds V_j for each integration



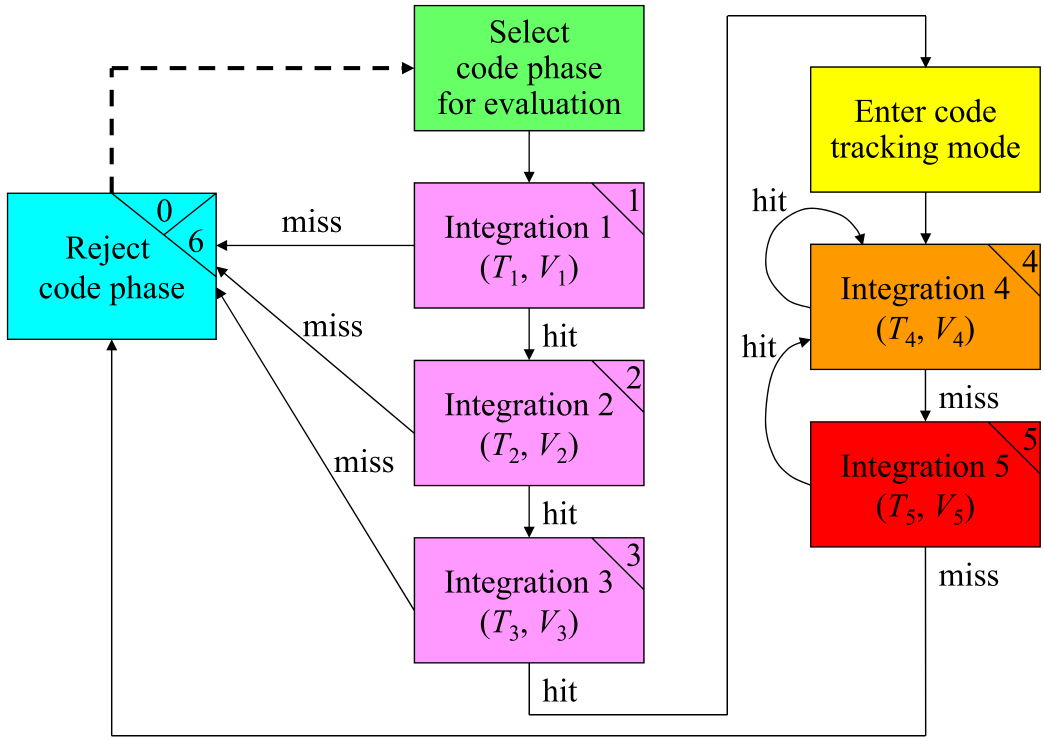
Multiple-Dwell Detection – Flow Diagram

- Fig. 5-14 is a flow diagram
- Integration 1: integrates for T_1 seconds, and the integrator output V is compared with a threshold V_1
  - If $V < V_1$, a "miss" (**incorrect** phase) is declared
  - If $V > V_1$, a "hit" (**potential correct** phase) is declared
- Integration k: integrates for T_k seconds, and the integrator output V is compared with a threshold V_k
- The "hit" output from **integration 3** generates the "**signal present**" output
- The "**signal absent**" output is generated when a "miss" occurs on any of the **first three integrations**

Multiple-Dwell Detection–Flow Diagram (Cont.)

- Integration 4 and integration 5 provide a means for detecting when the **code tracking loop** has lost lock
  - These two integration times may be quite large
  - Because an incorrect dismissal of the **correct code phase** is very serious
    - The correct code phase will appear after N cells
  - When a false alarm occurs, **Integrations 4 and 5** must be performed before the code phase can be rejected
- During **normal code tracking**, the logic continues to cycle around the loop connecting **Integrations 4 and itself**
  - Code tracking and demodulation
- Therefore, $\boldsymbol{T_1 < T_2 < T_3 < T_4 < T_5}$

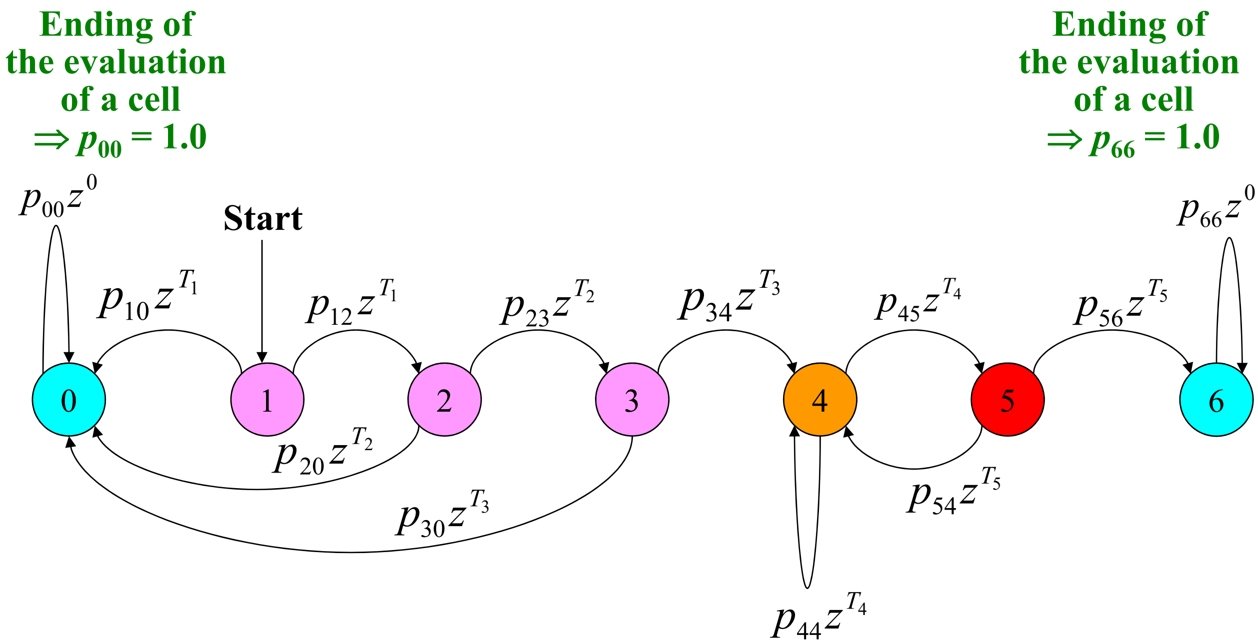
Multiple-Dwell Detection–Flow Diagram (Cont.)



Multiple-Dwell Detection – State Transition

- Fig. 5-14 can be represented by a state transition diagram as Fig. 5-15
  - The 0/6 state has been split into two states
  - T_{da}: the average time required for the system to progress from state 1 to state 0/6
  - Each transition of the diagram is labeled with
    - P_{jk}: **transition probability** from state j to state k (calculated from the integration time T_j, the threshold V_j, and the SNR)
    - z: raised to a power equal to the **integration time** associated with the "from" state

MD Detection–State Transition (Cont.)



MD Detection – Synchronization Time

- The mean time required to reject an incorrect phase is

$$T_{da} = \sum_{l \in L} \Pr(l) T_l$$

  - L: the set of all paths beginning at state 1 and ending at either state 0 or 6
  - l: a particular path in L
  - $\Pr(l)$: the probability of following path l
  - T_l: the time required to traverse path l

MD Detection – Synchronization Time (Cont.)

- For a specific path l_0 through the state diagram:
  - Path 1-2-3-0
  - The probability of this path is

$$\Pr(l_0) = p_{12} p_{23} p_{30}$$

  - The time required to traverse this path is

$$T_{l_0} = T_1 + T_2 + T_3$$

$$\Pr(l_0) T_{l_0} = p_{12} p_{23} p_{30} \left(T_1 + T_2 + T_3\right)$$

  - Let $B(l,z)$ denote the product of the path labels

$$\begin{aligned} B(l_0, z) &= p_{12} z^{T_1} p_{23} z^{T_2} p_{30} z^{T_3} \\ &= p_{12} p_{23} p_{30} z^{T_1+T_2+T_3} \end{aligned}$$

MD Detection – Synchronization Time (Cont.)

$$\frac{d}{dz} B(l_0, z) = p_{12} p_{23} p_{30} (T_1 + T_2 + T_3) z^{T_1+T_2+T_3-1}$$

  - Equals to Eq. (5-93) if $\boldsymbol{z = 1.0}$:

$$\Pr(l_0) T_{l_0} = \left[\frac{d}{dz} B(l_0, z) \right]_{z=1}$$

- The mean time required to reject an incorrect phase is

$$T_{da} = \sum_{l \in L} \Pr(l) T_l = \sum_{l \in L} \left[\frac{d}{dz} B(l, z) \right]_{z=1}$$

MD Detection – Synchronization Time (Cont.)

- The state-transition diagram can also be described by a **transition matrix**:
  - Rows: starting states
  - Columns: ending states
  - Elements: path labels

“to” state

$$\mathbf{Q}' = \begin{array}{c} \\ 0 \\ 6 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 6 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cc|ccccc} p_{00}z^{0} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p_{66}z^{0} & 0 & 0 & 0 & 0 & 0 \\ \hline p_{10}z^{T_1} & 0 & 0 & p_{12}z^{T_1} & 0 & 0 & 0 \\ p_{20}z^{T_2} & 0 & 0 & 0 & p_{23}z^{T_2} & 0 & 0 \\ p_{30}z^{T_3} & 0 & 0 & 0 & 0 & p_{34}z^{T_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{44}z^{T_4} & p_{45}z^{T_4} \\ 0 & p_{56}z^{T_5} & 0 & 0 & 0 & p_{54}z^{T_5} & 0 \end{array}\right] \end{array}$$

“from” state

MD Detection – Synchronization Time (Cont.)

- The matrix $\mathbf{Q}'$ can be partitioned into **four submatrices**: **U**, **R**, **Q**, and **0**
  - **Q** contains information about all the system **internal states**

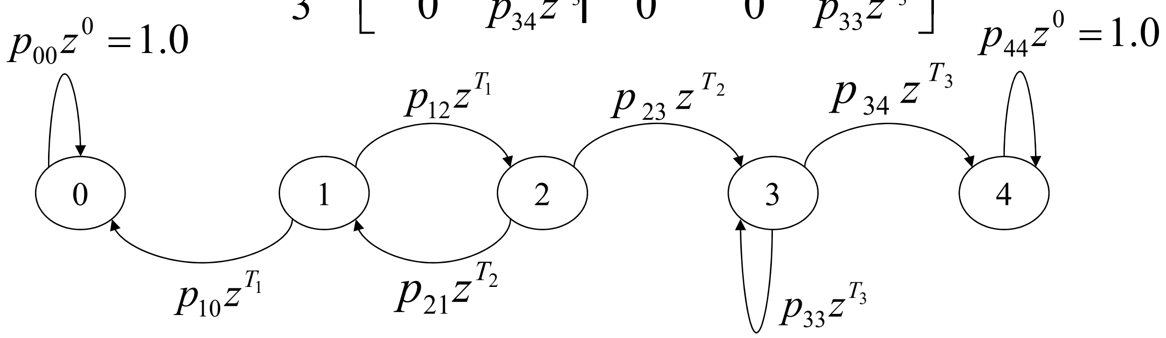
$$\mathbf{Q}' \triangleq \left[\begin{array}{c|c} \mathbf{U} & \mathbf{0} \\ \hline \mathbf{R} & \mathbf{Q} \end{array}\right]$$

- Define a new matrix $\mathbf{X} = \mathbf{Q}^n\mathbf{R}$:
  - Rows: inner states (starting states)
  - Columns: end states (ending states)
  - Element (x_{jk}): the sum of the product of path labels on the associated path of length $n + 1$ from state j to state k

Example

- Consider the state transition diagram:

$$
\mathbf{Q}' = \begin{array}{c} \\ 0 \\ 4 \\ 1 \\ 2 \\ 3 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 0 & 4 & 1 & 2 & 3 \end{array} \\
\left[\begin{array}{cc|ccc}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\ \hline
p_{10}z^{T_1} & 0 & 0 & p_{12}z^{T_1} & 0 \\
0 & 0 & p_{21}z^{T_2} & 0 & p_{23}z^{T_2} \\
0 & p_{34}z^{T_3} & 0 & 0 & p_{33}z^{T_3}
\end{array}\right]
\end{array}
$$



Example (Cont.)

$$
\mathbf{R} = \begin{bmatrix} p_{10}z^{T_1} & 0 \\ 0 & 0 \\ 0 & p_{34}z^{T_3} \end{bmatrix}
$$

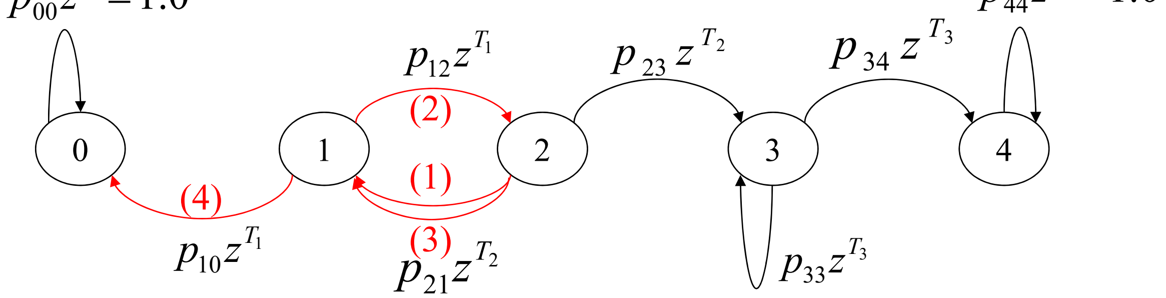
$$
\mathbf{Q} = \begin{bmatrix} 0 & p_{12}z^{T_1} & 0 \\ p_{21}z^{T_2} & 0 & p_{23}z^{T_2} \\ 0 & 0 & p_{33}z^{T_3} \end{bmatrix}
$$

$\mathbf{X} = \mathbf{Q}^3\mathbf{R} =$

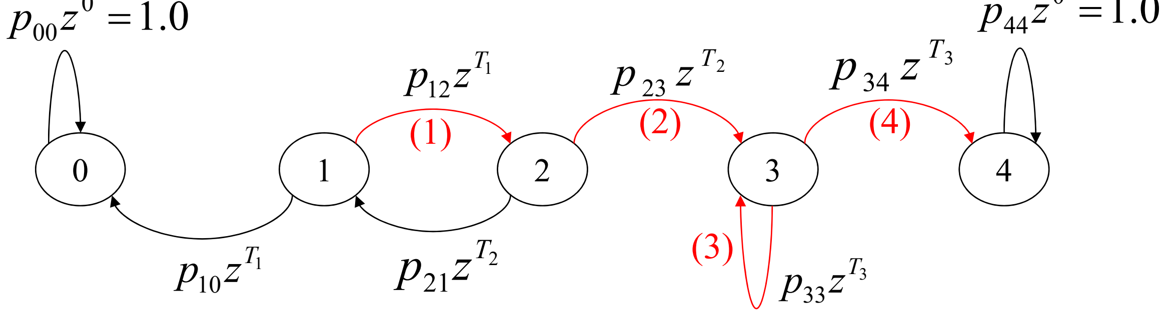
$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}
\begin{array}{c}
\begin{array}{cc} 0 & 4 \end{array} \\
\begin{bmatrix}
0 & p_{12}p_{23}p_{33}p_{34}z^{T_1+T_2+2T_3} \\
p_{12}p_{21}^2p_{10}z^{2T_1+2T_2} & p_{21}p_{12}p_{23}p_{34}z^{T_1+2T_2+T_3} + p_{23}p_{33}^2p_{34}z^{T_2+3T_3} \\
0 & p_{33}^3p_{34}z^{4T_3}
\end{bmatrix}
\end{array}
$$

Example (Cont.)

State 2 → State 0

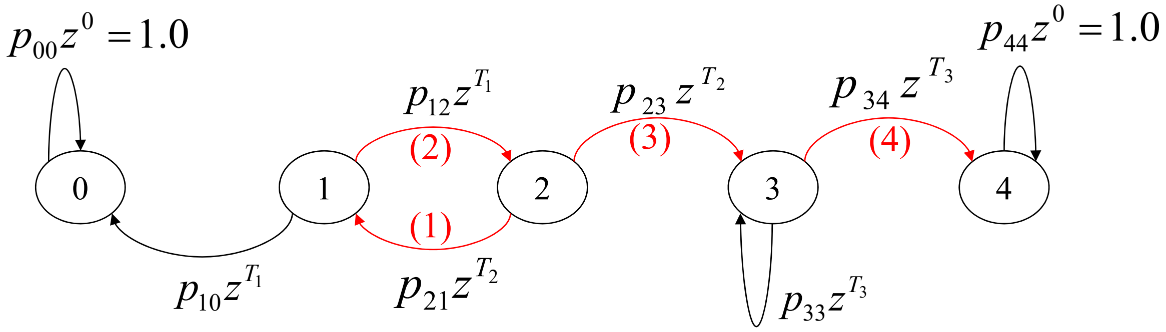


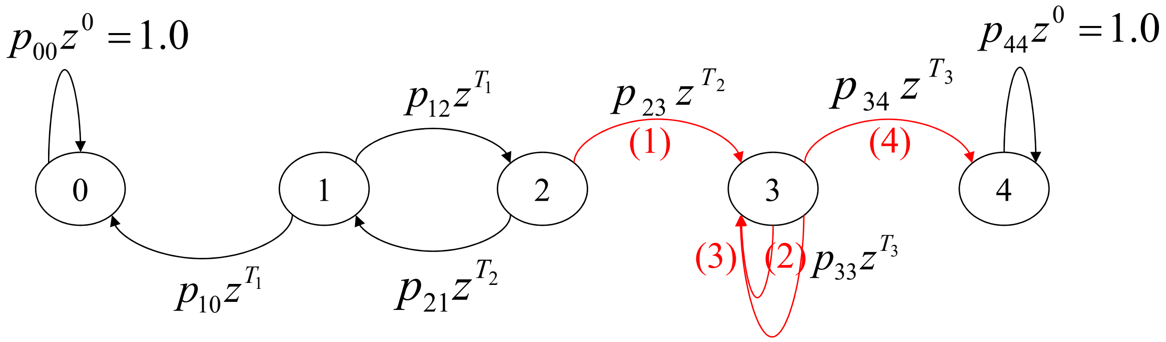
State 1 → State 4



Example (Cont.)

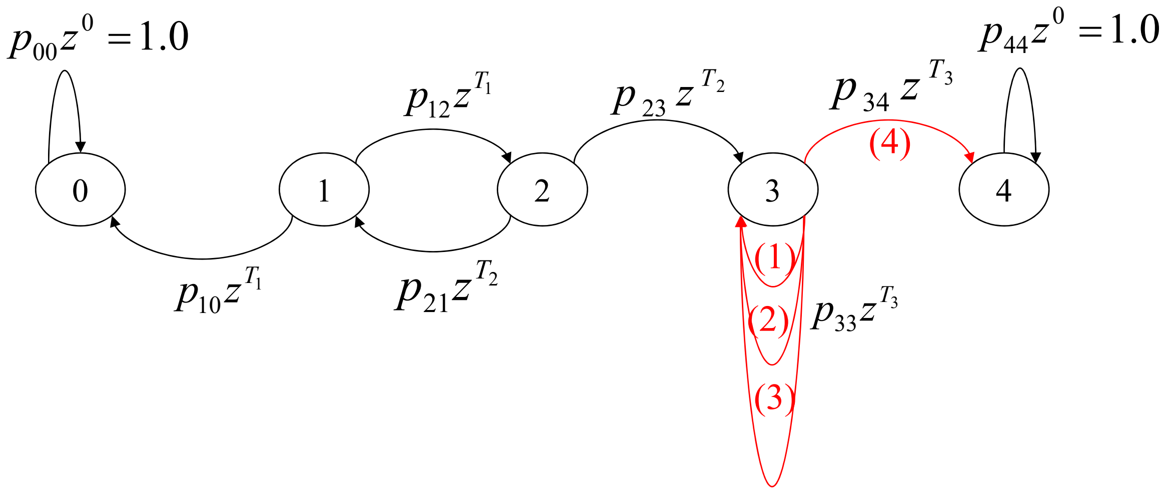
State 2 → State 4





Example (Cont.)

State 3 → State 4



MD Detection – Synchronization Time (Cont.)

- The infinite matrix sum **Y** enumerates **all paths of all lengths** between inner states and end states

$$\mathbf{Y} = \mathbf{R} + \mathbf{QR} + \mathbf{Q}^2\mathbf{R} + \mathbf{Q}^3\mathbf{R} + \cdots$$

$$y_{jk} = \sum_{l \in L(j,k)} B(l,z)$$

  - where $L(j,k)$ is the set of all paths beginning at state j and ending at state k
- For the average time to reject an incorrect cell
  - L denotes all paths between state 1 and state 0 or 6

$$T_{da} = \sum_{l \in L} \Pr(l) T_l = \sum_{l \in L} \left[\frac{d}{dz} B(l,z) \right]_{z=1}$$

$$L = L(1,0) + L(1,6)$$

MD Detection – Synchronization Time (Cont.)

- The average time to reject an incorrect cell is:

$$
\begin{aligned}
T_{da} &= \sum_{l \in L(1,0)} \left[\frac{d}{dz} B(l,z) \right]_{z=1} + \sum_{l \in L(1,6)} \left[\frac{d}{dz} B(l,z) \right]_{z=1} \\
&= \left\{ \frac{d}{dz} \left[\sum_{l \in L(1,0)} B(l,z) \right] + \frac{d}{dz} \left[\sum_{l \in L(1,6)} B(l,z) \right] \right\}_{z=1} \\
&= \left[\frac{d}{dz}(y_{10}) + \frac{d}{dz}(y_{16}) \right]_{z=1}
\end{aligned}
$$

- The matrix **Y** can be rewritten as

$$
\begin{aligned}
\mathbf{Y} &= (\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \cdots)\mathbf{R} \\
&= (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{R}
\end{aligned}
$$

MD Detection – Synchronization Time (Cont.)

- The derivative with respect to z is

$$
\begin{aligned}
\frac{d}{dz}(\mathbf{Y}) &= (\mathbf{I} - \mathbf{Q})^{-1} \left(\frac{d}{dz} \mathbf{R} \right) + \left[\frac{d}{dz} (\mathbf{I} - \mathbf{Q})^{-1} \right] \mathbf{R} \\
&= (\mathbf{I} - \mathbf{Q})^{-1} \left(\frac{d}{dz} \mathbf{R} \right) - (\mathbf{I} - \mathbf{Q})^{-1} \left[\frac{d}{dz} (\mathbf{I} - \mathbf{Q}) \right] (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{R}
\end{aligned}
$$

$$
\because \frac{d}{dz} \mathbf{A}^{-1} = -\mathbf{A}^{-1} \left(\frac{d}{dz} \mathbf{A} \right) \mathbf{A}^{-1}
$$

- Define a diagonal $n \times n$ matrix **T** of **integration times**

$$
\mathbf{T} = \begin{bmatrix} T_1 & & \mathbf{0} & \\ & T_2 & & \\ & & \ddots & \\ & \mathbf{0} & & T_n \end{bmatrix}
$$

MD Detection – Synchronization Time (Cont.)

$$\left(\frac{d}{dz}\mathbf{R}\right)_{z=1} = (\mathbf{TR})_{z=1}$$

From state *i* to outer states $\Rightarrow$ Time interval T_i

$$\left[\frac{d}{dz}(\mathbf{I}-\mathbf{Q})\right]_{z=1} = (-\mathbf{TQ})_{z=1}$$

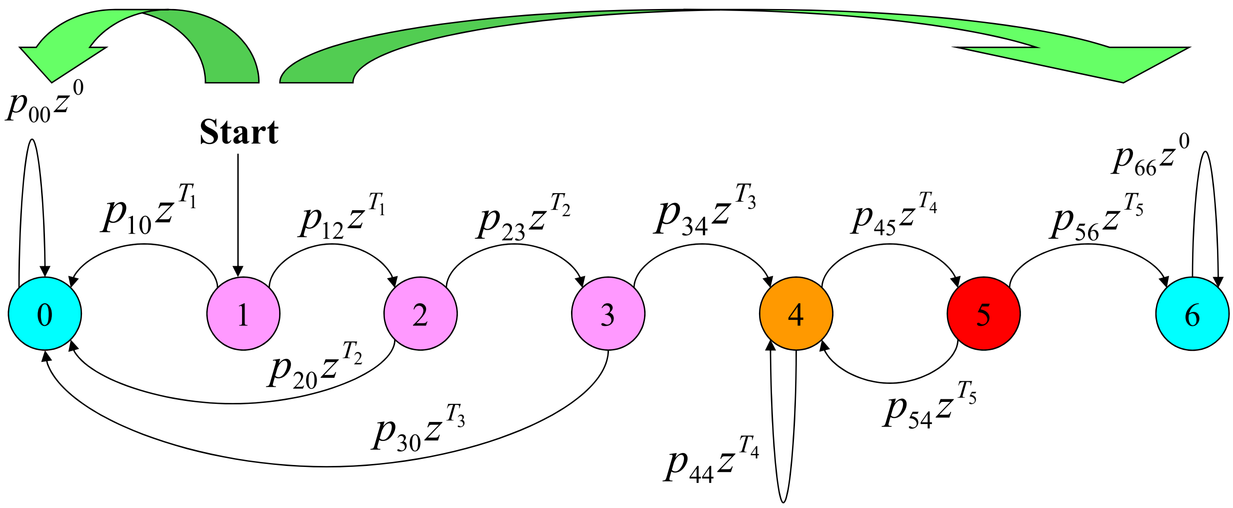
From state *i* to inner states $\Rightarrow$ Time interval T_i

- With $z = 1.0$:

$$\begin{aligned}\left(\frac{d}{dz}\mathbf{Y}\right)_{z=1} &= \left\{(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{TR}+(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{TQ}(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{R}\right\}_{z=1}\\ &= \left\{(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{T}\left[\mathbf{I}+\mathbf{Q}(\mathbf{I}-\mathbf{Q})^{-1}\right]\mathbf{R}\right\}_{z=1}\\ &= \left\{(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{T}\left[\mathbf{I}+\mathbf{Q}\left(\mathbf{I}+\mathbf{Q}+\mathbf{Q}^2+\cdots\right)\right]\mathbf{R}\right\}_{z=1}\\ &= \left[(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{T}(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{R}\right]_{z=1}\end{aligned}$$

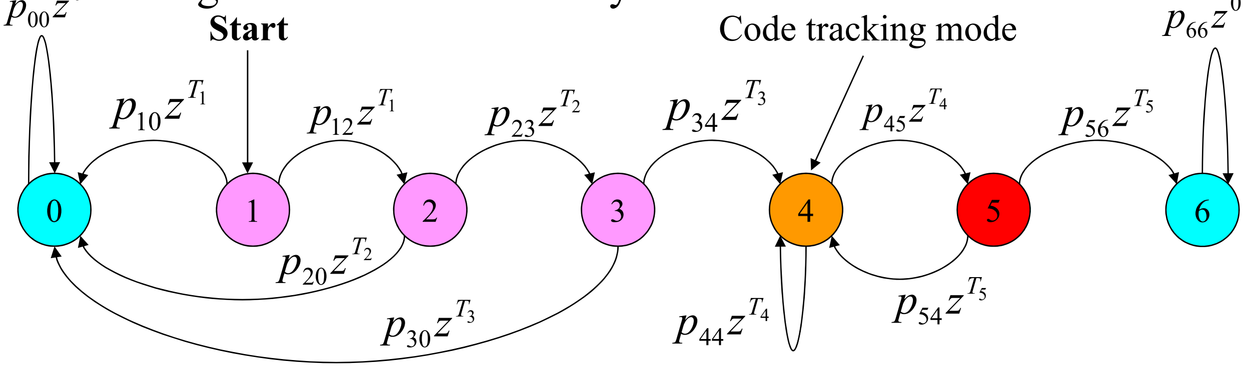
MD Detection – Synchronization Time (Cont.)

- The average time required to reject an **incorrect phase cell** is the sum of the elements on the **first row** of Eq. (5-113)
  - From state 1 to state 0 + From state 1 to state 6



MD Detection – Synchronization Time (Cont.)

- The probability of **detecting the correct phase cell**, P_d is
  - The probability of passing from state 1 to the “enter code tracking mode” function
  - The probability from the “code tracking mode” (pass through state 4 and 5) to state 6 is **one**
    - No other path to an end state and the system is guaranteed to eventually reach an end state



MD Detection – Synchronization Time (Cont.)

- The probability of **detecting the correct phase cell**, P_d, is equal to the probability of passing from state 1 to state 6

$$P_d = \sum_{l \in L(1,6)} \Pr(l) = \sum_{l \in L(1,6)} B(l,z)_{z=1}$$

$$P_d = (y_{16})_{z=1}$$

  - P_d is the element of the first row and second column of

$$(\mathbf{Y})_{z=1} = \left[(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{R}\right]_{z=1}$$

MD Detection – Synchronization Time (Cont.)

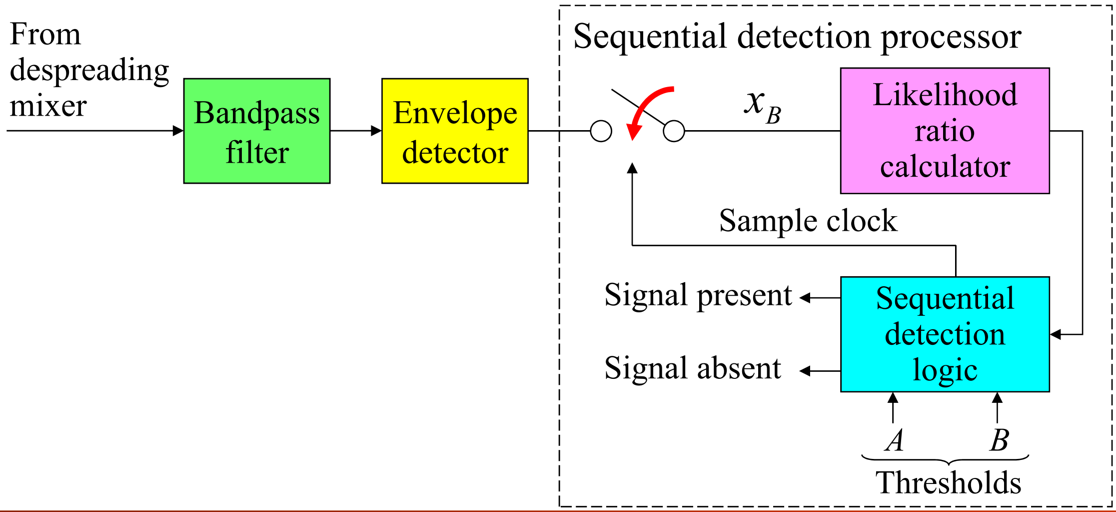
- To evaluate the average synchronization time: **two sets** of transition probabilities must be calculated
- For the **noise along** case:
  - To evaluate T_{da}
- For the **signal plus noise** case:
  - To evaluate T_i (the average time used in evaluating the correct phase cell): identical to the calculation of T_{da} except the **transition probabilities**
  - To evaluate P_d
- Note that average values for T_{da} and T_i can be used to calculate $\overline{T}_s$, but cannot be used to calculate the variance $\sigma_{T_s}^2$

$$\overline{T}_s = (C-1)T_{da}\left(\frac{2-P_d}{2P_d}\right) + \frac{T_i}{P_d}$$

Sequential Detection

Sequential Detection

- Sequential detection is used to detect the signal power reliably but without generating excessive false alarms, it includes
  - The **envelope detector**
  - The **sequential detection processor**



Sequential Detection (Cont.)

- For any $K = 1, 2, \ldots,$ let $\mathbf{x}_K = (x_1, x_2, x_3, \ldots, x_K)$ denote a sequence of samples of the envelope detector output
  - When **signal** is present: the joint pdf of samples is $p_s(\mathbf{x}_K)$
    - $p_s(\mathbf{x}_K)$ depends on the input signal power
  - When **noise alone** is present: the joint pdf of samples is $p_n(\mathbf{x}_K)$
- The samples are taken **one at a time** and input to the sequential detection processor
- After each sample, the **likelihood ratio** calculator computes:

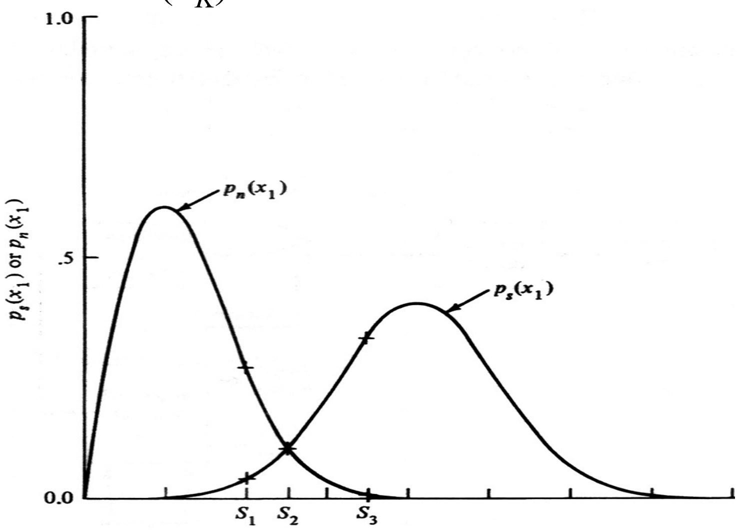
$$\Lambda(x_1, x_2, \cdots, x_K) = \frac{p_s(\mathbf{x}_K)}{p_n(\mathbf{x}_K)}$$

Sequential Detection (Cont.)

- The likelihood ratio is input to the sequential detection logic and compares with an **upper** threshold A and a **lower** threshold B, where $A > B$
  - The A and B are selected to achieved a specific reliability
  - If $\Lambda(\mathbf{x}_K) > A$: the signal is declared **present** and the **test ends**
  - If $\Lambda(\mathbf{x}_K) < B$: the signal is declared **absent** and the **test ends**
  - If $B < \Lambda(\mathbf{x}_K) < A$:
    - **No decision is made** about the present or absent of the signal
    - The test is **continuous** by taking another sample and calculating another likelihood ratio

Sequential Detection (Cont.)

- The likelihood ratio indicates whether the sequence of samples is more likely to have resulted from either
  - Signal and noise: $\Lambda(\mathbf{x}_K) > 1.0$, or
  - Noise alone: $\Lambda(\mathbf{x}_K) < 1.0$



Sequential Detection (Cont.)

- According to Fig. 5-20: $p_s(x_1)$ and $p_n(x_1)$
  - For sample S_1: $\Lambda(x_1) << 1.0$,
  - For sample S_2: $\Lambda(x_1) = 1.0$
  - For sample S_3: $\Lambda(x_1) >> 1.0$
- Assume that the samples of the envelope detector output are spaced sufficiently in time
  - They may be considered **independent**

  $\Rightarrow$ The joint pdf's are **products** of single sample pdf's

$$p_s(\mathbf{x}_K) = \prod_{k=1}^{K} p_s(x_k)$$

$$p_n(\mathbf{x}_K) = \prod_{k=1}^{K} p_n(x_k)$$

Sequential Detection (Cont.)

$$\Lambda(\mathbf{x}_K) = \prod_{k=1}^{K} \lambda(x_k)$$

- where

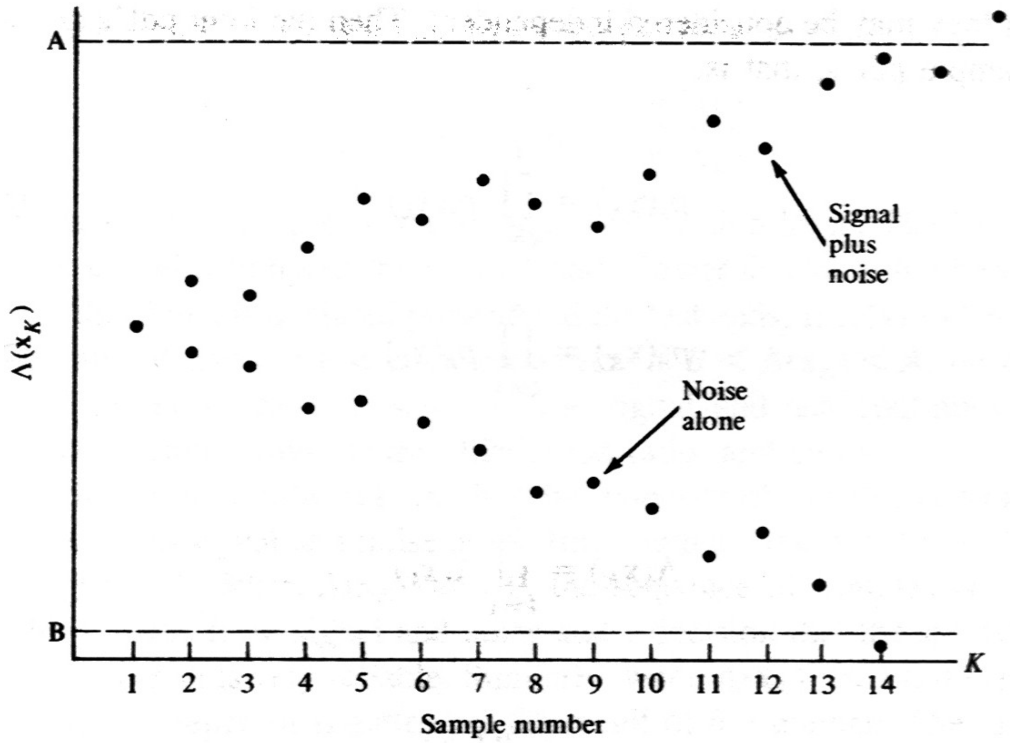
$$\lambda(x_k) \triangleq \frac{p_s(x_k)}{p_n(x_k)}$$

- When signal is **present** at the **design point SNR**
  - Envelope detector samples usually result in $\lambda(x_k) > 1.0$

  $\Rightarrow$ The product $\Lambda(\mathbf{x}_K)$ grows with increasing K
  - As $K \to \infty$, $\Lambda(\mathbf{x}_K)$ will always cross the upper threshold
- When **no signal** is present
  - Envelope detector samples usually result in $\lambda(x_k) < 1.0$

  $\Rightarrow$ The product $\Lambda(\mathbf{x}_K)$ approaches zero as K increases
  - As $K \to \infty$, $\Lambda(\mathbf{x}_K)$ will always cross the lower threshold

Sequential Detection (Cont.)



Sequential Detection (Cont.)

- When signal is indeed present:
  - A **missed detection** occurs whenever $\Lambda(\mathbf{x}_K)$ crosses the lower threshold before crossing the upper threshold
- When no signal is present:
  - A **false alarm** occurs whenever $\Lambda(\mathbf{x}_K)$ crosses the upper threshold before crossing the lower threshold

Thresholds and Error Probability

- Assume that the functions $p_s(\mathbf{x}_K)$ and $p_n(\mathbf{x}_K)$ are known for a particular SNR
- Let Γ_1 denote the set of all vectors that result in **a decision that signal is present**
  - For every $\mathbf{x}_K \in \Gamma_1$

$$A \le \frac{p_s(\mathbf{x}_K)}{p_n(\mathbf{x}_K)}$$

- When the bandpass filter output is **noise alone**, a false alarm occurs for any $\mathbf{x}_K \in \Gamma_1$

$$P_{fa} = \int_{\mathbf{x}_K \in \Gamma_1} p_n(\mathbf{x}_K)\, d\mathbf{x}_K$$

Thresholds and Error Probability (Cont.)

- When the bandpass filter output is **signal plus noise**, a correct detection occurs for any $\mathbf{x}_K \in \Gamma_1$

$$P_d = \int_{\mathbf{x}_K \in \Gamma_1} p_s(\mathbf{x}_K)\, d\mathbf{x}_K$$

$$A \le \frac{p_s(\mathbf{x}_K)}{p_n(\mathbf{x}_K)} \Rightarrow p_s(\mathbf{x}_K) \ge A p_n(\mathbf{x}_K)$$

$$\int_{\mathbf{x}_K \in \Gamma_1} p_s(\mathbf{x}_K)\, d\mathbf{x}_K \ge A \times \int_{\mathbf{x}_K \in \Gamma_1} p_n(\mathbf{x}_K)\, d\mathbf{x}_K$$

$$P_d \ge A P_{fa}$$

Thresholds and Error Probability (Cont.)

- Let Γ_2 denote the set of all vectors that result in **a decision that signal is absent**
  - For every $\mathbf{x}_K \in \Gamma_2$

$$\frac{p_s(\mathbf{x}_K)}{p_n(\mathbf{x}_K)} \le B$$

- When the bandpass filter output is **noise alone**, a correct dismissal occurs for any $\mathbf{x}_K \in \Gamma_2$

$$1 - P_{fa} = \int_{\mathbf{x}_K \in \Gamma_2} p_n(\mathbf{x}_K)\, d\mathbf{x}_K$$

Thresholds and Error Probability (Cont.)

- When the bandpass filter output is **signal plus noise**, a missed detection occurs for any $\mathbf{x}_K \in \Gamma_2$

$$1 - P_d = \int_{\mathbf{x}_K \in \Gamma_2} p_s(\mathbf{x}_K)\, d\mathbf{x}_K$$

$$\frac{p_s(\mathbf{x}_K)}{p_n(\mathbf{x}_K)} \le B \Rightarrow p_s(\mathbf{x}_K) \le B p_n(\mathbf{x}_K)$$

$$\int_{\mathbf{x}_K \in \Gamma_2} p_s(\mathbf{x}_K)\, d\mathbf{x}_K \le B \times \int_{\mathbf{x}_K \in \Gamma_2} p_n(\mathbf{x}_K)\, d\mathbf{x}_K$$

$$1 - P_d \le B\left(1 - P_{fa}\right)$$

Thresholds and Error Probability (Cont.)

- For a pair of desired P_d and P_{fa}, we have $A \le P_d / P_{fa}$
- If a smaller A is applied, we have a larger set of Γ_1
  - A larger P_d and a larger P_{fa} are obtained
  - Not all $A \le P_d / P_{fa}$ can achieve the desired P_d and P_{fa}

Only achieve the desired ratio

- For a pair of desired P_d and P_{fa}, we have $B \ge (1 - P_d) / (1 - P_{fa})$
- If a larger B is applied, we have a larger set of Γ_2
  - A larger $(1 - P_d)$ and a larger $(1 - P_{fa})$ are obtained
  - A smaller P_d and a smaller P_{fa} are obtained
  - Not all $B \ge (1 - P_d) / (1 - P_{fa})$ achieve the desired P_d and P_{fa}
- To achieve the desired P_d and P_{fa}, we make the equality hold

$$P_d = AP_{fa} \qquad 1 - P_d = B\left(1 - P_{fa}\right)$$

Thresholds and Error Probability (Cont.)

- To find the thresholds A and B
  - Functions of P_d and P_{fa}

$$P_d = AP_{fa} \Rightarrow A = P_d / P_{fa}$$

$$1 - P_d = B\left(1 - P_{fa}\right) \Rightarrow B = \left(1 - P_d\right) / \left(1 - P_{fa}\right)$$

- To determine the P_d and P_{fa}:
  - Functions of A and B

$$P_{fa} = \frac{1 - B}{A - B}$$

$$P_d = A\left(\frac{1 - B}{A - B}\right)$$

Synchronization by Using Matched Filters

Synchronization Using a Matched Filter

- In all of the synchronization schemes discussed above, the **duration of the correlation** is a function of:
  - The desired acquisition performance (P_d, P_{fa})
  - The received signal-to-noise ratio
  - The detection strategy (fixed integration time, multiple-dwell, or sequential detection)
- Assume that the duration of the correlation is $T_i = KT_c$
  - K may be from 10 to several thousand
- The spreading code phase is declared correct or incorrect after the correlation
- If the code phase is incorrect, the synchronization processor steps to another phase for evaluation

Synchronization Using a Matched Filter (Cont.)

- The average synchronization time, given a code phase uncertainty of M chips, is approximately

$$\overline{T}_s = \frac{2M+1}{2} T_i = \frac{2M+1}{2} K T_c \approx M K T_c$$

  - where it is assumed that $P_d = 1.0$, $P_{fa} = 0.0$ and the serial-search step size is ½ chips $\Rightarrow C = 2M$

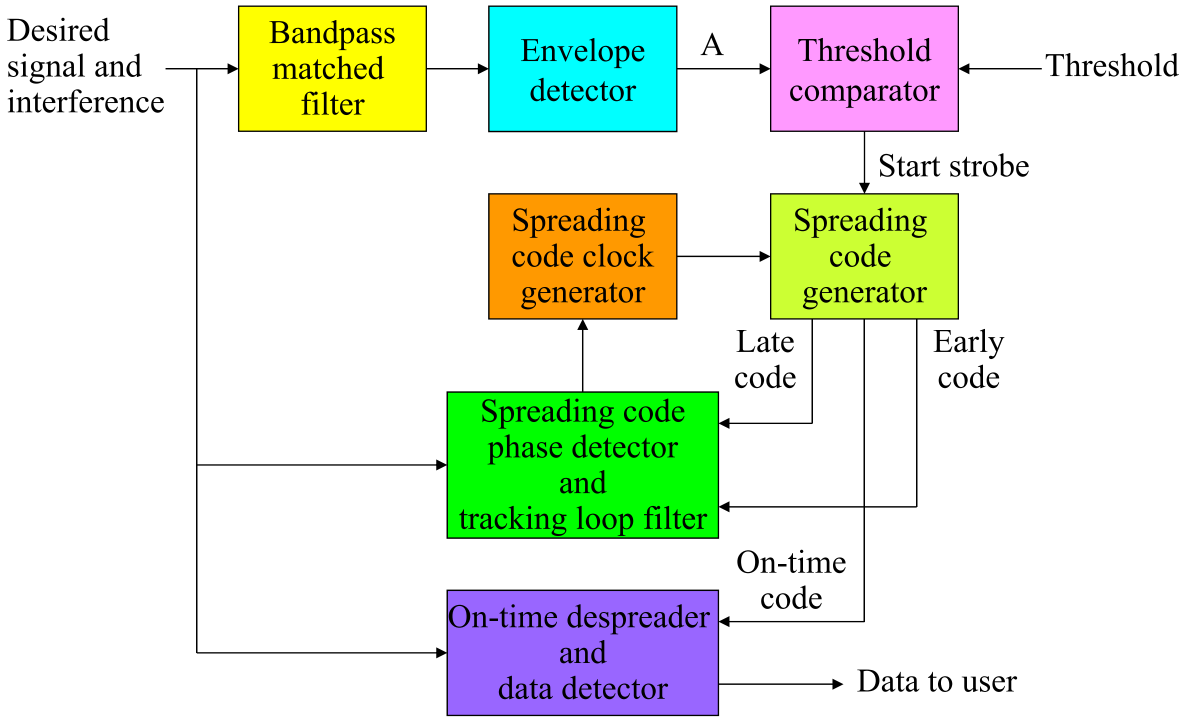
$$T_{da} = \left(1 - P_{fa}\right) T_i + P_{fa}\left(T_i + T_{fa}\right) = T_i + T_{fa} P_{fa} = T_i$$

$$\boxed{\begin{aligned} \overline{T}_s &= (C-1) T_{da} \left(\frac{2 - P_d}{2 P_d}\right) + \frac{T_i}{P_d} \\ &= \frac{2M-1}{2} T_i + T_i = \frac{2M+1}{2} T_i \end{aligned}}$$

Synchronization Using a Matched Filter (Cont.)

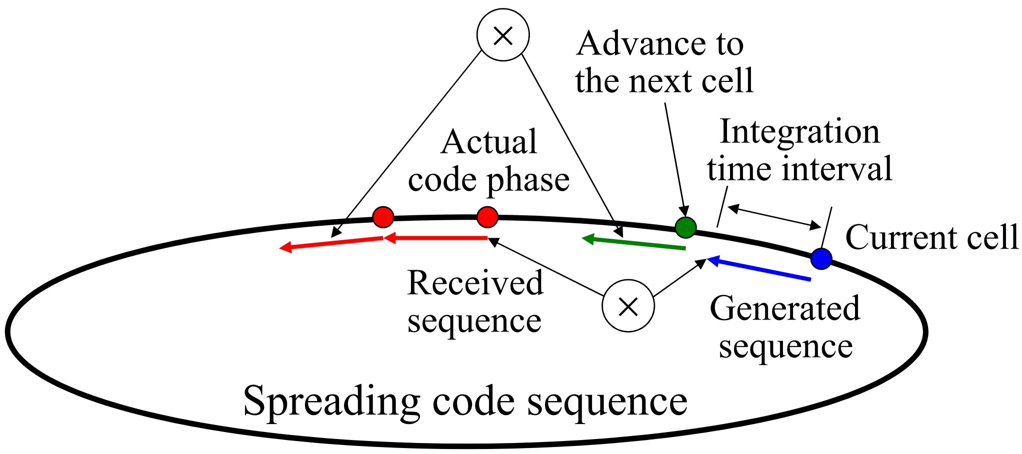
- If the time to evaluate each phase cell can be reduces, the average synchronization time can be reduces
- The matched-filter synchronizers reduce the time required to evaluated a phase cell from KT_c to approximately T_c
- Fig. 5-31 is a top-level block diagram of a **DSSS receiver**
  - The received signal is input to the spreading **code tracking loop** and to the on-time despreader for **data detection**
  - The received signal is input to a bandpass filter (matched to **a segment of the spreading waveform**)
  - When a matched spreading waveform is received, the matched filter produces a **output pulse**
  - The spreading code generator is **started at the correct phase** when the matched filter output pulse is sensed

Synchronization Using a Matched Filter (Cont.)



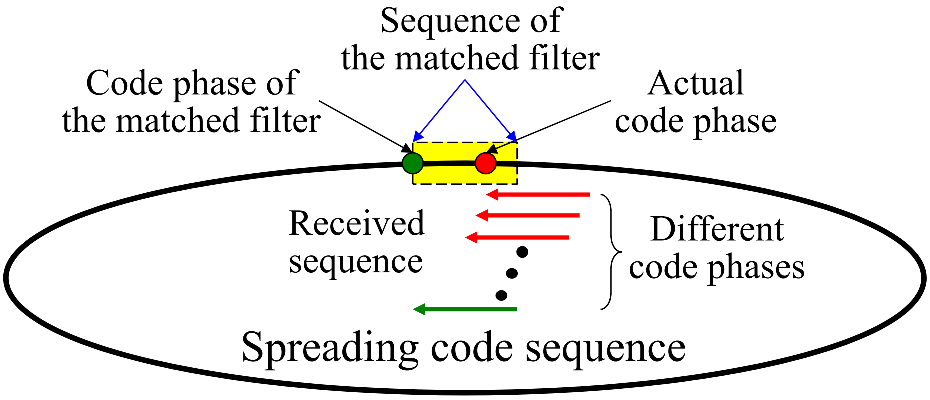
Comparison: Serial Search Synchronizer

- For serial search synchronizers:
  - Use **active** correlation and energy detection
  - Each received chip is used in the evaluation of **a single code phase** in stepped serial search



Synchronization Using a Matched Filter (Cont.)

- For matched-filter synchronizers:
  - Uses **passive** correlation within the filter
  - Each received chip is used in the evaluation of **many code phases**
- The flexibility of matched-filter synchronizers is poor



Synchronization Using a Matched Filter (Cont.)

- Suppose that the desired sequence of the matched filter is
  - "01110101"
- In the received code sequence:
  - Each received chip is used in the evaluation of **16 code phases**
  - The step size of a phase cell is assumed to be ½ chip

"… ": Sequence within filter

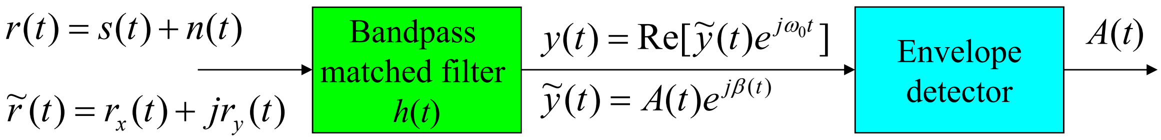
0111010"10010100"
0011101"01001010"
1001110"10100101"
1100111"01010010"
0110011"10101001"
0011001"11010100"
0001100"11101010"
0000110"01110101"

Time

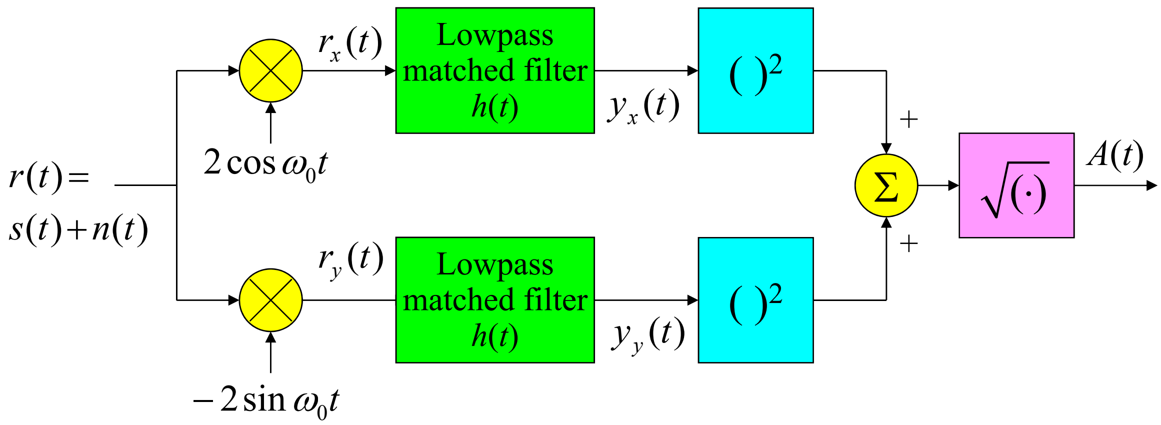
Is there any limitation on the sequence period?
The sequence period *N* cannot be too large!

- The implementation of matched filter includes bandpass or lowpass matched filter
  - These two implementations are equivalent and produce identical outputs when the inputs are identical

Matched Filter Implementations



Bandpass



Lowpass

Digital Implementations of Matched Filter