STANFORD

WELCOME TO EE379A !!

Data Transmission Design

2024 Winter Quarter

JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2024

STANFORD

Lecture 1

Discrete Message Encoding/Decoding

January 9, 2024

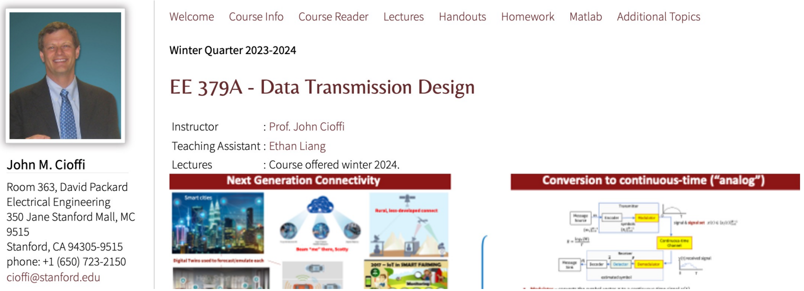
JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2024

Announcements & Agenda

- Announcements
  - People Introductions
  - Web site EE379A (see also canvas)
  - Chapters 1-8 on-line at this class web site (Course Reader)
  - Sections 1.1 and 1.2 today
  - See last/backup slide today for matlab toolboxes I use
  - Qualcomm and/or Samsung internships (send me email)
    - Apple, Intel, Ericsson



- Today
  - Course introduction
  - Discrete Data Transmission & Examples (1.1)
  - Modulation and Demodulation (1.2)
  - Detection(1.1.2)
  - Vector Channels (1.1.1)
    - Detection and Decision Regions (1.1.3)

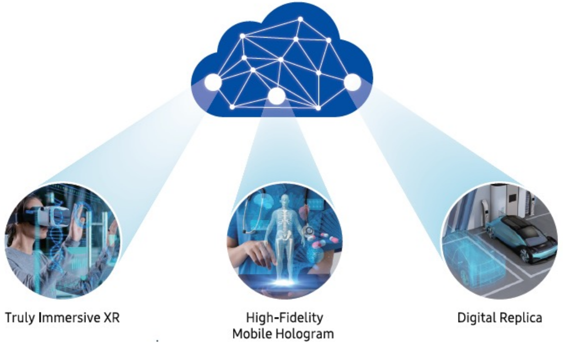
- Problem Set 1 = PS1 due Wednesday Jan 17 at 17:00
  1. 1.1 a constellation
  2. 1.2 inner products
  3. 1.3 same constellation, different modulators
  4. 1.7 Irrelevancy and decision regions
  5. 1.10 constellation invariances

Intro: Why Communications?

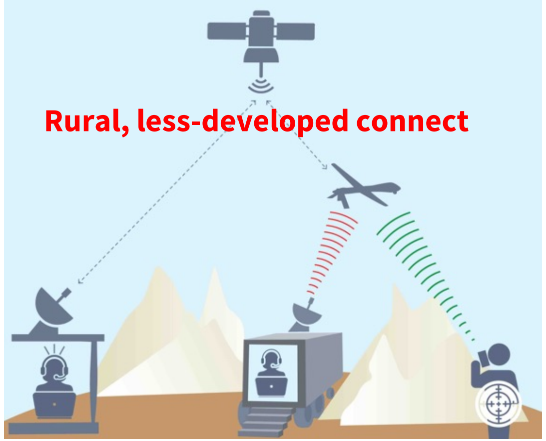
This fundamental area is always needed.
Many decades: it surges at times

If you love it, it is absolutely great area to enjoy working for your decades ahead.

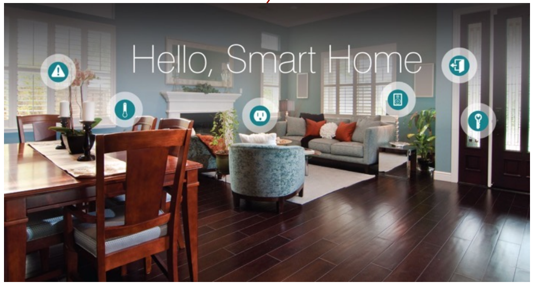
Next Generation Connectivity

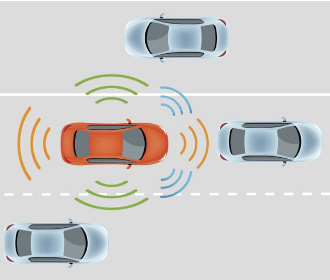


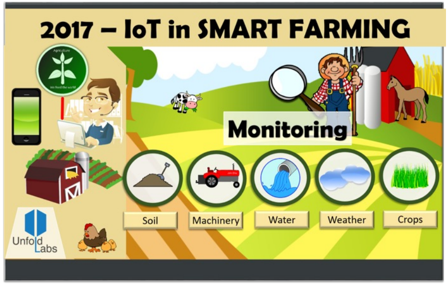
Beam “me” there, Scotty

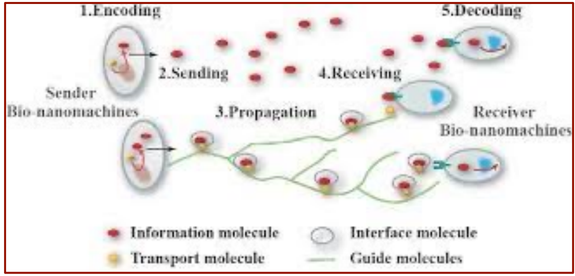


Digital Twins used to forecast/emulate each







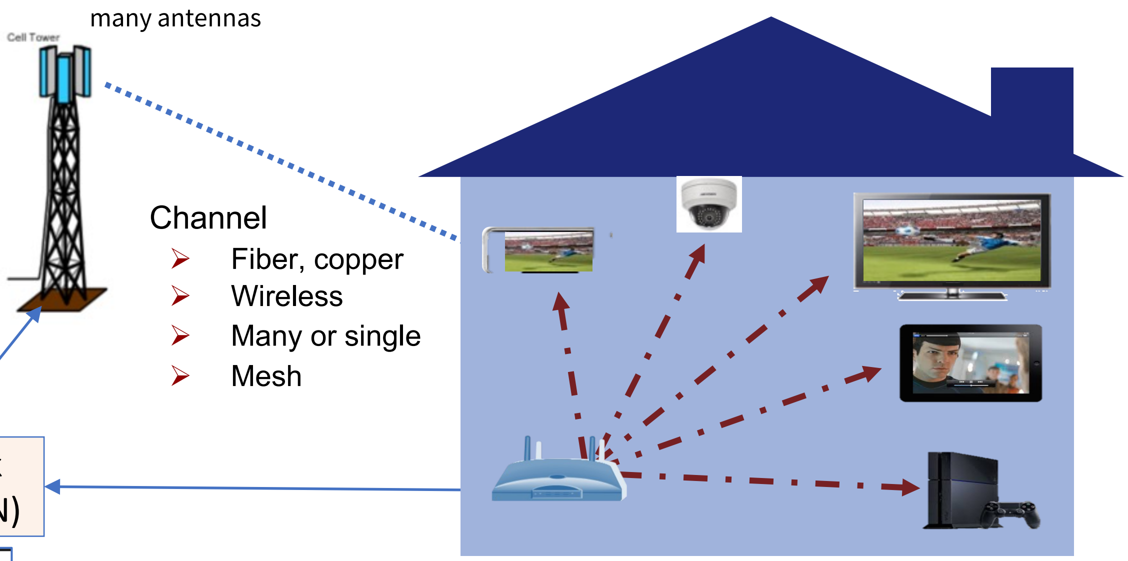


Defense (“5/6G.mil”)
Clothing, Computing, …

Samsung: 6G “hyper connected”

Broadband Internet Access ($1.5T/year)

- Messages
  - Internet
  - Email
  - Text
  - video, audio
  - Sensor/camera images



(almost) All signal processing, modulation, and coding done in software at edge with shared computing facility

OSI Model

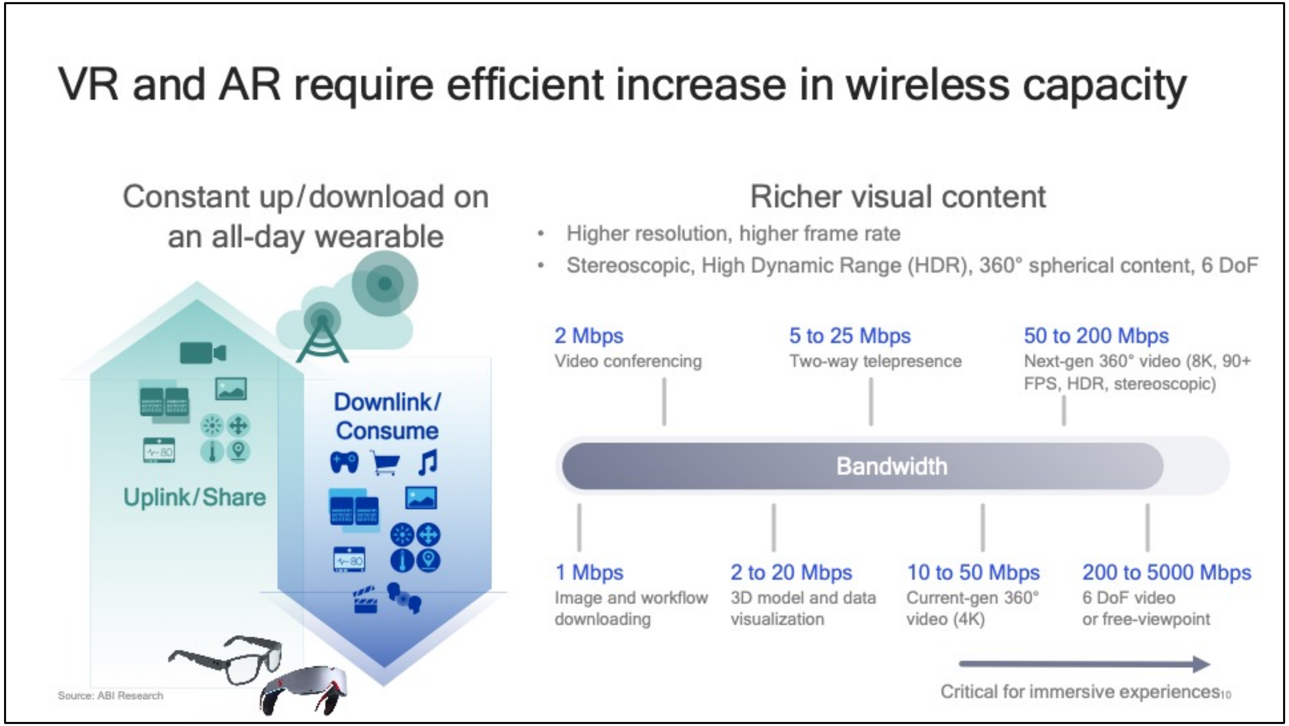
	Layer	Function	Example Protocols
7	Application Layer	network process to application	HTTP, SFTP, SSH
6	Presentation Layer	data representation & encryption	XML, JSON
5	Session Layer	interhost communication	Mostly theoretical
4	Transport Layer	end-to-end connections & reliability	TCP, UDP
3	Network Layer	path determination & logical addressing	IP Addresses
2	Data Link Layer	physical addressing	MAC Addresses
1	Physical Layer	medial signal & transmition	Ethernet, Bluetooth, Wireless

this class

OSI = **Open** Systems Interconnect

SDN = **Software** Defined Network

Metaverse - VR/AR focus and Bandwidth



Latency:

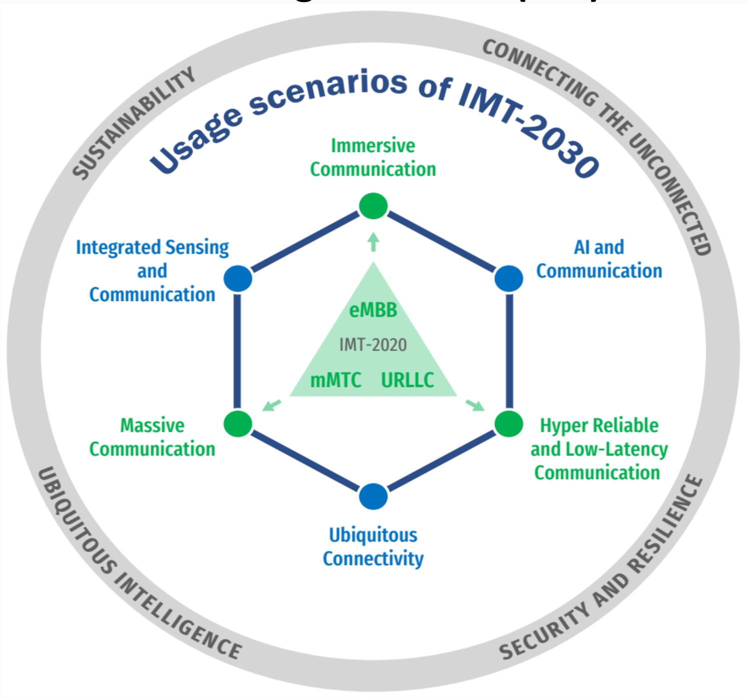
Edge ~ 1 ms

ISP Cloud 20-50 ms

Public Cloud 100 ms

International Mobile Telecommunications (IMT)

6 Usage Scenarios (3+3)

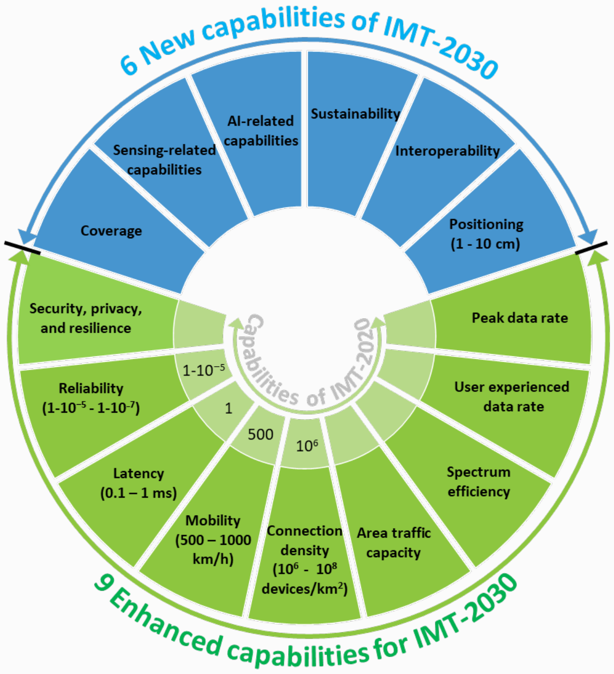


"Wheel diagram"

IMT-2030

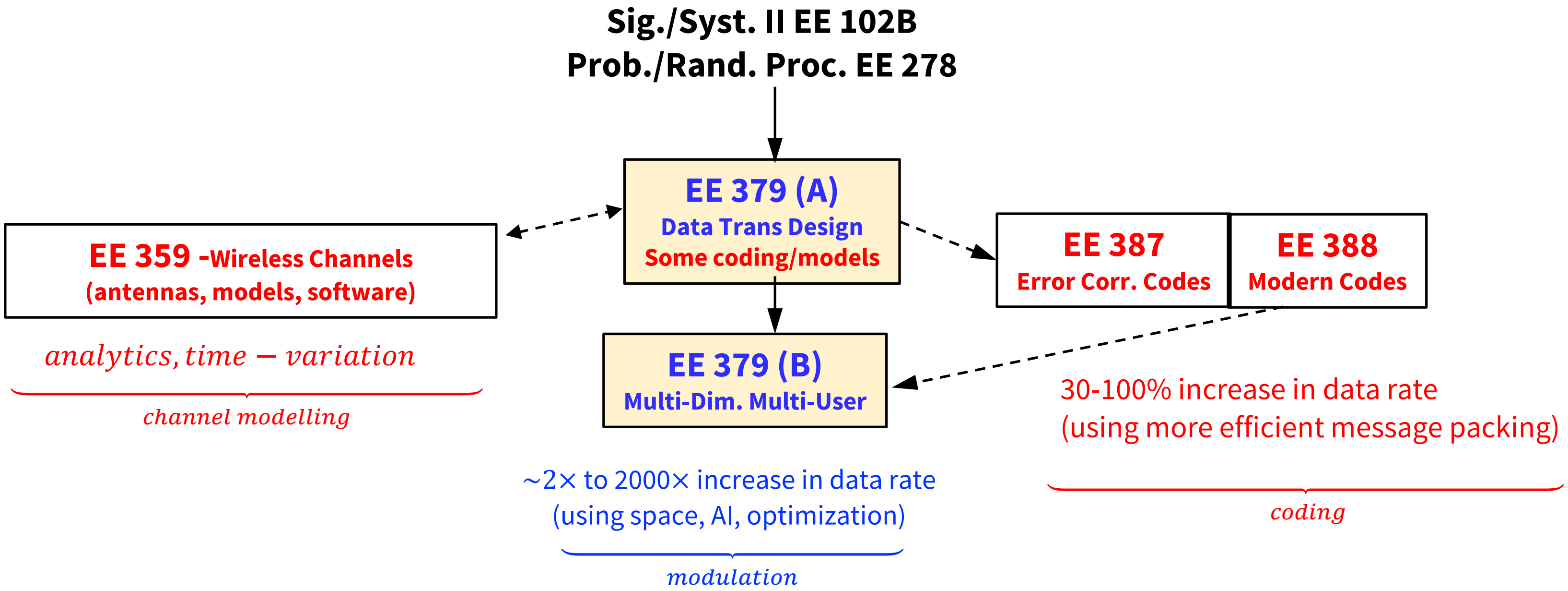
ITU-R M.2160-0 (11/2023)

15 Capabilities (9+6)



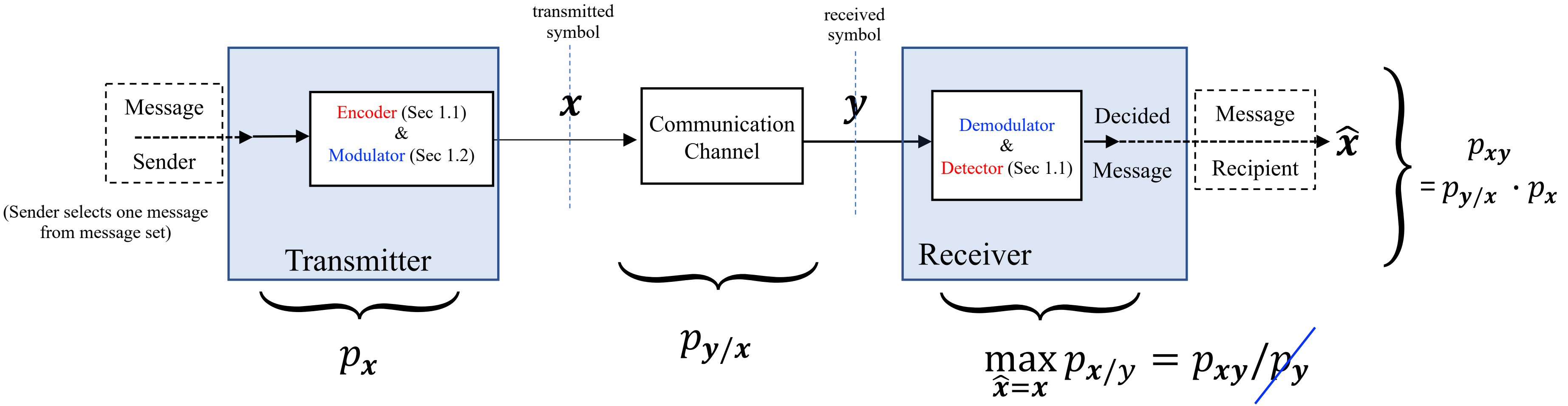
"Palette diagram"

Communications Depth Sequence



- **Modulation and Coding** compliment one another
  - Modulation = energy assignment to time/frequency/space, is separate from:
  - Coding = distinct message mapping.
  - If both done well, they separate.

Basic Communication (digital)



- The symbol $\boldsymbol{x}$ and messages are in a 1-to-1 relationship.
- Finding the best $\hat{\boldsymbol{x}}$ and designing $\boldsymbol{x}$ well → this class.
- Most general channel is represented by the conditional probability $p_{y/x}$.
- Partial source description is p_x - together, p_{xy} . (Mapping to $\boldsymbol{x}$ is important.)
- Optimum detector (minimizes ave error probability) is Maximum a Posteriori (MAP), max $p_{x/y} \sim p_{xy} = p_{y/x} \cdot p_x$. (1.1.2.1)
  - When input distribution is uniform (constant) → ML (maximum likelihood), $\max_{x} p_{y/x}$. (1.1.2.2)

EE 379A is single x user (time-domain) ;

EE 379B expands to multiuser, MIMO, multicarrier

3 Basic Problems to Solve

- **CHANNEL IDENTIFICATION** - what is $p_{\boldsymbol{y}/\boldsymbol{x}}$?

- **CODING & MODULATION** - What are good (best) $\boldsymbol{x}$ and $p_{\boldsymbol{x}}$ for a given channel?

- **DETECTION** – What is a good (best) receiver for deciding which $\boldsymbol{x}$?

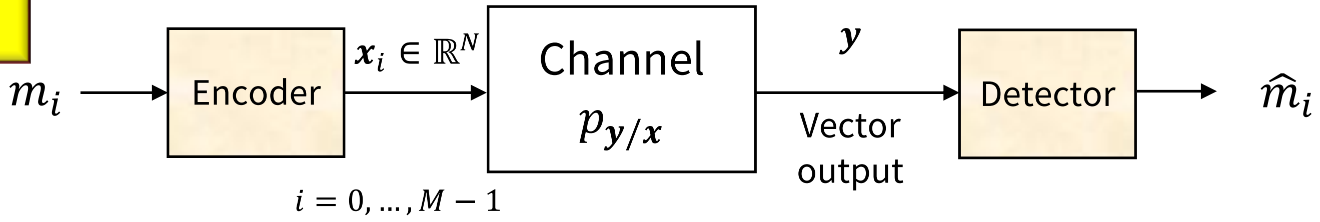
Discrete Data Transmission & Examples

Section 1.1.1

Discrete Message Transmission

Temporarily omit modulator/demodulator

$M = 2^b$
possible messages



Channel ∈ {fiber, copper, wireless, disk/SSD, flesh, qubit, … }

- Designers identify a channel's vector conditional-probability description $p_{\boldsymbol{y}/\boldsymbol{x}}$.
  - This allows a common design framework across different channel types/models.
- m_i, one of $M < \infty$ discrete messages (a vector "**symbol**" $\boldsymbol{x}$), is sent every **symbol period** of T seconds.
  - Messages are represented by symbol vectors, 1-to-1 mapping from messages.
  - Later $m_i \rightarrow m$.
  - $b = \log_2(M) =$ **number of bits (in message).**
- Conversion to/from actual (often continuous-time "analog") is "**modulation,**"
  - and left unspecified in this general discrete-message-transmission model.
  - Many different modulation methods can conform to the same discrete message-transmission system.

Transmitted-Symbol Quantities

Name	Definition
$M = 2^b \in \mathbb{Z}^+$	**number of messages**, corresponding to $b \geq 0$ information bits
$T \in \mathbb{R}$	**symbol period**; $1/T$ is the **symbol rate**
$R = \frac{b}{T} \in \mathbb{R}$	**data rate**.
$\boldsymbol{x}$	transmitted **symbol** value (typically a real or complex vector)
$C \subset \mathbb{R}^N$	**constellation** that consists of all possible symbol values $\{\boldsymbol{x}_i \; i = 0, ...\lvert C \rvert - 1\}$ with $\lvert C \rvert \geq M$ and in this chapter, always equal.
$p_m(i) \in \mathbb{R}$	**message's probability distribution**, $i = 0, ..., M-1$
$p_{\boldsymbol{x}}(i) \in \mathbb{R}$	**symbol value's probability distribution** $= p_m(i)$
$N \in Z^+$	**number** of real temporal **dimensions** per transmit symbol the receive symbol has $N + \nu$ dimensions with default $\nu = 0$ and $\nu \in \mathbb{Z}$
$L_x \in \mathbb{Z}^+$	number of **transmit spatial dimensions**
$L_y \in \mathbb{Z}^+$	number of **receive spatial dimensions**

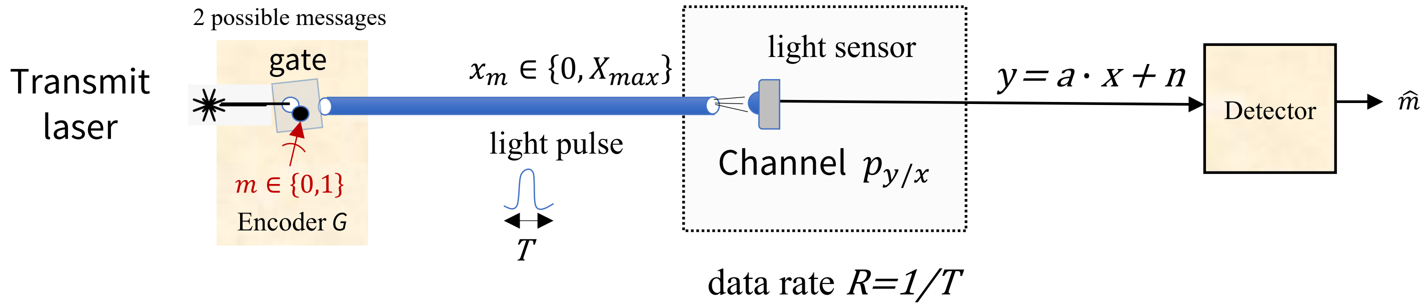
Table 1.1: Table of transmitted-symbol quantities' definitions.

- Derived: **Energy** $\mathcal{E}_x = \mathbb{E}[\|\boldsymbol{x}\|^2] = \sum_{i=1}^{64} p_x(i) \cdot \|\boldsymbol{x}_i\|^2$

  Power $P_x = \frac{\mathcal{E}_x}{T}$ *1.1.1.2*

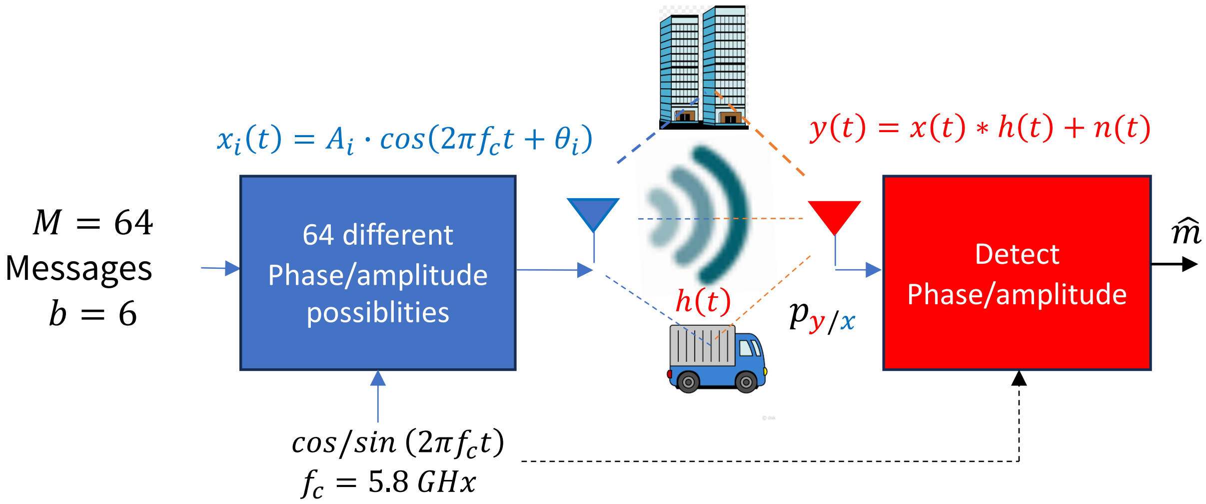
dB (decibels) are measured as
$10 \cdot \log_{10}$ *energy or power*
$= 20 \cdot \log_{10}$ *amplitude*
often relative to 1 mW (dBm)

Example 1: Light-Pulse Bit Transmission



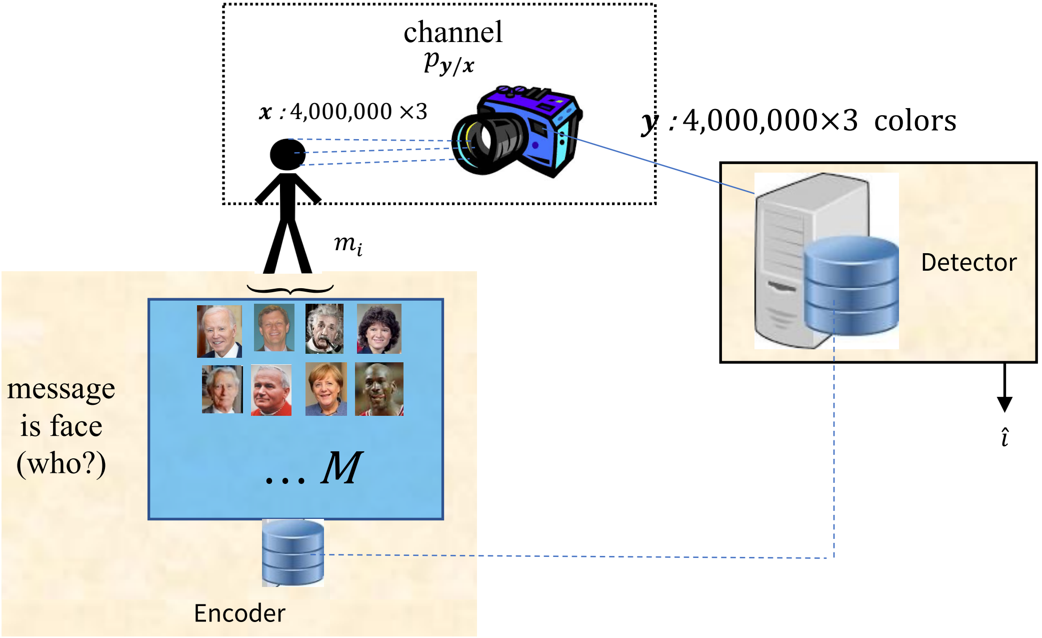
- On/off - $M = 2$ and $b = 1$ bit.
- If $T = 10\ ps$, then $R = 100$ Gbps (10^{11}) bits/second.
- $x \in \{0\ , X_{max}\}$.
- “non-coherent” or “direct-detection” optical transmission.

Example 2: Wireless



- Channel adds noise and has multiple paths that can align positively or negatively.
- Discrete modulator $\boldsymbol{x} = A \cdot e^{j\theta}$ **(64 complex vectors, one for each message)**; while $\boldsymbol{y} = H \cdot \boldsymbol{x} + \boldsymbol{n}$.
- $^1/_T = 100\ MHz$, so $R = 600\ Mbps$.

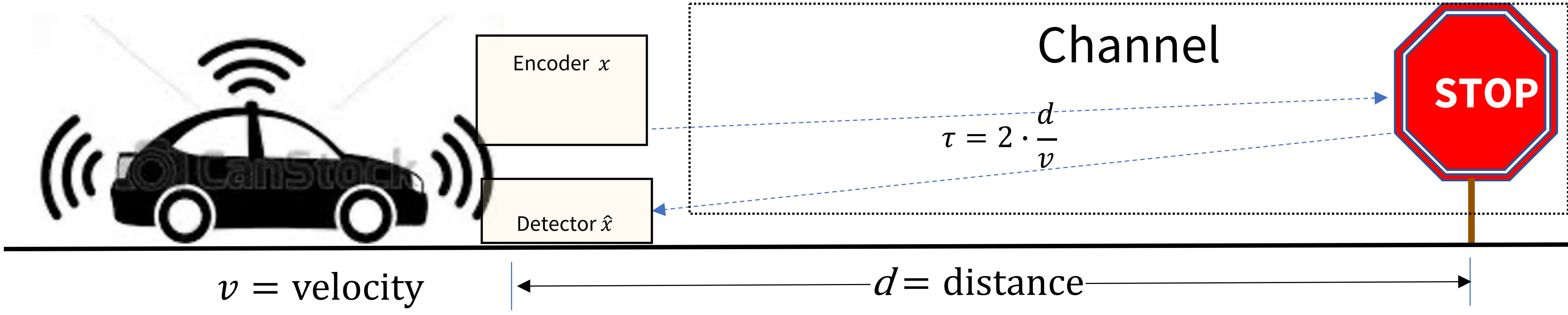
4 megapixel photograph



- Number of dimensions $N = 2280 \times 1640 \times 3$ (exactly 11,217,600).
- $M = 8$; $b = 3$.
- $R = 3$ bits/photo, where $T = 1$ photo.

Today's machine-learned facial ID reduces to this basic system, with perhaps large M

distance measurement (radar, lidar)

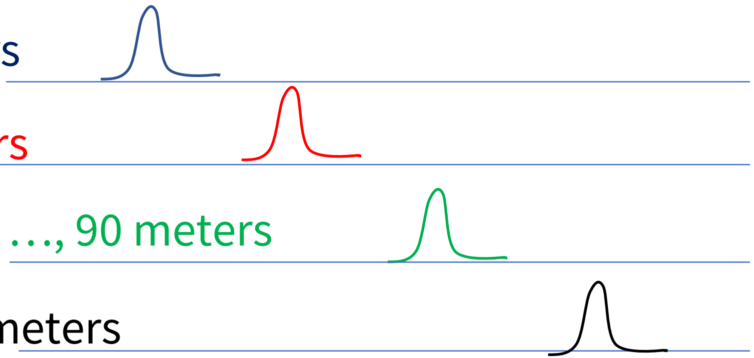


m_0 =10 meters

m_1 = 20 meters

$m_{i=2\ldots10}$ = 20, …, 90 meters

m_{M-1} = 100 meters



lidar pulse position is the message, equivalent to distance where decoder maps τ to the message index or d

$$\boldsymbol{x}_m = [x(0) \quad x(T') \quad \ldots \quad x([M-1]T')]$$

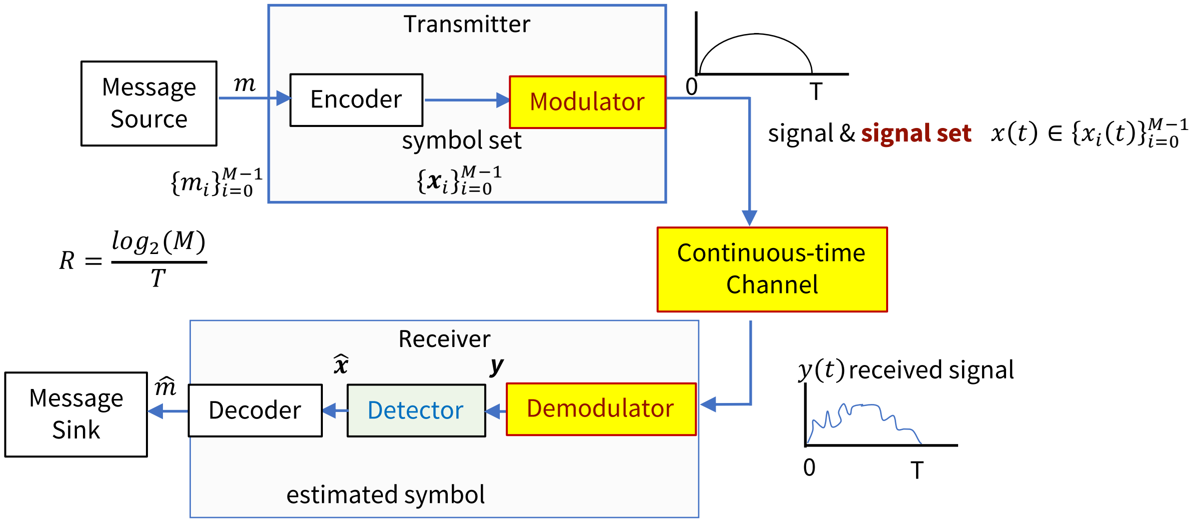
Radar, lidar, …., sensors
Are basic com systems

Modulation and Demodulation

(a foundation: Sections 1.1-1.2)

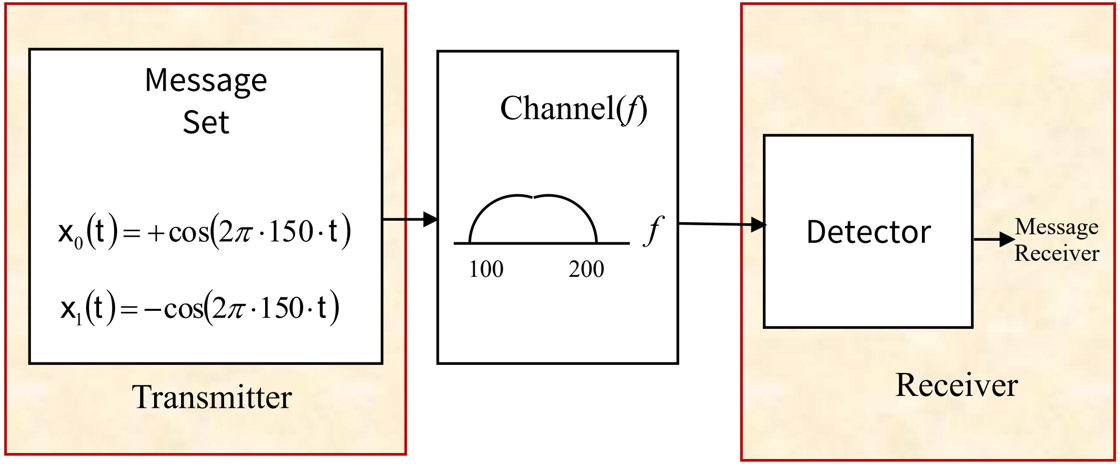
See PS1.1 and PS1.2

Conversion to continuous-time ("analog")



- **Modulator** – converts the symbol vector $\boldsymbol{x}$ to a continuous-time signal $x(t)$.
- **Demodulator** - reverses continuous time channel output $y(t)$ to the vector $\boldsymbol{y}$.
- **Detector** - decides which symbol decision $\hat{\boldsymbol{x}}$ minimizes the error probability that $\hat{\boldsymbol{x}} \neq \boldsymbol{x}$.

Example BPSK (wireless revisited)



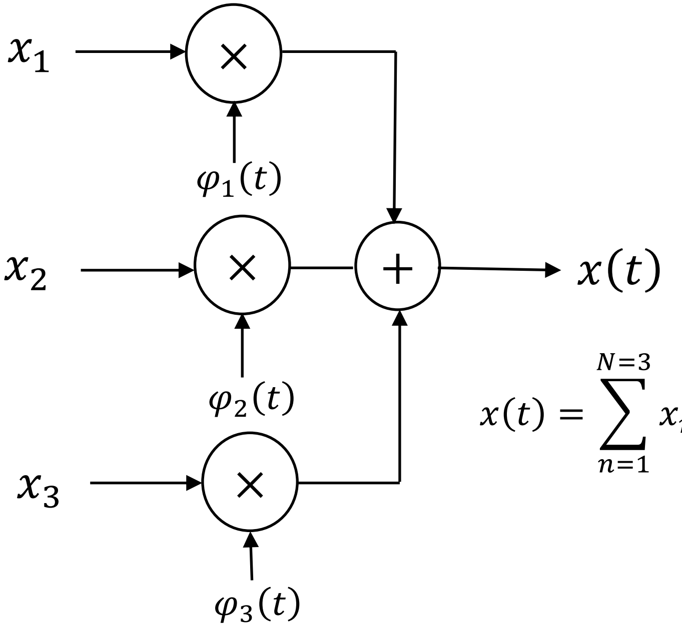
- Simple ± 1 inputs are not good for this channel, why?
- The modulator tries to match the signal vector to the channel;
  - while the discrete optical-pulse channel appears similar, it has a different modulator.
- However, designers may apply the same analysis to both, as we'll see.

Formal Modulation – math model

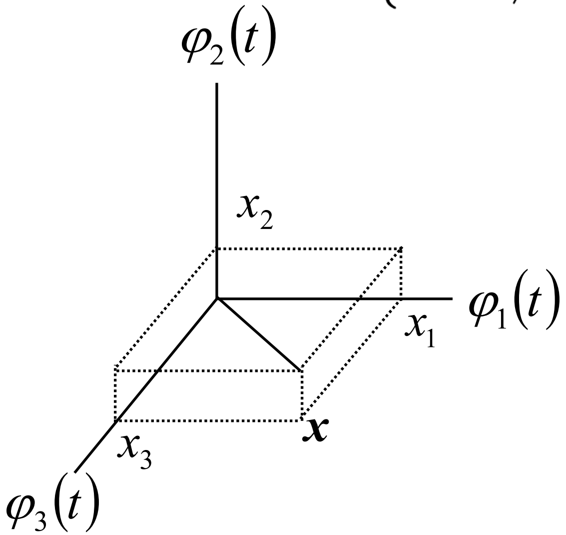
Modulator's basis functions $\varphi_n(t)$

$$\int_{-\infty}^{\infty} \varphi_m(t) \cdot \varphi_n(t) dt = \delta_{mn} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$



$$x(t) = \sum_{n=1}^{N=3} x_n \cdot \varphi_n(t)$$



- The basis functions match what the channel can pass well (in good designs).
- Symbol vector has a scalar component x_n for each dimension (one basis function/dimension, so N dimensions).
- The constellation $C = \{\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots \boldsymbol{x}_M\}$ (note set of *boldface* vectors, one for each message).

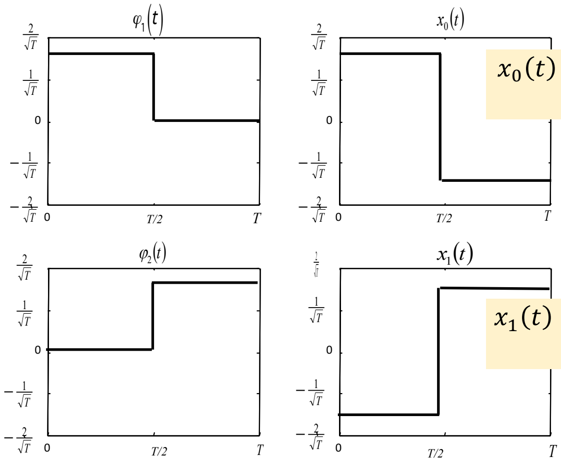
Binary Phase-Shift Keying (BPSK) and Manchester Examples

- **BPSK** with $T = 100ms$ (10 bits/second):

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cdot \cos\left(\frac{2\pi t}{T} + \frac{\pi}{4}\right) \quad 0 \le t \le T$$

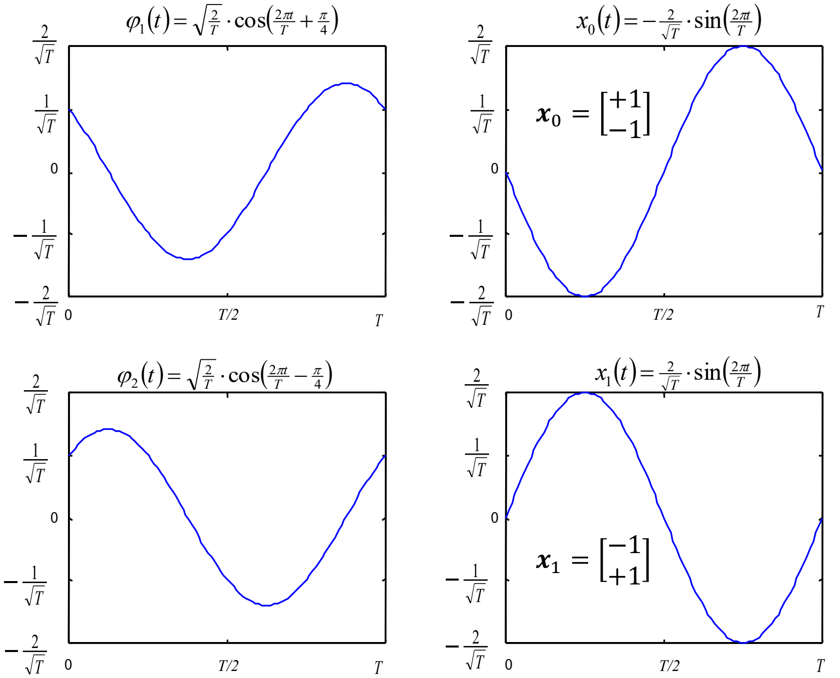
$$\varphi_2(t) = \sqrt{\frac{2}{T}} \cdot \cos\left(\frac{2\pi t}{T} - \frac{\pi}{4}\right) \quad 0 \le t \le T$$

- **Manchester Code** with $T = 100ms$ (10 bits/second) is better for simple channels with “binary” transmission:



$$x_0(t) = \underbrace{+1}_{x_{01}} \cdot \varphi_1(t) + \underbrace{(-1)}_{x_{02}} \cdot \varphi_2(t)$$

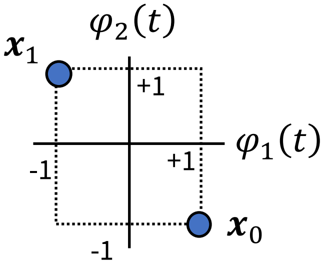
$$x_1(t) = \underbrace{-1}_{x_{11}} \cdot \varphi_1(t) + \underbrace{(+1)}_{x_{12}} \cdot \varphi_2(t)$$



Analysis will be same, but modulators differ
(because symbols $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are the same).

Common Constellations

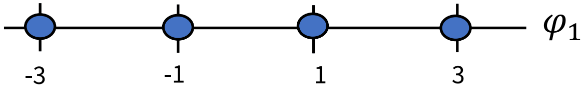
- Common constellation:



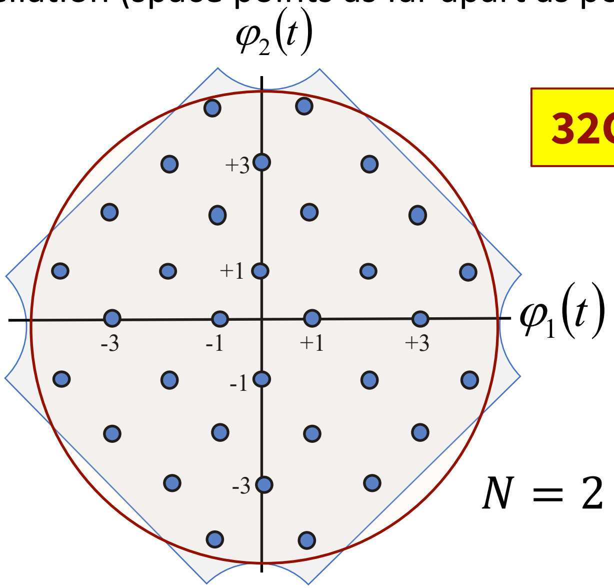
- All such systems' performance follows from the constellation (space points as far apart as possible within limited energy constraint).

- Other Examples:

4PAM or 2B1Q



$$N = 1 \; ; b = 2 \, ; M = 4$$



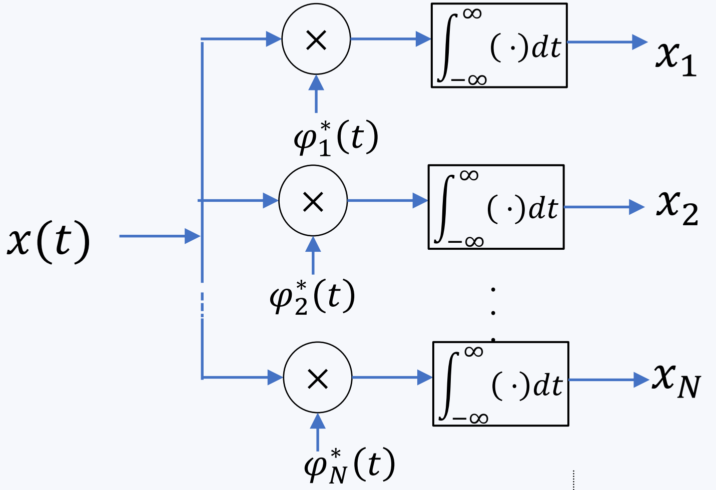
32CR or 32SQ

$$N = 2 \; ; b = 5 \, ; M = 32$$

Demodulator

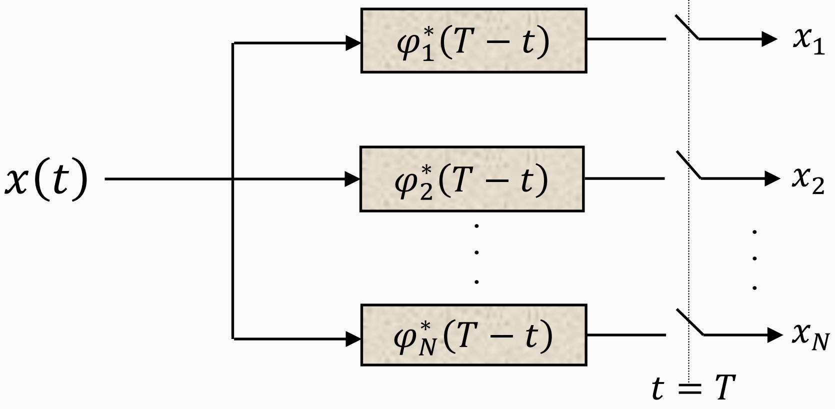
- **Correlative** demodulator

$$\int_{-\infty}^{\infty} x(t) \cdot \varphi_n^*(t) \cdot dt = x_n$$



Conjugate of real equals itself

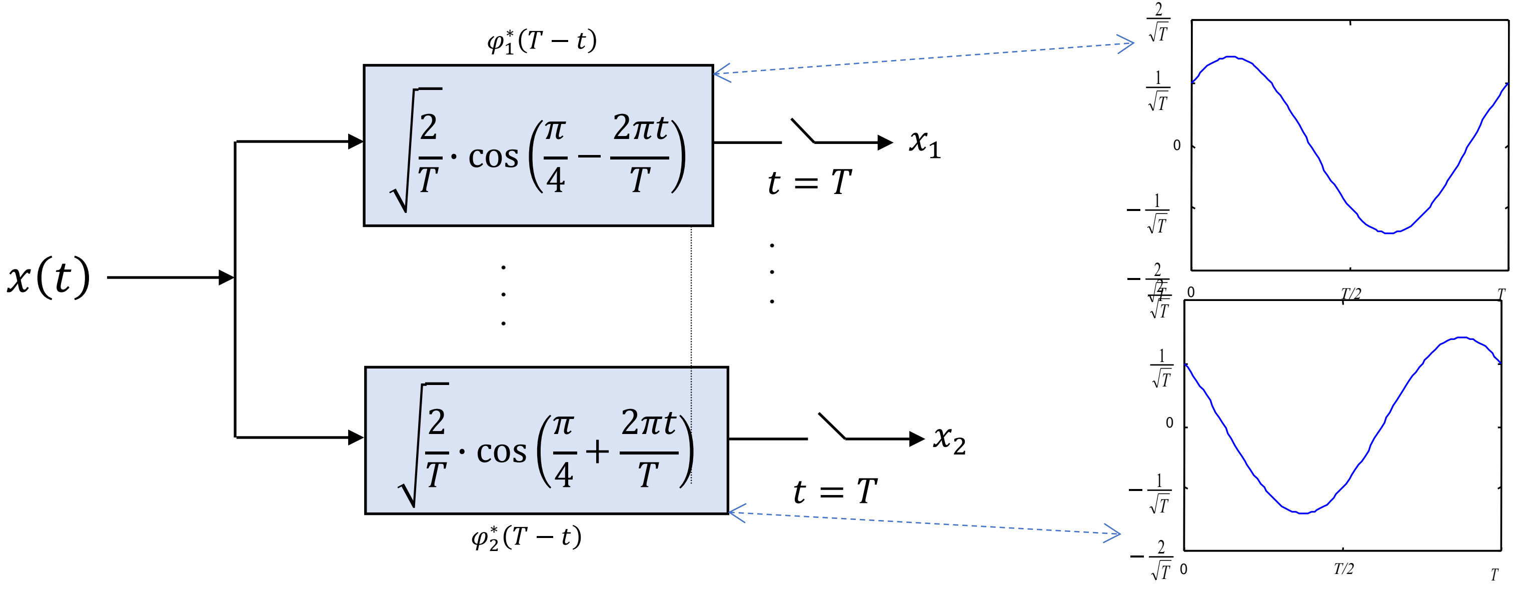
- **Matched-filter** demodulator



$$\int_{-\infty}^{\infty} x(t) \cdot \varphi_n^*(t) \cdot dt = x(t) * \varphi_n^*(T-t)|_{t=T}$$

BPSK matched-filter demodulator

- The BPSK modulator from slide 22 has corresponding matched-filter demodulator:



- The matched filter's convolution with original basis function produces unity norm at time T.
- The convolution is perfectly matched to maximize signal energy output at sampling time T.
- No other (unit-norm) filter sampled at any time can produce a larger SNR (see section 1.3.1.3 proof).

Inner Products

Definition 1.2.2 **[Inner Product]** *The* **inner product** *of two (real) functions of time* $u(t)$ *and* $v(t)$ *is*

$$\langle u(t), v(t)\rangle \triangleq \int_{-\infty}^{\infty} u(t) \cdot v(t) dt \ . \quad (1.41)$$

The inner product of two (real) vectors $\boldsymbol{u}$ *and* $\boldsymbol{v}$ *is*

$$u(t) = \sum_{n=1}^{N} u_n \cdot \varphi_n(t)$$

$$v(t) = \sum_{n=1}^{N} v_n \cdot \varphi_n(t)$$

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle \triangleq \boldsymbol{u}^* \boldsymbol{v} = \sum_{n=1}^{N} u_n \cdot v_n \ , \quad (1.42)$$

where $*$ *denotes vector transpose (and conjugate vector transpose later when complex signals are introduced).*

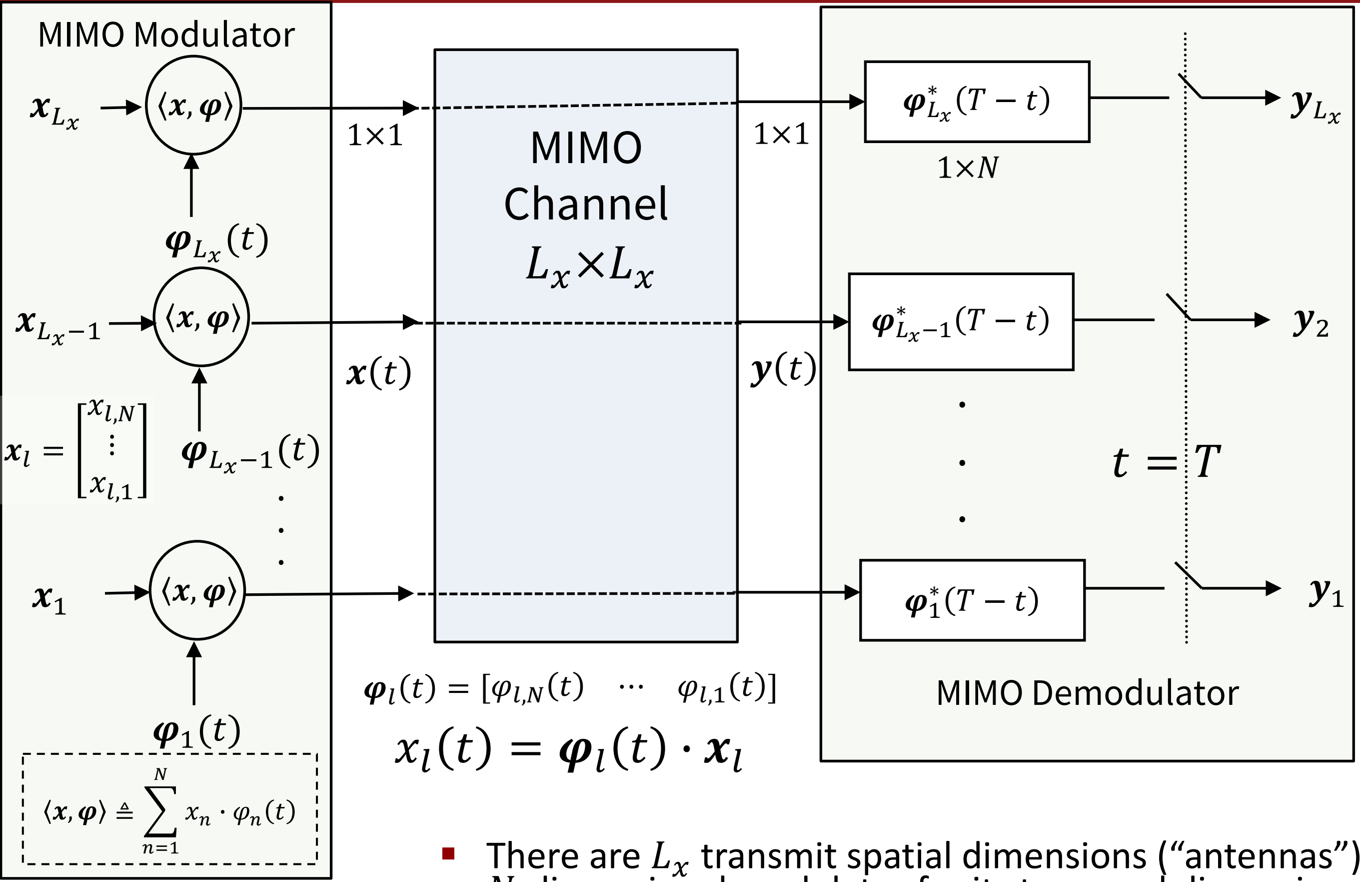
$$\|\boldsymbol{u}\|^2 = \langle \boldsymbol{u}, \boldsymbol{u}\rangle$$

- Easy to prove that $\langle u(t), v(t)\rangle = \langle \boldsymbol{u}, \boldsymbol{v}\rangle$
- And further that

Theorem 1.2.2 **[Parseval's Identity]** *The following relation holds true for any modulated waveform*

$$\mathcal{E}_{\boldsymbol{x}} = \mathbb{E}\left[||\boldsymbol{x}||^2\right] = \mathbb{E}\left[\int_{-\infty}^{\infty} x^2(t) dt\right] \ . \quad (1.48)$$

MIMO Channels



$$\Phi(t) = \begin{bmatrix} \varphi_{L_x}(t) & \dots & 0^* \\ \vdots & \ddots & \vdots \\ 0^* & 0^* & \varphi_1(t) \end{bmatrix}$$

$$x = \begin{bmatrix} x_{L_x} \\ \vdots \\ x_1 \end{bmatrix}$$

$$x(t) = \Phi(t) \cdot x$$

$$\langle \varphi_\ell^*(t), \varphi_\ell(t) \rangle = I \quad \forall\, \ell = 1, \dots, L_x$$

- There are L_x transmit spatial dimensions (“antennas”) and each has an N-dimensional modulator for its temporal dimensions.

Optimum Detection

Section 1.1.2

See PS1.3 (Prob 4.18), PS1.4 (Prob 4.7), and PS1.5 (Prob 4.25)

Minimum Error Probability

Definition 1.1.3 **[Error Probability]** *The* **Error Probability** *is*

$$P_e \triangleq P\{\hat{m} \neq m\} \quad .$$

The corresponding **probability of being correct** *is therefore*

$$P_c = 1 - P_e = 1 - P\{\hat{m} \neq m\} = P\{\hat{m} = m\} \quad .$$

- maximum *à* posteriori (MAP) detector – given some observed channel output $\boldsymbol{y} = \boldsymbol{v}$ is $p_{\boldsymbol{x}/\boldsymbol{y}}$.

$$P_c(\hat{m} = m_i,\ \boldsymbol{y} = \boldsymbol{v}) = p_{m/\boldsymbol{y}}(m_i, \boldsymbol{v}) \cdot p_{\boldsymbol{y}}(\boldsymbol{v}) = \underbrace{p_{\boldsymbol{x}/\boldsymbol{y}}(\boldsymbol{x}_i, \boldsymbol{v})}_{MAP} \cdot \cancel{p_{\boldsymbol{y}}(\boldsymbol{v})} = \underbrace{p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{v})}_{ML} \cdot \cancel{p_{\boldsymbol{x}_i}}$$

Not function of $\boldsymbol{x}_i$

- If $p_{\boldsymbol{x}_i} = \frac{1}{M}$, then Maximum Likelihood (ML) is $p_{\boldsymbol{y}/\boldsymbol{x}}$.

The MAP and ML Rules

Rule 1.1.1 **[MAP Detection Rule]**

$$\hat{m} \Rightarrow m_i \;\; if \;\; p_{\boldsymbol{y}/m}(\boldsymbol{v}, i) \cdot p_m(i) \geq p_{\boldsymbol{y}/m}(\boldsymbol{v}, j) \cdot p_m(j) \;\; \forall \;\; j \neq i \tag{1.8}$$

If equality holds in (1.8), then the decision can be assigned to either message m_i or m_j without changing the minimized error probability.

Rule 1.1.2 **[ML Detection Rule]**

$$\hat{m} \Rightarrow m_i \;\; if \;\; p_{\boldsymbol{y}/m}(\boldsymbol{v}, i) \geq p_{\boldsymbol{y}/m}(\boldsymbol{v}, j) \;\; \forall \;\; j \neq i \quad . \tag{1.10}$$

If equality holds in (1.10), then the decision can be assigned to either message m_i or m_j without changing the error probability.

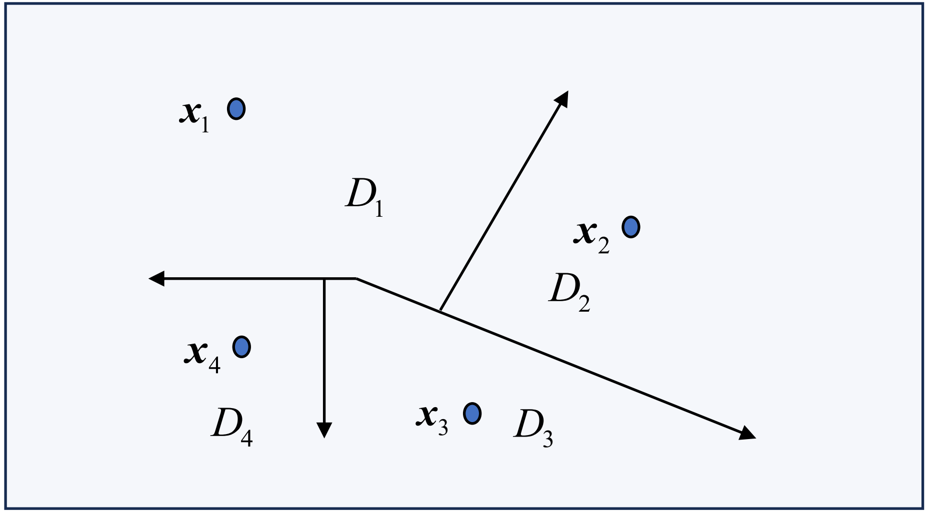
- Assume that all messages are equally likely, use ML (almost always).
- If the inputs have unequal probabilities, they can be compressed to lower rate with equally likely probabilities.

Examples in Lecture 2 and beyond.

Decision Regions

Definition 1.1.4 **[Decision Region]** *The* **decision region** *using a MAP detector for each message* m_i, $i = 0, \ldots, M-1$ *is defined as*

$$\mathcal{D}_i \triangleq \{\boldsymbol{v} \mid p_{\boldsymbol{y}/m}(\boldsymbol{v}, i) \cdot p_m(i) \geq p_{\boldsymbol{y}/m}(\boldsymbol{v}, j) \cdot p_m(j) \;\; \forall \;\; j \neq i\} \quad . \tag{1.12}$$



- Detector "precomputes" the associated decision for every specific value of $\boldsymbol{y} = \boldsymbol{v}$.

Decision Regions and Error Probability

- For any decision region

$$\mathbb{E}[P_c] = \sum_{i=0}^{M-1} \sum_{\boldsymbol{v} \in \boldsymbol{\mathcal{D}_i}} P_c(\hat{m} = m_i, \boldsymbol{y} = \boldsymbol{v} \in \boldsymbol{\mathcal{D}_i})$$

- For MAP

$$P_{c,max} = \mathbb{E}[P_c] = \sum_{i=0}^{M-1} \left\{ \sum_{\boldsymbol{v} \in \boldsymbol{\mathcal{D}_i}} P_{\boldsymbol{y}/m_i}(\boldsymbol{v}\,, m_i) \right\} \cdot p_{m_i}$$

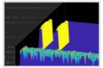
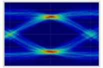
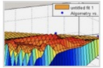
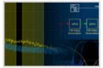
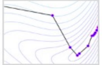
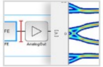
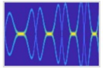
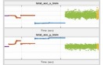
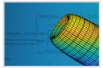
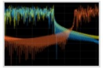
$$P_{e,min} = 1 - P_{c,max}$$

These (double) sums often simplify greatly, as we'll see.

End Lecture 1

- Discrete Data Transmission & Applications
- Encoding and Detection (decoding)
- Modulation and Demodulation
- Vector Channels
- Probability (see Appendix A) – happy to help, off hours, help

Cioffi uses these toolboxes

	Name
	**5G Toolbox** version 2.6
	**Communications Toolbox** version 8.0
	**Curve Fitting Toolbox** version 3.9
	**DSP System Toolbox** version 9.16
	**Optimization Toolbox** version 9.5
	**Parallel Computing Toolbox** version 7.8
	**SerDes Toolbox** version 3.0
	**Signal Processing Toolbox** version 9.2
	**Statistics and Machine Learning Toolbox** version 12.5
	**Symbolic Math Toolbox** version 9.3
	**WLAN Toolbox** version 3.6