Introduction to Deep Space Communications

Dr. Les Deutsch

Deputy Director, Interplanetary Network Directorate

Jet Propulsion Laboratory
California Institute of Technology

Course Concept

This is not meant to substitute for a complete academic course in communications

- The scope of these slides is usually covered in a one-year graduate-level course
- Prerequisites typically include
  - Abstract algebra
  - Differential equations
  - Basic electrical engineering

These slides should give you an idea of the overall concepts

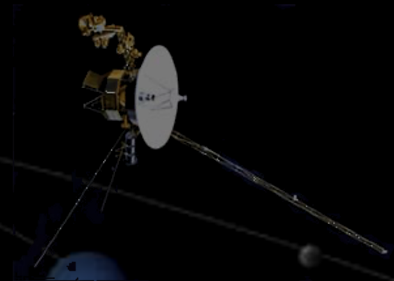
Hopefully they will inspire you to learn more in specific areas

Course Outline

1. Deep Space Communications Systems
2. Modulation
3. Coding
4. Design Control Table Example
5. Antenna Arraying
6. Performance History
7. Multiple Access
8. Bibliography

1. Deep Space Communications Systems

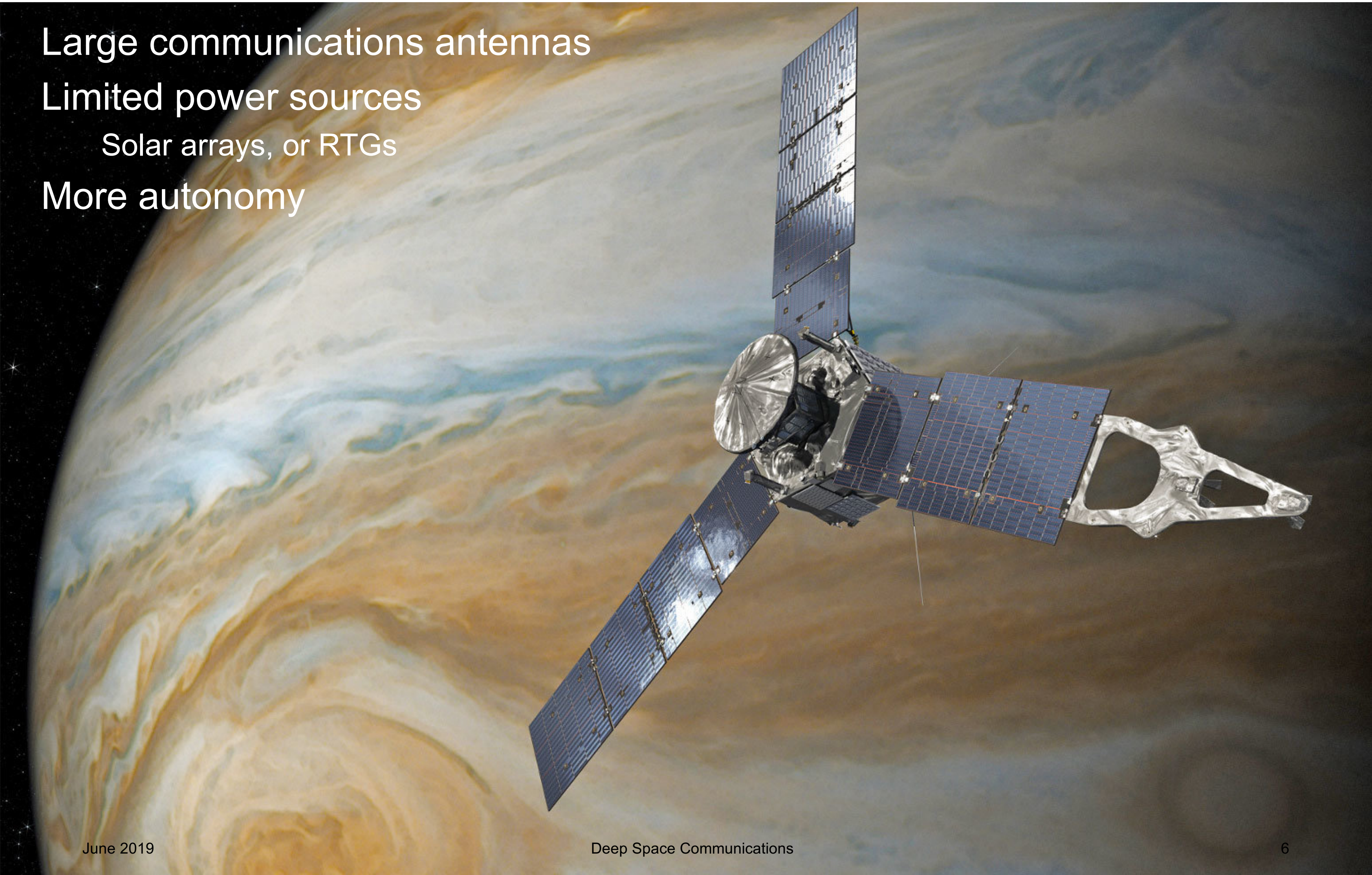
Deep Space is Unique



- Spacecraft mass and power are precious
- Huge communication distances:
  - Large spacecraft and Earth antennas
  - High power transmitters
  - Latency is large – leading to more autonomy
- Spacecraft power is precious
  - Small spacecraft size
  - Outer planets are far from the Sun
- Navigation is highly dependent communications system
  - No GPS in deep space
  - Target bodies must also be “navigated”
- Each spacecraft tends to be unique
  - Missions to specific targets tend to come only “once in a long while”
- Most systems are custom
  - Not much commercial market – though we leverage as much as possible

Deep Space Missions
Quite different from Earth orbiters

- Large communications antennas
- Limited power sources
  - Solar arrays, or RTGs
- More autonomy



The Deep Space Network
NASA's Connection to the Moon, Planets, & Beyond

- Large antennas at three global sites: California, Madrid, Canberra
- Very high power transmitters
- Very precise pointing and tracking

What it's all about

We assume a digital communications system, with information represented by a string of 0s and 1s

The key metric is "Signal to Noise Ratio" or "SNR"

I will use this definition of SNR: P_R/N_0

P_R is the power in each bit that gets to our receiving system

N_0 is the power spectral density of all the noise that gets to our receiving system

Don't worry about what that actually means yet, we'll get to it later!

The bottom line is that when SNR is high, we have good communications and when it is low we do not.

The rest is just a bunch of really important details.

Uplink and Downlink

We call signals sent from a spacecraft to Earth *"downlink"* and those from Earth to a spacecraft *"uplink"*

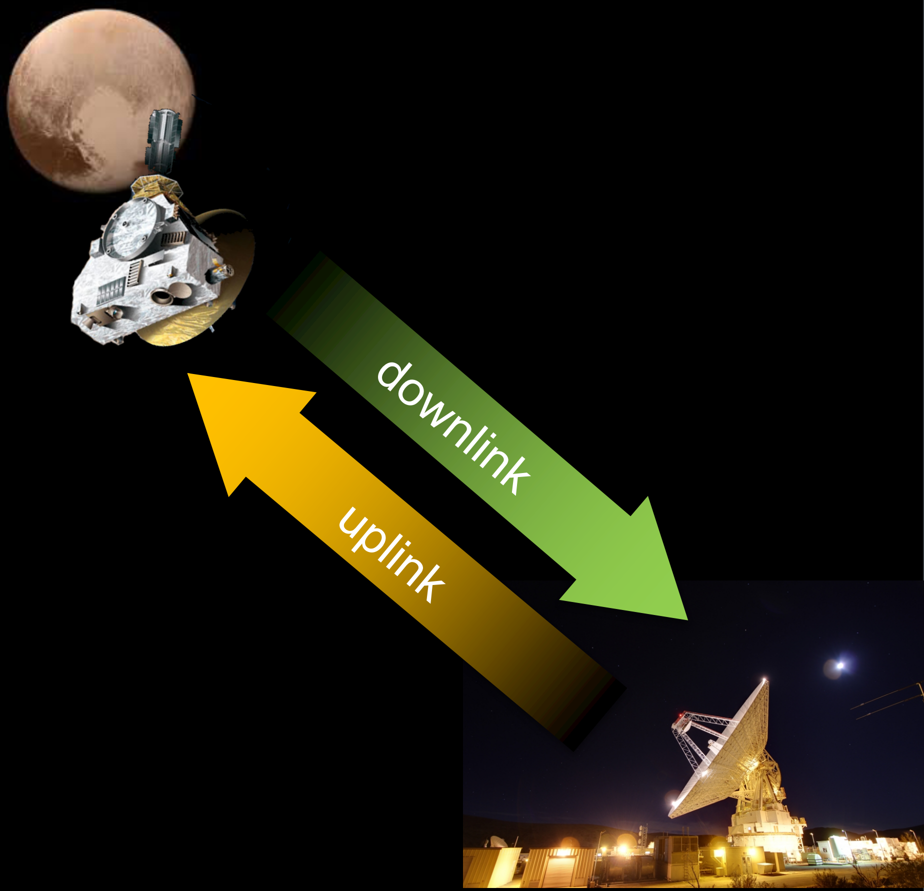
- There are other cases, such as when an intermediate space relay is used – but we will ignore those here

Uplink and downlink are governed by identical mathematics

For deep space, downlink is much harder

- Downlink contains all the important (and voluminous) science data
- Uplink is mostly short sets of instructions to the spacecraft
- This will change when humans go to deep space again

Hence, we will focus on downlink and assume whenever we can achieve good downlink, then uplink is an easy addition



Decibels (dB)

In deep space, numbers can be huge

In order to avoid writing large numbers and to help show the huge dynamic ranges, we tend to express numbers on a logarithm scale

We typically use decibels, or dBs, defined by

$$n_{dB} = 10 \log_{10}(n)$$

A *Decibel* is 10 *Bells*, a unit invented by the Bell Telephone Company in honor of Alexander Graham Bell. Since *Bell* is a proper name, the B is always capitalized in dB.

Examples:

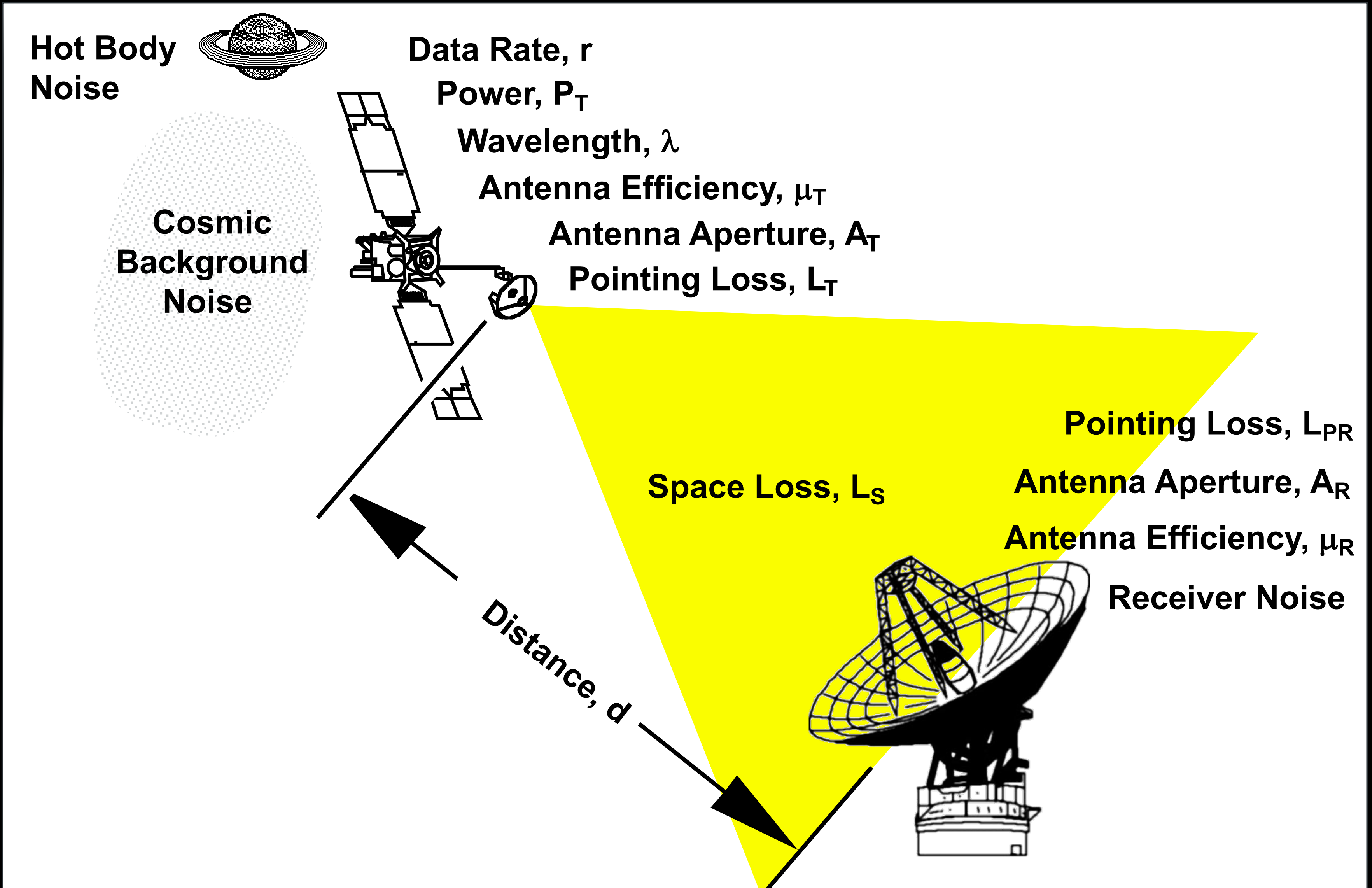
10 dB = 10

20 dB = 100

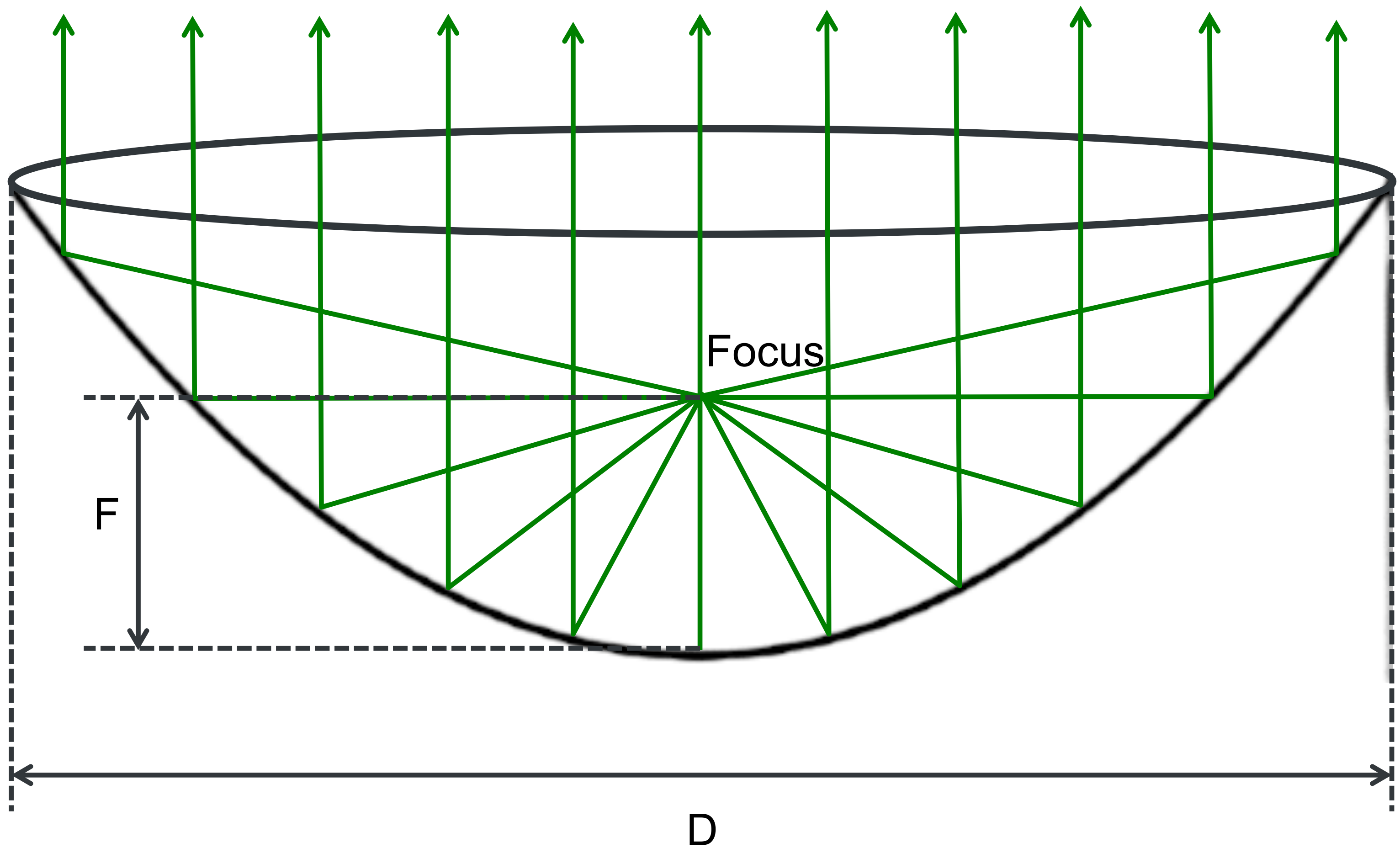
100 dB = 10,000,000,000

3 dB ≅ 2

The parameters of deep space comm



Parabolic Antennas



Antenna Gain

Antenna gain is a measure of how much energy is amplified through the antenna

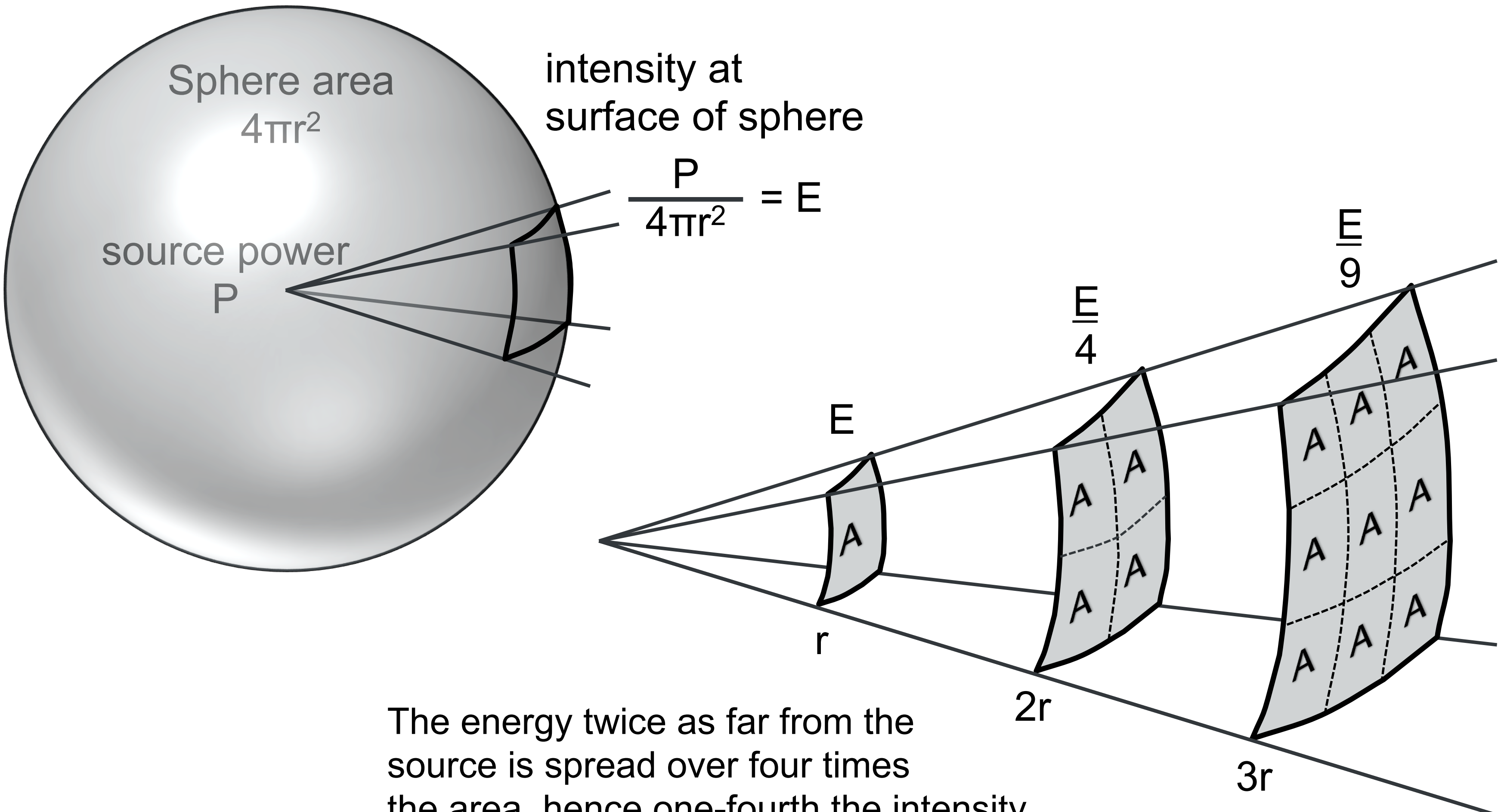
$$G_T = \frac{4\pi\mu_T(\lambda)A_T}{\lambda^2} \qquad G_R = \frac{4\pi\mu_R(\lambda)A_R}{\lambda^2}$$

It is proportional to the antenna's effective area, A

It is also proportional to the antenna's efficiency, μ , a number between 0 and 1 that is a function of the accuracy of the antennas shape and its surface roughness. This number is a function of the transmitted wavelength.

It is inveresly proportional to the square of the signal's wavelength, λ

Inverse Square Loss



The energy twice as far from the source is spread over four times the area, hence one-fourth the intensity

Space Loss

This is the real problem in deep space communications

$$L_S = \left(\frac{\lambda}{4\pi d}\right)^2$$

Space loss is inversely proportional to the distance between the transmitter and the receiver

It also is propositional to the square of the wavelength

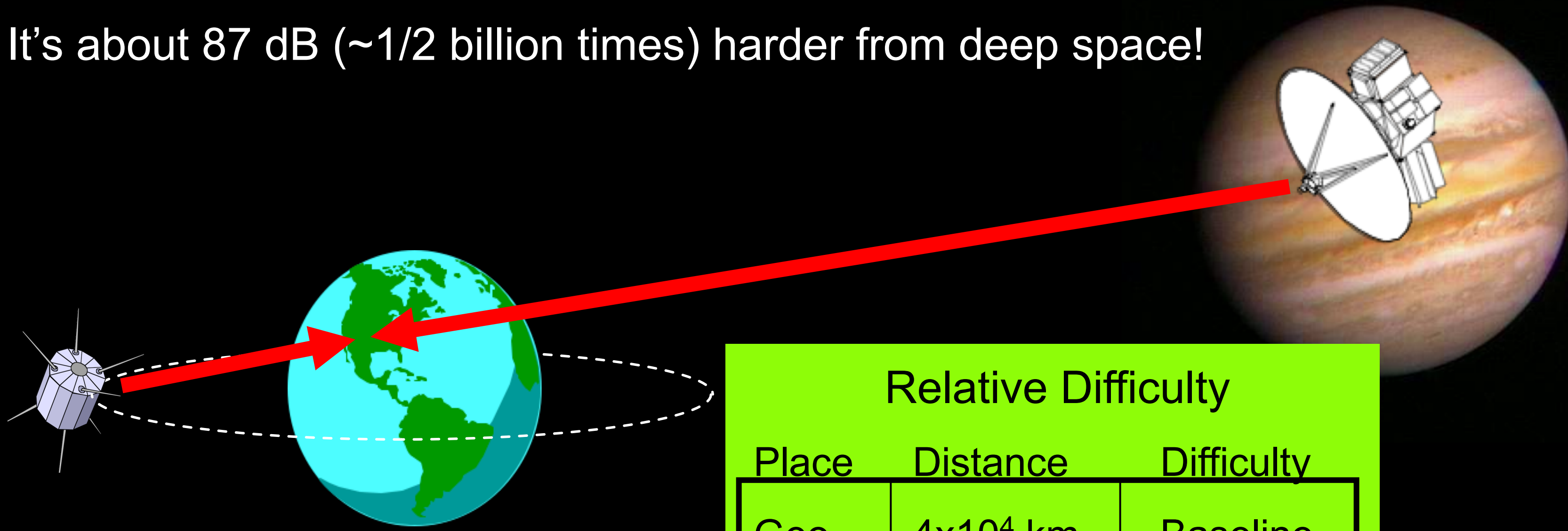
This is why, in general, higher frequencies are better for communications

Why is Deep Space Comm Difficult?

Communications performance decreases as the square of the distance.

Jupiter is nearly 1 *billion* km away, while a GEO Earth communications satellite is only about 40 *thousand* km away

– It's about 87 dB (~1/2 billion times) harder from deep space!



Relative Difficulty

Place	Distance	Difficulty
Geo	$4x10^4$ km	Baseline
Moon	$4x10^5$ km	100
Mars	$3x10^8$ km	$5.6x10^7$
Jupiter	$8x10^8$ km	$4.0x10^8$
Pluto	$5x10^9$ km	$1.6x10^{10}$

Received power per bit

So, the power in each bit that we find at the receiver is

$$P_R = P_T G_T L_S G_R / r$$

But there are other losses!

Antenna pointing losses

We can never point our antennas perfectly

This leads to two additional important losses, one for the spacecraft antenna and the other for the ground atenna

The final power formula is therefore

$$P_R = P_T G_T L_{PT} L_S L_{PR} G_R / r$$

Where L_{PT} and L_{PR} are the respective antenna pointing losses. These are both numbers between 0 and 1 – hopefully closer to 1!

Noise

- Noise is defined as any unwanted power that makes it to our receiver at or near the same part of the spectrum where our signal lies
  - This is just like the definition of “weed” with respect to a garden
- Since noise is not a defined waveform, but rather the sum of lots and lots of unwanted things, we presented it by a random process
- We make the assumption that noise sources are independent Gaussian processes
  - This is good assumption for deep space because there are so many noise sources
  - It also makes our calculations much easier
  - It means we can easily add noise sources
  - Experience has shown this assumption to be excellent
- Each noise source can then be described as a Gaussian process with noise temperature, T

Noise Sources

Cosmic background radiation – its always there and deep space communications helped find it in the first place

Hot bodies – e.g. planets or Moons that also lie in the beam of our receive antenna

Radio frequency interference (RFI) – Human-made signals which happen to be in our spectral band and receive antenna beam

Atmosphere – The Earth's atmosphere adds noise that is dependant on the signal's wavelength

Receiver – The most annoying of all noise sources, noise we generate ourselves by not have perfect systems

There may be other noise sources, but these are the most important for our case.

The total noise temperature is simply the sum of the individual temperatures

$$\begin{aligned} T = {} & T_{cosmic\ background} \\ & + T_{hot\ bodies} \\ & + T_{RFI} \\ & + T_{atmosphere} \\ & + T_{receiver} \end{aligned}$$

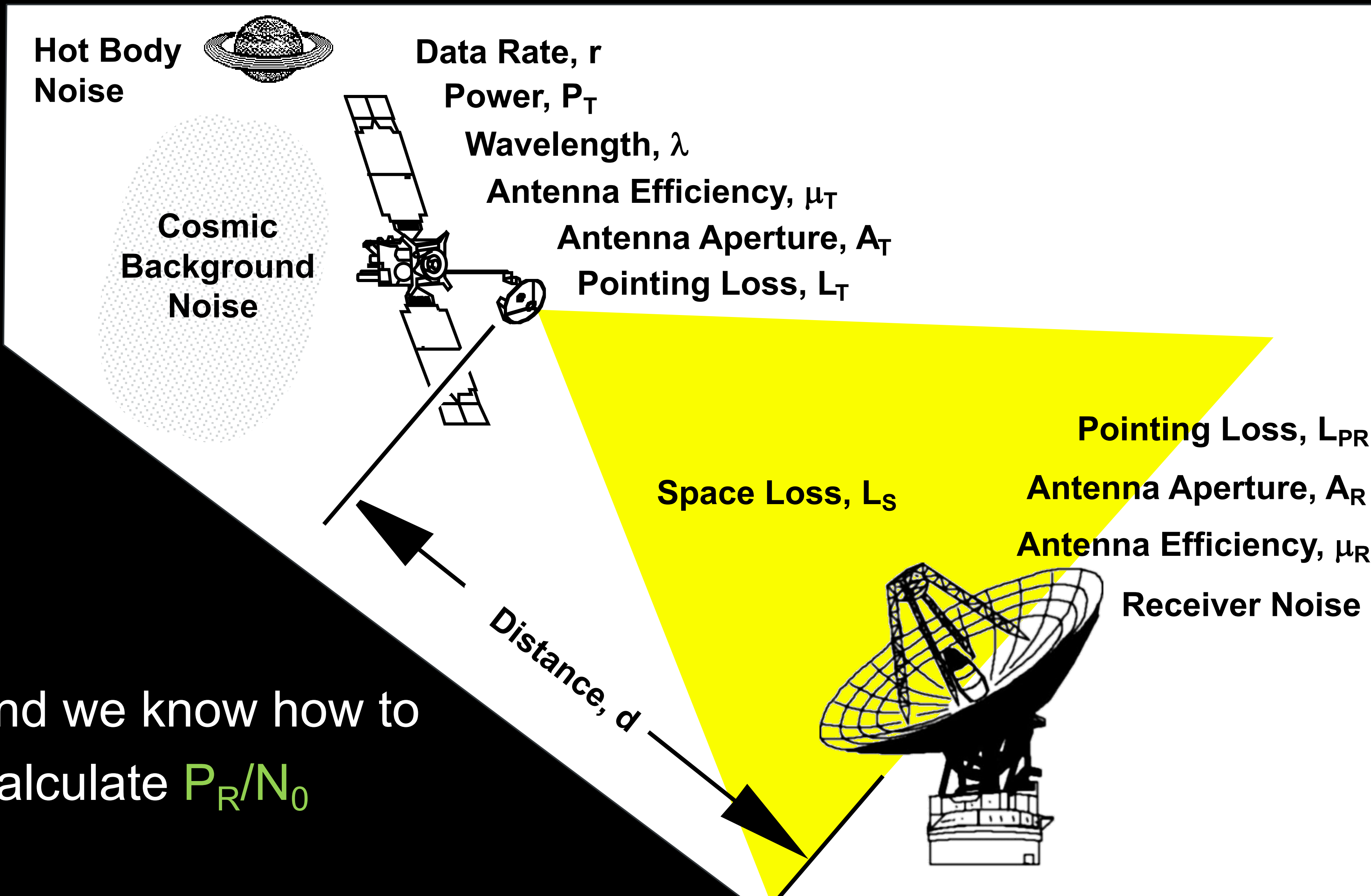
Total noise power

Using standard physics, we now can calculate the noise power spectral density as

$$N_0 = kT$$

Were k is Boltzmann's constant

So, here are all the parameters again



And we know how to Calculate P_R/N_0

2. Modulation

Introduction to modulation

We have looked at the various system parameters that determine communications performance

Now we have to determine how to actually use them to form a communications system

In this section we consider “modulation” which is simply a way of representing the but stream as an electromagnetic wave that can be physically transmitted and received

We start by studying a specific modulation algorithm: residual carrier binary phase shift keying (BPSK)

There are many other algorithms – we’ll get to them later

Residual carrier BPSK

We define our transmitted signal by

$$s(t) = A\sin(\omega t + \theta\, d(t))$$

Where A is its amplitude, ω is its frequency, and $d(t)$ is the binary data stream
In fact, relating these to previous parameters we see

$$\omega = c/\lambda$$

where c is the speed of light, and A is proportional to P_R
We use the convention

$$d(t) \in \{-1, +1\}$$

for the binary data rather than the more common “0s and 1s”. This will make the Math a lot simpler.

θ is defined as the modulation index

Some calculations

We use trigonometry identities to expand the signal:

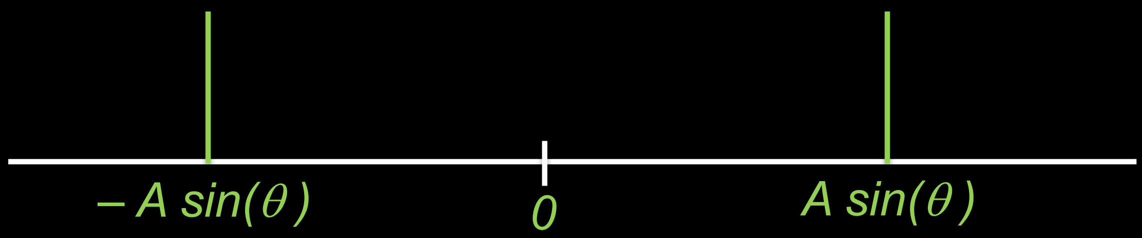
$$
\begin{aligned}
s(t) &= A \sin(\omega t + \theta d(t)) \\
&= A \sin(\omega t) \cos(\theta d(t)) + A \cos(\omega t) \sin(\theta d(t)) \\
&= A \cos(\theta) \sin(\omega t) + A d(t) \sin(\theta) \cos(\omega t)
\end{aligned}
$$

since $d(t)$ is always ± 1

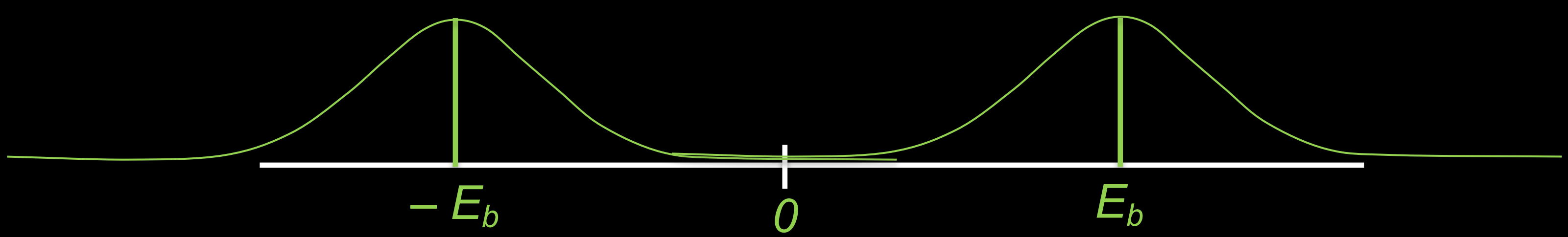
The first term is called the *residual carrier* while the second term is called the *data component.*

BPSK Performance

Consider the data component only for the moment and plot it in phase space



Now we add Gaussian noise, and define $E_b = A \sin(\theta)$

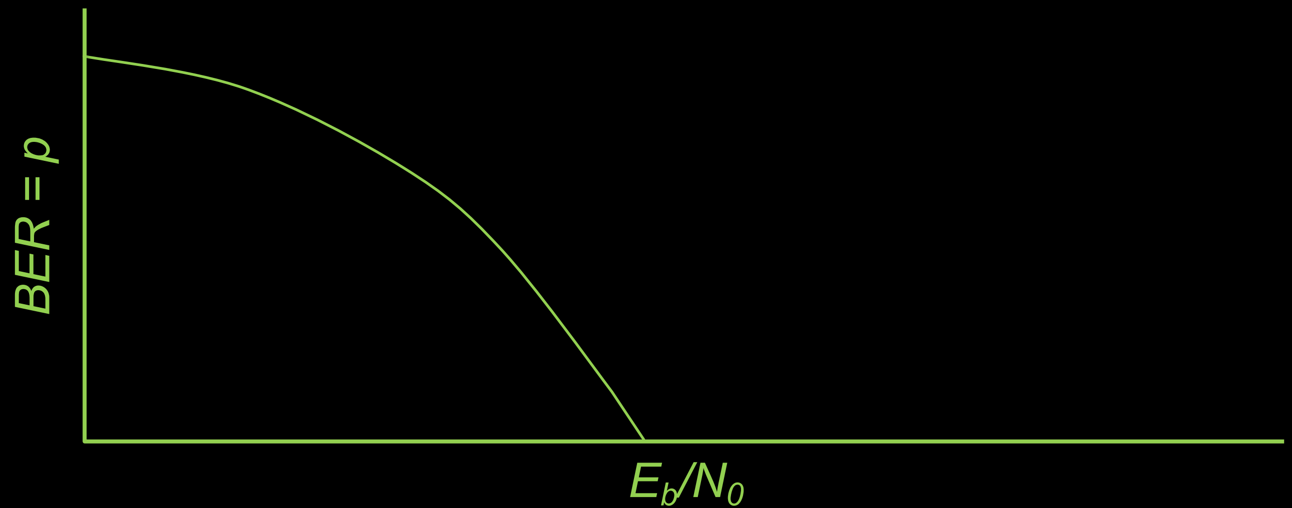


When there is so much noise that we mistake a -1 for a 1 (and visa versa) we get a bit error

BPSK Performance

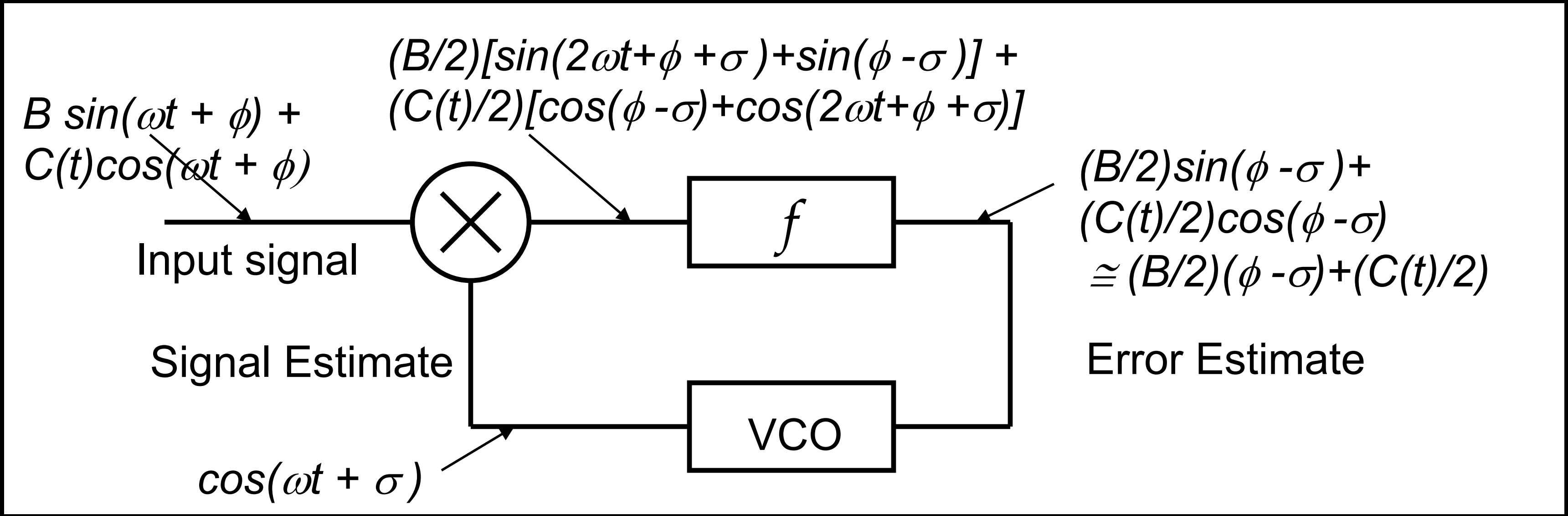
We can compute the bit error probability as a function of the signal to noise ratio, E_b/N_0. It is just probability that the noise is large enough during that bit time to push the value closer to the wrong side:

$$p = \frac{1}{\sqrt{\pi}} \int_{\sqrt{E_b/N_0}}^{\infty} e^{-t^2}\,dt = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{E_b/N_0}\right)$$



Phase lock loops

PLLs are a near-optimal method for tracking these dynamic signals, especially in presence of Doppler



The loop makes a guess, σ, for the phase. If it is close enough, the loop will converge and σ will tend to ϕ.

PLLs are not perfect. Even in the absence of noise, they will have inherent oscillations called "loop jitter" due to an uncertainty principle!

Doppler

The same phenomenon that makes a fire engine's siren appear to change frequency as it passes us, causes the apparent frequency from our spacecraft change as it accelerates in its path relative to our ground station

- Even if the spacecraft is "standing still" in space, we get this effect because of the Earth's rotation

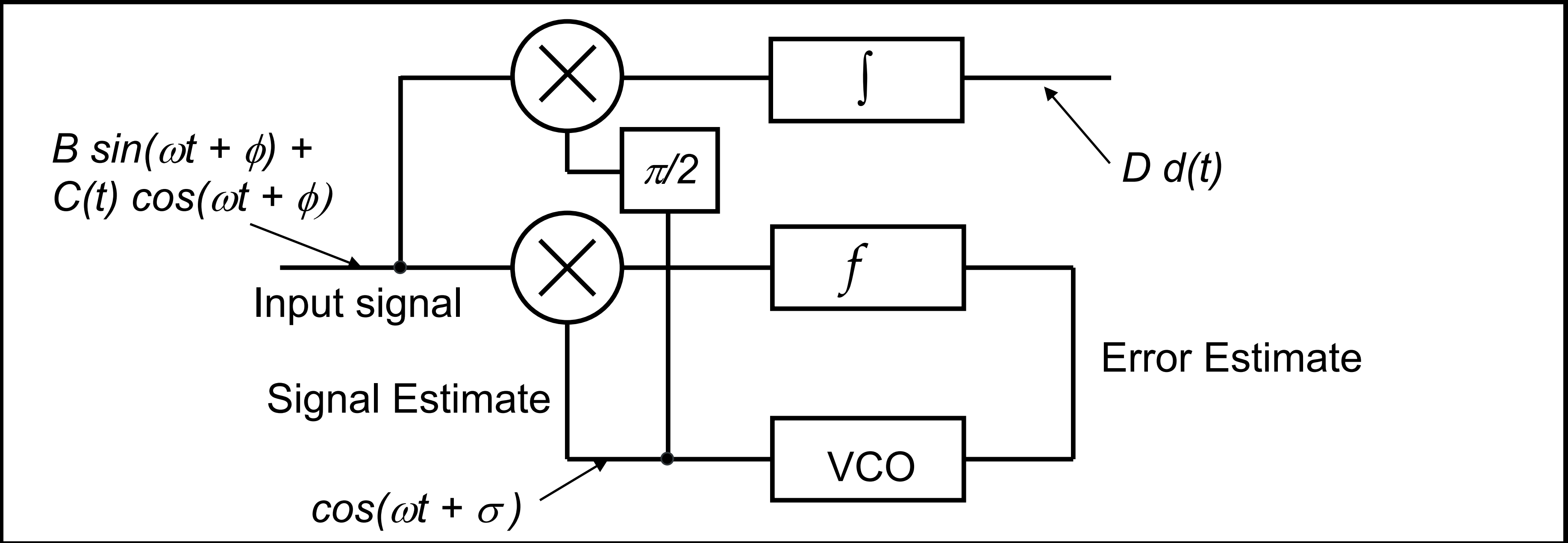
This effect is called *"Doppler"* and is named after Christopher Doppler who described it in the mid 19th century (hence, it is always capitalized)

We can model Doppler as yet another noise that perturbs the expected sinusoidal signal

Our Phase Lock Loop will track out the Doppler as long as the loop bandwidth is wide enough

Now, what about the data

We add a bit to the loop so we can extract the data stream



By shifting the loop estimate phase by $\pi/2$ we focus on the data part of the signal

The integrator integrates over a bit time, which we know a-priori. This is usually itself implemented as a second phase lock loop

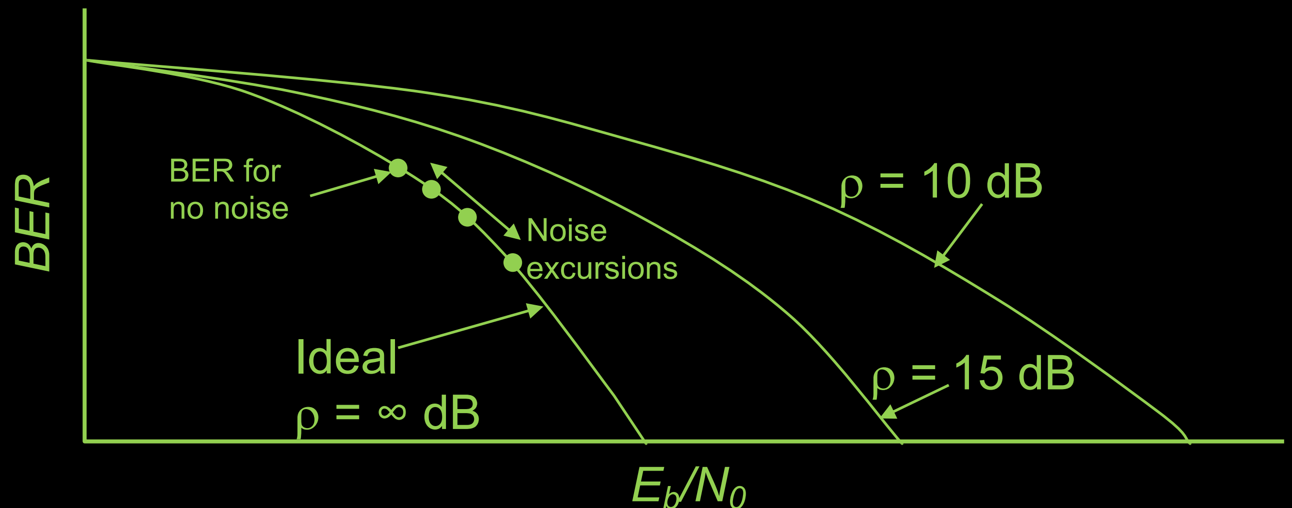
Radio Loss

The *erf* equation gives the correct performance of a PLL in the absence of noise and signal dynamics – e.g. ideal conditions

In a noisy system with PLLs, the effective instantaneous signal-to-noise ratio is degraded by poor PLL error estimates

These result in an effective “dancing around on the ideal curve” and leads to the degradation known as radio loss

Radio loss is a function of the relative amount of signal power in the loop bandwidth bandwidth of the PLL, called the Loop SNR and denoted by ρ.



Suppressed carrier modulation

The special case where $\theta = \pi/2$ yields

$$
\begin{aligned}
s(t) &= A\sin(\omega t + (\pi/2)\,d(t)) \\
&= A\sin(\omega t)\cos((\pi/2)\,d(t)) + A\cos(\omega t)\sin((\pi/2)\,d(t)) \\
&= A\cos(\pi/2)\sin(\omega t) + A\,d(t)\sin(\pi/2)\cos(\omega t) \\
&= A\,d(t)\cos(\omega t)
\end{aligned}
$$

Since there is no “carrier” portion in the modulation, this is called *suppressed carrier modulation* or sometimes *direct modulation*

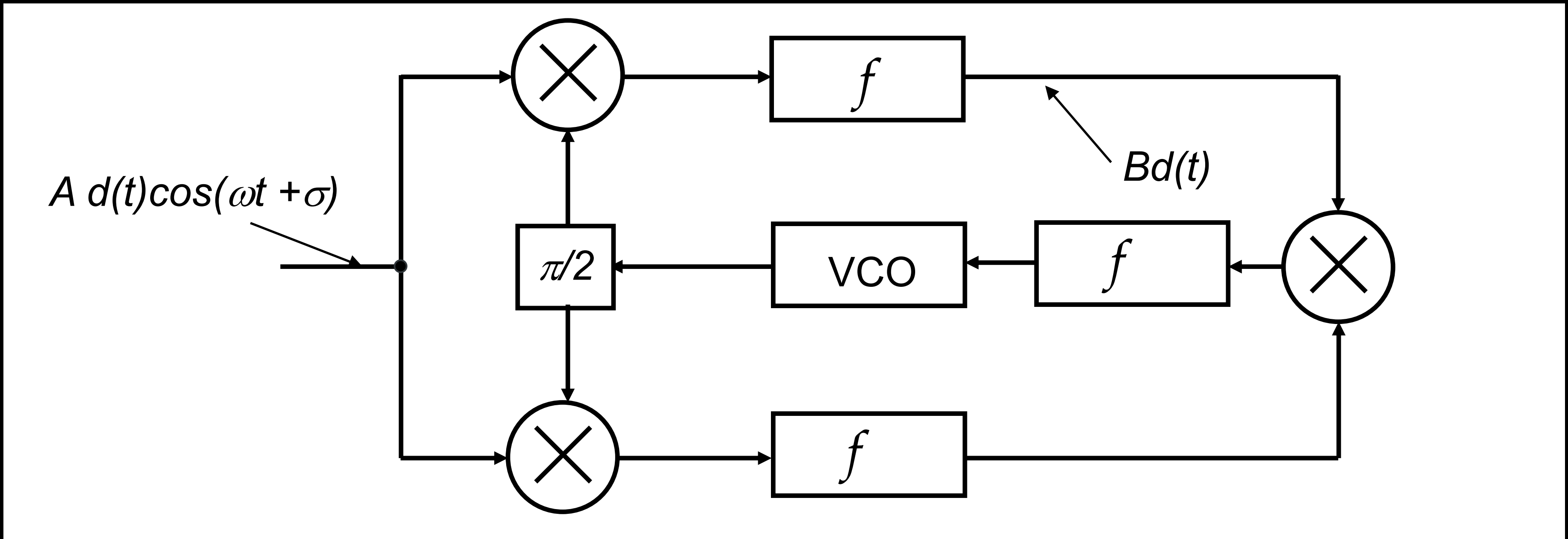
Since all the energy in the signal is in the data portion, this has the potential to perform better than residual carrier modulation

However, the phase lock loop we have already seen fails to work in this case

We need a different kind of loop

Squaring loop

A *squaring loop* is used for suppressed carrier modulation



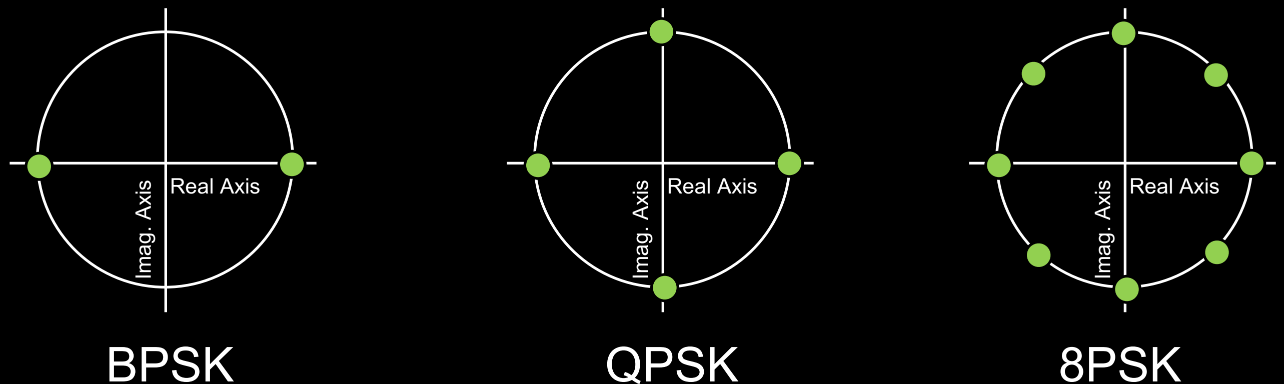
However, a phase shift of only $\pi/2$ will cause the bit stream to be inverted – which would only happen with π for suppressed carrier

Hence, residual carrier can only be used when the loop SNR is high enough so that these excursions are rare

Other Modulation Types

Some missions use *Quadrature Phase Shift Keying* (QPSK) and we have recommended using 8-Phase Shift Keying (8PSK) and higher dimensional schemes as it becomes necessary to fit more bits into the allocated spectral bandwidth

One can visualize all these schemes by saying that the modulator can choose from one of several signals at each bit time as follows



Since the "signals" are closer together, more power is required in the higher-dimensional schemes to maintain the same bit error performance

There are LOTS of variations on these schemes, e.g. *offset keying*

Subcarriers

Subcarrier modulation is often applied to keep the data modulation away from the carrier in the frequency domain

This requires an additional PLL for subcarrier demodulation!

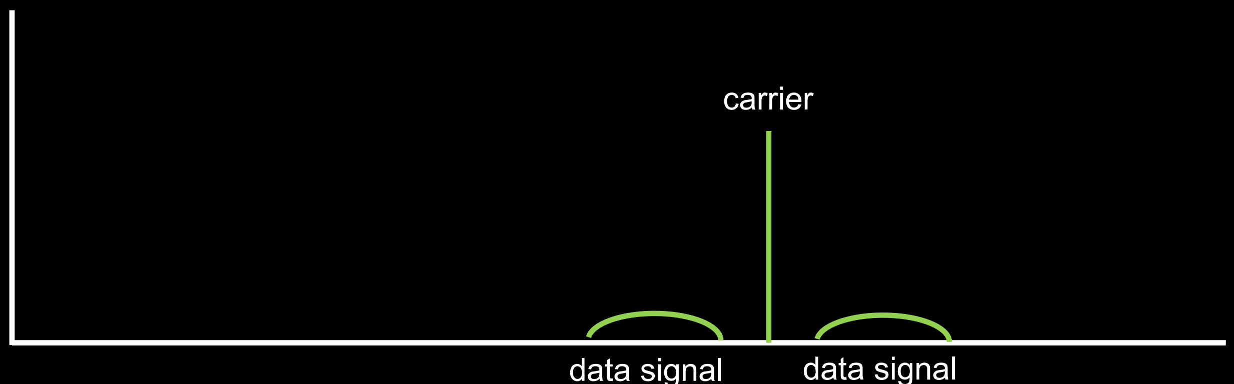
$$s(t) = A\sin(\omega t + \theta S(\omega_s t)d(t))$$

where

S is a square wave function
ω_s is the subcarrier frequency

Subcarriers

In frequency space, a signal modulated with a subcarrier places it a away from carrier so it is more easily distinguished from that carrier



It is also broadened by the subcarrier waveform, making it perhaps more protected from interference

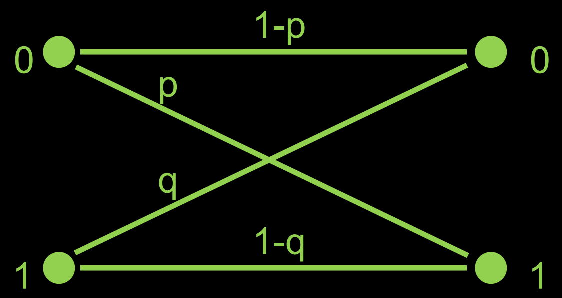
3. Coding

Coding: The deep space channel

A channel is everything that exists between the transmitter of a message and its receiver

If the elements of the messages are confined to a finite set of values, then the channel is called discrete

In most of digital communications (where messages consist of 0's and 1's) we can represent channels by the following diagram in which p and q are the probabilities of bit error



If p = q, then the channel is called symmetric

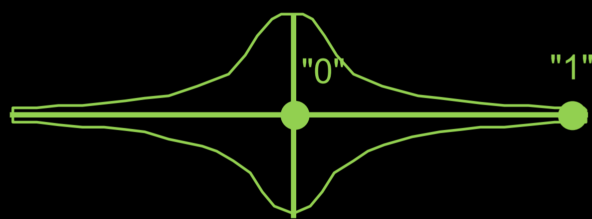
If p and q do not vary with time or the input message, then the channel is called memoriless

Uncoded performance – a review!

In most real world communications (including deep space) bits are sent as one of two analog waveforms that may be represented (in some convenient space) as two signal levels:

energy, magnitude is E_b

The error mechanism in the channel is often a form of additive noise. Most often, this noise has a Gaussian probability density.



Gaussian noise spectral density is N_0

In this case, we can compute the raw bit error probability of the channel as a function of the signal to noise ratio, E_b/N_0. It is just probability that the noise is large enough during that bit time to push the value closer to the wrong one:

$$p = \frac{1}{\sqrt{\pi}} \int_{\sqrt{E_b/N_0}}^{\infty} e^{-t^2} dt = \frac{1}{2} \mathrm{erfc}\left(\sqrt{E_b/N_0}\right)$$

Eror Korecting Cods

An *error correcting code* uses redundancy to control errors on a noisy channel

Trivial example: Repetition codes – just say your message multiple times

- This works well in advertising!
- Decoding is simple – take majority vote
- Repetition codes are very wasteful in power

Luckily, there are codes that are not wasteful

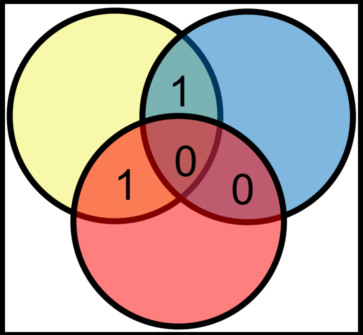
Claude Shannon proved “for any given degree of noise contamination of a communication channel, it is possible to communicate discrete data (digital information) nearly error-free up to a computable maximum rate through the channel”

Unfortunately, the proof give us no insight in finding what codes will actually work!

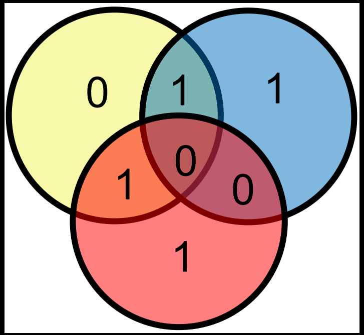
- The proof uses “random coding” to show some codes will work

A simple example

An example of coding: the (7, 4) Hamming code

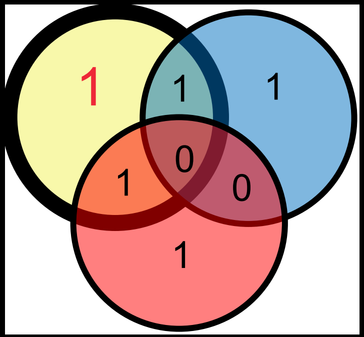


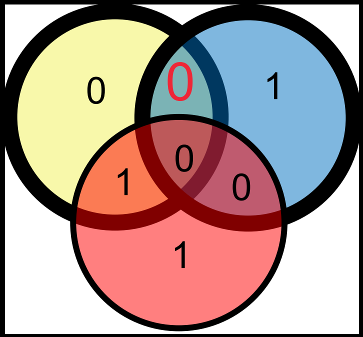
Place 4 information bits in the intersections of the Venn diagram



Fill in the diagram so that the circles have an even number of 1's

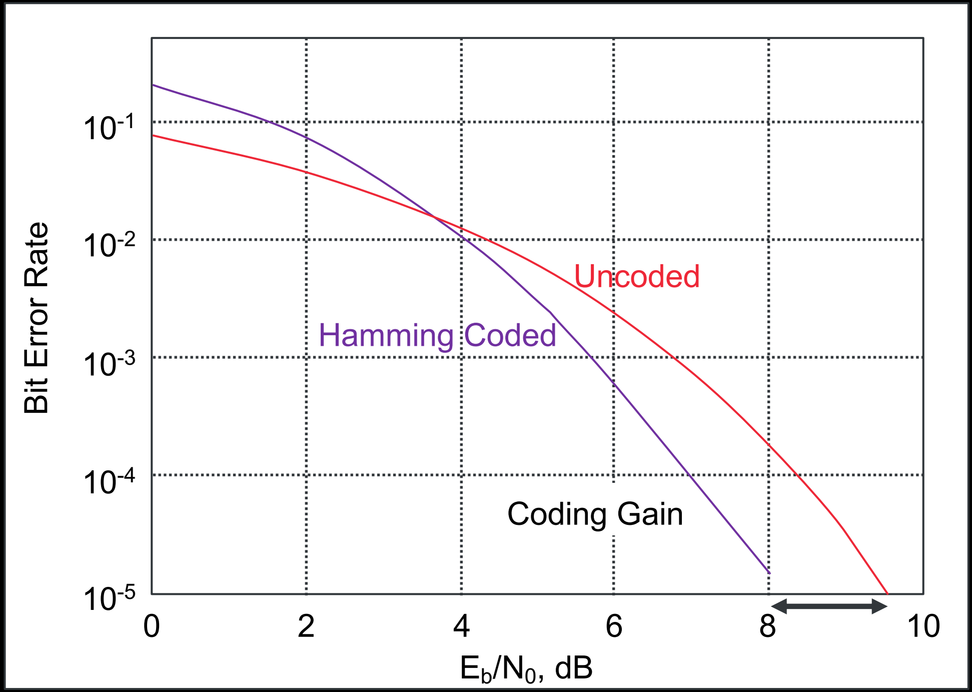
If a single error occurs, it can be corrected by locating the circles with an odd number of 1's and changing the bit in their intersection





(7, 4) Hamming code performance

Here is the bit-error rate of the (7, 4) Hamming Code as a function of SNR



We normalize the chart because the uncoded case uses four *bits* to send each message while the code uses seven *symbols* – this keeps things fair!

Notice that at low SNR, uncoded is better

This is typical – codes require a certain amount of SNR to work well

There is “coding gain” at higher enough SNR

Block Codes

Block Codes take fixed length data streams and produce fixed length codewords

The (7, 4) Hamming code is an example: sets of 4 bits are mapped to 7 bit codewords

In general, a (n, k) block code takes k bits of information and generates n bit codewords

The code rate is defined to be the ratio of the number of bits in the original message to the number in the codeword, i.e. $R = k/n$

A code is called *linear* if, whenever c_1 and c_2 are codewords, then so is $c_1 + c_2$.

All linear block codes can be expressed in algebraic notation using a generator matrix. For the Hamming code,

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad m = (x_1\ x_2\ x_3)$$

mH is a codeword in the (7, 4) Hamming code

If c is any codeword in the (7, 4) Hamming code, then $H^T c = 0$

If c is sent, but it is corrupted by noise, n, with at most one bit error, then

$$H(c+n)^T = Hc^T + Hn^T = Hn^T$$

which points to the position of the bit error

Synchronization

In order to decode a block code, we have to parse the incoming data stream into codewords

This means finding the position of the first bit of each codeword – called *code synchronization*

There are many techniques for this – the most common is inserting a known sequence of bits periodically in the data stream – called a *synchronization marker*

We make the sync marker as short as we can to minimize the loss due space in the data stream for information bits

- We typically use *pseudo-noise (pn)* sequences of bits for markers

The detector for the marker can integrate over several marker locations

Reed-Solomon Codes

We can generalize block coding to work over arbitrary finite fields

In fact, the most common block codes used in deep space are defined over Galois fields of size 2^n

For such fields, we simply take n consecutive data bits and identify them with an element in $GF(2^n)$

Reed Solomon (RS) codes are examples of these

The RS codes used in deep space are typically defined over GF(256) and represent sequences of eight bits

The CCSDS standard RS code for deeps pace is a (255, 223) block code over GF(256)

Everything we have said before about block codes still applies, but the elements in the matrices are now in GF(256) instead of being simply 0s and 1s.

How linear codes work: n-dimensional balls

Imagine a ball pit – but in n dimensions

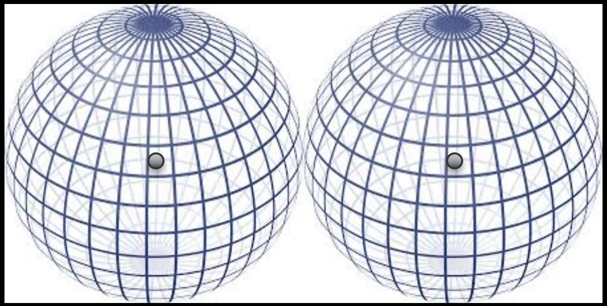
Map each k-bit chunk of data into the center one of the balls – which we call a codeword

The distance between points in this space is the number of bits in which they differ:

e.g. 1011000111 is distance 3 from 1001011111

Balls and error rates

Consider two adjacent codewords in this pit



If the balls have radius e, then we would have to change at least e+1 bits of our codeword to “move” it into the adjacent sphere

This means that this code can detect and correct up to e errors

An (n, k, d) linear block code is defined by

- n = code length
- k = the number of data bits represented each codeword
  - so, the number of codewords is 2^k
- d = minimum distance between codewords

For the codes we have seen thus far

The Hamming code example is a (7, 4, 3) code

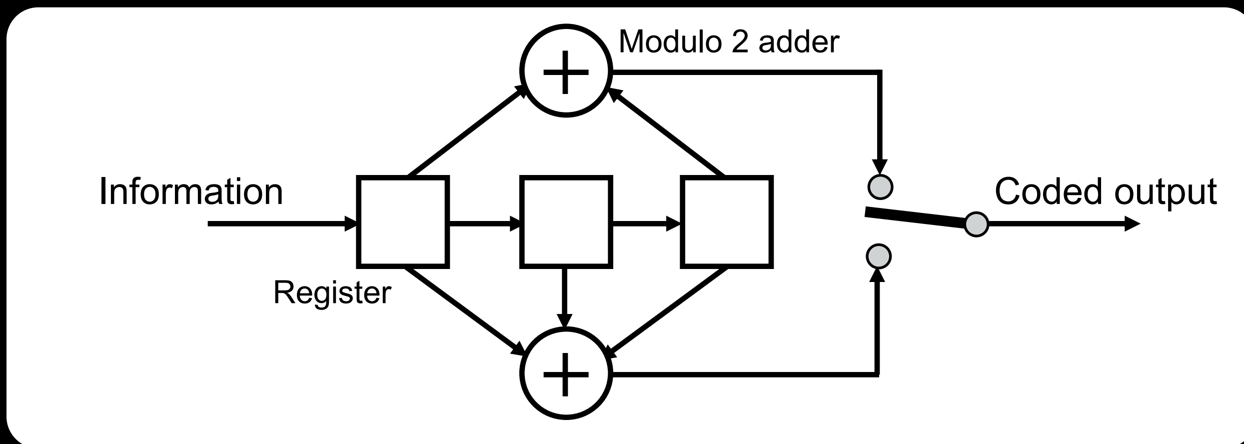
- It can correct single errors
- Its “balls” fit together perfectly with no space in between – called a *perfect code*

The CCSDS standard Reed Solomon code is a (255, 223, 33) code

- It can correct up to 16 errors
- Because its balls do not pack perfectly, it also can detect “erasures”, places in the data stream that are likely places for additional errors

Convolutional Codes

Convolutional Codes create redundancy in an iterative fashion rather than in predefined codeword sizes as in block codes



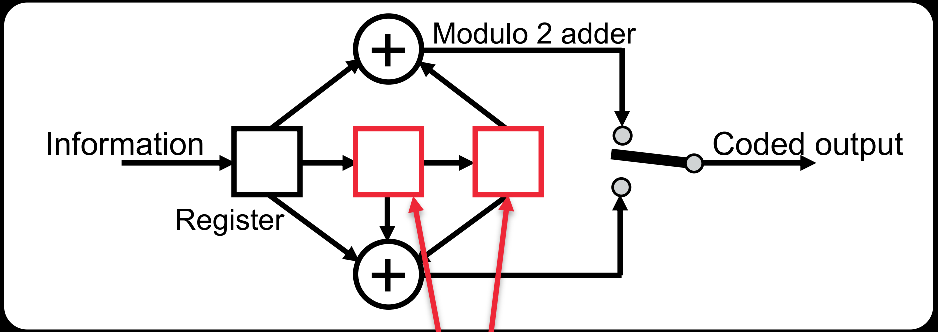
How it works:

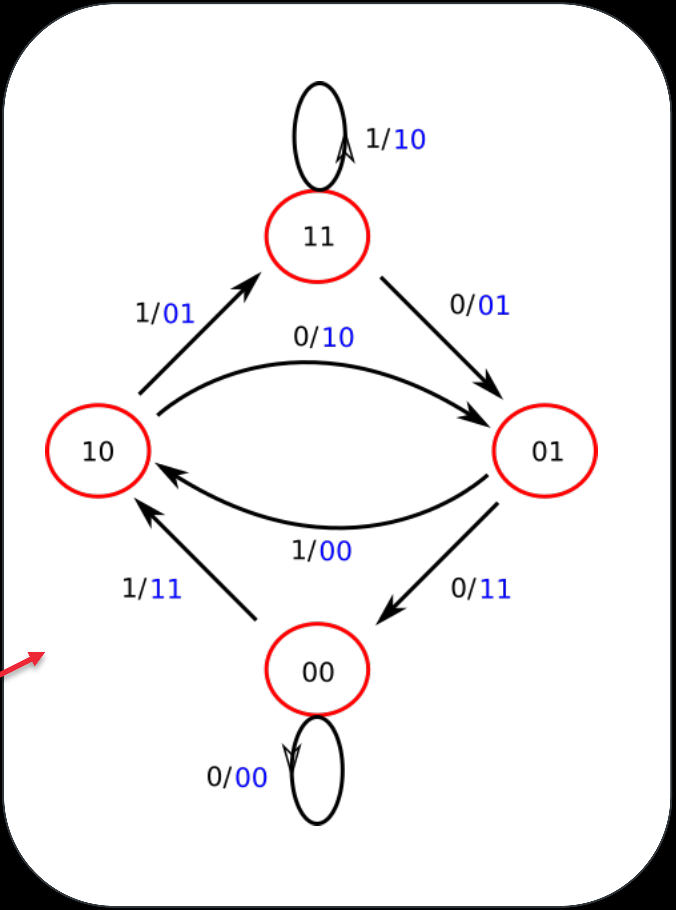
Time	Input	Register 1	Register 2	Register 3	Output
	10110	0	0	0	00
	01011	0	0	0	00 00
	00101	1	0	0	11 00 00
	00010	1	1	0	10 11 00 00
	00001	0	1	1	10 10 11 00 00
	00000	1	0	1	00 10 10 11 00 00

A *(n, r)* convolutional code has an n-bit shift register and r times as many bits on the input as on the output, i.e. its code rate is *r*

Viterbi decoding

Andy Viterbi calls it "maximum likelihood convolutional decoding"





We define the decoder state by the shift register minus the first stage

Expanding the state to get a trellis

Each time slot has a decoder state

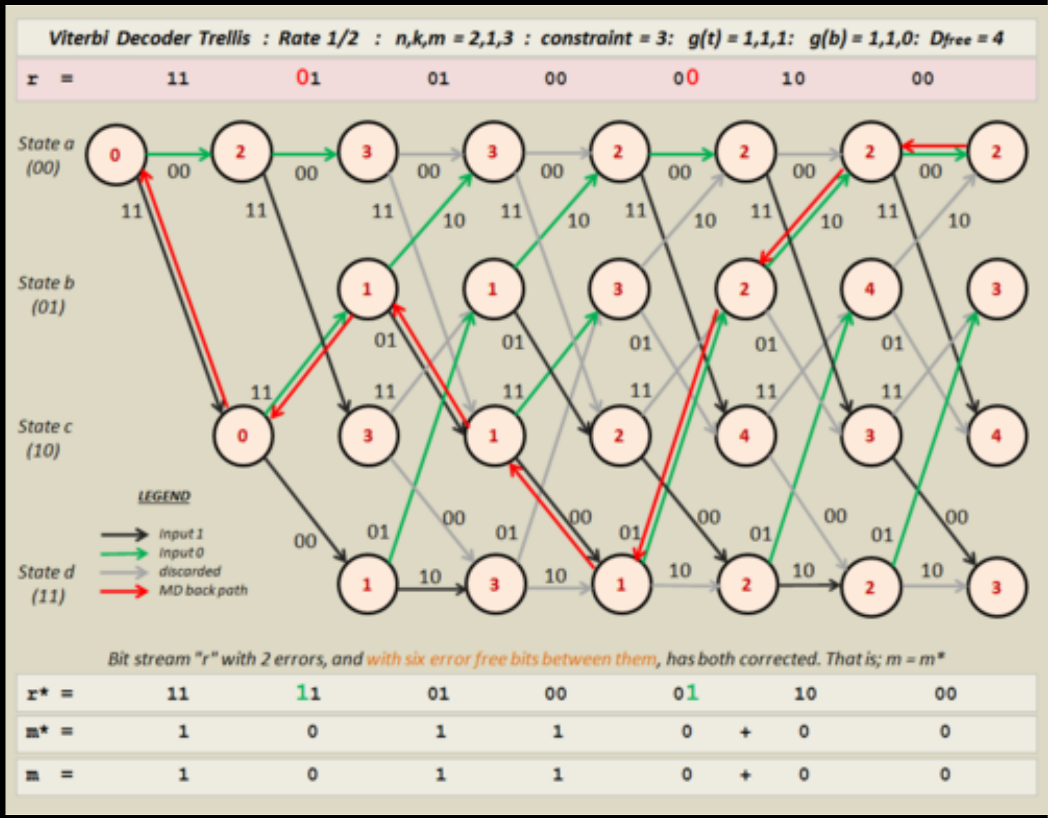
For each state at time t, we could have gotten there by one of two paths

We calculate the most likely of these and “forget” the other

We maintain the most likely path leading to each state – going back a finite distance in the trellis

Called the *traceback length*

At each time, we output the earliest bit from the best surviving path



Node Synchronization

Although convolutional codes have no codeword boundaries, we still have a synchronization challenge: finding the position of the parity checks in the symbol stream

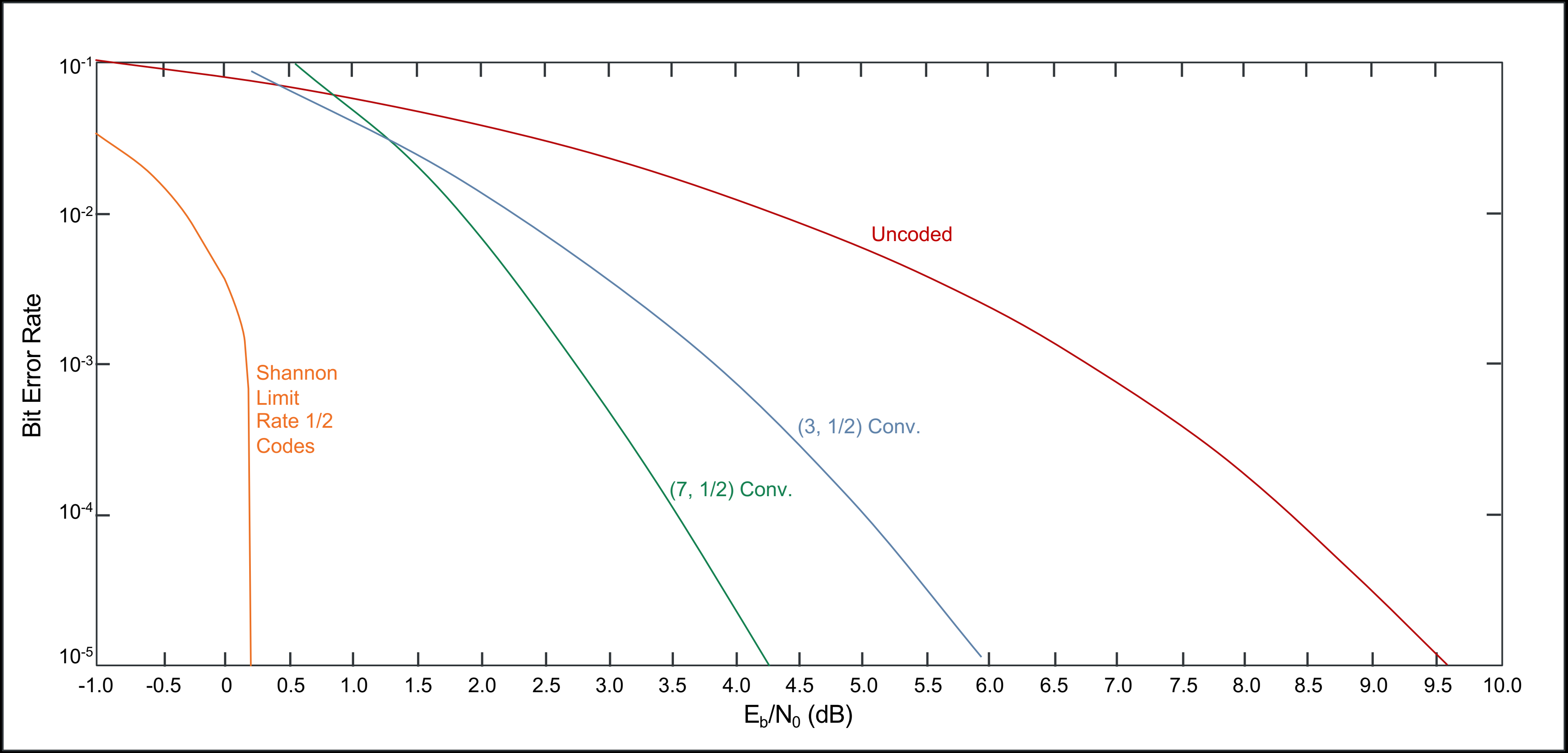
This is called *node synchronization*

Node sync is usually achieved by guessing a value and seeing if the decoder succeeds in finding meaningful bits! If not, we slip a symbol and try again.

Performance of convolutional codes

Here are performance curves for some of the convolutional codes we have used in deep space – the most common is the (7, ½) code

These codes offer substantial coding gain for very little encoder complexity



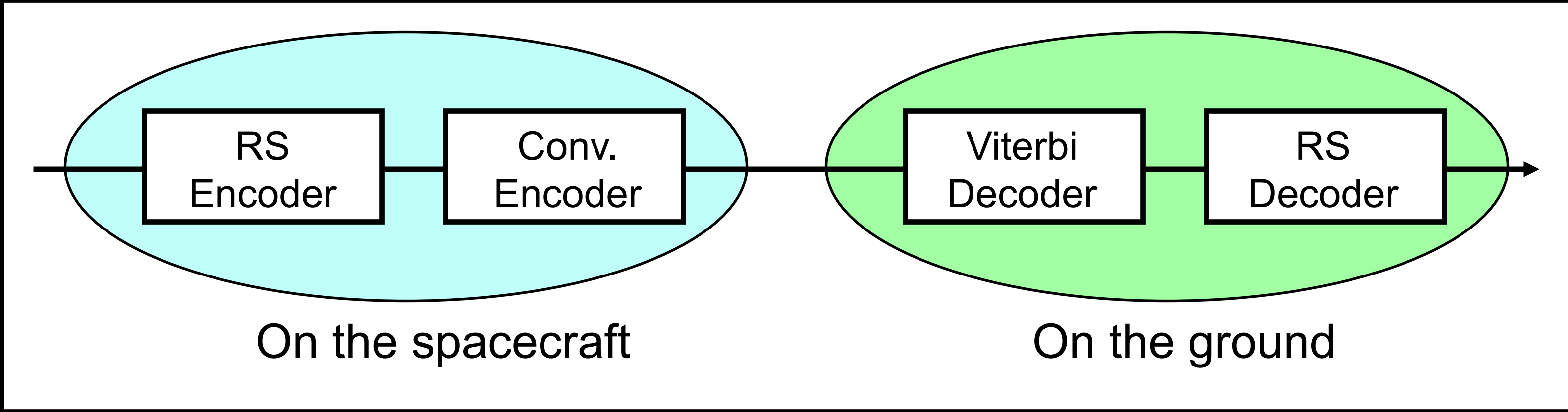
Concatenated coding

Now the down side!

Because of the decoding algorithm, undetected errors appear in clumps

Not all bits in these clumps are errors – but the density of errors is higher in the clumps

This leads to strategies to “clean up” errors from convolutional decoding



Concatenated coding – why it works

The (255, 223, 33) RS code uses 8-bit elements of GF(256) instead of bits

Hence, each error it corrects is actually a *byte error*, a string of 8 bits

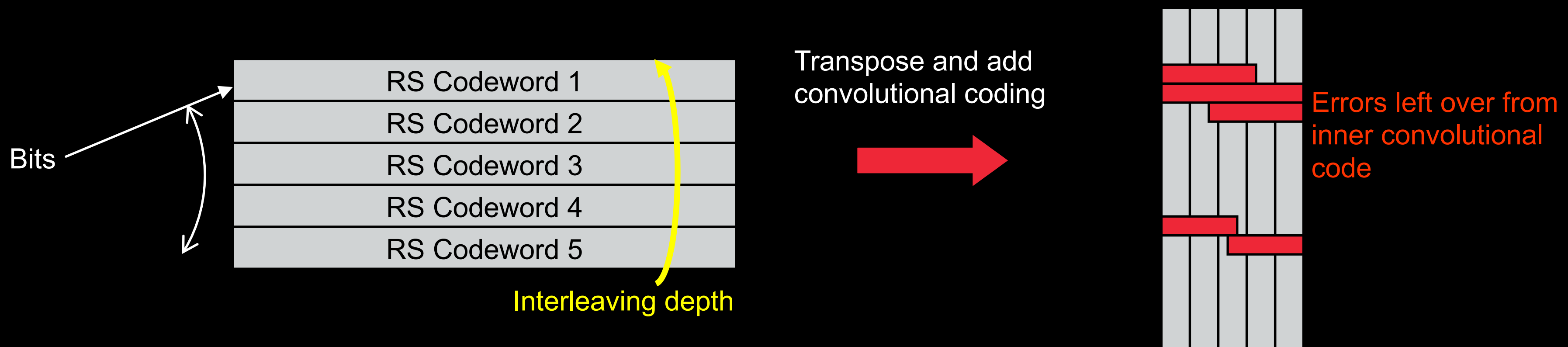
This simple concatenated code can clean up patterns of errors up to 8 bits long caused by the Viterbi decoding algorithm

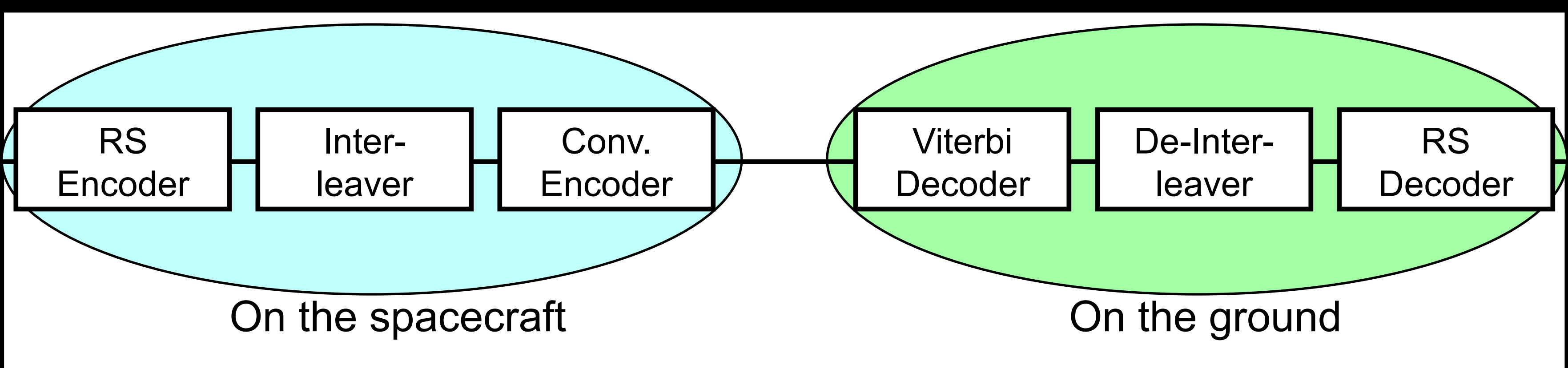
But what if the SNR on our channel is so low that we expect error sequences much longer than 8 bits will be common?

Interleaving

Concatenated coding is enhanced by using *interleaving*

- Distributes bursty errors from convolutional inner code among several Reed-Solomon codewords
- Decreased the errors per Reed-Solomon codeword, resulting in fewer words which fail to decode





Synchronization in concatenated codes

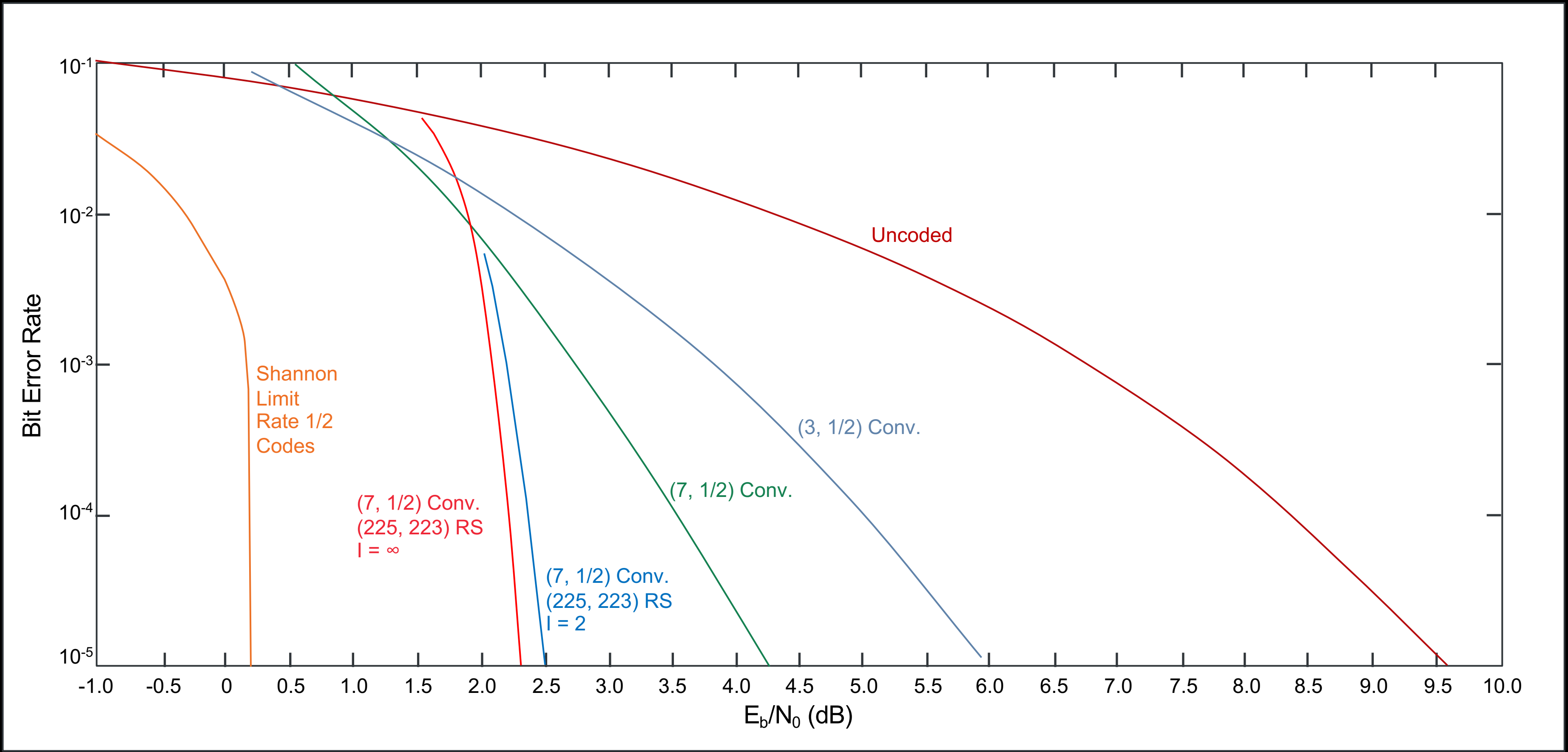
Because we now have an outer block code, we can use the periodic sync marker for all code synchronization

Placing the marker once per interleaved set of codewords suffices

Finding the image of that marker through the convolutional code in the underlying symbol stream points to all of these

- Position of each interleaved codeword
- Block code boundaries
- Node synchronization

Performance of concatenated coding systems



Note that an interleaving depth of I=5 would look essentially like the I=∞ case

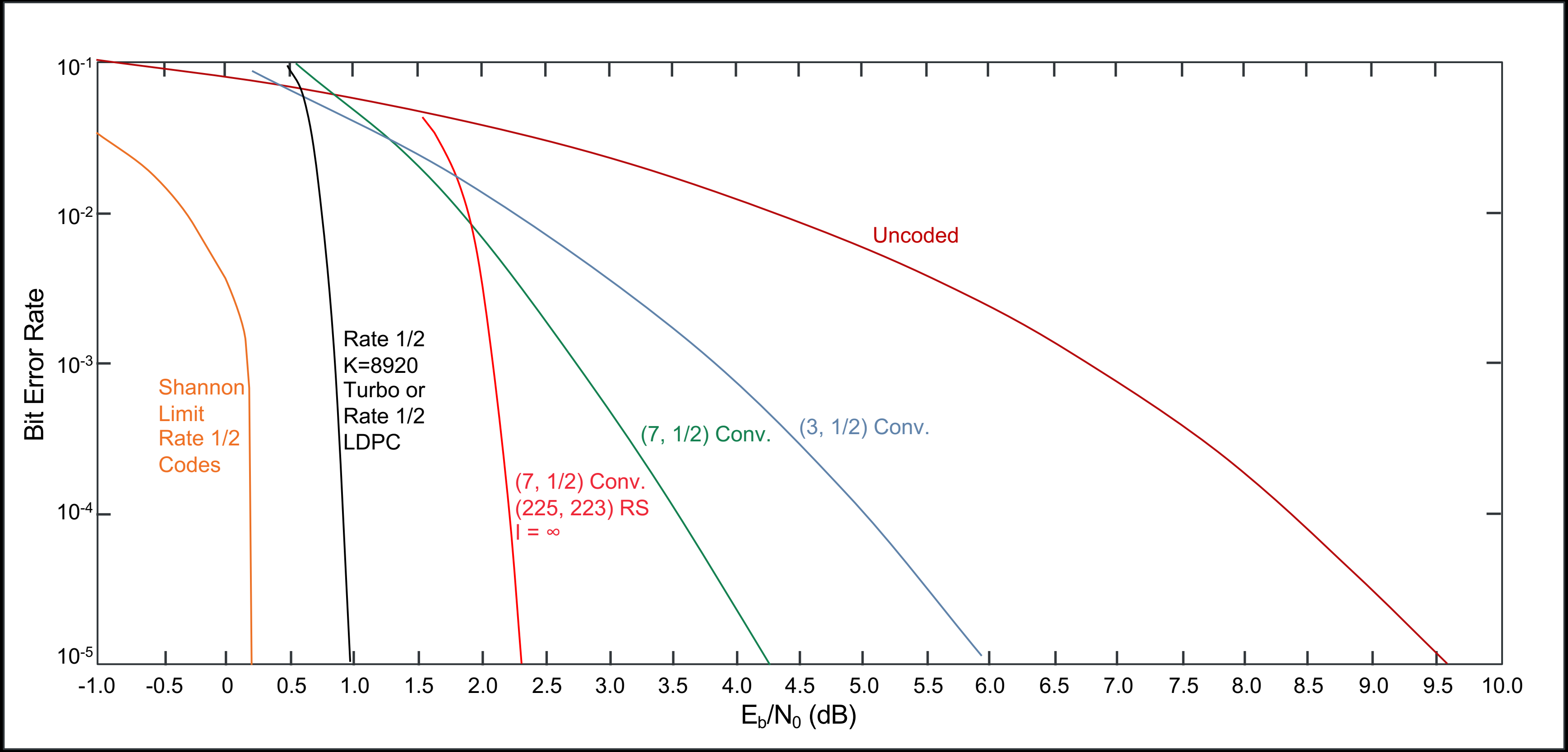
More advanced codes

Turbo codes

- Codes that are based on hybrid convolutional coding and interweavers

Low Density Parity Check (LDPC) codes

- Large block codes with sparse generator matrices, performance is equivalent to Turbo codes
- Spacecraft encoder is very simple, allowing higher data rates without lots of hardware/software



4. Design Control Table Example

Design Control Tables (Link Budgets)

Design Control Tables (DCT) are a tool that spacecraft telecommunications systems engineers use to predict communications performance at sopecific times in a mission

- DCT are also sometimes called *Link Budgets*

DCTs may be used to document tradeoffs in possible spacecraft and ground system configurations

DCTs are used to calculate the amount of *margin* on the link

- Margin is usually extra power used by the spacecraft to ensure that, even in the case of more-than-expected losses and noise, there is sufficnet power in the link to a specific error rate can be achieved to an acceptable level of confidence

We will look at a real (albeit old) DCT for the Voyager 2 spacecraft as an example

The DCT shows how all we have learned so far comes into play for real operations

Voyager 2 DCT – Part 1

Voy 2 (JSX), 70m/18 kW/12 Hz, 0 dB Rng, 0 dB Cmd, Clr Wthr X-Band TWT LP, HGA/NLC, 160 bps Coded, 2-Way Radio Losses					
Spacecraft 2	Station 43				
Time in Mission 96/001/00/00	Time from Epoch 35065 00:00				
	Design	Fav Tol	Adv Tol	Mean	Variance
Transmitter Parameters					
1) RF Power to Antenna, dBm				40.9	0.04
Transmitter Power, dBm	40.90	0.50	–0.50	40.9	0.04
Transmit Circuit Loss, dB	0.00	0.00	0.00	0.0	0.00
2) Antenna Circuit Loss, dB	0.00	0.30	0.00	0.0	0.00
3) Antenna Gain, dBi	48.20	0.26	–0.26	48.2	0.01
4) Pointing Error, dB	–0.10	0.10	–0.10	–0.1	0.00
Limit Cycle, deg	0.05	–0.05	0.00		
Angular Errors, deg	0.00	0.00	0.00		
Path Parameters					
5) Space Loss, dB	–308.19			–308.2	0.00
Freq = 8415.00 MHz					
Range = 7.273+09 km					
= 48.62 AU					
6) Atmospheric Attenuation, dB	–0.04	0.00	0.00	0.0	0.00

Voyager 2 DCT – Part 2

Receiver Parameters					
7) Polarization Loss, dB	–0.08	0.08	–0.11		
8) Antenna Gain, dBi	74.01	0.60	–0.60	73.7	0.14
9) Pointing Loss, dB	–0.20	0.20	–0.20		
10) Noise Spec Dens, dBm/Hz	–185.35	–0.97	0.80	–185.4	0.09
Total System Noise Temp, K	21.12	–4.24	4.24		
Receiver Temperature, K	13.20	–3.00	3.00		
Ground Contribution, K	2.88	–3.00	3.00		
Galactic Contribution, K	2.68	0.00	0.00		
Atmospheric Contrib, K	2.36	0.00	0.00		
Hot Body Noise, K	0.00	0.00	0.00		
Elev Angle = 58.01 deg					
11) Carr Thr Noise, BW, dB-Hz	14.77	–0.46	0.41	14.8	0.03
Power Summary					
12) Rcvd Power, P_t, dBm (1+2+3+4+5+6+7+8+9)				–145.5	0.19
13) Rcvd P_t/N_0, dB-Hz (12–10)				39.9	0.28
14) Ranging Suppression, dB	–0.22	0.05	0.05	–0.2	0.00
15) Telemetry Suppression, dB	–6.02	0.16	–0.17	–6.0	0.00
16) Carr Pwr/Tot Pwr, dB (14+15)				–6.2	0.00
17) Rcvd Carr Pwr, dBm (12+16)				–151.7	0.20
18) Carr SNR in 2BLO, dB (17–10–11)				19.0	0.31
				2.0S =	1.10

Voyager 2 DCT – Part 3

Data Channel Performance					
19) Data Bit Rate, dB	22.04	0.00	0.00	22.0	0.00
Bit Rate = 160.0 bps					
20) Data Pwr/Total Pwr, dB	–1.25	0.05	–0.06	–1.2	0.00
Tlm Mod Index = 60.0 deg					
21) Data Pwr to Rcvr, dBm (12+14+20)				–147.0	0.19
22) ST/N_0 to Rcvr, dB (21–19–10)				16.4	0.28
23) System Losses, dB	–0.72	0.06	–0.36	–0.8	0.01
Radio Loss, dB	–0.18	0.02	–0.02		
Demod, Detect Loss, dB	–0.36	0.04	–0.36		
Waveform Dist Loss, dB	–0.18	0.04	–0.03		
24) Data Bit Rate, dB (22+23)				15.6	0.29
25) Data Bit Rate, dB	2.34	0.00	0.00	2.3	0.00
26) Data Bit Rate, dB (24–25)				13.3	0.29
				2.0S =	1.10

5. Antenna Arraying

Basics of antenna arraying

We can use two or more antennas to synthesize a virtual antenna of a larger size in order to:

- Create a virtual antenna larger than we either have or can afford to build
- Create a greater capability that is only needed for a limited time
- Allow greater flexibility in the use of existing antennas

Can be used on a spacecraft or on the ground

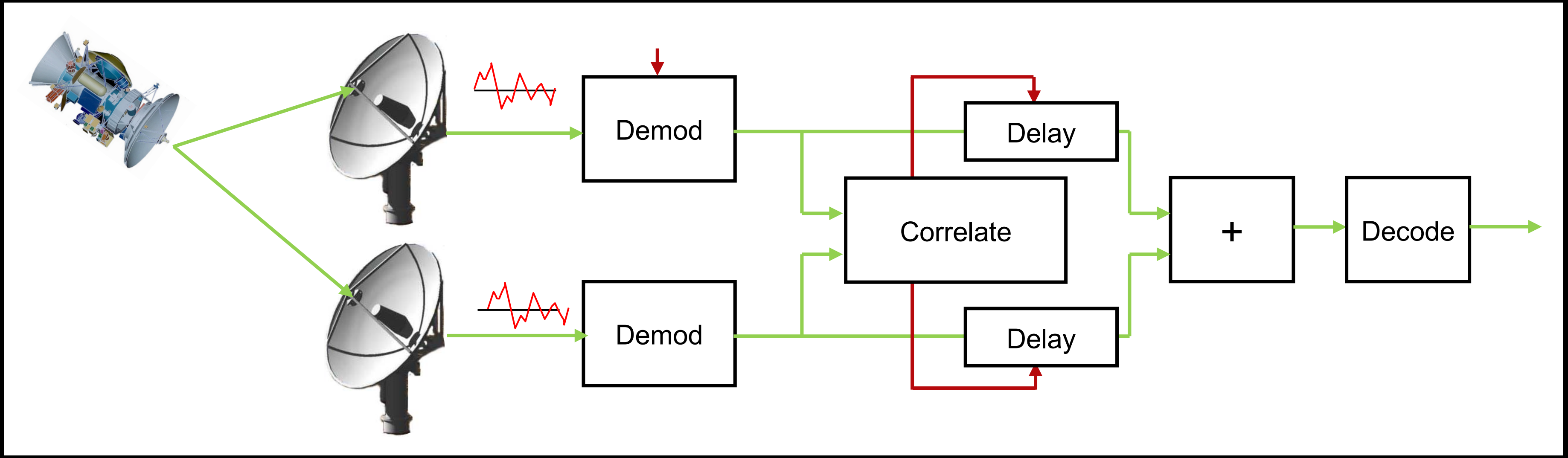
- In practice, it has only been used on Earth

Can be used for uplink or downlink

- In practice, it is MUCH simpler to use for downlink
- Uplink has been demonstrated, but is not operational for deep space applications at the moment

Symbol stream combining

If there is sufficient signal in each antenna for demodulation, then this scheme works very well

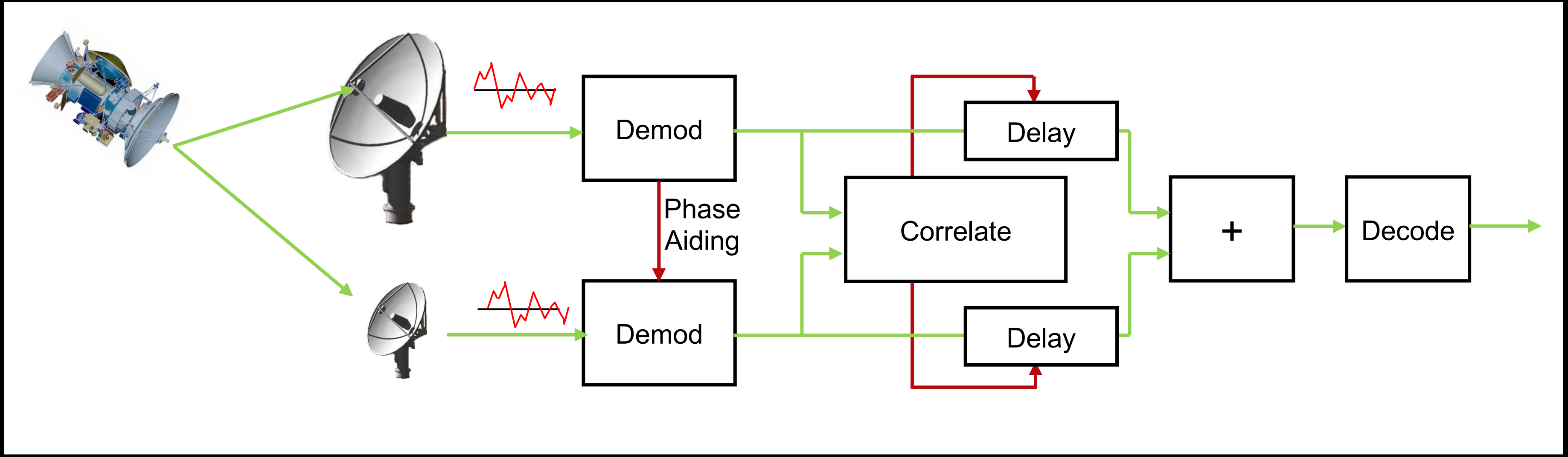


This was NASA's first operational arraying scheme

Used in Voyager's encounters with Uranus and Neptune

Master/Slave combining

If there is sufficient signal in one antenna for demodulation, then the phase estimates in its loop can be used to “aid” the loops in smaller antennas

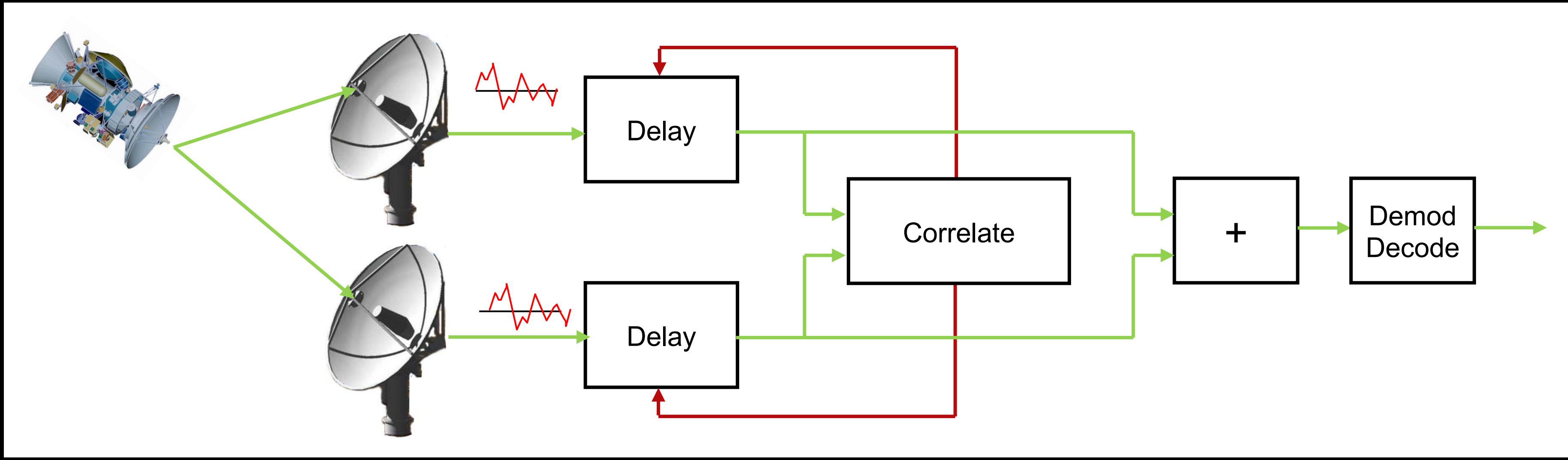


Some noise is common to all antennas, some is separate in each antenna path

The loops in the smaller antennas only have to track their sperate noise

Carrier combining

In carrier combining, the signals from each antenna are combined and the resulting sum is demodulated and decoded



Carrier arraying performance

The performance of the downlink is basically the performance one would get with an antenna with aperture equal to the sum of the apertures of the array elements

$$(G/T)_{array} = C \sum_{i} (G_i/T)$$

Where G_i is the gain for the *i*th element.

We assume all the noise is common

C is the combining loss

The performance of the other downlink arraying systems is more complex and can be found in the literature

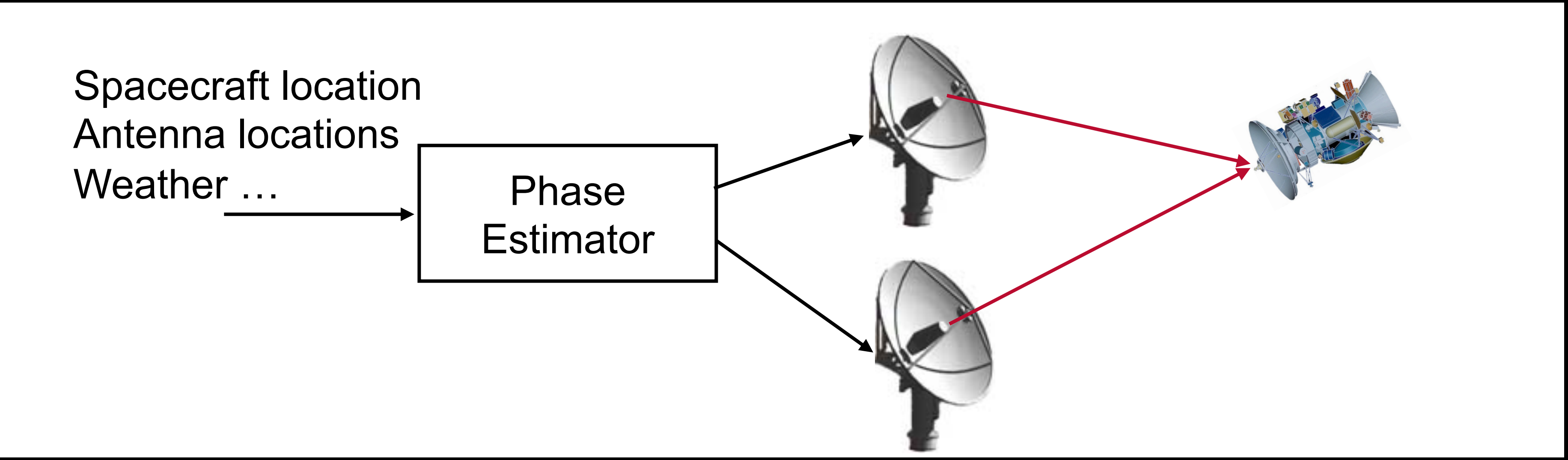
Uplink arraying

Unlike downlink, there are no signals to correlate

We have to estimate the phasing correctly on the ground

However, if we get it right, we are rewarded with a bonus

The voltages all add up coherently at the spacecraft giving us the sum of the squares on the performance of the individual transmitters



Uplink arraying performance

In ideal uplink arraying, the amplitudes of the transmitted signals line up coherently at the spacecraft

Hence, the total gain power product would be

$$(GP)_{array} = \Big(\sum_i \sqrt{(GP)_{element_i}}\Big)^2$$

In the special case of equal size transmitting antennas with equal sized transmitters, we get

$$(GP)_{array} = n^2 (GP)_{element}$$

Which is the famous “n^2” effect

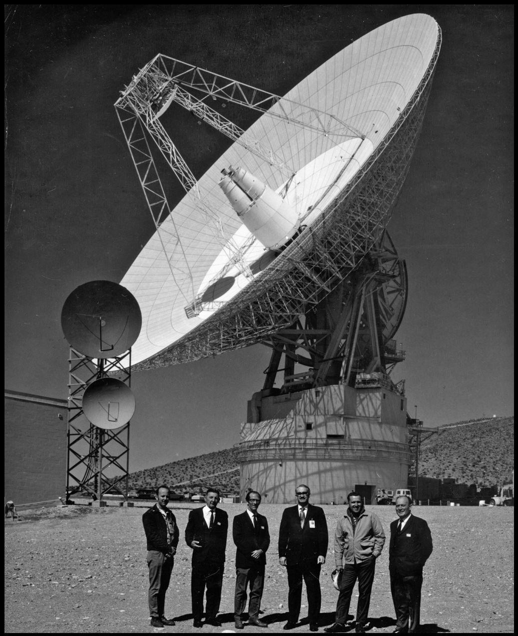
However, there are losses caused by misalignment of the phases at the spacecraft.

$$(GP)_{array} = C\, n^2 (GP)_{element}$$

6. Performance History

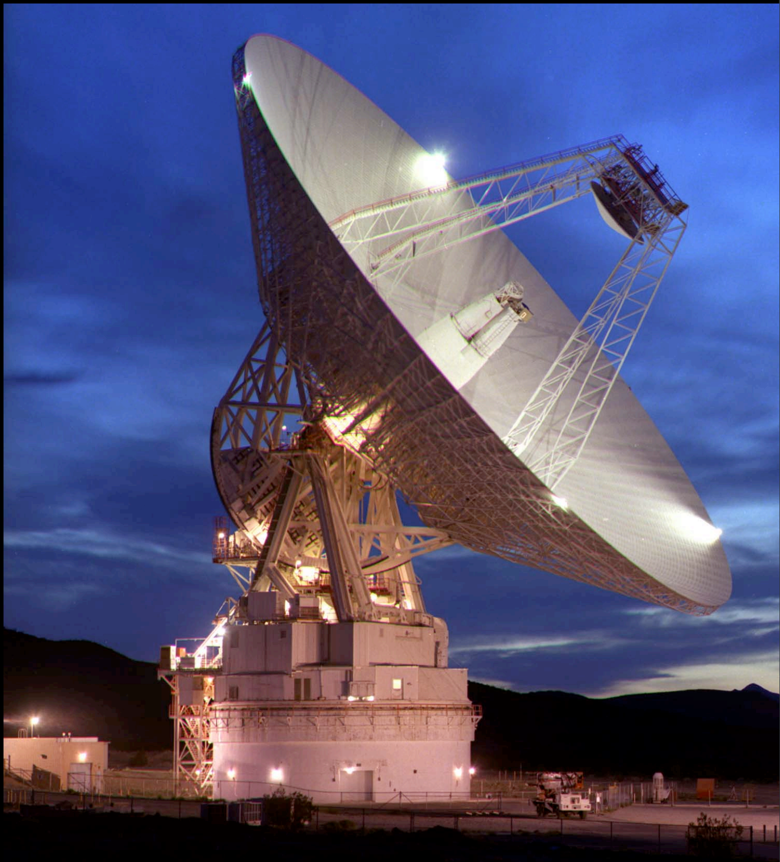
History of Ground Antennas

1958, 26m Station



1966, 64m Station

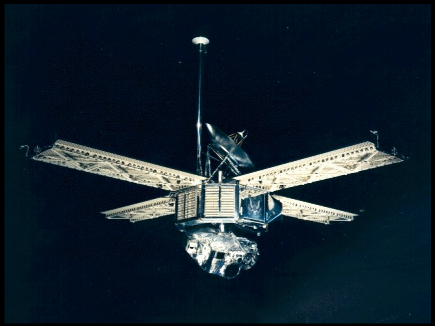
1979, 34m Station



1988, 70m Station
(converted from prior 64 antennas)

History of Spacecraft Antennas

1964, 0.12m Mariner 4



1966, 1m Mariner 6

1980, 1.5m Viking

2005, 3m Mars Reconnaissance Orbiter

History of Spacecraft Transmitter Power

1964, 10W Mariner 4

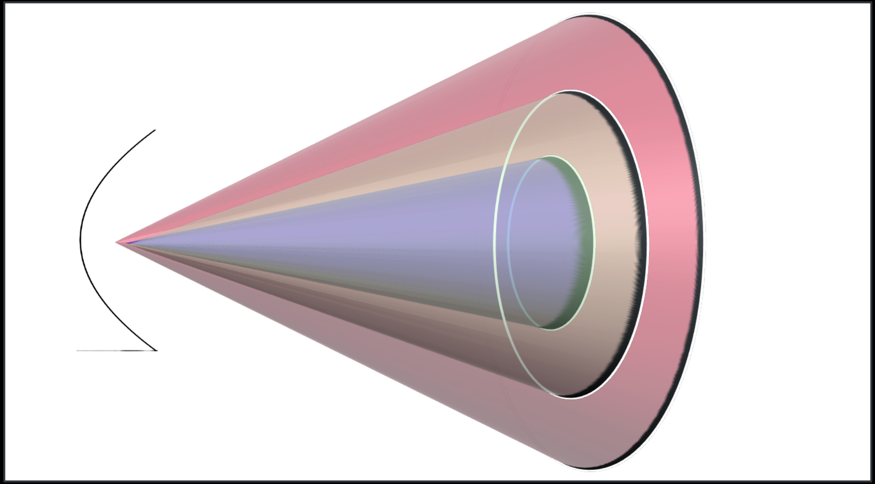
1980, 20W Viking

2005, 100W Mars Reconnaissance Orbiter

Higher Frequency is Good

Signal to Noise = constant/λ^2

- The first deep space missions transmitted at 960 MHz
- 2.2 GHz (S-band) became standard in 1969
- 8.4 GHz (X-band) became prevalent in the early 1970s
- 32 GHz (Ka-band) is now becoming the standard
- Optical communications is currently in demonstration phase and will become operational in the next decade



Why Ka-Band is not an 11.6 dB improvement

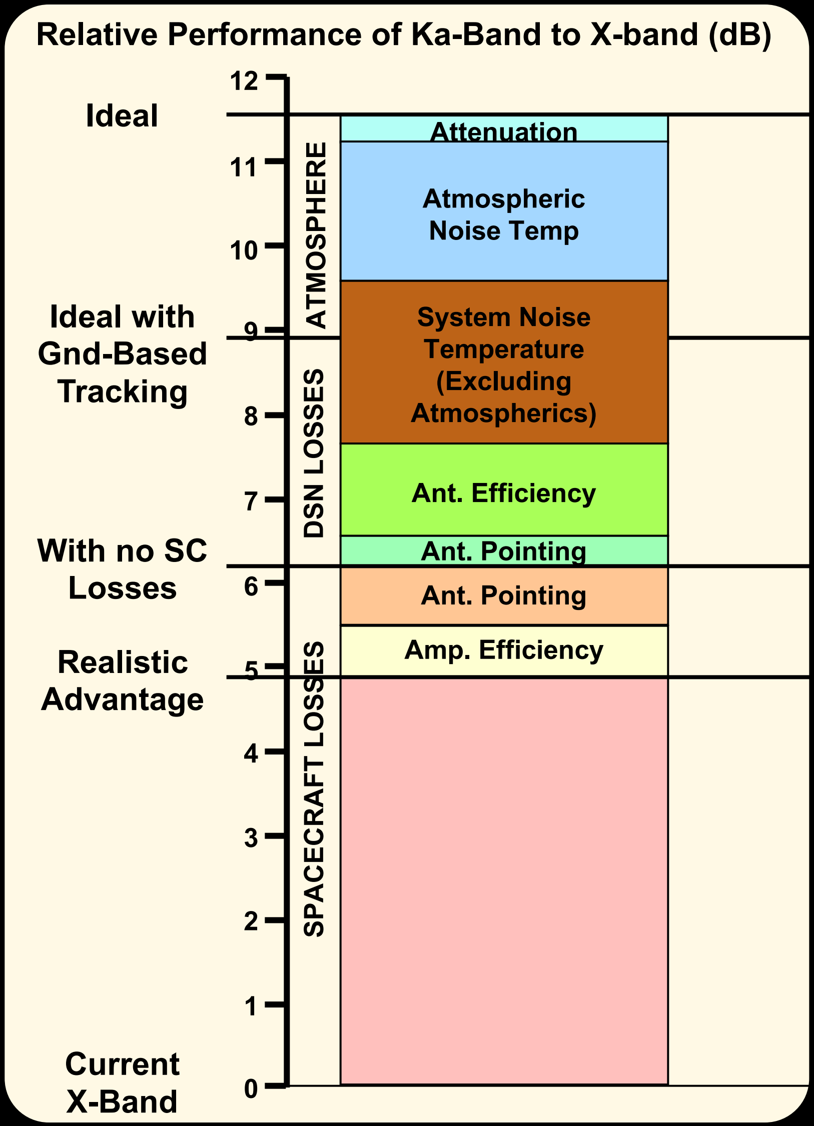
We would expect an 11.6 dB advantage over X-band based on the square of the ratio of frequencies

- Various degradations result in less

This old “thermometer” was a guide to technology investment

- Example of an *error budget*

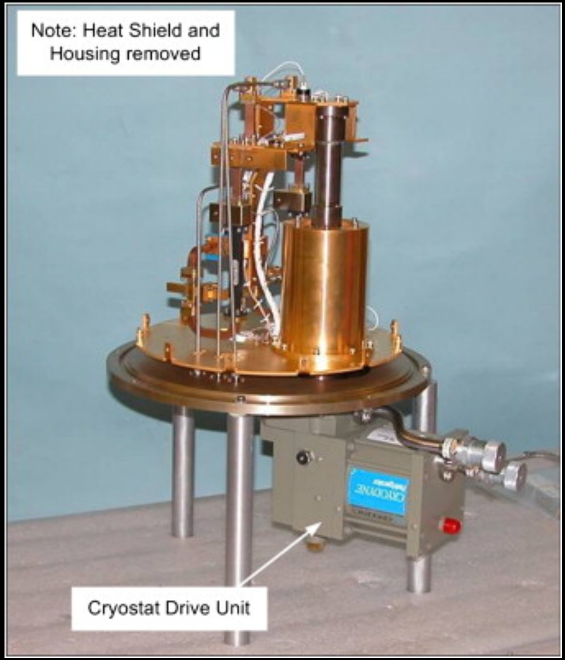
Final result turned out to be 5 to 7 dB advantage, depending on scenario



Lowering the System Noise

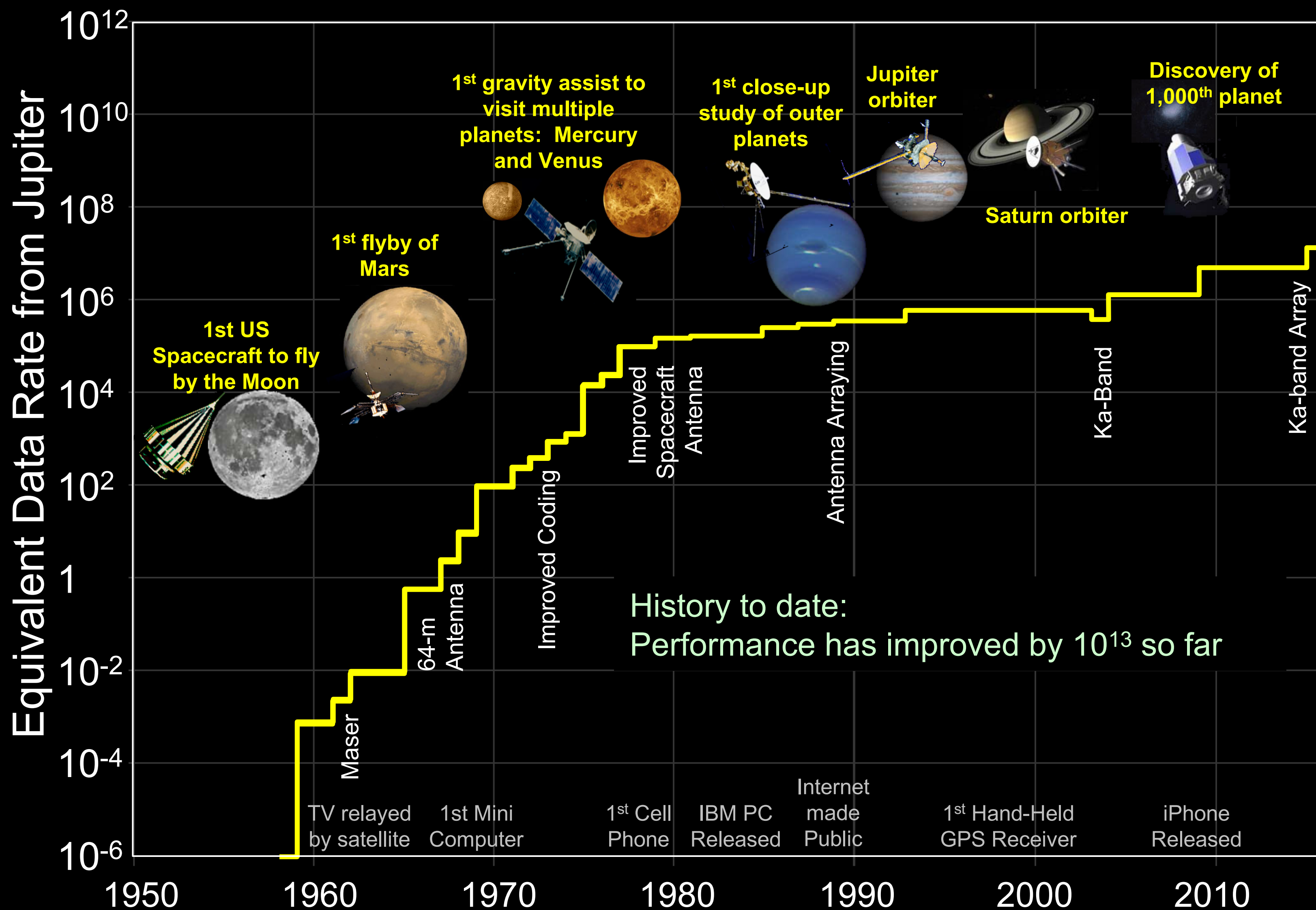
Signal to Noise = constant/T

- Some elements to T cannot be controlled
- We concentrate on the contributions of spacecraft and DSN electronics to T
- We carefully avoid RFI
  - Deep space research has its own spectrum assignments from the ITU
- DSN detectors use the best low noise amplifiers we can build or buy
  - Hydrogen masers or HEMTs
  - Physical temperature is ~12 K



Ka-band (32 GHz) low noise amplifier

A History of Improving Communications



Challenge: Future Missions Need More Data

Science Directions

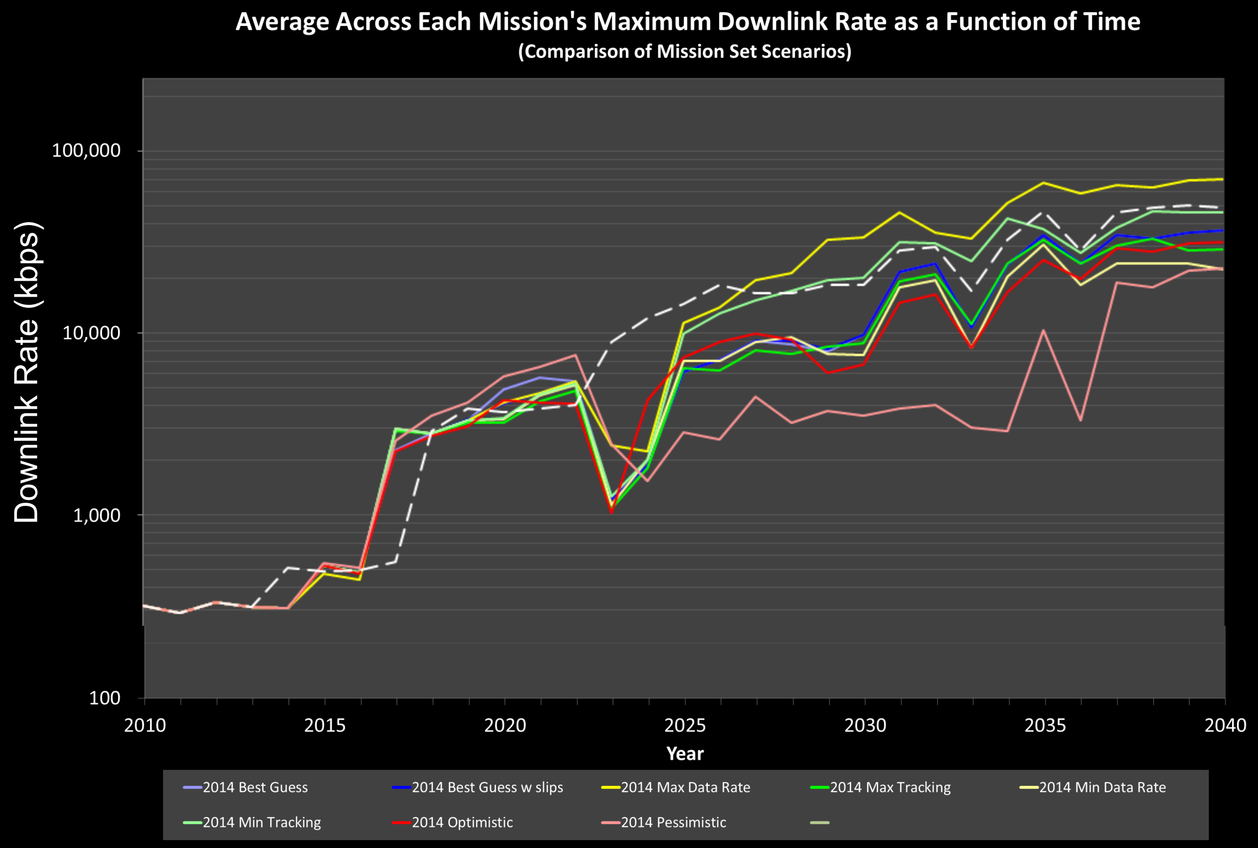
Have visited all major objects in Solar System, Global continuous presence on Mars since 2004

Trends: Revisit for more intense study, Smaller spacecraft and constellations, Humans beyond LEO

Mission modeling indicates desire for ~10X data improvement per decade through 2040

How can we improve even more?

Subject for another course!



7. Multiple Access

Making efficient use of ground antennas

Deep space Earth stations are big and expensive

We want to avoid building more than we absolutely need

We want to use the existing stations as efficiently as possible

If more than one spacecraft is in the beam of an Earth station, we can receive bits from them provided they all “play by a set of rules”

These rules are the various algorithms for *multiple access*

There are several ways to achieve multiple access – several sets of these rules

We will examine the most common methodes

Time Division Multiple Access (TDMA)

TDMA is the simplest form of multiple access

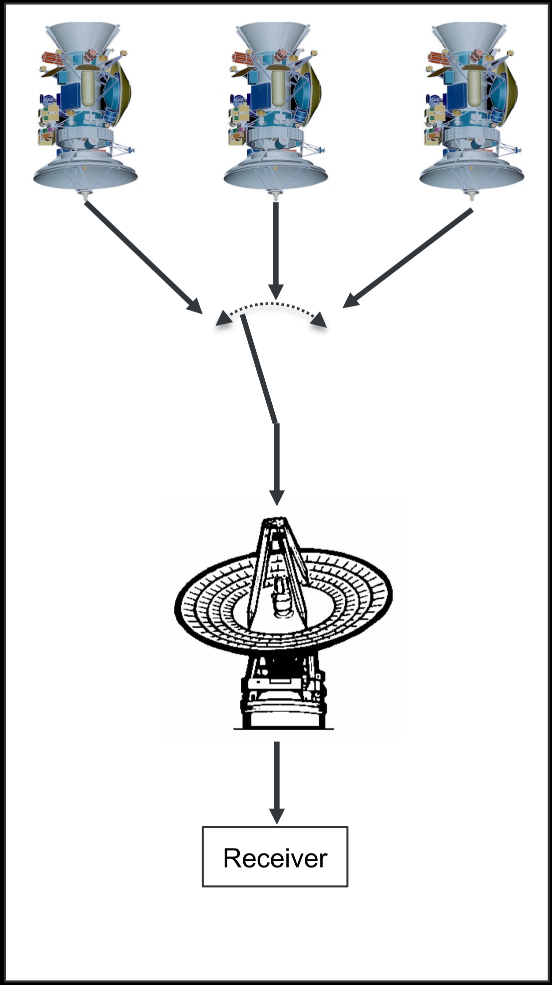
Time is divided into a set of predetermined slots

Each spacecraft is assigned one of these slots

- The spacecraft can transmit (or not) in its assigned slot
- It must be silent in all other time slots

Since the ground system knows about these slots, it is a simple matter to route bits received in the correct slot to the correct mission

- And to send bits to correct spacecraft during the correct slot



TDMA plusses and minuses

Plusses:

- Each spacecraft can use the entire allotted spectrum in its band
- No special processing algorithms are needed

Minuses:

- Clocks must be synchronized among all the spacecraft and Earth
- Special consideration must be given to mission critical events so communications occurs when it is required
- TDMA does not scale well when there are many spacecraft

Frequency Division Multiple Access (FDMA)

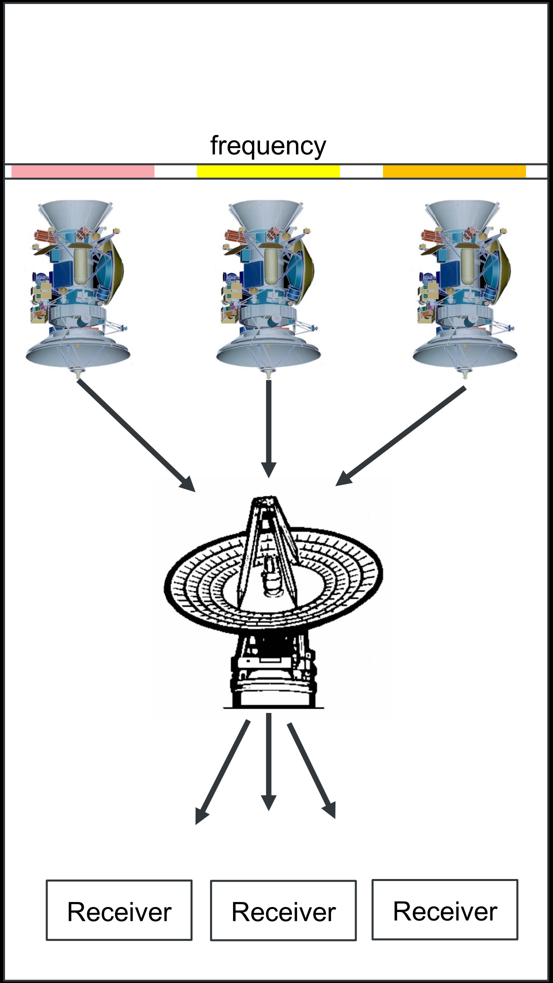
FDMA simply divides the allotted spectrum among the spacecraft

Each spacecraft is assigned a *channel* which is analogous to the channels used for television or commercial radio

Typically there must me *guard bands* to separate the channels to avoid inter-channel interference

The Earth antenna uses multiple receivers, one for each spacecraft

- Phase lock loops
- Decoders



FDMA plusses and minuses

Plusses:

- Clock synchronization is not required
- No special processing algorithms are needed
- Critical events can be tracked without special considerations

Minuses:

- We need multiple receivers
- Only a portion of the allotted spectrum is available to each spacecraft
- FDMA does not scale well when there are many spacecraft

NASA uses both TDMA and FDMA today

- Today, Mars spacecraft constitute most of NASA's deep space multiple access
  - There are many spacecraft at Mars and, when visible to Earth, they are all in the same 34m antenna beam
- For deep space downlink, FDMA is used for all spacecraft today
  - Channels are reused between spacecraft, if we know that they will not ever be in the same beam
  - This allows as much bandwidth as possible to missions
- For deep space uplink, TDMA is used for spacecraft at Mars

Code Division Multiple Access (CDMA)

Each data stream is xor-ed with a much higher frequency PN code and then modulated

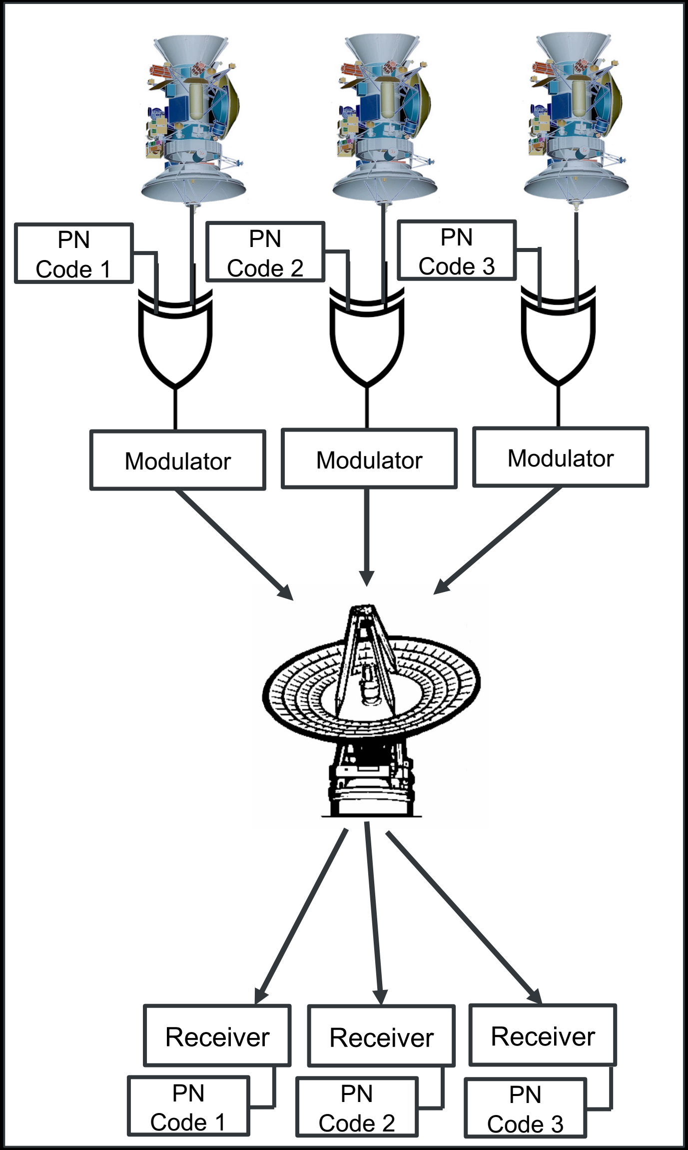
This *spreads* the frequency over a much large bandwidth

Although each transmitted signal occupies the same *spread spectrum*, they can be individually demodulated because the receivers know about the PN codes

This is how most cell phones work today

The system was actually invented by motion picture actress Hedy Lamar during WWII, but kept secret for a long time

- CDMA signals are, by their nature, encrypted
- They can only be understood if you know the PN code



CDMA plusses and minuses

Plusses:

- Clock synchronization is not required
- Critical events can be tracked without special considerations
- CDMA is, perhaps, the best for scaling up to large numbers
- CDMA provides natural encryption for data protection

Minuses:

- We need multiple receivers
- Code synchronization is added to the ground processing
- There are performance problems if the SNRs are not similar among the spacecraft

7. Bibliography

Some of my favorite books in this field

Principals of Communications Engineering, John Wozencraft and Irwin Jacobs, John Wiley & Sons, Inc, 1965.

Theory of Information and Coding, Encyclopedia of Mathematics and Its Applications Vol. 86, Robert McEliece, Cambridge University Press, 2002.

Deep Space Telecommunications Systems Engineering, Joseph Yuen, Editor, JPL Publication 82-76, 1982.

Uplink-Downlink, A History of the Deep Space Network, Douglass Mudgway, NASA Publication SP-2001-4227, 2001.

Large Antennas of the Deep Space Network, William Imbriale, Wiley-Interscience, 2003.

Antenna Arraying in the Deep Space Network, David Rogstad, Alexander Mileant, & Timothy Pham, Wiley-Interscience, 2003.

Spread Spectrum Communications (Volumes I, II, III), Marvin Simon, Jim Omura, Robert Scholtz, & Barry Levitt, Computer Science Press, 1985.

Rad

The *erf* e