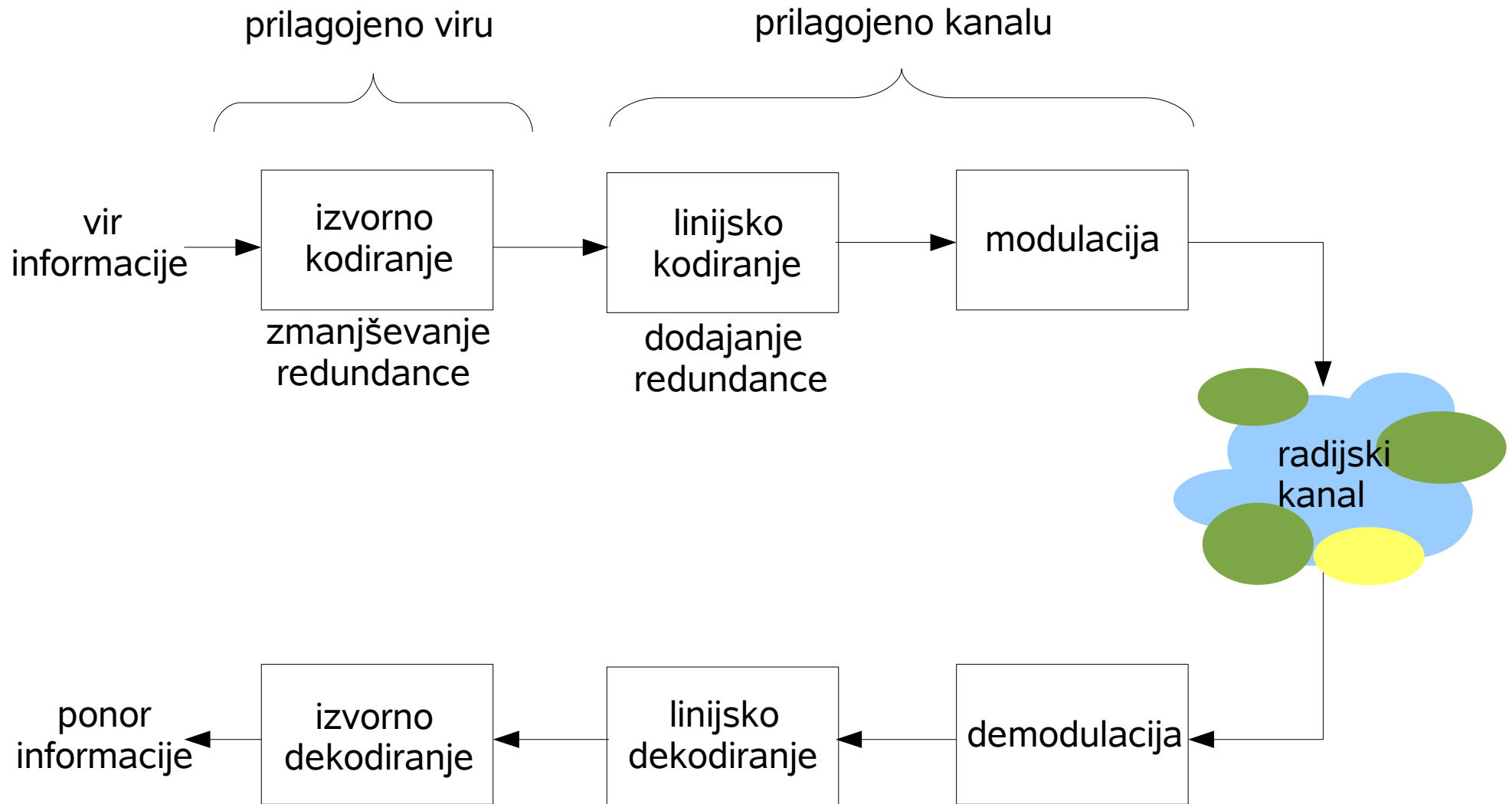


Novi načini dela na EME zvezah

Marko Čebokli S57UUU

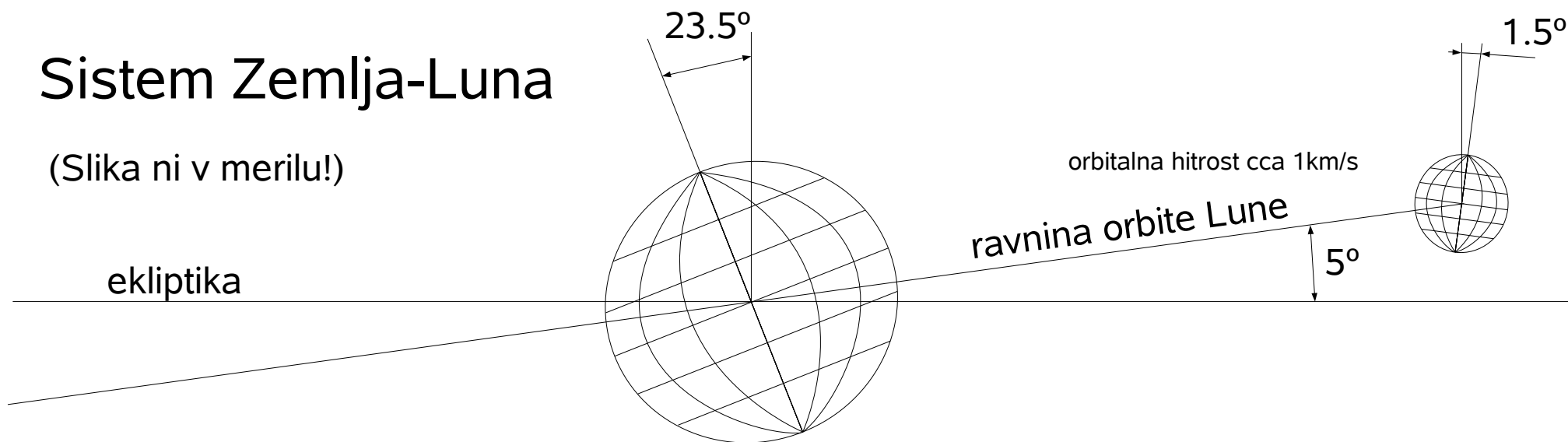
RIS-08

Radijska zveza



Sistem Zemlja-Luna

(Slika ni v merilu!)



Siderični mesec = 27.321661 dni (zvezde)

Tropski mesec = 27.321582 dni (pomladišče)

Anomalistični mesec = 27.554551 dni (apside)

Drakonični mesec = 27.21222 dni (vozli)

Srednji sinodski mesec = 29.530589 dni (sonce)

(cca 29.7 do 29.83 dni)

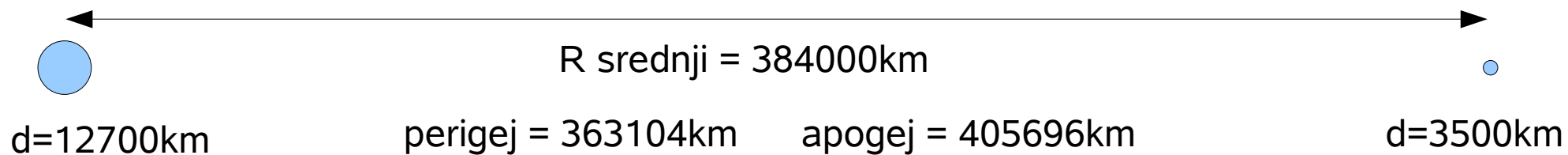
perioda precesije vozlov = 18.6 let regr.

perioda precesije osi = 18.6 let, sinhrono,

zato je nagib osi proti ekliptiki konstantno 1.5°

perioda precesije apsid = 8.85 let prog.

Slika v merilu:



Nivo signala

$$P_r = P_t G_t * \frac{1}{4 \pi r^2} * \pi r_l^2 \eta * \frac{1}{4 \pi r^2} * \frac{G_r \lambda^2}{4 \pi}$$

ekvivalentna oddajna moč (pointing to $P_t G_t$)
 razširjanje na poti tja (pointing to $\frac{1}{4 \pi r^2}$)
 odboj (pointing to $\pi r_l^2 \eta$)
 razširjanje na poti nazaj (pointing to $\frac{1}{4 \pi r^2}$)
 efektivna površina sprejemne antene (pointing to $\frac{G_r \lambda^2}{4 \pi}$)

$$r = 3.84 * 10^8$$

$$r_l = 1.74 * 10^6$$

$$\eta \approx 7\%$$

$$P_r/P_t = G_r G_t \lambda^2 * 1.31 * 10^{-26}$$

Klasični načini:
min S/N ≈ 10dB

Band	P_t	$G_r = G_t$	P_t/P_r	T_r	B za 10dB S/N	C
2m	60dBm	20dBi	213dB	500K	8Hz	27b/s
23cm	53dBm	30dBi	212dB	60K	16Hz	55b/s
3cm	45dBm	48dBi	193dB	100K	120Hz	400b/s

$$N = B * k_B * T$$

$$k_B = 1.38 * 10^{-23} \text{ J/K}$$

Shannon:

$$C = B \log_2(1 + S/N)$$

perigej - apogej
cca ± 0.95dB

Disperzija (razpršitev signala v časovnem in frekvenčnem prostoru)

Luna ni “polirana krogla”, ampak ima razgibano površino. Zaradi vrtenja zemlje, eliptičnosti Lunine orbite in nagnjenosti osi obeh, se površina Lune glede na opazovalca na površini Zemlje premika.

Odbiti signal je vsota prispevkov množice reflektorjev (hribi, kraterji, skale...)

Reflektorji so naključno porazdeljeni po razdalji (razpršitev signala po času, faza je naključna)

Reflektorji se glede na oddajnik in sprejemnik premikajo z različnimi hitrostmi (razpršitev signala po frekvenci - Doppler)

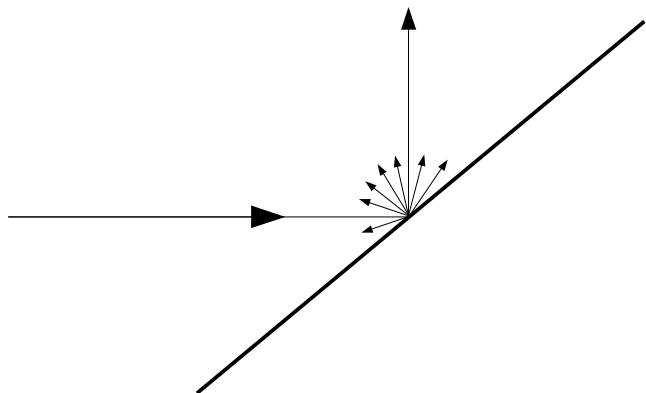
Večino Dopplerja pri EME zvezi povzroča vrtenje Zemlje okoli svoje osi

Na nižjih bandih prevladuje zrcalna, na višjih pa difuzna komponenta odboja

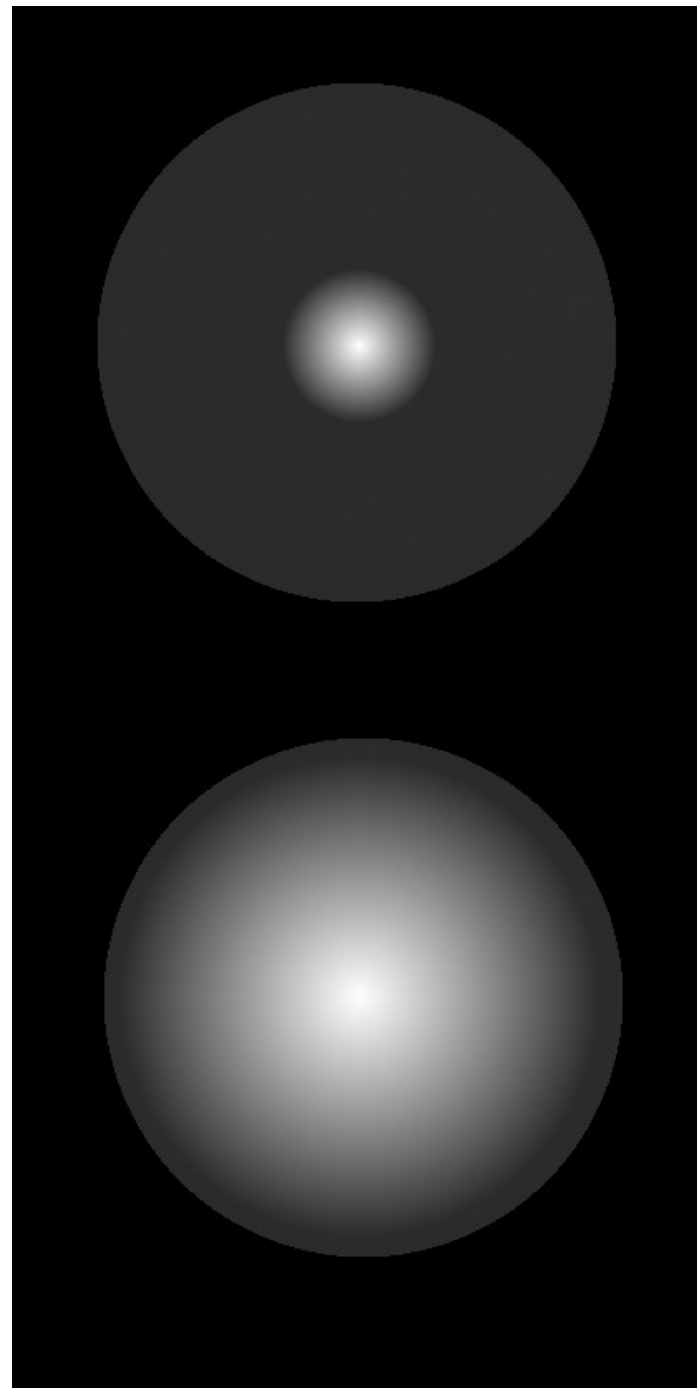
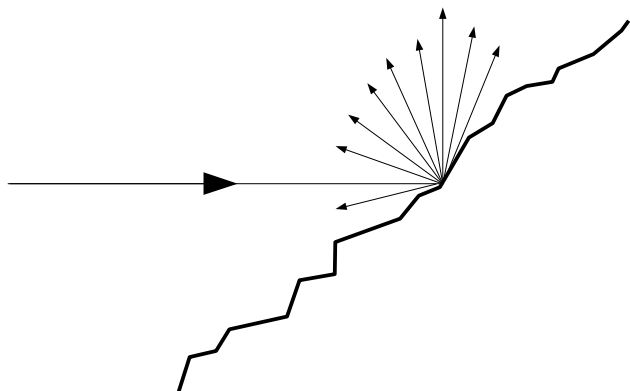
Razpršitev po frekvenci narašča hitreje kot linearno s frekvenco
Časovna “globina” Lune je večja na višjih bandih

Zrcalna in difuzna komponenta odboja

Če so neravnine na odbojni površini majhne v primerjavi z valovno dolžino, prevladuje zrcalna komponenta (nižji bandi)

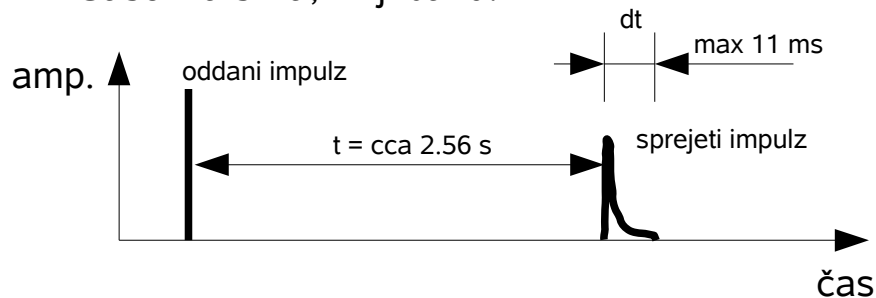


Če so neravnine na odbojni površini velike v primerjavi z valovno dolžino, prevladuje difuzna komponenta (višji bandi)

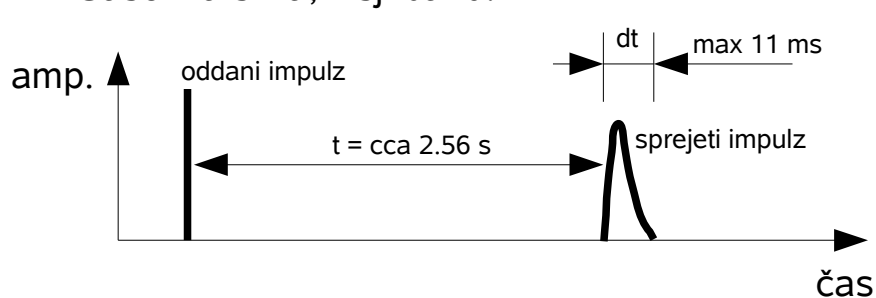


Disperzija

Časovna slika, nižji band:

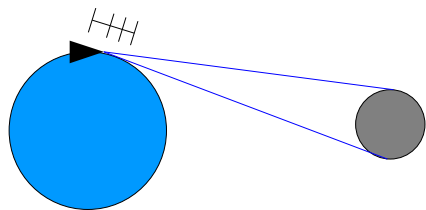
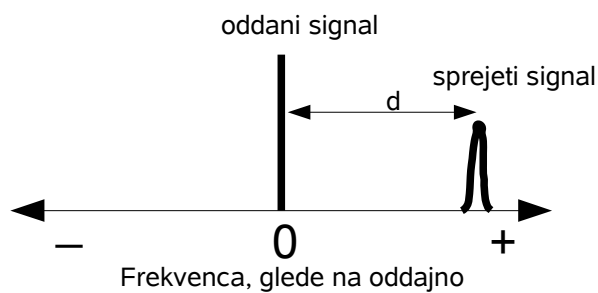


Časovna slika, višji band:



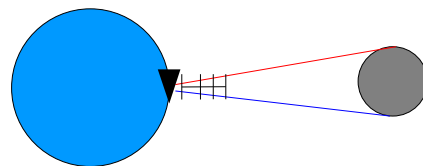
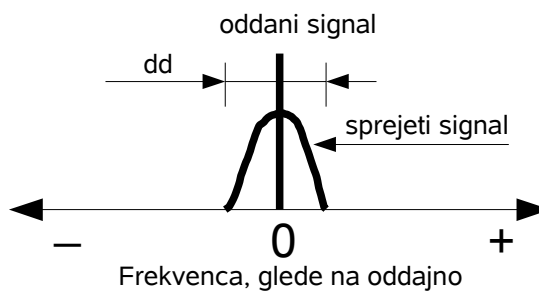
Frekvenčna slika, lasten odmev

Luna vzhaja:



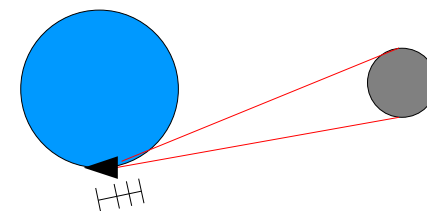
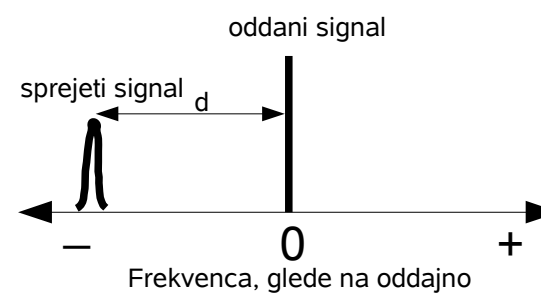
2m: $dt_{90\%} \approx 0.25\text{ms}$ $dd \approx 0.8\text{Hz}^*$

Luna kulminira:



23cm: $dt_{90\%} \approx 0.5\text{ms}$ $dd \approx 10\text{Hz}$

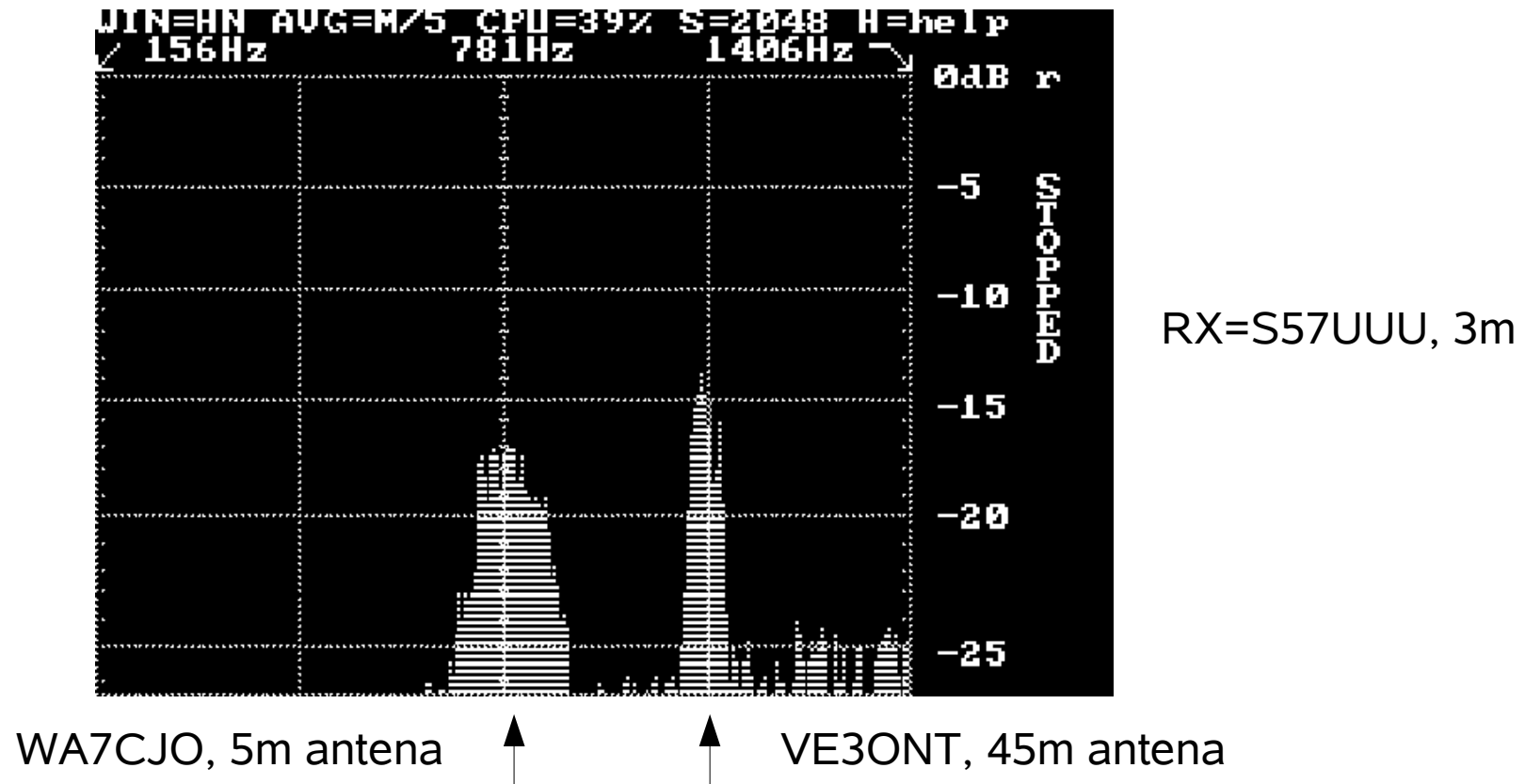
Luna zahaja:



3cm: $dt_{90\%} \approx 1.7\text{ms}$ $dd \approx 133\text{Hz}$

* znotraj teh meja je 90% energije, 3dB vrednosti so cca dvakrat do trikrat manjše

Spekter odbitega signala na 10GHz



Večja antena osvetljuje manjši del Lune, zato je razpršitev po frekvenci manjša

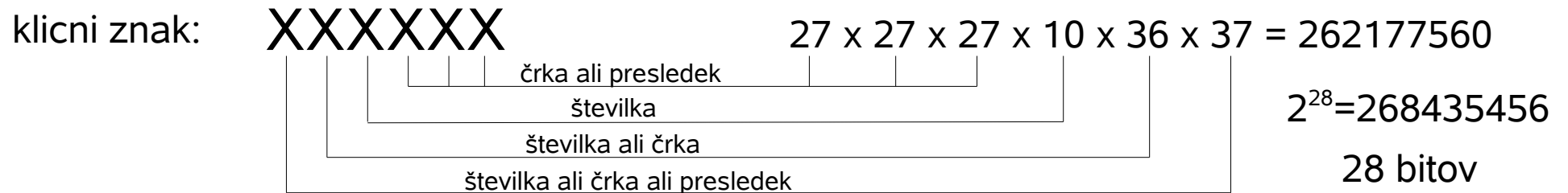
Primerjava nekaterih radijskih kanalov

	NASA DSN	Urbani mobilni	EME
nivo signala	ekstremno nizek	relativno visok	nizek
število poti	1	1...20	več tisoč
max disperzija po času	-	20us	11ms
max razpon Doppler hitrosti	-	m/s	km/s

Zaključek: EME radijski kanal se precej razlikuje od radijskih kanalov, za katere so bile razvite sodobne radijske tehnologije (GSM, WiMax itd...). Najpomembnejša razlika je, da zaradi naključne faze ne moremo uporabiti koherentnih tehnik. Uporabimo pa lahko močne metode korekcije napak, ki omogočajo dobro kvaliteto zveze pri slabših razmerjih S/N.

Koliko informacije prenašamo v EME zvezi?

ZNAK1 ZNAK2 INFO (info = raport, potrditev, lokator...)



Če kot “info” prenašamo samo O, M, RO, RM, QRZ? in 73, zadostujejo trije dodatni biti

Skupaj: $28 + 28 + 3 =$ cca 59 bitov

Če pa želimo prenesti še prve štiri znake deviškoglavega lokatorja, torej $180 \times 180 = 32400$ “kvadratov”, pa rabimo 15 bitov, torej skupaj 71 bitov

Klasika: MORSE

V klasični radiotelegrafski zvezi Morse opravlja funkcijo tako izvorne kot linijske kode.

Morse kot izvorno kodiranje

Morse je za odprt angleški tekst nekakšna kvazi Huffman koda (pogostejši znaki imajo krajše kode), vendar pa klicni znaki nimajo statističnih lastnost angleščine, zato to ni učinkovito.

6 mestni klicni znak v Morse kodi ima v povprečju 74.32 bita, proti 28 bitom je to 2.65X redundanca
MORSE NI UČINKOVITA IZVORNA KODA

Morse kot linijska koda

Kljub veliki redundanci v večini primerov ni možno niti odkrivanje napak, še manj pa popravljanje.

MORSE JE ZELO SLABA LINIJSKA KODA

1010101 = H

0010101 = S

1110101 = D

1000101 = EI

1011101 = R

1010001 = IE

1010111 = U

1010100 = S

Popravljanje napak

s povratnim kanalom
(retry, packet radio)

Če sprejemnik zazna napako, po povratnem kanalu od oddajnika zahteva ponovitev prizadetega paketa.

Da je ta metoda učinkovita, mora biti število napak zelo majhno, da je večina paketov brez napak.

brez povratnega kanala
(FEC, forward error correction)

Oddajnik sporočilu vnaprej doda redundantno informacijo, ki sprejemniku omogoča popravljanje napak.

Konvolucijske kode

Za (neprekinjen) tok podatkov, možni so preprosti strojni koderji in dekodeerji.
Za optimalno dekodiranje se uporablja Viterbijev algoritem.

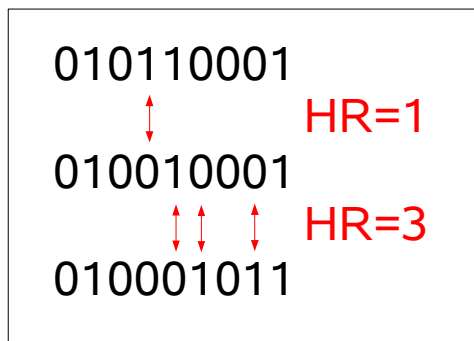
Blokovne kode

Podatki so v ločenih blokih, primerne so predvsem za paketno komunikacijo.

- Hamming kode
- Golay kode
- BCH kode
- Reed Solomon kode
- Turbo kode
- LDPC kode
- ... itd

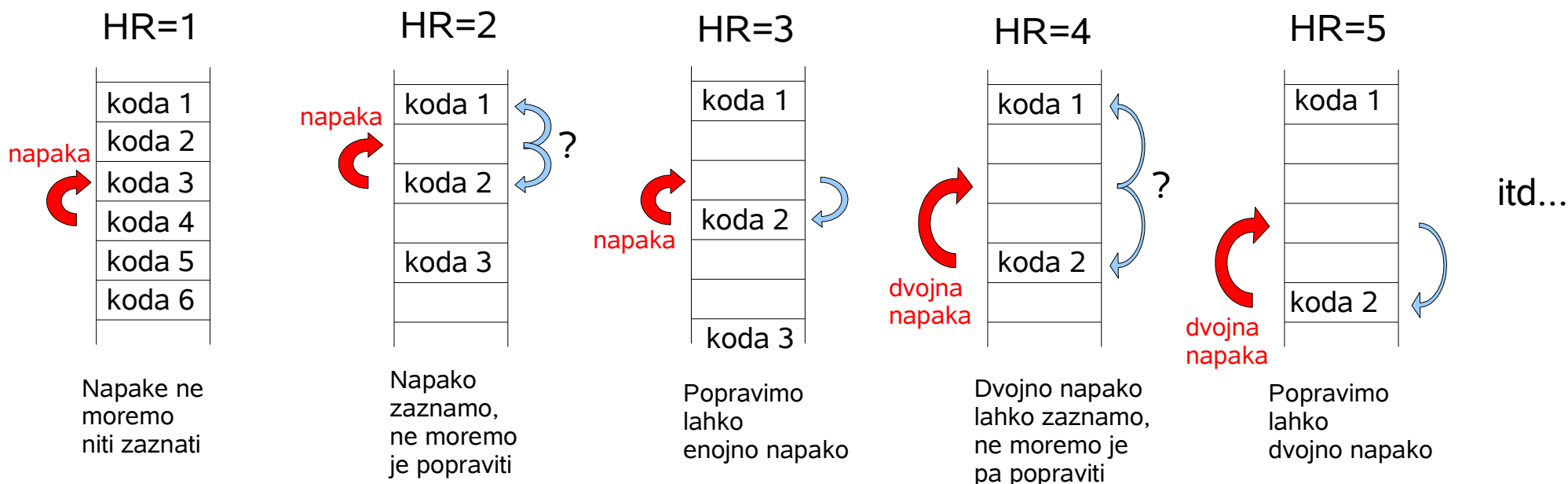
kombinirane kode (NASA DSN, Viterbi + RS)

Hammingova razdalja: na kolko mestih se kodni besedi razlikujeta?



Vsaka posamezna napaka nas v prostoru kod premakne za razdaljo 1. Če uporabimo vse možne kode v danem prostoru, je med njimi razdalja 1. V tem primeru vsaka napaka iz ene veljavne kode naredi drugo veljavno kodo, zato napak ne moremo niti zaznati, kaj šele popravljati.

Če želimo napake zaznavati in popravljati, moramo povečati Hammingovo razdaljo med uporabljenimi kodami. Kodni prostor mora biti večji (večbitne kodne besede), kot bi zahtevali vhodni podatki: dodajamo redundantno informacijo.



Pariteta in Hammingove kode

Zaznavanje enojnih napak: pariteta

Za zaznavanje enojnih napak rabimo Hammingovo razdaljo 2, torej mora med vsakim parom "legalnih" kodnih besed biti vsaj še ena "ilegalna". Prostor kod moramo torej najmanj podvojiti, oz. kodni besedi dodat vsaj še en bit.

Če v kodni besedi, ki vsebuje liho število enic spremenimo en bit, vedno dobimo besedo s sodim številom enic (in obratno). Med vsakim parom besed z lihimi (ali sodimi) števili enic je torej Hammingova razdalja vsaj 2! Če torej od kodnih besed dane dolžine uporabimo samo tiste z lihimi (ali sodimi) števili enic, bomo lahko odkrivali enojne napake.

Algoritem za generiranje ustreznih kod je preprost: če ima izvorna kodna beseda liho število enic, ji dodamo ničlo, sicer pa enico.

Popravljanje enojnih napak: Hammingove kode

Da bi lahko popravili napako v kodni besedi, mora biti iz dodanih bitov možno določiti na katerem bitu se je zgodila napaka. Zato mora v kodni besedi dolžine N biti malo več kot $\log_2 N$ dodatnih bitov: s štirimi dodanimi bitovi lahko popravimo eno napako v enajst bitni besedi, itd.

Richard Hamming je tik pred drugo svetovno vojno, ko je delal na elektromehanskih čitalnikih luknjanih kartic, razvil ustrezen algoritem za dodajanje paritetnih bitov, ki omogoča popravljanje ene napake v kodni besedi. Danes ga uporabljajo predvsem v RAM pomnilnikih visokozanesljivih računalnikov in v računalnikih v vesolju, kjer kozmični žarki radi "prevračajo" bite v RAMu.

Opis algoritma: p_x = paritetni bit d_x = podatkovni bit x = tem bitom je prirejen p_x

položaj																				
funkcija	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	p_5	d_{12}	d_{13}	d_{14}	d_{15}
p_1 je iz:	o		x		x		x		x		x		x		x		x		x	
p_2 je iz:		o	x			x	x			x	x			x	x			x	x	
p_3 je iz:				o	x	x	x					x	x	x	x					x
p_4 je iz:								o	x	x	x	x	x	x						
p_5 je iz:																o	x	x	x	x

Ponavljjanje

Najenostavnejši način dodajanja redundance je ponavljanje istega sporočila. “Klasične” EME zveze takorekoč temeljijo na neutrudnem ponavljanju klicnih znakov in raportov.

Kaj nam lahko o ponavljanju pove Hammingova teorija? Oglejmo si dve “sosednji” sporočili, in njuni izhodni kodni besedi:

sporočilo 1 = 00011100

koda 1 = 0001110000011100

sporočilo 2 = 00011101

koda 2 = 0001110100011101

Hammingova razdalja je 2 – to nam ravno omogoča zaznavanje napak, ne pa tudi popravljanja.

Z dvakratnim ponavljanjem (100% redundanco) smo torej dosegli komaj kaj več, kot prej pariteta z dodajanjem enega samega bita!

PONAVLJANJE NI UČINKOVIT NAČIN DODAJANJA REDUNDANCE

Reed – Solomon kode

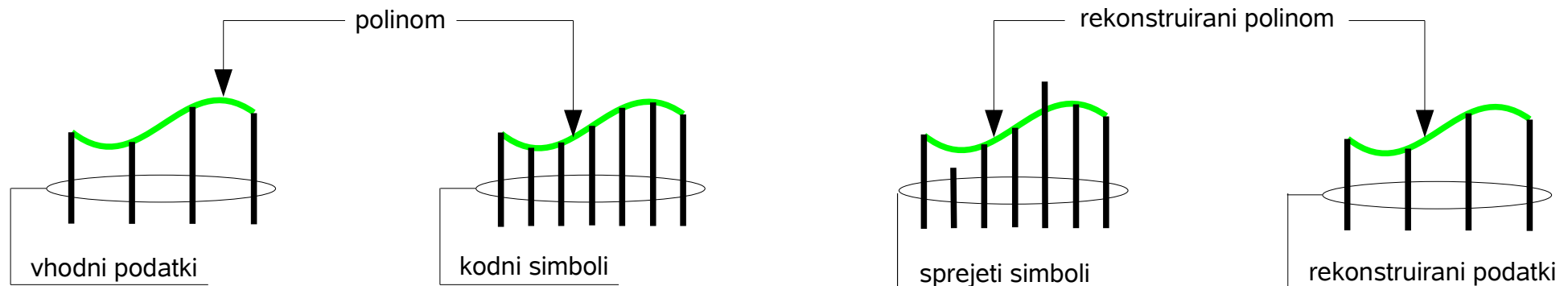
so družina močnih kod za popravljanje napak, ki temeljijo na lastnostih polinomov, in uporabljajo večbitne simbole. Danes se v praksi zelo veliko uporabljajo, n. pr. pri CDjih in trdih diskih. CDji so bili sploh prva širokopotrošna uporaba močnih kod za popravljanje napak.

Princip delovanja:

N točk popolnoma določa polinom stopnje $N-1$. N vhodnih simbolov uporabimo kot točke, skozi katere “napeljemo” polinom. Tega potem vzorčimo v M točkah ($M > N$), ki postanejo simboli izhodne kode.

Sprejemnik ima tako na voljo več točk, kot jih bi nujno potreboval za rekonstrukcijo polinoma, in ga lahko rekonstruira tudi, če je nekaj točk slabih.

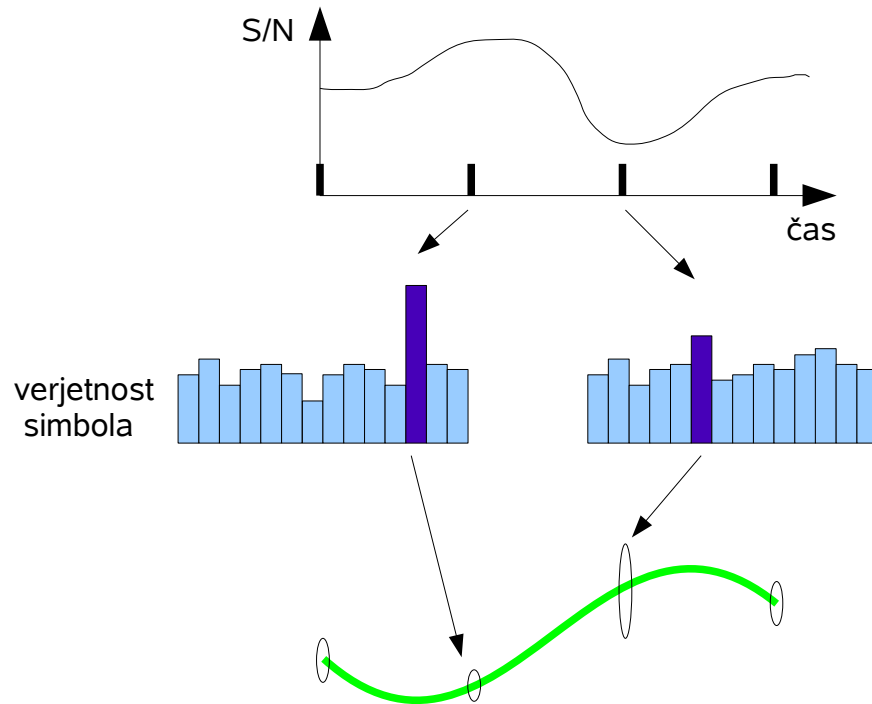
V praksi je najpopularnejša RS koda (255,223) za osembitne simbole, ki 223 vhodnih simbolov zakodira v 255 izhodnih (32 je dodanih) in lahko popravi do 16 napačnih ali do 32 izbranih simbolov. Več napačnih bitov znotraj istega simbola tu šteje kot ena sama napaka, tako da je RS koda še posebno odporna na kratke izbruhe napak.



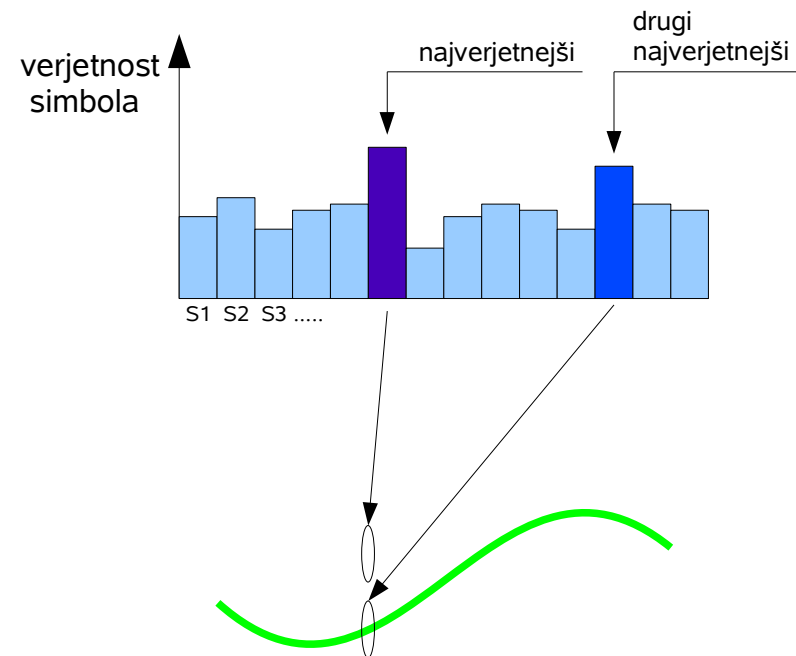
“Mehko” dekodiranje

Včasih nam sprejemnik poleg tega, kateri simbol je bil najverjetneje sprejet, da še nekatere druge informacije, n.pr. s kakšno gotovostjo trdi, da je bil sprejet dani simbol, in pa kateri je drugi (itd...) najverjetneje sprejeti simbol. Dobri Reed-Solomon dekodirji znajo to informacijo izkoristiti.

Kadar je razmerje signal/šum veliko, je simbol zelo verjetno pravilen, zato mu damo večjo utež pri rekonstrukciji polinoma, kot pa simbolu, ki je prišel med QSB.

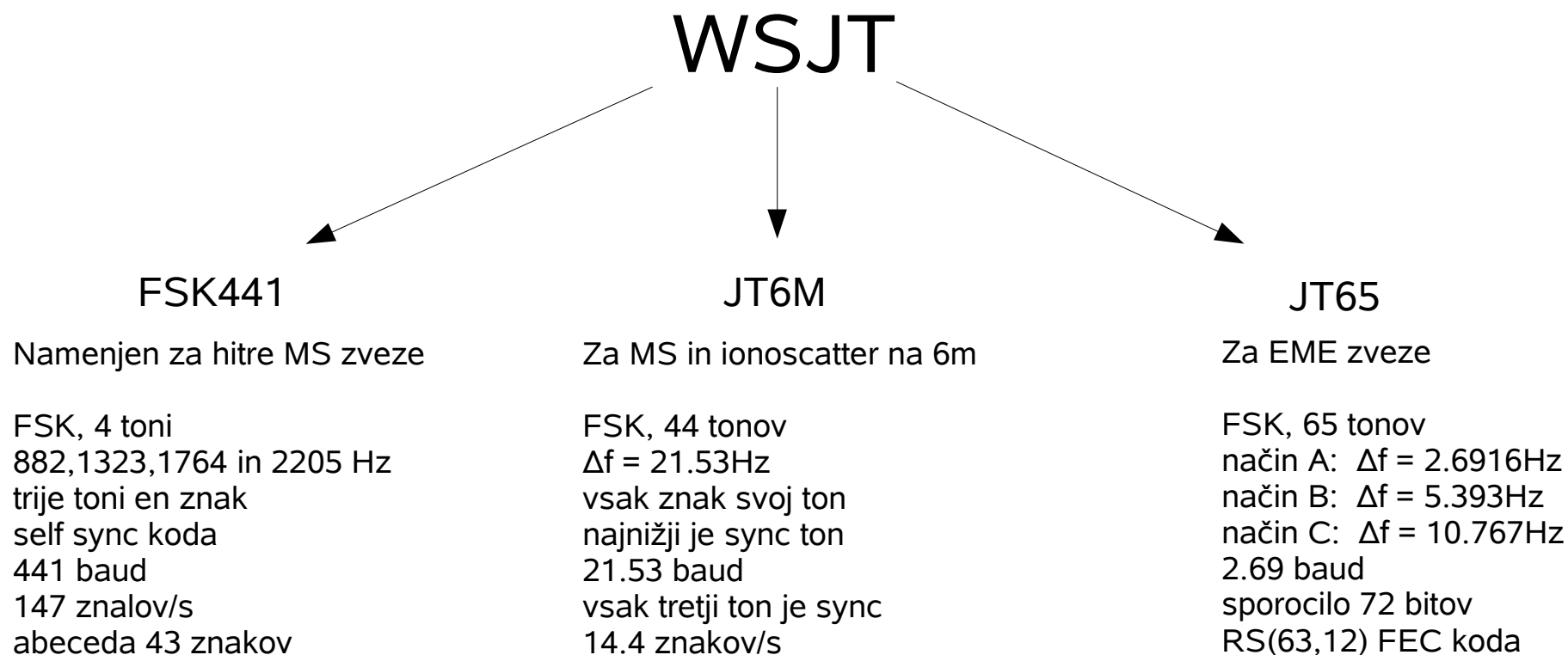


Če eden od simbolov zelo odstopa, lahko dekoder poskusi z naslednjim najverjetnejšim



Programski paket WSJT

Programski paket WSJT je namenjen radioamaterskemu delu s "šibkimi signali". Avtor Joe Taylor K1JT je poskusil prilagoditi načine kodiranja in modulacije raznim tipom radijskih kanalov, ki jih uporabljajo navdušenci za delo s šibkimi signali. Trenutno paket omogoča tri načine dela, prilagojene predvsem MS in EME delu.



JT65: izvorno kodiranje

JT65 uporablja tri različne načine izvornega kodiranja, za tri možne vrste sporočil:

Sporočilo tip 1:

dva klicna znaka + info

info = prvi 4 znaki lokatorja ali signal report 1...30 ali predpona države ali QRZ ali CQnnn ali /P itd

Za vsak znak je rezervirano 28 bitov plus 15 za lokator, skupaj 71 bitov. Ostala informacija je zapakirana v "višek" kod, ker je vseh možnih klicnih znakov manj kot 2^{28} in lokatorskih "kvadratov" manj ko 2^{15} . (poleg tega so izločeni še lokatorji 5 stopinj do severnega tečaja.)

Sporočilo tip 2:

poljuben tekst dolžine 13 znakov

JT65 abeceda ima 43 znakov, torej v 71 bitov lahko zakodiramo $71 / \log_2(43) = 13$ znakov

Sporočilo tip 3:

"shorthand" signali "RO", "RR" in "73"

Za ta tip sporočila ni uporabljeno RS kodiranje, ampak jih pošlje kot sekvenco izmeničnih tonov, ki jo je možno tudi dobro videti na spektrogramu in slišati.

JT65: linijsko kodiranje (FEC)

Sporočilo tipa 3 ne potrebuje dodatnega kodiranja, dvotonsko zaporedje direktno modulira oddajnik.

Sporočilom tipa 1 in 2 dodamo en bit, ki razločuje med tipoma 1 in 2. Tako dobimo sporočilo dolžine 72 bitov, ki ga razdelimo na 12 6-bitnih vhodnih simbolov za RS(63,12) enkoder.

RS enkoder generira izhodno sekvenco 63 6-bitnih simbolov, to je 378 bitov. Redundanca je torej $378 - 72 = 5.25$ krat, najmanjša Hammingova razdalja med izhodnimi spročili pa je 52. Učinkovitost te kode je mnogo boljša, kot pa bi bilo preprosto petkratno ponavljanje.

Po RS kodiranju program simbole permutira, tako da jih po vrsticah zapiše v 7x9 matriko in jih potem prebere po stolpcih. Avtor (K1JT) je pozneje ugotovil, da to nima nobenega vpliva na FEC, vendar je zaradi kompatibilnosti s prejšnjimi verzijami to ohranil.

V naslednjem koraku program 6-bitne simbole pretvori iz binarne v Gray kodo, kar naj bi malo izboljšalo odpornost na frekvenčne nestabilnosti.

JT65: modulacija

Modulacija je FSK, s 65 toni. 64 tonov je prirejenih 64 možnim 6-bitnim simbolom RS enkoderja, eden pa je sinhronizacijski. Hitrost signalizacije je 2.69 baud, tako da celotno sporočilo s sinhronizacijo traja 47.8 sekunde, in omogoča delo v enominutni periodi.

Frekvenčni razmik med toni je 2.69, 5.393 ali 10.767 Hz, odvisno od podnačina A, B ali C, kar omogoča prilagoditev na disperzijske lastnosti kanala, odvisno od frekvenčnega področja. Podnačin A se redko uporablja, saj je tudi na 2m bandu primernejši B, C pa se uporablja za 70 in 23 cm. Za višje bande bodo potrebni še večji razmiki, vendar pa bo za razmike večje od 20 Hz marsikomu problem pasovna širina SSB sprejemnika.

JT65: sinhronizacija

63 simbolom sporočila je dodanih 63 sinhronizacijskih simbolov (vsak drugi simbol) na posebni frekvenci, moduliranih s posebno binarno psevdonaključno sekvenco. Biti sekvence, ki so 1, dajo močno spektralno črto, po kateri JT65 dekode prilagodi frekvenco (Doppler, nestabilnosti postaj...), avtokorelacijske lastnosti sekvence pa omogočajo natančno časovno sinhronizacijo.

Časovno sinhronizacijo dodatno olajša sinhronizacija oddaje na UTC minuto, oddajati začne vedno eno sekundo po polni minuti – to seveda deluje samo, če je ura na računalniku točna...

Sprejem JT65

Na sprejemu program najprej poišče spektralno črto sinhronizacijskega tona. To frekvenco potem izloči s pasovnim sitom, in njeno amplitudno modulacijo korelira z znano psevdonaključno sinhronizacijsko kodo, kar mu da tako bitno kot sinhronizacijo okvirja.

Na osnovi tako določene referenčne frekvence in časa potem pomeri energijo v preostalih frekvenčnih kanalih in časovnih oknih, ki pripadajo posameznim simbolom sporočila.

Dobljene podatke potem preda RS dekodirnemu algoritmu z mehkim odločanjem “Koetter-Vardy”. Ta je sicer patentiran, vendar je K1JT od avtorjev dobil dovoljenje za brezplačno uporabo v neprofitne namene. Poleg “Koetter-Vardy” algoritma pa program WSJT vsebuje tudi navaden RS dekodir s trdim odločanjem, ki je v odprti kodi.

“Deep search”

“Koetter-Vardy” algoritem je sicer kak dB boljši od navadnega “trdega”, vendar pa še vedno ne dosega teoretičnega minimuma. Tega bi dosegli, če bi sprejeti signal korelirali s kodnimi sekvencami vseh možnih sporočil, in izbrali najverjetnejšega. Ker pa so sporočila 72 bitna, to praktično ni mogoče.

K1JT je zato dodal način dela, ki uporabi tabelo znanih pozivnih znakov, da zmanjša iskalni prostor, in tako lahko uporabi “idealno” metodo s koreliranjem vseh možnih sporočil, ter pridobi cca 4 dB. Seveda to precej zmanjša količino informacije, prenešene preko zveze – pri tabeli velikost nekaj tisoč pozivnih znakov, in če bi program vedno izbral enega od znakov, bi prenesli nekaj več kot deset bitov. Ker program ne da vedno rezultata, oz. ga opremi z oceno zanesljivosti, je prenesene informacije v resnici več.

Ta metoda je med HAMi malo kontroverzna. V zagovor lahko rečemo, da ima lahko, še posebej na “manj naseljenih” bandih, tudi klasični Morse operater “v glavi” znake vseh aktivnih postaj in (morda nezavedno) uporablja “bio” verzijo deep search algoritma...