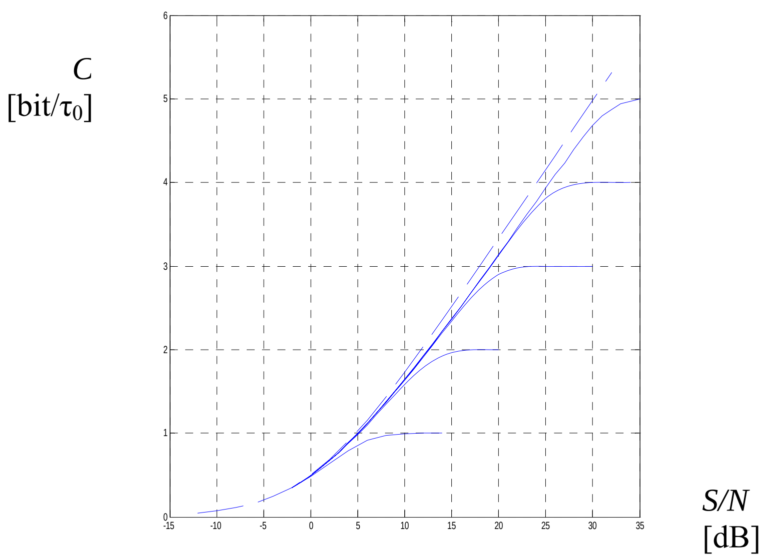
A kódolt moduláció

Több évtizeden keresztül a moduláció és a kódolás egymástól függetlenül művelt területe volt a hírközléselméletnek. Valamikor az 1970-es évek elején kezdtek azzal a kérdéssel foglalkozni, hogy miként lehetne a moduláció és a kódolás közös tervezésével elérni, hogy az információt képviselő jelsorozatok minél távolabbiak legyenek, és így zaj esetén a legjobb megkülönböztethetőséget biztosítsák.

Valójában arról van szó, hogy ebben az időben már kezdett világossá válni, hogy a teljesítménykorlátos, de csaknem szabad sávszélességű csatornákon jó eredményt mutató kódok szinte hatástalanok, ha korlátozott sávszélességű csatornákon akarjuk azokat alkalmazni. Mint már korábban elmondtuk, ha sávszélesség-korlátos a csatorna, akkor csak olyan módon adhatunk redundanciát az átviendő információhoz, hogy megnöveljük a használandó szimbólumkészletet. A problémát pedig alapvetően az jelenti, hogy bináris szimbólumkészlet és bináris jelkészlet (moduláció) esetén a Hamming távolság megfelelő mérték a kódtervezésre, mert megfelel a jelek euklédeszi távolságának, de nem bináris esetekben már nem ez a helyzet, és így a kódtervezésben új módszereket kellett keresni.

Egy igen érdekes, és alapvető jelentőségű felismerés a szükséges szimbólumkészlet bővítésének terén a következő. Ha megrajzoljuk a csatornakapacitás shannoni határát, valamint a digitális bemenetű és analóg kimenetű csatornák kapacitását additív gaussi fehér zaj esetén, akkor a következő ábrát kapjuk:



A szaggatottan rajzolt görbe azt mutatja, hogy elvileg hány bit információ továbbítható szimbólumidőnként egy additív Gauss zajú csatornán a jel-zaj viszony függvényében. A folytonos görbék a digitális bemenetű, analóg kimenetű csatornák kapacitását mutatják, a szimbólumkészlet 2, 4, 8, 16 és 32-es értékei mellett. Ily módon ezek a görbék 1, 2, 3, 4 és 5 bit/szimbólumidő értékeknél tetőznek, mindig az elvi határ alatt futnak, és a szimbólumkészlet méretének növekedésekor felismerhetően egy – az elvi határt jelentő görbével párhuzamos – görbébe olvadnak. (Erről a kb 1,53 dB értékű „elmaradásról” itt nem fogunk beszélni, csak megjegyezzük, hogy ennek leküzdésére is találtak megoldást.)

A diagramm egyik legfontosabb mondanivalója a következő: amennyiben egy adott szintű digitális csatornán a maximálisan lehetséges sebességet csökkentett jel-zaj viszony mellett akarjuk elérni, akkor a korlátozott sávszélesség miatt ki kell bővíteni a szimbólumkészletet, de ELÉG A KÉTSZERESRE TÖRTÉNŐ BŐVÍTÉS! Ugyanis, ha bármelyik folytonos görbe vízszintes szakaszától balra azonos magasságban elmozdulunk, akkor a többszintű görbéknek már a szinte egybefolyó részéhez érkezünk, tehát a kétszeresre bővített jelkészlettel már szinte az összes lehetséges nyereséget elérhetjük, így nem kell tovább bővíteni. „Csupán” az a feladat marad, hogy keressük meg azt a kódot, amely a következő jellemzőkkel rendelkezik:

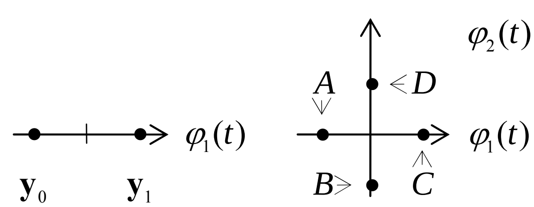
- Nem blokk kód, mert tudjuk, hogy az információt hordozó szimbólumokban mesterségesen képezett határolás veszteséget eredményez, tehát legyen trellis kód, de azon belül is elsősorban konvolúciós kód, amely – mint tudjuk – lineáris.
- Ha m bináris szimbólumot akarunk kódolatlan továbbítással időegységenként elküldeni, akkor 2^{m+1} elemű jelkészlet kell a kódolthoz sorozathoz, tehát
- A használandó kód $(m+1,m)$ paraméterű konvolúciós kód legyen.

A fenti kód megkeresésével párhuzamosan alakítsunk ki egy olyan leképezési szabályt, amely a keletkező kódsorozatot oly módon rendeli hozzá a választott jelkészlet egyes tagjaihoz, hogy a szabad távolságra lévő sorozatoknak megfelelő jelek euklédeszi távolsága a lehető legnagyobb legyen a jeltérben.

Az alábbiakban egy nagyon egyszerű példa keretében megmutatjuk, hogy miként lehet a feladatot megoldani, utalva arra, ahogyan Ungerboeck a róla elnevezett kódolást kidolgozta.

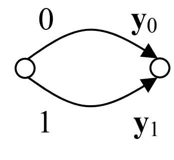
Legyen a példánk egy önmagában nem érdekes eset, amely azonban minden fontos lépésre rávilágít, miközben követhető bonyolultságú ábrákat és megfontolásokat kell megérteni! Ennek szellemében válasszuk azt az esetet, amikor egy kódolatlan BPSK átvitelt akarunk kódolással QPSK-vá alakítani, vagy precízebben fogalmazva, a kódolt sorozatott QPSK modulációval akarjuk továbbítani! Ezáltal m = 1-et, és 2^{m+1} = 4-et, választottuk, illetve egy (2,1)-es konvolúciós kódot keresünk.

A választott jelkészletek a jeltérben a mellékelt ábrán láthatók (csak szimbolikusan ábrázoljuk):



Az ábrán a kételemű jelkészletnél a modulációval foglalkozó fejezetben alkalmazott jelölést használtuk a két jelvektorra, a négyelemű esetben azonban annyit rövidítettünk, hogy az abc első négy betűjével jelöltük a vektorokat. (Csak emlékeztetőül mondjuk, hogy a vektorok koszinusz, illetve szinusz függvényeket képviselnek a τ_0 hosszúságú időrésben.)

Mivel a konvolúciós kódok leírására a trellis az egyik legszemléletesebb eszköz, hozzuk valamilyen kapcsolatba a jeltér elemeit a trellis-szel! Kezdjük azzal, hogy a kódolatlan sorozatot hordozó bináris jelkészletet is leírhatjuk egy trellis-szel, amely azonban a korábban megismertekhez képest elfajultnak tekinthető. A kódolatlan, bináris esetre egyállapotú trellis-t használhatunk, amelyik lényegében azt rögzíti, hogy emlékezet nélkül, az aktuális szimbólum hatására két lehetséges, „párhuzamos” ág valamelyikén léphetünk a következő, ugyanolyan csomópontra.



Az ágakat 0 és 1 bináris szimbólumokkal, illetve $\mathbf{y}_0$ és $\mathbf{y}_1$ jel-vektorokkal jelöltük, mely utóbbiak az $y_t^{(1)} = A \cdot \sin(2\pi/\tau_0 \cdot t)$, és $y_t^{(0)} = -A \cdot \sin(2\pi/\tau_0 \cdot t)$; $t \in [0, \tau_0)$ időfüggvényeket képviselik, és ezáltal válik megkülönböztethetővé a két ág.

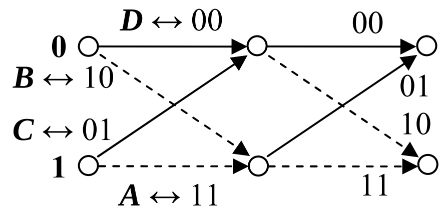
Térjünk rá a négyelemű jelkészletre! Kétszerannyi jelünk van, de a legközelebbiek közti távolság √2 –ed része a bináris esetben lévőnek, feltéve, hogy azonos a jelek átlagteljesítménye, ami a reális összehasonlítás alapja. A kétszerannyi jel azt teszi lehetővé, hogy mindenegyes kódolatlan szimbólumot négy lehetséges jel valamelyikével képviseltessük a csatornán, nyilván attól függően, hogy milyen szimbólum, vagy szimbólumok előzték meg az éppen aktuálisat. Azaz, mint már erről a kódolás bevezetőjében is elmélkedtünk, kapcsolatokat kell kialakítani a szimbólumsorozatban, ami másképpen úgy fogalmazható, hogy emlékezetet kell vinni a sorozatba.

Az első lépésben legyen az emlékezet egyetlen lépésnyi! Ekkor a kódolt sorozatot, illetve – ami most még érdekesebb – a használt jeleket egy kétállapotú trellis által írhatjuk le. Viszont annak érdekében, hogy a jelek legjobb választását biztosítsuk, végezzünk közöttük egy célszerű csoportosítást! Amennyiben a négyelemű jelkészletben az abc betűivel jelölt jeleket az abc sorrendjében egymásmellé helyezzük – *A, B, C, D* – akkor a szomszédosak közelebb lesznek a jeltérben, mint a távolabbiak, azzal a pontosítással, hogy a sor valójában gyűrűszerűen záródik, tehát a két szélső is szomszédnak számít. Bontsuk szét a négyelemű készletet két csoportra, de oly módon, hogy egy-egy csoportba mindig az egymástól távolabbiak kerüljenek, majd az így kapott két csoportot ismét bontsuk ketté! A keletkező csoportokat – az utolsóknak csak egyetlen tagja lesz – lássuk el bináris címkékkel, amint az alábbi ábrán az eddigieket szemléltettük:

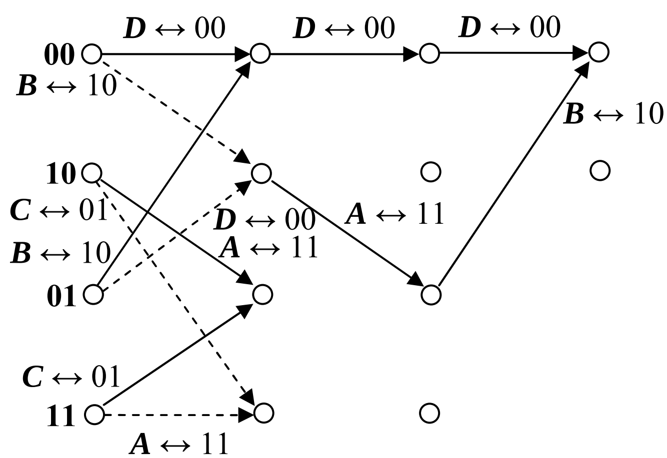
	A B C D			
	A C		***B D***	
d_0	1		0	
	A	***C***	***B***	***D***
d_1	1	0	1	0

A d_0 és d_1 a nulladik- és az első szintű csoportosítást jelöli, egyben utalva arra, hogy a megjelölt szinten lévő csoportok között a jeltérbeli távolság nulladik- és első szintű, amelyek a választott csoportosítás szerint növekvő értékeket mutatnak. Lényegében az egész csoportosításnak ez a növekvő távolság a célja, azaz a szereplők szisztematikus „távolítása” egymástól.

Ezután térjünk vissza a kétállapotú trellis kialakításához! Jelöljük 0-val és 1-el a két állapotot, valamint használjuk az egyes ágak jelölésére az ott alkalmazott jelnek a fenti táblázatban szereplő bináris címkéjét! Milyen módon rendeljük a jeleket az egyes ágakhoz? Mivel tudjuk, hogy a trellis-ben kialakuló hurkok az összetéveszthető utakat képviselik, ezért igyekszünk a hurkokat alkotó utak ágain minél eltérőbb jeleket használni. Ennek megfelelően a csomóponti elágazásokra és összefutásokra keresünk minél jobban különböző jeleket. A mellékelt ábrán bemutatunk egy lehetséges kiosztást, ahol a **0** állapotból 0 szimbólum hatására (folytonos vonal) kiinduló ághoz a ***D*** jelet



párosítottuk, amit a fenti táblázatban 00 címkével jelöltünk. A **0** állapotból 1-es szimbólum hatására kiinduló ághoz a ***B*** jelet rendeltük, amely azonos csoportban van az elsőként választott ***D*** jellel, amit címkéjük végének egyezése is mutat, és így távolságuk nagy. Ezután nem maradt túl sok választási lehetőségünk, lévén a trellis rendkívül egyszerű, és a másik két jelet az **1**-es állapotból kiinduló ágakhoz rendeltük hozzá.
A trellis-t még egy lépéssel folytattuk, ahol kicsit rövidítettük az ágak jelöléseit. Az ábrából láthatjuk, hogy várakozásunknak megfelelően a kétállapotú trellis esetén a legrövidebb hurok hossza két ágból tevődik össze. Az ágak jelölésére használt bináris címkék választott kiosztása azt eredményezi, hogy a fenti trellis-szel leírt (2,1) paraméterű kód lineáris lesz, és ezért a teljesítőképesség megvizsgáláshoz elegendő a csupa nullától eltérő út jellemzőinek a tekintetbe vétele.
A csupa nulla kódsorozatot két egymást követő ***D*** jellel továbbítjuk, míg a hurok másik útján egy ***B*** jel és egy ***C*** jel található. Ez azt eredményezi, hogy a zajos analóg csatornán a jeltévesztés valószínűségét meghatározó jeltávolságok a hurokban az első ágon d_1 a második ágon viszont csak d_0 értékűek lesznek, amint ezt a táblázatból kiolvashatjuk. Ennek alapján végezhetünk egy összehasonlítást a kódolatlan, bináris esettel. A bináris átvitel szabad távolsága (a két elemi jel közötti távolság a jeltérben) d_1 értékű, míg a kódolt esetben $\sqrt{d_1^2 + d_0^2} = d_1 \cdot \sqrt{1+\frac{1}{2}} = d_1 \cdot 1{,}2247$. Nem túl nagy javulás, de szerencsére könnyen túl lehet szárnyalni. Használjunk négyállapotú trellis-t!
A trellis megszerkesztése, és a használandó jelek kiosztása még könnyen elvégezhető, amit az alábbi ábrán bemutatunk, de az előzőnél kicsit kevesebb magyarázattal, mert a követendő módszer ugyanaz, csak a „választék” lett bővebb:



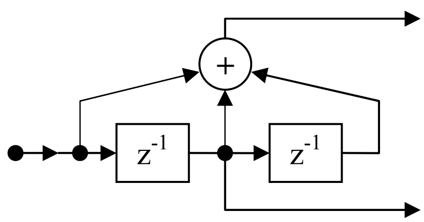
Kezdve a jelek kiosztását a **00** állapotnál, akkor megismételhetjük az előzőt, majd ugyancsak ismétlésszerűen leugrunk az **11** állapothoz. Innen kezdve azonban több lehetőségünk van, mint előbb volt. Most úgy tudunk jeleket kiosztani, hogy az **10** és a **01** állapotba összefutó ágakhoz rendelt jeleknek is nagy legyen a távolsága: ha az **10** –hoz fentről a ***B*** jel érkezik, akkor alulról a ***D*** jöhet, és ha a **01** állapotba alulról a ***C*** jön, akkor felülről jöhet az ***A***.
Az így kialakított trellis részletét még két lépésen keresztül folytattuk, ahol megrajzoltunk egy hurkot, amelynek felső útja csupa nulla, az alsó pedig 10-11-10. Illetve a felső úton mindhárom lépésben ***D*** jelet használunk, míg alul a ***B – A – B*** jeleket. Könnyen ellenőrizhetjük, hogy a két úton használt jelek távolsága a következő:

$$\sqrt{d_1^2 + d_0^2 + d_1^2} = d_1 \cdot \sqrt{1 + \tfrac{1}{2} + 1} = d_1 \cdot 1{,}5811 \ ,$$

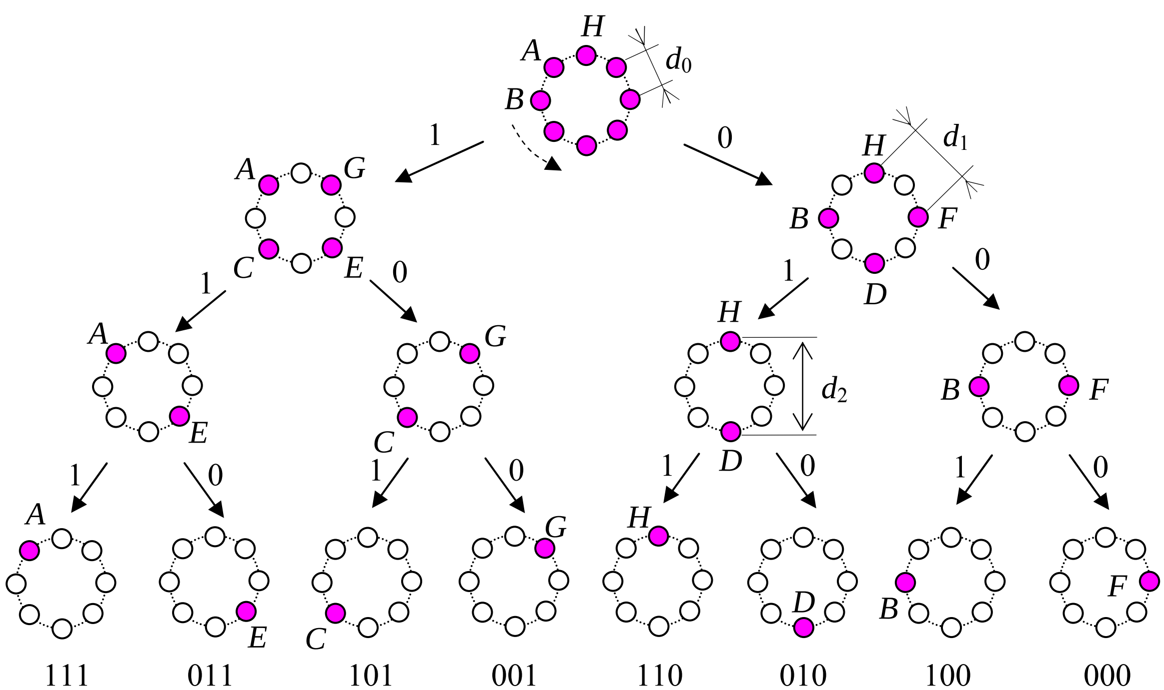
ami már nem jelentéktelen kódolási nyereséget eredményez.

Tovább növelve a kódolt sorozat készítésénél az emlékezetet, még további javulást érhetünk el, bár egyre nehezebb átlátni a sokállapotú trellis-szen használandó jelek kiosztását. A 8 állapotú trellis még megoldható „papíron ceruzával”, de a 16 állapotúhoz már számítógépes segítség kell. Bár ez a kérdés is érdekes gyakorlatilag, ezzel mégsem foglalkozunk, hanem azt akarjuk megválaszolni, hogy miként kell a kódolt sorozatot előállítani.

Mivel kikötöttük a lineáris kódot, ezért leszögezhetjük, hogy konvolúciós kódolóval előállítható a kód. A konvolúciós kódoló leírására több módszert megismertünk, most a leggyakorlatiasabb leírást, a kódot előállító shift-regisztert adjuk meg. A fenti négyállapotú trellis által leírt kódot az alábbi shift-regiszteres kódoló állítja elő:

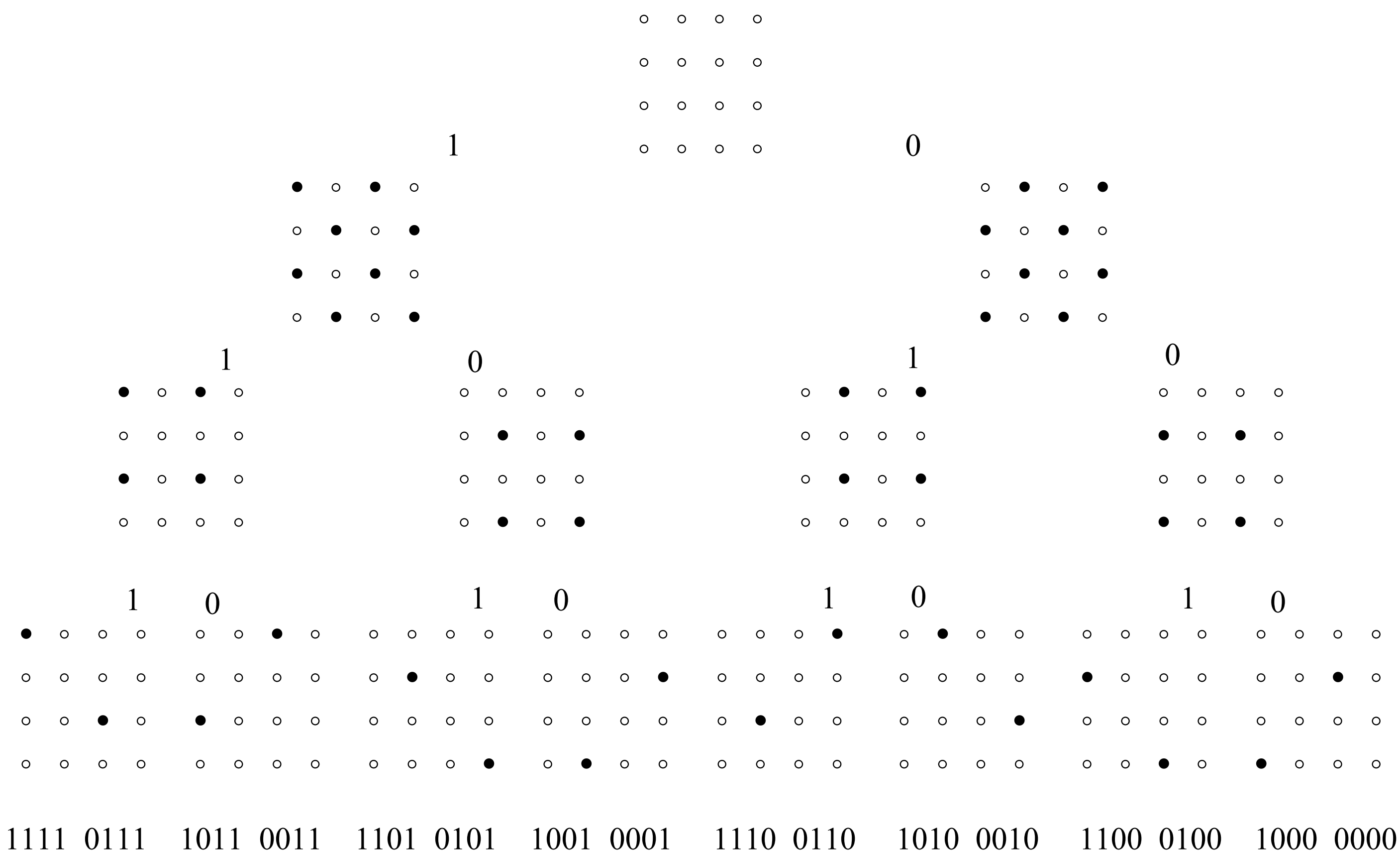


Befejezésül megmutatjuk, hogy a fenti részhalmazokra osztással történő leképezést miként lehet praktikusan is hasznos kódok előállítására fordítani. Egy 8PSK és egy 16QAM átvitel a leggyakrabban idézett két példa. Kezdjük a 8PSK-val:



Az ábra azt mutatja, hogy miként kell a 8 jelet szétosztani, a 3 elemű kódsorozatokhoz rendelni, hogy a legnagyobb távolságok legyenek elérhetők. Érdekes feladat a trellis-ek megszerkesztése, amit itt nem mutatunk meg.

A 16QAM-re vonatkozóan a „mapping by set partitioning” a következő ábrával szemléltethető:



Az első osztásnál √2-szörös lett a szomszédok távolsága, a másodiknál az eredeti duplája, a harmadiknál az eredeti 2▪√2-szöröse. Az ábrából az utolsó párok szétválasztását elhagytuk, csak a címkék alapján képződő kódokat írtuk le. Természetesen most is kell keresni egy trellis-t, amely meghatároz egy konvolúciós kódot, és a kódolt sorozathoz kell hozzárendelni a fenti jeleket.
A továbbiakban különböző példákon keresztül bemutattuk a kódolt moduláció lehetőségeit, beleértve a kettőnél több dimenziós jelterek kialakítását, a jelkészlet kedvező elhelyezését is.