

Intelligens, mint fogalom sokat változott az évek folyamán, az elektromos eszközökben.

Determinisztikus automata

- ugyanazokra a bemenetekre, ugyanazokat a válaszokat adja
- algoritmusok sorozata egy feladatra
- szoftver is lehet robot (pl. Google: hálózat figyel minket és próbál beazonosítani)

Robot vs automata

	Robot	Automata
Programozhatóság	+	+
Érzékelők	+	+
Memória	+	-(+)
Adaptációs képesség	+	-
Tanulási képességek	+	-

Adaptációs képesség: először hibásan, de tanulva hibáiból utána már helyesen oldja meg a feladatot.

Robotok felhasználása:

- Űrkutatás: űrszondák, leszálló egységek, mobil bolygókutató egységek
- Ipari robotok: összeszerelő robotok (hegesztő, szerelő, festő), megmunkáló robotok, karbantartó robotok
- Katasztrófaelhárító robotok: tűzoltó, felderítő (vegyi, sugár)
- Katonai robotok: akna felderítő
- Egyéb

B.E.A.M. robot: Biology Electronics Aesthetics Mechanics

napelem tölti a kondenzátort → szimmetrikus áramkör figyeli a feszültséget → szint elérve → motor mozgat → kapcsolók (bajusz) érzékeli az ütközést → cél: napfény

Szenzorok

Környezeti információk érzékelése és továbbítása a rendszer felé.

Infrareflexiós szenzor (fénysorompó)

- visszavert fényt érzékeli
- egyszerű
- könnyű számítógépre kötni (1bit – van/nincs akadály)
- csak egy pontban érzékel → több irányba kell érzékelőt elhelyezni

Ultrahangos/lézeres távolságmérő

- hasonló az infrához
- akadály távolságát tudja megadni

Kamera

- nagy mennyiségű információ → erőforrásigényes
- képfeldolgozás nagy számítási igényű
- részletes információ az akadályról
- sok energiát fogyaszt
- energia vs felbontás problémája

360 kamera

- nagyobb lefedett terület
- torzulás → információ veszteség

Szenzorok típusai/csoportosítása

Passzív szenzorok

Kívülről nem érzékelhetők → nem sugároz

- hőmérséklet
- nyomás
- fény (kamera)
- hang (mikrofon)

Aktív szenzorok

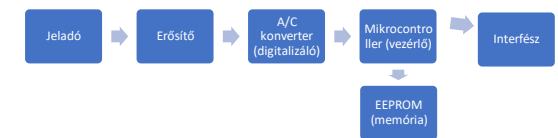
Mérőjelet bocsátanak ki, ennek visszaverődését más is érzékelheti.

- radarok
- fényérzékelő kamerák
- ultrahangos/lézeres távolságmérők

Strukturált fény kibocsátás

- lézercsík: a csík deformáltságából következtetünk a távolságra
- pontok: pl. xbox kinect

Intelligens szenzorok



Tulajdonságok – ZH

- önkalibrációs – akár több millió paraméter lehet
- öndiagnosztika – növeli a biztonságot, megbízhatóságot
- dinamikus érzékenység
- programozhatóság
- távvezérelhetőség

Adaptív szenzorok

Alkalmazkodik a változásokhoz

Fix értékek átlagát nézzük?

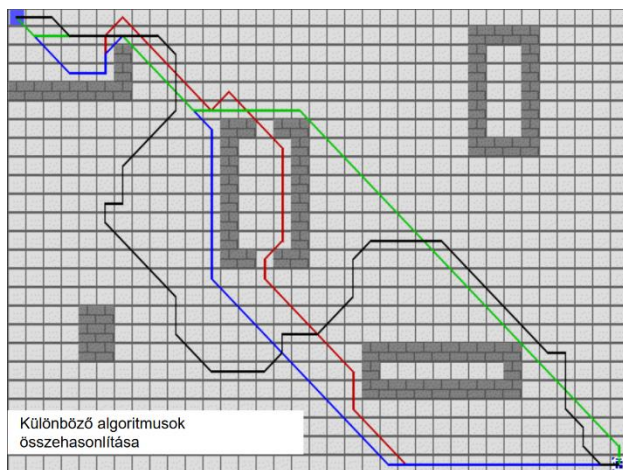
Lehető leg szélesebb körülmények között működjön

Navigáció

Robot egy kijelölt célpozícióra jusson (akár robotkarról is beszélhetünk)

Figyelembeveendő szempontok

- robot mozgási lehetőségei, képességei, mechanikai tulajdonságai
- mozgáshoz szükséges energia
- idő
- Terepviszonyok
 - Ismert terep (pl. autós navigáció)
 - Ismeretlen terep - látom, de nem ismerem



piros: szabályalapú algoritmus

kék: neurális-elvű algoritmus

zöld: hullám-továbbterjesztéses algoritmus

fekete: GVD-elvű, gráfbejáráson alapuló algoritmus

Szabályalapú navigáció/pályatervezés

Szabálytábla

- Szabályok a robot képességeihez szabva
- Minden egyes helyzetben a szabályrendszer határozza meg, hogy mit tud a robot csinálni
Itt: 1 helyzet (kocka) → 8 féle cselekvés (előre, hátra, jobbra, balra, jobb-felül, stb.)

Nincsen garancia, hogy eljut a célig, de sok esetben működik

probléma: „beragadás” → memória kell, hogy kijusson egy „mélyebb” akadályból, mert mindig a cél felé akar lépni

Öntanuló (tapasztalat szerzésen alapuló) navigáció

Szabályalapún alapszik, de a szabálytábla kitöltését a gépre bízuk.

Algoritmus:

Amíg el nem éri a célt:

1. Cél irányának meghatározása
2. Legjobb jóságú lépés kiválasztása
legmagasabb érték, vagy a sorrendben a legelső felé megy
3. Új lépés kiszámítása (lépés)
4. Távolság ellenőrzése (cselekvések értékelése)
 - ha csökken a céltól a távolság a cselekvés felértékelődik (+1)
 - ha nő, akkor leértékelődik (-1)
 - ha akadálnak ütközik, nagyon leértékelődik (-3)

Kis kóborlás (tréning) után egyre jobb válaszokat tud adni a rendszer

Előnyök

- Ismert és ismeretlen terepen is működik

Hátrányok

- Nem garantált a cél elérése
- Nem tudni, hogy elérhető-e a cél

Neurális elvű navigáció

Szabály alapú ez is

pl. 20 esetből tanul a hálózat, a többit ebből generálja

Soha nem látott mintára is tud választ adni

Nem ad egzakt eredményt → valószínűsíthető (pl. 98% eséllyel jut célba → egzakt megoldásokhoz nem alkalmazható (nincsen garancia a célba jutásra)

Hullám-továbbterjesztés tervezés

Hullám algoritmus

Pontból indul a hullám

Hullám nem gyengül, nem verődik vissza, mint a fizikában

1. Csupa nulla a pont köré
2. Utána 1, 2, 3, 4, ... körbe
3. Falak elnyelik → nem megy tovább
4. Amíg fel nem töltjük a terepet

Célből monoton csökkenve lépkedünk vissza, ha azonosak az értékek, akkor egy szabály dönti el, hogy merre (pl. mindig jobbra)

Előnyök

- Egzakt megoldás → már az elején tudja, hogy eljut e, s ha nem el se indul
- Ha létezik, a legrövidebb utat adja!

Hátrányok

- Csak ismert terepen működik

Hullám ismeretlen terepen

1. Szenzorok
2. Lépés a szenzorok által detektált akadályokat figyelembe véve
3. Amit látott beépíti a térképébe (emlékezőképesség)

GVD-elvű, gráfbejáráson alapuló tervezés

Legbiztonságosabb utat adja → minden akadálytól a legtávolabb megy

Gráf létrehozás → valamilyen tulajdonság alapján beveszek, vagy kizárok éleket

„Edény” faláról (terep széléről) indul a hullám, majd a lehető legmagasabb pontokon haladunk

Neurális hálózatok

biológiai inspirációjú információ feldolgozás, ahol a modell az idegrendszer

Jellemzői

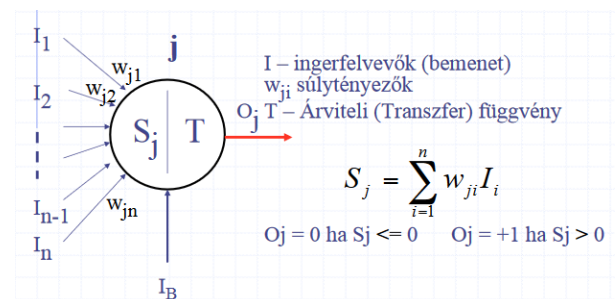
- neuronokból (egyszerű processzorokból) épül fel
- hálózatot alkotnak, ahol az összeköttetések változtatható súlytényezőjűek
- a tárolt információk a hálózatban elosztottan a súlytényezők segítségével ábrázolhatók
- nem programozzuk, hanem tanítjuk
- neurális hálózatok hibatűrők

Neuronokat az **idegsejtekről** mintázzuk

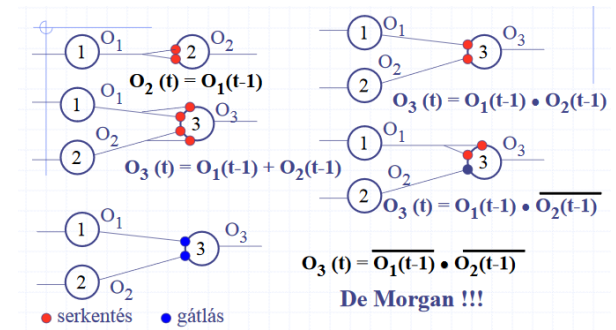
- működése mindent, vagy semmit jellegű
- ingerületbe hozáshoz legalább két bemenetet ingerelni kell
- jelentős késleltetés csak a szinapszisoknál jön létre
- bármilyen gátló szinapszis megakadályozhatja az idegsejt ingerületbe kerülését
- összeköttetési hálózat időben nem változik

Mesterséges neuronok hogyan épülnek fel?

A McCulloch és Pitts formális neutron (1943)



Logikai műveletek a McCulloch-Pitts neuronokkal



Perceptron (Frank Rosenblatt - 1957)

Első mesterséges neurális hálózat: vetített nyomtatott betűk felismerése tanítás alapján

Jellemzők

- 20x20 fotóérzékelő
- McCulloch-Pitts neuronok
- Előre-csatolt egyrétegű hálózat
- Csak lineárisan elválasztható csoportokat tud osztályozni

Felügyeletes tanítás algoritmus

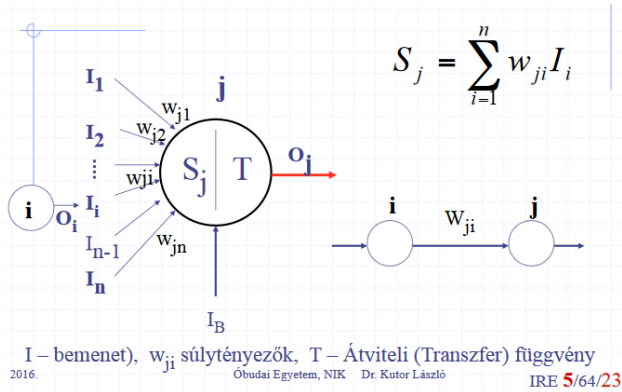
Addig változtatjuk a súlytényezőket, amíg a bemenő mintákra a hálózat a megfelelő-, előre kiszámított válaszokat nem adja

- Kezdeti súlytényezők beállítása
- Tanítóminta bemeneti értékei alapján a hálózat kimeneti értékeinek kiszámítása
- Tanítóminta célértékeinek összehasonlítása a hálózat célértékével
- Szükség esetén a súlytényezők módosítása
- Tanítás addig, amíg az összes tanítómintával az előre meghatározott hibahatárnál kisebb hibával elő nem állítjuk a meghatározott kimeneti értéket

Neurális hálózat legfontosabb tényezői

- neuronok (neuron/node/unit/cell)
- topológia (összeköttetési séma)
súlymátrix
- tanító szabályokat tartalmazó algoritmus

Alap neuron felépítése

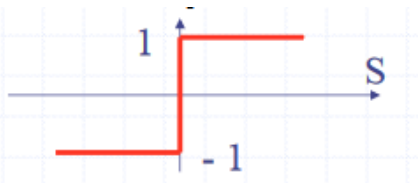


Leggyakoribb átviteli függvények

Ugrásfüggvény

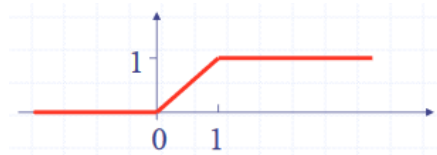


$$O_j = 0, \text{ vagy } -1, \text{ ha } S \leq 0$$



$$O_j = 1, \text{ ha } S > 0$$

Korlátozott lineáris függvény



$$O_j = 0, \text{ ha } S \leq 0$$

$$O_j = S, \text{ ha } 0 \leq S < 1$$

$$O_j = 1, \text{ ha } S \geq 1$$

Szimoid függvény



$$O_j = \frac{1}{1 + e^{-S}}$$

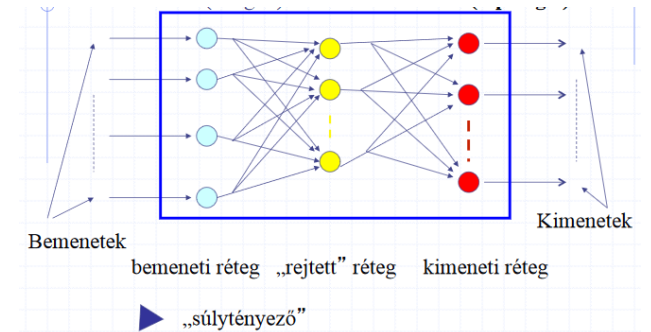


$$O_j = 1 - \frac{1}{1 + S}, \text{ ha } S \geq 0$$

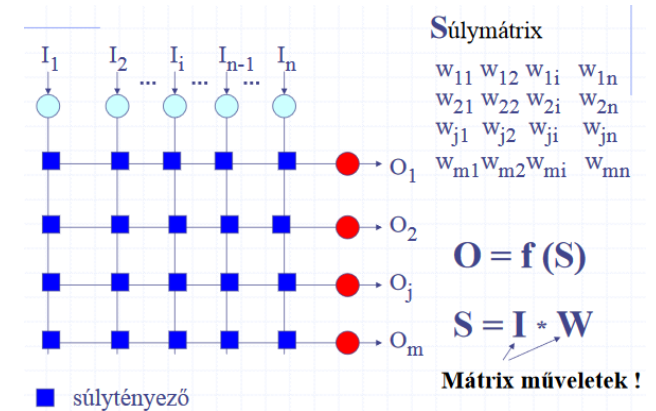
$$O_j = -1 + \frac{1}{1 - S}, \text{ ha } S < 0$$

Neurális hálózat összeköttetések (topológiák)

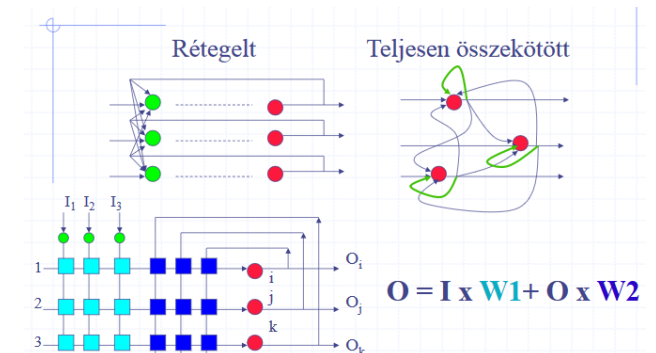
Előrecsatolt (rétegelt) hálózat



alternatív ábrázolás:



Visszacsatolt



IRE

Mikor használjuk?

Meddig tanítunk egy neurális hálót?

Mi a rétegszám?

Hibatűrés

Inicializálás

Genetikus algoritmus



Biológiai inspiráció alapján: egy egyedhalmazból, az újonnan létrejött populáció (jól szabályozott rendszer esetén) jobb „minőségű” egyedek jönnek létre

Mikor használjuk?

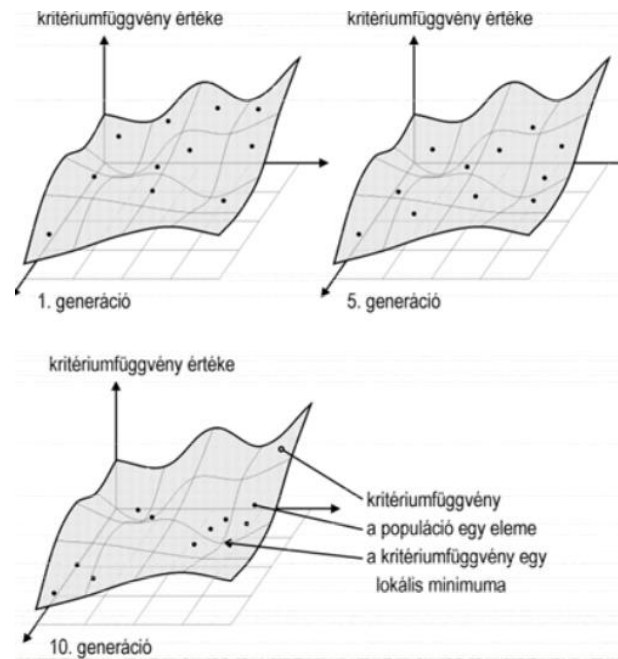
- Ismert az elérni kívánt eredmény - **kritériumfüggvény**
- Nem ismert, de az egyes részeredmények egymáshoz való viszonyából megállapítható
- Nem ismert az eredményhez vezető optimális út
- Több megoldási mód is ismert, de nincs olyan, ami optimális eredményt ad

A genetikus/evolúciós algoritmusokat *sztochasztikus szélsőérték keresésnek* nevezzük

Heurisztikus algoritmus

Kiindulás

Létezik egy probléma tér, amit egy felület reprezentál:



Probléma lehet, hogy olyan lokális minimumba, vagy maximumba kerülnek a kezdeti egyedek, amelyekből nem tudnak továbbfejlődni, „beragadnak”. (gradiens módszer esetén)

Zajt (*véletlenszám generátort*) használunk a kezdeti egyedek létrehozására, hogy:

- a „beragadást” elkerüljük
- beletaláljunk abba a területbe, ahonnan a legsikeresebb egyedek kifejlődhetnek

Eljárás sebességét befolyásolják

1.) Kezdeti populációszám

Nagy populáció elvileg gyorsabb eredményt ad, de nagyobb a számítási igénye → meg kell keresni az *optimális populáció számot*

A populáció mértéke nincsen befolyással az eredmény jóságára

2.) Mutáció

Biztosítja a rendszer folyamatos működését (jóság növekedését)

- Mutáció nélkül a rendszer gyorsan leáll
- Lassú mutáció esetén lassú eredmény
- Gyors mutáció esetén „szétesik” a rendszer

konklúzió: *optimális mutációs sebesség* kell

Algoritmus

- 1.) Induló egyedek (individuumok) számának meghatározása
- 2.) Szelekciós eljárás kiválasztása
- 3.) Populáció kezdőértékeinek megadása (generálása)
- 4.) Individuumok értékelése (fitness számolása)

Generációs ciklus:

- 1.) szülők kiválasztása → utódok generálása
- 2.) utódok értékelése (fitness számolás)
- 3.) új populáció kiválasztása (változó stratégiákkal)

megállási feltétel: általában adott számú ismétlés (generációs szám), vagy adott jóság elérése után

Szelekciós eljárások**Fitness arányos szelekció (Roland-kerekes szelekció?)**

Egy rulett kerekét felosztunk úgy szeletekre, hogy minden egyednek jut egy szelet, az egyed fitnessének megfelelő arányban.

Forog → egy egyed

Probléma: ha azonos fitnessűek az egyedek, akkor minden egyednek közel azonos esélye van

Rang szerinti szelekció

Hasonlóan rulett-kerek elven, a szeletek a sorba rendezett populációban betöltött hely szerint kerülnek megállapításra. A leggyengébb egyed egy egység, a legjobb n egységnyit. (n a populációszám)

Verseny szelekció

Sorrend szerint rendezünk.

Két egyed véletlenszerűen a populációból.

Véletlen számot r-t generálunk 0 és 1 között, ha $r < k$, ahol k egy paraméter (pl. 0.75), akkor a jobb oldali elem kerül kiválasztásra, különben a bal.

Legjobb szelekció

Mindig a populáció legjobb eleme kerül kiválasztásra.

Véletlen szelekció

Véletlenszerűen egy egyedet választunk ki.

Interaktív szelekció

Olyan helyzetekben, ahol a felhasználó tud választani az egyedek közül, mert a fitness függvényt nem tudjuk megalkotni.

Sokszor grafikai elemeknél használjuk

Tervezéshez és alakfelismeréshez

Sigma scaling

Rulettkerék alapú

Szelet meghatározása:

- egyed fitness értéke
- populáció átlaga
- populáció szórása

$$f^*(i) = \begin{cases} 1 + \frac{f(i) - F}{2s}, & \text{ha } s \neq 0 \\ 1, & \text{ha } s = 0 \end{cases}$$

Ahol,

$f^*(i)$ - az i egyed módosított fitness értéke

$f(i)$ - az i egyedhez tartozó eredeti fitness érték

F - a populáció átlag fitness értéke

s - a populáció fitness értékeinek a szórása

Boltzmann szelekció

A szigma scaling, a „szelekciós nyomást” egy konstans érték körül tartja, szükség lehet, hogy jobban kihangsúlyozzunk a magasabb fitness értékű egyedeket.

Egy szelet meghatározása:

$$f^*(i) = \frac{\exp\left(\frac{f(i)}{T}\right)}{\sum \exp\left(\frac{f(i)}{T}\right)}$$

Ahol,

$f^*(i)$ - az i egyed módosított fitness értéke

$f(i)$ - az i egyedhez tartozó eredeti fitness érték

\sum - az aktuális populáció középértékét jelöli

T - egy hőmérséklet, amely egyenletesen csökken a folyamat során. Alacsonyabb T érték mellett nő a különbség az $f^*(i)$ -ben az alacsony és magas fitness értékek között

IRE
GNSS - Global Navigational Satellite System

GLONASS (orosz)

GALILEO/EGNOS (EU)

GPS (amerikai)

Föld körül pontosan ismert pályákon keringő műholdak rendszere

GPS rendszer

GPS műholdak

- NAVSTAR rendszerben
 - 21 aktív műhold / 3 tartalék
 - távolság (pálya sugár) 26.370km
 - orbitális sík 55 fok az egyenlítő síkjához
 - keringési idő 12 óra
- élettartam 7-8 év
 - elektronika elavul
 - üzemanyag elfogy (pályamódosításhoz)
- időt cézium atomórával

Földi állomások feladata (Nemzetvédelmi minisztérium)

- műholdak nyomon követése
- állapot, pozíció figyelés
- pályaadat korrekciókat küldeni
radarral vizsgálják a holdakat (magasság, helyzet, sebesség), megadják a pályahibát
- holdak óráját összeszinkronizálni

GPS vezérterv

A műholdak pályája úgy van megtervezve, hogy a föld minden pontján legalább öt műhold egyszerre látható legyen

Minden vevőkészülékben van egy *almanach*, amely pontosan megmondja, hogy az egyes műholdak adott pillanatban épp hol tartózkodnak, új adat esetén frissül

GPS vevő paraméterek

csak vevőkészülék

csatornaszám: egyszerre hány műholdat tud érzékelni

- 5-6 még rossz pozíció
- 7-8 már elfogadható
- 12: drón

update rate (ismétlési frekvencia)

- autó/séta: 1Hz
- robotrepülő: 5-10-20Hz

acquisition time (készenléti idő)

- warm: 15s
- cold: 45s (akár 30 min)

Pontosság

- hol és milyen irányban pontos?
- abszolút pontosság?

Helymeghatározás

A GPS műhold pályáját ismerjük → műhold helyét tudjuk

A vevő egység csak a műhold távolságát tudja mérni

- műhold rádió jelet bocsát ki, aminek ismerjük a terjedési sebességét (300.000km/s)
- vevő a rádió jel futási idejét méri, s ebből számítja a távolságot
- probléma:
 - a mért idő nagyon rövid
 - a vevő órahibája (adó-vevő óra nincsen szinkronban) miatt nem lesz pontos a távolság → késleltetés: a vevő késleltetési ideje megegyezik a műhold jelének futási idejével (ál-véletlen kód)

Műholdas trilateráció (háromszögelés)

Adott:

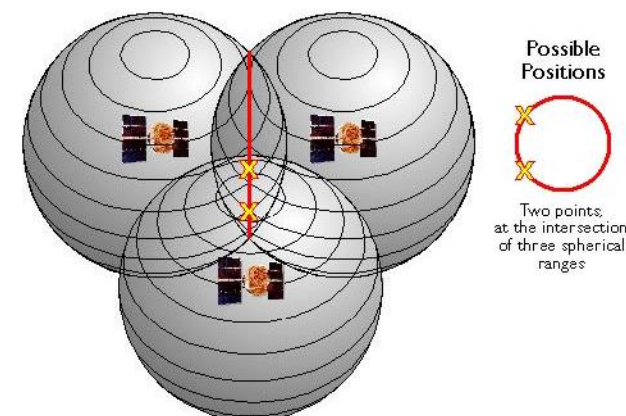
- első műhold 20,000km távolságra
- második műhold 21,000km távolságra
- harmadik műhold 22,000km távolságra

Ekkor:

1 műhold megmutatja, hogy ha a műhold egy 20,000km-s sugarú gömb középpontja, akkor mi ennek a gömbnek a felületén vagyunk valahol.

2 műhold megmutatja, hogy a két gömb áthallásán: egy körön vagyunk

3 műhold esetén 3 gömb áthallásán: két ponton vagyunk



4 műhold mutatja meg, hogy a két pont közül melyiken vagyunk, de általában az egyik pont túl messze esik a földtől, hogy valós legyen. Emiatt sok esetben a negyedik mérés nélkül is eldönthető a kérdés.

GPS jelek**L1 vivőhullám**

- 1575,42 MHz
- helyzeti üzenetek
- ál-véletlen kód időmeghatározáshoz

L2 vivőhullám

- 1227,60 MHz
- katonai ál-véletlen kód

Ál-véletlen kódok

- durva kód (C/A Coarse Acquisition)
 - vivő hullámot modulálja
 - 1023 bitenként ismétlődik
 - minden műholdnak saját ál-véletlen kód az azonosításhoz
 - polgári GPS alapja
- P (precise) kód
 - 266,4 naponként ismétli önmagát
 - egyedi kód azonosításhoz
 - kód generálás minden GPS-hét kezdetekor (vasárnaponként)
 - modulálja L1 és L2 vivőfrekvenciát
 - katonai felhasználás
 - bonyolult, ezért a katonai is először C/A-ra csatlakozik
 - ellenállóbb a zavarásra
- Y kód
 - P titkosított változata

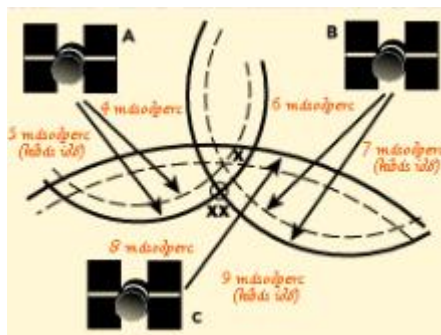
Ál-véletlen kód előnye

- időmérő jel
- minden műhold használhatja ugyanazt a frekvenciát, anélkül, hogy egymást zavarják. (egyedi azonosítás)
- lehetővé teszi a gyenge GPS jelek erősítését

Problémák / Pontosság**1. Órahiba kiejtése**

A vevő az órahibája miatt nem tudja pontosan megmérni a műhold távolságát, amíg ez a helyzet fennáll úgynevezett Pszeudo-mérést végeztünk csak el.

Pszeudo-mérés: a mérésben még szerepel az órahiba



Probléma: A és B műhold pszeudo-távolságai egy pontot metszenek ki, XX-et, a C pszeudo-távolsága nem tud átmenni ezen a ponton. (itt csak 2D-ben vázoltuk fel a problémát, a 3D-s megoldás, plusz egy mérést tesz szükségessé)

Az órahiba minden mérést befolyásol

Megoldás:

A vevő számítógépe egy olyan egyszerű korrekciót keres, amellyel eléri, hogy az összes mérés egy pontot metsszen ki.

Ezt a korrekciót már a többi méréshez is fel tudja használni → így már szinkronban lesz a műhold és a vevő órája

Ezeket a korrekciós eljárásokat ismételni kell

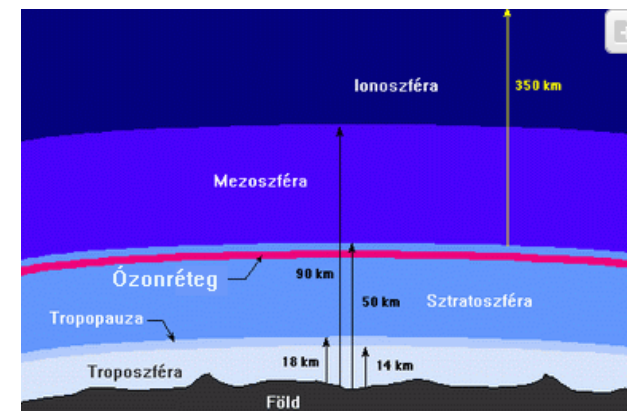
2. Pályahibák

- hold és nap gravitációs hatásától
- napszél hatása a műholdra

Megoldás:

- Földi állomás korrekciós információkat küld a műholdnak szükség esetén
- A tipikus pályahibákat óránként javítják

Ezután ellenőrzésképpen „kimérik” a műhold időjelét (ál-véletlen kódját), így úgymond a kód már pályahiba információ szerzésre is alkalmas.

3. Légköri hibák**Ionosféra (50-500km)**

ionizált részecskék → GPS jelt zavarja (a jel, mint a fény törik, vagy lassul)

Troposzféra (alsó 51km)

- vízgőzzel telített
- változó légnyomás/hőmérséklet
- viszonylag kisebb hibát okoz

IRE

Megoldások:

a) Matematikai modellek

A jel útjának késleltetése az atmoszférán keresztül, jól jósolható

Az atmoszféra matematikai modelljei számolnak a ionizált részecskék és a gáznemű összetevők hatásaival is.

A vevő figyeli, hogy a jel milyen szögben lépett be az atmoszférába, mert a belépési szög meghatározza a zavaró közegben megtett út hosszát.

b) Kétfrekvenciás mérés

Megoldás lehet még a légkör hatásainak kiküszöbölésére, ha kétfrekvenciás mérést használunk, ám ez csak fejlettebb eszközökön elérhető.

Ha egy fény áthalad egy közegen, akkor az alacsony frekvenciájú jelek megtörnek, vagy jobban lassulnak, mint a magas frekvenciájúak.

A GPS L1 és L2 jelét összehasonlítva következtethetünk az atmoszféra hatásaira és ennek megfelelően javításokat alkalmazhatunk.

Csak a katonai vevők tudnak vivőfrekvencián mérni

Polgári eszközökön „csalással” megy

4. Földfelszíni terjedési hibák (visszhang)



Többutas terjedés

A jel nem egyenes vonal mentén terjed, mert visszaverődik a környezeti tárgyakra → ugyanaz a jel időben késve többször érkezik meg a vevőre (először a közvetlen, majd a visszavert)

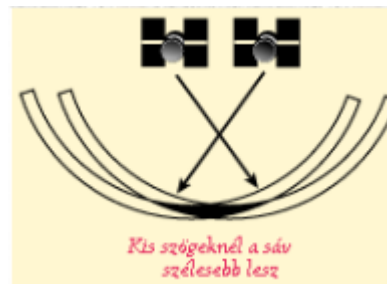
Ha a visszavert jel elég erős, hibát okozhat

Megoldás:

Jobb vevők ki tudják választani a közvetlen jelet

5. GDOP (Geometric Dilution of Precision) hiba

Pontosság geometriai felhígulása: ha túl közeli műholdakat választ ki a vevő, akkor a metszéspontok/körök nagyon lapos szögben metszik egymást:



Megoldás:

- Jó vevők figyelembe veszik a GDOP hibát
- Legjobb metszés a derékszögű

6. Szándékos zavarás (SA = Selective Availability)

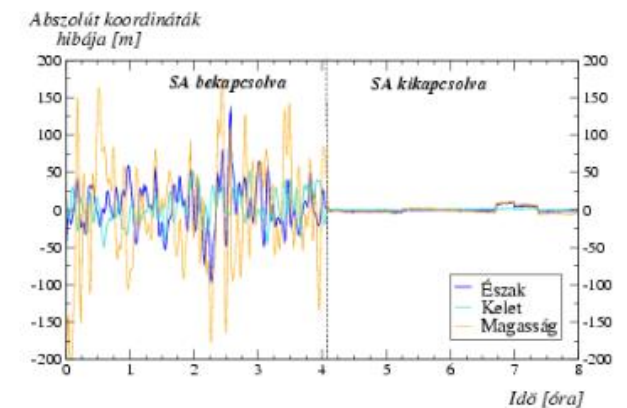
Alapvetően a Nemzetvédelmi Minisztérium egy bizonyos algoritmus szerint „elrontja” a műholdak idő- és pályainformációit.

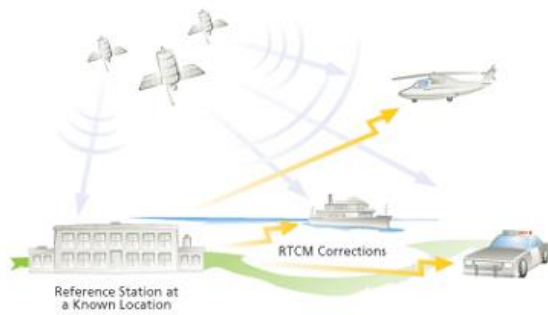
Egy vevős felhasználókat érinti leginkább

Katonai dekóderek ezt el tudják tüntetni

GPS differenciális módszerrel ezek a „hibák” jelentősen csökkenthetők

Clinton 2000 május 2-án kikapcsoltatta → helymeghatározás pontossággal 10x nőt



Real-Time Differential GPS**Differenciális javítási technika**

Létezik egy bázis (referencia) állomás, ami pontosan tudja a helyzetét, de meghatározza azt a GPS műholdak segítségével is és ezt a két értéket összehasonlítja.

A bázis által kiszámolt differencia átadható egy másik vevőnek, amit az alkalmazhat saját számításaihoz.

GPS vs DGPS hibák összehasonlítása

	szabályos GPS	DGPS
Műhold óra	1,5m	0m
Pályahiba	2,5m	0m
Ionoszféra	5,0m	0,3m
Troposzféra	0,5m	0,2m
Vevő zaj	0,3m	0,3m
Többutas terjedés	0,6m	0,6m
SA	30m	0m