

***Koncepció Rendszeranalízis  
Megvalósítására SIMTONIA-ban***

Asbóth Árpád

2015.március

## Tartalom

1.	Bevezetés.....	2
2.	Eredő átviteli függvény számítása.....	3
2.1.	Alapfüggvények SISO esetben (Single Input, Single Output).....	3
2.1.1.	Soros kapcsolás .....	3
2.1.2.	Párhuzamos kapcsolás.....	3
2.1.3.	Visszaesatolás.....	4
2.2.	Hatásvázlatok átalakítása.....	4
2.2.1.	Ekvivalens kapcsolások: .....	5
2.3.	Példák hatásvázlatok átalakítására.....	5
2.4.	Példa MIMO rendszer eredő átviteli függvényének számítására .....	9
2.5.	Algoritmizálhatóság.....	10
3.	Eredő átviteli függvény rekurzívvá tétele.....	12
3.1.	Analitikus átalakítás.....	12
3.1.1.	Elsőrendű tag átalakítása: .....	12
3.1.2.	Másodrendű tag átalakítása: .....	12
4.	Rendszeranalízis.....	15
4.1.	Ugrásválasz.....	15
4.1.1.	Görbeanalízis másodrendű átviteli függvény esetén .....	16
4.2.	Impulzusválasz: .....	17
4.3.	Zérus-Pólus elhelyezkedés.....	18
4.4.	Bode-diagram .....	18
4.5.	Nyquist diagram .....	20
5.	Összefoglalás:.....	22
6.	Matlab Control System Toolbox .....	23

## 1. BEVEZETÉS

Egy rendszer analíziséhez és szimulációjához szükséges modell között alapvető különbségek vannak, azonban az ezek közötti átjárhatóság megoldható. Mindkettő matematikai modell, de az adott folyamat megközelítése más. Egy szimulációnál a valós fizikai, kémiai vagy egyéb jelenségeket igyekszünk közelíteni valamilyen összefüggéssel, vagy differenciálegyenlettel. Így sikerül egy olyan matematikai modellt felállítani, mely a valódi társa által produkált jelenségeket jó közelítéssel adja vissza. A rendszertechnikai modell esetében azonban már elrugaszkodunk a fizikai jelentéstől, és kizárólag matematikai jelekkel foglalkozunk, illetve ezek transzformáltjaival.

A szabályozástechnika alapelveit, módszereit a legtöbb egyetemen hasonlóképpen oktatják. Valamint a szakemberek többsége is e megközelítéseket ismeri. Ezeket a módszereket foglalja egybe a MathWorks MATLAB szoftver Control System Toolbox programcsomagja. Így a rendszeranalízis bármilyen programba történő implementálása során nem kerülhetjük el a CST megvalósítási elveit. A továbbiakban e programcsomag megközelítése, illetve más megkérdezett mérnökök véleménye alapján állítom össze, mire érdemes koncentrálni egy rendszeranalízist megvalósító szoftverrésznél. Külön figyelve azokra a részekre, amelyek az oktatásban hasznosak lehetnek, illetve hangsúlyt kapnak.

## 2. EREDŐ ÁTVITELI FÜGGVÉNY SZÁMÍTÁSA

A jelátviteli tagok között a következő alapkapsolások adhatók meg:

- soros kapsolás
- párhuzamos kapsolás
- visszacsatolás

Ezekre az alapkapsolásokra, könnyen számolható függvények vannak. Azonban egy rendszerben általában ennél jóval bonyolultabb viszonyokat is találhatunk. Az eredő átviteli függvény meghatározásához a komplexebb kapsolásokat vissza lehet vezetni az elemiekre, amiket egyenként ki tudunk számolni.

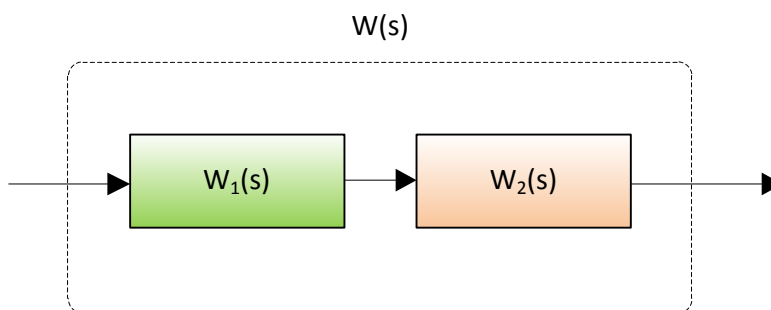
Fontos megjegyezni, hogy az alábbi matematikai műveletek az alábbi alakú Laplace tartománybeli átviteli függvényekre igazak, ahol  $U(s)$ -el jelöljük a rendszer bemenetének differenciálegyenletének Laplace tartományát, ahol  $s$ , a Laplace operátor.  $Y(s)$  jelöli a kimenet differenciálegyenletének Laplace transzformáltját. A két egyenlet hányadosa adja a  $W(s)$ , átviteli függvényt, melynek már az alakja is sok információt elárul a rendszer tulajdonságairól.

$$W(s) = \frac{U(s)}{Y(s)} = \frac{a_0 + a_1s + a_2s^2 + \dots + a_ns^n}{b_0 + b_1s + b_2s^2 + b_ms^m}$$

### 2.1. Alapfüggvények SISO esetben (Single Input, Single Output)

#### 2.1.1. Soros kapsolás

Két átviteli függvényvel rendelkező rendszer soros kapsolását mutatja a 2.1. ábra. Az így kapott rendszer átviteli függvényét a két differenciálegyenlet konvolúciója adja. Ez Laplace tartományban egyszerű szorzást jelent. Az analitikus szorzás könnyen algoritmizálható a konvolúciós algoritmus segítségével.

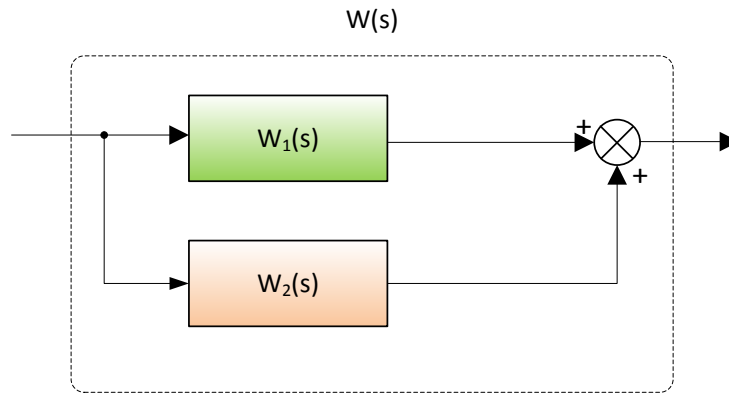


2.1. ábra: soros kapsolás

$$W(s) = W_1(s)W_2(s)$$

#### 2.1.2. Párhuzamos kapsolás

Amennyiben a  $W_1(s)$  átviteli függvényvel és a  $W_2(s)$  átviteli függvényvel rendelkező alrendszerek párhuzamosan vannak kapsolva, akkor mindkét alrendszer ugyanazt a bemenetet kapja, a kimeneteik pedig összeadódnak, vagy kivonódnak.



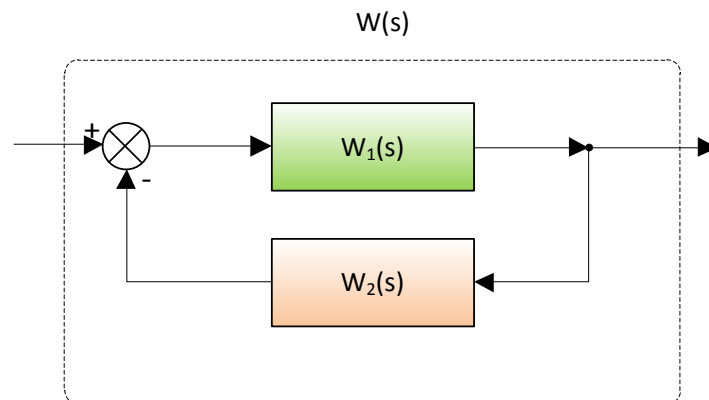
2.2. ábra: párhuzamos kapcsolás

Az eredő átviteli függvény a két alrendszer átviteli függvényének összegével állítható elő:

$$W(s) = W_1(s) + W_2(s)$$

### 2.1.3. Visszacsatolás

A szabályozási körök alapvető jellemzője a visszacsatolás. Tekintsük a 2.3. ábrán látható negatív visszacsatolással rendelkező rendszert.



2.3. ábra: negatív visszacsatolás

A rendszer bemenete és kimenete között az alábbi összefüggés vezethető le:

$$y(s) = W_1(s) [r(s) - W_2(s)y(s)]$$

$$[1 + W_1 W_2]y = W_1 r$$

$$W = \frac{y}{r} = \frac{W_1}{1 + \underbrace{W_1 W_2}_{W_0}} = \frac{W_1}{1 + W_0}$$

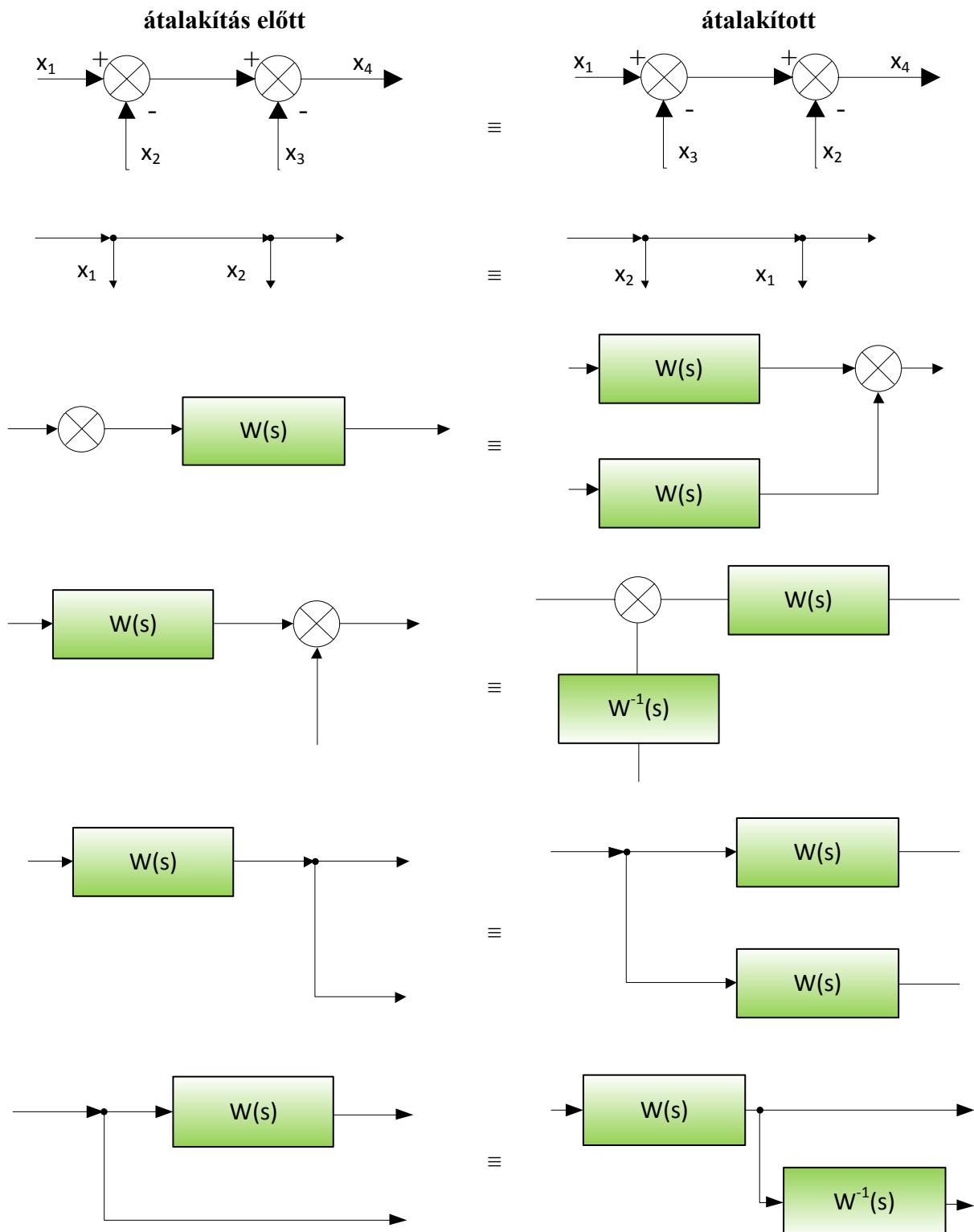
Pozitív visszacsatolás esetén az előjelek megfordulnak:

$$W = \frac{W_1}{1 - W_1 W_2}$$

## 2.2. Hatásvázlatok átalakítása

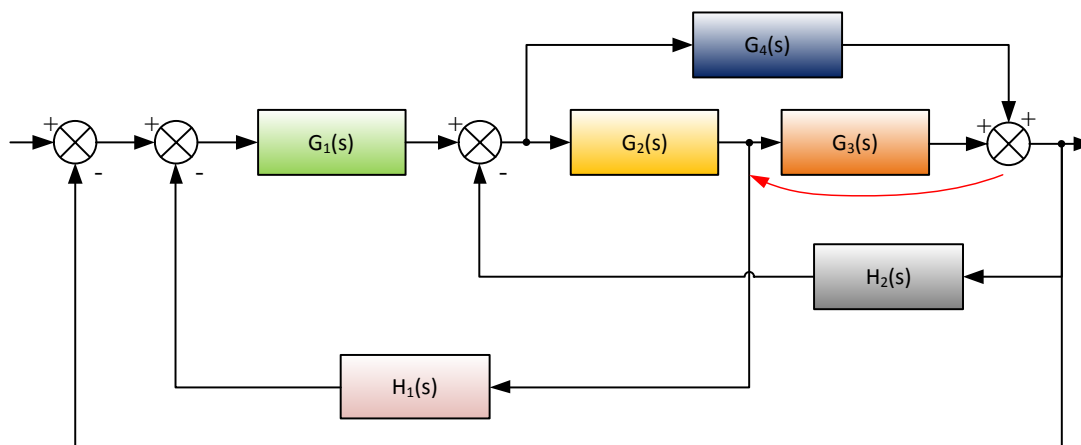
Az összetett rendszerek vizsgálatához szükséges a hatásvázlat ismerete. Az eredő átviteli függvény egyszerű kiszámításához ezekkel az ekvivalenciákkal lehet olyan alakra kell hozni a blokk-diagramot, mellyel egyszerűen kiszámolható az átviteli függvény.

### 2.2.1. Ekvivalens kapcsolások:

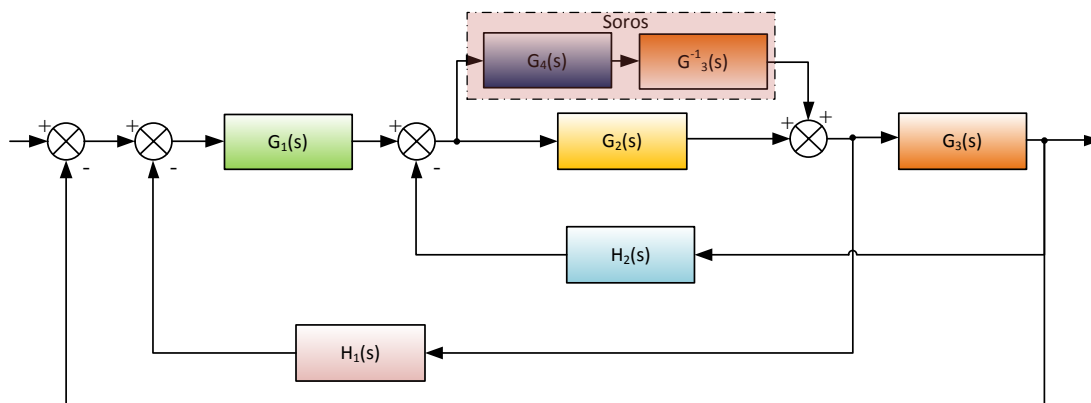


### 2.3. Példák hatásvázlatok átalakítására

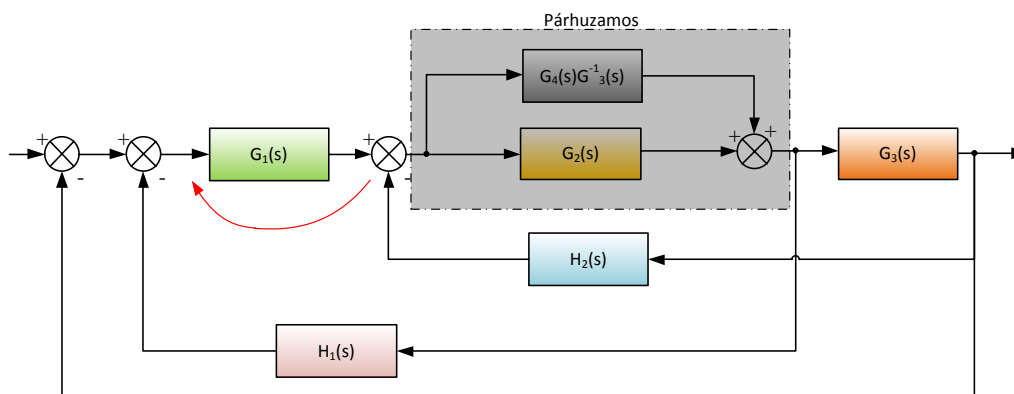
Egy hatásvázlat átalakítása több módszerrel lehetséges. A következőekben nem feltétlenül a leglogikusabb sorrendet fogom követni a műveletekben, inkább törekszem elemi átalakításokra lebontani a lépéssort. Így könnyebben látható az algoritmizálhatóság.

**1. példa:**

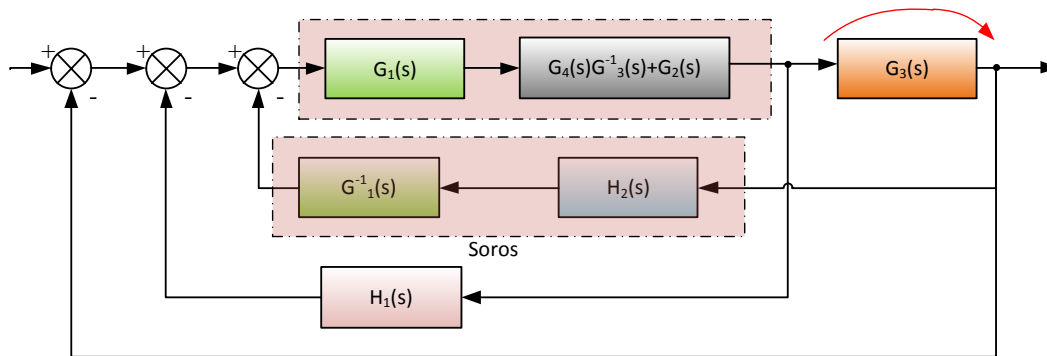
Első lépésként a jobb oldalon levő összegzőt egy blokkal ballra helyezzük át az előbb ismertetett ekvivalens kapcsolásokat felhasználva. Így kapjuk az alábbi diagramot:



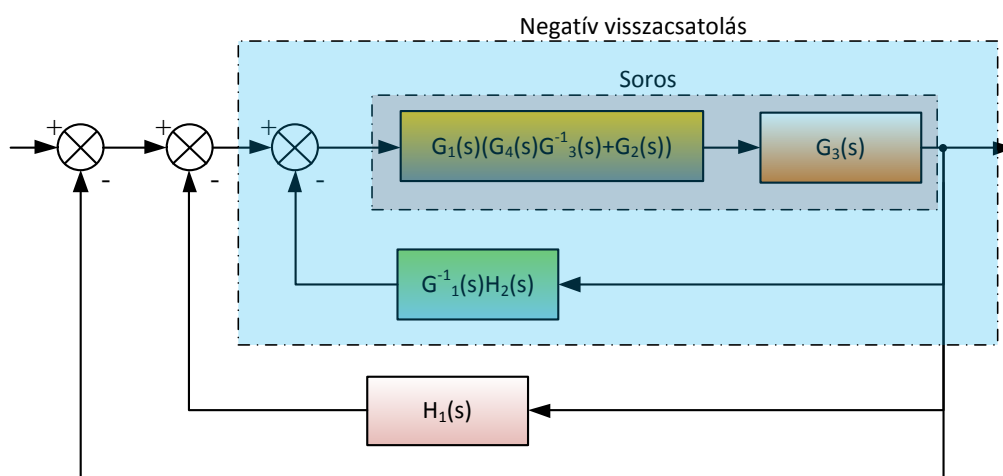
Láthatjuk, hogy az áthelyezésnek köszönhetően a  $G_4$  rendszer az ágában megjelenő új taggal együtt soros kapcsolás jön létre, ami már összevonható, így egyszerűsödik az ábra:



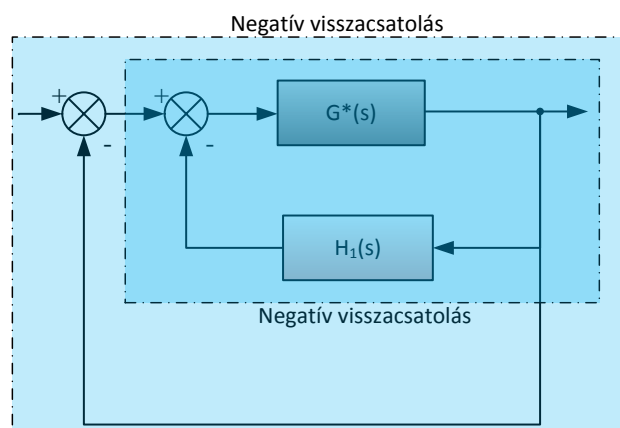
Az összevont blokkok kiadnak egy párhuzamos kapcsolást is. Tovább lépve pedig a  $G_1$  blokk elé kell hoznunk a következő összegzőt.



Így két új sorosan összevonható ágot kapunk. Mivel már minden összegző baloldalon van, ezért áttérhetünk a csomópontok jobbra tolására. Vesszük balról az első csomópontot, és egy blokkal hátrébb toljuk.



Az így kapott új hatásvázlaton ismét fellelhetünk egy soros kapcsolást, valamint egy visszacsatolást is össze tudunk vonni. Az összevonások következtében jutunk az alábbi ábrához:

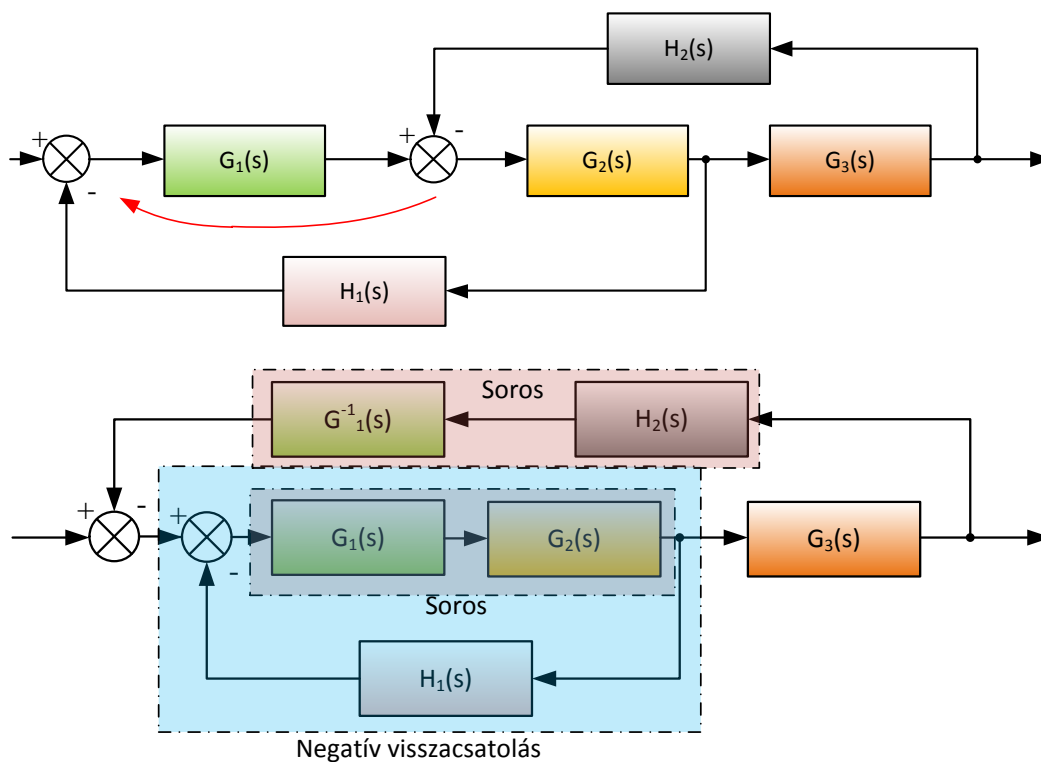


Látható, hogy már mindössze két negatív visszacsatolásra egyszerűsödött le rendszerünk. A szokásos műveleteket elvégezve végül megkapjuk az eredő átviteli függvényt.

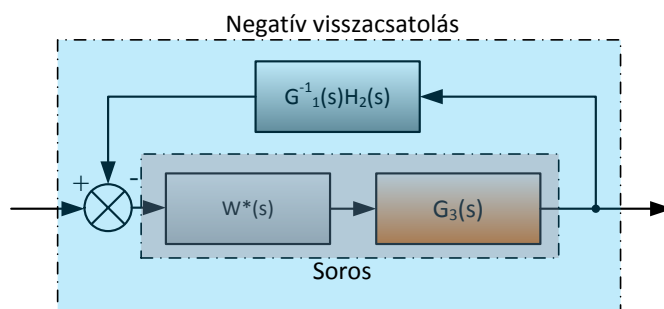
## 2. példa:

Az alábbi hatásvázlat egy igen klasszikus problémát vet fel. Látható, hogy két negatív visszacsatolás van a rendszerben, azonban a két részrendszer szerves részét alkotja egymásnak, így nem választhatók egyszerűen ketté. A megoldás az előző példához hasonlóan kezdődik. Jobbról az első összegzőt és a tőle ballra elhelyezkedő blokkot megcseréljük az azonosságok felhasználásával.



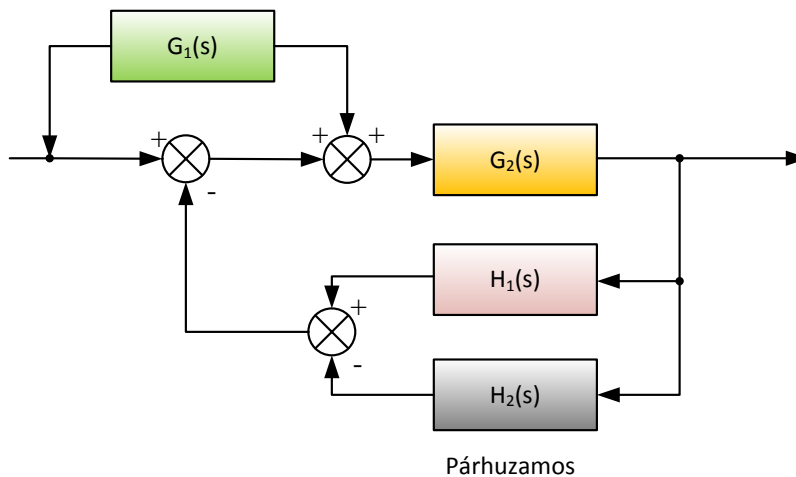


Az elmozgatással rögtön megkapunk egy párhuzamos kapcsolást, és egy negatív visszacsatolást.

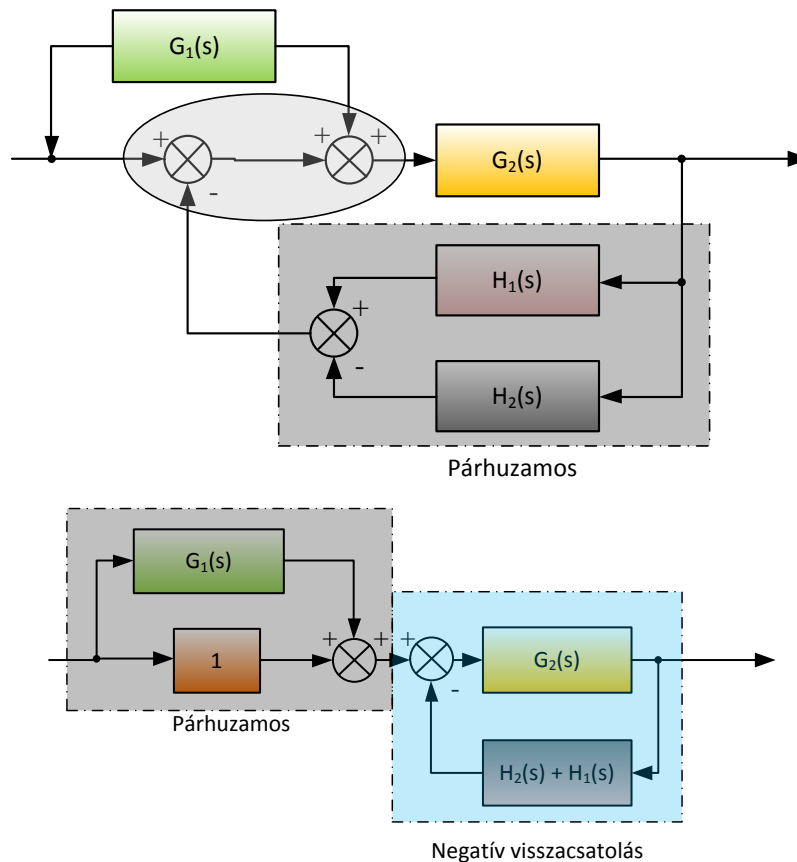


Az összevonásokat elvégezve láthatjuk, hogy nincs szükség további mozgításokra. A kapott soros kapcsolat, és negatív visszacsatolás összevonásának elvégzésével megkapjuk az eredő átviteli függvényünket.

### 3. példa:



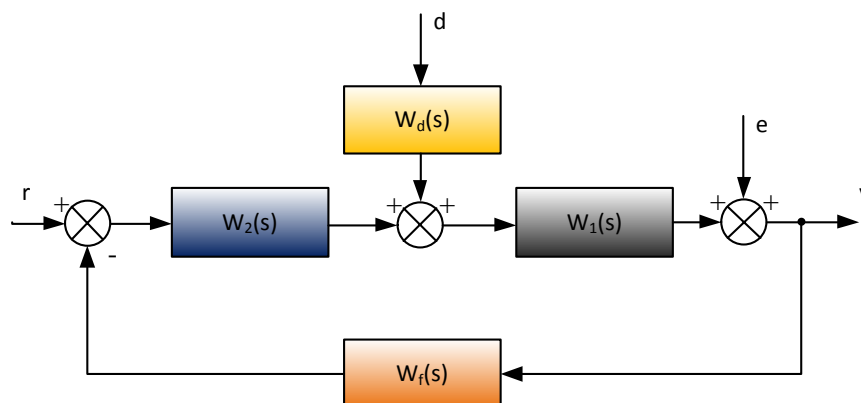
A következő rendszer bonyolultabbnak tűnik, azonban a párhuzamos rendszer összevonásával, és az összegzők egyneműnek tekintésével, és némileg átrendezve kapjuk a második ábrát, ami pedig egy újabb párhuzamos kapcsolást ad, illetve egy negatív visszacsatolást.



## 2.4. Példa MIMO rendszer eredő átviteli függvényének számítására

A gyakorlatban leginkább MIMO rendszerekkel találkozhatunk (Multi Input Multi Output). A többszörös bemenetet nem csak más inputok adhatják, hanem a számításba vett zavar is. Ezt úgy szokás jelölni, hogy az részrendszer utolsó tagja után, de még a visszacsatolás leágazása előtt pozitívan hozzáadjuk a jelhez. Ezáltal jelölve a zaj superponálódását.

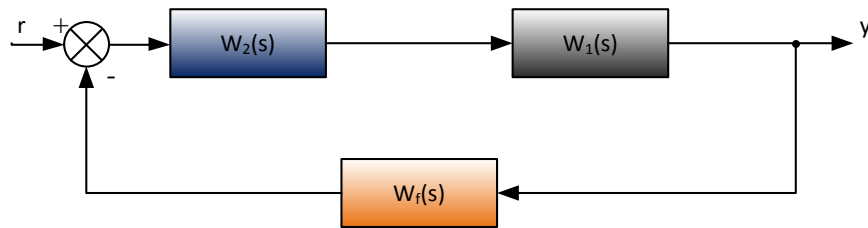
Tekintsük az alábbi ábrát. Látható, hogy egy három bemenetű egy kimenetű rendszerrel



A komponens átviteli függvényeket a szuperpozíció elve alapján és a visszacsatolt rendszerek alapképleteinek felhasználásával határozhatjuk meg. Az átviteli függvény itt valójában egy  $1 \times 3$  méretű mátrix, mivel 1 kimenetű, és 3 bemenetű rendszerről beszélünk.

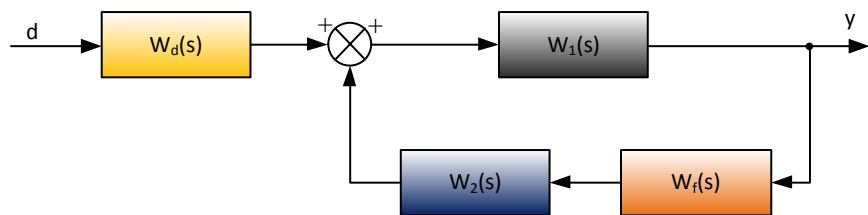
Az előzőekben megfogalmazott elvek miatt a kimenet és az egyes bemenetek között az eredő átviteli függvények a következőképpen alakulnak:

Az  $r$  bemenettől az  $y$  kimenetig



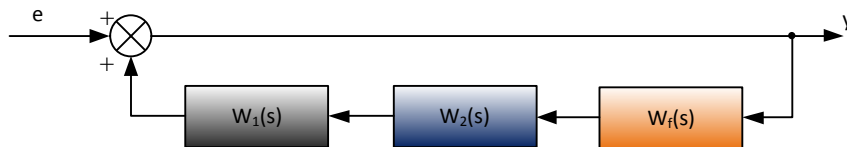
$$W_{yr} = \frac{W_1 W_2}{1 + W_1 W_2 W_f}$$

A  $d$  bemenettől az  $y$  kimenetig:



$$W_{yd} = \frac{W_1 W_d}{1 - W_1 W_2 W_f}$$

Az  $e$  zavarójeltől az  $y$  kimenetig vett hatásvázlat:



$$W_{ye} = \frac{1}{1 - W_1 W_2 W_f}$$

A szuperpozíció elve alapján a kimenő jel Laplace-transzformáltja az egyes bemenetekről gerjesztetett kimenetek Laplace-transzformáltjának összege. Tehát:

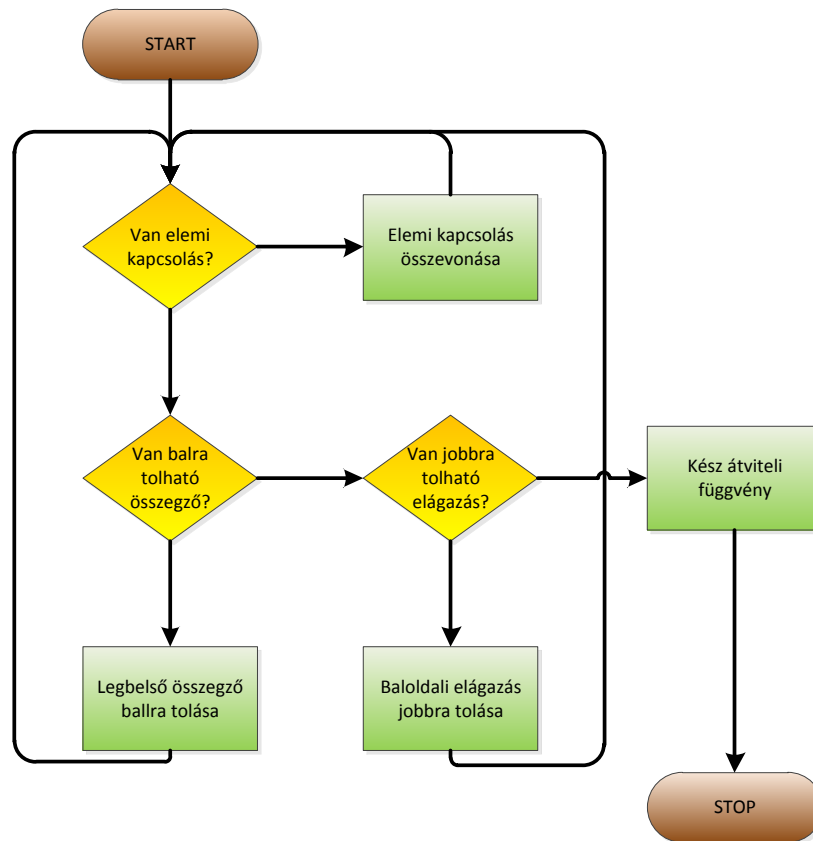
$$Y(s) = W_{yr}(s)R(s) + W_{yd}(s)D(s) + W_{ye}(s)E(s)$$

## 2.5. Algoritmizálhatóság

A fentiekben láthattuk, hogy bonyolultabb rendszerek esetén is, egy egyértelmű algoritmus alapján egyszerűbb alakra hozható a hatásvázlat, végül pedig meghatározható az eredő átviteli függvény.

Egy ilyen belső reprezentáció létrehozásához fontos kezdeti feltétel a rendszerblokkok maradéktalan definiálása, mely magában foglalja az átviteli függvényt, a be és kimenetek elnevezéseit. Továbbá egzakt módon definiálni kell a blokkok egymáshoz való kapcsolatait. Valamint biztosítani kell az egymás szomszédságában lévő összegzők, illetve elágazási pontok ekvivalenciáját, amivel megvalósítjuk a kapcsolásoknál alkalmazott felcserélhetőséget.

Így a fent bemutatott példák megoldása során alkalmazott elveket az alábbi folyamatábrában tudjuk összefoglalni:



A módszer segítségével az eredő függvény meghatározásához mindössze 3 matematikai műveletre van szükség a Laplace-tartományon:

- összeadás
- kivonás
- szorzás

Továbbá teljesülnie kell, hogy rendszer invertálható legyen, azaz legyen kauzális, és ne rendelkezzen holtidővel.

### 3. EREDŐ ÁTVITELI FÜGGVÉNY REKURZÍVÁ TÉTELE

A fentiek során elkészült átviteli függvényünket már fel tudjuk használni a következő analízisekhez. Azonban néhány esetben szükség van az egyenlet időtartománybeli rekurzív alakjához. Ezek segítségével lehet készíteni egy megfelelő időfelbontással egy gyors analízist. A lineáris rendszerek nem tartalmaznak nemlineáris tagot, ami egy jó könnyítés az átalakításban. Elsőként az analitikus kézi módszert tekintjük meg, amelyből levezethető az algoritmizálható megoldás.

#### 3.1. Analitikus átalakítás

Nézzünk két példát a Laplace-tartománybeli átviteli függvényünk gyors átalakítására analitikusan

##### 3.1.1. Elsőrendű tag átalakítása:

Tekintsük a következő átviteli függvényű rendszert:

$$W(s) = \frac{K}{\tau s + 1}$$

Ezt alakítsuk át, majd tegyük át folytonos időbe:

$$W(S) = \frac{Y(S)}{U(S)} \rightarrow Y(S)(\tau s + 1) = sY(S)\tau + Y(S) = KU(S)$$

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t)$$

Ezt diszkrétizálva:

$$\tau \frac{y_{k+1} - y_k}{T_0} + y_k = Ku_k$$

Majd átalakítva:

$$\frac{\tau}{T_0} y_{k+1} = y_k \left( \frac{\tau}{T_0} - 1 \right) + Ku_k$$

$$y_{k+1} = \left( 1 - \frac{T_0}{\tau} \right) y_k + \frac{T_0}{\tau} Ku_k$$

Új változók bevezetésével az új rekurzív alakunk:

$$p_1 = 1 - \frac{T_0}{\tau}$$

$$p_2 = \frac{KT_0}{\tau}$$

$$y_{k+1} = p_1 y_k + p_2 u_k$$

##### 3.1.2. Másodrendű tag átalakítása:

Legyen a rendszerünk átviteli függvénye:

$$W(s) = \frac{1}{36s^2 + 12s + 1}$$

Az átalakítás:

$$(36s^2 + 12s + 1)Y(s) = U(s)$$

$$36 \frac{d^2 y}{dt^2} + 12 \frac{dy}{dt} + y = u$$

$$36 \frac{\frac{y_{k+1} - y_k}{T_0} - \frac{y_k - y_{k-1}}{T_0}}{T_0} + 12 \frac{y_{k+1} - y_{k-1}}{2T_0} + y_k = u_k$$

$$36 \frac{y_{k+1} - 2y_k + y_{k-1}}{T_0^2} + 12 \frac{y_{k+1} - y_{k-1}}{2T_0} + y_k = u_k$$

$$\left( \frac{36}{T_0^2} + \frac{12}{2T_0} \right) y_{k+1} + \left( 1 - \frac{72}{T_0^2} \right) y_k + \left( \frac{36}{T_0^2} - \frac{12}{2T_0} \right) y_{k-1} = u_k$$

$$p_0 = \left( \frac{36}{T_0^2} + \frac{12}{2T_0} \right)$$

$$p_0 = \left( \frac{36}{T_0^2} + \frac{12}{2T_0} \right)$$

$$p_0 = \left( \frac{36}{T_0^2} + \frac{12}{2T_0} \right)$$

$$p_0 y_{k+1} + p_1 y_k + p_2 y_{k-1} = u_k$$

$$p_0 y_{k+1} + p_1 y_k + p_2 y_{k-1} = u_k$$

Amint látható a másodrendű tag átalakítása már analitikusan is elég hosszadalmas, és ez hatványozottan bonyolódik a fokszám növelésével. Azonban az elv használható a rekurzív alakításnál. A megoldást a Taylor sor visszafejtése adja.

Az  $f$  függvényt  $x_0$  kis környezetében a következő Taylor-sorral közelíthetjük:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{6}(x - x_0)^3$$

Ha a környezet intervallumát  $x_0 - h$  és  $x_0 + h$  alakban írjuk fel, akkor a következő két képletet kapjuk. Ezek az úgynevezett centrális képletek:

$$f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{f''(x_0)}{2}h^2 - \frac{f^{(3)}(x_0)}{6}h^3$$

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2}h^2 + \frac{f^{(3)}(x_0)}{6}h^3$$

Ha ezeket összevonjuk:

$$f(x_0 - h) + f(x_0 + h) = 2f(x_0) + f''(x_0)h^2 + \frac{f^{(4)}(x_0)}{12}h^4$$

Ebből a 3 egyenletből ki lehet fejezni akárhányadik deriváltat.

Az első:

$$f(x_0 - h) = f(x_0) - f'(x_0)h$$

$$f'(x_0)h = f(x_0) - f(x_0 - h)$$

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h}$$

Ami a klasszikus Newton-formula

A második derivált:

$$f(x_0 - h) + f(x_0 + h) = 2f(x_0) - f''(x_0)h^2$$

$$f''(x_0)h^2 = f(x_0 - h) - 2f(x_0) + f(x_0 + h)$$

$$f''(x_0) = \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2}$$

És így tovább.

A megfelelő formula kiválasztása után az egyenletből kivesszük az időben „legújabb” tagot, és arra fejezzük ki az egyenletet.

*Összefoglalva a következő lépésekkel jutunk el a rekurzív egyenlethez*

1. átviteli függvény átrendezése
2. folytonos idejűvé alakítás
3. megfelelő Taylor-képlet behelyettesítése a megfelelő helyekre
4. rekurzívvá alakítás

## 4. RENDSZERANALÍZIS

A továbbiakban felvázolom, az egyes módszereknek milyen kimeneteinek kell lennie, amelyek egy tervező mérnök számára fontosak. A következő analíziseket használják leggyakrabban:

- Ugrásválasz (Step-response)
- Impulzusválasz (Impulse-response)
- Bode-diagram
- Nyquist-diagram
- Zérus-Pólus térkép (Pole-Zeros Map)
- Pólusvándorlás térképe

Az analízis során a példafüggvényünk, amelyről az analízis ábráit készítettem. Úgy választottam meg, hogy a bemutatott paraméterek többsége jól látszódjon:

$$W(s) = \frac{s + 5}{40s^3 + 54s^2 + 15s + 1}$$

Valamint, ahol kiemelten a másodrendű rendszerről beszélek, ott a következő átviteli függvényű rendszert használtam:

$$W(s) = \frac{s + 5}{5s^2 + s + 1}$$

### 4.1. Ugrásválasz

Ha ismerjük egy lineáris rendszer adott gerjesztéshez (vagy ún. vizsgálójelhez) tartozó válaszát, akkor ennek ismeretében meg tudjuk határozni a rendszer tetszőleges gerjesztéséhez tartozó válaszát is. Ilyen vizsgálójel például az egységugrás-jel és a Dirac-impulzus. Az előbbire adott választ nevezzük a rendszer ugrásválaszának, a másodikra adottat pedig az impulzusválaszának.

Az egységugrás jelet a következőképpen definiáljuk:

$$\varepsilon(t) = \begin{cases} 0, & \text{ha } t < 0 \\ 1, & \text{ha } t \geq 0 \end{cases}$$

Az egységugrás-jel generálása a legegyszerűbb egy valós rendszeren, így könnyen ellenőrizhetjük a számolt értékeket. Egyszerűen bekapcsoljuk a kapcsolót, megnyitunk egy szelepet, azaz valamilyen értéket 0-ból 1-esre állítunk egy bemeneten, amivel olyan kimenetet kapunk, amivel tudjuk jellemezni a rendszer tranzienseit, és alapvető paramétereit. Értelemszerűen a cél, hogy a rendszer minél jobban, stabilan, és minél kisebb túllendüléssel, vagy anélkül kövesse le a vezérlő jelet.

A rendszer egységugrásra adott válaszát nevezzük átmeneti függvénynek (jelölése:  $v(t)$ ). Ez rendszerjellemző. Az ugrásfüggvény változtatásával a következő alapvető információkat tudjuk megkapni a rendszerről:

#### 1. Időinvariancia:

Időinvariáns, azaz időtől független lefolyású egy függvény, ha a viselkedése nem függ az időben való elhelyezkedésétől. Azaz ha elindítjuk, akkor ugyanúgy viselkedik a  $t = 0$  és bármely  $t$  időpillanatban.

Legyen egy lineáris, invariáns és kauzális rendszer ugrásválasza az  $s(t) = \varepsilon(t)$  gerjesztésre a következő:

$$v(t) = \varepsilon(t)e^{-2t}$$

Ha ugyanezen rendszer gerjesztése  $s(t) = \varepsilon(t - 4)$ , azaz  $t = 4$  s-al eltolott, akkor a rendszer kimenetén az invariancia miatt a következő válaszjel jelenik meg:

$$y(t) = v(t - 4) = \varepsilon(t - 4)e^{-2(t-4)}$$

Továbbá ez igaz bármely  $t$ -re.



## 2. Linearitás

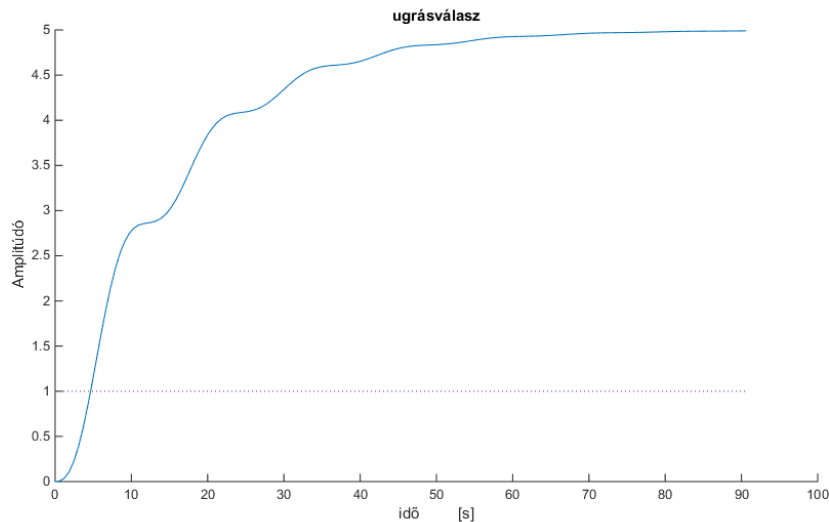
Egy rendszer lineáris, ha adott bemenethez tartozó kimenet esetén, valós konstanssal szorzott bemenetre valós konstans-szoros kimenetet kapunk eredményként.

Azaz, ha a rendszer gerjesztése:  $s(t) = 2 * \varepsilon(t)$

Akkor a rendszer kimenete:  $y(t) = 2 * v(t)$

### 4.1.1. Görbeanalízis másodrendű átviteli függvény esetén

Az átmeneti függvény görbéjének közvetlen az ugrás ráadása utáni szakaszának vizsgálatából számos rendszerparamétert tudunk megkapni.



4.1. ábra: Negyedrendű rendszer ugrásválasza

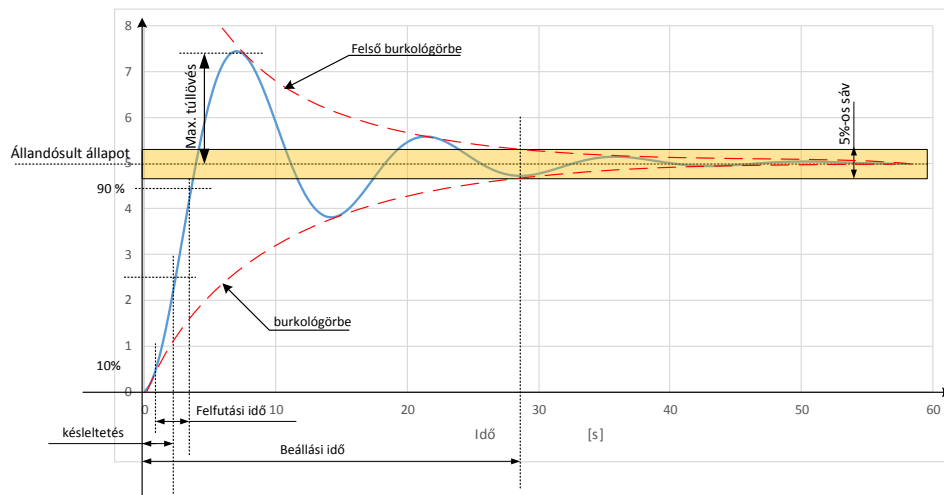
Mivel a szabályozástechnikában sokszor a tervezésnél az átviteli függvény pólusait úgy választjuk meg, hogy a domináns póluspár az origóhoz közelebb legyen, míg a többi, az ennek minimum 3-4-szerese. Így az azok által befolyásolt tényezők túl hamar lejátszódnak, és nem vesznek részt számottevően a transziens folyamatokban. Így a tervezés során sokszor csak a domináns póluspárhoz tartozó másodrendű függvénnyel foglalkozunk, amelynek sajátos jelleggörbéi vannak.

Egy másodfokú rendszer esetében az ugrásválasz görbéjéről a következő paramétereket tudjuk meghatározni:

- állandósult állapot
- késleltetés
- felfutási idő (az állandósult állapot 10%-ától szükséges idő, annak 90%-ának eléréséig)
- Beállási idő (a burkológörbe adott hibaszázalékos sávba érkezésének ideje)
- Maximum túllövés (az ugrásra adott legnagyobb válasz)

#### Kvalitatív információk:

- lecsengés amplitúdója, hossza, és ezek viszonya jól jellemzi a rendszer stabilitás-állapotát
- a felső és alsó burkológörbe látszólagos szimmetriája. Az első hullámok kiugró csúcscértékei jelzik az integráló tagok (szabályozók) túlvezérlődését.



4.2. ábra: Másodrendű rendszer ugrásválasza, és rajta a legfontosabb paraméterek

#### 4.2. Impulzusválasz:

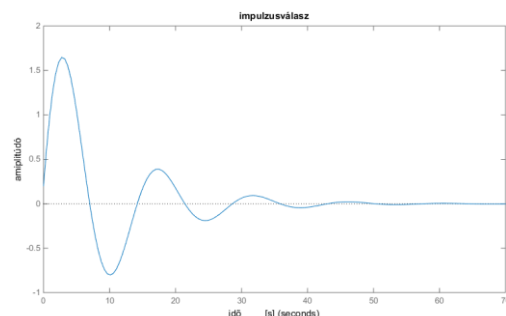
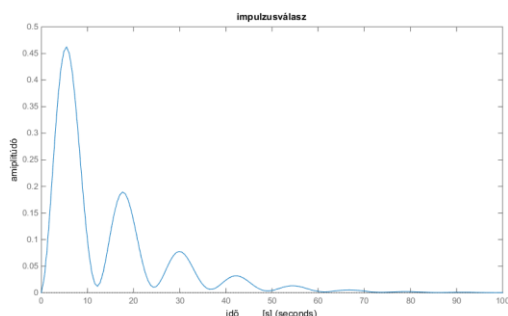
Az impulzusválasz egy másik rendszerjellemző, amellyel lehet vizsgálni az előzőekhez hasonlóan a rendszer időinvarianciáját, illetve linearitását. Ha két rendszer azonosságát, vagy egy időben folytonos függvényt, és időben diszkrét függvényt hasonlóságot vizsgáljuk, akkor az impulzus-ekvivalencia szigorúbb követelmény, mint az egységugrás ekvivalencia.

Az impulzust adó Dirac-delta a következőképpen definiálható:

$$\delta_k = \begin{cases} 0, & \text{ha } k < 0 \\ 1, & \text{ha } k = 0 \\ 0, & \text{ha } k > 0 \end{cases}$$

Az impulzusválaszból hasonló paraméterek következtethetők ki, mint az ugrásválaszból:

- maximum túllendülés
- burkológörbe
- felfutási idő
- késleltetés
- belengés vizsgálata, periódusidő
- csillapítás



4.3. ábra: Impulzusválasz negyedrendű és másodrendű rendszer esetén

### 4.3. Zérus-Pólus elhelyezkedés.

Az analízisnek ez a módszere egy egyszerűbb műveletet igényel, azonban kézzelfogható, és azonnal láthatóan megmutatja, hogy a rendszer stabil-e vagy sem.

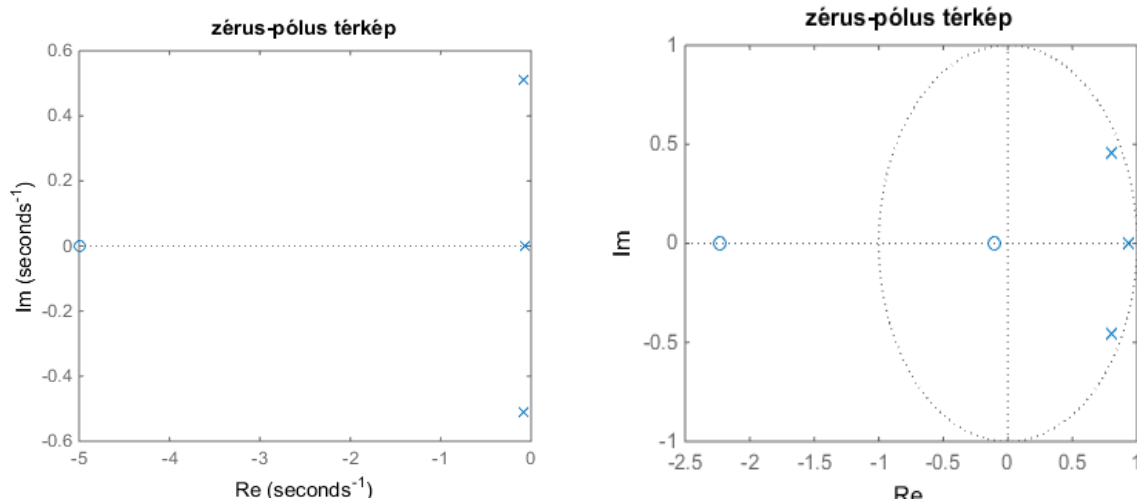
A rendszer zérusait, az átviteli függvény gyökei adják, azaz, ahol az egyenlet 0 értéket vesz fel. A rendszer pólusai pedig az átviteli függvény gyökei adják, melyek a rendszer viselkedését nagyban befolyásolják.

Ha a rendszer pólusai:  $s_1, s_2 \dots s_i$

A rendszer stabil, ha:

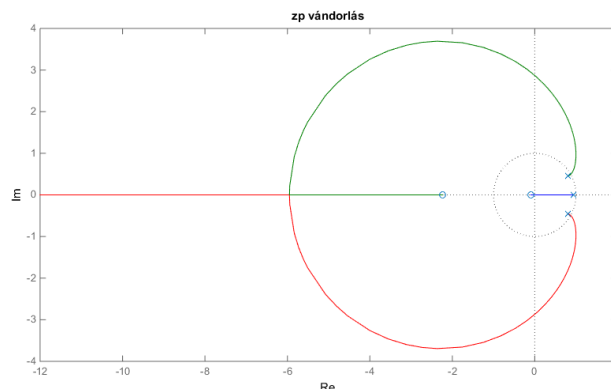
- folytonos időben a pólusok valós része negatív, azaz  $Re(s_i) < 0$
- diszkrét időben az egységkörön belül helyezkedik el, azaz  $|s_i| < 1$

Abban az esetben, ha a feltételnél egyenlőség teljesül, stabilitási határhelyzetben vagyunk, további vizsgálat szükséges, azonban ez kerülendő, mivel bármi átlendítheti a rendszer instabil állapotba.



4.4. ábra: Zérus-pólus térkép negyedrendű rendszer esetén. Baloldalt folytonos időben, jobboldalt diszkrét időben ( $T_s=1$  s)

Tervezés során érdemes figyelni, hogy változnak a rendszer zérushelyei, és pólusai. Erre szolgál a pólusvándorlást bemutató grafikon. Az alábbi ábrán a zérusok és pólusok vándorlása látható a körerősítés függvényében.



4.5. ábra: Zérusok-Pólusok vándorlása a példarendszerben

### 4.4. Bode-diagram

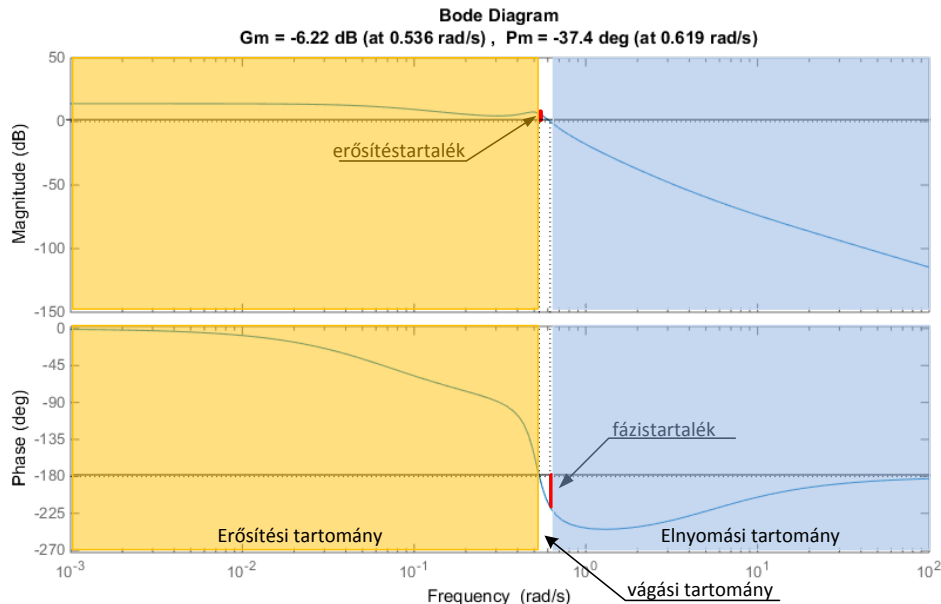
A Bode-diagram széles körben alkalmazott jellemzése egy rendszernek. Két ábra szerepeltetését jelenti általában egymás alatt. Mindkét ábrán, a vízszintes tengelyen a frekvencia van feltüntetve

**logaritmikus rendben, de lineáris értékekkel.** Először az átviteli karakterisztika abszolút értékét ábrázoljuk a frekvencia függvényében, másodszor pedig az átvitel fázisfüggvényét. Az **amplitúdó-karakterisztikánál a függőleges tengelyen logaritmikus skálázást alkalmazunk, és logaritmikus értékeket is tüntetünk fel, míg a másodikban lineáris a skálázás, az értékek feltüntetése is.**

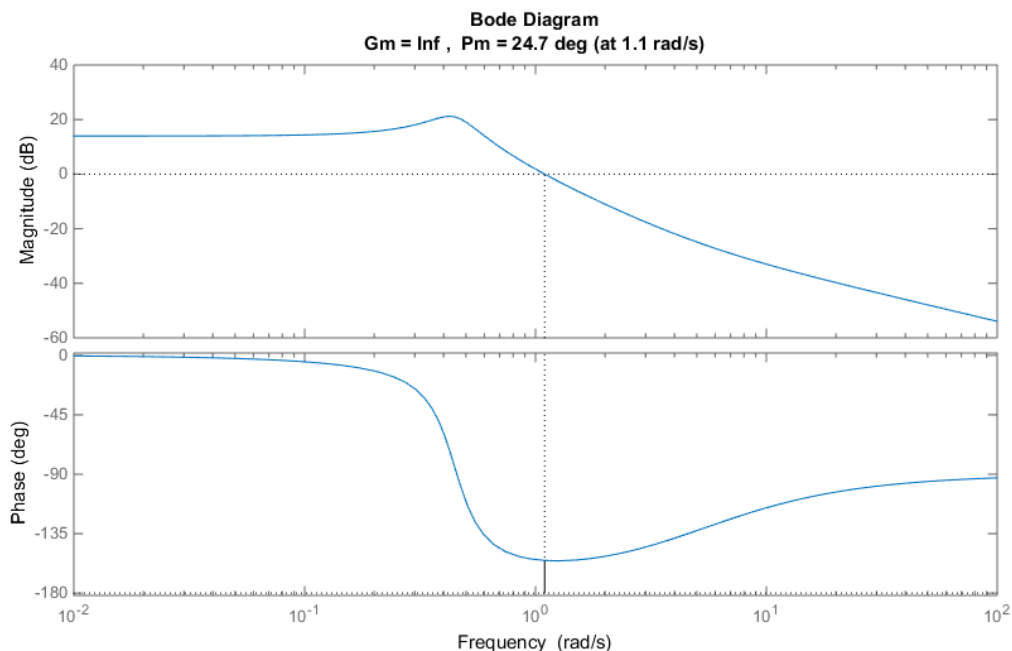
Az egyes karakterisztikákat az alábbi összefüggések szerint kapjuk:

$$a_{db} = 20 \lg |W(j\omega)|$$

$$\varphi(\omega) = \arg(W(j\omega))$$



4.6. ábra: Negyedrendű rendszer Bode-diagramja, és a fontosabb paraméterek ábrázolása



4.7. ábra: másodrendű rendszer Bode-diagramja

A diagramokról egyértelműen leolvasható mely frekvenciatartományokon erősítik a bemenetet, és melyeken gyengítik. Illetve e kettő határán található két fontos érték, amit fontos kijelezni.

Ahhoz a frekvenciához, melyen a fázismenet eléri a  $-180^\circ$ -ot, tartozik az amplitúdó-karakterisztikán az erősítés tartalék, vagy Gain Margin (GM). Szemléletesen szólva ez azt jelenti, hogy ha a

hurokerősítést ennyiszeresére emelnénk, akkor kerülnénk a stabilitás határterületére. Mivel ez a szám itt negatív, ezért jelen esetben csökkentésről van szó.

A másik fontos érték a vágási frekvenciához ( $\omega_c$ ) tartozó fázistartalék ( $\varphi_t$ ), vagy Phase Margin (PM). Azaz azon a frekvenciaértéken felvett fázisérték, amelyen az amplitúdó-karakterisztika metszi a 0 egyenest.

A fázistartalék, vagy más néven fázistöbblet a következő képlettel határozható meg:

$$\varphi_t = \pi + \varphi(\omega_c) \text{ [rad]}$$

#### 4.5. Nyquist diagram

Legyen a rendszer  $W(s)$ . Ha  $s = j\omega$  konverzíóval áttérünk  $W(j\omega)$  alakra, akkor felírhatjuk az átviteli függvényt

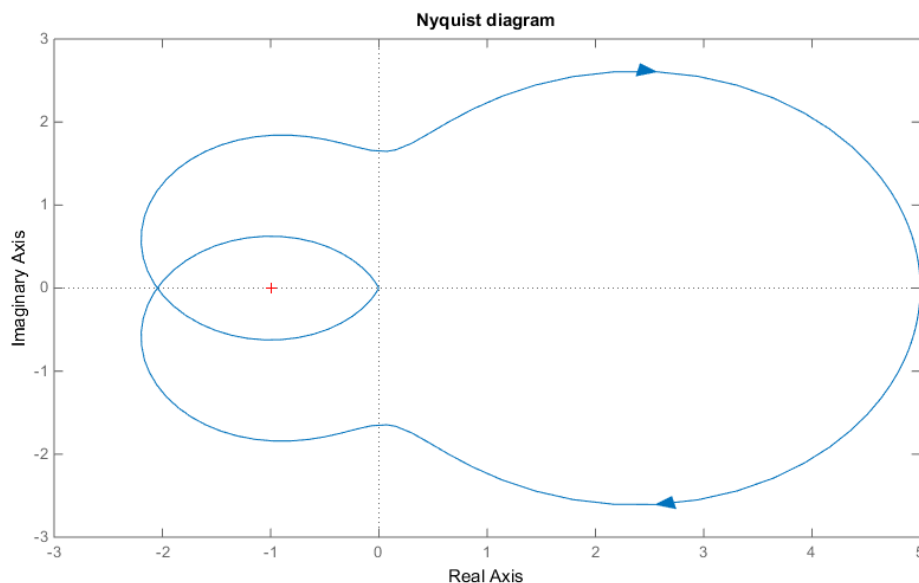
$$W(j\omega) = A(j\omega)e^{j\varphi(\omega)}$$

alakban.

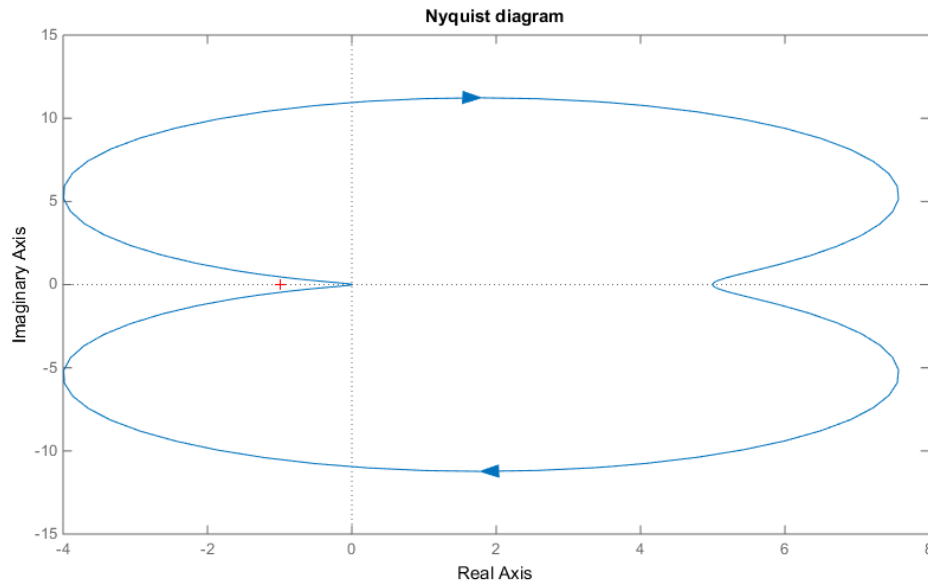
Ha ábrázoljuk  $A(j\omega)$  függvényt, akkor megkapjuk a Bode diagram amplitúdó-karakteristikáját,  $\varphi(\omega)$  pedig megadja a fázismenetet. Azonban ha az egész  $W(j\omega)$  imaginárius részét ábrázoljuk a reális részének függvényében, akkor megkapjuk a Nyquist diagramot.

Másképpen a Nyquist diagram minden egyes frekvenciára a kimenő jelnek a bemenő jelhez viszonyított fáziseltolási szöge és az amplitúdó a komplex sík egy pontját határozzák meg. A pontokat összekötő görbe az amplitúdó-fázis jelleggörbe. Ez egy irányított görbe, amely a frekvencia növekedésének irányát mutatja. A rendszer állandósult állapotának a görbe egy pontja felel meg.

Fontos az ábrázolásnál az exponenciális pontsűrűség. Mert így a kritikus pontok nem esnek ki a tartomány elején. A zárt rendszer (visszacsatolt) stabilis, ha a felnyitott rendszer (tehát, amely nem tartalmaz hurkot) Nyquist diagramja nem veszi körül a  $(-1 + 0j)$  pontot. Ezért ezt fontos jelölni az ábrán.



4.8. ábra: A példarendszer Nyquist diagramja. Látható, hogy a kritikus pontot körbefutja a görbe.



**4.9. ábra:** *A másodrendű rendszer Nyquist diagramja. Itt a kritikus pontot még éppen nem éri el a görbe.*

Stabilitásvizsgálat esetén érdemes továbbá figyelni, hogy változik a diagram a körerősítés növelése során. Ezt érdemes lehet a grafikonon is jelölni.

## 5. ÖSSZEFOGLALÁS:

Véleményem szerint egy mérnök, ha tervezni szeretne, akkor azt egy ahhoz megfelelő szoftverrel fogja elvégezni. A SIMTONIA ehhez egy támogatást tud nyújtani a megfelelő paraméterek kinyerésében. Előnye a grafikus felület lehet, melynél az interaktivitásra kell helyezni a hangsúlyt, melyre jelenleg a piacon erre kevés példát találni. Fontos, hogy a felhasználó egy egyszerű felületen ki tudja választani a megfelelő rendszert, esetleg segítse az eredő átviteli függvény meghatározását azzal, hogy rámutat, mely blokkokat kell eltolni, vagy összevonni. Ezek után a kapott átviteli függvényből a szoftvernek le kell generálnia egy diszkrét rekurzív képletet. Ami egy finom időbeli felbontással elégséges a szabályozástechnikában szükséges analízisekhez. Ennek a gyors rekurzív képletnek az előnye, hogy a tervezés során nem szükséges az leszimulálni a teljes rendszeren az analíziseket, hanem kaphatunk egy gyors választ. Azonban emellett rendelkezésre kell állnia egy olyan lehetőségnek is, hogy a szimulációban is megvalósítható legyen ez.

Az interaktivitás egy ilyen grafikus felületen nagyon fontos fejlesztés lehet véleményem szerint. Ez sok szoftverben nem áll rendelkezésre. A következő funkciók jelentős segítséget adnának:

- Grafikonon való kereshetőség. Szálkereszttel a vízszintes és a függőleges tengelyen lehessen keresni egy oszcilloszkópokhoz hasonlóan. A szálkereszt metszéspontját „vonzza” a görbe, azaz ha a környezetébe kerül, akkor ugorjon rá. Ehhez természetesen szükség van egy gyors interpolációra, hogy a funkció a meghatározott pontok között felrajzolt vonaldarabokon is működjön. Erre egy általánosan alkalmazható megoldás a legkisebb négyzetek módszere (Least Square Method – LSM)
- Legyen lehetőség a grafikon módosítására. Kisebb módosításokra legyen lehetőség, amik hatását azonnal meg lehet jeleníteni a grafikonon. Ilyen lehet például a visszacsatolás hatása a rendszerre, körerősítés változtatása.
- Kritikus tartományok bejelölése. Ilyen lehet például a stabilitás tartománya a zérus-pólus térképen, a fontos paraméterek megjelenítési lehetősége az ugrásválasznál, illetve a Bode-diagramon.

## 6. MATLAB CONTROL SYSTEM TOOLBOX

A MATLAB-ot (matrix laboratory) az 1970-es évek elején Cleve Moler kezdte el fejleszteni, az akkori Új-Mexikói Egyetem. Kezdetben csak a diákjai munkáját tervezte megkönnyíteni. Hamarosan elterjedt más egyetemek hallgatói és munkatársai között is és így erős érdeklődésre tett szert az alkalmazott matematikával foglalkozók körében. Jack Little, egy mérnök, Molernél tett látogatása során felismerte a MATLAB-ban lévő lehetőségeket 1983-ban. Utána nem sokkal csatlakozott hozzá és Steve Bangert-hez, majd újraírták a MATLAB-ot C nyelven és megalapították a The MathWorks-öt 1984-ben. 2000-ben a MATLAB-ot ismét újraírták, hogy újabb módszereket alkalmazzon a mátrixokkal való műveletekre, ebből született a LAPACK csomag. A legutóbbi verziószám ugrás 2012 második félévében történt, amikor megújult a user interface, új motorokat integráltak, illetve MATLAB applikációk készítése vált lehetővé.

A MATLAB az első szoftver, ami univerzálisan alkalmazható a legtöbb kutatási területen, és a mérnöki tervezésben. Gyorsan tanulható nyelve miatt előszeretettel alkalmazzák a lineáris algebra, numerikus analízis oktatásban is. Továbbá kiemelt szerepet kap a numerikus képanalízisben.

A Control System Toolbox egy ipari-standardizált algoritmusokat és alkalmazásokat tartalmaz a rendszeranalízishez. Lehetővé teszi a lineáris rendszerek tervezését, és beállítását. Lehet specifikálni a rendszer átviteli függvényét, állapotter modelljét, zérus-pólus felírást, vagy frekvencia-válasz modelljét. Továbbá az egyes szakaszok, átviteli függvénnyel rendelkező részrendszerek összekapcsolása is megvalósítható elemi kapcsolásokkal (sorba kapcsolás, párhuzamos kapcsolás, és visszacsatolás, illetve általános kapcsolás létrehozása) Az alkalmazáscsomagban fellelhetőek a főbb rendszeranalízishez szükséges függvények, illetve grafikonok. Segítséget nyújt a szabályozáshoz fontos számítások elvégzésében, szabályozók szakaszhoz való tervezésében. Továbbá még számos paraméterének kiszámolására tartalmaz algoritmusokat.

Az alábbi függvényeket használtam a fenti dokumentum elkészítésénél. Az ábrákat legtöbbször egy az egyben másoltam. A függvényeknek pedig az elveit igyekeztem megfogni:

- Transfer Function

```
sys=tf(num,den)
```

Átviteli függvény képzése annak számlálójával, és nevezőjével. A CST saját struktúrát alkot ekkor, amelyben benne van az átviteli függvény, a bementeket, kimeneteket lehet jelölni, valamint lehet külön definiálni a bemeneti, kimeneti, vagy együttes holtidőt. Külön el lehet nevezni, valamint megjegyzéseket fűzni hozzá.

- State-Space model

```
sys=ss(A,B,C,D)
```

Állapotter modell készítése a megfelelő mátrixokkal. Sajátos CST struktúra.

- Continuous to Discrete

```
sys_z=c2d(sys,Ts)
```

Átalakítás diszkrétidőbe megadott mintavételi idővel.

- Step function

```
step(sys)
```

Ugrásválasz számítása

- Impulse function

```
impulse(sys)
```

Impulzusválasz számítása

- Bode diagram számítása

```
bode(sys)
```



```
margin(sys)
```

Előbbi ábrázolja a klasszikus bode diagrammot, második pedig bejelöli az erősítéstartalékot illetve a fázistartalékot

- Nyquist diagram

```
nyquist(sys)
```

- Pole-Zeros Map

```
pzmap(sys)
```

```
rlocus(sys)
```

Előbbi ábrázolja a zérusokat, pólusokat, utóbbi pedig ezek vándorlását a körerősítés függvényében

A CST a következő függvényekkel dolgozik az eredő átviteli függvény meghatározása során. A függvények legtöbbször első lépésben, ha nem átviteli függvényt kapnak, akkor átalakítják azzá, mivel két vektorban lehet tárolni egyszerűbben (számláló, nevező). A későbbiekben pedig többségben vektorműveletekkel számolnak.

- Párhuzamos kapcsolás:

```
parallel(sys1,sys2, ...)
```

Összeadja a két rendszert. Ha negatívan vannak összekötve az egyiket negatív előjellel kell átadni a függvénynek.

- Sorba kapcsolás:

```
series(sys1,sys2, ...)
```

A rendszereket összeszorozza konvolúciós függvény segítségével.

- Negatív visszacsatolás:

```
feedback(sys1,sys2)
```

Megadja a visszacsatolt rendszer eredő függvényét

- Általános összevonás:

```
connect(sys1,sys2, ..., inputs, outputs, connections)
```

A rendszereket miután elneveztük a ki- és bemeneteiket átadhatjuk a connect függvénynek.

Továbbá átadjuk, hogy melyek a teljes rendszer ki és bemenetei, illetve a még nem meghatározott, úgymond levegőben lógó vezetékeket összekötjük összegzőkkel, vagy elágazásokkal (pl.  $y(2)-y(15)+y(6)$ ).