Szabtech. LabZH

# MatLab parancsok

* S = zpk(’s’); Z = zpk(’z’, Ts);
	+ Zéró-pólus alakhoz való megadás
* P = tf(1/L)
	+ Transferfunction – átviteli függvény (Rendszer felvételéhez.)
* P = tf(A,B,C,D)
	+ Állapot teres leírásból átviteli függvény.
* P = tf(NUM, DEN)
	+ Számláló, nevezőből átviteli függvényt csinál.
* [NUM, DEN] = tfdata(P, ’v’);
	+ Számláló, nevező vektorban visszaadva.
* [mag, pha, w] = bode(P);
	+ W-hoz tartozó erősítést és fázisát adja vissza. (Lista)
* [mag, pha, Wcg, Wcp] = margin(P)
	+ Erősítési tartalék, fázistartalék, az az w ahol az erősítési tartalékot méred, az az w ahol a fázistartalékot méred.
* [mag, pha, Wcg, Wcp] = margin(MAG,PHASE,W)
	+ Bode által adott érték alapján számol erősítési- és fázistartalékot.
	+ Konkrét fázistartalékot lehet meghatározni vele, ha PHASE-70 írunk be neki.
* Poles = eig(A)
	+ A mátrix sajátértékei
	+ A rendszer pólusai is egyben.
* [Ad, bd, cd, dd] = canon(A,b,c,d)
	+ A mátrixot kanonikus (diagonális) alakra hozza.
* SYS = ss(A,B,C,d)
	+ Állapot leíró mátrixokból rendszert készít.
* p = step(P)
	+ Egységugrást ad a rendszernek, ábrázolja és visszaadja egy tömbbe, amin később lehet dolgozni. (max, min, stb.)
* [Y,T,X] = step(SYS)
	+ Állapottérben lévő rendszert adsz meg neki és megadja a kimenőjelet, állapotváltozó értékeit.
	+ Y – kimenet, T – idő, X – Állapotváltozók tömbje x(:,1) az első érték x1.
* H = feedback(C\*P, 1)
	+ Egyszeresen visszacsatolt rendszert ad meg. (C\*P / (1 + C\*P))
* K = acker(A,B,P)
	+ K lesz a visszacsatoló vektor. A visszacsatol rendszerben az A mátrix A – B\*K lesz.
	+ A, B a rendszer mátrixai
	+ P a kívánt rendszered pólusai (tfdata(P, ’v’), ennek a DEN-jére hívsz egy roots(DEN) –t, ami a P lesz.)
* R = roots(DEN)
	+ Polinomnak a gyökeit adja vissza egy tömben.
* [R,P,K] = residue(B,A)
	+ Részlet törtekre bont.
	+ R és P egy n elemű tömb, annyi tört lesz.
	+ K sima konstans
* impulse(P)
	+ diracot ad a rendszerre és ábrázolja
	+ hasonlóan paraméterezhető, mint a step
* K = dcgain(SYS)
	+ A rendszer végértékét határozza meg
* SYSD = c2d(SYSC,TS,METHOD)
	+ SYSD – Mintavett rendszer
	+ SYSC – folytonos rendszer
	+ TS – mintavételi idő
	+ METHOD – mindig ’zoh’
	+ Folytonos rendszerből -> diszkrét rendszer
* SYSC = d2c(SYSD,METHOD)
	+ Diszkrét rendszer -> Folytonos rendszer
	+ METHOD – ’zoh’
* nyquist(SYS)
	+ Nyquist-diagramját adja meg a rendszernek
* [RE,IM] = nyquist(SYS,W)
	+ Adott frekvenciára megadja, hogy hol vagy a komplex síkon
* [RE,IM,W] = nyquist(SYS)
	+ Visszaadja egy tömbben a frekvenciát és a frekvenciához tartozó komplex számot
* MSYS = minreal(SYS,TOL)
	+ Sys-ből kiveszi a TOL-nál kisebb együtthatójú tényezőket (illik sűrűn hívogatni)
* [NUM,DEN] = pade(T,N)
	+ Megadja a számlálóját és a nevezőjét egy holtidős tag pade közelítésének
	+ T a holtidő, N a közelités rendje
* [A,B,C,D] = tf2ss(NUM,DEN)
	+ Visszaadja egy átviteli függvényből képzett rendszer állapotteres leírását
* [NUM,DEN] = ss2tf(A,B,C,D,iu)
	+ Visszaadja egy állapotteres leírásból képzett rendszer átviteli függvényét
* lsim(SYS,U,T)
	+ A rendszer válasza adott bemenőjelre T időre (tartomány)
* F = ilaplace(L) (HSZKBAN NINCS)
	+ Inverz laplacol
* Mc = ctrb(A,b)
	+ Irányíthatósági mátrix
	+ A a determinása 0 akkor nem irányítható
* Mo = obsv(A, c)
	+ megfigyelhetőségi mátrix
	+ A a determinása 0 akkor nem megfigyelhető
* Rank(M)
	+ M mátrix rangját adja vissza
* [y,t,x] = initial(ss(A,b,c,d), x0)
	+ Állapottrajekció, ha x0 meg van adva
	+ Pl.: X0 = [a,b,c]
	+ Plot(x(:,1), x(:,2)) // ábrázolás
* plot(X,Y)
	+ Ábrázolja X és Y-t

# Képletek

* PID szabályozó (diszkrét, ideális PD-vel)
	+ Ha nem ideális akkor , ahol T2\* < T2
* PID szabályozó (folytonos, közelítő)
* Z trafó (folytonosból diszkrétbe pólus)
* Youla – parametrizálás
	+ 
	+ G = c2d(P, Ts, ’zoh’);
	+ G = G \* z^(-Td/Ts); //Holtidő
	+ Gm = parazitazérusok / z
	+ Gm = Gm / dcgain(Gm); //1-re álljon be, hogy ne módosítsa a szorzás után
	+ Gp = G/ Gm;
	+ Q = Rn / Gp;
	+ C = Q / (1 – Q \* G)
	+ Y = minreal((Rn/Rr) \* ((C \* G) / (1 + C\*G)))
	+ U = minreal((Rn/Rr) \* (C / (1 + C\*G)))
	+ u = step(U)
	+ umax = max(u)
* Állapot visszacsatolás
	+ Kt = acker(A,B,poles);
	+ A = A-B\*Kt;
	+ Tnew = ss(A, B, C, d);
	+ Kr = 1 / dcgain(Tnew);
	+ Tnew = ss(A, B\*Kr, C, d);
* Differencia egyenlet
	+ 
	+ Oda-vissza kell tudni
* Analitikus kimeneti jel
	+ Y = P \* U;
	+ [num, den] = tfdata(Y, ’v’);
	+ [R, P, K] = residue(num, den)
	+ Y = R / (s - P) // és ezt inverzlaplacolni
* Adott késleltetés mellett fázistartalék és erősítési tartalék
	+ [gm, pm, wg, wc] = margin(L)
	+ Phad = Td \* wc \* 180/pi
	+ Az erősítési tartalék nem változik holtidőre, a fázistartalék pedig, phaD.
	+ Simulinkkel: 
	+ Túllővés a kimenet max-a a beállt jelhez képest, x%os túllövés ideje értelemszerűen
* Állapottrajekció
	+ [y,t,x]= step(ss(A,b,c,d));
	+ Plot(x(:,1), x(:,2))
	+ Ha van kezdő érték, akkor step helyett:
		- x0 = [kezdőértékek felsorolása]
		- [y, t, x] = initial(ss(A,b,c,d),x0);

# NagyZH

* Mintavételezett rendszerek,
	+ e-ados tag úgy mint f esetén aztán behelyettesítem az integrálba és b-vel szorzok így csak egy oszlopvektor lesz amit elemenként integrálok és kész.
	+ Ha A invertálható akkor g képlete lehet az is hogy
* Folytonos megfigyelhetőségi mátrix előállítása átviteli fgv-ből (Átviteli fgv alakja ), ahol a1..an az A együtthatói és b1…bn a B együtthatói. ahol a1 a legnagyobb kitevőjű tag együtthatója .) Ezért fontos is hogy ha ezt felírjuk a polionok a megfelelő alakban legyenek:
* csak ott nem 1-etől kezdjük hanem b1\*-től



* Folytonos irányíthatósági mátrix előállítása átviteli fgv-ből (ez diszkrét esetben tükröződik valamiért)



* Holtidős integráló maradékrendszer:
	+ Itt az a lényeg hogy a PID után az L-be csak egy marad tehát csak holtidő és integráló meg valami konstans, ilyenkor igaz az alábbi egyenlet
	+ mert integráló miatt –pi/2-be indul és még a késleltető fázistolását kell hozzáadni és ez egyenlő a fázistöbblettel majd
	+ mert |L|-ben -ben 1-nek kell lennie.
	+ ha a fenti képletbe Ti-t kérdezik az ugye éppen 1/K-val egyenlő. Lehet kérdés ez is akkor K-t kiszámolod és abból megvan a Ti.
	+ P átalakítása G-vé
	+ 
* Adott szakasz átviteli függvény esetén PI szabályozó:
	+ P-nek a legnagyobb időállandóját vesszük, ebből csinálunk egy PI-t a megadott képlet alapján.
	+ Ezek után a Kc meghatározása adott bemeneti jelnek a kezdeti értéke u(0) = X
	+ U(0) = Kc = X
	+ A szabályozó átvitele 0-ban az egyenlő Kc-vel, mert ugye lim s-> esetén az egész tart 1-be! Így Kc \* 1 = X kell.
* Bode-diagram
	+ Megnézzük a felnyitott szab. Kör pólusait és zérusait. Minden pólusnál -20-at tör és minden zérusnál +20 tör. A vágási körfrekvencia kiszámolásához abs(L) = 1 –et kell vegyük. Ilyenkor általában a törések hatása messzebb van, mint a körfrekvencia ezért azokat a pólusokat vagy zérusokat nem kell figyelembe venni, így általában 1 = X/s jön ki amit könnyű kiszámolni.
	+ 
	+ Fázistartalék számításához elsőnek vesszük a pólusokat, amikben átírjuk az s-ket jw-ra, és ábrázoljuk őket a komplex számsíkon, majd nézzük az x tengellyel bezárt szögét, ,ahhoz hogy ezt megkapjuk arctg-ét kell venni.
	+ Ft = (180 – ((integrátorok száma) \* 90 + (pólusok száma) \* arctg(…) – (zérusok száma) \* arctg(…))) – W\*Td\*180/PI
* Olyan feladatnál aminél a mátrixok vannak megadva és állapotvisszacsatolással szabályozunk és van egy tervezési polinom megadva a feladat pedig a kT meghatározása, úgy kell dolgozni, hogy a karakterisztikus egyenlet = tervezési polinom
	+ a tervezési polinom meg lehet adva úgy is hogy konkrétan mi legyen a pólus ebből simán vissza kell írni
* Olyan feladatnál amikor átviteled van megadva ahol az egyik pólus labilis, és tükrözni kell hogy megkapd a tervezési polinomt úgy kell eljárni, hogy felírod a zárojeleket felbontva a két polionomt ekkor lesz egy (folyamat átviteli fgv-ének a nevezője felbontva), ekkor a megoldás vagyis kT úgy jön ki hogy kT = [r1-a1 r2-a2] és kész is.
* Predikciós szabályzós feladatnál, azt kell csinálni, hogy G(z) = és két polinomot keresünk F-et és P-t amik úgy állnak elő hogy:
	+ ahol F-et és P-t a filteres alakba keressük ami azt jelenti hogy (ezt onnan lehet látni hogy a z^-d amúgy is ilyen alakban van benne.
	+ F fokszáma: d-1 (d a holtidő vagy megadják hány lépés alatt álljon be)
	+ P fokszáma pedig n-1 ahol n a folyamat fokszáma
	+ Ha ezeket felírtuk akkor meg kell oldani a diophantoszi egyenletet és megkapjuk a polinomok együthatóit.
	+ A szabályzó úgy áll elő hogy (akár meg is van adva a feladatba hogy milyen alakba várjuk)
* Véges beállású szabályozó
	+ Itt a teljes rendszer egyenletét úgy keressük hogy
	+ Ebből a szabályozó képlete: ha a lengéseket kerülni akarjuk.
	+ Ha lengéseket megengedünk, akkor:

# Elméleti kérdések

* Mi a z-transzformáció? Hova képzi le az s komplex s = -3 pontját? Hogyan deifiniáljuk egy jel z-transzofrmáltját?
	+ 
	+ 1-ről indulva veszed a hatványokat. 1; 0.6 \* z^-1; 0.36 \* z^-2; …
	+ Y-ban a 0.6 hatványai vannak benn i = 0-tól indulva
	+ Elvileg Geometria sor felbontása
* Impulzus átviteli függvény deifiníciója:
	+ v: átmeneti fg.
		- Karakterisztikus egyenlet:
		- 0 < K Ts < 2
		- Abszolút érték miatt egy egység kört nézel, ami miatt K Ts nem lehet nagyobb 2-nél, illetve kisebb mint 0.
* Szakasz állapotegyenlete irányíthatósági kanonikus alakban
	+
	+ y(z) = v(z) ()
	+ 
* Laplace-trafó
	+ Lineáris
	+ 
	+ 
	+ 
* Inverz Laplace-trafó
	+ dt