

6. Gyakorlat

6. Tantermi gyakorlat – Diszkrétidejű szabályozások tervezése állapotterben

A gyakorlat célja, hogy bemutassuk a diszkrétidejű (mintavételes) szabályozók állapotteres tervezésének módszereit: pólus (sajátérték) áthelyezés állapotvisszacsatolással; állapotmegfigyelő tervezése; az alapjel figyelembevétele; integráló hatás vagy terhelésbecslő beépítése. A gyakorlat során csak egybemenetű és egykimenetű (SISO – Single Input, Single Output) rendszerekkel foglalkozunk, a módszerek ugyanakkor egyszerűen kiterjeszthetők több bemenet és kimenet tartalmazó rendszerek kompenzálásához. MIMO rendszerek esetén egyedül az Ackermann-képlet helyett kell MIMO rendszerekre is alkalmazható más technikát (place, LQ optimalizálás stb.) választani a pólusáthelyezéshez, vagy az arra visszavezethető feladathoz.

Diszkrétidejű szabályozások tervezése állapotterben

A folytonosidejű és diszkrétidejű időinvariáns lineáris rendszerek állapotterbeli elveken alapuló irányítását a közöttük fennálló algebrai hasonlóság miatt hasonló módszerekkel lehet megtervezni, amint azt az 5. gyakorlat során bemutattuk. A folytonosidejű szabályozott szakasz állapotegyenletének alakja legyen a szokásos $\dot{x} = Ax + Bu$, $y = Cx$, és legyen a szakasz előtt digitális/analóg átalakító (DAC), amely nulladrendű tartószervvel modellezhető, utána pedig álljon analóg/digitális átalakító (ADC). Mivel ilyenkor a szakasz bemenetén lépcsős analóg jel van, ezért az eredő mintavételes rendszert a következő alakra lehet hozni:

$$\begin{aligned}x_{i+1} &= \Phi x_i + \Gamma u_i, \\ y_i &= C x_i.\end{aligned}\tag{6.1}$$

Az irányíthatósági és megfigyelhetőségi mátrix a folytonosidejű rendszerekhez hasonlóan:

$$M_c = [\Gamma \quad \Phi\Gamma \quad \dots \quad \Phi^{n-1}\Gamma]\tag{6.2}$$

$$M_o = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix}.\tag{6.3}$$

Megjegyezzük azonban, hogy míg folytonos időben a teljes irányíthatóság (nulla állapotba) és teljes elérhetőség (nulla állapotból) szükséges és elégséges feltétele egyformán $\text{rank } M_c = n = \dim x$, addig diszkrét időben az (analóg) rang feltétel a teljes elérhetőségnek szükséges és elégséges feltétele, míg a teljes irányíthatóságnak csak elégséges feltétele (azaz van olyan rendszer, amely teljesen irányítható, de M_c nem maximális rangú).

Hasonlóan diszkrét időben $\text{rank } M_o = n = \dim x$ a teljes megfigyelhetőségnek (jövőbeli bemeneti-kimeneti adatokból) szükséges és elégséges feltétele, míg a teljes rekonstruálhatóságnak (múltbeli bemeneti-kimeneti adatokból) csak elégséges feltétele.

Ennek oka az, hogy míg folytonos időben az állapotegyenlet (lokálisan) csökkenő és növekvő idő irányban is egyértelműen megoldható, addig a diszkrétidejű állapotegyenlet csak akkor oldható meg csökkenő idő irányában, ha a Φ mátrix invertálható (azaz $\det \Phi \neq 0$, vagy más megfogalmazásban $z=0$ nem sajátérték, pl. a rendszer nem holtidős). Az elégségesség miatt szerencsére a rangfeltételek teljesülésekor nincs probléma az irányíthatósággal és a rekonstruálhatósággal sem. Ha a Φ mátrix invertálható, akkor a diszkrétidejű rendszert reverzibilisnek nevezzük. Reverzibilis rendszer esetén a rangfeltételek az irányíthatóságnak illetve a rekonstruálhatóságnak szükséges feltételei is.

A továbbiakban feltesszük, hogy a mintavételes rendszer teljesen elérhető (irányítható) és teljesen megfigyelhető, vagyis $\text{rank } M_c = \text{rank } M_o = \dim x = n$.

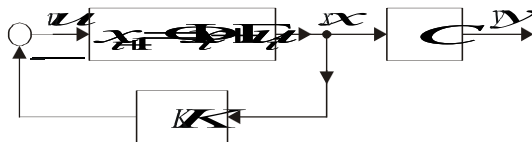
Pólusát helyezés állapot-visszacsatolással

A 6.1. ábrának megfelelően alkalmazzunk

$$u_i = -Kx_i \quad (6.4)$$

állapot-visszacsatolást, akkor a zárt rendszer állapotegyenlete és karakterisztikus egyenlete rendre a következő lesz:

$$\begin{aligned} x_{i+1} &= (\Phi - \Gamma K) x_i, \\ \varphi_c(z) &= \det(zI - (\Phi - \Gamma K)). \end{aligned} \quad (6.5)$$



6.1. ábra. Állapot-visszacsatolás diszkrét időben

Ha előírjuk a zárt rendszer karakterisztikus egyenletét, és keressük az ehhez megfelelő K állapot-visszacsatolást, akkor a folytonosidejű feladathoz algebrailag hasonló feladatra jutunk, és ezért K meghatározható a folytonosidejű rendszerekre kidolgozott módszerekkel. Egyváltozós (SISO) rendszereknél a pólusáthelyezési feladat megoldása meghatározható az Ackermann-képlettel:

$$K = (0 \dots 0 \ 1) M_c^{-1} \varphi_c(\Phi), \quad (6.6)$$

$$(\Phi, \Gamma) \xrightarrow{M_c} K. \quad (6.7)$$

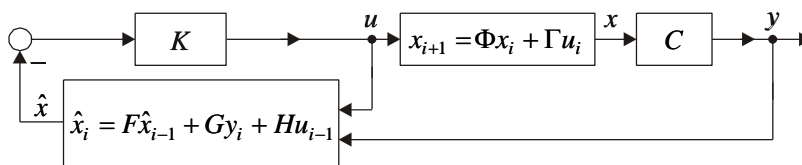
Ha a kívánt sajátértékek vektora $[\lambda_1 \ \lambda_2 \ \dots \ \lambda_n]^T$, akkor az $\Phi - \Gamma K$ mátrix karakterisztikus egyenlete $\varphi_c(z) = \prod_{i=1}^n (z - \lambda_i)$.

Az Ackermann-képletben szerepel az M_c mátrix inverze, amely SISO esetben kvadratikusan, és a feltevés szerint maximális rangú. A Φ mátrix szintén kvadratikusan, ezért hatványozható és Φ^i behelyettesíthető a z^i hatványok helyére a zárt rendszer $\varphi_c(z)$ karakterisztikus polinomjában. A Control System Toolbox-ban (CST) rendelkezésünkre áll az `acker` utasítás is K meghatározására, amint azt az 5. gyakorlat keretében már láttuk.

Ne felejtjük azonban el, hogy diszkrét időben a stabilitáshoz a sajátértékeknek az egységkör belsejében kell lenniük, továbbá a gyors sajátértékek $z=0 = e^{-\infty T}$ közelében vannak. Másrészt viszont a rendszer gyorsítása a beavatkozó jel növekedésével jár, amely telítést okozhat a szabályozót követő beavatkozó szervben vagy a DAC-ban (mivel a bitszám limitált). Nem szabad arról sem megfeledkezni, hogy a Shannon-tételnek nemcsak a szakaszra, hanem a felgyorsított rendszer minden jelére, így a szabályozóra (állapot-visszacsatolás, megfigyelő) is teljesülni kell, ezért a gyorsításnak és a mintavételi időnek is összhangban kell lennie.

Állapot-visszacsatolás realizálása aktuális megfigyelővel

A rendszerint nem mérhető x állapotváltozó becslésére diszkrétidejű rendszereknél is állapotmegfigyelőt célszerű alkalmazni. Diszkrét időben aktuális állapotmegfigyelőt célszerű alkalmazni, lásd 6.2. ábra.



6.2. ábra. Aktuális állapotmegfigyelő diszkrét időben

Feltesszük, hogy $(\Phi, C\Phi)$ megfigyelhető, pl. $\exists \Phi^{-1}$ és (Φ, C) megfigyelhető. Akkor választható az

$$\hat{x}_i = F\hat{x}_{i-1} + Gy_i + Hu_{i-1} \quad (6.8)$$

aktuális állapotmegfigyelő. Jelölje $\tilde{x}_i = x_i - \hat{x}_i$ az állapotbecslés hibáját, akkor

$$\tilde{x}_i = F(x_{i-1} - \hat{x}_{i-1}) + (\Gamma - GCF - H)u_{i-1} + (\Phi - GC\Phi - F)x_{i-1}, \quad (6.9)$$

ezért aszimptotikus állapotmegfigyelőhöz jutunk, ha $\tilde{x}_i \rightarrow 0$ biztosítása érdekében a következő választással élünk:

$$F = \Phi - GC\Phi, \quad (6.10a)$$

$$H = \Gamma - GCF, \quad (6.10b)$$

$$\tilde{x}_i = F\tilde{x}_{i-1} \text{ stabil és gyors.} \quad (6.10c)$$

Ha a megfigyelő tranziensének gyorsaságát a megfigyelő $\varphi_o(z)$ karakterisztikus egyenletével írjuk elő, ahol

$$\varphi_o(z) = \det(zI - F) = \det(zI - (\Phi - GC\Phi)),$$

akkor G és F meghatározható

$$(\Phi, C)_I \leftrightarrow (\Phi^T, \Phi^T C^T)_{II} \xrightarrow[M_{c,II}]{\varphi_o(z)} K_{II} \rightarrow G = K_{II}^T \rightarrow F = \Phi - GC\Phi \quad (6.11)$$

alapján, SISO rendszer esetén pedig az Ackermann-képlettel (vagy más ekvivalens módszerrel). Végül G ismeretében számítható H is (6.10b) alapján.

Az \hat{x}_i becsült állapot számítása valós idejű megvalósítás szempontjából kedvezőbb alakra is hozható a következő átalakítással:

$$\begin{aligned} \hat{x}_i &= (\Phi - GC\Phi)\hat{x}_{i-1} + Gy_i + (\Gamma - GCF)u_{i-1} = \\ &= \Phi\hat{x}_{i-1} + \Gamma u_{i-1} + G\{y_i - C(\Phi\hat{x}_{i-1} + \Gamma u_{i-1})\}. \end{aligned} \quad (6.12)$$

Jól látható, hogy a számítások közül $\Phi\hat{x}_{i-1} + \Gamma u_{i-1}$ az utolsó mintavételnél azonnal indítható, míg Gy_i csak a következő mintavételnél, miután y_i mérése már megtörtént. Ez a processzor idő jobb kihasználását eredményezi, különösen, ha $n = \dim x$ nagy. Az aktuális megfigyelő \tilde{x}_i bevezetésével (amelynek számítása két mintavétel közötti időben is elvégezhető) a következő alakra hozható:

$$\begin{aligned}\bar{x}_i &= \Phi \hat{x}_{i-1} + \Gamma u_{i-1} && \text{"time-update"} \\ \hat{x}_i &= \bar{x}_i + G(y_i - C \bar{x}_i) && \text{"measurement-update"}\end{aligned}\quad (6.13)$$

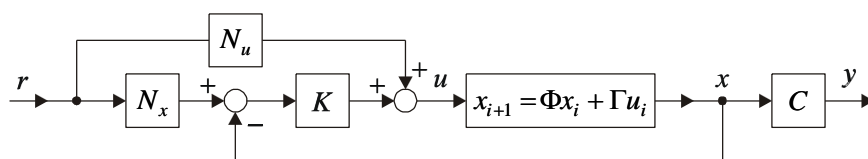
Ez a felbontás más állapotbecslési struktúráknál is tipikus, így a sztochasztikus elven alapuló Kalman-szűrőnél is. Vegyük észre, hogy $C \bar{x}_i$ azt mutatja, hol várjuk a kimenetet a régi megfigyelések alapján.

Egy rokon területen, a mobilis ágensek navigációjánál is felhasználható ez az elv. Ha az előző ütemben már azonosítottuk, melyik ágens hol van, és a mozgásuk mellett a következő mérésnél meg tudjuk állapítani a navigációs rendszer érzékelőinek jeleiből, hogy hol vannak ágensek, de nem tudjuk azonosítani, hogy melyik új helyzet melyik ágenshez tartozik, mert az ágensek szimultán mozognak és keresztezhetik egymás pályáját, akkor $\hat{y}_i := C \bar{x}_i$ segíthet abban, hogy becslést adjon arra, minek a környezetében kell keresni az előző ütemben megtalált ágens a sok új helyzet között. Ez a koncepció repülő objektumok követésénél is alkalmazható. A rendszernek ilyenkor megfelel az objektumok/ágensek kinematikai (sebesség és szögsebesség) modellje. Ez a probléma azonban nemlineáris, ezért lokálisan linearizálni kell az utolsó \hat{x}_{i-1} körül. Erre az elvre épül a kiterjesztett Kalman-szűrő.

Az alapjel miatti korrekció diszkrét időben

Az alapjel figyelembevételére a folytonos időben megismert módszer analógiájára a 6.3. ábra szerinti megoldás választható. Bár az alapjel miatti korrekció elvei ugyanazok lesznek, tekintettel kell lenni arra, hogy az állapotegyenlet megoldását állandósult állapotban az jellemzi, hogy a régi és az új állapot azonos:

$$x_\infty = \Phi x_\infty + \Gamma u_\infty. \quad (6.14)$$



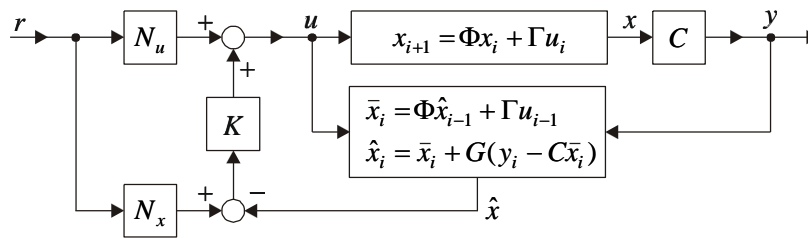
6.3. ábra. Alapjel miatti korrekció diszkrét időben

Most is feltesszük, hogy az alapjel konstans (pontosabban ritkán változik meg és utána hosszabb ideig állandó marad). Úgy döntünk, hogy állandósult állapotban az $N_x r - x_\infty$ különbségi jel legyen nulla, a kimenet állandósult állapotbeli értéke pedig legyen hiba nélküli, tehát $y_\infty = r$. Az ehhez szükséges bemenőjelet állandósult állapotban $u_\infty = N_u r$ fogja szolgáltatni. Legyen $\dim y = \dim r = \dim u = m$, akkor

N_x egy $n \times m$, N_u pedig egy $m \times m$ méretű mátrix (SISO rendszer esetén nyilván $m=1$). A folytonosidejű esetben N_x és N_u számítására megismert módszert kismértékben korrigálni kell, mivel diszkrét időben az állandósult állapotra vonatkozó matematikai összefüggés más:

$$\begin{aligned} N_x r = x_\infty &\Rightarrow y_\infty = C x_\infty = C N_x r = r \Rightarrow C N_x = I_m, \\ N_u r = u_\infty &\Rightarrow \Phi x_\infty + \Gamma u_\infty = x_\infty \Rightarrow [(\Phi - I) N_x + \Gamma N_u] r = 0 \Rightarrow \\ &(\Phi - I) N_x + \Gamma N_u = 0_{n \times m} \\ \begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix} \begin{pmatrix} N_x \\ N_u \end{pmatrix} &= \begin{pmatrix} 0_{n \times m} \\ I_m \end{pmatrix} \Rightarrow \begin{pmatrix} N_x \\ N_u \end{pmatrix} = \begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0_{n \times m} \\ I_m \end{pmatrix}. \end{aligned} \quad (6.15)$$

A 6.4. ábra állapot-visszacsatolásból (ÁV), állapotmegfigyelőből (ÁM) és alapjel miatti korrekcióból (N_x, N_u) felépülő egyszerű szabályozás megvalósításának elvét mutatja be, figyelembe véve az állapotmegfigyelő valós idejű realizálásának szempontjait is. A szabályozó bemenete r és y , kimenete pedig u .



6.4. ábra. Állapot-visszacsatolást és aktuális megfigyelőt alkalmazó egyszerű szabályozás megvalósításának elve diszkrét időben

Integráló szabályozás

A zavarás hatásának csökkentésére célszerű a szabályozóban integrátort elhelyezni. Egészítsük ki ezért a rendszert az integrátor $x_I = \int y dt$ jelével, és alkalmazzuk számítására a bal oldali téglalapszabályt (LSR):

$$x_{I,i+1} = x_{I,i} + T y_i = x_{I,i} + T C x_i \quad (6.16)$$

Az állapotegyenlet az $\tilde{x} = (x^T, x_I^T)^T$ bővített állapotváltozó bevezetése után a következő lesz:

$$\begin{pmatrix} x_{i+1} \\ x_{I,i+1} \end{pmatrix} = \begin{bmatrix} \Phi & 0 \\ TC & I \end{bmatrix} \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} + \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} u_i \Rightarrow \tilde{x}_{i+1} = \tilde{\Phi} \tilde{x}_i + \tilde{\Gamma} u_i, \quad (6.17)$$

$$y_i = [C \quad 0] \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} \Rightarrow y_i = \tilde{C} \tilde{x}_i.$$

Az integrátor jelével bővített állapot-visszacsatolás ekkor

$$u_i = -\tilde{K} \tilde{x} = -[K \quad K_I] \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} = -K x_i - K_I x_{I,i}, \quad (6.18)$$

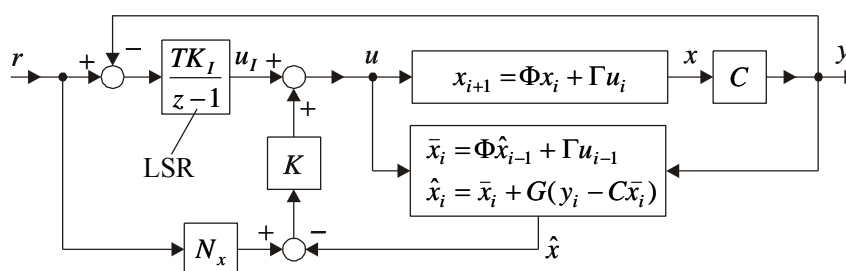
a zárt rendszer állapotegyenlete pedig

$$\tilde{x}_{i+1} = (\tilde{\Phi} - \tilde{\Gamma} \tilde{K}) \tilde{x}_i. \quad (6.19)$$

Ha előírjuk a zárt rendszer $\tilde{\varphi}_c(z) = \det(zI - (\tilde{\Phi} - \tilde{\Gamma} \tilde{K}))$ karakterisztikus egyenletét, akkor a pólusát helyezési feladat megoldásához szükséges $\tilde{K} = [K \quad K_I]$ állapot-visszacsatolás meghatározható a

$$(\tilde{\Phi}, \tilde{\Gamma}) \xrightarrow[\tilde{M}_c]{\tilde{\varphi}_c(z)} \tilde{K}. \quad (6.20)$$

séma alapján, egyváltozós (SISO) rendszereknél pl. az Ackermann-képlettel.



6.5. ábra. Integráló szabályozás megvalósításának elve diszkrét időben

Az integráló szabályozás megvalósítását a 6.5. ábrán mutatjuk be. A realizációnál felhasználjuk, hogy az integrátor állandósult állapotban nulla

bemenetnél is tud nemnulla kimenetet biztosítani, ezért $N_u r$ hatását is realizálja. Az integrátort a jobb oldali téglalapszabállyal (RSR) is implementálhatjuk.

A tervezés lépései a következők:

$$\begin{aligned}(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}) &\rightarrow \tilde{K} = [K \ K_1] \quad (\text{ÁV}) \\(\Phi, \Gamma, C) &\rightarrow (F, G, H) \quad (\text{ÁM}) \\(\Phi, \Gamma, C) &\rightarrow (N_x, N_u)\end{aligned} \quad (6.21)$$

Terhelésbecslés

A zavarást a szakasz bemenetére redukálnak képzeljük ("load change"), hasonlóan az 5. gyakorlathoz. Ha a zavarás jellegéről ismereteink vannak, akkor modellezhető. A továbbiakban feltesszük, hogy a zavarás konstans, ezért differenciálegyenlete

$$\dot{d} = 0 \quad (d \text{ konstans, értéke ismeretlen}) \Rightarrow d_{i+1} = d_i. \quad (6.22)$$

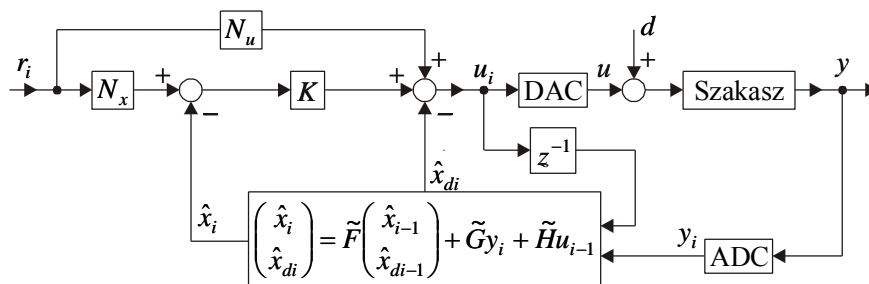
Ha a rendszert bővítjük az $x_d = d$ állapotváltozóval és bevezetjük az $\tilde{x} = (x^T, x_d^T)^T$ jelölést, akkor az állapotegyenlet a következő lesz:

$$\begin{aligned}\begin{pmatrix} x \\ x_d \end{pmatrix}_{i+1} &= \begin{bmatrix} \Phi & \Gamma \\ 0 & I \end{bmatrix} \begin{pmatrix} x \\ x_d \end{pmatrix}_i + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u_i \Rightarrow \tilde{x}_{i+1} = \tilde{\Phi} \tilde{x}_i + \tilde{\Gamma} u_i, \\ y_i &= [C \ 0] \begin{pmatrix} x \\ x_d \end{pmatrix}_i \Rightarrow y_i = \tilde{C} \tilde{x}_i.\end{aligned} \quad (6.23)$$

Mivel az $(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C})$ rendszer nem irányítható (az x_d kívülről érkező jelet nem lehet belülről u -val irányítani), ezért az állapot-visszacsatolást és az alapjel miatti korrekciót az eredeti rendszerhez kell meghatározni. A megfigyelőt a bővített rendszerhez kell megtervezni, hogy biztosítani tudja mind az \hat{x} , mind pedig az \hat{x}_d becslést a szabályozó számára:

$$\begin{aligned}(\Phi, \Gamma, C) &\rightarrow K \quad (\text{ÁV}) \\(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}) &\rightarrow (\tilde{F}, \tilde{G}, \tilde{H}) \quad (\text{ÁM}) \\(\Phi, \Gamma, C) &\rightarrow (N_x, N_u)\end{aligned} \quad (6.24)$$

A terhelésbecslést is tartalmazó szabályozás megvalósításának elvét a 6.6. ábra mutatja be.



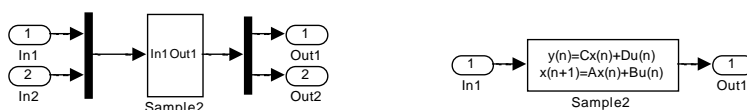
6.6. ábra. Terhelésbecslést alkalmazó szabályozás megvalósításának elve diszkrét időben

A szabályozók megvalósítása

A továbbiakban egy harmonikus oszcillátor (csillapítatlan lengőtag) szabályozása kapcsán két módszert mutatunk be a szabályozó realizálására.

1. módszer: Szabályozó realizálás a megtervezett komponensek összekötésével

A szabályozók egy lehetséges megvalósítását a három megvizsgált üzemmódra a 6.4. ábra (normál állapot-visszacsatolás megfigyelővel és alapjel-korrekcióval), a 6.5. ábra (integráló állapot-visszacsatolás a hibajel integrálásával, megfigyelővel és alapjel-korrekcióval) és a 6.6. ábra (állapot-visszacsatolás állapotmegfigyelővel, terhelésbecslővel és alapjel-korrekcióval) mutatja be. A 6.4 és 6.5 ábrákat ki kell egészíteni az $u_{i-1} = z^{-1}u_i$ taggal (Unit Delay a mintavételi idő megadásával). A folytonosidejű $r(t)$ és $y(t)$ jelek korrekt mintavételezését meg lehet oldani egy elfajuló diszkrétidejű állapotegyenlettel, lásd 6.7. ábra, ahol $A = B = C = []$ és $D = eye(2)$.



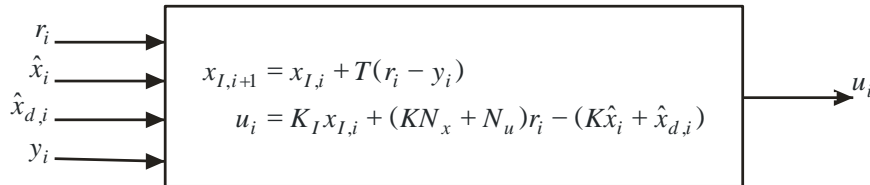
6.7. ábra. Folytonosidejű jelek mintavételezése

Az 1. módszert normál állapot-visszacsatolás és integráló állapot-visszacsatolás esetén fogjuk bemutatni.

2. módszer: Állapotteres realizálás összevont megfigyelővel és szabályozóval

A szabályozó kimenőjele $u_i = K(N_x r_i - \hat{x}_i) + N_u r_i + K_I x_{I,i} - \hat{x}_{d,i}$ alapján képezhető az általános esetben, ahol $x_{I,i+1} = x_{I,i} + T(r_i - x_i)$ része a szabályozó dinamikájának.

Hatásvázlatát a 6.8. ábra mutatja, ahol x_I a szabályozó belső állapota, becslésére nincs szükség.



6.8. ábra. Az integráló állapot-visszacsatolás hatásvázlata szeparált megfigyelő esetén

A megfigyelő belső állapotát jelölje $\bar{X} = (\bar{x}^T, \bar{x}_d^T)^T$, akkor kihasználva azt, hogy a Simulink előbb képezi a kimenőjelet és csak utána a következő állapotot, az S és S_d szelektormátrixok bevezetésével írhatjuk, hogy

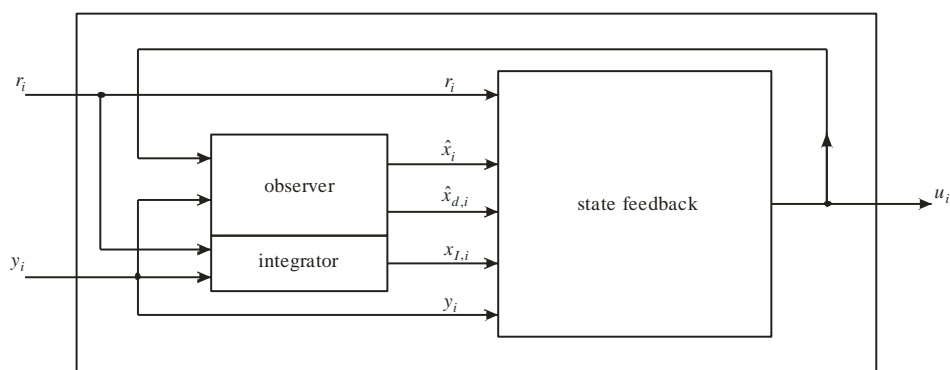
$$\begin{aligned}
 \hat{X}_i &= \tilde{F}\hat{X}_{i-1} + \tilde{G}y_i + \tilde{H}u_{i-1} = \\
 &= (\tilde{\Phi} - \tilde{G}\tilde{C}\tilde{\Phi})\hat{X}_{i-1} + \tilde{G}y_i + (\tilde{\Gamma} - \tilde{G}\tilde{C}\tilde{\Gamma})u_{i-1} = \\
 &= \tilde{\Phi}\hat{X}_{i-1} + \tilde{\Gamma}u_{i-1} + \tilde{G}\{y_i - \tilde{C}(\tilde{\Phi}\hat{X}_{i-1} + \tilde{\Gamma}u_{i-1})\} \\
 \bar{X}_i &= \tilde{\Phi}\hat{X}_{i-1} + \tilde{\Gamma}u_{i-1} \\
 \hat{X}_i &= \bar{X}_i + \tilde{G}\{y_i - \tilde{C}\bar{X}_i\},
 \end{aligned} \tag{6.25}$$

ahonnan a megfigyelő a Matlab-ban megszokott standard alakra hozható:

$$\begin{aligned}
 \bar{X}_{i+1} &= \tilde{\Phi}\hat{X}_i + \tilde{\Gamma}u_i = \tilde{\Phi}\{\bar{X}_i + \tilde{G}[y_i - \tilde{C}\bar{X}_i]\} + \tilde{\Gamma}u_i = \\
 &= (\tilde{\Phi} - \tilde{\Phi}\tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \tilde{\Gamma}u_i = \\
 &= \tilde{\Phi}(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \tilde{\Gamma}u_i \\
 \hat{X}_i &= (I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{G}y_i \\
 \hat{x}_i &= S\hat{X}_i = S(I - \tilde{G}\tilde{C})\bar{X}_i + S\tilde{G}y_i \\
 \hat{x}_{d,i} &= S_d\hat{X}_i = S_d(I - \tilde{G}\tilde{C})\bar{X}_i + S_d\tilde{G}y_i
 \end{aligned} \tag{6.26}$$

A (6.26) egyenletben az első egyenlet az állapotegyenlet, az utolsó három pedig a kimeneti leképezés. Az állapotváltozó \bar{X} , \hat{X} pedig kimenőjel. Jól látható továbbá, hogy a Simulink adottságainál fogva az állapotteres megoldás egyszerre kéri bemenetként u_i értékét, miközben u_i számításához szükség van \hat{x}_i , $\hat{x}_{d,i}$ értékére.

A keletkezett algebrai hurkot a Simulink általában nem tudja numerikusan korrektül feloldani, mitöbb a Simulink program beágyazott processzorra történő kihozatala ilyen esetben nem is lenne lehetséges. Ezért azt a megoldást választjuk, hogy a megfigyelőt és az állapot-visszacsatolást algebrailag összevonjuk, azaz \hat{x}_i , $\hat{x}_{d,i}$ mellett u_i is a közös szabályozó állapotegyenlet kimenete lesz. A szabályozó algebrai hurkot tartalmazó elvi hatásvázlatát a 6.9. ábra mutatja.



6.9. ábra. Az összevont megfigyelő és az állapot-visszacsatolás elvi hatásvázlata

A korábbi \bar{X} állapothoz hozzávehetjük a szabályozó x_I integrátorát: $\bar{X}_1 := (\bar{X}^T, x_I^T)^T$. Ezen kívül \bar{X} állapotegyenletében az algebrai hurok feloldására u_i helyére behelyettesíthető

$$\begin{aligned} u_i &= K(N_x r_i - \hat{x}_i) + N_u r_i + K_I x_{I,i} - \hat{x}_{d,i} = \\ &= (KN_x + N_u)r_i - (KS + S_d)\{(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{G}y_i\} + K_I x_{I,i} = \\ &= -(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i, \end{aligned} \quad (6.27)$$

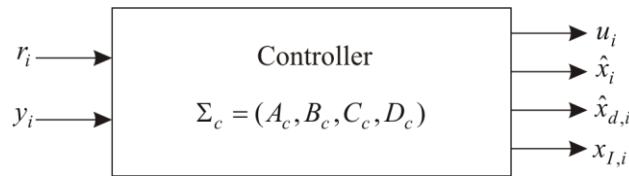
ahonnan kapjuk, hogy

$$\begin{aligned} \bar{X}_{i+1} &= \tilde{\Phi}(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \\ &+ \tilde{\Gamma}\{-(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i\} \Rightarrow \end{aligned}$$

$$\begin{aligned}
\bar{X}_{i+1} &= [\tilde{\Phi} - \tilde{T}(KS + S_d)](I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{T}K_I x_{I,i} + \\
&+ \tilde{T}(KN_x + N_u)r_i + [\tilde{\Phi} - \tilde{T}(KS + S_d)]\tilde{G}y_i = \\
&= A_{c11}\bar{X}_i + A_{c12}x_{I,i} + B_{c11}r_i + B_{c12}y_i \\
x_{I,i+1} &= x_{I,i} + T(r_i - y_i) = x_{I,i} + Tr_i - Ty_i = \\
&= A_{c22}x_{I,i} + B_{c21}r_i + B_{c22}y_i
\end{aligned} \tag{6.28}$$

$$\begin{aligned}
u_i &= -(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i = \\
&= C_{c11}\bar{X}_i + C_{c12}x_{I,i} + D_{c11}r_i + D_{c12}y_i \\
\hat{x}_i &= S(I - \tilde{G}\tilde{C})\bar{X}_i + S\tilde{G}y_i = C_{c21}\bar{X}_i + D_{c22}y_i \\
\hat{x}_{d,i} &= S_d(I - \tilde{G}\tilde{C})\bar{X}_i + S_d\tilde{G}y_i = C_{c31}\bar{X}_i + D_{c32}y_i \\
x_{I,i} &= x_{I,i} = C_{c4,2}x_{I,i}
\end{aligned} \tag{6.29}$$

Az állapotmegfigyelő és az állapot-visszacsatolás összevonása után keletkezett kompakt szabályozót a 6.10. ábrán mutatjuk be. A szabályozó egy diszkrétidejű dinamikus rendszer, amelynek bemenőjelei r_i , y_i , kimenőjelei pedig u_i és a becült állapotok, szükség esetén kiegészítve a hibajel integráljával (ekkor N_u hiányzik).



6.10. ábra. Az összevont kompakt szabályozó

A 2. módszert a terhelésbecslőt alkalmazó szabályozás esetén mutatjuk be.

A harmonikus oszcillátor diszkrétidejű szabályozása

Normál szabályozás

Az alábbi kód lefutása után a szabályozó következő paramétereit kaptuk (a kódból a rendszer, a specifikációk, a mintavételi idő stb. azonosítható, ezért ezeket nem részletezzük):

```
%Oscillator_Control.m
```

```
clear all
close all
```

```

clc

%System
w0=100;
W=tf(w0^2,[1 0 w0^2]);
sys=ss(W);
Ts=0.001;
sysd=c2d(sys,Ts);    %'zoh'
[Phi,Gamma,C,D]=ssdata(sysd)

%Specifications
xi=0.7;
s1=-xi*w0+j*sqrt(1-xi^2)*w0;
scinf=3*real(s1);
soinf=-5*w0;
wN=pi/Ts;
z1=exp(s1*Ts);
zcinf=exp(scinf*Ts);
zoinf=exp(soinf*Ts);

%-----
%Normal control
%-----
fprintf('Normal Control\n')
%Controller
pcl=[z1 conj(z1)]';
K=acker(Phi,Gamma,pcl)
%Actual observer
pobs=[zoinf zoinf]';
KII=acker(Phi',Phi'*C',pobs);
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma
%Reference signal correction
Nxu=inv([Phi-eye(2) Gamma; C 0])*[0 0 1]';
Nx=Nxu(1:2);
Nu=Nxu(3);
%Simulation
open_system('OscNormalCl')
fprintf('OscNormalCl.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
pause

```

A megtervezett szabályozó paramétere:

```

Phi =
    0.9950    -0.0780
    0.1278     0.9950

```

```
Gamma =  
    0.0080  
    0.0005
```

```
C =  
  
    0    9.7656
```

```
D =  
    0
```

```
Normal Control
```

```
K =  
    16.3157   -0.6528
```

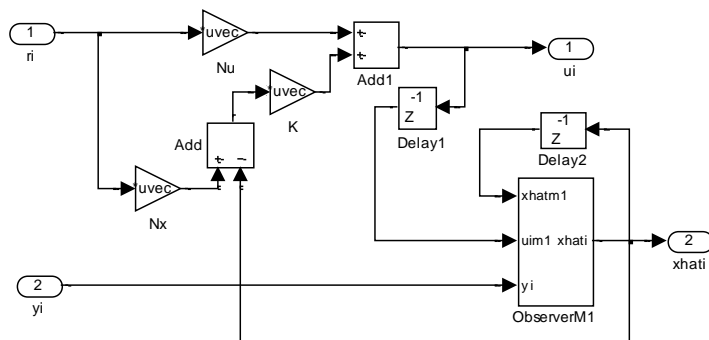
```
G =  
    0.1186  
    0.0647
```

```
F =  
    0.8470   -1.2303  
    0.0470    0.3660
```

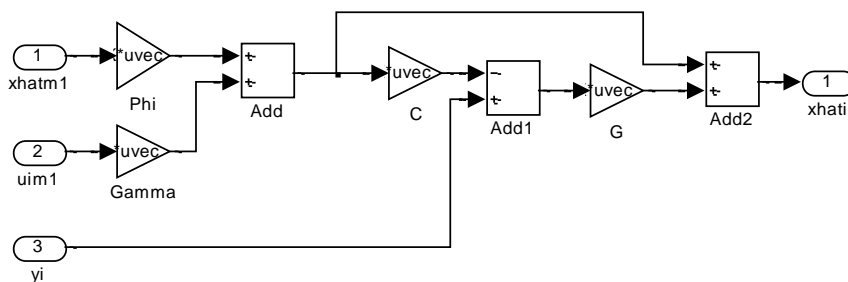
```
H =  
    0.0074  
    0.0002
```

```
OscNormalCl.mdl has been loaded  
Activate Scopes and Start Simulation  
>>
```

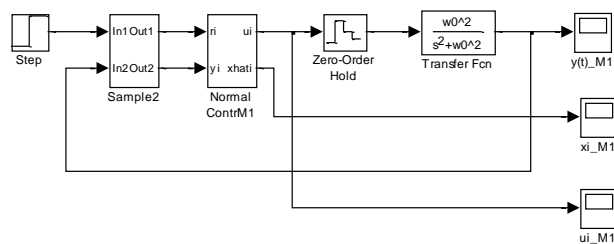
A rendszer és a normál szabályozó Simulink megvalósítása az 1. módszerrel:



6.11. ábra. A normál szabályozó Simulink modellje (1. módszer)

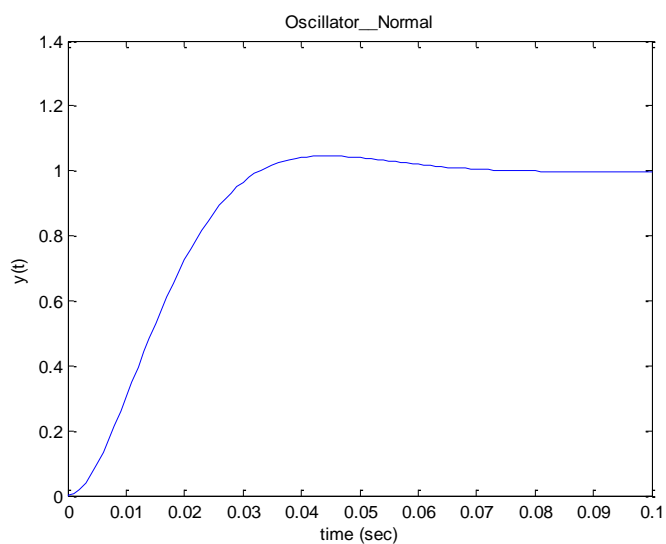


6.12. ábra. A megfigyelő Simulink modellje (1. módszer)

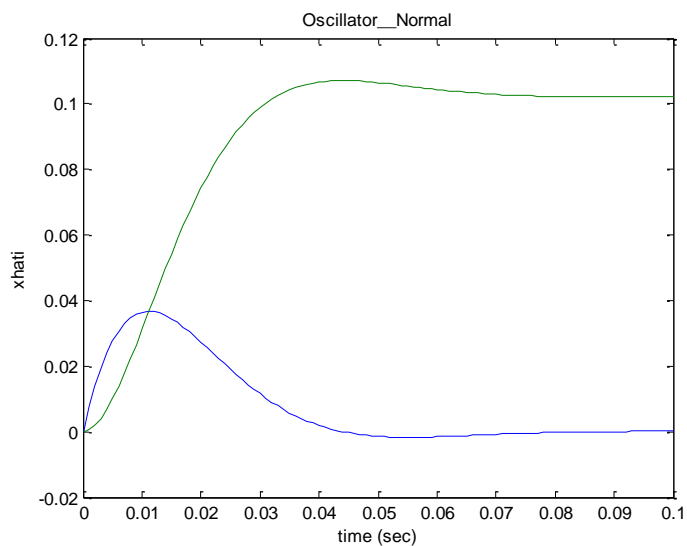


6.13. ábra. A zárt szabályozási kör Simulink modellje normál szabályozóval (1. módszer)

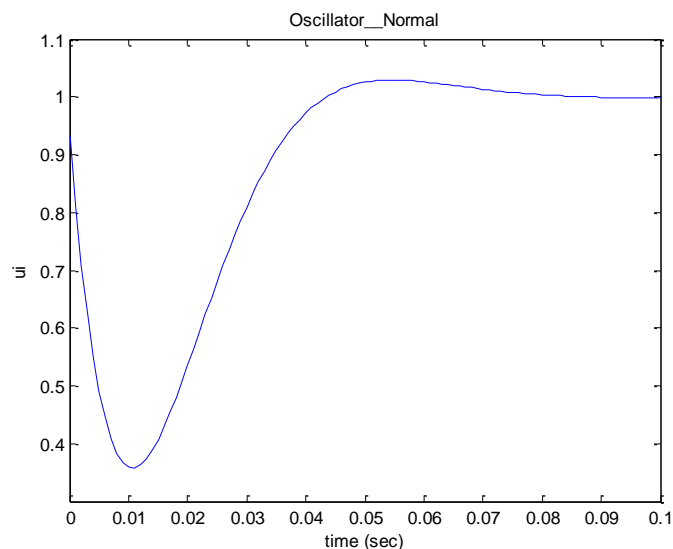
A zárt szabályozási rendszer tranziensei alapjelugrás esetén az $y(t)$, x_i , u_i oszcilloszkópokon jelennek meg, onnan kimenthető és felrajzolhatóak.



6.14. ábra. A szakasz kimenőjele normál szabályozóval (1. módszer)



6.15. ábra. A becstelt állapotváltozók normál szabályozóval (1. módszer)



6.16. ábra. A szabályozó kimenőjele normál szabályozóval (1. módszer)

Integráló szabályozás

A fenti kód folytatásaként az alábbi lefuttatva az integrátorral kiegészített szabályozó következő paramétereit kaptuk:

```

%-----
%Integrating control
%-----
fprintf('Integrating Control\n')
%State augmenting
Phit=[Phi zeros(2,1); Ts*C eye(1)]; %t instead of tilde
Gammat=[Gamma; zeros(1,1)];
Ct=[C zeros(1,1)];
Dt=0;
%Controller
pclt=[z1 conj(z1) zcinf]';
Kt=acker(Phit,Gammat,pclt);
K=Kt(1:2)
KI=Kt(3)
%Actual observer
pobs=[zoinf zoinf]';
KII=acker(Phi',Phi'*C',pobs);
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma
%Reference signal correction

```

```
Nxu=inv([Phi-eye(2) Gamma; C 0])*[0 0 1]';
Nx=Nxu(1:2);
Nu=Nxu(3);
%Simulation
open_system('OscIntegralCl')
fprintf('OscIntegralCl.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause
```

A megtervezett szabályozó paramétereit:

```
Phi =
    0.9950   -0.0780
    0.1278    0.9950
```

```
Gamma =
    0.0080
    0.0005
```

```
C =
         0    9.7656
```

```
D =
    0
```

Integrating Control

```
K =
    38.4277    24.3961
```

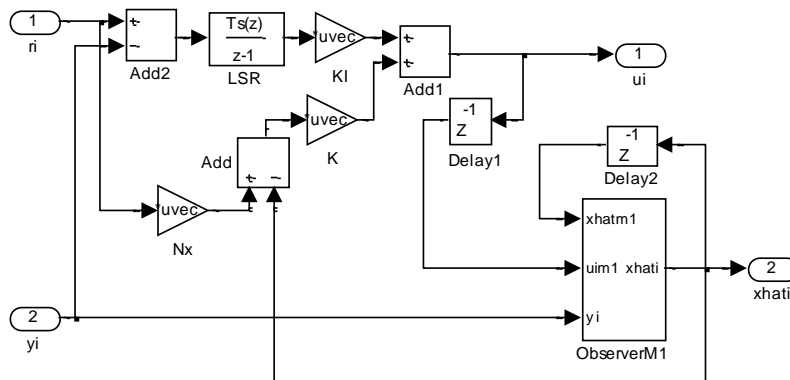
```
KI =
    176.7544
```

```
G =
    0.1186
    0.0647
```

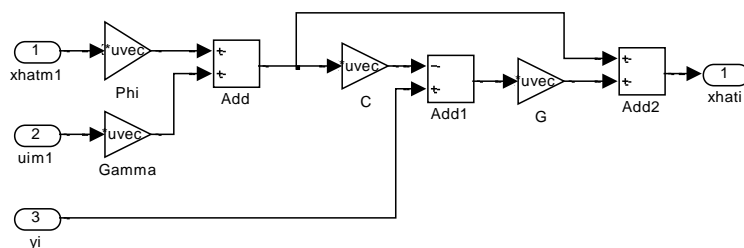
```
F =
    0.8470   -1.2303
    0.0470    0.3660
```

```
H =
    0.0074
    0.0002
```

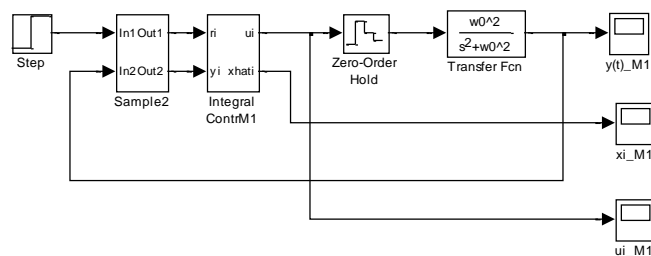
```
OscIntegralCl.mdl has been loaded
Activate Scopes and Start Simulation
>>
```

A rendszer és az integráló szabályozó Simulink megvalósítása az 1. módszerrel:


6.17. ábra. Az integráló szabályozó Simulink modellje (1. módszer)

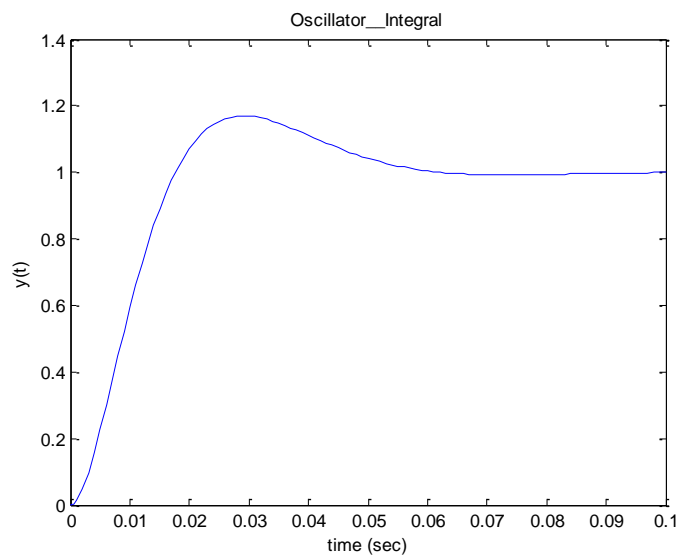


6.18. ábra. A megfigyelő Simulink modellje (1. módszer)

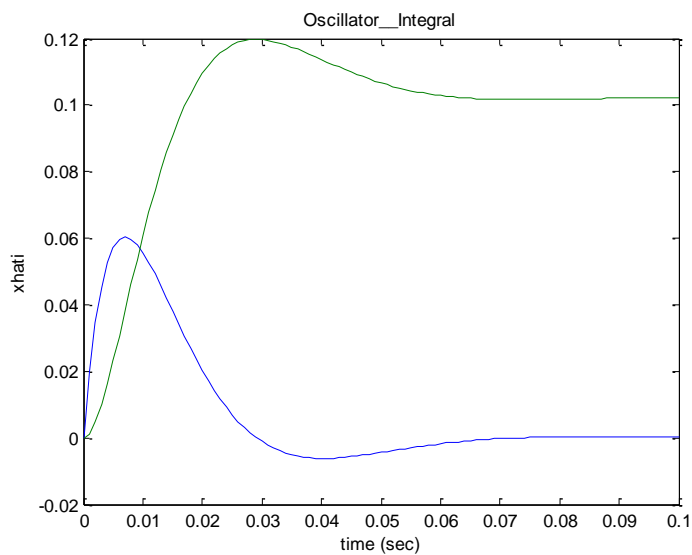


6.19. ábra. A zárt szabályozási kör Simulink modellje integráló szabályozóval (1. módszer)

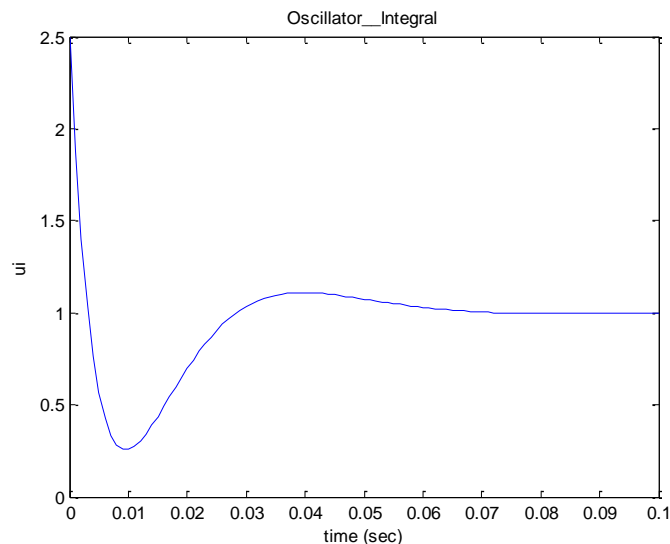
A zárt szabályozási rendszer tranziensei alapjelugrás esetén az $y(t)$, x_i , u_i oszcilloszkópokon jelennek meg, onnan kimentethők és felrajzolhatók. Figyeljük meg, hogy a kimenőjel túllövése megnőtt a normál esethez képest.



6.20. ábra. A szakasz kimenőjele integráló szabályozóval (1. módszer)



6.21. ábra. A becsült állapotváltozók integráló szabályozóval (1. módszer)



6.22. ábra. A szabályozó kimenőjele integráló szabályozóval (1. módszer)

Terhelésbecslés komplex szabályozóval

A fenti kód folytatásaként az alábbi lefuttatva a terhelésbecslővel kiegészített szabályozó következő paramétereit kaptuk:

```
%-----
%Load estimating control
%-----
fprintf('Load Estimating Control\n')
%State augmenting
Phit=[Phi Gamma; zeros(1,2) 1]; %t instead of tilde
Gammat=[Gamma; zeros(1,1)];
Ct=[C zeros(1,1)];
Dt=0;
%Controller
pcl=[z1 conj(z1)]';
K=acker(Phi,Gamma,pcl)
%Actual observer
ptobs=[zoinf zoinf zoinf]';
KtII=acker(Phit',Phit'*Ct',ptobs);
Gt=KtII'
Ft=Phit-Gt*Ct*Phit
Ht=Gammat-Gt*Ct*Gammat
%Reference signal correction
Nxu=inv([Phi-eye(2) Gamma; C 0])*[0 0 1]';
Nx=Nxu(1:2);
Nu=Nxu(3);
```

```
%Simulation M2
[Ac,Bc,Cc,Dc]=Controller_M2('load',Ts,Phit,Gammat,Ct,Gt,K,N
x,Nu)
open_system('OscLoadCl_M2')
fprintf('OscLoadCl_M2.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause
```

A megtervezett szabályozó paramétere:

```
Phi =
    0.9950   -0.0780
    0.1278    0.9950

Gamma =
    0.0080
    0.0005

C =
         0    9.7656

D =
    0

Load Estimating Control

K =
    16.3157   -0.6528

Gt =
    0.2941
    0.0796
    6.0967

Ft =
    0.6280   -2.9354    0.0065
    0.0285    0.2220    0.0001
   -7.6082  -59.2406    0.9695

Ht =
    0.0065
    0.0001
   -0.0305

Ac =
    0.8647   -2.4994         0
    0.1194   -0.1209         0
         0  -59.5381    1.0000
```

```

Bc =
    0.0075    0.2485
    0.0005    0.1143
         0     6.0967

Cc =
   -16.3157   106.5379   -1.0000
    1.0000    -2.8717         0
         0     0.2231         0
         0   -59.5381    1.0000

Dc =
    0.9332   -10.8426
         0     0.2941
         0     0.0796
         0     6.0967

```

OscLoadCl_M2.mdl has been loaded
 Activate Scopes and Start Simulation

A komplex szabályozó állapotegyenlete

A komplex szabályozó állapotegyenletének A_c , B_c , C_c , D_c mátrixait az

```
[Ac,Bc,Cc,Dc]=Controller_M2(ctype,Ts,Phit,Gammat,Ct,Gt,Kt,Nx,Nu)
```

függvénnyel határoztuk meg, melyet a 2. módszer állapotegyenletei alapján írtunk meg. A függvény MIMO rendszerek esetén is alkalmazható és mindhárom üzemmódra meghívható. Az üzemmódot *ctype* azonosítja a híváskor. Csak terhelésbecslésre a kód egyszerűsíthető. A függvény kódja az általános esetben a következő:

```

function
[Ac,Bc,Cc,Dc]=Controller_M2(ctype,Ts,Phit,Gammat,Ct,Gt,Kt,Nx,Nu)
%Generate complex state space model of observer and state feedback

m=size(Ct,1); %m=r assumed

switch ctype
    case 'normal'
        nbar=size(Phit,1);
        K=Kt;
        S=eye(nbar);
        Sd=zeros(m,nbar);
        Ac11=(Phit-Gammat*(K*S+Sd))*(eye(nbar)-Gt*Ct);
        Ac12=[];
        Bc11=Gammat*(K*Nx+Nu);
        Bc12=(Phit-Gammat*(K*S+Sd))*Gt;
        Ac21=[];
        Ac22=[];

```

```

Bc21=[];
Bc22=[];
Cc11=-(K*S+Sd)*(eye(nbar)-Gt*Ct);
Cc12=[];
Dc11=K*Nx+Nu;
Dc12=-(K*S+Sd)*Gt;
Cc21=S*(eye(nbar)-Gt*Ct);
Cc22=[];
Dc21=zeros(nbar,m);
Dc22=S*Gt;
Cc31=[];
Cc32=[];
Dc31=[];
Dc32=[];
Cc41=[];
Cc42=[];
Dc41=[];
Dc42=[];
case 'integral'
nbar=size(Phit,1);
K=Kt(1:m,1:nbar);
KI=Kt(1:m,nbar+1:nbar+m);
Nu=zeros(m,m);
S=eye(nbar);
Sd=zeros(m,nbar);
Ac11=(Phit-Gammat*(K*S+Sd))*(eye(nbar)-Gt*Ct);
Ac12=Gammat*KI;
Bc11=Gammat*(K*Nx+Nu);
Bc12=(Phit-Gammat*(K*S+Sd))*Gt;
Ac21=zeros(m,nbar);
Ac22=eye(m);
Bc21=Ts*eye(m);
Bc22=-Ts*eye(m);
Cc11=-(K*S+Sd)*(eye(nbar)-Gt*Ct);
Cc12=KI;
Dc11=K*Nx+Nu;
Dc12=-(K*S+Sd)*Gt;
Cc21=S*(eye(nbar)-Gt*Ct);
Cc22=zeros(nbar,m);
Dc21=zeros(nbar,m);
Dc22=S*Gt;
Cc31=[];
Cc32=[];
Dc31=[];
Dc32=[];
Cc41=zeros(m,nbar);
Cc42=eye(m);
Dc41=zeros(m,m);
Dc42=zeros(m,m);
case 'load'
nbar=size(Phit,1);
K=Kt;
S=[eye(nbar-m) zeros(nbar-m,m)];
Sd=[zeros(m,nbar-m) eye(m)];
Ac11=(Phit-Gammat*(K*S+Sd))*(eye(nbar)-Gt*Ct);

```



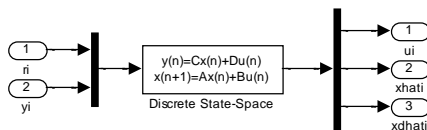
```

Ac12=[];
Bc11=Gammat*(K*Nx+Nu);
Bc12=(Phit-Gammat*(K*S+Sd))*Gt;
Ac21=[];
Ac22=[];
Bc21=[];
Bc22=[];
Cc11=-(K*S+Sd)*(eye(nbar)-Gt*Ct);
Cc12=[];
Dc11=K*Nx+Nu;
Dc12=-(K*S+Sd)*Gt;
Cc21=S*(eye(nbar)-Gt*Ct);
Cc22=[];
Dc21=zeros(nbar-m,m);
Dc22=S*Gt;
Cc31=Sd*(eye(nbar)-Gt*Ct);
Cc32=[];
Dc31=zeros(m,m);
Dc32=Sd*Gt;
Cc41=[];
Cc42=[];
Dc41=[];
Dc42=[];
otherwise
Ac=[]; Bc=[]; Cc=[]; Dc=[];
fprintf('Nonexisting ctype=%s\n',ctype);
return;
end;

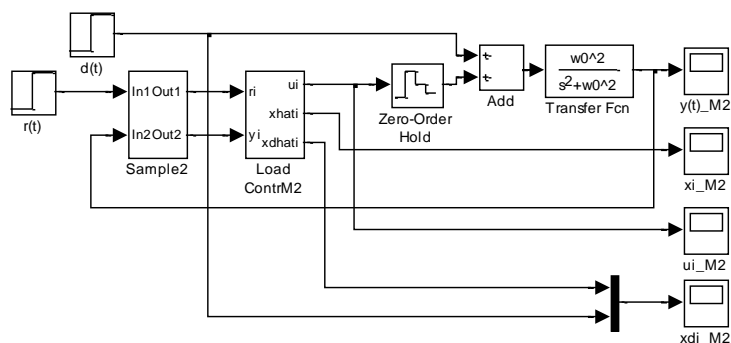
Ac=[Ac11 Ac12; Ac21 Ac22];
Bc=[Bc11 Bc12; Bc21 Bc22];
Cc=[Cc11 Cc12; Cc21 Cc22; Cc31 Cc32; Cc41 Cc42];
Dc=[Dc11 Dc12; Dc21 Dc22; Dc31 Dc32; Dc41 Dc42];

```

A rendszer és a terhelésbecsléssel kiegészített szabályozó Simulink megvalósítása az 2. módszerrel

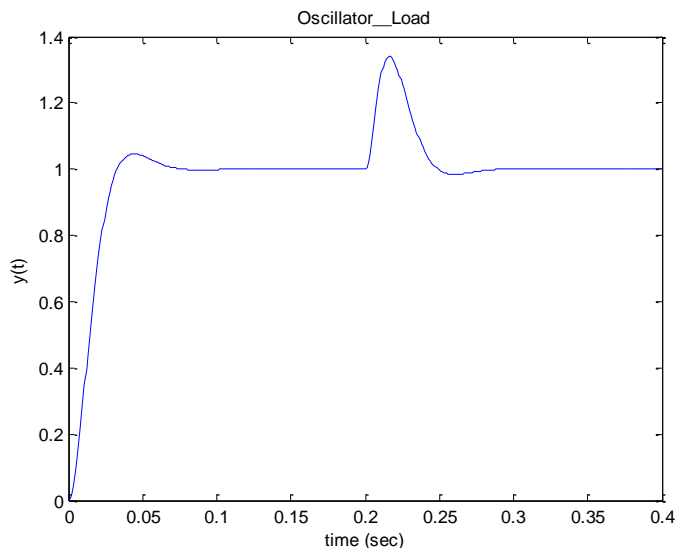


6.23. ábra. A terhelésbecslőt tartalmazó szabályozó komplex megvalósítása (2. módszer)

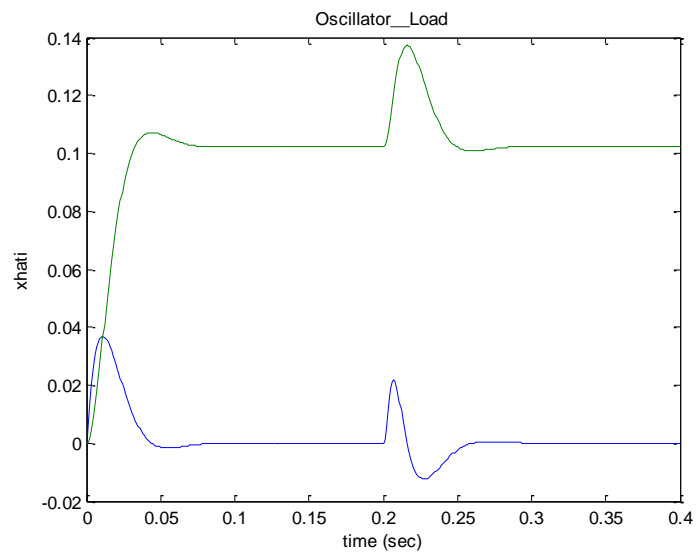


6.24. ábra. A zárt szabályozási kör terhelésbecslővel (2. módszer)

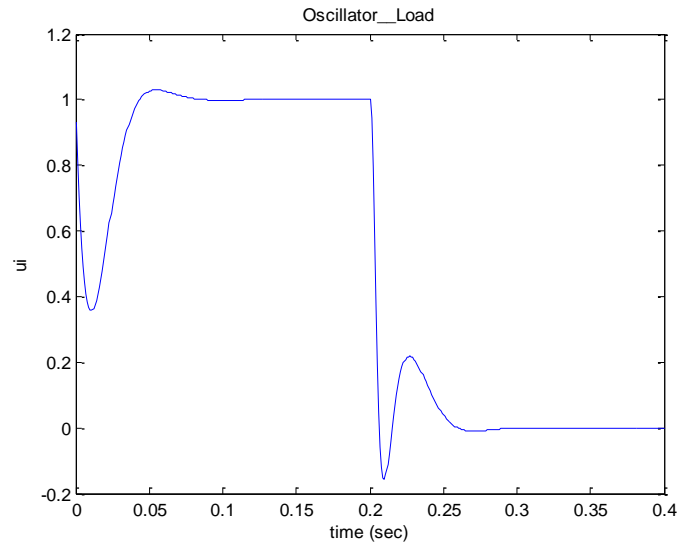
A zárt szabályozási rendszer tranziensei alapjelugrás esetén az $y(t)$, x_i , u_i , x_{di} oszcilloszkópokon jelennek meg, ahonnan `Workspace` változóba menthetők és felrajzolhatók. A tárolást a `Scope` paramétereit között lehet beállítani. A változók a `ScopeData`, `ScopeData1`, ... struktúrákban keletkeznek. A struktúra tartalmazza az időt, a jeleket és azok értékeit.



6.25. ábra. A szakasz kimenőjele terhelésbecslés esetén (2. módszer)

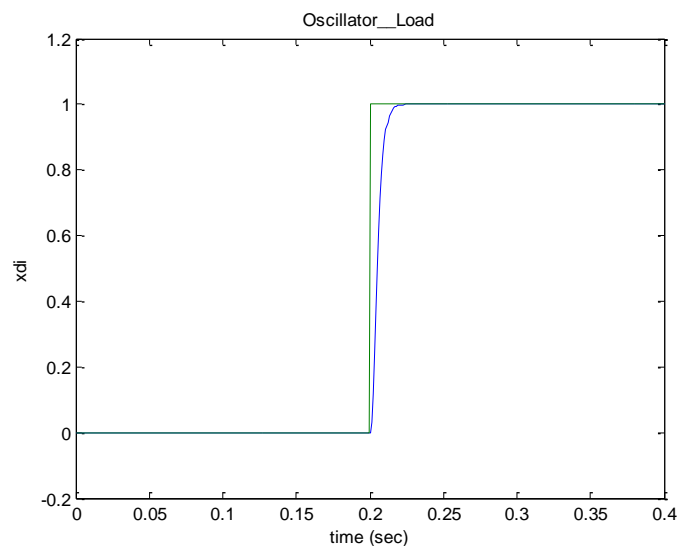


6.26. ábra. A szakasz becült állapotváltozói terhelésbecslés esetén (2. módszer)



6.27. ábra. A szabályozó kimenőjele terhelésbecslés esetén (2. módszer)

Az alapjelugrást követően a tranziensek lecsengése után megvizsgáltuk a bemeneten ható zavarójel (terhelés) hatását is. Az egységugrás alakú zavarójel 0.2 sec-nál kezdődik, kompenzálását az x_d állapotváltozó végzi, lásd 6.28. ábra.



6.28. ábra. A zavarójel kompenzálása terhelésbecslés esetén (2. módszer)

Terhelésbecslés algebrai hurkot tartalmazó szabályozóval

Algebrai hurkot tartalmazó Simulink modell nem emelhető át beágyazott rendszere célprocesszorára a Realtime Workshop és a target compiler segítségével. Ezért okulásképpen megmutatjuk, hogyha a szabályozót megfigyelő állapotegyenletre és alapjel korrekcióval is ellátott állapot-visszacsatolásra bontjuk, akkor a Simulink futtatáskor algebrai hurkot jelez ki, bár MATLAB környezetben a szimuláció (helyes konfiguráció paraméterválasztás esetén) jó eredményt ad. Hangsúlyozzuk azonban, hogy ez a szabályozó megoldás nem emelhető át beágyazott processzorra, ezért ilyen célú fejlesztés esetén ez a megoldás nem, hanem helyette a komplex szabályozó választandó.

A vizsgálathoz az `xOscillator_Load.m` script fájl használható, amelynek induló része azonos a komplex szabályozónál megismerttel, de az utolsó része a következőre módosul:

```
%Simulation M3
%Prepare M3
S=[eye(2) zeros(2,1)]
```

```

Sd=[zeros(1,2) eye(1)]
Ao=Phit*(eye(3)-Gt*Ct)
Bo=[Gammat Phit*Gt]
Co=[S; Sd]*(eye(3)-Gt*Ct)
Do=[zeros(3,1) [S; Sd]*Gt]
%
open_system('xOscLoadCl_M3')
fprintf('xOscLoadCl_M3.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause

```

Az új eredmények az S, S_d szelektorokat, továbbá az u_i, y_i bemenőjelű és $\hat{x}_i, \hat{x}_{d,i}$ kimenőjelű $\Sigma_o = (A_o, B_o, C_o, D_o)$ megfigyelő állapotegyenletének mátrixait tartalmazzák:

S =

1	0	0
0	1	0

Sd =

0	0	1
---	---	---

Ao =

0.9950	-3.3503	0.0080
0.1278	-0.1754	0.0005
0	-59.5381	1.0000

Bo =

0.0080	0.3351
0.0005	0.1199
0	6.0967

Co =

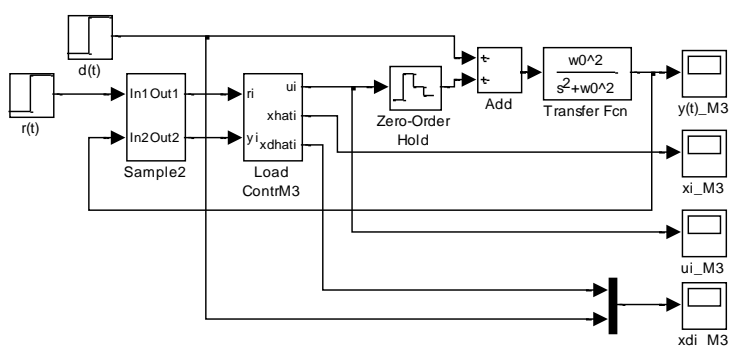
1.0000	-2.8717	0
0	0.2231	0
0	-59.5381	1.0000

Do =

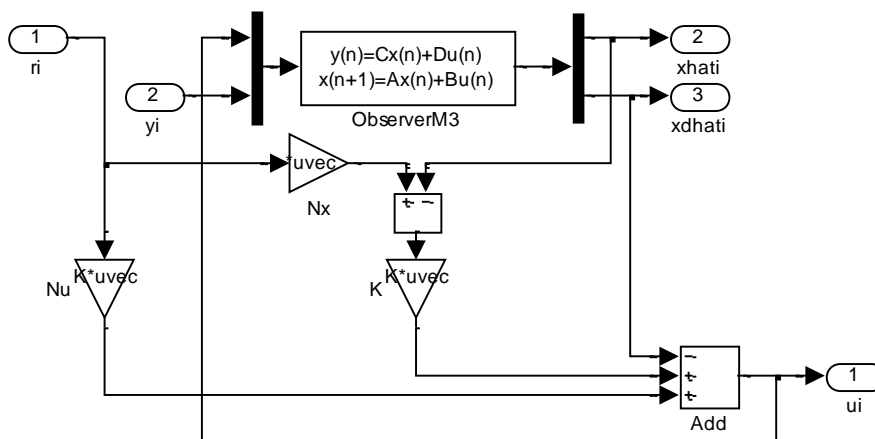
0	0.2941
0	0.0796
0	6.0967

xOscLoadCl_M3.mdl has been loaded
 Activate Scopes and Start Simulation

A zárt rendszer és a szabályozó Simulink modelljét a 6.29. és 6.30. ábrák mutatják be.



6.29. ábra. A zárt rendszer Simulink modellje algebrai hurkot tartalmazó terheléscsülés esetén (3. módszer)



6.30. ábra. A szabályozó Simulink modellje algebrai hurkot tartalmazó terheléscsülés esetén (3. módszer)

A szimuláció indításakor a Simulink kijelzi az algebrai hurkot:

```
Warning: Block diagram 'xOscLoadCl_M3' contains 1 algebraic
loop(s). To see more details about the loops use the
command line Simulink debugger by typing "sldebug
xOscLoadCl_M3" in the MATLAB command window. To eliminate
this message, set the Algebraic loop option in the
Diagnostics page of the Simulation Parameters Dialog to
"None".
```

Found algebraic loop containing:

```
'xOscLoadCl_M3/Load ContrM3/ObserverM3'
'xOscLoadCl_M3/Load      ContrM3/Subtract'      (algebraic
variable)
'xOscLoadCl_M3/Load ContrM3/K'
'xOscLoadCl_M3/Load ContrM3/Add' (algebraic variable)
```

Az üzenet ellenére helyes szimulációs eredmények születnek, az oszcilloszkópokon most is a 6.25-6.28. ábrákon megadott eredmények jelennek meg, ezért ezeket nem ismétljük meg.

A következő kiegészítő kódrészlet alkalmazásával a komplex szabályozó állapotegyenlete előállítható a megfigyelő és a statikus szabályozó állapotegyenleteinek összekapcsolásával is:

```
%Connection to complex controller
fprintf('\nConnect compenents to complex controller\n');
fprintf('\nsysobs:\n');
sysobs=ss(Ao,Bo,Co,Do,Ts);
sysobs.InputName={'ui';'yi'};
sysobs.OutputName={'xhati1';'xhati2';'xdhati'}
%Static controller
fprintf('\nStatic controller:\n');
Ac0=[]
Bc0=[]
Cc0=[]
Dc0=[K*Nx+Nu -K -1]
syscstat=ss(Ac0,Bc0,Cc0,Dc0,Ts);
syscstat.InputName={'ri';'xhati1'; 'xhati2'; 'xdhati'};
syscstat.OutputName={'ui'}
%
fprintf('\nConnected system:\n')
sysc=connect(sysobs,syscstat,{'ri';'yi'},{'ui';'xhati1';
'xhati2'; 'xdhati'})
%Complex controller
fprintf('\nComplex controller');
[Ac,Bc,Cc,Dc]=ssdata(sysc)
```

A kódrészlet lefutásakor a következő eredmények jelennek meg:

Connect components to complex controller

sysobs:

```
a =
      x1      x2      x3
x1    0.995   -3.35   0.007987
x2    0.1278  -0.1754  0.0005116
x3     0      -59.54     1
```

```
b =
      ui      yi
x1    0.007987  0.3351
x2    0.0005116  0.1199
x3     0         6.097
```

```
c =
      x1      x2      x3
xhati1  1  -2.872   0
xhati2  0  0.2231  0
xdhati  0  -59.54   1
```

```
d =
      ui      yi
xhati1  0  0.2941
xhati2  0  0.07955
xdhati  0  6.097
```

Sampling time: 0.001

Discrete-time model.

Static controller:

Ac0 =

[]

Bc0 =

[]

Cc0 =

[]

Dc0 =

```
0.9332  -16.3157   0.6528  -1.0000
```

d =

```
      ri  xhati1  xhati2  xdhati
ui  0.9332 -16.32  0.6528   -1
```

Static gain.

Connected system:

a =

```
      x1      x2      x3
x1  0.8647  -2.499   0
x2  0.1194  -0.1209  0
x3      0  -59.54   1
```

b =

```
      ri      yi
x1  0.007453  0.2485
x2  0.0004774 0.1143
x3      0      6.097
```

c =

```
      x1      x2      x3
ui  -16.32  106.5   -1
xhati1  1  -2.872   0
xhati2  0  0.2231  0
xdhati  0  -59.54  1
```

d =

```
      ri      yi
ui  0.9332 -10.84
xhati1  0  0.2941
xhati2  0  0.07955
xdhati  0  6.097
```

Sampling time: 0.001

Discrete-time model.

Complex controller

Ac =

```
0.8647  -2.4994   0
0.1194  -0.1209   0
      0  -59.5381  1.0000
```

Bc =

0.0075	0.2485
0.0005	0.1143
0	6.0967

Cc =

-16.3157	106.5379	-1.0000
1.0000	-2.8717	0
0	0.2231	0
0	-59.5381	1.0000

Dc =

0.9332	-10.8426
0	0.2941
0	0.0796
0	6.0967

Vegyük észre, hogy a keletkezett komplex szabályozó azonos a korábban más technikával kapottal.

6. Számítógéptermi gyakorlat: Állapot-visszacsatolás és megfigyelő tervezése diszkrét időben

A szabályozott szakasz leírása nemcsak átviteli függvény segítségével, hanem állapotteretes leírással is történhet. Az állapotterben megadott modell információban gazdagabb, figyelembe tudja venni a korszerű érzékelőket (GPS, 3D gyorsulás és szögsebesség érzékelők, képfeldolgozáson alapuló pozíció és orientáció meghatározás), ezért a szabályozótervezés sok esetben az állapotteretes leíráson alapul. Másrészt a korszerű szabályozók valamilyen processzoron, esetleg beágyazott szabályozási rendszerként kerülnek megvalósításra, ezért pontosabb megoldást kapunk, ha nem a folytonosidejű állapotteretes szabályozó mintavételes közelítésével próbálkozunk, hanem közvetlenül, egzakt módszerekkel diszkrét időben tervezzük és valósítjuk meg az állapotteretes szabályozást.

Az alábbiakban összefoglaljuk a diszkrétidejű szabályozótervezés hasonlóságát és esetleges eltéréseit a folytonosidejű esethez képest. Az elméleti alapokat a tankönyv részletesen tárgyalja, a tantermi gyakorlat bevezető része pedig összefoglalja a fontosabb eredményeket. A számítógépi gyakorlaton a tantermi gyakorlat elméleti összefoglalóját komprimáltan megismételjük, majd a módszereket tervezési feladatok keretében alkalmazzuk.

Az elméleti alapok összefoglalása

A folytonosidejű és diszkrétidejű időinvariáns lineáris rendszerek állapotterbeli elveken alapuló irányítását a közöttük fennálló algebrai hasonlóság miatt hasonló módszerekkel lehet megtervezni. A folytonosidejű szabályozott szakasz állapotegyenletének alakja legyen a szokásos $\dot{x} = Ax + Bu$, $y = Cx$, tömören $\Sigma_c = (A, B, C, D)$, és álljon a szakasz előtt digitális/analóg átalakító (DAC), amely nulladrendű tartószervvel modellezhető, a szakasz után pedig analóg/digitális átalakító (ADC). Mivel ilyenkor a szakasz bemenetén lépcsős analóg jel van, ezért az eredő mintavételes rendszert a $\Sigma_d = (\Phi, \Gamma, C, D)$ alakra lehet hozni:

$$\begin{aligned} (A, B, C, D) &\xrightarrow[\text{zoh}]{c2d} (\Phi, \Gamma, C, D) \\ x_{i+1} &= \Phi x_i + \Gamma u_i, \\ y_i &= C x_i. \end{aligned} \tag{6.30}$$

Az irányíthatósági és megfigyelhetőségi mátrix a folytonosidejű rendszerekhez hasonlóan:

$$M_c = [\Gamma \ \Phi\Gamma \ \dots \ \Phi^{n-1}\Gamma] \quad (6.31)$$

$$M_o = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix}. \quad (6.32)$$

Megjegyezzük azonban, hogy diszkrét időben $\text{rank } M_c = n = \dim x$ feltétel a teljes elérhetőségnek (nulla állapotból) szükséges és elégséges feltétele, míg a teljes irányíthatóságnak (nulla állapotba) csak elégséges feltétele. Hasonlóan diszkrét időben $\text{rank } M_o = n = \dim x$ a teljes megfigyelhetőségnek (jövőbeli bemeneti-kimeneti adatokból) szükséges és elégséges feltétele, míg a teljes rekonstruálhatóságnak (múltbeli bemeneti-kimeneti adatokból) csak elégséges feltétele.

A továbbiakban feltesszük, hogy a mintavételes rendszer teljesen elérhető (irányítható) és teljesen megfigyelhető, vagyis $\text{rank } M_c = \text{rank } M_o = \dim x = n$ teljesül.

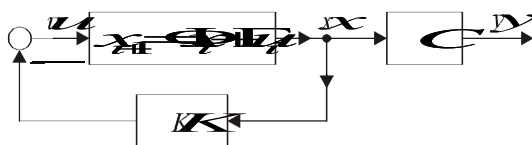
Pólusát helyezés állapot-visszacsatolással

A 6.31. ábrának megfelelően alkalmazzunk

$$u_i = -Kx_i \quad (6.33)$$

állapot-visszacsatolást, akkor a zárt rendszer állapotegyenlete és karakterisztikus egyenlete rendre a következő lesz:

$$\begin{aligned} x_{i+1} &= (\Phi - \Gamma K) x_i, \\ \varphi_c(z) &= \det(zI - (\Phi - \Gamma K)). \end{aligned} \quad (6.34)$$



6.31. ábra. Állapot-visszacsatolás diszkrét időben

Ha előírjuk a zárt rendszer $\varphi_c(z)$ karakterisztikus egyenletét, és keressük az ehhez megfelelő K állapot-visszacsatolást, akkor a folytonosidejű feladathoz algebraiailag hasonló feladatra jutunk, és ezért K meghatározható a folytonosidejű

rendszerekre kidolgozott módszerekkel. Egyváltozós (SISO) rendszereknél a pólusáthelyezési feladat megoldása meghatározható az Ackermann-képlettel:

$$K = (0 \dots 0 \ 1) M_c^{-1} \varphi_c(\Phi), \quad (6.35)$$

$$(\Phi, \Gamma) \xrightarrow{\frac{\varphi_c(z)}{M_c}} K. \quad (6.36)$$

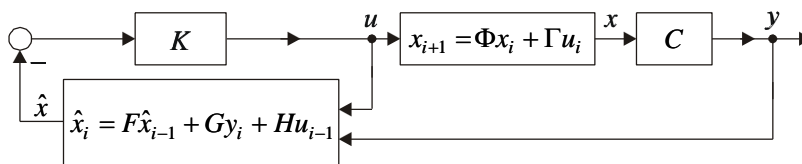
Ha a kívánt sajátértékek vektora $[\lambda_1 \ \lambda_2 \ \dots \ \lambda_n]^T$, akkor az $\Phi - \Gamma K$ mátrix karakterisztikus egyenlete $\varphi_c(z) = \prod_{i=1}^n (z - \lambda_i)$.

Az Ackermann-képletben szerepel az M_c mátrix inverze, amely SISO esetben kvadratikusan, és a feltevés szerint maximális rangú. A Φ mátrix szintén kvadratikusan, ezért hatványozható és Φ^i behelyettesíthető a z^i hatványok helyére a zárt rendszer $\varphi_c(z)$ karakterisztikus polinomjában. A Control System Toolbox-ban (CST) rendelkezésünkre áll az `acker` utasítás is K meghatározására, amint azt az 5. gyakorlat keretében már láttuk.

Ne felejtjük azonban el, hogy diszkrét időben a stabilitáshoz a sajátértékeknek az egységkör belsejében kell lenniük, továbbá a gyors sajátértékek $z=0 = e^{-\infty T}$ közelében vannak. Másrészt viszont a rendszer gyorsítása a beavatkozó jel növekedésével jár, amely telítést okozhat a szabályozót követő beavatkozó szervben vagy a DAC-ban (mivel a bitszám limitált). Nem szabad arról sem megfeledkezni, hogy a Shannon-tételnek nemcsak a szakaszra, hanem a felgyorsított rendszer minden jelére, így a szabályozóra (állapot-visszacsatolás, megfigyelő) is teljesülni kell, ezért a gyorsításnak és a T mintavételi időnek is összhangban kell lennie, lásd $\omega_h \leq \pi/T = \omega_N$. A specifikációkat most is célszerű folytonos időben $\xi, \omega_0, s_{c\infty}, s_{o\infty}$ révén megfogalmazni, betartván hogy közülük pl. a leggyorsabbra is teljesüljön kb. $|s_{o\infty}| \leq 0.1\omega_N$.

Állapot-visszacsatolás realizálása aktuális megfigyelővel

A rendszerint nem mérhető x állapotváltozó becslésére diszkrétidejű rendszereknél is állapotmegfigyelőt célszerű alkalmazni. Diszkrét időben azonban aktuális állapotmegfigyelőt célszerű alkalmazni, lásd 6.32. ábra.



6.32. ábra. Aktuális állapotmegfigyelő diszkrét időben

Feltesszük, hogy $(\Phi, C\Phi)$ megfigyelhető, pl. $\exists \Phi^{-1}$ és (Φ, C) megfigyelhető. Akkor választható az

$$\hat{x}_i = F\hat{x}_{i-1} + Gy_i + Hu_{i-1} \quad (6.37)$$

aktuális állapotmegfigyelő. Jelölje $\tilde{x}_i = x_i - \hat{x}_i$ az állapotbecslés hibáját, akkor

$$\tilde{x}_i = F(x_{i-1} - \hat{x}_{i-1}) + (\Gamma - GCF - H)u_{i-1} + (\Phi - GC\Phi - F)x_{i-1}, \quad (6.38)$$

ezért aszimptotikus állapotmegfigyelőhöz jutunk, ha $\tilde{x}_i \rightarrow 0$ biztosítása érdekében a következő választással élünk:

$$F = \Phi - GC\Phi, \quad (6.39a)$$

$$H = \Gamma - GCF, \quad (6.39b)$$

$$\tilde{x}_i = F\tilde{x}_{i-1} \text{ stabil és gyors.} \quad (6.39c)$$

Ha a megfigyelő tranziensének gyorsaságát a megfigyelő $\varphi_o(z)$ karakterisztikus egyenletével írjuk elő, ahol

$$\varphi_o(z) = \det(zI - F) = \det(zI - (\Phi - GC\Phi)),$$

akkor G és F meghatározható

$$(\Phi, C)_I \leftrightarrow (\Phi^T, \Phi^T C^T)_{II} \xrightarrow[M_{c,II}]{\varphi_o(z)} K_{II} \rightarrow G = K_{II}^T \rightarrow F = \Phi - GC\Phi \quad (6.11)$$

alapján, SISO rendszer esetén pedig az Ackermann-képlettel (vagy más ekvivalens módszerrel). Végül G ismeretében számítható H is (6.39b) alapján.

Az \hat{x}_i becslt állapot számítása valós idejű megvalósítás szempontjából kedvezőbb alakra is hozható a következő átalakítással:

$$\begin{aligned} \hat{x}_i &= (\Phi - GC\Phi)\hat{x}_{i-1} + Gy_i + (\Gamma - GCF)u_{i-1} = \\ &= \Phi\hat{x}_{i-1} + \Gamma u_{i-1} + G\{y_i - C(\Phi\hat{x}_{i-1} + \Gamma u_{i-1})\}. \end{aligned} \quad (6.40)$$

Jól látható, hogy a számítások közül $\Phi\hat{x}_{i-1} + \Gamma u_{i-1}$ az utolsó mintavételnél azonnal indítható, míg Gy_i csak a következő mintavételnél, miután y_i mérése már megtörtént. Ez a processzor idő jobb kihasználását eredményezi, különösen, ha $n = \dim x$ nagy. Az aktuális megfigyelő \tilde{x}_i bevezetésével (amelynek számítása két mintavétel közötti időben is elvégezhető) a következő alakra hozható:

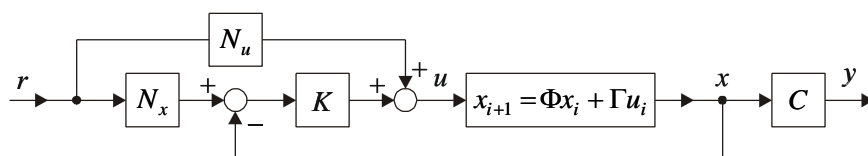
$$\begin{aligned}\bar{x}_i &= \Phi \hat{x}_{i-1} + \Gamma u_{i-1} && \text{"time-update"} \\ \hat{x}_i &= \bar{x}_i + G(y_i - C \bar{x}_i) && \text{"measurement-update"}\end{aligned}\quad (6.41)$$

Ez a felbontás más állapotbecslési struktúráknál is tipikus, így a sztochasztikus elven alapuló Kalman-szűrőnél is. Vegyük észre, hogy $C \bar{x}_i$ azt mutatja, hol várjuk a kimenetet a régi megfigyelések alapján.

Az alapjel miatti korrekció diszkrét időben

Az alapjel figyelembevételére a folytonos időben megismert módszer analógiájára a 6.33. ábra szerinti megoldás választható. Bár az alapjel miatti korrekció elvei ugyanazok lesznek, tekintettel kell lenni arra, hogy az állapotegyenlet megoldását állandósult állapotban az jellemzi, hogy a régi és az új állapot azonos:

$$x_\infty = \Phi x_\infty + \Gamma u_\infty. \quad (6.42)$$



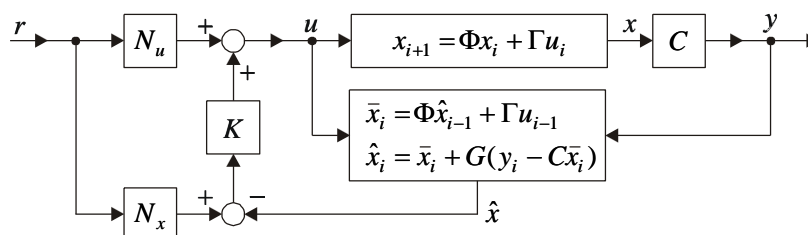
6.33. ábra. Alapjel miatti korrekció diszkrét időben

Most is feltesszük, hogy az alapjel konstans (pontosabban ritkán változik meg és utána hosszabb ideig állandó marad). Úgy döntünk, hogy állandósult állapotban az $N_x r - x_\infty$ különbségi jel legyen nulla, a kimenet állandósult állapotbeli értéke pedig legyen hiba nélküli, tehát $y_\infty = r$. Az ehhez szükséges bemenőjelet állandósult állapotban $u_\infty = N_u r$ fogja szolgáltatni. Legyen $\dim y = \dim r = \dim u = m$, akkor N_x egy $n \times m$, N_u pedig egy $m \times m$ méretű mátrix (SISO rendszer esetén nyilván $m=1$). A folytonosidejű esetben N_x és N_u számítására megismert módszert kismértékben korrigálni kell, mivel diszkrét időben az állandósult állapotra vonatkozó matematikai összefüggés más:

$$\begin{aligned}N_x r = x_\infty &\Rightarrow y_\infty = C x_\infty = C N_x r = r \Rightarrow C N_x = I_m, \\ N_u r = u_\infty &\Rightarrow \Phi x_\infty + \Gamma u_\infty = x_\infty \Rightarrow [(\Phi - I) N_x + \Gamma N_u] r = 0 \Rightarrow \\ &(\Phi - I) N_x + \Gamma N_u = 0_{n \times m}\end{aligned}$$

$$\begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0_{n \times m} \\ I_m \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0_{n \times m} \\ I_m \end{bmatrix}. \quad (6.43)$$

A 6.34. ábra állapot-visszacsatolásból (ÁV), állapotmegfigyelőből (ÁM) és alapjel miatti korrekcióból (N_x, N_u) felépülő egyszerű szabályozás megvalósításának elvét mutatja be, figyelembe véve az állapotmegfigyelő valós idejű realizálásának szempontjait is. A szabályozó bemenete r és y , kimenete pedig u .



6.34. ábra. Állapot-visszacsatolást és aktuális megfigyelőt alkalmazó egyszerű szabályozás megvalósításának elve diszkrét időben

Integráló szabályozás

A zavarás hatásának csökkentésére célszerű a szabályozóban integrátort elhelyezni. Egészítsük ki ezért a rendszert az integrátor $x_I = \int y dt$ jelével, és alkalmazzuk számítására a bal oldali téglalapszabályt (LSR):

$$x_{I,i+1} = x_{I,i} + T y_i = x_{I,i} + T C x_i \quad (6.44)$$

Az állapotegyenlet az $\tilde{x} = (x^T, x_I^T)^T$ bővített állapotváltozó bevezetése után a következő lesz:

$$\begin{aligned} \begin{pmatrix} x_{i+1} \\ x_{I,i+1} \end{pmatrix} &= \begin{bmatrix} \Phi & 0 \\ T C & I \end{bmatrix} \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} + \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} u_i \Rightarrow \tilde{x}_{i+1} = \tilde{\Phi} \tilde{x}_i + \tilde{\Gamma} u_i, \\ y_i &= [C \quad 0] \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} \Rightarrow y_i = \tilde{C} \tilde{x}_i. \end{aligned} \quad (6.45)$$

Az integrátor jelével bővített állapot-visszacsatolás ekkor

$$u_i = -\tilde{K} \tilde{x} = -[K \quad K_I] \begin{pmatrix} x_i \\ x_{I,i} \end{pmatrix} = -K x_i - K_I x_{I,i}, \quad (6.46)$$

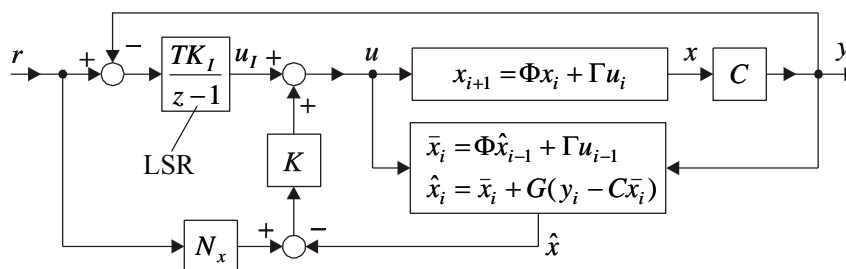
a zárt rendszer állapotegyenlete pedig

$$\tilde{x}_{i+1} = (\tilde{\Phi} - \tilde{\Gamma} \tilde{K}) \tilde{x}_i. \quad (6.47)$$

Ha előírjuk a zárt rendszer $\tilde{\varphi}_c(z) = \det(zI - (\tilde{\Phi} - \tilde{\Gamma} \tilde{K}))$ karakterisztikus egyenletét, akkor a pólusát helyezési feladat megoldásához szükséges $\tilde{K} = [K \ K_I]$ állapot-visszacsatolás meghatározható a

$$(\tilde{\Phi}, \tilde{\Gamma}) \xrightarrow[\tilde{M}_c]{\tilde{\varphi}_c(z)} \tilde{K}. \quad (6.48)$$

séma alapján, egyváltozós (SISO) rendszereknél pl. az Ackermann-képlettel.



6.35. ábra. Integráló szabályozás megvalósításának elve diszkrét időben

Az integráló szabályozás megvalósítását a 6.35. ábrán mutatjuk be. A realizációnál felhasználjuk, hogy az integrátor állandósult állapotban nulla bemenetnél is tud nemnulla kimenetet biztosítani, ezért $N_u r$ hatását is realizálja. Az integrátort a jobb oldali téglalapszabállyal (RSR) is implementálhatjuk.

A tervezés lépései a következők:

$$\begin{aligned} (\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}) &\rightarrow \tilde{K} = [K \ K_I] \quad (\text{ÁV}) \\ (\Phi, \Gamma, C) &\rightarrow (F, G, H) \quad (\text{ÁM}) \\ (\Phi, \Gamma, C) &\rightarrow (N_x, N_u) \end{aligned} \quad (6.49)$$

Terhelésbecslés

A zavarást a szakasz bemenetére redukálnak képzeljük ("load change"). A továbbiakban feltesszük, hogy a zavarás konstans, ezért differenciálegyenlete

$$\dot{d} = 0 \quad (d \text{ konstans, értéke ismeretlen}) \Rightarrow d_{i+1} = d_i. \quad (6.50)$$

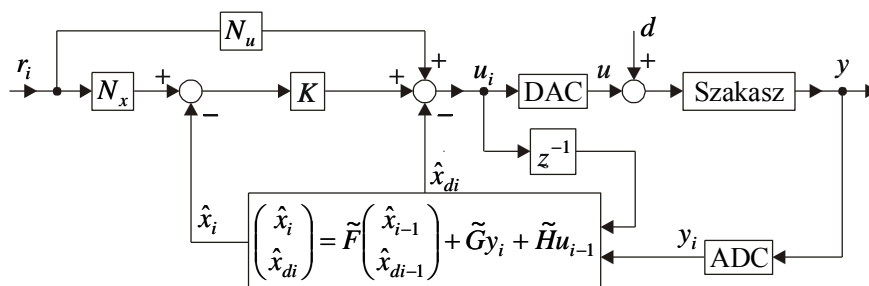
Ha a rendszert bővítjük az $x_d = d$ állapotváltozóval és bevezetjük az $\tilde{x} = (x^T, x_d^T)^T$ jelölést, akkor az állapotegyenlet a következő lesz:

$$\begin{aligned} \begin{pmatrix} x \\ x_d \end{pmatrix}_{i+1} &= \begin{bmatrix} \Phi & \Gamma \\ 0 & I \end{bmatrix} \begin{pmatrix} x \\ x_d \end{pmatrix}_i + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u_i \Rightarrow \tilde{x}_{i+1} = \tilde{\Phi} \tilde{x}_i + \tilde{\Gamma} u_i, \\ y_i &= [C \quad 0] \begin{pmatrix} x \\ x_d \end{pmatrix}_i \Rightarrow y_i = \tilde{C} \tilde{x}_i. \end{aligned} \quad (6.51)$$

Mivel az $(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C})$ rendszer nem irányítható (az x_d kívülről érkező jelet nem lehet belülről u -val irányítani), ezért az állapot-visszacsatolást és az alapjel miatti korrekciót az eredeti rendszerhez kell meghatározni. A megfigyelőt a bővített rendszerhez kell megtervezni, hogy biztosítani tudja mind az \hat{x} , mind pedig az \hat{x}_d becslést a szabályozó számára:

$$\begin{aligned} (\Phi, \Gamma, C) &\rightarrow K \quad (\text{ÁV}) \\ (\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}) &\rightarrow (\tilde{F}, \tilde{G}, \tilde{H}) \quad (\text{ÁM}) \\ (\Phi, \Gamma, C) &\rightarrow (N_x, N_u) \end{aligned} \quad (6.52)$$

A terhelésbecslést is tartalmazó szabályozás megvalósításának elvét a 6.36. ábra mutatja be.



6.36. ábra. Terhelésbecslést alkalmazó szabályozás megvalósításának elve diszkrét időben

A szabályozók megvalósítása

A szabályozók megvalósítására a következő lehetőségek kínálkoznak:

- A szabályozó realizálása a megtervezett komponensek összekötésével a Simulink alapelemkészletének felhasználásával.
- A szabályozó felbontása megfigyelőre (állapotbecslőre) és állapot-visszacatolásra az alapjel figyelembevételével, és a komponensek összekötése. Ez a módszer a megfigyelőt állapotegyenlettel írja le, amelyben az állapotváltozó nem lehet \hat{x} , mivel ekkor \hat{x}_{i+1} jobb oldalán megjelenne Gy_{i+1} , amely nem áll rendelkezésre. A helyes megfigyelő állapot ezért \bar{x} , mert ekkor \bar{x}_{i+1} jobb oldalán Gy_i áll, amely rendelkezésre áll. Ez a dekompozíciós módszer azonban Simulink környezetben algebrai hurkot eredményez az u_i visszavezetése miatt a megfigyelő bemenetére, amely problémát okoz, ha a szabályozót ki akarjuk hozni a Simulink alól egy beágyazott processzorra.
- A CST-ben van azonban két jobb lehetőség is. A két alrendszer (dinamikus megfigyelő és statikus szabályozó) realizálható egy-egy rendszerben (`sysobs`, `sysstat`), majd képezhető eredőjük a CST szolgáltatásaival (`connect stb.`), amely már átadható a Simulink-nek algebrai hurok nélkül. Egy még célszerűbb megoldás azonban közvetlenül egy lépésben a teljes komplex szabályozó állapotegyenletét meghatározni egy saját fejlesztésű függvényvel, és az eredő $\Sigma_c = (A_c, B_c, C_c, D_c)$ rendszert diszkrétidejű állapotegyenlet formájában átadni a Simulink számára

1. módszer: Szabályozó realizálás a megtervezett komponensek összekötésével

A szabályozók egy lehetséges megvalósítását a három megvizsgált üzemmódra a 6.34. ábra (normál állapot-visszacatolás megfigyelővel és alapjel-korrekcióval), a 6.35. ábra (integráló állapot-visszacatolás a hibajel integrálásával, megfigyelővel és alapjel-korrekcióval) és a 6.36. ábra (állapot-visszacatolás állapotmegfigyelővel, terhelésbecslővel és alapjel-korrekcióval) mutatja be. A 6.36. és 6.37. ábrákat ki kell egészíteni az $u_{i-1} = z^{-1}u_i$ taggal (Unit Delay a mintavételi idő megadásával). A folytonosidejű $r(t)$ és $y(t)$ jelek korrekt mintavételezését meg lehet oldani egy elfajuló diszkrétidejű állapotegyenlettel, lásd 6.35. ábra, ahol $A = B = C = []$ és $D = eye(2)$.

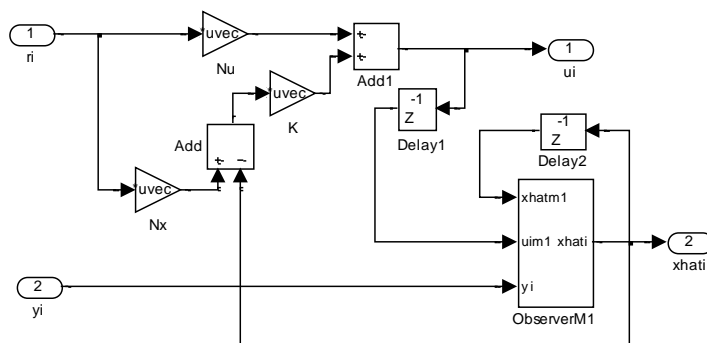
Az 1. módszert normál állapot-visszacatolás esetén a 6.38 és 6.39. ábra, integráló állapot-visszacatolás esetén pedig a 6.40. és 6.41. ábra mutatja be. A megfigyelő mindkét esetben azonos. Ügyeljünk arra, hogy a Gain Simulink blokk mintavételes legyen, és `matrix*` `vector` üzemmód legyen beállítva, és az értékeket a `Workspace` tartalmazza (`function` nem látja a `Workspace` változóit).



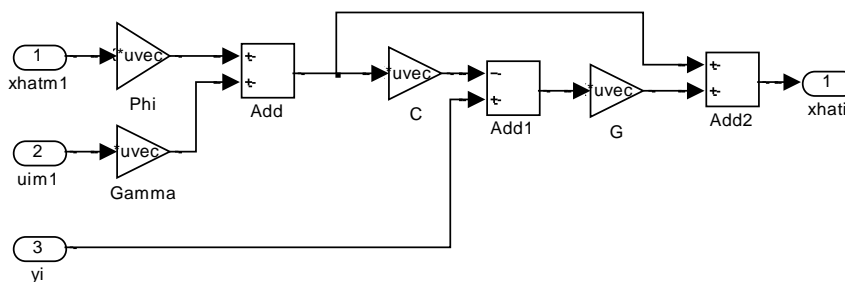
6.37. ábra. Folytonosidejű jelek mintavételezése (ADC)

A normál szabályozó Simulink megvalósítása az 1. módszerrel:

A módszer közvetlenül a 6.34. ábra szabályozó részét valósítja meg.



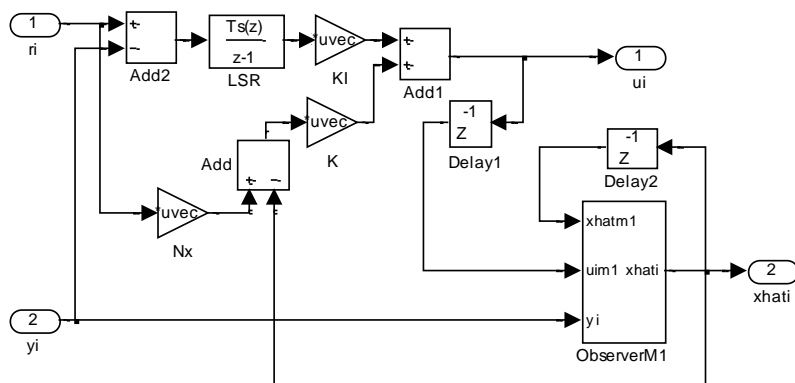
6.38. ábra. A normál szabályozó Simulink modellje (1. módszer)



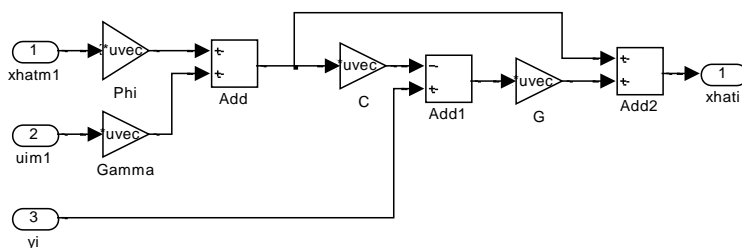
6.39. ábra. A megfigyelő Simulink modellje (1. módszer)

Az integráló szabályozó Simulink megvalósítása az 1. módszerrel:

A módszer közvetlenül a 6.35. ábra szabályozó részét valósítja meg. A hibajel integrálására a bal oldali téglalapszabályt (LSR) alkalmazza.



6.40. ábra. Az integráló szabályozó Simulink modellje (1. módszer)



6.41. ábra. A megfigyelő Simulink modellje (1. módszer)

2. módszer: Állapotteres realizálás összevont megfigyelővel és szabályozóval

A szabályozó kimenőjele $u_i = K(N_x r_i - \hat{x}_i) + N_u r_i + K_I x_{I,i} - \hat{x}_{d,i}$ alapján képezhető az általános esetben, ahol $x_{I,i+1} = x_{I,i} + T(r_i - x_i)$ része a szabályozó dinamikájának.

A megfigyelő belső állapotát jelölje $\bar{X} = (\bar{x}^T, \bar{x}_d^T)^T$, akkor kihasználva azt, hogy a Simulink előbb képezi a kimenőjelet és csak utána a következő állapotot, az S és S_d szelektormátrixok bevezetésével írhatjuk, hogy

$$\begin{aligned} \hat{X}_i &= \tilde{F}\hat{X}_{i-1} + \tilde{G}y_i + \tilde{H}u_{i-1} = \\ &= (\tilde{\Phi} - \tilde{G}\tilde{C}\tilde{\Phi})\hat{X}_{i-1} + \tilde{G}y_i + (\tilde{\Gamma} - \tilde{G}\tilde{C}\tilde{\Gamma})u_{i-1} = \\ &= \tilde{\Phi}\hat{X}_{i-1} + \tilde{\Gamma}u_{i-1} + \tilde{G}\{y_i - \tilde{C}(\tilde{\Phi}\hat{X}_{i-1} + \tilde{\Gamma}u_{i-1})\} \Rightarrow \end{aligned}$$

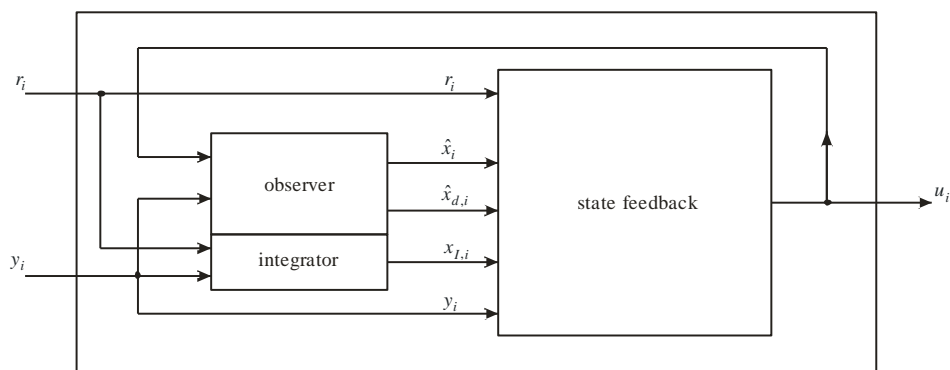
$$\begin{aligned}\bar{X}_i &= \tilde{\Phi} \hat{X}_{i-1} + \tilde{\Gamma} u_{i-1} \\ \hat{X}_i &= \bar{X}_i + \tilde{G}\{y_i - \tilde{C}\bar{X}_i\},\end{aligned}\quad (6.53)$$

ahonnan a megfigyelő a Matlab-ban megszokott standard alakra hozható:

$$\begin{aligned}\bar{X}_{i+1} &= \tilde{\Phi} \hat{X}_i + \tilde{\Gamma} u_i = \tilde{\Phi}\{\bar{X}_i + \tilde{G}[y_i - \tilde{C}\bar{X}_i]\} + \tilde{\Gamma} u_i = \\ &= (\tilde{\Phi} - \tilde{\Phi}\tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \tilde{\Gamma} u_i = \\ &= \tilde{\Phi}(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \tilde{\Gamma} u_i \\ \hat{X}_i &= (I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{G}y_i \\ \hat{x}_i &= S \hat{X}_i = S(I - \tilde{G}\tilde{C})\bar{X}_i + S\tilde{G}y_i \\ \hat{x}_{d,i} &= S_d \hat{X}_i = S_d(I - \tilde{G}\tilde{C})\bar{X}_i + S_d\tilde{G}y_i\end{aligned}\quad (6.54)$$

A (6.54) egyenletben az első egyenlet az állapotegyenlet, az utolsó három pedig a kimeneti leképezés. Az állapotváltozó \bar{X} , \hat{X} pedig kimenőjel. Jól látható továbbá, hogy a Simulink adottságainál fogva az állapotteres megoldás egyszerre kéri bemenetként u_i értékét, miközben u_i számításához szükség van \hat{x}_i , $\hat{x}_{d,i}$ értékére.

A szabályozó elvi hatásvázlatát a 6.42. ábra mutatja.



6.42. ábra. Az megfigyelő és az állapot-visszacsatolás elvi hatásvázlata

Az egyik lehetőség a két komponens külön-külön történő megvalósítása, majd összekötése a CST szolgáltatásaival (lásd `connect`, az összekötendő a jelek nevének megadásával), és az összekötés eredményeként $\Sigma_c = (A_c, B_c, C_c, D_c)$ előállítás.

A másik lehetőség a komplex szabályozó állapotegyenletének egyszerre történő felírása, és $\Sigma_c = (A_c, B_c, C_c, D_c)$ előállítását egyetlen lépésben külső összeköttetés nélkül, ugyanis \bar{X} állapotegyenletében u_i helyére behelyettesíthető

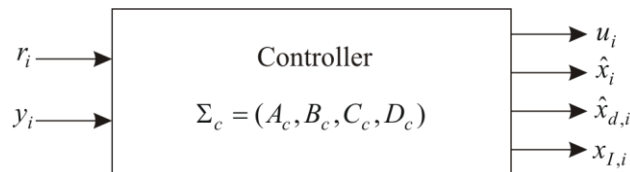
$$\begin{aligned} u_i &= K(N_x r_i - \hat{x}_i) + N_u r_i + K_I x_{I,i} - \hat{x}_{d,i} = \\ &= (KN_x + N_u)r_i - (KS + S_d)\{(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{G}y_i\} + K_I x_{I,i} = \\ &= -(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i, \end{aligned} \quad (6.55)$$

ahonnan kapjuk, hogy

$$\begin{aligned} \bar{X}_{i+1} &= \tilde{\Phi}(I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{\Phi}\tilde{G}y_i + \\ &+ \tilde{\Gamma}\{-(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i\} \Rightarrow \\ \bar{X}_{i+1} &= [\tilde{\Phi} - \tilde{\Gamma}(KS + S_d)](I - \tilde{G}\tilde{C})\bar{X}_i + \tilde{\Gamma}K_I x_{I,i} + \\ &+ \tilde{\Gamma}(KN_x + N_u)r_i + [\tilde{\Phi} - \tilde{\Gamma}(KS + S_d)]\tilde{G}y_i = \\ &= A_{c11}\bar{X}_i + A_{c12}x_{I,i} + B_{c11}r_i + B_{c12}y_i \\ x_{I,i+1} &= x_{I,i} + T(r_i - y_i) = x_{I,i} + Tr_i - Ty_i = \\ &= A_{c22}x_{I,i} + B_{c21}r_i + B_{c22}y_i \end{aligned} \quad (6.56)$$

$$\begin{aligned} u_i &= -(KS + S_d)(I - \tilde{G}\tilde{C})\bar{X}_i + K_I x_{I,i} + (KN_x + N_u)r_i - (KS + S_d)\tilde{G}y_i = \\ &= C_{c11}\bar{X}_i + C_{c12}x_{I,i} + D_{c11}r_i + D_{c12}y_i \\ \hat{x}_i &= S(I - \tilde{G}\tilde{C})\bar{X}_i + S\tilde{G}y_i = C_{c21}\bar{X}_i + D_{c22}y_i \\ \hat{x}_{d,i} &= S_d(I - \tilde{G}\tilde{C})\bar{X}_i + S_d\tilde{G}y_i = C_{c31}\bar{X}_i + D_{c32}y_i \\ x_{I,i} &= x_{I,i} = C_{c4,2}x_{I,i} \end{aligned} \quad (6.57)$$

A nulla blokkokat az állapotegyenletben nem tüntettük fel. Az állapotmegfigyelő és az állapot-visszacsatolás összevonása után keletkezett kompakt szabályozót a 6.43. ábrán mutatjuk be. A szabályozó egy diszkrétidejű dinamikus rendszer, amelynek bemenőjelei r_i , y_i , kimenőjelei pedig u_i és a becsült állapotok, szükség esetén kiegészítve a hibajel integráljával (ekkor N_u hiányzik).



6.43. ábra. Az összevont kompakt szabályozó

A $\Sigma_c = (A_c, B_c, C_c, D_c)$ komplex szabályozó állapotegyenlete mátrixainak meghatározására külön függvény készült, amelynek prototípus alakja

```
function [Ac,Bc,Cc,Dc]=Controller_M2(ctype,Ts,Phi,Gamma,C,G,K,Nx,Nu)
```

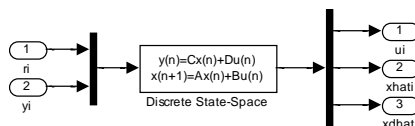
A függvény m-fájlként rendelkezésre áll. Hívásai a normal, integráló és terhelésbecslő üzemmódban rendre a következők:

```
[Ac,Bc,Cc,Dc]=Controller_M2('normal',Ts,Phi,Gamma,C,G,K,Nx,Nu)
```

```
[Ac,Bc,Cc,Dc]=Controller_M2('integral',Ts,Phi,Gamma,C,G,[K KI],Nx,Nu)
```

```
[Ac,Bc,Cc,Dc]=Controller_M2('load',Ts,Phi,Gamma,Ct,Gt,K,Nx,Nu)
```

Híváskor a bemeneti változóknak ott kell lenniük a Workspace területen (a tervezés lépései már lezárultak). A visszatérési mátrixok beépíthetők a Simulink modellbe.



6.44. ábra. Az összevont kompakt szabályozó terhelésbecslés esetén (2. módszer)

A $\Sigma_c = (A_c, B_c, C_c, D_c)$ szabályozó leírásból a CST standard szolgáltatásaival meghatározható többek között a komplex szabályozó $D_c(z)$ diszkrétidejű átviteli függvénye is.

A tervezés és megvalósítás lépéseinek bemutatása harmadrendű rendszeren

A korábbiakhoz hasonlóan legyen a szabályozott szakasz átviteli függvénye:

$$W_p(s) = \frac{A}{(1+sT_1)(1+sT_2)(1+sT_3)}$$

$$\begin{aligned} A &= 5 \\ T_1 &= 10 \text{ sec} \\ T_2 &= 4 \text{ sec} \\ T_3 &= 1 \text{ sec} \end{aligned}$$

Megkötés, hogy az állapotváltozók rendre az egytárolós tagok kimenetei az időállandók sorrendjében, a rendszer fizikai felépítésének megfelelően!

Mivel az $Y(s) = \frac{A}{1+sT}U(s)$ tag állapotegyenlete $TsY(s) + Y(s) = AU(s)$ miatt

$x = y$ állapotválasztással $\dot{x} = -\frac{1}{T}x + \frac{A}{T}u$, ezért a szabályozott szakasz állapotegyenlete, figyelembe véve, hogy $u_2 = x_1$, $u_3 = x_2$ és az A erősítés közvetlenül a bemenetre hat, a következő:

$$\frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} -1/T_1 & 0 & 0 \\ 1/T_2 & -1/T_2 & 0 \\ 0 & 1/T_3 & -1/T_3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} A/T_1 \\ 0 \\ 0 \end{pmatrix} u \quad (6.58)$$

$$y = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

A tervezés lépéseit egy-egy script fájlban adjuk meg, ahol $n = \dim x$ és $m = \dim y$.

Normál szabályozás

A normál szabályozás a T123_Normal.m script fájlban lett megvalósítva.

```
%T123_Normal.m
```

```
clear all
close all
clc
```

```
%System
```

```

n=3; %dimx
m=1; %dimy=dimu
A=5; T1=10; T2=4; T3=1;
Ap=[-1/T1 0 0; 1/T2 -1/T2 0; 0 1/T3 -1/T3]
Bp=[A/T1; 0; 0]
Cp=[0 0 1]
Dp=0
sys=ss (Ap, Bp, Cp, Dp) ;
Wp=tf(sys)
nump=Wp.num{1}
denp=Wp.den{1}
Ts=0.2 %Sampling time
sysd=c2d(sys, Ts); %'zoh'
[Phi, Gamma, C, D]=ssdata (sysd)

%Specification
xi=0.7; w0=1;
s1=-xi*w0+j*sqrt(1-xi^2)*w0;
scinf=3*real(s1);
soinf=5*real(s1);
wN=pi/Ts %Nyquist-frequency
cmpwN=[abs(s1) abs(scinf) abs(soinf) wN]
z1=exp(s1*Ts)
zcinf=exp(scinf*Ts)
zoinf=exp(soinf*Ts)

%-----
%Normal control
%-----
fprintf('Normal Control\n')
%Controller
pcl=[z1 conj(z1)]';
if n>2, pcl=[pcl; repmat(zcinf,n-2,1)]; end;
pcl
K=acker(Phi, Gamma, pcl)
%Actual observer
pobs=repmat(zoinf, n, 1)
KII=acker(Phi', Phi'*C', pobs);
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma
%Reference signal correction
Nxu=inv([Phi-eye(n) Gamma; C 0])*[zeros(n,m); eye(m)];
Nx=Nxu(1:n,:);
Nu=Nxu(n+1:n+m,:);

%Simulation M1
open_system('T123NormalCl_M1')
fprintf('T123NormalCl_M1.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');

```

```

%Activate Scopes
%pause

% %Simulation M2
%
[Ac,Bc,Cc,Dc]=Controller_M2('normal',Ts,Phi,Gamma,C,G,K,Nx,
Nu)
% open_system('T123NormalCl_M2')
% fprintf('T123NormalCl_M2.mdl has been loaded\n');
% fprintf('Activate Scopes and Start Simulation\n');
% %Activate Scopes
% %pause

```

A T123_Normal.m script indítása után a következő eredményeket kaptuk:

Ap =

```

-0.1000    0    0
 0.2500  -0.2500    0
 0    1.0000  -1.0000

```

Bp =

```

 0.5000
 0
 0

```

Cp =

```

 0    0    1

```

Dp =

```

 0

```

Transfer function:

```

 0.125

```

```

-----
s^3 + 1.35 s^2 + 0.375 s + 0.025

```

nump =

```

 0    0    0    0.1250

```

denp =

1.0000 1.3500 0.3750 0.0250

Ts =

0.2000

Phi =

0.9802 0 0
0.0483 0.9512 0
0.0046 0.1767 0.8187

Gamma =

0.0990
0.0024
0.0002

C =

0 0 1

D =

0

wN =

15.7080

cmpwN =

1.0000 2.1000 3.5000 15.7080

z1 =

0.8605 + 0.1237i

zcinf =

0.6570

zoinf =

0.4966

Normal Control

pcl =

0.8605 - 0.1237i
0.8605 + 0.1237i
0.6570

K =

3.6048 5.9792 3.8404

pobs =

0.4966
0.4966
0.4966

G =

13.3214
2.8986
0.8396

F =

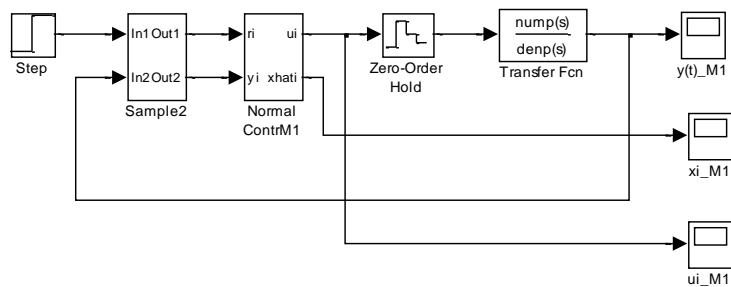
0.9193 -2.3534 -10.9067
0.0350 0.4391 -2.3732
0.0007 0.0283 0.1313

H =

0.0969
0.0020
0.0000

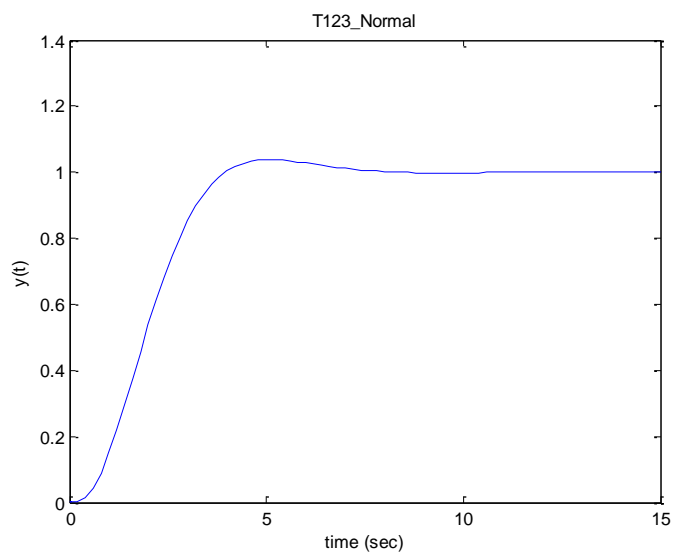
T123NormalCl_M1.mdl has been loaded
Activate Scopes and Start Simulation

A normál szabályozóval felépítettük a zárt rendszer Simulink modelljét az 1. módszer szerint, lásd 6.45. ábra. NormalContrM1 és abban ObserverM1 a 6.38. és 6.39. ábra szerint lett megvalósítva.

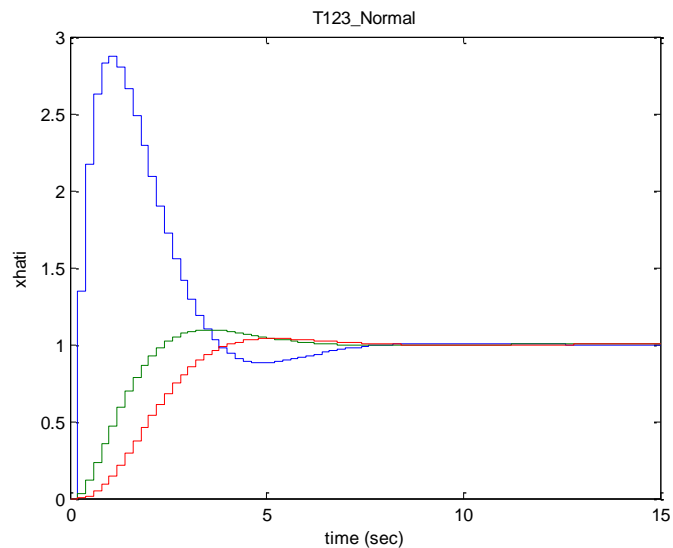


6.45. ábra. A zárt rendszer Simulink modellje normál szabályozó esetén (1. módszer)

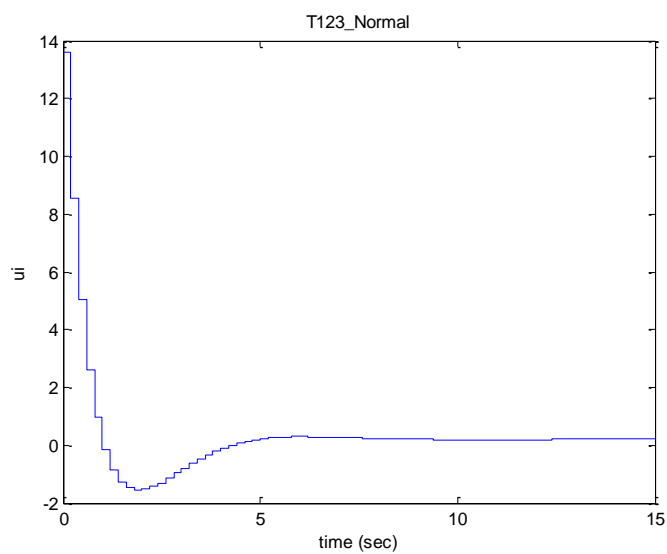
A zárt rendszer tranziensei a következőképp alakultak:



6.46. ábra. A zárt rendszer kimenőjele normál szabályozó esetén (1. módszer)



6.47. ábra. A zárt rendszer becsült állapotváltozói normál szabályozó esetén (1. módszer)



6.48. ábra. A szabályozó kimenőjele normál szabályozó esetén (1. módszer)

Integráló szabályozás

Az integráló szabályozás a T123_Integral.m script fájlban lett megvalósítva.

```
%T123_Integral.m

clear all
close all
clc

%T123_Normal.m

clear all
close all
clc

%System
n=3; %dimx
m=1; %dimy=dimu
A=5; T1=10; T2=4; T3=1;
Ap=[-1/T1 0 0; 1/T2 -1/T2 0; 0 1/T3 -1/T3]
Bp=[A/T1; 0; 0]
Cp=[0 0 1]
Dp=0
sys=ss(Ap,Bp,Cp,Dp);
Wp=tf(sys)
nump=Wp.num{1}
denp=Wp.den{1}
Ts=0.2 %Sampling time
sysd=c2d(sys,Ts); %'zoh'
[Phi,Gamma,C,D]=ssdata(sysd)

%Specification
xi=0.7; w0=1;
s1=-xi*w0+j*sqrt(1-xi^2)*w0;
scinf=3*real(s1);
soinf=5*real(s1);
wN=pi/Ts %Nyquist-frequency
cmpwN=[abs(s1) abs(scinf) abs(soinf) wN]
z1=exp(s1*Ts)
zcinf=exp(scinf*Ts)
zoinf=exp(soinf*Ts)

%-----
%Integrating control
%-----
fprintf('Integrating Control\n')
%State augmenting
Phit=[Phi zeros(n,m); Ts*C eye(m)] %t instead of tilde
Gammat=[Gamma; zeros(m,m)]
```



```

Ct=[C zeros(m,m)]
Dt=0
%Controller
pclt=[z1 conj(z1)]';
pclt=[pclt; repmat(zcinf,n-1,1)]
Kt=acker(Phit,Gammat,pclt);
K=Kt(:,1:n)
KI=Kt(:,n+1:n+m)
%Actual observer
pobs=repmat(zoinf,n,1)
KII=acker(Phi',Phi'*C',pobs);
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma
%Reference signal correction
Nxu=inv([Phi-eye(n) Gamma; C 0])*[zeros(n,m); eye(m)];
Nx=Nxu(1:n,:);
Nu=Nxu(n+1:n+m,:);

%Simulation_M1
open_system('T123IntegralCl_M1')
fprintf('T123IntegralCl_M1.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause

% %Simulation M2
%
[Ac,Bc,Cc,Dc]=Controller_M2('integral',Ts,Phi,Gamma,C,G,[K
KI],Nx,Nu)
% open_system('T123NormalCl_M2')
% fprintf('T123IntegralCl_M2.mdl has been loaded\n');
% fprintf('Activate Scopes and Start Simulation\n');
% %Activate Scopes
% %pause

```

A T123_Integral.m script indítása után a következő eredményeket kaptuk:

Ap =

```

-0.1000    0    0
 0.2500  -0.2500    0
    0    1.0000  -1.0000

```

Bp =

```

0.5000
    0
    0

```

Cp =

0 0 1

Dp =

0

Transfer function:

0.125

s^3 + 1.35 s^2 + 0.375 s + 0.025

nump =

0 0 0 0.1250

denp =

1.0000 1.3500 0.3750 0.0250

Ts =

0.2000

Phi =

0.9802 0 0
0.0483 0.9512 0
0.0046 0.1767 0.8187

Gamma =

0.0990
0.0024
0.0002

C =

0 0 1

D =

0

wN =

15.7080

cmpwN =

1.0000 2.1000 3.5000 15.7080

z1 =

0.8605 + 0.1237i

zcinf =

0.6570

zoinf =

0.4966

Integrating Control

Phit =

0.9802	0	0	0
0.0483	0.9512	0	0
0.0046	0.1767	0.8187	0
0	0	0.2000	1.0000

Gammat =

0.0990
0.0024
0.0002
0

Ct =

0 0 1 0

Dt =

0

pclt =

0.8605 - 0.1237i
0.8605 + 0.1237i
0.6570
0.6570

K =

6.4227 30.9868 22.3513

KI =

23.3627

pobs =

0.4966
0.4966
0.4966

G =

13.3214
2.8986
0.8396

F =

0.9193 -2.3534 -10.9067
0.0350 0.4391 -2.3732
0.0007 0.0283 0.1313

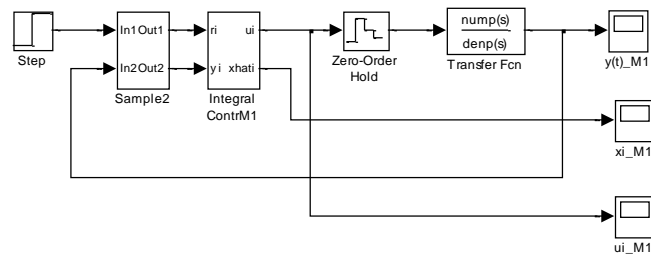
H =

0.0969
0.0020
0.0000

T123IntegralCl_M1.mdl has been loaded

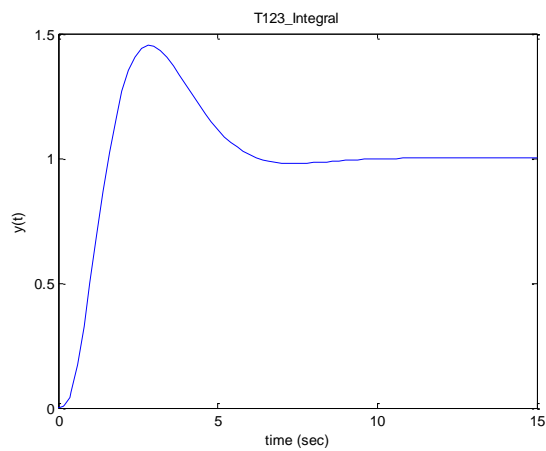
Activate Scopes and Start Simulation

Az integráló szabályozóval felépítettük a zárt rendszer Simulink modelljét az 1. módszer szerint, lásd 6.49. ábra. IntegralContrM1 és abban ObserverM1 a 6.40. és 6.41. ábra szerint lett megvalósítva.

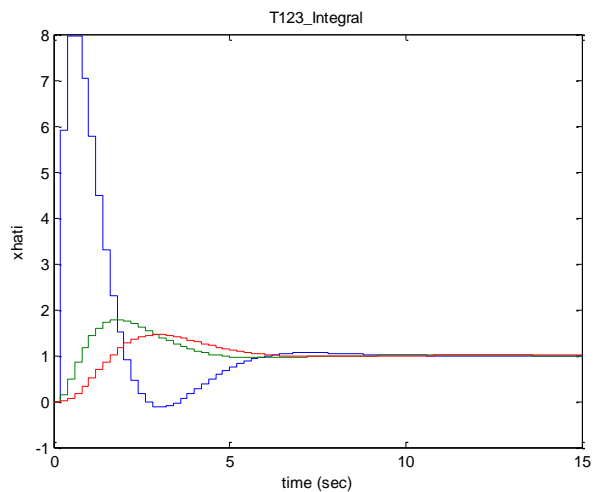


6.49. ábra. A zárt rendszer Simulink modellje integráló szabályozó esetén (1. módszer)

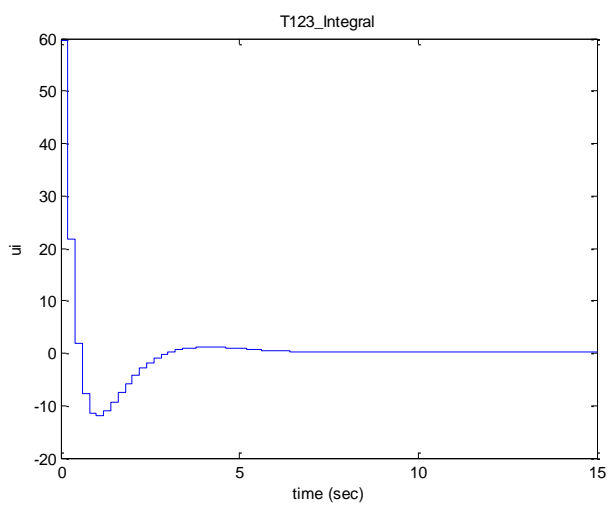
A zárt rendszer transziensei a következőképp alakultak:



6.50. ábra. A zárt rendszer kimenőjele integráló szabályozó esetén (1. módszer)



6.51. ábra. A becsült állapotváltozók integráló szabályozó esetén (1. módszer)



6.52. ábra. A szabályozó kimenőjele integráló szabályozó esetén (1. módszer)

Terhelésbecslő szabályozás

A terhelésbecslő szabályozás a T123_Load.m script fájlban lett megvalósítva.

```
%T123_Load.m
```

```
clear all  
close all
```

```

clc

%T123_Normal.m

clear all
close all
clc

%System
n=3; %dimx
m=1; %dimy=dimu
A=5; T1=10; T2=4; T3=1;
Ap=[-1/T1 0 0; 1/T2 -1/T2 0; 0 1/T3 -1/T3]
Bp=[A/T1; 0; 0]
Cp=[0 0 1]
Dp=0
sys=ss(Ap,Bp,Cp,Dp);
Wp=tf(sys)
nump=Wp.num{1}
denp=Wp.den{1}
Ts=0.2 %Sampling time
sysd=c2d(sys,Ts); %'zoh'
[Phi,Gamma,C,D]=ssdata(sysd)

%Specification
xi=0.7; w0=1;
s1=-xi*w0+j*sqrt(1-xi^2)*w0;
scinf=3*real(s1);
soinf=5*real(s1);
wN=pi/Ts %Nyquist-frequency
cmpwN=[abs(s1) abs(scinf) abs(soinf) wN]
z1=exp(s1*Ts)
zcinf=exp(scinf*Ts)
zoinf=exp(soinf*Ts)

%-----
%Load estimating control
%-----
fprintf('Load Estimating Control\n')
%State augmenting
Phit=[Phi Gamma; zeros(m,n) eye(m)]; %t instead of tilde
Gammat=[Gamma; zeros(m,m)];
Ct=[C zeros(m,m)];
Dt=zeros(m,m);
%Controller
pcl=[z1 conj(z1)]';
if n>2, pcl=[pcl; repmat(zcinf,n-2,1)]; end;
pcl
K=acker(Phi,Gamma,pcl)

```

```

%Actual observer
ptobs=repmat(zoinf,n+m,1)
KtII=acker(Phit',Phit'*Ct',ptobs);
Gt=KtII'
Ft=Phit-Gt*Ct*Phit
Ht=Gammat-Gt*Ct*Gammat
%Reference signal correction
Nxu=inv([Phi-eye(n) Gamma; C zeros(m,m)])*[zeros(n,m);
eye(m)];
Nx=Nxu(1:n,:);
Nu=Nxu(n+1:n+m,:);

%Complex Controller
[Ac,Bc,Cc,Dc]=Controller_M2('load',Ts,Phit,Gammat,Ct,Gt,K,N
x,Nu)
syscx=ss(Ac,Bc,Cc,Dc,Ts);
Dcx=tf(syscx); %Controller transfer function
if m==1 %SISO
    numc11=Dcx.num{1,1};
    numc12=Dcx.num{1,2};
    denc11=Dcx.den{1,1};
    denc12=Dcx.den{1,2};
    fprintf('Controller transfer functions:\n');
    fprintf('\n');
    fprintf('Dcur(z):\n');
    Dcur=tf(numc11,denc11,Ts)
    fprintf('\n');
    fprintf('Dcuy(z):\n');
    Dcuy=tf(numc12,denc12,Ts)
else %MIMO
    fprintf('Complex controller transfer functions\nLeading
terms belong to u\n');
    Dcuxxd=tf(syscx)
end;

%Simulation
open_system('T123LoadCl_M2')
fprintf('T123LoadCl_M2.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause

```

A T123_Integral.m script indítása után a következő eredményeket kaptuk:

Ap =

```

-0.1000    0    0
 0.2500  -0.2500    0
    0    1.0000  -1.0000

```


Bp =

```

0.5000
  0
  0

```

Cp =

```

0  0  1

```

Dp =

```

0

```

Transfer function:

```

0.125
-----

```

```

s^3 + 1.35 s^2 + 0.375 s + 0.025

```

nump =

```

0  0  0  0.1250

```

denp =

```

1.0000  1.3500  0.3750  0.0250

```

Ts =

```

0.2000

```

Phi =

```

0.9802  0  0
0.0483  0.9512  0
0.0046  0.1767  0.8187

```

Gamma =

```

0.0990
0.0024

```

0.0002

C =

0 0 1

D =

0

wN =

15.7080

cmpwN =

1.0000 2.1000 3.5000 15.7080

z1 =

0.8605 + 0.1237i

zcinf =

0.6570

zoinf =

0.4966

Load Estimating Control

pcl =

0.8605 - 0.1237i
0.8605 + 0.1237i
0.6570

K =

3.6048 5.9792 3.8404

ptobs =

0.4966
 0.4966
 0.4966
 0.4966

Gt =

41.5295
 4.5790
 0.9203
 73.3765

Ft =

0.7903	-7.3368	-34.0015	0.0925
0.0273	0.1423	-3.7490	0.0017
0.0004	0.0141	0.0652	0.0000
-0.3356	-12.9631	-60.0756	0.9886

Ht =

0.0925
 0.0017
 0.0000
 -0.0114

Nx =

1.0000
 1.0000
 1.0000

Nu =

0.2000

Ac =

0.6233	-0.5920	-23.2050	0
0.0395	0.9366	-5.9290	0
0.0040	0.1757	-0.9061	0
0	0	-73.3765	1.0000

Bc =

1.3489	22.8248
0.0333	5.9196

```

0.0021    1.7242
      0    73.3765

```

Cc =

```

-3.6048   -5.9792   250.1544   -1.0000
 1.0000         0   -41.5295         0
      0    1.0000   -4.5790         0
      0         0    0.0797         0
      0         0   -73.3765    1.0000

```

Dc =

```

13.6244  -253.9948
      0    41.5295
      0    4.5790
      0    0.9203
      0    73.3765

```

Controller transfer functions:

Dcur(z):

Transfer function:

```

13.62 z^4 - 27.06 z^3 + 20.16 z^2 - 6.674 z + 0.8285
-----
z^4 - 1.654 z^3 + 0.9826 z^2 - 0.3683 z + 0.03956

```

Sampling time: 0.2

Dcuy(z):

Transfer function:

```

-254 z^4 + 660.3 z^3 - 573.3 z^2 + 166.1 z - 2.9e-015
-----
z^4 - 1.654 z^3 + 0.9826 z^2 - 0.3683 z + 0.03956

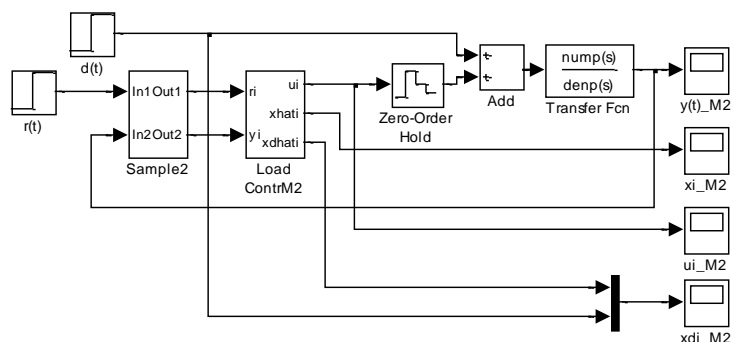
```

Sampling time: 0.2

T123LoadC1_M2.mdl has been loaded
 Activate Scopes and Start Simulation

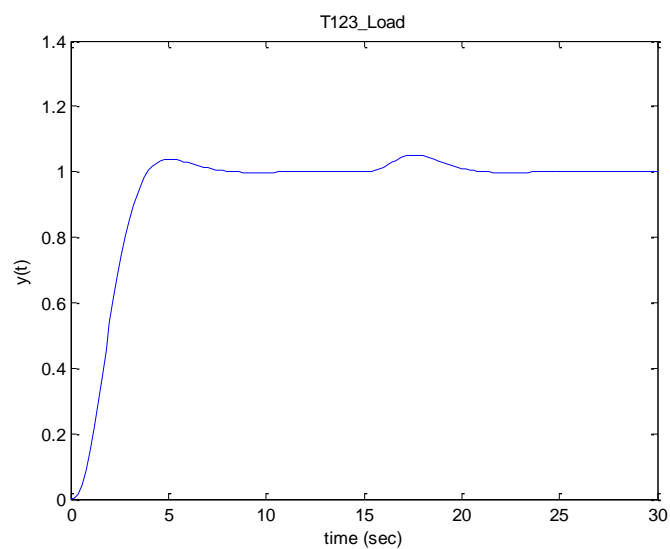
Vegyük észre, hogy a komplex szabályozó diszkrétidejű átviteli függvénye is meghatározásra került.

A terhelésbecslő szabályozóval felépítettük a zárt rendszer Simulink modelljét az 2. módszer szerint, lásd 6.53. ábra. LoadContrM1 komplex szabályozó a 6.44. ábra szerint lett megvalósítva (vigyázzunk a demultiplexer jelméreteinek megadásánál).

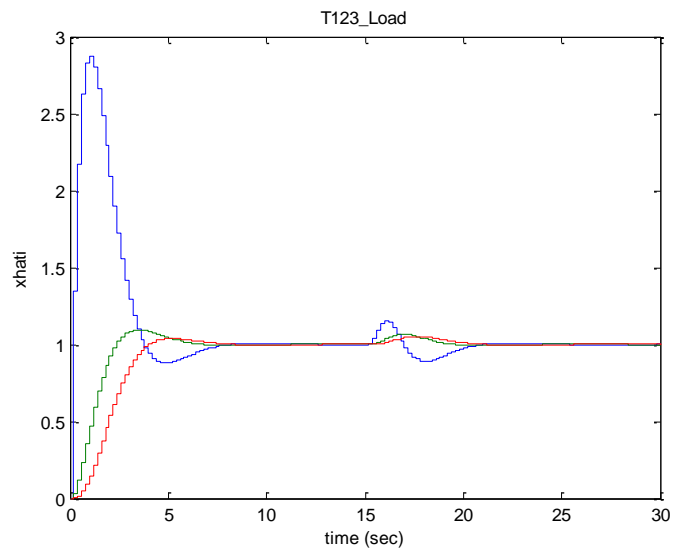


6.53. ábra. A zárt rendszer Simulink modellje terhelésbecslő szabályozó esetén (2. módszer)

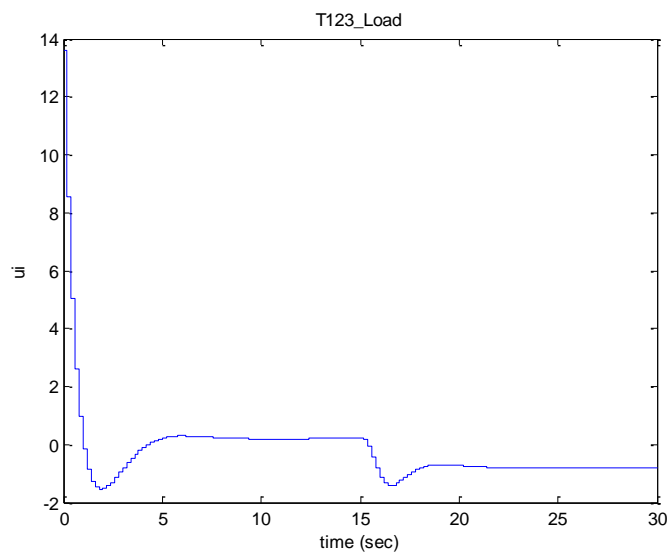
A zárt rendszer tranziensei a következőképp alakultak alapjelugrás és a tranziensek lezajlása utáni zavarójelugrás esetén:



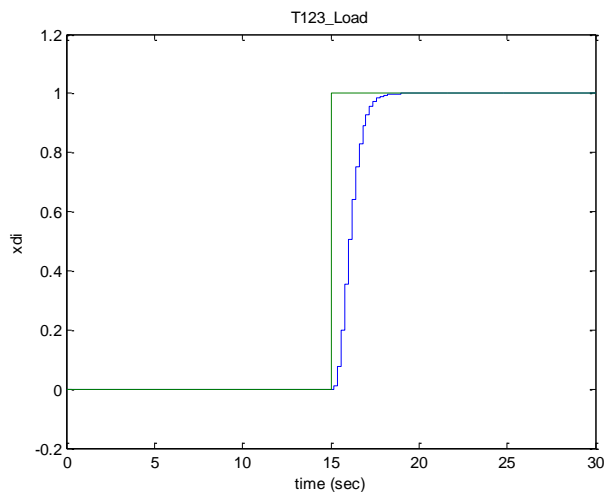
6.54. ábra. A zárt rendszer kimenőjele terhelésbecslő szabályozó esetén (2. módszer)



6.55. ábra. A rendszer becsült állapotváltozói terhelésbecslő szabályozó esetén (2. módszer)



6.56. ábra. A szabályozó kimenőjele terhelésbecslő szabályozó esetén (2. módszer)



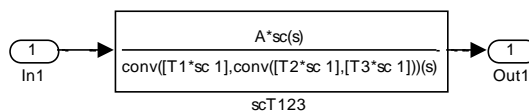
6.57. ábra. A zavarás és becslt értéke terhelésbecslő szabályozó esetén (2. módszer)

A paraméterváltozások hatásának vizsgálata terhelésbecslő esetén

A következő kódrészlet segítségével ellenőrizhető a szakaszparaméterek megváltozásának hatása, ha továbbra is a nominális rendszerhez tervezett szabályozót használjuk.

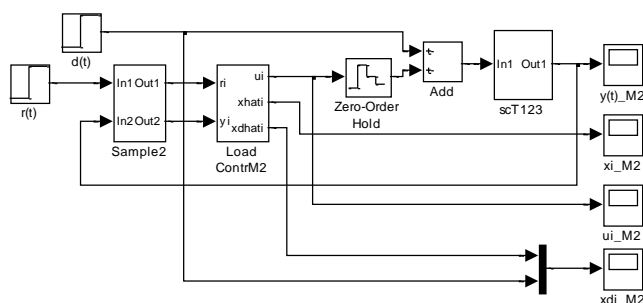
```
%Simulation with sc
sc=1 %1, 0.75, 1.25
open_system('scT123LoadC1_M2')
fprintf('scT123LoadC1_M2.mdl has been loaded\n');
fprintf('Activate Scopes and Start Simulation\n');
%Activate Scopes
%pause
```

A zárt rendszer Simulink modelljében a szakasz helyére a megváltozott rendszer modellje kerül. Az sc skálatényező a `command` prompt-ból adható meg. A vizsgált értékek $sc=0.75$ és $sc=1.25$.



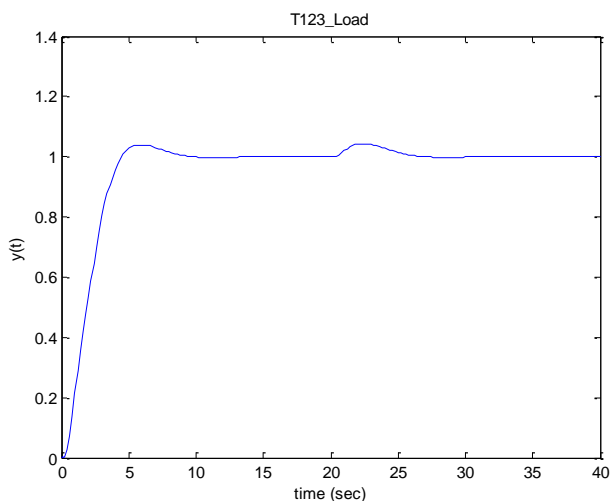
6.58. ábra. A megváltozott paraméterű szabályozott szakasz (sc skálatényező)

A zárt rendszer modelljében a zavarójel ugrás 20 sec-nál kezdődik, hogy az alapjel ugrás és a zavarójel ugrás tranziensei külön is megfigyelhetők legyenek.

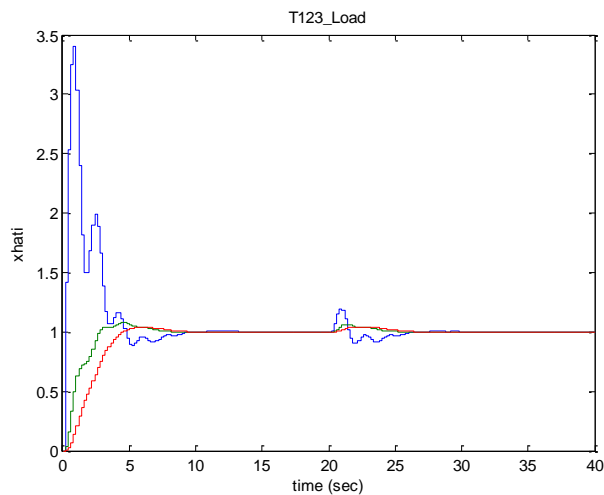


6.59. ábra. A zárt rendszer Simulink modellje a megváltozott szakasz (scT123) és a nominális terhelésbecslő esetén

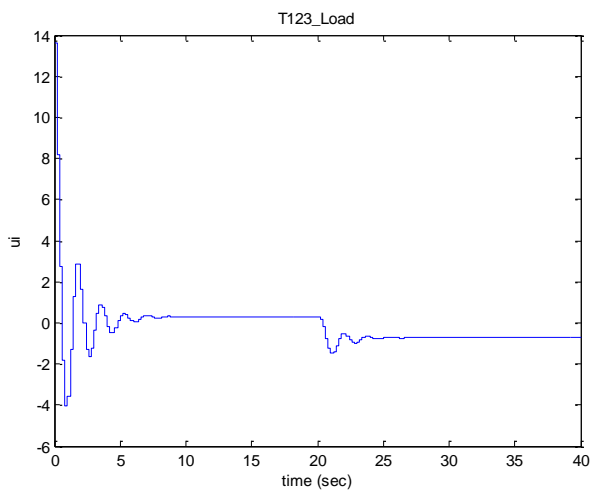
Az eredményeket $s_c=0.75$ esetén a 6.60.-6.63. ábrák, $s_c=1.25$ esetén pedig a 6.64.-6.67. ábrák mutatják be. Jól látható, hogy a vizsgált paramétertartományban a szabályozási rendszer stabil marad, és a szabályozott jellemző megfelelő. Az is látszik, hogy a nominálisnál kisebb erősítés esetén \hat{x}_d a zavarójelnél kisebb, a nominálisnál nagyobb erősítés esetén pedig a zavarójelnél nagyobb értékre áll be, kompenzálva ezáltal az erősítésváltozás hatását a szabályozott jellemzőre.



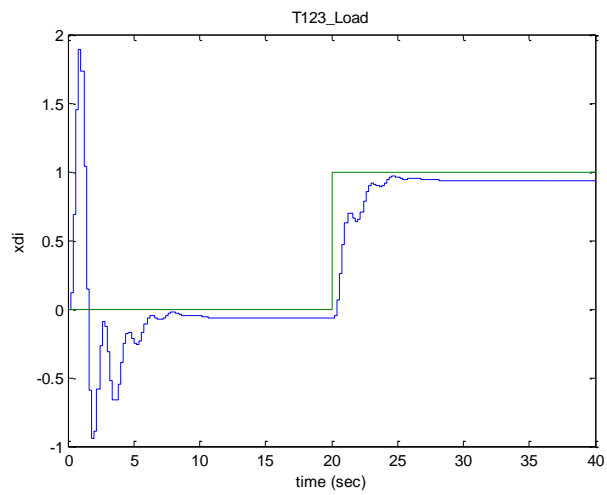
6.60. ábra. A szabályozott jellemző megváltozott szakasz ($s_c=0.75$) és nominális terhelésbecslő esetén



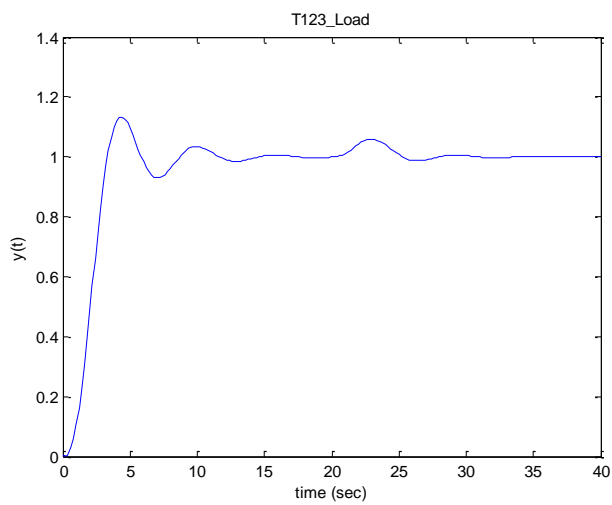
6.61. ábra. A becstt állapotvltzozok megvltzozott szakasz ($sc=0.75$) és nominális terhelésbecslő esetén



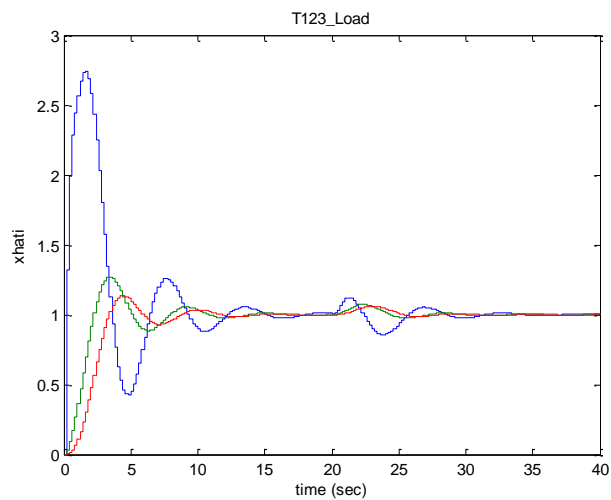
6.62. ábra. A szabályozó kimenőjele megvltzozott szakasz ($sc=0.75$) és nominális terhelésbecslő esetén



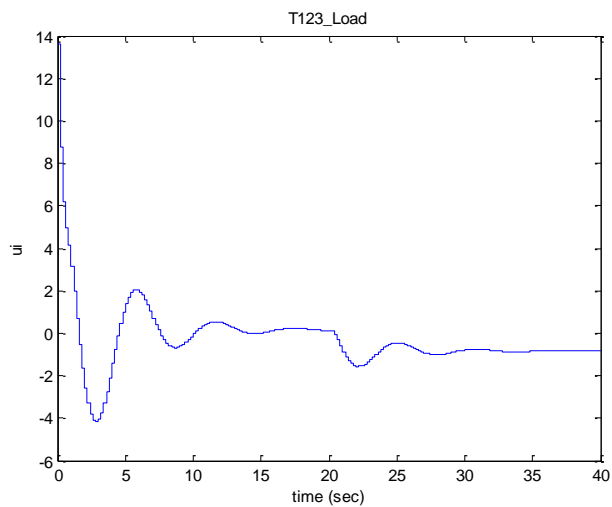
6.63. ábra. A zavaró jellemző és becslt értéke megváltozott szakasz ($sc=0.75$) és nominális terhelésbecslő esetén



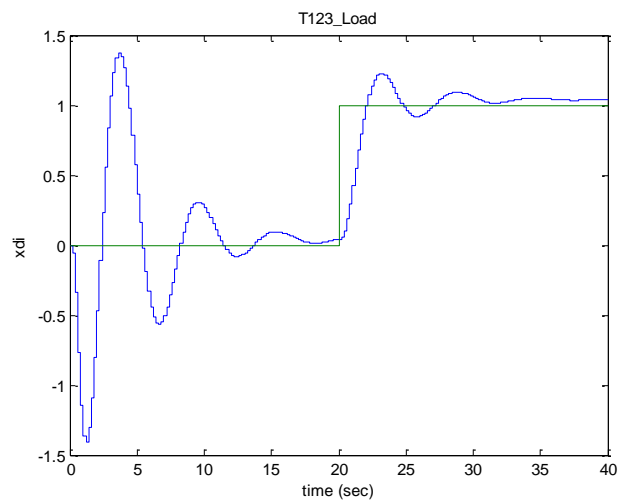
6.64. ábra. A szabályozott jellemző megváltozott szakasz ($sc=1.25$) és a nominális terhelésbecslő esetén



6.65. ábra. A becstelt állapotváltozók megváltozott szakasz ($sc=1.25$) és nominális terhelésbecslő esetén



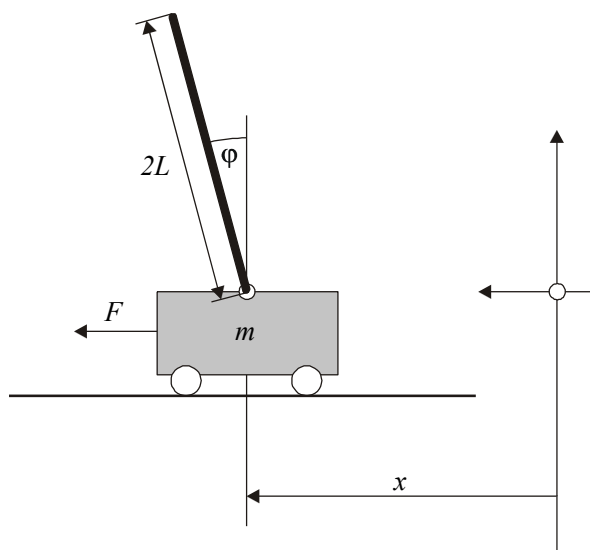
6.66. ábra. A szabályozó kimenőjele megváltozott szakasz ($sc=1.25$) és nominális terhelésbecslő esetén



6.67. ábra. A zavaró jellemző és becült értéke megváltozott szakasz ($sc=1.25$) és nominális terhelésbecslő esetén

Invertált inga diszkrétidejű szabályozása állapotér módszerrel

Az invertált inga dinamikus modelljét a 3. fejezetben a 3.11 pontban már tárgyaltuk. Az invertált inga szerkezeti vázlatát a 6.65. ábrán megismételjük.



6.65. ábra. Mozgó kocsihoz szerelt invertált inga szerkezeti vázlat

A kocsihoz szerelt rúd hossza $2L$, tömege M és a tömegközéppontra vonatkozó tehetetlenségi nyomatéka $\Theta = ML^2/3$. A kocsi tömege m , a φ szöget a függőleges helyzettől mérjük. A kocsi mozgásakor súrlódási erő is fellép, amelyet sebességfüggő viszkózus súrlódással modellezünk, értéke $b\dot{x}$. A 3.11 pontban megmutattuk, hogy a rendszer nemlineáris, differenciálegyenlete kiegészítve a súrlódási veszteséggel a következő ($C_\varphi := \cos \varphi$, $S_\varphi := \sin \varphi$ tömör jelölés):

$$\begin{aligned} (m + M)\ddot{x} + MLC_\varphi\ddot{\varphi} - MLS_\varphi\dot{\varphi}^2 + b\dot{x} &= F \\ MLC_\varphi\ddot{x} + (\Theta + ML^2)\ddot{\varphi} - MgLS_\varphi &= 0 \end{aligned} \quad (6.59)$$

Bemenőjelnek itt az F erőt fogjuk tekinteni, a kimenőjelek x és φ . Jól látható, hogy a rendszer szabadságfoka 2, ezzel szemben a bemenetek száma csak 1, vagyis a rendszer ún. alulaktuált rendszer. A nemlineáris rendszert linearizáljuk az $x = x_0, \dot{x} = 0, \varphi = 0, \dot{\varphi} = 0$ álló függőleges helyzet körül, és irányítását eme helyzet környezetében állapot-visszacsatolás, állapotmegfigyelő és alapjel korrekció bevonásával fogjuk megtervezni és ellenőrizni. A rendszer a 2 kimenet miatt nem

SISO rendszer, ezért a pólusáthelyezési feladatot nem lehet az Ackermann-képlettel megoldani. Helyette a MATLAB `CST place` függvényét fogjuk használni, amelynek azonban nem írhatók elő ismétlődő sajátértékek.

A linearizált rendszer differenciálegyenlete kis változásokra, ha a változásokat is x és φ jelöli, továbbá alkalmazzuk a $C_\varphi \approx 1$ és $S_\varphi \approx \varphi$ közelítéseket, a következő alakú lesz:

$$\begin{aligned}(m+M)\ddot{x} + ML\ddot{\varphi} + b\dot{x} &= F \\ ML\ddot{x} + (\Theta + ML^2)\ddot{\varphi} - MgL\varphi &= 0\end{aligned}\quad (6.60)$$

Figyelembe véve, hogy 2x2 méretű mátrixok esetén

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

kapjuk, hogy $\det = ad - bc = (m+M)(\Theta + ML^2) - ML \cdot ML = \Theta(m+M) + mML^2$,

$$\begin{aligned}\begin{pmatrix} \ddot{x} \\ \ddot{\varphi} \end{pmatrix} &= \frac{1}{\det} \begin{bmatrix} \Theta + ML^2 & -ML \\ -ML & m+M \end{bmatrix} \begin{pmatrix} F - b\dot{x} \\ MgL\varphi \end{pmatrix} \Rightarrow \\ \ddot{x} &= \frac{-(\Theta + ML^2)b\dot{x} - M^2L^2g\varphi + (\Theta + ML^2)F}{\det} \\ \ddot{\varphi} &= \frac{MLb\dot{x} + (m+M)MgL\varphi - MLF}{\det}\end{aligned}\quad (6.61)$$

A linearizált rendszer állapotvektora $(x, \dot{x}, \varphi, \dot{\varphi})^T$, mérhető kimenőjel vektora $(x, \varphi)^T$ és bemenete F , ezért állapotegyenletének mátrixai:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -(\Theta + ML^2)b/\det & -M^2L^2g/\det & 0 \\ 0 & 0 & 0 & 1 \\ 0 & MLb/\det & (m+M)MgL/\det & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ (\Theta + ML^2)/\det \\ 0 \\ -ML/\det \end{bmatrix}\quad (6.62)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

A következő kódrészlet $M = 0.2$, $m = 0.5$, $L = 0.3$, $\Theta = ML^2/3 = 0.006$, $b = 0.1$ esetén (minden paraméter SI egységben értendő) kiszámítja a folytonosidejű linearizált rendszer modelljét és rendszerjellemezőit.

```
%InvertedPendulum.m
%Discrete time state space control

clear all
close all
clc

%Parameters
M=0.2; %mass of pendelum
m=0.5; %mass of cart
L=0.3; %length to pendelum COG
Theta=M*L^2/3; %0.006; %inertia belonging to COG
b=0.1; %viscous friction
g=9.8; %gravity acceleration

%Continuous time LTI system
det=Theta*(m+M)+m*M*L^2
A = [0 1 0 0; ...
     0 -(Theta+M*L^2)*b/det -M^2*L^2*g/det 0; ...
     0 0 0 1; ...
     0 M*L*b/det (m+M)*M*g*L/det 0]
B = [0; (Theta+M*L^2)/det; 0; -M*L/det]
C = [1 0 0 0; 0 0 1 0]
D=zeros(2,1)
sys=ss(A,B,C,D);
W=tf(sys)
[z,p,k]=zpkdata(sys);
z1=z{1};
p1=p{1};
k1=k(1);
z2=z{2};
p2=p{2};
k2=k(2);
sys1=zpk(z1,p1,k1)
sys2=zpk(z2,p2,k2)
figure(1)
pzmap(sys1)
title('Pole-Zero Map (F\rightarrow x)')
figure(2)
pzmap(sys2)
title('Pole-Zero Map (F\rightarrow phi)')
```

A kapott eredmények (állapotegyenlet mátrixai, az $F \rightarrow x$ és $F \rightarrow \varphi$ alrendszerek átviteli függvényei, zérusai, pólusai és P/Z térképei) a következők:

A =

```

      0   1.0000   0   0
      0  -0.1818  -2.6727  0
      0   0   0   1.0000
      0   0.4545  31.1818  0

```

B =

```

      0
      1.8182
      0
     -4.5455

```

C =

```

      1   0   0   0
      0   0   1   0

```

D =

```

      0
      0

```

Transfer function from input to output...

```

      1.818 s^2 + 1.825e-013 s - 44.55
#1:  -----
      s^4 + 0.1818 s^3 - 31.18 s^2 - 4.455 s

      -4.545 s + 1.359e-016
#2:  -----
      s^3 + 0.1818 s^2 - 31.18 s - 4.455

```

Zero/pole/gain:

```

      1.8182 (s+4.95) (s-4.95)
-----
s (s-5.565) (s+5.604) (s+0.1428)

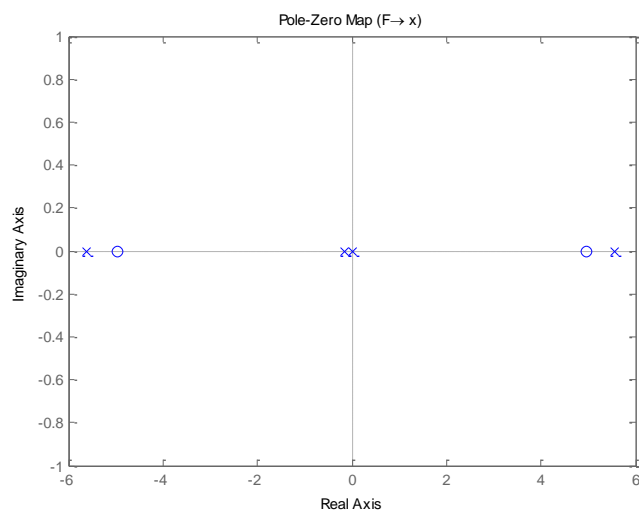
```

Zero/pole/gain:

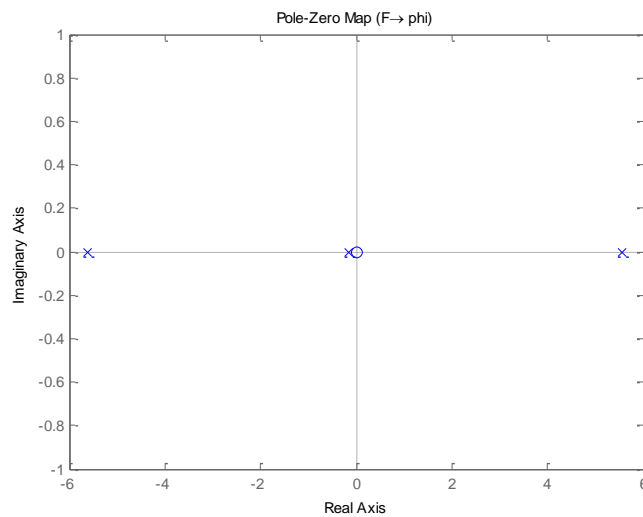
```

      -4.5455 s
-----
(s-5.565) (s+5.604) (s+0.1428)

```

6.66. ábra. Folytonosidejű szakasz pólus/zérus eloszlása ($F \rightarrow x$)



6.67. ábra. Folytonosidejű szakasz pólus/zérus eloszlása ($F \rightarrow \varphi$)

Mint az fizikailag várható volt, a folytonosidejű linearizált rendszer instabillnak adódott, ami a legjobban az átviteli függvények gyöktényezős alakjából látszik

(pólus $s = +5.565$ értéknél). Ezenkívül az is látszik, hogy az $F \rightarrow x$ első rendszernek van egy nem-minimumfázisú zérushelye is az $s = +4.95$ helyen, ami a stabilitás tulajdonságokat (a negatív fázisszög miatt) szintén rontja. Ezért a zárt rendszer eredő pólusainak megválasztásakor körültekintően kell eljárni.

Figyelembe véve, hogy az állapot-visszacsatolás nem mozdítja ki a szakasz zérushelyeit (!), a zárt rendszer 4 megválasztható pólusa közül az elsőt úgy helyezzük el, hogy $s_1 = -4.95$ legyen, ezáltal a zárt rendszerben pólus/zérus kiejtés jöjjön létre ezen a helyen. A második pólust ugyanide választjuk, tehát legyen $s_2 = -4.95$. Ezáltal biztosítjuk, hogy a zárt rendszer (!) Bode-diagramja 0 dB/dek meredekséggel folytatódik $\omega = 4.95$ után is, mivel ugyanitt van a jobb félsíkon lévő nem-minimumfázisú zérushely törésfrekvenciája is. A zárt rendszer határfrekvenciáját ezután gyorsra és aperiódikus határesetűre tervezett csillapítású konjugált komplex póluspárral állítjuk be, amelyhez $\xi = 1/\sqrt{2}$ és $\omega_0 = 3|s_1|/\xi$ értékeket választunk, vagyis $s_{3,4} = -\xi\omega_0 \pm j\omega_0\sqrt{1-\xi^2}$.

A megfigyelő számára $\text{soinf} = -5 \cdot \max(\text{abs}(\text{real}([s_1 \ s_2 \ s_3 \ s_4])))$ ideális lenne, mivel így a megfigyelő sajátértékei stabilak és gyorsabbak a zárt rendszer sajátértékeinél. Másrészt azonban a rendszer nem SISO, és így az Ackermann-képlet nem alkalmazható, ezért a MIMO rendszerekre is alkalmazható place CST-függvényt fogjuk a megfigyelő tervezésnél használni, amely azonban kizárja az ismétlődő sajátértékeket. Ezért a megfigyelő sajátértékeihez rendre hozzáadunk egy-egy kis értéket, miután a specifikációkat átszámítottuk z -be $z = e^{sT_s}$ szerint. Egyúttal ellenőrizzük, hogy a felgyorsított rendszerre is teljesül-e a Shannon-tétel előírása.

Az alábbi kódrészlet elvégzi a specifikációk meghatározását, átszámítását z -be $T_s = 0.01$ sec mintavételi idő választás mellett, és a folytonosidejű $\Sigma = (A, B, C, D)$ LTI rendszer konvertálását diszkrétidejű $\Sigma_d = (\Phi, \Gamma, C, D)$ LTI rendszerré.

```
%Specification
s1=-4.95
s2=s1
xi=1/sqrt(2), w0=3*abs(s1)/xi
s3=-xi*w0+j*w0*sqrt(1-xi^2)
s4=conj(s3)
soinf=-5*max(abs(real([s1 s2 s3 s4])))
Ts=0.01
fprintf('Comparison of eigenvalues to Nyquist-frequency:\n');
[abs([s1 s2 s3 s4 soinf]) pi/Ts]
%
z1=exp(s1*Ts)
z2=exp(s2*Ts)
```

```
z3=exp(s3*Ts)
z4=exp(s4*Ts)
zoinf=exp(soinf*Ts)

%discrete time LTI system
sysd=c2d(sys,Ts,'zoh');
[Phi,Gamma,C,D,Ts]=ssdata(sysd)
```

Az eredmények a következők:

```
s1 =
    -4.9500

s2 =
    -4.9500

xi =
    0.7071

w0 =
    21.0011

s3 =
    -14.8500 +14.8500i

s4 =
    -14.8500 -14.8500i

soinf =
    -74.2500

Ts =
    0.0100
```

Comparison of eigenvalues to Nyquist-frequency:

ans =

4.9500 4.9500 21.0011 21.0011 74.2500 314.1593

z1 =

0.9517

z2 =

0.9517

z3 =

0.8525 + 0.1275i

z4 =

0.8525 - 0.1275i

zoinf =

0.4759

Phi =

1.0000	0.0100	-0.0001	-0.0000
0	0.9982	-0.0267	-0.0001
0	0.0000	1.0016	0.0100
0	0.0045	0.3119	1.0016

Gamma =

0.0001
0.0182
-0.0002
-0.0454

C =

1	0	0	0
0	0	1	0

D =

```
0
0
```

Ezután a következő kóddal elvégeztük a K állapot-visszacsatolás megtervezését az acker CST-függvénnyel, és az $F \rightarrow x$ rendszerhez (amelyhez C és D első sora tartozik) az alapjel miatti N_x , N_u korrekció meghatározását. Világos ugyanis, hogy az $F \rightarrow \varphi$ alrendszer differenciáló jellegű, alapjelváltásnál végértéke a helyes nulla szögérték (függetlenes invertált inga helyzet) és így azzal nem kell foglalkozni. Felrajzoltuk a kimenetek tranzienseit is a zárt rendszerben mérhető állapotok esetén.

```
%State feedback design
zc=[z1 z2 z3 z4] '
K=acker(Phi,Gamma,zc)

%Nx,Nu design
Cx=C(1,:); Dx=D(1,:);
Nxu=inv([Phi-eye(4) Gamma; Cx Dx])*[zeros(4,1); 1];
Nx=Nxu(1:4)
Nu=Nxu(5)
Gff=K*Nx+Nu

%Closed loop simulation with measured state
tt=[0:0.01:2]';
FF=0.2*ones(size(tt,1),1);
[yy,xx]=dlsim(Phi-Gamma*K,Gamma*(K*Nx+Nu),C,D,FF);
figure(3)
stairs(tt,yy)
title('x and phi in cloed loop (with measured state)')
legend('x','phi','Location','Best')
```

Az eredmények a következők:

zc =

```
0.9517
0.9517
0.8525 - 0.1275i
0.8525 + 0.1275i
```

K =

```
-199.1915 -95.0564 -228.6709 -45.9084
```

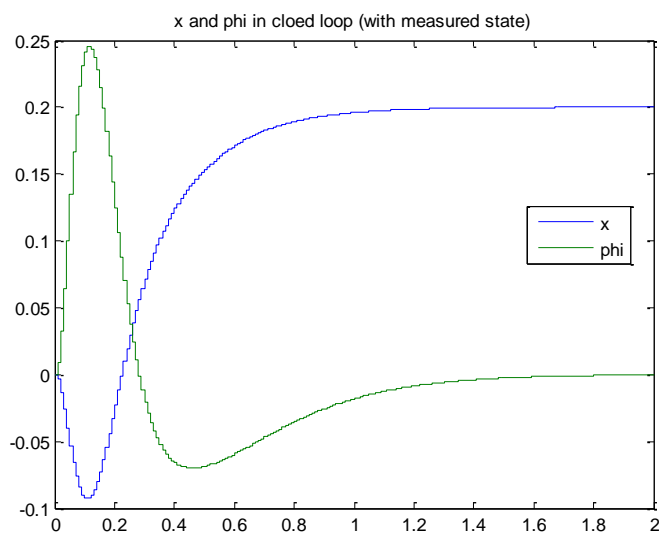
$N_x =$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$N_u =$

$$0$$

$G_{ff} =$

$$-199.1915$$


6.68. ábra. A zárt rendszer kimenőjelei mérhető állapotok esetén

Ezután az alábbi kódrészlettel meghatároztuk a megfigyelő mátrixait, majd ezt követően az állapot-visszacsatolásból, megfigyelőből és alapjel korrekcióból álló teljes diszkrétidejű $\Sigma_c = (A_c, B_c, C_c, D_c)$ szabályozó állapotegyenletének mátrixait. Ez utóbbi kódrészlet egyrészt egyszerűbb, mint a Controller_M2 függvény (de amely itt változatlan formában $\dim y \neq \dim u$ miatt nem lenne alkalmazható),

másrészt viszont a normal esetre elvégzi azokat a minimális korrekciókat, amelyek lehetővé teszik alkalmazását az alulaktuált esetben is.

```
%Observer design
zo=[zoinf zoinf+0.01 zoinf+0.02 zoinf+0.03]'
%repeated eigenvalues are forbidden for place
%MIMO -> acker cannot be used
KII=place(Phi',Phi'*C',zo)
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma

%Don't use the following function
%[Ac,Bc,Cc,Dc]=Controller_M2('normal',Ts,Phi,Gamma,C,G,K,Nx,Nu)

%Resulting controller for dimy~=dimu
nbar=size(Phi,1);
dimu=size(Gamma,2);
Ac=(Phi-Gamma*K)*(eye(nbar)-G*C);
Bc1=Gamma*(K*Nx+Nu);
Bc2=(Phi-Gamma*K)*G;
Cc1=-K*(eye(nbar)-G*C);
Dc11=K*Nx+Nu;
Dc12=-K*G;
Cc2=eye(nbar)-G*C;
Dc21=zeros(nbar,dimu);
Dc22=G;
Bc=[Bc1 Bc2];
Cc=[Cc1; Cc2];
Dc=[Dc11 Dc12; Dc21 Dc22];
Ac, Bc, Cc, Dc
sysc=ss(Ac,Bc,Cc,Dc,Ts);
```

A megfigyelő és a komplett szabályozó állapotegyenleteinek mátrixai a következők:

zo =

```
0.4759
0.4859
0.4959
0.5059
```

KII =

```
0.7591    25.8490    0.0057    0.7629
0.0052     0.5180    0.7585    26.0348
```

G =

0.7591	0.0052
25.8490	0.5180
0.0057	0.7585
0.7629	26.0348

F =

0.2409	0.0024	-0.0052	-0.0001
-25.8490	0.7399	-0.5421	-0.0053
-0.0057	-0.0001	0.2418	0.0024
-0.7629	-0.0037	-25.7634	0.7411

H =

0.0000
0.0159
-0.0001
-0.0396

Ac =

-0.2396	0.0186	-0.1185	0.0042
-70.2293	2.7251	-22.1440	0.8339
0.5417	-0.0216	0.2518	-0.0004
110.2309	-4.3145	28.0795	-1.0844

Bc =

-0.0181	1.2577	0.1392
-3.6188	73.8481	26.2716
0.0453	-0.5869	0.6978
9.0506	-119.2815	-38.1576

Cc =

1.0e+003 *			
-2.4455	0.0951	-1.1903	0.0459
0.0002	0	-0.0000	0
-0.0258	0.0010	-0.0005	0
-0.0000	0	0.0002	0
-0.0008	0	-0.0260	0.0010

Dc =

1.0e+003 *		
-0.1992	2.6447	1.4189


```

0      0.0008      0.0000
0      0.0258      0.0005
0      0.0000      0.0008
0      0.0008      0.0260

```

Ezután a $\Sigma_c = (A_c, B_c, C_c, D_c)$ szabályozó (sysc) és a $\Sigma = (\Phi, \Gamma, C, D)$ szakasz (sysd) számára megadtuk a bemenőjelek és kimenőjelek neveit, majd név alapján a CST connect függvényével előállítottuk a zárt szabályozási rendszert a két alrendszer összekötésével.

```

%Connection by name
sysc.InputName={'r','x','phi'};
sysc.OutputName={'F','xhat1','xhat2','xhat3','xhat4'};
sysd.InputName={'F'};
sysd.OutputName={'x','phi'};
syscl=connect(sysd,sysc,{'r'},{'x','phi','F'});
[Phic1,Gamma1,C1,D1,Ts]=ssdata(syscl)

```

Az eredő $\Sigma_{cl} = (\Phi_{cl}, \Gamma_{cl}, C_{cl}, D_{cl})$ zárt szabályozási kör számára a következő eredményeket kaptuk:

Phic1 =

Columns 1 through 7

```

1.2403    0.0100    0.1288   -0.0000   -0.2222    0.0086   -0.1081
48.0463    0.9982   25.7516   -0.0001  -44.4275    1.7269  -21.6239
-0.6009    0.0000    0.6792    0.0100    0.5556   -0.0216    0.2704
-120.1649    0.0045  -64.1601    1.0016   111.1142   -4.3191   54.0819
1.2577         0    0.1392         0   -0.2396    0.0186   -0.1185
73.8481         0   26.2716         0  -70.2293    2.7251  -22.1440
-0.5869         0    0.6978         0    0.5417   -0.0216    0.2518
-119.2815         0  -38.1576         0   110.2309   -4.3145   28.0795

```

Column 8

```

0.0042
0.8340
-0.0104
-2.0859
0.0042
0.8339
-0.0004
-1.0844

```

Gamma1 =

```

-0.0181
-3.6188
0.0453
9.0506
-0.0181
-3.6188

```

```

0.0453
9.0506

Ccl =
1.0e+003 *
Columns 1 through 7
    0.0010         0         0         0         0         0
         0         0    0.0010         0         0         0
    2.6447         0    1.4189         0    -2.4455    0.0951    -1.1903

Column 8
         0
         0
    0.0459

Dcl =
         0
         0
   -199.1915

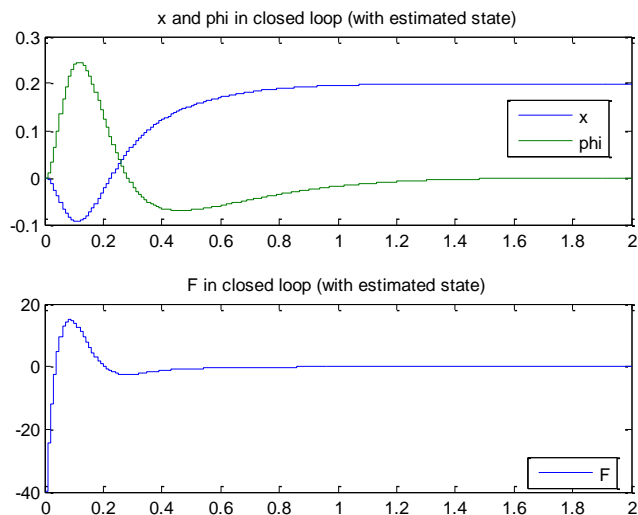
Ts =
    0.0100

```

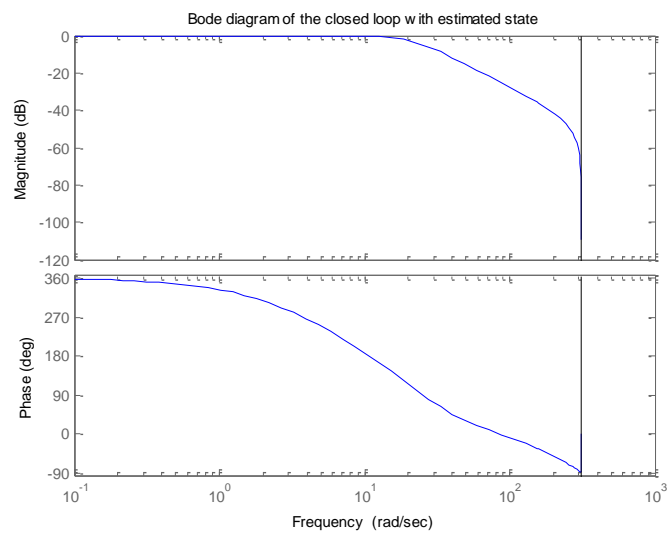
A Σ_{cl} rendszer kimenőjelei x , φ , F (az utóbbi a szabályozó kimenőjele). A Zárt rendszer tranzienseit és a zárt rendszer Bode-diagramját a 6.69.-6.70. ábrákon mutatjuk be.

A Bode-diagram az $r \rightarrow x$ alrendszerhez tartozik. A zárt rendszer Bode-diagramjából látható, hogy valóban sikerült a specifikáció megtervezésekor megcélzott Bode-diagramot megvalósítani. Vegyük észre, hogy a fázisszög 360 fok értékről indul, az induló érték levonása után a várt negatív fázisszög értékeket kapjuk (például $360^\circ \rightarrow 0^\circ$, $270^\circ \rightarrow -90^\circ$, stb.).

Nem vizsgáltuk a diszkrétidejű lineáris szabályozó viselkedését a nemlineáris szakasszal összekapcsolva, amely a Simulink kínálta lehetőségekkel viszonylag könnyen elvégezhető.



6.69. ábra. A zárt rendszer kimenőjelei és a szabályozó kimenő jele becstült állapotok esetén



6.70. ábra. A zárt rendszer Bode-diagramja becstült állapotok esetén

Az alábbiakban megadjuk a teljes `InvertedPendulum.m` script fájlt, melynek részleteit korábban már használtuk.

```

%InvertedPendelum.m
%Discrete time state space control

clear all
close all
clc

%Parameters
M=0.2; %mass of pendelum
m=0.5; %mass of cart
L=0.3; %length to pendelum COG
Theta=M*L^2/3; %0.006; %inertia belonging to COG
b=0.1; %viscous friction
g=9.8; %gravity acceleration

%Continuous time LTI system
det=Theta*(m+M)+m*M*L^2
A = [0 1 0 0; ...
      0 -(Theta+M*L^2)*b/det -M^2*L^2*g/det 0; ...
      0 0 0 1; ...
      0 M*L*b/det (m+M)*M*g*L/det 0]
B = [0; (Theta+M*L^2)/det; 0; -M*L/det]
C = [1 0 0 0; 0 0 1 0]
D=zeros(2,1)
sys=ss(A,B,C,D);
W=tf(sys)
[z,p,k]=zpkdata(sys);
z1=z{1};
p1=p{1};
k1=k(1);
z2=z{2};
p2=p{2};
k2=k(2);
sys1=zpk(z1,p1,k1)
sys2=zpk(z2,p2,k2)
figure(1)
pzmap(sys1)
title('Pole-Zero Map (F\rightarrow x)')
figure(2)
pzmap(sys2)
title('Pole-Zero Map (F\rightarrow phi)')

%Specification
s1=-4.95
s2=s1
xi=1/sqrt(2), w0=3*abs(s1)/xi
s3=-xi*w0+j*w0*sqrt(1-xi^2)
s4=conj(s3)
soinf=-5*max(abs(real([s1 s2 s3 s4])))
Ts=0.01

```

```

fprintf('Comparison of eigenvalues to Nyquist-
frequency:\n');
[abs([s1 s2 s3 s4 soinf]) pi/Ts]
%
z1=exp(s1*Ts)
z2=exp(s2*Ts)
z3=exp(s3*Ts)
z4=exp(s4*Ts)
zoinf=exp(soinf*Ts)

%discrete time LTI system
sysd=c2d(sys,Ts,'zoh');
[Phi, Gamma, C, D, Ts]=ssdata(sysd)

%State feedback design
zc=[z1 z2 z3 z4]'
K=acker(Phi, Gamma, zc)

%Nx, Nu design
Cx=C(1,:); Dx=D(1,:);
Nxu=inv([Phi-eye(4) Gamma; Cx Dx])*[zeros(4,1); 1];
Nx=Nxu(1:4)
Nu=Nxu(5)
Gff=K*Nx+Nu

%Closed loop simulation with measured state
tt=[0:0.01:2]';
FF=0.2*ones(size(tt,1),1);
[yy,xx]=dlsim(Phi-Gamma*K, Gamma*(K*Nx+Nu), C, D, FF);
figure(3)
stairs(tt,yy)
title('x and phi in cloed loop (with measured state)')
legend('x','phi','Location','Best')

%Observer design
zo=[zoinf zoinf+0.01 zoinf+0.02 zoinf+0.03]'
%repeated eigenvalues are forbidden for place
%MIMO -> acker cannot be used
KII=place(Phi', Phi'*C', zo)
G=KII'
F=Phi-G*C*Phi
H=Gamma-G*C*Gamma

%Don't use the following function
%[Ac, Bc, Cc, Dc]=Controller_M2('normal', Ts, Phi, Gamma, C, G, K, Nx
, Nu)

%Resulting controller for dimy~=dimu
nbar=size(Phi,1);
dimu=size(Gamma,2);

```

```

Ac=(Phi-Gamma*K)*(eye(nbar)-G*C);
Bc1=Gamma*(K*Nx+Nu);
Bc2=(Phi-Gamma*K)*G;
Cc1=-K*(eye(nbar)-G*C);
Dc11=K*Nx+Nu;
Dc12=-K*G;
Cc2=eye(nbar)-G*C;
Dc21=zeros(nbar,dimu);
Dc22=G;
Bc=[Bc1 Bc2];
Cc=[Cc1 Cc2];
Dc=[Dc11 Dc12; Dc21 Dc22];
Ac, Bc, Cc, Dc
sysc=ss(Ac,Bc,Cc,Dc,Ts);

%Connection by name
sysc.InputName={'r','x','phi'};
sysc.OutputName={'F','xhat1','xhat2','xhat3','xhat4'};
sysd.InputName={'F'};
sysd.OutputName={'x','phi'};
syscl=connect(sysd,sysc,{'r'},{'x','phi','F'});
[Phicl,Gammacl,Ccl,Dcl,Ts]=ssdata(syscl)

%Closed loop simulation with estimated state
%tt=[0:0.01:2]';
%
FF=0.2*ones(size(tt,1),1); %Step=0.2m
[yyy,xxx]=dlsim(Phicl,Gammacl,Ccl,Dcl,FF);
figure(4)
subplot(211)
stairs(tt,yyy(:,1:2))
title('x and phi in closed loop (with estimated state)')
legend('x','phi','Location','Best')
subplot(212)
stairs(tt,yyy(:,3))
title('F in closed loop (with estimated state)')
legend('F','Location','Best')
%
figure(5)
dbode(Phicl,Gammacl,Ccl(1,:),Dcl(1,:),Ts)
title('Bode diagram of the closed loop with estimated
state')

```

6. Ellenőrző kérdések a gyakorlathoz

1. Adja meg az M_c irányíthatósági mátrixot és a teljes elérhetőség/irányíthatóság feltételét a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén. Mit értünk reverzibilis rendszer alatt és az hogyan függ össze a teljes irányíthatósággal?
2. Fogalmazza meg a pólusáthelyezési feladatot állapot-visszacsatolás esetén, és a megoldás meghatározására szolgáló Ackermann-képletet $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű SISO rendszert feltételezve. Adja meg a zárt rendszer hatásvázlatát az állapot-visszacsatolás és mérhető állapot esetén.
3. Adja meg az alapjel miatti korrekcióhoz szükséges N_x, N_u mátrixok számítási szabályát a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén, méretüket speciálisan SISO rendszer esetén, és a zárt rendszer hatásvázlatát állapot-visszacsatolás és az alapjel miatti korrekció feltüntetésével.
4. Adja meg az M_o megfigyelhetőségi mátrixot és a teljes megfigyelhetőség/rekonstruálhatóság feltételét a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén. Mit értünk reverzibilis rendszer alatt és az hogyan függ össze a teljes rekonstruálhatósággal?
5. Adja meg a diszkrétidejű teljesrendű aktuális megfigyelő állapotegyenletét és a benne szereplő mátrixok megválasztását a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén. Adja meg a megfigyelő valósídejű szempontból kedvező realizálásának alakját.
6. Adja meg a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén a teljesrendű aktuális megfigyelőtervezési feladat megoldásának sémáját a dualitás elve és az Ackermann-képlet felhasználásával.
7. Adja meg a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén az állapot-visszacsatolás, alapjel miatti korrekció és aktuális állapotmegfigyelő együttes alkalmazása esetén a zárt rendszer hatásvázlatát.
8. Fogalmazza meg a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén az integrátort is tartalmazó állapot-visszacsatolási feladatot, adja meg a tervezés lépéseit és rajzolja fel alkalmazása esetén a zárt rendszer hatásvázlatát.
9. Adja meg a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén a terhelésbecslést (bemeneti zavarás kompenzálást) alkalmazó állapotmegfigyelő tervezési lépéseit, a benne szereplő mátrixok megválasztását és az Ackermann-képletre visszavezethető feladat alakját.

-
10. Adja meg a $\Sigma_d = (\Phi, \Gamma, C, D)$ diszkrétidejű rendszer esetén az állapot-visszacsatolást, alapjel miatti korrekciót és terhelésbecslőt alkalmazó szabályozó tervezési lépéseit, és a zárt rendszer hatásvázlatát együttes alkalmazásukkor.