

LINEÁRIS SZABÁLYOZÁSOK OPTIMÁLIS SZINTÉZISE DIGITÁLIS SZÁMÍTÓGÉPEN

I. RÉSZ

CSÁKI FRIGYES*

az MTA levelező tagja

KEVICZKY LÁSZLÓ** — KOVÁCS TIVADAR***

[Beérkezett: 1970. január 20-án]

Jelen cikkben egy olyan ALGOL programrendszer kerül ismertetésre, mellyel végrehajtható a lineáris szabályozási rendszerek optimális szintézise a négyzetes integrálkritérium alapján.

Bevezetés

Az állandó, koncentrált paraméterű valamint a holtidős tagokat tartalmazó egyváltozós lineáris szabályozási rendszerekben determinisztikus jelekre létrejött válaszfüggvények optimális lefolyását biztosító rendszerparaméterek meghatározására nagyon kézenfekvőnek látszik a korszerű digitális számológépek felhasználása.

A megoldáshoz szükséges numerikus módszerek — itt utalunk a lineáris egyenletrendszerek megoldására, határozott integrálok kiszámítására, többváltozós függvények lokális szélső értékének megkeresésére stb. — jól ismertnek tekinthetők. Így érdekes az a körülmény, hogy tudomásunk szerint e szakirodalomból hiányzik egy olyan egységes programcsalád széles körű ismertetése, mely a felvetett feladatnak megfelel.

Ezt pótlandó, jelen cikkünkben olyan ALGOL-programcsaládot kívánunk ismertetni, amelynek segítségével végrehajtható a lineáris szabályozási körök optimális szintézise valamely jel tranziens lefolyására jellemző integrálnak minimumra állítása alapján (integrálkritérium).

A tárgyalásunk során röviden kitérünk az elkészült programok elméleti alapjaira, felhasználási területükre és az alkalmazásuknál fellépő korlátozásokra.

1. ALGOL — eljárások a négyzetes integrálkritériumok számítására

Az állandó paraméterű, lineáris szabályozási körök legismertebb — ma már klasszikussá váló — tervezési módszerei általában a frekvencia és komplex operátoros tartományt használják fel. Ezek lassan eljutnak a lezárttság álla-

* Dr. Csáki Frigyes, Budapest, V., Váci u. 8.

** Dr. Keviczky László, Budapest, VII., Damjanich u. 25/a.

*** Kovács Tivadar, Budapest, XI., Stoczek u. 17/b.

potához, különösen abból a szempontból, hogy a számítógépes technika elterjedésével az említett tartományokra való áttérés (L-, Z- és F-transzformáció) valamint a visszatérés (az előbbieket inverze) viszonylag gyorsan végrehajtható. Ezen módszerek talán egyik legnagyobb problémája, hogy a lineáris szabályozásokra általánosan elfogadott optimalitási követelményt, azaz a minimális túllendülést és minimális szabályozási időt, igen nehéz az ω vagy s tartományban megfogalmazni. Így az egyes paraméterváltoztatások hatását is csak közvetve, az időfüggvény értékelése után kaphatjuk meg.

Célszerűnek látszik a tranzien্স szabályozási folyamatokra vonatkozó többé-kevésbé ellentmondó követelményeket közvetlenül az időtartománybeli viselkedés alapján optimálisan kiegyenlíteni. A kompromisszumot (többek között) az integrálkritériumok segítségével valósíthatjuk meg. [1, 2, 4]

Valamely integrálkritériumot kiértékelhetünk:

- a) a rendszer analóg vagy digitális szimulációjával valamint
- b) a frekvenciatartományra vonatkozó összefüggések segítségével.

A rendszernek analóg-számológépes szimulációjakor még tervezett kísérletek segítségével is igen nehézkes és hosszadalmas az optimális feltételek beállítása. Csak kevés változóra alkalmazható.

A digitális szimuláció megvalósítható egy eléggé általános modell programozásával vagy pedig a rendszer differenciálegyenletének lépésenkénti megoldásával. [8]

Mivel egy jól működő szimulációs programrendszer kidolgozása maga is komoly feladat, ezért az integrálkritériumok kiértékelésére a frekvencia tartományban illetőleg a komplex operátoros tartományban érvényes összefüggéseket választottunk.

Optimálisnak nevezünk egy dinamikus szabályozási folyamatot, ha egy általunk megválasztott integrál (funkcionál) minimumot ér el. Az integrálkritérium általános alakja:

$$I = \int_0^{\infty} F(x(t), t) dt = \min. \quad (1.1)$$

Itt F bizonyos kétváltozós függvényét jelenti a t időnek és egy alkalmasan megválasztott $x(t)$ jelnek. Az integrálandó függvényt úgy kell megválasztani, hogy egyrészt a tranzien্স folyamat minőségét megfelelően jellemezze, másrészt viszonylag egyszerű alakú legyen és a rendszer-paraméterekkel való kapcsolatát lehetőleg egyszerű formában fejezze ki. A felsoroltak többé-kevésbé ellentmondó igények. Így nem hat meglepetésként, ha a szakirodalomban a legkülönbözőbb alakú integrálkritériumokkal találkozunk [1—7].

Míg a lineáris kritériumok elsősorban aperiodikus folyamatok értékelésére használhatók, a kvadratis integrálkritériumok akár aperiodikus, akár lengő folyamatok vizsgálatához felhasználhatók [1, 4, 7]. A legkézenfekvőbb kritériumok az abszolút értékekre vonatkozó integrálkritériumok lennének,

amelyek mind az aperiodikus, mind a túllendülő folyamatok értékelésére használhatók. Mivel az abszolút értékek matematikai kezelése nehézkes, ezért inkább a négyzetes kritériumokat szokták a matematikai elemzésekben felhasználni. Az analóg számológépen azonban nem nehéz az abszolút érték kritériuma alapján vizsgálni a szabályozási rendszert.

Az optimalizálási probléma megoldásakor a szabályozási rendszer átmeneti függvényeiből, vagyis az időtartománybeli viselkedésből célszerű kiindulni. Fel kell azonban hívni a figyelmet az időtartomány és a frekvenciatartomány viszonylag szoros kapcsolatára. Lényeges a tervezés szempontjából, hogy milyen összefüggés mutatkozik egy bizonyos integrálkritérium és az átviteli függvény paraméterei (például együtthatói vagy időállandói, átviteli tényezője) között. Ennek a problémának az ad jelentősége, hogy rendszerint a frekvencia — (vagy komplex operátoros) tartományban vannak megadva a szabályozási rendszer összefüggései és paraméterei.

Optimalitási kritériumként mi a négyzetes integrálkritériumot (a továbbiakban ISE: Integral of Square of Error) választottuk. Kidolgoztunk olyan ALGOL eljárásokat, amelyek állandó, koncentrált paraméterű valamint holtidős tagokat tartalmazó egyváltozós lineáris szabályozási rendszerekben determinisztikus jelekre létrejött tranzienst időfüggvények négyzetes integrálkritériumának kiszámítására alkalmasak [2, 3].

A programok jól felhasználhatók szabályozási rendszerek tervezésében akár önállóan, akár valamely több változós szélső érték-kereső programba beépítve.

A négyzetes integrálkritérium választását indokolja, hogy ez a matematikai elemzésekben legáltalánosabban használt kritérium; kiértékelése a frekvenciatartományban is elvégezhető; igen jól gépesíthető; ez integrál minimuma „elég jó” időfüggvényt biztosít; szelektivitása nem rosszabb más idővel súlyozatlan kritériumoknál.

Ez utóbbival kapcsolatban megjegyezzük, hogy nem látjuk elég indokoltnak GRAHAM, D. és LATHROP, R. C. [6] azon megállapítását, hogy az ISE kritérium nem alkalmas szabályozási rendszerek optimalitásának eldöntésére. A [6]-ban végzett szelektivitási vizsgálatoknál ugyanis idővel súlyozott és súlyozatlan kritériumokat is vetettek össze egymással és ez nyilván az előbbieik javára döntött. A szelektivitási vizsgálatoknál nyilván csak az időfüggő súlyozó függvényt azonos formában tartalmazó integrálok összevetéséből vonhatunk le helyes következtetéseket, hiszen az idő valamely függvényével súlyozó integrálkritériumok későbbi időpontban fellépő ugyanakkora hibát nagyobb súllyal vesznek figyelembe.

Azt a tényt, hogy az ISE kritérium szelektivitása viszonylag kicsi (tehát a függvény a szélső értéke közelében elég lapos), az optimalizáláskor megfelelő (deriváltakat nem használó) szélső érték-kereső programmal ellensúlyozhatjuk.

Az ISE kritérium választása mellett szól az a tény is, hogy a [6]-ban optimálisnak mondott, az idővel súlyozott abszolút érték kritérium (a továbbiak-

ban ITAE) alapján kidolgozott standard formulák is vezethetnek helytelen eredményekre. (Erre a cikk második részében térünk vissza.)

Az $x(t)$ időfüggvény négyzetes integrálkritériumának definíciója:

$$\text{ISE} = \int_{-\infty}^{\infty} x^2(t) dt. \quad (1.2)$$

A Parseval-tétel [1, 2, 8] szerint az ISE integrálnak az időtartománybeli alakjával egyenértékű a frekvenciatartományban felírt következő kifejezés:

$$\text{ISE} = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} X(-s) X(s) ds = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} |X(s)|^2 ds. \quad (1.3)$$

ahol $s = j\omega$ és $X(s) = L\{x(t)\}$.

Ez utóbbi kifejezés még a következő alakban is írható :

$$\text{ISE} = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(j\omega)|^2 d\omega = \frac{1}{\pi} \int_0^{\infty} |X(j\omega)|^2 d\omega, \quad (1.4)$$

ahol kihasználtuk, hogy $|X(j\omega)|^2$ páros függvény.

Ha $X(s)$ racionális törtfüggvény, azaz a szabályozási kör nem tartalmaz holtidős tagot, tehát

$$X(s) = \frac{C(s)}{D(s)} = \frac{\sum_{k=0}^{n-1} c_k s^k}{\sum_{k=0}^n d_k s^k}, \quad (1.5)$$

akkor ISE kiszámítása egy lineáris egyenletrendszer megoldására vezethető vissza [1, 8, 9]. A számítás során feltételezzük, hogy a $D(s)$ valós együtthatójú polinomnak csak bal oldali zérushelyei vannak, valamint azt, hogy a $C(s)$ valós együtthatójú polinom legalább eggyel alacsonyabb fokú, mint $D(s)$. Az integrál kiszámításához megoldandó egyenletrendszer:

$$\mathbf{B} = \mathbf{D}\mathbf{A}. \quad (1.6)$$

Itt

$$\mathbf{B} = \begin{bmatrix} b_0 \\ b_2 \\ \vdots \\ b_{2n-2} \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} a_0 \\ -a_1 \\ \vdots \\ (-1)^{n-1} a_{n-1} \end{bmatrix}, \quad (1.7)$$

ahol

$$2b_m = \begin{cases} \sum_{k=0}^m (-1)^k C_k C_{m-k} & 0 \leq m \leq n-1 \\ \sum_{k=m-n+1}^{n-1} (-1)^k C_k C_{m-k} & n \leq m \leq 2n-2 \end{cases} \quad (1.8)$$

Ha n páratlan, akkor

$$D = \begin{bmatrix} d_0 & 0 & 0 & \dots & 0 \\ d_2 & d_1 & d_0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ d_{n-1} & d_{n-2} & d_{n-3} & \dots & d_0 \\ 0 & d_n & d_{n-1} & \dots & d_2 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & d_{n-1} \end{bmatrix}. \quad (1.9)$$

Ha n páros, akkor

$$D = \begin{bmatrix} d_0 & 0 & 0 & \dots & 0 \\ d_2 & d_1 & d_0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ d_n & d_{n-1} & d_{n-2} & \dots & d_1 \\ 0 & 0 & d_n & \dots & d_3 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & d_{n-1} \end{bmatrix}. \quad (1.10)$$

A fenti egyenletrendszert az a_{n-1} együtthatóra megoldva, a keresett integrál:

$$I = \frac{a_{n-1}}{d_n}. \quad (1.11)$$

Az „ISEL” elnevezésű ALGOL-eljárásunk a szabályozási kör paramétereiből a kívánt módon képezi a $C(s)$ és $D(s)$ polinomokat, a c_k és d_k együtthatókból előállítja az (1.6) egyenletrendszer együtthatóinak mátrixát, majd az egyenletrendszer megoldásával kiszámítja a négyzetes integrál értékét. A szabályozás struktúrájának változtatásakor a programban csak a c_k és d_k paramétereket előállító eljárást kell megváltoztatni. A legtöbb nagy teljesítményű számítógép által a $D(s)$ polinom fokszámára megadott korlátozás a gyakorlatban előforduló feladatokra ($n < 20$) nem jelent megkötést.

Röviden ismertetjük a programot.

procedure ISEL (n , y , parameter, transformer, generator, les)

value n ; *integer* n ; *real* y ; *array* parameter;

procedure transformer, generator, les;

transformer (Parameter, C , D);

generator (n , C , D , A);

les (N , A , X);

$y := (-1) \uparrow (n-1) \times X[n-1]/D[n]$

end ISE;

Kimeneti paraméter: y — az integrál értéke. A programban szereplő többi eljárás szerepe a következő:

1.

procedure transformer (parameter, C , D);

array parameter, C , D ;

.....

($C[1 : n-1]$ és $D[1 : n]$ tömbök előállítása a rendszer paramétereiből)

.....

end transformer;

A programnak tehát ez az eljárása az, amelyiket a szabályozási rendszer struktúrájának megfelelően kell mindig kialakítani.

Az eljárás kialakításához nyújtunk segítséget a függelékben közölt „MULT” és „ADD” elnevezésű eljárásokkal, melyek segítségével polinomok összeadása és összeszorozása végezhető el.

2.

Az (1.6) egyenletrendszer B és D mátrixát előállító „generator” elnevezésű eljárást a függelékben közöljük. Az eljárás kimeneti A tömbje tartalmazza az egyenletrendszer kibővített mátrixát.

3.

procedure les (n , A , X);

.....

(Az (1.6) egyenletrendszer megoldását adja az X vektorban.)

.....

end les;

Az egyenletrendszer megoldására mi a Gauss—Jordan-féle eliminációs módszert használtuk.

Az „ISEL” eljárást ALGOL-programmá kiegészítve, ismert eredményű számítésképeken próbáltuk ki. [9] és [10]-ben beszámolnak egy olyan ALGOL-eljárás elkészültéről, melynek segítségével kiszámítható ISE és ISTSE (lásd (1.15)-nél) értéke. Az ott ismertetett program felépítése igen bonyolult, a számítás is hosszadalmas, így nem felelt volna meg céljainknak, azaz az optimalizálásra történő felhasználásnak. Egyébként magát a programot nem közölték.

Ha a szabályozási rendszerben holtidős tag is található, akkor az integrálkritérium értékét nem tudjuk a lineáris egyenletrendszer megoldásához hasonló egyszerű úton kiszámítani, mivel $X(s)$ nem racionális törtefüggvény. Célszerűnek látszott ilyenkor ismét az (1.3) összefüggéshez fordulni, amelyet (1.4) alapján a frekvenciatartományban tudunk kiértékelni.

Az (1.4) improprius integrál kiértékelését a numerikus Laplace-transzformációban szokásos módszerekhez hasonlóan végeztük el. Eszerint a $[0; \infty]$ intervallumot az

$$\Omega = e^{-\omega} \quad (1.12)$$

transzformáció segítségével a $[0; 1]$ intervallumba vittük át.

Tehát

$$\text{ISE} = \frac{1}{\pi} \int_0^{\infty} |X(j\omega)|^2 d\omega = \frac{1}{\pi} \int_0^{\infty} G^2(\omega) d\omega = \frac{1}{\pi} \int_0^1 G^2\left(\frac{-\ln \Omega}{\Omega}\right) d\Omega. \quad (1.13)$$

alakú lesz kiértékelendő integrálunk. Itt $G(\omega) = |X(j\omega)|$.

Mivel az integrandusznak szakadása van az $\Omega = 0$ helyen, ezért az integrál alsó határának egy kis értékű a számot választunk ($a \sim 10^{-3} - 10^{-6}$), a számológép számtartománya és a kívánt pontosság függvényében. Így

$$\text{ISE} \cong \frac{1}{\pi} \int_a^1 G^2\left(\frac{-\ln \Omega}{\Omega}\right) d\Omega. \quad (1.14)$$

Ezen, most már határozott integrál kiértékelésére a Romberg—Stiefel—Bauer-féle numerikus módszert használtuk fel [11, 12]. A numerikus analízis általános megállapítása ugyanis, hogy a Newton—Cotes formulák közül egyszerűsége és gyors konvergenciája miatt a trapézformulán alapuló Romberg-féle módszer a legjobban alkalmazható [13].

Az (1.4) integrál kiszámításához nem feltétlenül szükséges az (1.12) átalakítás. Meg kell jegyeznünk, hogy azonos ω -nak megfelelő tartományra a kétféle módon számított integrálnál, azonos pontosság mellett az (1.14) ad kevesebb iterációt és így gyorsabb is.

Hasznossága miatt közöljük ezt az eljárást is:

```
procedure ISER (n, y, parameterek, romberg);
value n; integer n; real y; real procedure romberg; array parameterek;
romberg (a, b, f, eps, ord);
y := romberg/3.141592
end ISER;
```

A „romberg” eljárást a függelékben közöljük. Itt a *real procedure* f eljárás az, amely a rendszer paramétereiből az integrandusz függvényt előállítja, tehát megfelel az „ISEL” *procedure* transformer eljárásának.

Az eljárás felhasználásával készült programot egy és kéttárolós holtidős szakaszokat tartalmazó szabályozási körökben próbáltuk ki, sikerrel.

Az „ISEL” és „ISER” eljárások az

$$\text{ISTSE} = \int_{-\infty}^{\infty} t^2 x^2(t) dt. \quad (1.15)$$

idő négyzetével súlyozott négyzetes integrálkritérium (ISTSE: Integral of Squared Time multiplied Square of Error) is kiértékelhető [1, 8, 9], ugyanis

$$\text{ISTSE} = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \chi(-s) \chi(s) ds = \frac{1}{\pi} \int_0^{\infty} |\chi(j\omega)|^2 d\omega, \quad (1.16)$$

ahol

$$\chi(s) = -\frac{d}{ds} X(s). \quad (1.17)$$

Változás tehát csak a *procedure* transformer és *real procedure* f eljárásokban következik be.

Az elkészült eljárásokat igen hasznosan tudjuk alkalmazni szabályozás-technikai rendszerek tervezésében. Önmagukban is felhasználhatók, de a legjobb alkalmazási módot többváltozós szélső érték-kereső programba történő beépítés jelenti, így ugyanis az adott struktúrájú szabályozási rendszer optimális paramétereit is megkaphatjuk.

2. Az optimális paraméterek megkeresésére szolgáló algoritmusok

Optimálisnak nevezünk egy szabályozási rendszert, ha valamely megfelelően megválasztott átmeneti folyamatára az integrálkritérium kiszámított értéke minimumot ér el. A feladat tehát ilyen megfogalmazásban visszavezethető egy többváltozós függvény szélső-értékének (minimumának) megkeresésére.

Mivel az alkalmazott optimalitási kritérium (ISE) szelektivitása viszonylag kicsiny, azaz a függvény meglehetősen lapos az optimum környezetében, célszerűnek látszott olyan szélső érték-kereső eljárást választani, amely nem használja fel a függvény deriváltjait. Így a tiszta kereső módszernek tekinthető algoritmusok közül kettőre készítettünk ALGOL eljárást.

Az „OPTADJ-1” a minimum helyét **Rosenbrock** módszerével határozza meg [17]. Az eljárás nem használja fel a függvény deriváltjait. Az adott irány mentén a minimum meghatározása egy optimális elhatároló eljárással történik (**Fibonacci** módszer [16]).

Az „OPTADJ-2” a minimum helyét a szekvenciális szimplex módszerrel [18, 19] határozza meg. Ez az eljárás sem használja fel a függvény deriváltját.

Az elkészült programokat ismert szélső értékű n dimenziós függvényeken próbáltuk ki eredményesen.

A választott módszerek nem a leggyorsabb eljárások közül valók, viszont igen jól kézbentartathatók és a korlátozásokat könnyű figyelembe venni. Az eljárások az n változós függvény lokális minimumának helyét szolgáltatják egy adott korlátnál kisebb hibával. (Erre vonatkozó általános követelményt a deriváltakat használó módszereknél nehéz felállítani [14, 15].)

Mindkét eljárásnál a következő korlátozásokat lehet beiktatni a keresés közben:

- a) A paraméterek csak pozitív értékeket vehetnek fel;
- b) ISE értéke nem lehet negatív;
- c) ISE értéke egy adott korlátot nem haladhat meg.

A keresést az algoritmusok mindig egy előre megadott pontból kezdik. A szimplex módszernél a kezdeti szimplex élet is meg kell adni. A kezdeti szimplex elhelyezését úgy választottuk meg, hogy az a) feltételt minden új lépéshossznál biztosítsa.

A kiindulási pont és a kezdeti élhossz megállapításához, valamint annak eldöntésére, hogy a függvénynek létezik-e szélső értéke, és ha igen, akkor unimodális-e, minden esetben részletesen elemezni kell a szabályozási rendszer struktúráját. Az elkészült eljárások ugyanis csak a függvény lokális szélső értékét keresik meg. Az optimum abszolút voltát eldöntő más eljárások lényegesen lassabbak és bonyolultabbak a közölteknél.

Összefoglalás

Az elkészült ALGOL-programcsalád lehetővé teszi, hogy a négyzetes integrálkritérium alapján optimális lineáris szabályozási rendszereket tervezzünk.

Az eljárások egymáshoz jól illeszthetők és megbízható eredményeket szolgáltatnak.

Ha a szabályozási kör nem tartalmaz holtidős tagot, akkor a vizsgált időfüggvény négyzetes integrálját az „ISEL” programmal számíthatjuk ki.

Megoldottuk az ISE számítását holtidős tagokat tartalmazó szabályozási rendszereknél is, ezt a feladatot az „ISER” eljárásunk végzi el.

Az optimalizálási feladat megoldására felhasznált szélső érték-kereső eljárásaink (OPTADJ-1, OPTADJ-2) a többváltozós függvény (jelen esetben az ISE értékének) lokális minimumát határozzák meg adott pontossággal.

A létrehozott eljárások elve már régen ismeretes, munkánk fő eredménye, hogy azokra kipróbált algoritmusokat készítettünk és azokat a lineáris szabályozási rendszerek optimalizálási feladatának megfelelő egységbe foglaltuk.

Ezen ALGOL-programok lineáris szabályozási rendszerek optimális tervezésére történő felhasználását néhány példán és cikkünk második részében mutatjuk be.

IRODALOM

1. CSÁKI F.: Szabályozások dinamikája, Akadémiai Kiadó, Budapest 1966.
2. CSÁKI, F.: A Short Survey of Deterministic Optimisation Techniques Based on Integral Criteria, *Periodica Polytechnica* 12, 1.
3. KOVÁCS, T.—KEVICZKY, L.: ALGOL Procedures for the Evaluation of Quadratic Integral Criterious, *Periodica Polytechnica*, (In the press.).
4. FELDBAUM, A. A.: Integralnője kriterii kaceszta regulirovanija. *Avtomatika i telemenika* 9. (1948) 1.
5. NIMS, P. T.: Some Design Criteria for Automatic Controls, *Trans. AIEE*, 70, (1951), 606—611.
6. GRAHAM, D.—LATHROP, R. C.: The Synthesis of Optimum Transient Response. Criteria and Standard Forms. *Trans. AIEE*, 72, (1953), 273—288.
7. SCHULTZ, M. A.—RIDEOUT, V. C.: The Selection and Use of Servo Performance Criteria. *Trans. AIEE*, Appl. and Ind., January, (1958) 383—388.
8. GILBERT, E. G.: Dynamic-Error Analysis of Digital and Combined Analogue-Digital Computer Systems, *Simulation*, (1966), 241—257.
9. SCHNEIDER, F.: Geschlossene Formeln Berechnung der quadratischen und der zeitbeschwerten quadratischen Regelfläche für kontinuierliche und diskrete Systeme, *Regelungstechnik*, 14, (1966), 159—166.
10. SCHNEIDER, F.: Die Auswertung der Geschlossenen Formeln zur Berechnung der quadratischen und der zeitbeschwerten quadratischen Regelfläche für kontinuierliche und diskrete Systeme mit Hilfe des Digitalrechners, *Regelungstechnik*, 14, (1966), 557—562.
11. BAUER, F. L.: Romberg Integration, Algorithm 60 CACM, 4—61, 255.
12. FILIPPI: Das Verfahren von Romberg—Stiefel—Bauer . . *Mathematik, Techn. Wirtschaft*, 11. (1964) 2.3.
13. RALSTON, A.: A First Class in Numerical Analysis. Mc Graw-Hill Book Co., New York—London 1965.
14. WILDE, J. D.: Optimum Seeking Methods, Prentice-Hall, Inc. 1964.
15. BROOKS, S. H.: A Comparison of Maximum-Seeking Methods, *Operations Research*, 7, 430, 1959.
16. JOHNSON, S. M.: Optimal Search for a Maximum is Fibonacciian, Rand Corp. report, P-856, 1956.
17. ROSENBROCK, H. H.: An Automatic Method for Finding the Greatest or Least Value of a Function, *Computer Journal*, 3, 175.
18. SPEDLEY, W., HEXT, G. R., HIMSWORTH, F. R.: Sequential Application of Simplex Design in Optimization and Evolutionary Operation, *Technometrics*, 4, 441, 1962.
19. ERMURATSKIJ, P. V.: Szimpleksnőj metod optimizacii, Trudi MEI, Vőpuszk 67, Moszkva, 1966.
20. VÁMOS, T.—GYŐRKY, F.: Some Remarks on the Optimum Setting of Variable Parameter Process- Controllers, *Acta Techn. Hung.* 54, (1966), 439—452.

Főggelék

```

procedure generator (n, c, d, a);
value n; integer n; array c, d, a;
begin integer i, k, l, m, li, lk; real s;
array b[1 : n], dd[1 : n, 1 : n];
for m := 0, m + 2 while m ≤ n - 1 do begin s := 0;
for k := 0 step 1 until m do s := s + (-1) ↑ k × c[k + 1] × c[m - k + 1];
b[m/2 + 1] := s/2; l := m end;
for m := 1 ÷ 2, m + 2 while m ≤ 2 × n - 2 do begin s := 0;
for k := m - n + 1 step 1 until n - 1 do
s := s + (-1) ↑ k × c[k + 1] × c[m - k + 1];
b[m/2 + 1] := s/2 end;
if (n ÷ 2) × 2 = n then begin li := n/2 + 1; lk := n/2 - 1 end
else begin li := (n + 1)/2; lk := (n - 1)/2 end;
for k := 1 step 1 until n do for l := 1 step 1 until n do
if 2 × k - 1 ≥ 1 ∧ k ≤ li then dd[k, l] := d[2 × k - 1] else
dd[k, l] := 0;

```

```

for k := 0 step 1 until n - 1 do for l := 0 step 1 until n - 1 do
if  $2 \times k - 1 + 1 \geq 0 \wedge 2 \times k - 1 \leq n - 1 \wedge k \leq lk$  then dd[n - k, n - 1] :=
d[n - 2 \times k + 1];
for k := 1 step 1 until n do begin for l := 1 step 1 until n do
a[k, l] := dd[k, l]; a[k, n + 1] := b[b]
end generator;
real procedure romberg (a, b, f, eps, ord);
value a, b, eps, ord; integer ord; real a, b, eps;
real procedure f;
begin real q, s, f0, i1, i2, x, delta;
integer j, k, p; array t[1 : ord + 1];
s := b - a; x := a; i1 := f(x); x := b;
t[1] := (i1 + f(x)) \times s/2; i1 := 0
for k := 1 step 1 until ord do begin
i2 := 0; s := s/2; p := 2 \uparrow k
for j := p - 1 step -1 until 1 do begin
x := j/p; x := x \times a + (1 - x) \times b; f0 := f(x);
i2 := i2 + f0 end;
i2 := t[k + 1] - t[k]/2 + i2 \times s; q := 1
for j := k step -1 until 1 do begin q := 4 \times q;
i2 := t[j] := (i2 \times q - t[j])/(q - 1) end;
delta := abs((i2 - i1)/i2);
if delta \leq eps then goto fine; i1 := i2 end;
fine: romberg := i2
end romberg;
procedure mult (a, b, c, m, n, mn);
array a, b, c; value m, n; inter m, n, mn;
begin real s; integer i, k;
c[0] := a[0] \times b[0]; mn := m + n;
for k := 1 step 1 until m + n do begin s := 0;
for i := 0 step 1 until k do if i \leq m \wedge k - i \leq n
then s := s + a[i] \times b[k - i]; c[k] := s
end end mult;
procedure add(a, b, c, m, n, mn);
array a, b, c; value m, n; integer m, n, mn;
begin integer max, i;
if m \geq n then max := m else max := n; mn = max;
for i := 0 step 1 until max do begin if m < n then goto c1;
if n - i \geq 0 then c[m - i] := a[m - i] + b[n - i] else
c[m - i] := a[m - i] goto c2;
c1 : if m - i \geq 0 then c[n - i] := a[m - i] + b[n - i] else
c[n - i] := b[n - i]; c2 : end
end add;
procedure optadj2(n, l, eps, t, ISEL); value n, l, eps;
integer n; real l, eps; array t; procedure ISEL;
begin real p1, q1, p, q, yold, y, a1, s; integer i, j, k, ku,
jj, jo;
real array v[1 : n + 1], pq[1 : n + 1, 1 : n], sor[1 : n],
yk[1 : n + 1], u[1 : n + 1, 1 : n];
integer array rend[1 : n + 1];
p1 := 1/(n \times 1.4142) \times (n - 1 + sqrt(n + 1)) q1 := 1/(n \times 1.4142) \times (sqrt(n + 1) - 1);
for i := 1 step 1 until n + 1 do for j := 1 step 1 until n do;
begin if i = 1 then pq[i, j] := t[j] else if i - 1 = j then
pq[i, j] := p + t[j] else pq[i, j] := q + t[j] end;
for i := 1 step 1 until n + 1 do
begin for j := 1 step 1 until n do sor[j] := pq[i, j];
ISEL(n, y, sor, transformer, generator, les);
yk[i] := y end; k := 0; goto c10; c9: if 1 le eps then goto c16;
l := 0.5 \times l; for i := 1 step 1 until n do t[i] := pq[i, i];
goto c2; c10: for i := 1 step 1 until n + 1 do
begin rend[i] := i; v[i] := yk[i]; for i := 1 step 1 until n do
u[i, j] := pq[i, j] end; for i := 1 step 1 until n do

```

```

for j := i + 1 step 1 until n + 1 do begin if v[i] < v[j]
then begin al := v[i]; v[i] := v[j]; v[j] := al; jo := rend[i];
rend[i] := rend[j]; rend[j] := jo end end: ku := 1;
c13: if rend[ku] = k then goto c15; k := rend[ku]; yold :=
yk[k]; for j := 1 step 1 until n do begin s := 0;
for i := 1 step 1 until n + 1 do s := s + 2/n × pq[i, j];
pq[k, j] := s - (2 + n)/n × pq[k, j] end:
for j := 1 step 1 until n do begin if pq[k, j] < 0 then
goto c7; sor[j] := pq[k, j] end:
ISEL(n, y, sor, transformer, generator, les); yk[k] := y;
if yk[k] le yold then goto c10; c7: for j := 1 step 1
until n do pq[k, j] := u [k, j]; yk [k] := yold; if ku < n + 1
then goto c15;
jj := rend[n + 1]; goto c9; c15: ku := ku + 1; goto c13;
c16: text a fuegveeny minimumaanak becsleese;
output (yk[jj]); line: for i := 1 step 1 until n do
begin output(pq[jj, i]); line end end;

```

LINEÁRIS SZABÁLYOZÁSOK OPTIMÁLIS SZINTÉZISE DIGITÁLIS SZÁMÍTÓGÉPEN

II. RÉSZ

CSÁKI FRIGYES*—KOVÁCS TIVADAR**—KEVICZKY LÁSZLÓ***

[Beérkezett: 1970. jan. 20-án]

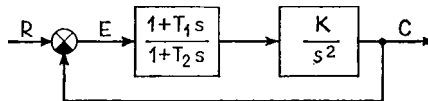
Az ismertett eljárások alkalmazása

Az [1]-ben ismertett ALGOL-eljárások olyan eszközöket adtak kezünkbe, amelyek segítségével minimális ráfordítással, minimális idő alatt végezhető el optimális tranziens tulajdonságú állandó paraméterű, lineáris szabályozási körök szintézise.

Jelen cikkünkben példákat közlünk az algoritmusok alkalmazására és beszámolunk az itt nyert tapasztalatokról, majd utalunk a továbbfejlesztés lehetőségeire.

Hogy a közölt példák szemléletesek legyenek, a bemutatott példákat a szabályozási körökben gyakran előforduló PI, PD, PID kompenzálás témaköréből vettük.

1. Példa.



1. ábra

Állítsuk be az 1. ábrán látható szabályozási körben a T_1 és T_2 időállandókat úgy, hogy a hibajel négyzetes integrálkritériuma (ISE) minimális legyen, ha az alapjel egységgrásfüggvény. A rendszer körerősítése legyen például $K = 777 \text{ l/sec}^2$. A hibajel L-transzformáltja ekkor:

$$E(s) = \frac{s + T_2 s^2}{777 + 777 T_1 s + s^2 + T_2 s^3}$$

* Dr. Csáki Frigyes, V., Váci u. 8.

** Kovács Tivadar, Bpest, XI., Stoczek u. 17/b

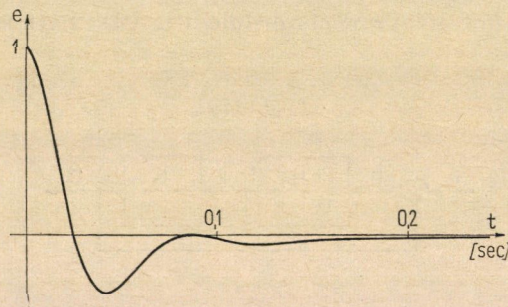
*** dr. Keviczky László, Budapest, VII. Damjanich u. 25/a

[1]-ben rávilágítottunk arra, hogy milyen fontos az optimalizálási feladatoknál a rendszer előzetes vizsgálata. Az ebben a példában alkalmazott PD kompenzáció $T_1 > T_2$ esetben strukturálisan stabilis rendszert hoz létre. Analitikusan is bebizonyítható, de a fizikai képből is látható, hogy a leggyorsabb átmeneti folyamat $T_1 \rightarrow \infty$ és $T_2 = 0$ értékekre következik be, tehát ISE-nek nincs minimuma a végesben. A hibajel L-transzformáltja ezen paraméter értékeknél

$$E(s) = \frac{1}{777 T_1} \frac{1}{\left(1 + \frac{1}{777 T_1} s\right)},$$

tehát egy zérushoz tartó időállandójú és átviteli tényezőjű egytárolós differenciáló tagnak megfelelő.

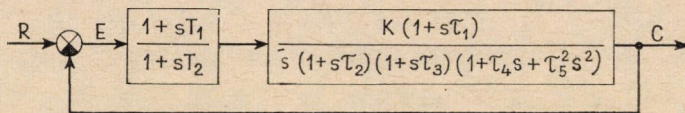
Érdekes megjegyezni azt, hogy ha T_1 és T_2 -t úgy választjuk meg, hogy biztosítsák a Graham—Lathrop-féle standard átviteli függvényeket [12], akkor $T_1 = 0,11$ sec és $T_2 = 0,012$ sec értékeket kapunk. Az ezekhez tartozó hibaátmeneti függvényt a 2. ábrán látjuk. Az átmeneti függvény alakja valóban „szépnek” mondható, viszont meglehetősen távol van, az előzőleg tárgyalt optimális tranzienstől.



2. ábra

Az elmondottakat támasztja alá az a tény is, hogy a programok is a T_1 és T_2 értékeire megadott korlátozási tartomány határáig keresték a szélsőértéket.

2. Példa.



3. ábra

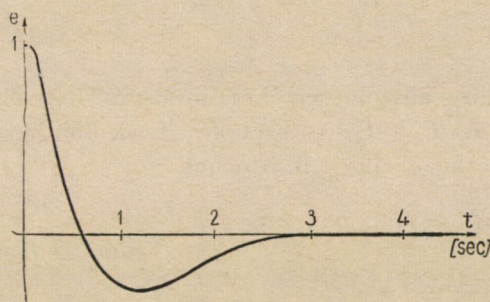
Válasszuk meg a 3. ábrán látható PD kompenzáció T_1 és T_2 paramétereit úgy, hogy ISE minimális legyen. ($R(s) = 1/s$) A szabályozott szakasz időállandói a következők legyenek:

$$K = 30 \text{ 1/sec}; \quad \tau_1 = 1,25 \text{ sec}; \quad \tau_2 = 0,667 \text{ sec};$$

$$\tau_3 = 2,5 \text{ sec}; \quad \tau_4 = 0,02 \text{ sec}; \quad \tau_5^2 = 0,0025 \text{ sec}^2.$$

Az optimalizálás eredményeként kapott hibaátmeneti függvényt a 4. ábrán láthatjuk. Az optimális paraméterértékek:

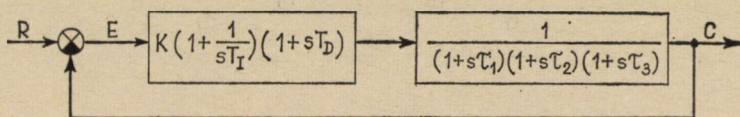
$$T_{1 \text{ opt}} = 1,06 \text{ sec}; \quad T_{2 \text{ opt}} = 0,08 \text{ sec}.$$



4. ábra

(Mivel a szabályozási körök felépítésének megváltozásakor a *procedure* transzformer eljárást kell megváltoztatni, így a 2. példára készült eljárás eléggé általános, magában foglalja az 1. példát is.)

3. Példa.



5. ábra

Egy háromtárolós szabályozott szakasz optimális kompenzálására alkalmazzuk az 5. ábrán látható felépítésű szabályozót. Paramétereit úgy kell megválasztani, hogy biztosítsák a négyzetes integrál minimumát a hibaátmeneti függvényre. A szabályozott szakasz időállandói legyenek a feladatban

$$\tau_1 = 1 \text{ sec}; \quad \tau_2 = 2 \text{ sec}; \quad \tau_3 = 3 \text{ sec}$$

értékűek.

A számítógépes optimalizálás igen érdekes eredményeket hozott ki optimális eredményként:

$$T_{I \text{ opt}} = 1,64 \text{ sec}; \quad K_{\text{opt}} = 5,22; \quad T_{D \text{ opt}} = 6,52 \text{ sec}.$$

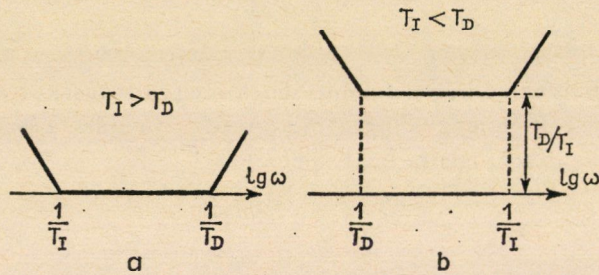
Az optimalizálás során tehát $T_I < T_D$ állapot jött létre, ami általában nem kívánatos, mivel T_I -vel a kisfrekvenciás, T_D -vel pedig a nagyfrekvenciás viselkedést szokták befolyásolni.

Vizsgáljuk meg a kérdés mögött húzódó strukturális problémákat. Legyen a szabályozó átviteli függvénye a következő:

$$G_{\text{PID}}^{(1)} = K \left(1 + \frac{1}{s T_I} \right) (1 + s T_D) = K \frac{1 + s(T_I + T_D) + s^2 T_I T_D}{s T_I}.$$

A frekvenciafüggvény abszolút értékét ábrázoló Bode-diagram látható a 6. ábrán, $T_I > T_D$ (a) és $T_I < T_D$ (b) esetekre. Ezek alapján az $G_{\text{PID}}^{(1)}$ frekvenciafüggő részét a következőképpen is írhatjuk:

$$\frac{T_D}{T_I} \frac{1 + s(T_I + T_D) + s^2 T_I T_D}{s T_D}, \quad \text{ahol } T_D > T_I.$$



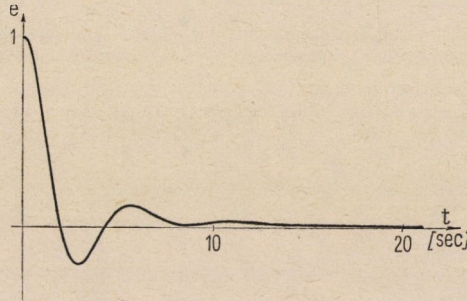
6. ábra

Ez pedig ugyanolyan átviteli függvényű tagot szolgáltat, amilyen az eredeti. Ebből következik, hogy az $G_{\text{PID}}^{(1)}$ típusú szabályozóknál két, egymással teljesen ekvivalens kompenzáció lehetséges a T_I és T_D egymáshoz képesti nagyságrendi viszonyának függvényében. Mivel a számológépi programjaink lokális minimumot keresnek, így előfordulhat, hogy a $T_I < T_D$ tartományban levő optimumot kapjuk eredményként. Az előbb elmondottak szerint ilyenkor tehát nincs más teendőnk, mint T_I és T_D értékének felcserélésével elvégeznünk a megfelelő korrekciót. Ennek megfelelően az optimális kompenzáció paramétereit:

$$T_{I \text{ opt}} = 6,52 \text{ sec}; \quad K = 5,22; \quad T_{D \text{ opt}} = 1,64 \text{ sec}.$$

Az optimális hibaátmeneti függvény a 7. ábrán látható. Ha szükségünk van az $G_{PID}^{(1)}$ -nek megfelelő

$$G_{PID}^{(2)} = K^* \left(1 + \frac{1}{sT_I^*} + sT_D^* \right) = K^* \frac{1 + sT_I^* + s^2 T_I^* T_D^*}{sT_I^*}$$



7. ábra

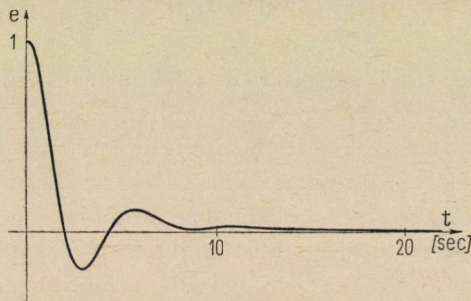
ideális szabályozó paramétereire, akkor azokat a következő képletekkel számíthatjuk:

$$T_I^* = T_I + T_D; \quad K^* = K \frac{T_I + T_D}{T_I}; \quad T_D^* = \frac{T_I T_D}{T_I + T_D}.$$

Az előzőleg kapott optimális paraméterek alapján:

$$T_{I\text{opt}}^* = 8,16 \text{ sec}; \quad K_{\text{opt}}^* = 6,52; \quad T_{D\text{opt}}^* = 1,294 \text{ sec}.$$

Az ideális PID kompenzációhoz tartozó optimális hibaátmeneti függvény a 8. ábrán található, megegyezik az előző optimális időfüggvénnyel.



8. ábra

4. Példa.

Az 5. ábrán látható szabályozási körben alkalmazzuk most a következő átviteli függvényű szabályozót:

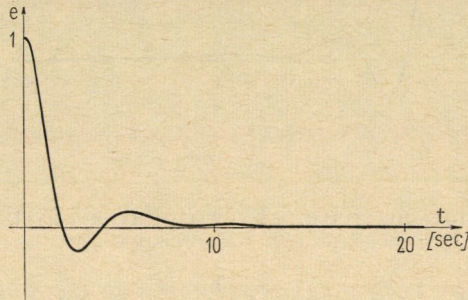
$$G_{PID}^{(3)} = K \left(1 + \frac{1}{sT_I} + \frac{sT_D}{1 + s\tau} \right) = K \frac{1 + s(T_I + \tau) + s^2 T_I(T_D + \tau)}{sT_I(1 + s\tau)},$$

ahol $\tau = 0,1$ sec. A szabályozott szakasz időállandói legyenek az előző példa szerintiek.

Az optimalizálás eredményeként most is $T_I < T_D$ adódott. Ezen paraméterek értékét megcserélve, a következő beállítási feltételeket kapjuk:

$$T_I = 6,42 \text{ sec}; \quad K = 5,31; \quad T_D = 1,69 \text{ sec.}$$

A megfelelő hibaátmeneti függvény képe a 9. ábrán látható.



9. ábra

Megjegyezzük, hogy ha az $G_{PID}^{(2)}$ vagy $G_{PID}^{(3)}$ szabályozót alkalmazzuk, akkor analitikusan nem lehet megindokolni az előzőekben felhasznált korrekciót, mivel a számláló csak $T_I > 4 T_D$ esetben alakítható szorzattá és akkor érvényesek az $G_{PID}^{(1)}$ -nél elmondottak. A 4. példából látható, hogy T_I és T_D felcserélése még ideális PID szabályozónál is megengedhető az optimális értékeknél.

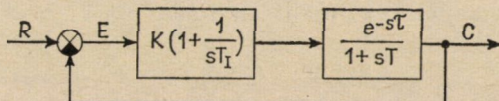
Az eddig megoldott feladatokban a szabályozási körök nem tartalmaztak holtidős tagot, tehát a hibajel L-transzformáltját mindig kifejezhettük racionális törtfüggvénnyel (bár ezeket itt nem részleteztük). Így a négyzetes integrálkritérium értékének kiszámítására az „ISEL” eljárásunkat használhattuk fel. A szélső érték megkeresésére a simplex módszert alkalmaztuk, mivel az gyorsabbnak bizonyult a másik módszerünkénél. A számítás ideje egy és két perc között változott (RAZDAN-3 digitális számítógépen) a kezdeti pont megválasztásától függően. Általában mondhatjuk, hogy jó kezdeti pontnak bizonyulnak a szabályozókra vonatkozó más, nem optimális, beállítási módok alapján választott értékek.

Az érdekesség kedvéért néhány példánál az optimális paramétereket az idő négyzetével súlyozott négyzetes integrálkritérium alapján is meghatároztuk. Ez nem eredményezett lényeges változásokat az optimális paraméterek értékében, viszont sokkal bonyolultabb L-transzformáltakat kellett beprogramozni.

5. Példa.

Tekintsük a 10. ábrán látható PI szabályozóval kompenzált, egytárolós holtidős szabályozott szakaszt tartalmazó rendszert. Válasszuk meg a szabályozó K és T_I paramétereit úgy, hogy a hibaátmeneti függvény ISE szerint optimális legyen. A szabályozott szakasz időállandói a következők:

$$\tau = 2 \text{ sec}; \quad T = 10 \text{ sec.}$$

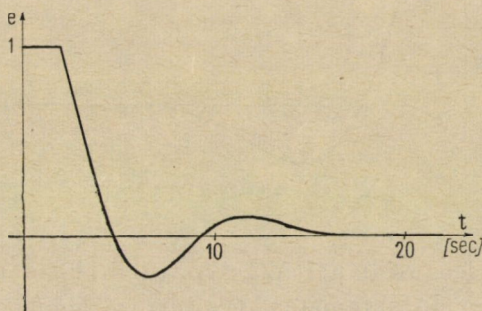


10. ábra

Az optimalizálást elvégezve

$$K_{\text{opt}} = 3,92 \text{ és } T_{I \text{ opt}} = 13,77 \text{ sec}$$

optimális értékeket kapunk. A hibaátmeneti függvényt a 11. ábra mutatja.



11. ábra

Ez a számítás több mint tíz percig tartott, tehát általában egy nagyságrenddel több, mint a holtidő nélküli rendszerekre. Ebből azt a következtést vonhatjuk le, hogy ez a számítási mód a jelenleg rendelkezésünkre álló számológépek gyorsasága mellett nem alkalmas beállítási célokra szolgáló diagramok kidolgozására, viszont egyedi tervezési feladatok megoldására nagyon jól alkalmas.

Mivel távolabbi céljaink között szerepel beállítási diagramok kidolgozása is, megpróbáltuk a számítási időt lecsökkenteni. Ha a holtidős tag helyén annak valamely PADE sorfejtését [6] alkalmazzuk, akkor az $E(s)$ L-transzformáltra

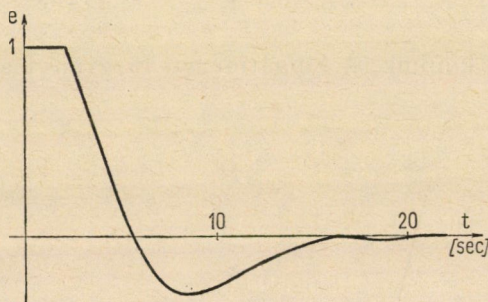
ismét racionális törtfüggvényt kapunk, tehát felhasználhatjuk a gyorsabb „ISEL” eljárást. Legyen a közelítés a következő:

$$e^{-s\tau} = \frac{1 - \frac{1}{2} s\tau + \frac{1}{10} s^2 \tau^2 - \frac{1}{120} s^3 \tau^3}{1 + \frac{1}{2} s\tau + \frac{1}{10} s^2 \tau^2 + \frac{1}{120} s^3 \tau^3}$$

Ezt felhasználva, a kapott optimális eredmények:

$$K_{\text{opt}} = 2,8; \quad T_{l \text{ opt}} = 5,5 \text{ sec.}$$

A hibaátmeneti függvény a 12. ábrán található. Az időfüggvényről látható, hogy még távol van a 11. ábrán bemutatott ideális optimális függvénytől, de beállítási célokra teljesen megfelelő. A pontosság természetesen az $e^{-s\tau}$ -t közelítő tag rendszámának növelésével fokozható.



12. ábra

A tervezési feladatokat a RAZDAN-3 digitális számítógépen oldottuk meg. A kapott optimális paraméterekhez tartozó időfüggvények meghatározásához a holtidő nélküli rendszereket a MEDA-40 TA analóg számológépen, a holtidős rendszereket pedig a MINSZK-22-es digitális számítógépen a BOCS digitális szimulációs program segítségével modelleztük. [13]

Következtetések

A szemléltetésre került példák jól illusztrálják azt, hogy az elkészült ALGOL programok igen hasznos segítőtársnak bizonyulnak a lineáris szabályozási rendszerek optimális tervezésekor. Segítségükkel a közölt, viszonylag egyszerűbb feladatok mellett bonyolultabb, speciális esetet is magába foglaló problémák is megoldhatók.

A módszerek amellet, hogy igen gyorsan szolgáltatják a kívánt optimális eredményt, értékes felvilágosításokat nyújthatnak a rendszer strukturális kérdéseiről, mint azt az előzőekben is láttuk.

Megmutattuk, hogy a feladatok megoldásakor milyen nagy szerepe van a rendszer előzetes strukturális elemzésének. Ekkor kell valamely hagyományos tervezési mód (pl. a Bode-diagramok) segítségével az optimalizálás kezdőpontját valamint egyéb segédmenyiségeket (hiba, lépéshossz stb.) megválasztani. Ha lehetséges, előre meg kell vizsgálnunk, hogy a szélső érték létezik-e (lásd 1. példa), és ha igen, akkor a függvénynek csak egy szélső értéke van-e vagy pedig több (lásd 3. példa).

A PID kompenzáció vizsgálata során megállapítottuk, hogy a négyzetes integrálnak, mint többváltozós függvénynek két egymással egyenértékű minimuma van. Ha a kereső algoritmus a nem kívánt ($T_i < T_D$) minimumot határozza meg, akkor a korrekció csupán T_i és T_D értékének megcserélésében áll.

Legértékesebb eredményünk, hogy megoldottuk a holtidős szabályozási körök optimalizálásának problémáját az „ISER” eljárással (Lásd 5. példa).

Legközelebbi feladatunknak tekintjük, hogy az elkészült programok segítségével beállítási diagramokat dolgozzunk ki egyszerű szabályozott szakaszok optimális PI és PID kompenzálásához.

Mivel a programok felhasználási területe rendkívül széles körű lehet, távolabbi céljaink között szerepel azok kiterjesztése digitális, sztochasztikus és nemlineáris rendszerekre, majd ezek többhurkú változatára is.

IRODALOM

1. CSÁKI, F.—KEVICZKY, L.—KOVÁCS, T.: Optimal Synthesis of Linear Control Systems by Digital Computer, Part. I, *Acta Techn. Hung.* 1970 (In the press).
2. CSÁKI, F.: Control Dynamics, Akadémia Kiadó, Budapest 1966
3. KOVÁCS, T.—KEVICZKY, L.: ALGOL Procedures for the Evaluation of Quadratic Integral Criteria, *Periodica Polytechnica*, (In the press).
4. GIBSON, J. E.: Szisztemü szamooptimizacii ili szamonasztravajucsieszja szisztemü avtomaticheskogo regulirobanyija, IFAC, 1966 Moszkva.
5. FELDBAUM, A. A.: Osznovü teorii optimalnüh avtomaticheskikh szisztem, Moszkva 1963.
6. TRUXAL, J. G.: Control System Synthesis, Mc Graw-Hill Book C. Inc., New York 1955.
7. OLDENBOURG, R.—SARTORIUS, H.: The Dynamics of Automatic Controls, *ASME*, New York 1948.
8. HERRIOT, P.: Process Control, Mc Graw-Hill Book C. Inc., New York—San Francisco 1964.
9. TUSTIN, A.: Method of Analysing the Behaviour of Linear Systems in Terms of Time Series, *Proc. IEE*, **94**, (1947), 130—142.
10. OLDENBOURG, G.—SARTORIUS, H.: A Uniform Approach to the Optimum Adjustment of Control Loops, *AMSE-Transaction*, **76**, (1954), 1265—1279.
11. NEWTON, G.—GOULD, L.—KAISER, J.: Analytical Design of Linear Feedback Controls, John Wiley Et Sons, New York 1957.
12. GRAHAM, D.—LATHROP, R. C.: The Synthesis of Optimum Transient Response. Criteria and Standard Forms, *Trans. AIEE*, **72**, (1953), 278—288.
13. KÁLMÁN, R.—BAKONYI, P.: BOCS blokk orientációs programrendszer a MINSZK-22 digitális számítógépre, *Mérés és Automatika*, **15**, (1967) 3.

Optimal Synthesis of Linear Control Systems by a Digital Computer. The present paper describes an ALGOL program system whereby the optimal synthesis of linear control systems can be performed on the basis of the quadratic integral criterion.

Optimaler Entwurf von linearen Regelungssystemen mit Hilfe von Digitalrechnern. Es wird eine Programmfamilie in algorithmischer Sprache ALGOL für den optimalen Entwurf der linearen Regelsysteme auf Grund der quadratischen Regelfläche beschrieben.