یک الگوریتم بازگشتی برای افزایش سرعت در محاسبه تبدیل سیستم‌های گسسته و پیوسته به یکدیگر

محمد باقر منهاج
استادیار
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

حامد شکوری گنجوی
دانشجوی دکترا
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

چکیده:

آنچه در این مقاله ارائه می‌شود یك الگوریتم سریع براساس، فرمول های بازگشتی است که می تواند تا حد مطلوبی به انجام تبدیل سیستم های گسسته و پیوسته به یکدیگر سرعت بخشد. در این مقاله نشان داده می شود که تعداد عملیات شناور در محاسبه تبدیل ضرایب به میزان سه برابر کاهش می یابد. نویسندگان مقاله معتقدند که این میزان کاهش سرعت برای زمانی است که ضرایب به طور مستقیم و یا با به توان رساندن و ضرب چند جمله ای به دست آیند و این روش در مقایسه با فرمول های محاسباتی دیگر [۱] بسیار کاراتر می باشد.

A Fast Recursive Algorithm for Continious-Discrete Model Conversion

M.B. Menhaj

Assist. Prof. Elect. Eng. Faculty
Amirkabir Univ. of Tech.

H. Shakouri. G.

Ph.D. Student, Elect. Eng. Faculty
Amirkabir Univ. of Tech.

Abstract

This paper presents a fast recursive algorithm for transforming Continious-Discrete linear systems to each other. The conversion formula consists of a map on the polynomials that construct the transfer functions. By the proposed formula the speed of conversion is highly improved (about 3 times faster). The efficiency of this method, compared with the direct calculations on polynomials, is verified by obtaining the floating point operations (FPO). The FPO for the recursive method is one third of direct calculations. The method is developed through two Lemmas and the FPO formulas are derived in an entirely general scheme.

۱- مقدمه

دیرزمانی است که تبدیل Bilinear منسوب به Tustin به صورت $s=\frac{2}{T}\frac{z-1}{z+1}$ از فضای صفحه s به صفحه z و یا تبدیل معکوس آن به نام تبدیل -w به صورت $z=\frac{1+wT/2}{1-wT/2}$ از فضای صفحه z به صفحه w به منظور تقریب گسسته نگهدارنده صفر برای سیستم های پیوسته و یا مطالعات پایداری در سیستم های گسسته مورد استفاده قرار می گیرد [۳-۱]. با در نظر گرفتن این حقیقت که این تبدیلات به طور مکرر در مطالعات شناسایی و کنترل سیستم های دیجیتال کاربرد می یابند [۵-۴] توجه به سرعت محاسبات در انجام آنها از اهمیتی درخور توجه برخوردار خواهند بود.

در این مقاله یک الگوریتم سریع براساس فرمول های بازگشتی ارائه می شود که می تواند تا حد مطلوبی به انجام این تبدیل سرعت بخشد. نکته حائز اهمیت دیگر در اینجا، استفاده از الگوریتم های بازگشتی است که در بیشتر علوم مخصوصاً تئوری تخمین ها نقش مهمی بازی می کنند [۷-۶]. این الگوریتم ها دارای دو ویژگی اصلی هستند: یکی سهولت محاسبات online و دیگری حجم پایین محاسبات می باشد. (لازم به ذکر است که تمامی متدهای کلاسیک این تبدیل را به صورت Batch یا غیربازگشتی انجام می دهند.) در این مقاله سعی شده است با ایجاد تغییر و تحولاتی در ماتریس تبدیل قادر باشیم ضرایب چند جمله ای درجه n+1ام را از روی ضرایب چند جمله ای تبدیل درجه nام به دست آوریم. این روش عیناً در الگوریتم های بازگشتی تئوری تخمین ها جهت تخمین پارامترها به کار می رود؛ به این ترتیب که از اطلاعات جدید برای تخمین بهتر پارامترهای تخمین زده شده استفاده می شود. ساختار مقاله از این قرار می باشد:

ابتدا مروری بر معرفی تبدیل مزبور و کاربردهای آن شده است. سپس روش محاسبه آن به طور معمول بیان شده و الگوریتمی برای آن طرح می گردد. در بخش بعد با به دست آوردن چند فرمول، مسأله به طور دیگری طرح و الگوریتم جدیدی براساس نمایش ماتریسی پی ریزی می شود. سرانجام به فرمول تعداد عملیات نقطه شناور (Floating point opperation) برای هرمرحله از محاسبات در دو روش معمولی و روش بازگشتی پیشنهاد شده پرداخته می شود. پس از حصول نتایج مبنی بر اینکه تعداد عملیات در هر دو روش با توان سوم درجه سیستم های گسسته (یا پیوسته) متناسب است، نشان داده می شود که روش بازگشتی تعداد عملیات برای محاسبه تبدیل را در حدود سه برابر کاهش می دهد.

۲- بازگویی مسأله

مسأله مورد بررسی، الگوریتم محاسبه تبدیل Bilinear است. بنابراین ابتدا روش معمول در این محاسبه را از نظر می گذرانیم. تعریف مسأله با بیان ساده چنین است: چند جمله ای $C_n(P)$ به صورت زیر داده شده است:

$$C_n(p)=\sum_{i=0}^{n} c_i p^i \qquad (۱)$$

که می تواند یک چند جمله ای Monic باشد. می خواهیم با قرار دادن $p=\frac{q-1}{q+1}$ در این چند جمله ای کسر گویای $\frac{D_n(q)}{(q+1)^n}$ را محاسبه نماییم.

$$\frac{D_n(q)}{(q+1)^n}=\sum_{i=0}^{n} c_i\left(\frac{q-1}{q+1}\right)^i \qquad (۲)$$

رابطه (۲) را می توان طی عملیات ساده ای به شکل ماتریسی زیر نمایش داد:

$$D_n(q)=\sum_{i=0}^{n} c_i(q-1)^i(q+1)^{n-i}=\sum_{i=0}^{n} d_i q^i \qquad (۳)$$
$$=\left[(q+1)^n,(q+1)^{n-1}(q-1),\ \dots,(q+1)(q-1)^{n-1},(q-1)^n\right]\underline{c}(n)$$

که در آن بردار $\underline{c}$ بردار ضرایب چند جمله ای $C_n(P)$ است:

$$\underline{c}(n)=[c_0,\ c_1,\ \dots,\ c_n]^T \qquad (۳-۱)$$

بدیهی است در روش معمول محاسبهٔ $D_n(q)$ می بایست ابتدا بسط نیوتن دو جمله ای های $(q+1)^{n-i}$ و $(q-1)^i$ را برای کلیه مقادیر $i=2, \dots, n$ به دست آورده، سپس با ضریب c_i مناسب در یکدیگر ضرب و نهایتاً پس از یافتن توان های یکسان با هم جمع نماییم. قبل از آن که روند انجام این عملیات نسبتاً طولانی را به طور خلاصه مرور کنیم، لازم است توجه کنیم که در تبدیل Bilinear که عموماً برا ی محاسبه معادل گسسته سیستم های پیوسته یا بالعکس مورد استفاده واقع می گردد، چند جمله ای $C_n(p)$ تبدیل به کسر گویایی برحسب q شده است؛ امّا در حالی که صفرهای چند جمله ای $D_n(q)$ مورد علاقه می باشند (مثلاً چنانچه مسأله پایداری مورد

تحقیق باشد) نیازی به محاسبهٔ مخرج این کسر نداریم. به هر حال طبق رابطه (۳) قسمت اصلی این محاسبه را یافتن چند جمله ای های$\phi_{n,i}(q)$ با تعریف زیر تشکیل می دهد:

$$\phi_{n,i}(q) = (q-1)^i (q+1)^{n-i} \; ; \; i = 0, 1, \ldots, n \qquad (۴)$$

می دانیم:

$$(x+1)^m = \sum_{k=0}^{m} \frac{m!}{k!(m-k)!} x^{m-k} \qquad (۵)$$

پس محاسبه چند جمله ای $\phi_{n,i}(q)$ در حالت کلی از طریق به دست آوردن بسط فوق، دوبار برای m=i و m=n-i و ضرب آن دو در هم میسر است. یک مرور دوباره به رابطهٔ (۳) معلوم می دارد که در انجام این تبدیل، عملیات مشابهی در حال اتفاق است. این دقت ما را بر آن داشته است که به دنبال روش بازگشتی باشیم تا در انجام این عملیات تکراری در محاسبه چند جمله ای های $\phi_{n,i}(q)$ صرفه جویی شده و زمان محاسبات تا حد قابل توجهی کاهش یابد. رابطه (۴) را با اعمال فرمول بسط خیام در رابطه (۵) بازنویسی می کنیم:

$$\psi_{n,i}(q) = \left[\sum_{j=0}^{i} (-1)^j . f_{i,j} . q^{i-j}\right] . \left[\sum_{l=0}^{n-i} f_{n-i,l} . q^{n-i-l}\right] \qquad (۶)$$

که در آن برای اختصار تعریف شده است:

$$f_{i,j} = \frac{i!}{j!(i-j)!} \qquad (۷)$$

ملاحظه می شود که نه تنها محاسبه ضریب $f_{i,j}$ خود شامل عملیات تکراری است، بلکه در جملات مختلف حاصلضرب (۴) نیز تکرار می گردد. در بخش بعد با طرح قضیه ای همین نکته، اساس کاهش محاسبات قرار خواهد گرفت.

۳ـ مقدمات ریاضی در جستجوی الگوریتم بازگشتی

در این بخش سعی خواهیم کرد که با انجام یک سری عملیات ریاضی به نتایجی دست یابیم که بتوان محاسبات را خلاصه کرد.

ابتدا از محاسبهٔ چند جمله ای های $\phi_{n,i}(q)$ شروع می کنیم. با تعریف m = n-i و با پیگیری مراحل ضرب در چند جمله ای بسط یافته زیر:

$$(q-1)^i (q+1)^m = \left[q^i - iq^{i-1} + \ldots + (-1)^l \frac{i!}{l!\,(i-l)!} q^{i-l} + \ldots\right] . \left[q^m - mq^{m-1} + \ldots + \frac{m!}{k!\,(m-k)!} q^{m-k} + \ldots\right] \qquad (۸)$$

می توان ملاحظه کرد که:

$$\psi_{n,i}(q) = \sum_{j=0}^{n} h_{j,i}\, q^{n-j} \qquad (۹)$$

که در آن ضرایب $h_{j,i}$ به نحو زیر به دست می آیند. در جدول زیر این مراحل را بررسی می کنیم:

برای ضریب جملهٔ عمومی j‌ام که در آن $k = j-l$ تعریف شده است ملاحظه می شود که دو شرط زیر همواره ضروری اند:

(۱۰) با توجه به توان های غیر منفی برای q^{i-l} $\quad l \le i$

i	0	1	2	j ....
جمله j ام	q^n	q^{n-1}	q^{n-2}	q^{n-j}
حاصل ازضرب جملات	$q^m . q^i$	$q^i.q^{m-1}, q^{i-1}.q^m$	$q^i.q^{m-2}, q^{i-1}. q^{m-1}, q^{i-2}. q^m$	$q^i. q^{m-j}, \ldots, q^{i-l}. q^{m-k}, \ldots q^{i-j}. q^m$
ضریب جمله j ام	1.1	1.m + (-i).1	$1.\frac{m(m-1)}{2} + (-i).m + \frac{i(i-1)}{2}.1$	$\sum_{l=0}^{j} (-1)^l \frac{i!}{l!(i-l)!} . \frac{m!}{k!(m-k)!}$

(۱۱) با توجه به توان‌های غیر منفی برای q^{m-k} $k \le m$

بدیهی است شرط (۱۰) بایستی برای کلیهٔ مقادیر l برقرار باشد، بنابر این:

$j \le i$

همچنین از جایگزینی برای شرط (۱۱) به دست می‌آید:

$j-l \le n-i$

و از آنجا:

$l \ge i+j-n$

در ادامه از لم زیر برای پی‌ریزی الگوریتم خود بهره می‌گیریم:

لم ۱

چند جمله‌ای $\phi_{n,i}(q)$ با تعریف (۶) معادل با چند جمله‌ای (۸) می‌باشد، جائیکه ضرایب $h_{j,i}$ چنین به دست آمده باشند:

$$h_{j,i} = \sum_{l=l_{min}}^{l_{max}} (-1)^l \cdot f_{i,l} \cdot f_{m,k} \qquad (۱۲)$$

که در آن:

$$l_{min} = \max(0, i+j-n) \qquad (۱۳)$$

$$l_{max} = \min(i, j) \qquad (۱۴)$$
$$m = n-i, k = j-l$$

و ضرایب $f_{i,l}$ و $f_{m,k}$ نیز طبق (۷) تعریف شده‌اند. اثبات براساس آنچه گذشت و نیز از طریق استقراء ریاضی روشن است. حال پیش از استخراج رابطه بازگشتی، این قسمت را با تشکیل یک نمایش ماتریسی تکمیل می‌نماییم.

اوّل رابطه (۹) را می‌توان به صورت ماتریسی زیر درآورد:

$$\phi_{n,i}(q) = [q^n, q^{n-1}, \dots, q^0]\, \underline{h}_i \qquad (۱۵)$$

که در آن $\underline{h}_i$ بردار ضرایب $h_{j,i}$ عبارت است از:

$$\underline{h}_i = [h_{0,i}, h_{1,i}, \dots, h_{n,i}]$$

از طرفی با تعریف (۴)، رابطه (۳) تبدیل می شود به:

$$D_n(q) = [\phi_{n,0}(q), \phi_{n,1}(q), \dots, \phi_{n,n}(q)]\, \underline{c}(n) \qquad (۱۶)$$

با قرار دادن (۱۵) در (۱۶) خواهیم داشت:

$$D_n(q) = q^T(n)\, H(n)\, \underline{c}(n) \qquad (۱۷)$$

که در آن تعریف شده است:

$$q(n) = [q^n, \dots, q^0]^T \qquad (۱-۱۷)$$

$$H(n) = [\underline{h}_0, \underline{h}_1, \dots, \underline{h}_n] \qquad (۲-۱۷)$$

٤ـ استخراج رابطهٔ بازگشتی

در این قسمت، هدف آن است که با داشتن رابطه (۲) برای درجهٔ nام، این رابطه را از روش کوتاه‌تری برای درجهٔ n+1 به دست آوریم. رابطه مذکور از درجه n+1 چنین است:

$$\frac{D_{n+1}(q)}{(q+1)^{n+1}} = c_{n+1}\left(\frac{q-1}{q+1}\right)^{n+1} + \sum_{i=0}^{n} c_i \left(\frac{q-1}{q+1}\right)^i$$

به سادگی می‌توان نشان داد که سمت راست عبارت فوق برابر است با:

$$c_{n+1}\left(\frac{q-1}{q+1}\right)^{n+1} + \frac{1}{(q+1)^n} q^T(n)\, H(n)\, \underline{c}(n)$$
$$= \frac{1}{(q+1)^{n+1}} \left[c_{n+1}(q-1)^{n+1} + (q+1)\, q^T(n)\, H(n)\, \underline{c}(n)\right] \qquad (۱۸)$$

بر طبق نمایش (۱۷) با تبدیل n به n+1 عبارت داخل کروشه معادل است با:

$$q^T(n+1)\, H(n+1)\, \underline{c}(n+1) = c_{n+1}(q-1)^{n+1} + (q+1)\, q^T(n)\, H(n)\, \underline{c}(n) \qquad (۱۹)$$

با تعریف ماتریس های زیر:

$$S(n)=\begin{bmatrix} \underline{0}_{(n\times 1)} & H(n) \\ 0 & \underline{0}^T_{(1\times n)} \end{bmatrix}_{(n+2)\times(n+2)}$$

$$T(n)=\begin{bmatrix} H(n) & \underline{0}_{(n\times 1)} \\ \underline{0}^T_{(1\times n)} & 0 \end{bmatrix}_{(n+2)\times(n+2)}$$

ترم دوم از عبارت سمت راست رابطه (۱۹) به صورت ماتریس زیر نوشته می شود:

$$q\,.\,q^T(n)\,H(n)\,\underline{c}(n)+q^T(n)\,H(n)\,\underline{c}(n)$$

$$q^T(n+1)\,S(n)\,\underline{c}(n+1)+q^T(n+1)\,T(n)\,\underline{c}(n+1)$$

حال اگر بسط $(q+1)^{n+1}$ را به صورت ماتریسی درآوریم:

$$(q+1)^n=\sum_{i=0}^{n+1} f_{n+1,\,i}\,q^{n+1-i} = q^T(n+1)\,f(n+1) \qquad (۲۰)$$

که در آن:

$$f(n+1)=\left[f_{n+1,0}\,,f_{n+1,1},\ldots,f_{n+1,n+1}\right]^T \qquad (۱-۲۰)$$

آنگاه صورت رابطه (۱۸) عبارت خواهد بود از:

$$D_{n+1}(q)=c_{n+1}\,q^T(n+1)\,f(n+1)+q^T(n+1)\,.$$
$$S(n)\,\underline{c}(n+1)-q^T(n+1)\,T(n)\,\underline{c}(n+1)$$

اجازه دهید تعریف دیگری را بیافزاییم:

$$f_0(n+1)=\left[f(n+1),0_{(n+1)\times(n+1)}\right]$$

بنابر این:

$$D_{n+1}(q)=\underline{q}^T(n+1)\left[f_0(n+1)+S(n)-T(n)\right]\underline{c}(n+1) \qquad (۲۱)$$

از مقایسه (۲۱) و (۱۹) ملاحظه می شود که بایستی داشته باشیم:

$$H(n+1)=f_0(n+1)+S(n)-T(n)$$

بدین ترتیب با داشتن H (n) که در مرحلهٔ nام محاسبه شده است، H (n+1) به راحتی قابل تشکیل است:

$$H(n+1)=\begin{bmatrix} f_{n+1,0} & H(n) \\ f_{n+1,n+1} & \underline{0}^T_{1x(n+1)} \end{bmatrix}-\begin{bmatrix} 0 & \underline{0}^T_{1x(n+1)} \\ \underline{0}_{(n+1)x1} & H(n) \end{bmatrix} \qquad (۲۲)$$

رابطه (۲۲) همان رابطه بازگشتی است که به دنبال آن بودیم. با توجه به این که به جز از ستون اول ماتریس اول عملیات دیگری جز جمع دو ماتریس موجود نیست و به نظر می رسد تعداد عملیات محاسباتی کاملاً کاهش یافته باشند، در بخش بعد حجم عملیات مورد لزوم در این روش برای حصول $D_n(q)$ را با روش مستقیم تبدیل مقایسه خواهیم نمود.

لم ۲

تبدیل Bilinear از چند جمله ای $C_n(p)$ با قرار دادن $p=\frac{q-1}{q+1}$ عبارت است از کسر گویای $\frac{D_n(q)}{(q+1)_n}$ و چند جمله ای $D_n(q)$ برابر است با:

$$D_n(q)=\underline{q}^T(n)\,H(n)\,\underline{C}(n) \qquad (۲۳)$$

$$H(k)=\left[f(k)\ :\frac{H(k-1)}{\underline{0}^T}\right]-\left[\underline{0}\ :\frac{\underline{0}^T}{H(k-1)}\right]_{(k+1)(k+1)} ;k=1,\cdots,n \qquad (۱-۲۳)$$

جائیکه:

$$H(0)=1 \qquad (۲-۲۳)$$

و بردارهای $\underline{f}$، $\underline{q}$، $\underline{c}$ به ترتیب طبق روابط (۳-۱)، (۱۷-۱)، (۲۰-۱) تعریف شده اند.

مثال ۱: فرض کنیم بخواهیم تبدیل چند جمله ای $C(p)=p^2+2p+3$ را تحت نگاشت مختلط $p=\frac{q-1}{q+1}$ به دست آوریم. خواهیم داشت:

$$f(1)=\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad , \quad H(0)=1$$

از آنجا:

$$H(1)=\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}-\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad ; \quad f(2)=\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

و به همین ترتیب:

$$H(2)=\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix}$$

بنابر این:

$$D(q)=\begin{bmatrix} q^2 & q & 1 \end{bmatrix} H(2)\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}=6q^2+4q+2$$

یعنی:

$$C(P)\xrightarrow{p=\frac{q-1}{q+1}}\frac{D(q)}{(q+1)^2}$$

۵ـ مقایسه حجم عملیات محاسباتی

آنچه نتایج بخش پیش را بیشتر درخور توجه جلوه می‌دهد، تعداد عملیات محاسباتی در محاسبه این تبدیل است. در این بخش نشان خواهیم داد که این حجم به میزان تقریباً ۲ تا ۳ برابر در مقایسه با روش محاسبه مستقیم کاهش می یابد. ابتدا متذکر می شویم که آنچه در اینجا مورد نظر است، تعداد عملیات نقطهٔ شناور[1] شامل چهار عمل اصلی می باشد. به عنوان مثال برای ضرب دو ماتریس به ابعاد $m\times n$ و $n\times p$ تعداد (fpo) برابر است با $mnp\times(n-1)$. نتایج ریاضی این بخش به توسط نرم افزار MATLAB نیز مورد آزمایش و تصدیق قرار گرفته اند که در ادامه می‌آید.

۵-۱ـ fpo در محاسبه مستقیم تبدیل

منظور از محاسبه مستقیم انجام عملیات، به توان رسانی و ضرب چند جمله ای ها طبق رابطه (۳) می باشد، می توان نشان داد که برای ضرب دو چند جمله ای از درجه های n_1 و n_2 که منجر به چند جمله ای با درجه n_1+n_2 می شود، داریم:

$$fpo_{conv}(n_1, n_2)=2\min(n_1, n_2)(n_1+n_2-1) \qquad (۲۴)$$

از آنجا ضرب چند جمله ای‌ها در واقع حالتی از ضرب کانولوشن است. زیرنویس conv برای آن انتخاب شده است.

تعداد عملیات محاسبه توان nام هر چند جمله ای $q\pm1$ نیز متعاقباً و با اندکی دقت قابل حصول است. می توان نشان داد که این میزان برابر است با:

$$fpo_{pow}(n)=2n^2+6n-8\ ;\ (n\neq0) \qquad (۲۵)$$

در ادامه می توان fpo را برای انجام مجموعه عملیات لازم در محاسبه $(q+1)^i\ (q-1)^{n-i}$ به دست آورد. این عملیات شامل دو عملیات pow با درجات n-i و i و یک عملیات conv با همین درجات است. با اختصار در نوشتار خواهیم داشت:

$$\begin{aligned} fpo_{tot}(n,i) &= 2n^2+4i^2-2ni+8n+2i-14\ ;\ 0<i\le\frac{n}{2} \\ &= 2n^2+8n-6 \qquad ;\ i=0\ \text{یا}\ n=i \\ &= 4n^2+4i^2-6ni+11n-2i-14\ ;\ \frac{n}{2}\le i<n \end{aligned}$$

به دست آوردن $D_n(q)$ نیاز به انجام اعمال فوق برای $i=0, \dots, n$ برای تشکیل ماتریسی مشابه $H(n)$ دارد. در این صورت fpo را برای nهای زوج طبق:

$$fpo_{even}(n)=2fpo_{tot}(n,0)+fpo_{tot}(n,\frac{n}{2})+2\sum_{i=1}^{n/2-1} fpo_{tot}(n,i)$$

به دست می‌آوریم که ساده می شود به:

$$fpo_{even}(n)=\frac{11}{16}n^3+\frac{19}{2}n^2-\frac{16}{3}n+2 \qquad (۲۷)$$

و برای مقادیر فرد n نیز روشن است که باید حساب کنیم:

$$fpo_{odd}(n)=2fpo_{tot}(n,0)+2\sum_{i=1}^{(n-1)/2} fpo_{tot}(n,i)$$

و این نهایتاً منجر می شود به:

$$fpo_{odd}(n)=\frac{11}{6}n^3+\frac{21}{2}n^2-\frac{35}{6}n+\frac{3}{2} \qquad (۲۸)$$

۵-۲ـ fpo در محاسبه براساس لم (۲)

امر مهم در به کارگیری لم(۲) همانا یافتن ماتریس $H(n)$ به روش بازگشتی است وگرنه ضرب $H(n)\ \underline{C}(n)$ به هر حال اجتناب ناپذیر است. در روش بازگشتی با شروع از $H(0)$ نیاز به محاسبه عناصر بردار $\underline{f}(n)$ داریم. براساس تعریف (۷) حداقل: $fpo_{comb}(n,i)=2(n-i)-1$ عمل برای محاسبه عنصر $f_{n,i}$ انجام می‌شود. بنابر این ایجاد تمام عناصر $f(n)$ مجموعاً:

$$fpo_f(n)=\sum_{i=0}^{n} fpo_{comb}(n,i)=n^2-1 \qquad (۲۹)$$

عمل نقطه شناور می طلبد.

امّا جالب است که عملیات تولید $H(k)$ در هر مرحله تنها $(k+1)^2$ عمل جمع است که بین عناصر ماتریس ها در تعریف ۲۳-۱ صورت می گیرد. با افزودن تعداد $fpo_t(k)$ به هر مرحله، به سادگی می توان نشان داد که تشکیل $H(n)$ تعداد عملیات نقطه شناور کمتری را نسبت به روش ضرب مستقیم نیاز دارد:

$$fpo_H(n)=\sum_{i=0}^{n} 2i(i+1)=\frac{2}{3}n^3+2n^2+\frac{4}{3}n \qquad (۳۰)$$

از مقایسه روابط (۲۷ - ۲۸) و (۳۰) ملاحظه می شودکه:

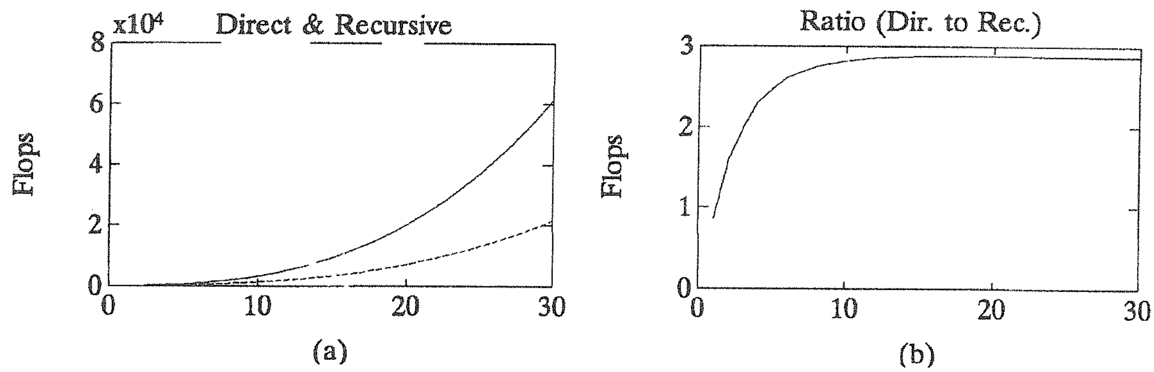
$$\lim_{n\to\infty}\frac{fpo_{dir}(n)}{fpo_H(n)}=\frac{\frac{11}{6}}{\frac{2}{3}}=\frac{11}{4}\approx 3$$

در شکل (۱) این دو مقدار و نسبت آنها را برای مقادیر مختلف n می بینیم.

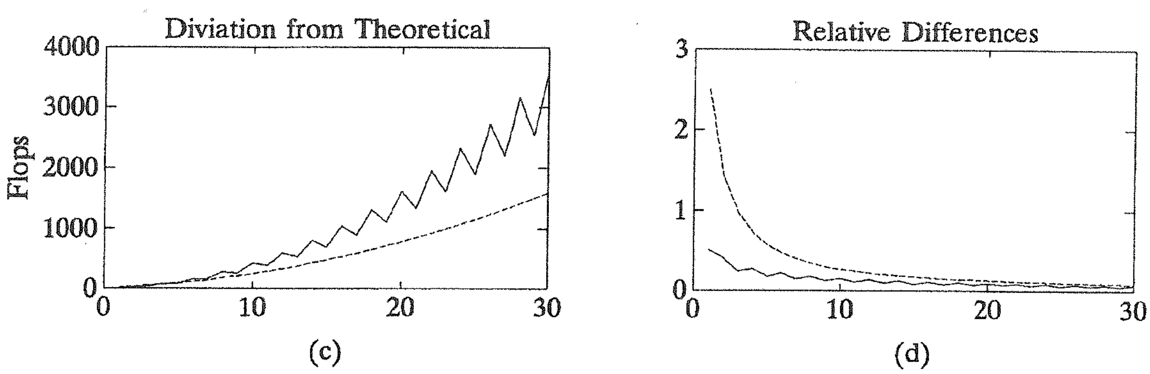
در انتها لازم است به دو نکته توجه داشته باشیم.

اولاً مقادیر fpo محاسبه شده در فوق با نتایج عملی در کامپیوتر قدری متفاوتند و این به دلیل ساختار خاصی است که در نرم افزار وجود دارد. استخراج ریشه تفاوت مختصر، چندان مشکل نیست امّا جهت اختصار از آن صرفنظر می کنیم. تفاوت مقادیر واقعی با مقادیر تئوری را برای هر دو حالت روش مستقیم و روش بازگشتی در شکل (۲) ملاحظه می کنیم.

ثانیاً نتیجه مهم دیگری که از لم (۲) حاصل می شود این است که با محاسبه $H(n)$ یک بار و برای همیشه، تبدیل Bilinear به صورت یک ترکیب خطی از ضرایب چند جمله‌ای $C(p)$ از طریق این ماتریس قابل محاسبه است. این بدان معنی است که به انجام عملیات ضرب چند جمله‌ای ها یا به توان رساندن آنها به کرّات و در هر تبدیل نی



شکل (۱)



شکل (۲)

۶ـ نتیجه‌گیری

در این مقاله نشان داده شد که تبدیل Bilinear چیزی جز یک ترکیب خطی از ضرایب چند جمله‌ای حوزه زمان پیوسته تحت یک ماتریس تبدیل نمی‌باشد. همچنین اثبات شد که ماتریس این تبدیل می‌تواند تحت یک رابطه بازگشتی به سادگی به دست آید. در نهایت نشان داده شد که تعداد عملیات نقطه شناور در محاسبه ضرایب چند جمله‌ای تبدیل یافته تقریباً ۳ برابر کمتر است نسبت به وقتی که ضرایب به طور مستقیم با بتوان رساندن و ضرب چند جمله‌ای‌ها به دست آیند.

زیرنویس

1- floating point opperation (fpo)

منابع:

[1] G.F. Fralin, J. D. Powell and Workmann L: Digital control of Dynamic systems, Addison-Wesley, 1990.

[2] K.J. Astrom, Wihenmark, B., Computer Controlled Systems, Theory & Design, Prentice-Hall, 1990.

[3] J.J. D'Azzo, Houples, C.H. Linear Control System, Analysis and Design, Conventional & Modern, McGraw-Hill, 1981.

[4] G.F. Franklin, Powell, J.D., Feedback Control of Dynamic Systems, Addison-Wesley, 1986.

[5] Ogata, k, Discrete Time Control Systems, Prentice-Hall, 1987.

[6] Mendel, J. M., Lessons in Digital Estimation Theory, Prentice-Hall, 1987.

[7] Scharf, L.L., Statistical Signal Processing Detection, Estimation and Time Series Analysis, Addison-Wesley, 1991.