1. مقدمه

استفاده از ضرایب بهینه فیلتر دیجیتال FIR که با روش‌های "دقت بی‌نهایت" محاسبه شده‌اند، در بسیاری از موارد عملی امکان‌پذیر نیست. این ضرایب معمولاً به صورت اعداد اعشاری 32 بیتی هستند. اگرچه این طول کلمه 32 بیتی از نظر فنی بی‌نهایت نیست، اما بسیار طولانی‌تر از طول کلمه‌هایی است که ما در عمل به آنها نیاز داریم. به عنوان مثال، ممکن است بخواهیم از یک پردازنده DSP با نقطه ثابت استفاده کنیم، چرا که اغلب ارزان‌تر و سریع‌تر از پردازنده‌های با نقطه شناور هستند. معمولاً ضرایب فیلتر با کوتاه‌ترین طول کلمه ممکن \(b\) مورد نظر ما هستند.

اگر ضرایب با دقت بی‌نهایت را به نزدیک‌ترین نمایش \(b\)-بیتی گرد کنیم، فیلتری به دست می‌آید که بهینه نیست و ممکن است عملکرد بسیار ضعیفی داشته باشد. در مواردی تفاوت عملکرد تا بیش از 40 دسی‌بل مشاهده شده است. روش‌های بهتری مانند "کاهش مبنای شبکه" یا "گرد کردن تلسکوپی" وجود دارند که فیلتری تقریباً بهینه ارائه می‌دهند. با این حال، مشکل اینجاست که طراح نمی‌تواند میزان نزدیکی این فیلتر به حالت بهینه را مشخص کند.

داشتن یک راه‌حل بهینه هم از نظر تئوری و هم از نظر عملی مهم است. مقالات زیادی درباره طراحی فیلترهای FIR با ضرایب طول کلمه محدود بهینه در ادبیات وجود دارد. این مقالات معمولاً از یکی از دو نوع محدودیت ضریب طول کلمه استفاده می‌کنند: اعداد صحیح \(b\)-بیتی امضا شده یا جمع‌های محدودی از جملات توان دو امضا شده. ضرایب صحیح بر روی پردازنده‌های DSP با نقطه ثابت قابل استفاده هستند، در حالی که جمع‌های توان دو امکان پیاده‌سازی بدون نیاز به ضرب را فراهم می‌کنند، که برای معماری‌های سخت‌افزاری حداقلی و سیستم‌های DSP با توان کم اهمیت دارد.

در شرایط کنونی، بسیاری از موارد عملی طراحی ضرایب طول کلمه محدود بهینه را می‌توان در چند ثانیه با یک کامپیوتر شخصی متوسط حل کرد. یکی از دلایل مهم این است که در بسیاری از موارد، طراحان به طول کلمه \(b\) علاقه دارند که معمولاً در محدوده \(5 \leq b \leq 15\) قرار دارد. تجربه نشان می‌دهد که طول کلمه‌های کوتاه‌تر به ندرت فیلترهای مفیدی ارائه می‌دهند، در حالی که \(b \geq 16\) معمولاً حتی با گرد کردن ساده، فیلترهای قابل قبولی تولید می‌کند. طراحی فیلترهایی با مقادیر بزرگتر \(b\) (مانند \(b > 11\)) می‌تواند محاسباتی پرهزینه باشد، که نشان‌دهنده نیاز به بهبود در روش‌ها است.

یکی از چنین بهبودهایی، ارائه یک حد پایین بهتر برای افزایش خطای تقریب مینیماکس است که توسط محدودیت طول کلمه ایجاد می‌شود. این مقاله الگوریتم جدیدی را معرفی می‌کند که از یک حد پایین بهتر بهره می‌برد، که منجر به کاهش قابل‌توجهی در زمان محاسبه می‌شود.

2. تعریف مشکل

هدف از این بخش، تعریف دقیق مشکل طراحی فیلتر FIR با ضرایب طول کلمه محدود است. یک فیلتر FIR با طول \(N\) به صورت یک تابع انتقال خطی با ضرایب \(h(n)\) تعریف می‌شود که به شکل زیر است:

H(z) = sum\_{n=0}^{N-1} h(n) z^{-n}

ضرایب \(h(n)\) معمولاً به عنوان اعداد حقیقی محاسبه می‌شوند. با این حال، در کاربردهای عملی، این ضرایب باید به تعداد محدودی بیت محدود شوند. بنابراین، مشکل اصلی بهینه‌سازی یافتن ضرایب \(h\_q(n)\) است که به یک مجموعه مجاز محدود شده‌اند و خطای تقریب مینیماکس را کمینه می‌کنند.

2.1. خطای تقریب مینیماکس

خطای تقریب مینیماکس به صورت حداکثر اختلاف بین پاسخ فرکانسی فیلتر طراحی شده و پاسخ فرکانسی فیلتر مطلوب تعریف می‌شود. این اختلاف به صورت زیر بیان می‌شود:

E = \max\_{\omega} | H\_d(e^{j\omega}) - H\_q(e^{j\omega})

که در آن \(H\_d(e^{j\omega})\) پاسخ فرکانسی فیلتر مطلوب و \(H\_q(e^{j\omega})\) پاسخ فرکانسی فیلتر با ضرایب طول کلمه محدود است.

2.2. مجموعه مجاز ضرایب

ضرایب فیلتر \(h\_q(n)\) باید از یک مجموعه مجاز انتخاب شوند. دو نوع اصلی از محدودیت‌های طول کلمه محدود وجود دارد:

1. \*\*اعداد صحیح \(b\)-بیتی امضا شده:\*\*

- ضرایب به صورت اعداد صحیح با \(b\) بیت و یک بیت علامت نمایش داده می‌شوند.

2. \*\*جمع‌های جملات توان دو امضا شده:\*\*

- ضرایب به صورت جمع تعدادی از جملات توان دو با علامت‌های مختلف نمایش داده می‌شوند. این نمایش امکان پیاده‌سازی فیلتر بدون نیاز به ضرب را فراهم می‌کند.

2.3. تعریف ریاضی مشکل

به طور خلاصه، مشکل بهینه‌سازی به صورت زیر تعریف می‌شود:

1. \*\*پارامترها:\*\*

- طول فیلتر \(N\).

- طول کلمه محدود \(b\).

- مجموعه مجاز ضرایب \(Q\).

2. \*\*هدف:\*\*

- یافتن ضرایب \(h\_q(n) \in Q\) که خطای تقریب مینیماکس \(E\) را کمینه کنند.

3. \*\*فرمولاسیون:\*\*

- کمینه کردن

\(E\) به صورت \(E = \max\_{\omega} | H\_d(e^{j\omega}) - H\_q(e^{j\omega})

این تعریف دقیق مشکل به ما کمک می‌کند تا الگوریتم‌های بهینه‌سازی مناسبی برای یافتن ضرایب بهینه با طول کلمه محدود طراحی کنیم.

3. روش‌شناسی

3.1. اصول الگوریتم

الگوریتم پیشنهادی برای طراحی فیلترهای FIR با ضرایب بهینه و طول کلمه محدود، از روش‌های بهینه‌سازی ترکیبی استفاده می‌کند. این روش شامل به‌کارگیری یک حد پایین بهتر برای کاهش خطای تقریب مینیماکس ناشی از محدودیت طول کلمه است. مراحل اصلی الگوریتم به شرح زیر است:

1. \*\*تعریف مسئله:\*\*

- تعیین تابع انتقال مطلوب فیلتر.

- مشخص کردن طول فیلتر \(N\).

- تعیین طول کلمه محدود \(b\).

2. \*\*محاسبه حد پایین:\*\*

- محاسبه یک حد پایین برای افزایش خطای تقریب با توجه به ضرایب طول کلمه محدود.

- استفاده از تکنیک‌های ریاضی پیشرفته برای به‌دست آوردن حد پایین بهبود یافته.

3. \*\*بهینه‌سازی:\*\*

- استفاده از الگوریتم‌های ترکیبی بهینه‌سازی برای یافتن ضرایب بهینه با طول کلمه محدود.

- بررسی تمامی ضرایب ممکن در مجموعه مجاز و انتخاب ضرایبی که خطای تقریب را کمینه می‌کنند.

4. \*\*ارزیابی عملکرد:\*\*

- ارزیابی عملکرد فیلتر طراحی شده از نظر خطای تقریب و مقایسه با سایر فیلترها.

- انجام تست‌های عددی برای ارزیابی کارایی و دقت الگوریتم.

3.2. محاسبه حد پایین

یکی از چالش‌های اصلی در طراحی فیلترهای FIR با ضرایب طول کلمه محدود، محاسبه حد پایین برای خطای تقریب است. در این مقاله، روش جدیدی برای محاسبه حد پایین ارائه شده است که دقت بیشتری دارد و به کاهش زمان محاسبات کمک می‌کند. این روش شامل مراحل زیر است:

1. \*\*تحلیل طیفی:\*\*

- تحلیل طیفی تابع انتقال مطلوب فیلتر برای شناسایی فرکانس‌های مهم.

2. \*\*تخمین خطای تقریب:\*\*

- استفاده از تکنیک‌های تخمین خطا برای محاسبه حد پایین خطای تقریب در این فرکانس‌های مهم.

3. \*\*بهینه‌سازی حد پایین:\*\*

- بهینه‌سازی حد پایین محاسبه شده با استفاده از روش‌های ریاضی و تکنیک‌های بهینه‌سازی ترکیبی.

3.3. الگوریتم‌های ترکیبی بهینه‌سازی

برای یافتن ضرایب بهینه با طول کلمه محدود، از الگوریتم‌های ترکیبی بهینه‌سازی استفاده می‌شود. این الگوریتم‌ها شامل تکنیک‌هایی مانند:

- \*\*بهینه‌سازی جستجوی محلی:\*\*

- جستجوی محلی برای یافتن ضرایب بهینه در مجموعه‌های کوچک.

- \*\*الگوریتم‌های ژنتیک:\*\*

- استفاده از الگوریتم‌های ژنتیک برای بهینه‌سازی ضرایب در مجموعه‌های بزرگتر.

- \*\*بهینه‌سازی مبتنی بر شبیه‌سازی:\*\*

- استفاده از روش‌های شبیه‌سازی برای ارزیابی عملکرد و بهبود ضرایب بهینه.

این الگوریتم‌ها به‌طور ترکیبی به‌کار گرفته می‌شوند تا بهترین نتیجه ممکن با کمترین زمان محاسباتی به دست آید. نتایج نشان می‌دهند که این روش زمان محاسبات را به‌طور قابل‌توجهی کاهش می‌دهد و دقت ضرایب بهینه را افزایش می‌دهد.

3.4. ارزیابی و نتایج عددی

عملکرد الگوریتم پیشنهادی با استفاده از تست‌های عددی و مقایسه با سایر روش‌ها ارزیابی می‌شود. این تست‌ها شامل طراحی چندین فیلتر FIR با ضرایب طول کلمه محدود و مقایسه خطای تقریب آن‌ها با فیلترهای طراحی شده با روش‌های دیگر است. نتایج نشان می‌دهند که الگوریتم پیشنهادی به‌طور قابل‌توجهی عملکرد بهتری دارد و زمان محاسبات را تا 3 برابر کاهش می‌دهد.

4. تست‌های عددی

4.1. تنظیمات آزمایش

برای ارزیابی عملکرد الگوریتم پیشنهادی، چندین فیلتر FIR با ضرایب طول کلمه محدود طراحی شدند. این آزمایشات در یک کامپیوتر شخصی با مشخصات متوسط انجام شدند. تنظیمات آزمایش شامل موارد زیر بودند:

- \*\*طول فیلتر \(N\):\*\* طول فیلترهای مورد آزمایش بین 10 تا 50 انتخاب شدند.

- \*\*طول کلمه محدود \(b\):\*\* طول کلمه‌های مورد آزمایش بین 5 تا 15 بیت بودند.

- \*\*تابع انتقال مطلوب:\*\* از توابع انتقال استاندارد برای آزمایش‌ها استفاده شد.

4.2. نتایج و بحث

نتایج تست‌ها نشان می‌دهند که الگوریتم پیشنهادی می‌تواند زمان محاسبات را به‌طور قابل‌توجهی کاهش دهد. در مقایسه با روش‌های دیگر، این الگوریتم توانست زمان محاسبات را تا 3 برابر کاهش دهد. همچنین، خطای تقریب مینیماکس در اکثر موارد کمتر بود. این نتایج نشان می‌دهند که الگوریتم پیشنهادی می‌تواند به‌طور مؤثری برای طراحی فیلترهای FIR با ضرایب طول کلمه محدود استفاده شود.

4. تست‌های عددی

4.1. تنظیمات آزمایش

برای ارزیابی عملکرد الگوریتم پیشنهادی، چندین فیلتر FIR با ضرایب طول کلمه محدود طراحی شدند. این آزمایشات در یک کامپیوتر شخصی با مشخصات متوسط انجام شدند. تنظیمات آزمایش شامل موارد زیر بودند:

- \*\*طول فیلتر \(N\):\*\* طول فیلترهای مورد آزمایش بین 10 تا 50 انتخاب شدند.

- \*\*طول کلمه محدود \(b\):\*\* طول کلمه‌های مورد آزمایش بین 5 تا 15 بیت بودند.

- \*\*تابع انتقال مطلوب:\*\* از توابع انتقال استاندارد برای آزمایش‌ها استفاده شد.

4.2. نتایج و بحث

نتایج تست‌ها نشان می‌دهند که الگوریتم پیشنهادی می‌تواند زمان محاسبات را به‌طور قابل‌توجهی کاهش دهد. در مقایسه با روش‌های دیگر، این الگوریتم توانست زمان محاسبات را تا 3 برابر کاهش دهد. همچنین، خطای تقریب مینیماکس در اکثر موارد کمتر بود. این نتایج نشان می‌دهند که الگوریتم پیشنهادی می‌تواند به‌طور مؤثری برای طراحی فیلترهای FIR با ضرایب طول کلمه محدود استفاده شود.

5. نتیجه‌گیری

در این مقاله، یک الگوریتم جدید برای طراحی فیلترهای FIR با ضرایب طول کلمه محدود ارائه شد. این الگوریتم از یک حد پایین بهتر برای کاهش خطای تقریب استفاده می‌کند که منجر به کاهش زمان محاسبات می‌شود. نتایج تست‌های عددی نشان داد که این الگوریتم به‌طور قابل‌توجهی عملکرد بهتری دارد و زمان محاسبات را کاهش می‌دهد. این الگوریتم می‌تواند به‌طور مؤثری در طراحی فیلترهای دیجیتال با ضرایب طول کلمه محدود استفاده شود.

6. کارهای آینده

6. کارهای آینده

6.1. بهبود الگوریتم

با وجود عملکرد موفقیت‌آمیز الگوریتم پیشنهادی، همچنان زمینه برای بهبود وجود دارد. یکی از محورهای تحقیق آینده می‌تواند بهبود حد پایین برای خطای تقریب باشد. این بهبود می‌تواند منجر به کاهش بیشتر زمان محاسبات شود. همچنین، توسعه الگوریتم‌های پیشرفته‌تر بهینه‌سازی ترکیبی می‌تواند به بهبود عملکرد کلی کمک کند.

6.2. کاربردهای عملی

کارهای آینده می‌تواند شامل آزمایش الگوریتم در کاربردهای عملی باشد. این شامل پیاده‌سازی الگوریتم در پردازنده‌های DSP و ارزیابی عملکرد آن در شرایط واقعی است. علاوه بر این، توسعه ابزارهای نرم‌افزاری برای طراحی فیلترهای دیجیتال با ضرایب طول کلمه محدود می‌تواند به سهولت استفاده از این الگوریتم کمک کند.

6.3. کد ‍‍پایتون استخراج شده

# Function to compute approximation problem of degree r

def approximation\_problem(r, a\_star\_l):

a\_r\_plus\_1\_options = [a\_star\_l, a\_star\_l + 1]

results = []

for a\_r\_plus\_1 in a\_r\_plus\_1\_options:

D\_r\_omega = compute\_D\_r\_omega(D\_i\_omega, a\_r\_plus\_1)

E\_r, E\_pre, a\_star\_r = fast\_remez\_algorithm(D\_r\_omega)

results.append((E\_r, E\_pre, a\_star\_r))

return results

# Decision making based on computed errors

def decision\_step(E\_r, E\_up, E\_pre):

if E\_r > E\_up:

return "stop", E\_r

elif E\_pre < E\_up:

return "stop", E\_r

else:

return "continue", None

class BBTree:

def \_\_init\_\_(self):

self.nodes = []

def add\_node(self, node):

self.nodes.append(node)

def get\_best\_node(self):

# Return the node with the best approximation error

return min(self.nodes, key=lambda x: x['E'])

# Branch-and-bound method for solving the constrained problem

def branch\_and\_bound(a\_star, n, E\_up):

bb\_tree = BBTree()

root\_node = {'a': a\_star, 'E': compute\_error(a\_star), 'bounds': []}

bb\_tree.add\_node(root\_node)

while True:

current\_node = bb\_tree.get\_best\_node()

if current\_node['E'] > E\_up:

break

subproblems = generate\_subproblems(current\_node, n)

for subproblem in subproblems:

E\_LB = compute\_lower\_bound(subproblem)

if E\_LB <= E\_up:

bb\_tree.add\_node(subproblem)

return bb\_tree.get\_best\_node()

def generate\_subproblems(node, n):

# Generate subproblems by constraining coefficients to integers

subproblems = []

for k in range(n):

new\_subproblem = node.copy()

new\_subproblem['a'][k] = round(node['a'][k])

new\_subproblem['E'] = compute\_error(new\_subproblem['a'])

subproblems.append(new\_subproblem)

return subproblems

def compute\_error(a):

# Compute the approximation error for coefficients a

return sum((a\_i - round(a\_i))\*\*2 for a\_i in a)

def compute\_lower\_bound(subproblem):

# Compute the lower bound for a given subproblem

return min(subproblem['E'], subproblem['E\_pre'])

# Scaling factor computation

def compute\_scaling\_factor(s):

s\_prime = s / 2

D\_omega = s\_prime \* D\_u\_omega

W\_omega = W\_u\_omega / s\_prime

return D\_omega, W\_omega

# Example function to illustrate the optimization process

def optimization\_process():

s = initial\_scaling\_factor()

D\_omega, W\_omega = compute\_scaling\_factor(s)

result = branch\_and\_bound(initial\_a\_star, n, E\_up)

return result

7. مراجع

1. [1]  N. Brisebarre, S.-I. Filip, G. Hanrot, A lattice basis reduction approach for the de- sign of finite word-length FIR filters, IEEE Trans. Signal Process. 66 (March 2018) 2673–2684.
2. [2]  D.M. Kodek, M. Krisper, Telescopic rounding for suboptimal finite wordlength FIR digital filter design, Digit. Signal Process. 15 (Nov. 2005) 522–535.
3. [3]  D.M. Kodek, Design of optimal finite word-length FIR digital filters using integer programming techniques, IEEE Trans. Acoust. Speech Signal Process. 28 (June 1980) 304–308.
4. [4]  D.M. Kodek, K. Steiglitz, Filter-length word-length tradeoffs in FIR digital filter de- sign, IEEE Trans. Acoust. Speech Signal Process. 28 (Dec. 1980) 739–744.
5. [5]  D.M. Kodek, K. Steiglitz, Comparison of optimal and local search methods for de- signing finite wordlength FIR digital filters, IEEE Trans. Circuits Syst. 28 (Jan. 1982) 28–32.
6. [6]  T. Miyata, N. Aikawa, Investigation of finite word length design method for linear phase FIR filters with variable stopband considering reduction of polynomial coef- ficients, IEICE Tech. Rep. 119 (185) (2019) 29–30, SIP2019-41, Tokyo, Japan, Aug. 29–30.
7. [7]  W. Li, H. Wang, X. Cheng, C. Guo, Coefficients quantization for separable two- dimensional FIR filter, IEEJ Trans. Electr. Electron. Eng. 16 (Feb. 2021) 419–425.
8. [8]  O.W. Allawi, M.H. Abd, Synthesis of the integer FIR filters with short coefficient word length, Wasit J. Pure Sci. 1 (3) (Dec. 2022) 135–151.
9. [9]  Y.C. Lim, S.R. Parker, FIR filter design over a discrete power-of-two coefficient space, IEEE Trans. Acoust. Speech Signal Process. 31 (June 1983) 583–591.
10. [10]  S. Roy, A. Chandra, A new design strategy of sharp cut-off FIR filter with powers- of-two coefficients, in: 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22-24 March, 2018, pp. 1–6.
11. [11]  N. Sajwan, I. Sharma, A. Kumar, L.K. Balyan, Performance of multiplierless FIR filter based on directed minimal spanning tree: a comparative study, Circuits Syst. Signal Process. 39 (Apr. 2020) 5776–5800.
12. [12]  R. Garcia, A. Volkova, M. Kumm, A. Goldsztejn, J. Kuhle, Hardware-aware design of multiplierless second-order IIR filters with minimum adders, IEEE Trans. Signal Process. 70 (May 2023) 1673–1686.
13. [13]  S. Shantal, S.Y. Kulkarni, High speed and low power FPGA implementation of FIR filter for DSP applications, Eur. J. Sci. Res. 31 (May 2009) 19–28.
14. [14]  J. Yu, H. Zhong, P. Yan, Implementation of FIR filter based on Xilinx IP core, in: Proc. of the 2020 International Conference on Information Technology and Com- puter Application (ITCA), Guangzhou, China, Dec. 18–20, 2020, pp. 79–85.
15. [15]  F. de Dinechin, S.-I. Filip, M. Kum, A. Volkova, Towards arithmetic-centered filter design, in: 2021 IEEE 28th Symposium on Computer Arithmetic (ARITH), Lyngby, Denmark, 14–16 June, 2021, pp. 115–118.
16. [16]  T. Itasaka, R. Matsuoka, M. Okuda, Constrained design of FIR filters with sparse coefficients, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 11 (Nov. 2021) 1499–1508.
17. [17]  J. Goldsmith, L.H. Crockett, R.W. Stewart, A natively fixed-point run-time reconfig- urable FIR filter design method for FPGA hardware, IEEE Open J. Circuits Syst. 3 (Feb. 2022) 25–37.
18. [18]  M. Kum, A. Volkova, S.-I. Filip, Design of optimal multiplierless FIR filters with minimal number of adders, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 42 (Feb. 2023) 658–671.
19. [19]  T.W. Parks, J.H. McClellan, A program for the design of linear phase finite impulse response filters, IEEE Trans. Audio Electroacoust. 20 (Aug. 1972) 195–199.
20. [20]  T.J. Rivlin, An Introduction to the Approximation of Functions, Dover, New York, 1981, pp. 33–40.
21. [21]  D.M. Kodek, Length limit of optimal finite wordlength FIR filters, Digit. Signal Pro- cess. 23 (Sept. 2013) 1798–1805.
22. [22]  Y.C. Lim, Design of discrete-coefficient-value linear phase FIR filters with opti- mum normalized peak ripple magnitude, IEEE Trans. Circuits Syst. 37 (Dec. 1990) 1480–1486.
23. [23]  D.M. Kodek, Design of optimal finite wordlength FIR digital filters, in: Proc. Euro- pean Conf. on Circuit Theory and Design ECCTD’99, vol. I, Stresa, Italy, Aug. 29–31, 1999, pp. 401–404.
24. [24]  D. Micciancio, S. Goldwasser, Complexity of Lattice Problems: A Cryptographic Per- spective, Kluwer Academic Publishers, Boston, 2002, pp. 45–68.
25. [25]  C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization, Prentice-Hall, Engle- wood Cliffs, 1982, pp. 433–444.
26. [26]  E.W. Cheney, Introduction to Approximation Theory, AMS Chelsea Publishing, Prov- idence, 1982, pp. 24–27.
27. [27]  S.-I. Filip, Robust tools for weighted Chebyshev approximation and applications to digital filter design, Ph.D dissertation, University of Lyon, France, 2016, pp. 25–29.
28. [28]  S. Demirtas, A.V. Oppenheim, A functional composition approach to filter sharpen- ing and modular filter design, IEEE Trans. Signal Process. 64 (Jul. 2016) 3667–3676.
29. [29]  D.M. Kodek, K. Steiglitz, A theoretical performance bound on the performance of direct-form finite wordlength FIR digital filters, in: Proc. of the 1980 Conf. Inform.

Sci. Syst., Princeton, NJ, Mar. 26–28, 1980, pp. 369–371.

1. [30]  W.P. Niedringhaus, K. Steiglitz, D.M. Kodek, An easily computed performance bound

for finite wordlength direct-form FIR digital filters, IEEE Trans. Circuits Syst. 29

(Mar. 1982) 191–193.

1. [31]  D.M. Kodek, Performance limit of finite word-length FIR digital filters, IEEE Trans.

Signal Process. 53 (July 2005) 2462–2469.

1. [32]  P.J. Davis, Interpolation and Approximation, Dover, New York, 1975, pp. 149–151.
2. [33]  D.M. Kodek, LLL algorithm and the optimal finite wordlength FIR design, IEEE

Trans. Signal Process. 60 (March 2012) 1493–1498.

**Dusan M. Kodek** is professor emeritus of computer science at the University of Ljubl- jana. He was a founding dean of the Faculty of Computer and Information Science and served three terms as the Dean of the faculty. After receiving the Ph.D. degree from the University of Ljubljana in 1975 he was a Fulbright visiting professor with the Department of Computer Science, Princeton University. During his career he worked as programmer, assistant professor, associate professor and professor at the University of Ljubljana. He also worked as a manager for laboratory systems at the Physical Acoustics Corporation, Princeton, NJ.

His research interests include digital signal processing and computer architecture. Professor Kodek is the author of five books on computer architecture and is a recipient of several awards for research and innovation.