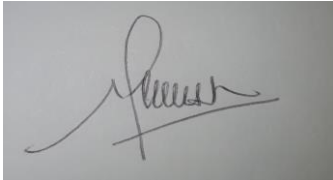




**UNIVERSITAS PAKUAN
FAKULTAS TEKNIK
PRODI ELEKTRO**

*(Disi Kode
Dokumen)*

RENCANA PEMBELAJARAN SEMESTER

Mata Kuliah (MK)	Kode	Rumpun MK	Bobot SKS		Semester	Tgl Penyusunan
Pengolahan Sinyal Digital	554KB62801 / 3 sks	Wajib	T = 3	P = 3	6	15 Mei 20223
Otorisasi/Pengesahan	Dosen Pengembang RPS  Ir Herry Satria Utama MT	Koordinator RMK (jika ada) TTD		Kaprodi TTD		
Capaian Pembelajaran	CPL-PRODI yang dibebankan pada MK					
	CPL1 (S4)	Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang				

		<p>keahliannya secara mandiri: Menguasai pengetahuan sains dasar, sains komputer dan sains rekayasa yang diperlukan untuk menganalisis dan merancang perangkat lunak, serta sistem yang terdiri atas perangkat keras dan perangkat lunak</p>
	CPL2 (P3)	<p>a. Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang keahliannya, berdasarkan hasil analisis informasi dan data;</p> <p>b. Mampu merencanakan, menganalisis, merancang dan memadukan teknik pengolahan sinyal audio dan video .</p>
	CPL3 (KU2)	<p>Diharapkan mahasiswa mampu menguasai pengetahuan untuk menganalisis menghasilkan kualitas sinyal audio dan sinyal video . Sehingga mahasiswa mampu memberikan solusi untuk memecahkan masalah gangguan dan dapat meningkatkan keterampilan dalam memahami aplikasi pengolahan sinyal Audio dan video secara seperti Mp3 dan Mp4.</p>
	CPL4 (KK4)	<p>Diharapkan mahasiswa mampu menguasai pengetahuan untuk menganalisis menghasilkan kualitas sinyal audio dan sinyal video . Sehingga mahasiswa mampu memberikan solusi untuk memecahkan masalah gangguan dan dapat meningkatkan keterampilan dalam memahami aplikasi pengolahan sinyal Audio dan video secara seperti Mp3 dan Mp4.</p>

- 1. Bertakwa kepada Tuhan Yang Maha Esa dan mampu menunjukkan sikap religius yang diaktualisasikan dalam kehidupan sebagai warganegara yang cinta tanah air, memiliki nasionalisme serta rasa tanggungjawab pada negara dan bangsa;**
- 2. Menginternalisasi semangat kemandirian, kejuangan, dan kewirausahaan.**
- 3. Menguasai pengetahuan sains dasar, sains komputer dan sains rekayasa yang diperlukan untuk menganalisis dan merancang perangkat lunak, serta sistem yang terdiri atas perangkat keras dan perangkat lunak.**
- 4. Memiliki pengetahuan yang cukup luas dalam bidang teknik elektro termasuk teknik tenaga listrik, teknik telekomunikasi, teknik kendali, dan sistem komputer.**
- 5. Mampu menerapkan pemikiran logis, kritis, sistematis, dan inovatif dalam konteks pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora yang sesuai dengan bidang keahliannya.**
- 6. Mampu mengidentifikasi permasalahan-permasalahan yang timbul dalam bidang teknik elektro, dengan menggunakan prinsip dasar matematika dan fisika dalam menyelesaikan permasalahan, untuk memberikan petunjuk dalam memilih berbagai alternatif solusi secara mandiri dan kelompok**
- 7. Mampu merencanakan, menganalisis, merancang dan memadukan teknik energi dan ketenagalistrikan, serta evaluasi penggunaan energi.**
- 8. Mampu menerapkan konversi energi baru dan terbarukan, serta energi konvensional.**
- 9. Mampu mengidentifikasi kebutuhan komponen sistem tenaga listrik dan elektronika daya**
- 10 Mampu merencanakan, menganalisis, merancang dan mengevaluasi sistem komunikasi, serta pengolahan sinyal informasinya.**
- 11 Mampu menggunakan mikrokontroler dalam permasalahan bidang teknik elektro.**

Capaian Pembelajaran Maya Kuliah (CPMK)

CPMK1

Ruang Lingkup DSP:

Pemrosesan Sinyal Digital adalah salah satu teknologi paling kuat yang akan membentuk sains dan teknik di abad kedua puluh satu. Perubahan revolusioner telah terjadi dibuat dalam berbagai bidang: komunikasi, pencitraan medis, radar & sonar, tinggi reproduksi musik kesetiaan, dan pencarian minyak, untuk menyebutkan beberapa saja. Masing-masing bidang ini telah mengembangkan teknologi DSP yang dalam, dengan algoritmanya sendiri, matematika, dan teknik khusus. Kombinasi nafas dan kedalaman ini membuat tidak mungkin bagi siapapun satu individu untuk menguasai semua teknologi DSP yang telah dikembangkan. DSP pendidikan melibatkan dua tugas: mempelajari konsep umum yang berlaku di lapangan secara keseluruhan, dan mempelajari teknik khusus untuk bidang minat khusus Anda. Bab ini dimulai perjalanan kita ke dunia Pemrosesan Sinyal Digital dengan menjelaskan efek dramatisnya yang telah dibuat DSP di berbagai bidang. Revolusi telah dimulai.

2.Akar dari DSP ada di tahun 1960-an dan 1970-an ketika komputer digital pertama kali menjadi tersedia. Komputer mahal selama era ini, dan DSP hanya terbatas pada beberapa aplikasi kritis. Upaya perintisan dilakukan di empat bidang utama: radar & sonar, di mana keamanan nasional terancam; eksplorasi minyak, di mana uang dalam jumlah besar bisa dibuat; eksplorasi ruang angkasa, dimana data tidak tergantikan; dan pencitraan medis, di mana nyawa bisa diselamatkan.

Revolusi komputer pada 1980-an dan 1990-an menyebabkan DSP

		<p>meledak dengan yang baru aplikasi. Alih-alih dimotivasi oleh kebutuhan militer dan pemerintah, DSP lah tiba-tiba didorong oleh pasar komersial. Siapapun yang mengira mereka bisa membuat uang di bidang yang berkembang pesat tiba-tiba menjadi vendor DSP. DSP mencapai publik dalam produk-produk seperti: telepon seluler, pemutar CD, dan suara elektronik surat. Gambar 1-1 mengilustrasikan beberapa dari beragam aplikasi ini.</p> <p>Revolusi teknologi ini terjadi dari atas ke bawah. Pada awal 1980-an, DSP diajarkan sebagai kursus tingkat pascasarjana di bidang teknik listrik. Satu dekade kemudian, DSP melakukannya menjadi bagian standar dari kurikulum sarjana. Saat ini, DSP adalah keterampilan dasar dibutuhkan oleh para ilmuwan dan insinyur: pada bidang-bidang seerti berikut:</p> <p>DSP Pada Telekomunikas DSP pada Multiplexing DSP pada Kompresi sinyal</p>
	CPMK2	<p>ADC & DAC yaitu Interface untuk proses sinyal analog menjadi sinyal digital, dan konversi sinyal digital menjadi sinyal Analog. Sebagian besar sinyal yang ditemui secara langsung dalam sains dan teknik bersifat kontinu: intensitas cahaya yang berubah seiring jarak; tegangan yang bervariasi dari waktu ke waktu; laju reaksi kimia yang bergantung pada suhu, dll. Konversi Analog-ke-Digital (ADC) dan Konversi Digital-ke-Analog (DAC) adalah proses yang memungkinkan komputer digital berinteraksi dengan sinyal sehari-hari ini. Digital informasi berbeda dari rekan berkelanjutannya dalam dua hal penting: diambil sampelnya, dan itu dikuantisasi. Keduanya membatasi seberapa banyak informasi yang dapat</p>

		<p>dikandung sinyal digital. Ini bab tentang manajemen informasi: memahami informasi apa yang perlu Anda simpan, dan informasi apa yang Anda mampu untuk kehilangan. Pada gilirannya, ini menentukan pemilihan pengambilan sampel frekuensi, jumlah bit, dan jenis pemfilteran analog yang diperlukan untuk konversi antar analog dan dunia digital..</p>
	<p>CPMK3</p>	<p>SISTEM LTI (<i>Linear Time Invariant</i>)</p> <p>Sistem ini adalah invarian waktu linier (LTI) jika memenuhi properti linieritas dan varians waktu Bab ini akan mempelajari sistem LTI hampir secara eksklusif, karena paling mudah sistem untuk bekerja, dan ideal untuk dianalisis dan dirancang</p> <p>Linier :</p> <p>Ada 3 persyaratan untuk linieritas. Sebuah fungsi harus memenuhi ketiganya untuk disebut "linier".</p> <ol style="list-style-type: none"> 1. Aditivitas: Input dari $x_3(t) = x_1(t) + x_2(t)$ menghasilkan output $y_3(t) = y_1(t) + y_2(t)$. Homogenitas: Input dari ax_1 menghasilkan output ay_1 Jika $x(t) = 0, y(t) = 0$. 2. "Linear" dalam pengertian ini bukanlah kata yang sama seperti yang digunakan dalam aljabar konvensional atau geometri. Secara khusus, linieritas dalam aplikasi sinyal tidak ada hubungannya dengan garis lurus garis. Berikut ini contoh kecilnya: $y(t) = x(t) + 5$ <p>Fungsi ini tidak linier, karena ketika $x(t) = 0, y(t) = 5$ (gagal memenuhi persyaratan 3). Ini mungkin mengejutkan orang, karena persamaan ini adalah persamaan untuk garis lurus.</p>

Kita dapat menggabungkan tiga persyaratan menjadi satu persamaan: Dalam sistem linier, sebuah masukan dari $a_1x_1(t) + a_2x_2(t)$ menghasilkan output $a_1y_1(t) + a_2y_2(t)$.

Aditif

Suatu sistem dikatakan aditif jika sejumlah masukan menghasilkan sejumlah keluaran. Untuk menguji aditif, kita perlu membuat dua input yang berubah – ubah, $x_1(t)$ dan $x_2(t)$. Kami kemudian menggunakan masukan ini untuk menghasilkan dua keluaran masing-masing:

$$y_1(t) = f(x_1(t))$$

$$y_2(t) = f(x_2(t))$$

Dimana

$$x_1(t) = cx_2(t)$$

Di mana c adalah konstanta arbitrer. Jika ini kasusnya maka sistem homogen jika untuk sembarang c .

$$y_1(t) = cy_2(t)$$

Varians Waktu

Jika sinyal masukan $x(t)$ menghasilkan keluaran $y(t)$ maka setiap kali masukan digeser, $x(t + \delta)$, hasilnya dalam output bergeser waktu $y(t + \delta)$. Properti ini dapat dipenuhi jika fungsi transfer sistem bukan merupakan fungsi waktu kecuali diekspresikan oleh input dan output.

Selain linier, atau time-invariant, ada sejumlah properti lain yang kita bisa mengidentifikasi dalam suatu fungsi:

Memori

Suatu sistem dikatakan memiliki memori jika keluaran dari sistem bergantung pada masukan sebelumnya (atau masukan masa depan) ke sistem. Suatu sistem disebut tanpa memori jika keluarannya hanya tergantung pada masukan saat ini. Sistem tanpa memori

lebih mudah digunakan, tetapi sistem dengan memori lebih umum dalam aplikasi pemrosesan sinyal digital.

Kasualitas

Kausalitas adalah properti yang sangat mirip dengan memori. Suatu sistem disebut kausal jika hanya tergantung pada masukan masa lalu atau saat ini. Suatu sistem disebut non kausal jika keluaran dari sistem tersebut tergantung pada masukan masa depan.

Stabilitas

Stabilitas adalah konsep yang sangat penting dalam sistem, tetapi juga salah satu fungsi yang paling sulit properti untuk membuktikan. Ada beberapa kriteria berbeda untuk stabilitas sistem, tetapi yang paling banyak persyaratan umum adalah bahwa sistem harus menghasilkan keluaran yang terbatas ketika mengalami sebuah masukan yang terbatas. Misalnya, jika kita menerapkan 5 volt ke terminal input dari rangkaian tertentu, kita akan menyukainya jika keluaran rangkaian tidak mendekati tak terhingga, dan rangkaian itu sendiri tidak meleleh atau meledak. Jenis stabilitas ini sering disebut dengan stabilitas "Input Terikat, Output Terikat", atau BIBO.

Operator Linear

Operator matematika yang memenuhi properti linieritas dikenal sebagai operator linier. Berikut beberapa operator linier yang umum:

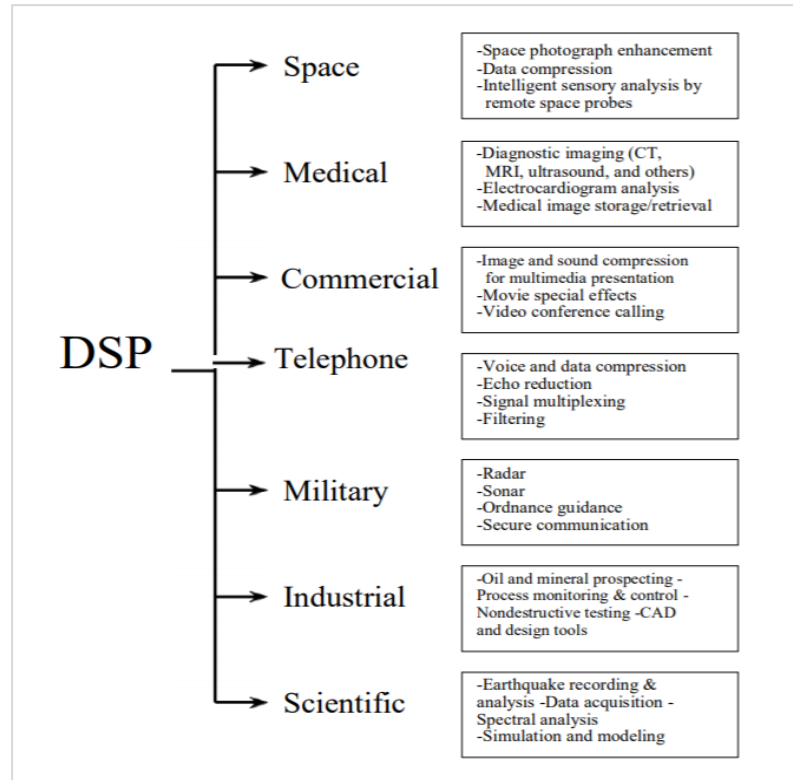
1. Turunan
2. Integral
3. Transformasi Fourier

	CMK4	FIR Filtering and Convolution: 4.1.1 Convolution 4.1.2 Direct Form 4.1.3 Convolution 4.1.4 LTI Form 4.1.5 Matrix Form, 4.1.6 Flip-and-Slide Form 4.1.7 Transient and Steady-State Behavior 4.1.8 Convolution of Infinite Sequences,
	CMK5	z-Transforms: 5.1 Basic Properties 5.2 Region of Convergence 5.3 Causality and Stability 5.4 Frequency Spectrum 5.5 Inverse z-Transforms
	CMK6	Transfer Functions: 6.1 Equivalent Descriptions of Digital Filters 6.2 Transfer Functions, 6.3 Sinusoidal Response, 6.3.1 Steady-State Response 6.3.2 Transient Response 6.4 Pole/Zero Designs, 6.4.1 First-Order Filters 6.4.2 Parametric Resonators and Equalizers 6.4.3 Notch and Comb Filters 6.5 Deconvolution, Inverse Filters, and Stability
	CMK7	Digital Filter Realizations 7.1 Direct Form 7.2 Canonical Form

	7.3 Cascade Form 7.4 Cascade to Canonical 7.5 Hardware Realizations and Circular Buffers
CMK8	Ujian Tengah Semester (UTS) Tes tertulis untuk materi I s/d VII
CMK9	Signal Processing Applications: 1.Digital Waveform Generators, 2. Digital Audio Effects 3. Noise Reduction and Signal Enhancement Noise Reduction Filters Notch and Comb Filters
CMK10	1.DFT/FFT Algorithms: 2.DTFT at a SingleFrequency 3.DTFT over Frequency Range 4.DFT 5,Zero Padding,
CMK11	FIR Digital Filter Design: 1.Window Method 2.Ideal Filters, 532 3.Rectangular Window, 4.Hamming Window
CMK 12	Design: 1.Bilinear Transformation 2.First-Order Lowpass and Highpass Filters 3.Butterworth Filters
CMK 13	Desain filter : Menggunakan 1metoda Recursif filter

	2. Buterworth filter
CMK14	The Chebyshev and Butterworth Responses: Pembuatan filter chebisyhev dan butterworth lebih mudah lagi dengan ada konstanta 2 pole, 4 pole, 6 pole. Kemudian dimasukan kedalam konstanta recursive filter.
CMK 15	Dapat membuat filter untuk low pass , high pass, dan band pass menggunakan cara recursive filter.
CPMK 16	Ujian Akhir Semester (UAS) Tes tertulis untuk materi IX s/d XV
Sub-CMK1	Ruang Lingkup DSP ada di tahun 1960-an dan 1970-an ketika komputer digital pertama kali menjadi tersedia. Komputer mahal selama era ini, dan DSP hanya terbatas pada beberapa aplikasi kritis. Upaya perintisan dilakukan di empat bidang utama: radar & sonar, di mana keamanan nasional terancam; eksplorasi minyak, di mana uang dalam jumlah besar bisa dibuat; eksplorasi ruang angkasa, dimana data tidak tergantikan; dan pencitraan medis, di mana nyawa bisa diselamatkan. Revolusi komputer pada 1980-an dan 1990-an menyebabkan DSP meledak dengan yang baru aplikasi. Alih-alih dimotivasi oleh kebutuhan militer dan pemerintah, DSP lah tiba-tiba didorong oleh pasar komersial. Siapapun yang mengira mereka bisa membuat uang di bidang yang berkembang pesat tiba-tiba menjadi vendor DSP. DSP mencapai publik dalam produk-produk seperti: telepon seluler, pemutar CD, dan suara elektronik surat. Gambar 1-1 mengilustrasikan beberapa dari beragam aplikasi ini.

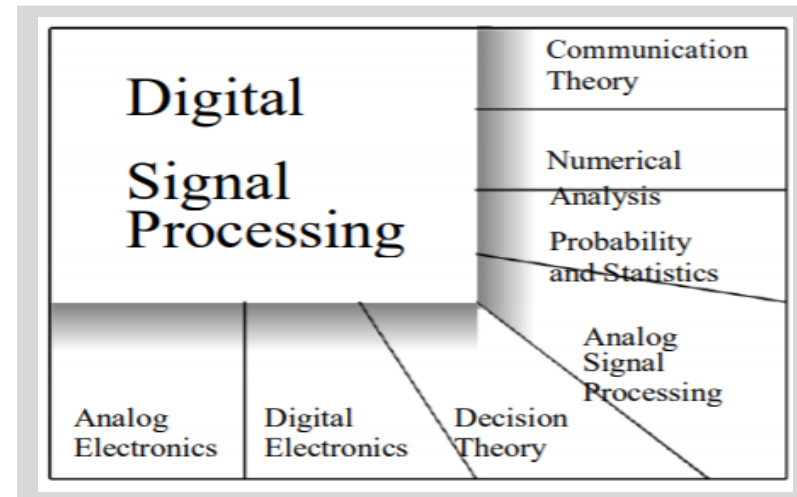
Revolusi teknologi ini terjadi dari atas ke bawah. Pada awal 1980-an, DSP diajarkan sebagai kursus tingkat pascasarjana di bidang teknik listrik. Satu dekade kemudian, DSP melakukannya menjadi bagian



Gambar 1-1

DSP telah merevolusi banyak bidang di bidang sains dan teknik, beberapa dari beragam aplikasi ini ditampilkan di sini

Disini digambarkan bidang-bidang dimana DSP telah menghasilkan Prubahan revolusioner. Saat anda menelusuri setiap aplikasi, perhatikan bahwa DSP sangat bermanfaat antar interdisipliner, mengendalikan pekerjaan teknis dibanyak bidang yang berdekatan. Seperti pada Gambar 1-2 dapat dilihat, batas antara DSP dan disiplin teknik lainnya terdefinisi dengan baik, tetapi agak kabur dan tumpang tindih. Jika anda ingin berspesialisasi dalam DSP, ini adalah area kesatuan yang juga perlu anda pelajari:



Gambar 1-2
Pemrosesan Sinyal Digital memiliki batas yang tidak jelas dan tumpang tindih dengan banyak lainnya bidang sains, teknik dan matematika.

Telekomunikasi

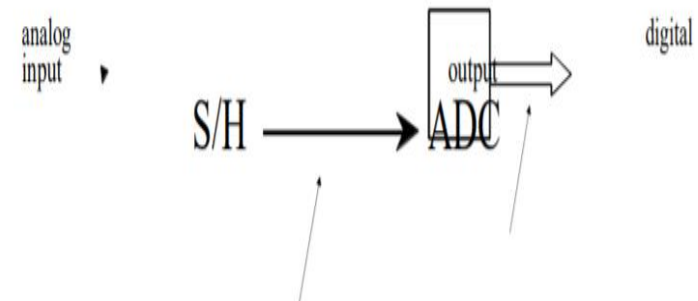
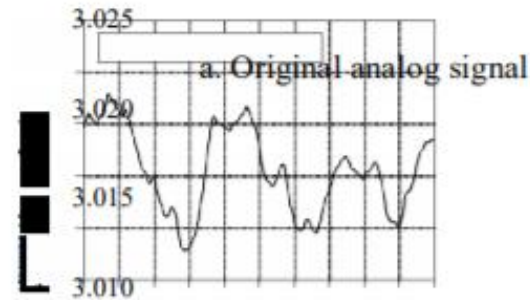
Telekomunikasi adalah transfer informasi dari satu lokasi ke lokasi lain. Ini mencakup banyak bentuk informasi: percakapan telepon, sinyal televisi, file komputer, dan jenis data lainnya. Untuk mentransfer informasi, Anda memerlukan saluran di antara dua lokasi. Ini mungkin sepasang kabel, sinyal radio, serat optik, dll. Perusahaan telekomunikasi menerima pembayaran untuk mentransfer pelanggan mereka informasi, sementara mereka harus membayar untuk membangun dan memelihara saluran tersebut. DSP telah merevolusi industri telekomunikasi di banyak era. Pembuatan dan deteksi nada sinyal, pergeseran pita frekuensi, pemfilteran ke lepaskan degungan saluran listrik, dll.

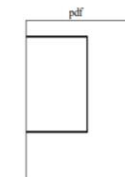
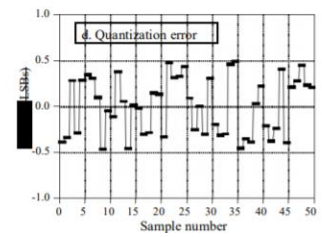
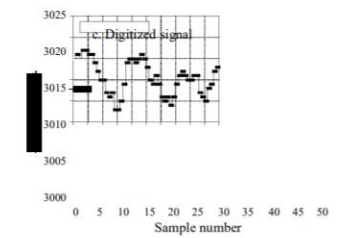
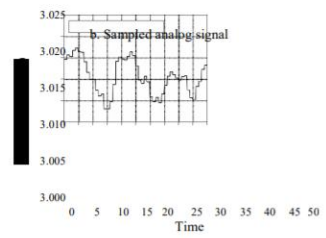
Multiplexing

Di beberapa pers tombol, jaringan switching memungkinkan salah satu dari ini untuk dihubungkan ke yang lain hanya dalam beberapa detik. Besarnya tugas ini sangat membingungkan! Sampai tahun 1960-an, ada hubungan antara dua telepon diperlukan melewati sinyal suara analog melalui mekanis sakelar dan

		<p>amplifire. Misalnya, standar telepon yang dikenal sebagai sistem T-carrier bisa secara bersamaan mengirimkan 24 sinyal suara. Setiap sinyal suara diambil sampelnya 8000 kali per kedua menggunakan konversi analog-ke-digital 8 bit companded (logarithmic compressed). Ini menghasilkan setiap sinyal suara yang direpresentasikan sebagai 64.000 bit / detik, dan semua 24 saluran yang terkandung dalam 1,544 megabit / detik. Sinyal ini dapat ditransmisikan sekitar 6000 kaki menggunakan saluran telepon biasa dari kawat tembaga 22 gauge, sebuah interkoneksi yang khas jarak. Keuntungan finansial dari transmisi digital sangat besar.</p> <p>Kompresi Lusinan algoritma DSP telah dikembangkan untuk dikonversi sinyal suara digital menjadi aliran data yang membutuhkan lebih sedikit bit per detik, ini disebut data algoritma kompresi. Algoritme ini bervariasi dalam jumlah kompresi yang dicapai dan kualitas suara yang dihasilkan. Secara umum, mengurangi kecepatan data dari 64 kilobit / detik menjadi 32 kilobit / detik tidak menghasilkan kualitas suara yang menurun. Saat dikompresi ke kecepatan data 8 kilobit / detik, suara sangat terpengaruh, tetapi masih dapat digunakan untuk telepon jarak jauh jaringan. Kompresi yang dapat dicapai tertinggi adalah sekitar 2 kilobit / detik, menghasilkan suara yang sangat terdistorsi, tetapi dapat digunakan untuk beberapa aplikasi seperti militer dan komunikasi bawah laut.</p>
	<p>Sub-CPMK2</p>	<p>ADC & DAC Sebagian besar sinyal yang ditemui secara langsung dalam sains dan teknik bersifat kontinu: intensitas cahaya yang berubah seiring jarak; tegangan yang bervariasi dari waktu ke waktu; laju reaksi kimia yang bergantung pada suhu, dll. Konversi Analog-ke-Digital (ADC) dan Konversi Digital-ke-Analog (DAC) adalah proses yang memungkinkan</p>

komputer digital berinteraksi dengan sinyal sehari-hari ini. Digital informasi berbeda dari rekan berkelanjutannya dalam dua hal penting: diambil sampelnya, dan itu dikuantisasi. Keduanya membatasi seberapa banyak informasi yang dapat dikandung sinyal digital. Ini bab tentang manajemen informasi: memahami informasi apa yang perlu Anda simpan, dan informasi apa yang Anda mampu untuk kehilangan. Pada gilirannya, ini menentukan pemilihan pengambilan sampel frekuensi, jumlah bit, dan jenis pemfilteran analog yang diperlukan untuk konversi antar analog dan dunia digital.



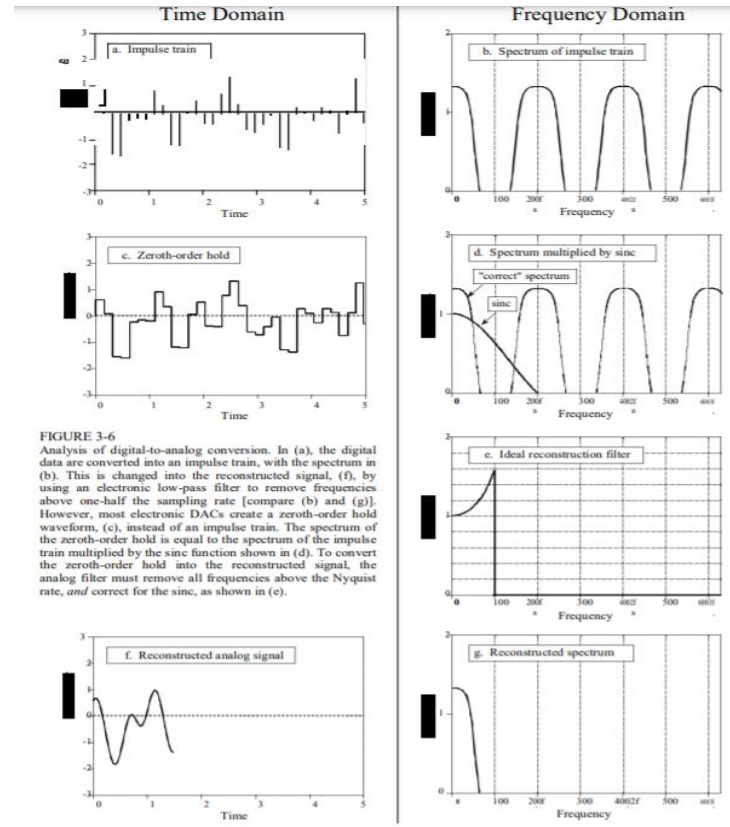


Gambar 2-1 Konversi Digital ke Analog

Bentuk gelombang yang menggambarkan proses digitalisasi. Itu konversi dipecah menjadi duatahap untuk memungkinkan efek pengambilan sampel dipisahkan dari efek kuantisasi. Tahap pertama adalah sampel-and-hold (S/H), dimana satu-satunya informasi yang dipertahankan adalah nilai sesaat dari sinyal saat periodic pengambilan sampel berlangsung. Pada tahap kedua, ADC mengubah tegangan ke bilangan bulat terdekat. Ini menghasilkan setiap sampel dalam sinyal digital memiliki plus minus $\frac{1}{2}$ LSB. Seperti yang ditunjukkan pada (d). sebagai akibatnya, kuantisasi biasayanta dapat dimodelkan sebagai hanya menambahkan noise kie sinyal.

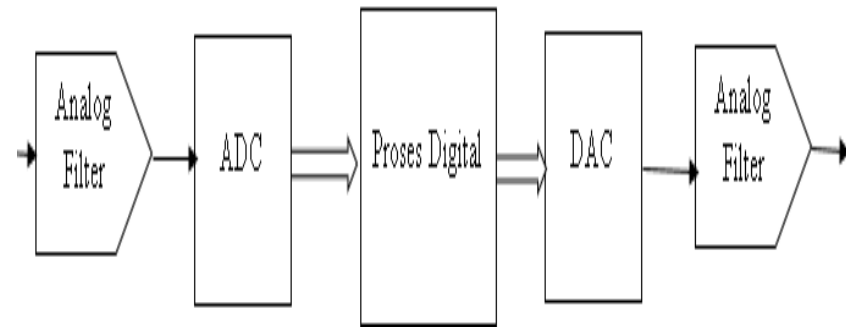
Konversi Digital ke Analog

Konversi digital ke analog adalah dengan menarik sampel dari memori dan mengubahnya menjadi kereta implus. Seperti gbr dibawah ini:



Gambar 2-2 Analog To Digital

Filter Analog untuk Konversi Data



Gambar 2.3 Filter Analog to Digital dan Digital to Analog

Filter elektronik analog yang digunakan untuk memenuhi teorema pengambilan sampel. Filter elektronik ditempatkan sebelum ADC disebut filter antialias. Ini digunakan untuk menghapus komponen frekuensi di atas setengah dari laju pengambilan sampel yang seharusnya alias selama pengambilan sampel. Filter elektronik yang ditempatkan setelah DAC disebut filter rekonstruksi. Itu juga menghilangkan

frekuensi di atas tingkat Nyquist, dan mungkin termasuk koreksi untuk penahanan orde-nol. sinyal input diproses dengan filter low-pass elektronik untuk menghilangkan semua frekuensi di atas frekuensi Nyquist (setengah laju pengambilan sampel). Ini adalah dilakukan untuk

mencegah aliasing selama pengambilan sampel, dan juga disebut sebagai filter antialias. Di sisi lain, sinyal digital dilewatkan melalui konverter digital ke analog dan filter low-pass lainnya diatur ke frekuensi Nyquist. Ini filter keluaran disebut filter rekonstruksi, dan mungkin termasuk filter sebelumnya menjelaskan peningkatan frekuensi penahan urutan ke-nol

TABEL 2-4 Karakteristik dari tiga filter klasik

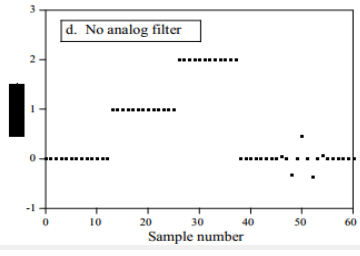
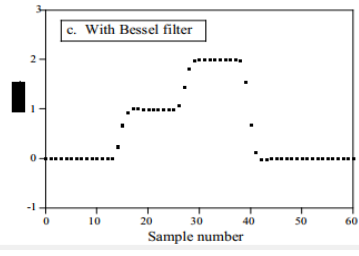
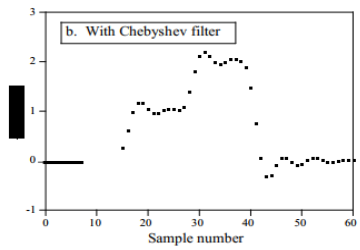
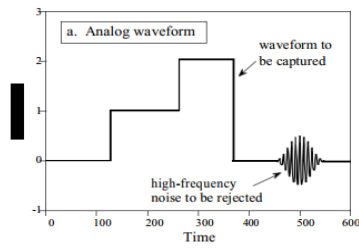
	Voltage gain at DC	Step Response			Frequency Response		
		Overshoot	Time to settle to 1%	Time to settle to 0.1%	Ripple in passband	Frequency for x100 attenuation	Frequency for x1000 attenuation
Bessel							
2 pole	1.27	0.4%	0.60	1.12	0%	12.74	40.4
4 pole	1.91	0.9%	0.66	1.20	0%	4.74	8.45
6 pole	2.87	0.7%	0.74	1.18	0%	3.65	5.43
8 pole	4.32	0.4%	0.80	1.16	0%	3.35	4.53
Butterworth							
2 pole	1.59	4.3%	1.06	1.66	0%	10.0	31.6
4 pole	2.58	10.9%	1.68	2.74	0%	3.17	5.62
6 pole	4.21	14.3%	2.74	3.92	0%	2.16	3.17
8 pole	6.84	16.4%	3.50	5.12	0%	1.78	2.38
Chebyshev							
2 pole	1.84	10.8%	1.10	1.62	6%	12.33	38.9
4 pole	4.21	18.2%	3.04	5.42	6%	2.59	4.47
6 pole	10.71	21.3%	5.86	10.4	6%	1.63	2.26
8 pole	28.58	23.0%	8.34	16.4	6%	1.34	1.66

		<p>Karakteristik dari tiga filter klasik. Filter Bessel memberikan respons langkah terbaik, menjadikannya pilihan untuk sinyal yang dikodekan domain waktu. Filter Chebyshev dan Butterworth digunakan untuk menghilangkan frekuensi di stopband, menjadikannya ideal untuk sinyal yang dikodekan dalam domain frekuensi. Nilai dalam tabel ini adalah dalam satuan detik dan hertz, untuk frekuensi cutoff satu hertz. Sementara ada banyak cara agar informasi dikodekan hanya dalam bentuk gelombang analog dua metode umum, pengkodean domain waktu, dan domain frekuensi pengkodean. Perbedaan antara keduanya sangat penting dalam DSP, dan akan menjadi tema berulang di seluruh buku ini.</p>
--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Dalam pengkodean domain frekuensi, informasi tersebut terkandung dalam gelombang sinusoidal yang bergabung untuk membentuk sinyal. Sinyal audio adalah contoh yang bagus untuk ini. Ketika seseorang mendengar pidato atau musik, suara yang dirasakan bergantung pada frekuensi yang ada, dan bukan pada bentuk gelombang tertentu. Ini bisa jadi ditunjukkan dengan melewati sinyal audio melalui rangkaian yang mengubah fase berbagai sinusoid, tetapi mempertahankan frekuensi dan amplitudonya. Sinyal yang dihasilkan terlihat sangat berbeda pada osiloskop, tetapi terdengar identik.

Konversi Data Multirate

Konversi data adalah contoh yang bagus untuk ini. Pertimbangkan desainnya perekam suara digital, sistem yang akan mendigitalkan sinyal suara, menyimpan file data dalam bentuk digital, dan kemudian merekonstruksi sinyal untuk diputar ulang. Untuk membuat ulang ucapan yang dapat dimengerti, sistem harus menangkap frekuensi antara sekitar 100 dan 3000 hertz. Namun sinyal analog juga dihasilkan oleh mikrofon mengandung frekuensi yang jauh lebih tinggi, katakanlah 40 kHz. Pendekatan kekerasan adalah melewati sinyal analog melalui filter Chebyshev delapan kutub low-pass pada 3 kHz, dan kemudian ambil sampel pada 8 kHz. Di sisi lain, DAC merekonstruksi analog sinyal pada 8 kHz dengan penahanan orde nol. Filter Chebyshev lain pada 3 kHz adalah digunakan untuk menghasilkan sinyal suara akhir

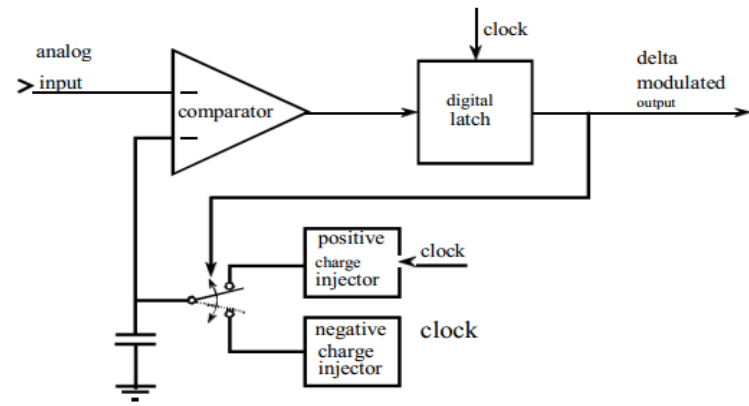


GAMBAR 2.5 Konversi Data Multirate

Tiga opsi filter antialias untuk sinyal yang dikodekan domain waktu. Tujuannya adalah untuk menghilangkan frekuensi tinggi (alias itu selama pengambilan sampel), sekaligus mempertahankan ketajaman tepi (yang membawa informasi). Gambar (a) menunjukkan contoh sinyal analog yang mengandung kedua sisi tajam dan ledakan kebisingan frekuensi tinggi Gambar (b) menunjukkan digitalisasi tersebut sinyal menggunakan filter Chebyshev. Sementara frekuensi tinggi telah dihilangkan secara efektif, ujung-ujungnya terlalu banyak terdistorsi. Ini biasanya

solusi yang buruk. Filter Bessel, yang ditunjukkan pada (c), memberikan perataan tepi yang lembut menghilangkan frekuensi tinggi. Gambar (d) menunjukkan sinyal digital menggunakan filter tanpa antialias. Dalam hal ini, tepinya telah mempertahankan ketajaman sempurna; Namun, ledakan frekuensi tinggi telah alias menjadi beberapa sampel yang tidak berarti. Ada banyak manfaat berguna dalam pengambilan sampel yang lebih cepat daripada analisis langsung ini. Untuk Misalnya, bayangkan mendesain ulang perekam suara digital menggunakan sampling 64 kHz menilai. Filter antialias sekarang memiliki tugas yang lebih mudah: meneruskan semua frekuensi di bawah 32 kHz, sambil menolak semua frekuensi di atas 32 kHz. Penyederhanaan serupa terjadi untuk filter rekonstruksi. Singkatnya, tingkat pengambilan sampel yang lebih tinggi memungkinkan delapan filter kutub harus diganti

		<p>dengan jaringan resistor-kapasitor (RC) sederhana. Masalahnya, sistem digital kini dibanjiri dengan data-data dari yang lebih tinggi tingkat pengambilan sampel.</p> <p>Konversi Data Bit Tunggal</p> <p>Teknik yang populer di bidang telekomunikasi dan reproduksi musik dengan ketelitian tinggi adalah satu bit ADC dan DAC. Ini adalah teknik multirate dimana lebih tinggi sampling rate diperdagangkan untuk jumlah bit yang lebih rendah. Secara ekstrim, hanya satu bit diperlukan untuk setiap sampel. Meskipun ada banyak sirkuit yang berbeda konfigurasi, sebagian besar didasarkan pada penggunaan modulasi delta</p>
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



		<p>Gambar 2.6 Diagram blok dari rangkaian modulasi delta</p> <p>Diagram blok dari rangkaian modulasi delta. Tegangan input dibandingkan dengan tegangan disimpan pada kapasitor, menghasilkan nol digital atau satu yang diterapkan ke input memalangi. Output dari latch diperbarui dalam sinkronisasi dengan jam, dan digunakan dalam file loop umpan balik untuk menyebabkan tegangan kapasitor melacak tegangan input. Sebuah loop umpan balik dibentuk dengan mengambil keluaran digital dan menggunakannya untuk menggerakkan saklar elektronik. Jika outputnya digital, sakelar menghubungkan kapasitor ke injektor muatan positif. Ini adalah istilah yang sangat longgar untuk sirkuit yang meningkat tegangan pada kapasitor dengan jumlah yang tetap, katakanlah 1 milivolt per siklus clock. Ini mungkin tidak lebih dari resistor yang terhubung ke tegangan positif yang besar. Jika outputnya adalah nol digital, sakelar terhubung ke injektor muatan negatif. Ini menurunkan tegangan pada kapasitor dengan jumlah tetap yang sama.</p>
	<p>Sub-CPMK3</p>	<p>SISTEM LTI (Linear Time Invariant)</p> <p>Sistem ini adalah invarian waktu linier (LTI) jika memenuhi properti linieritas dan varians waktu Bab ini akan mempelajari sistem LTI hampir secara eksklusif, karena paling mudah sistem untuk bekerja, dan ideal untuk dianalisis dan dirancang.</p> <p>Linier</p> <p>Ada 3 persyaratan untuk linieritas. Sebuah fungsi harus memenuhi ketiganya untuk disebut "linier".</p> <ol style="list-style-type: none"> 1. Aditivitas: Input dari $x_3(t) = x_1(t) + x_2(t)$ menghasilkan

		<p>output $y_3(t) = y_1(t) + y_2(t)$. Homogenitas: Input dari x_1 menghasilkan output y_1 Jika $x(t) = 0, y(t) = 0$.</p> <p>2. "Linear" dalam pengertian ini bukanlah kata yang sama seperti yang digunakan dalam aljabar konvensional atau geometri. Secara khusus, linieritas dalam aplikasi sinyal tidak ada hubungannya dengan garis lurus. Berikut ini contoh kecilnya:</p> $y(t) = x(t) + 5$ <p>Fungsi ini tidak linier, karena ketika $x(t) = 0, y(t) = 5$ (gagal memenuhi persyaratan 3). Ini mungkin mengejutkan orang, karena persamaan ini adalah persamaan untuk garis lurus.</p> <p>Kita dapat menggabungkan tiga persyaratan menjadi satu persamaan: Dalam sistem linier, sebuah masukan dari $a_1x_1(t) + a_2x_2(t)$ menghasilkan output $a_1y_1(t) + a_2y_2(t)$.</p> <p>Aditif</p> <p>Suatu sistem dikatakan aditif jika sejumlah masukan menghasilkan sejumlah keluaran. Untuk menguji aditif, kita perlu membuat dua input yang berubah – ubah, $x_1(t)$ dan $x_2(t)$. Kami kemudian menggunakan masukan ini untuk menghasilkan dua keluaran masing-masing:</p> $y_1(t) = f(x_1(t))$ $y_2(t) = f(x_2(t))$ <p>Dimana</p> $x_1(t) = cx_2(t)$ <p>Di mana c adalah konstanta arbitrer. Jika ini kasusnya maka sistem homogen jika untuk sembarang c.</p> $y_1(t) = cy_2(t)$ <p>Varians Waktu</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Jika sinyal masukan $x(t)$ menghasilkan keluaran $y(t)$ maka setiap kali masukan digeser, $x(t + \delta)$, hasilnya dalam output bergeser waktu $y(t + \delta)$. Properti ini dapat dipenuhi jika fungsi transfer sistem bukan merupakan fungsi waktu kecuali diekspresikan oleh input dan output.

Selain linier, atau time-invariant, ada sejumlah properti lain yang kita bisa mengidentifikasi dalam suatu fungsi:

Memori

Suatu sistem dikatakan memiliki memori jika keluaran dari sistem bergantung pada masukan sebelumnya (atau masukan masa depan) ke sistem. Suatu sistem disebut tanpa memori jika keluarannya hanya tergantung pada masukan saat ini. Sistem tanpa memori lebih mudah digunakan, tetapi sistem dengan memori lebih umum dalam aplikasi pemrosesan sinyal digital.

Kasualitas

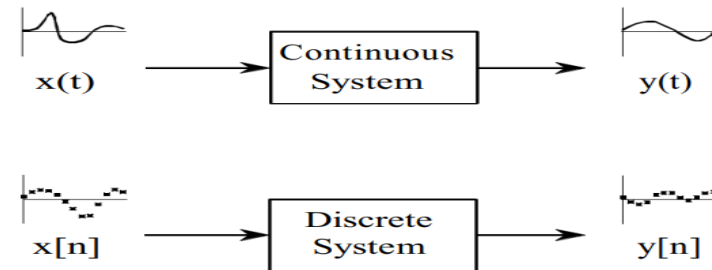
Kausalitas adalah properti yang sangat mirip dengan memori. Suatu sistem disebut kausal jika hanya tergantung pada masukan masa lalu atau saat ini. Suatu sistem disebut non kausal jika keluaran dari sistem tersebut tergantung pada masukan masa depan.

Stabilitas

Stabilitas adalah konsep yang sangat penting dalam sistem, tetapi juga salah satu fungsi yang paling sulit properti untuk membuktikan. Ada beberapa kriteria berbeda untuk stabilitas sistem, tetapi yang paling banyak persyaratan umum adalah bahwa sistem harus menghasilkan keluaran yang terbatas ketika mengalami sebuah masukan yang terbatas. Misalnya, jika kita

		<p>menerapkan 5 volt ke terminal input dari rangkaian tertentu, kita akan menyukainya jika keluaran rangkaian tidak mendekati tak terhingga, dan rangkaian itu sendiri tidak meleleh atau meledak. Jenis stabilitas ini sering disebut dengan stabilitas "Input Terikat, Output Terikat", atau BIBO.</p> <p>Operator Linear</p> <p>Operator matematika yang memenuhi properti linieritas dikenal sebagai operator linier. Berikut beberapa operator linier yang umum:</p> <ol style="list-style-type: none"> 1. Turunan 2. Integral 3. Transformasi Fourier
	<p>Sub-CPMK4</p>	<p>Sistem Linier</p> <p>Kebanyakan teknik DSP didasarkan pada strategi divide-and-conquer yang disebut superposisi. Sinyalnya sedang diproses dipecah menjadi komponen sederhana, setiap komponen diproses secara individual, dan hasilnya dipertemukan kembali. Pendekatan ini memiliki kekuatan luar biasa untuk memecahkan satu masalah rumit menjadi banyak yang mudah. Superposisi hanya dapat digunakan dengan sistem linier, istilah yang berarti bahwa aturan matematika tertentu berlaku. Untungnya, sebagian besar aplikasi ditemui dalam sains dan teknik termasuk dalam kategori ini. Bab ini menyajikan fondasi DSP: apa artinya sistem menjadi linier, berbagai cara untuk memecah sinyal komponen yang lebih sederhana, dan bagaimana superposisi menyediakan berbagai teknik pemrosesan sinyal.</p> <p>Sinyal dan Sistem</p>

Sinyal adalah deskripsi tentang bagaimana satu parameter bervariasi dengan parameter lainnya. Misalnya tegangan berubah seiring waktu di sirkuit elektronik, atau kecerahan bervariasi dengan jarak pada gambar. Sebuah sistem adalah setiap proses yang menghasilkan sinyal keluaran sebagai respons terhadap sinyal masukan. Ini diilustrasikan oleh diagram blok pada Gambar 4.1. Input dan output sistem kontinu sinyal kontinu, seperti di analog elektronik. Input dan output sistem diskrit sinyal diskrit, seperti program komputer itu memanipulasi nilai yang disimpan dalam array



Gambar 4.1 Sistem Linear Continue dan Sistem Discrete

Terminologi untuk sinyal dan sistem. Sistem adalah proses apa pun yang menghasilkan sinyal keluaran menanggapi sinyal input. Sinyal kontinu biasanya diwakili dengan tanda kurung, sedangkan sinyal diskrit menggunakan tanda kurung. Semua sinyal menggunakan huruf kecil, menggunakan huruf besar untuk domain frekuensi (disajikan dalam bab-bab selanjutnya). Kecuali ada nama yang lebih baik yang tersedia, masukan sinyal disebut: $x(t)$ atau $x[n]$, sedangkan keluarannya disebut: $y(t)$ atau $y[n]$. Sinyal dan sistem sering kali didiskusikan tanpa mengetahui

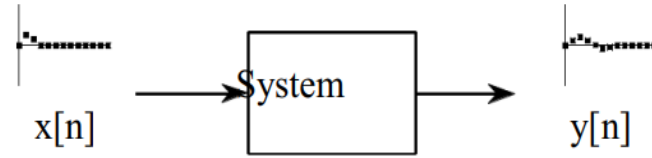
parameter pasti yang diwakili. Ini sama dengan menggunakan x dan y dalam aljabar, tanpa memberikan arti fisik pada variabel. Ini membawa aturan keempat untuk sinyal penamaan. Jika nama yang lebih deskriptif tidak tersedia, masukan sinyal ke sistem diskrit biasanya disebut: $x[n]$, dan sinyal keluaran: $y[n]$. Untuk sistem kontinu, sinyal: $x(t)$ dan $y(t)$ digunakan. Dalam hal teori system, masalahnya adalah menemukan system itu mengubah sinyal yang ditransmisikan menjadi sinyal yang diterima

Persyaratan untuk Linearitas:

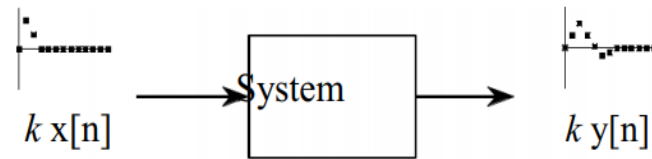
Suatu sistem disebut linier jika memiliki dua sifat matematis: homogenitas (homogeneity) dan aditif. Jika Anda dapat menunjukkan bahwa suatu sistem memiliki kedua properti tersebut, maka Anda telah membuktikan bahwa sistem tersebut benar linier. Demikian juga, jika Anda dapat menunjukkan bahwa suatu sistem tidak memiliki satu atau kedua properti, Anda telah membuktikannya bahwa itu tidak linier. Properti ketiga, shift invariance, bukanlah persyaratan yang ketat untuk linieritas, tetapi ini adalah sebuah properti wajib untuk sebagian besar teknik DSP. Ketika Anda melihat istilah sistem linier yang digunakan di DSP, Anda harus menganggap itu termasuk invariansi shift kecuali Anda memiliki alasan untuk percaya sebaliknya.

Dalam istilah matematika, jika sinyal input $x[n]$ menghasilkan sinyal keluaran $y[n]$, masukan $kx[n]$ menghasilkan keluaran $ky[n]$, untuk setiap sinyal masukan dan konstan, k .

IF



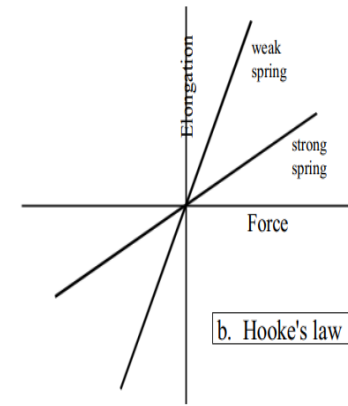
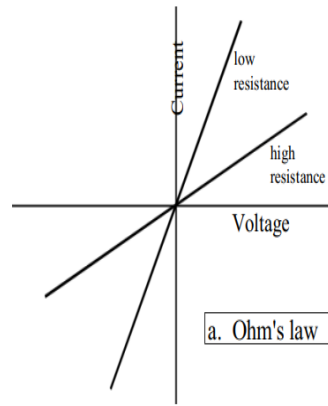
THEN



Gambar 4.2 Sistem Diskrete

Definisi homogenitas. Suatu sistem dikatakan homogen jika amplitudo berubah dalam hasil masukan dalam perubahan amplitudo yang identik pada keluaran. Artinya, jika $x[n]$ menghasilkan $y[n]$, kemudian $kx[n]$ menghasilkan $ky[n]$, untuk sinyal apa pun, $x[n]$, dan konstanta apa pun, k .

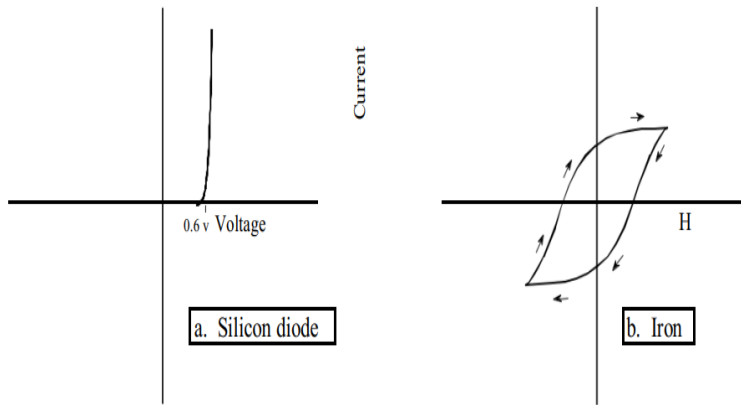
Definisi homogenitas. Suatu sistem dikatakan homogen jika amplitudo berubah dalam hasil masukan dalam perubahan amplitudo yang identik pada keluaran. Artinya, jika $x[n]$ menghasilkan $y[n]$, kemudian $kx[n]$ menghasilkan $ky[n]$, untuk sinyal apa pun, $x[n]$, dan konstanta apa pun, k .



Gambar 4.3 Dua contoh linieritas statis

Dua contoh linieritas statis. Dalam (a), hukum Ohm: arus yang melalui resistor sama dengan tegangan melintasi resistor dibagi dengan resistansi. Dalam (b), hukum Hooke: Perpanjangan pegas sama dengan gaya yang diterapkan dikalikan dengan koefisien kekakuan pegas.

Dalam sistem ini tidak masalah jika sinyal input statis atau berubah. Ini adalah disebut sistem tanpa memori, karena keluarannya hanya bergantung pada keadaan masukan saat ini, dan bukan dalam sejarahnya. Misalnya, arus sesaat dalam resistor hanya bergantung pada tegangan sesaat di atasnya, dan bukan pada bagaimana sinyal menjadi nilainya. Jika sebuah sistem memiliki linieritas statis, dan tanpa memori, maka sistem harus linier. Ini memberikan cara penting untuk memahami (dan membuktikan) linieritas sistem sederhana ini

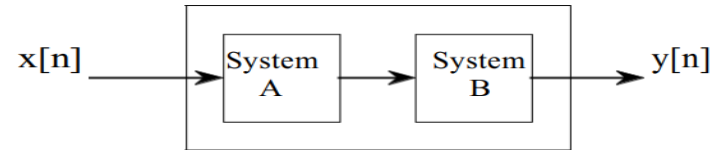


Gambar 4.4 Dua contoh nonlinier DC

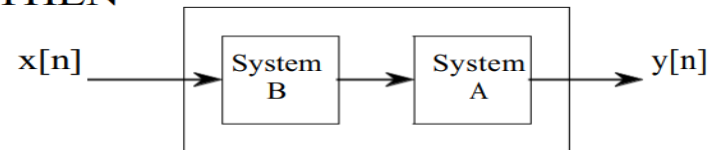
Dua contoh nonlinier DC. Dalam (a), dioda silikon memiliki

		<p>hubungan eksponensial antara tegangan dan arus. Dalam (b), hubungan antara intensitas magnet, H, dan kerapatan fluks, B, pada besi tergantung pada sejarah sampel, suatu perilaku yang disebut histeresis.</p> <p>Karakteristik penting dari system linier adalah bagaimana mereka berperilaku dengan sinusoid, sifat yang akan kita lakukan memanggil kesetiaan sinusoidal. Jika masukan ke system linier adalah gelombang sinusoidal, keluarannya juga akan menjadi gelombang sinusoidal dan pada frekuensi yang sama persis dengan input. Sinusoidal adalah satu-satunya bentuk gelombang itu memiliki property ini. Misalnya, tidak ada alasan untuk mengharapkan gelombang persegi memasuki linier system akan menghasilkan gelombang persegi pada keluarannya. Meskipun sinusoidal pada masukan menjamin sinusoidal pada output, keduanya mungkin berbeda dalam amplitude dan fase</p> <p>Properti Khusus Linearitas</p> <p>Properti komutatif menyatakan bahwa urutan sistem dalam kaskade dapat diatur ulang tanpanya mempengaruhi karakteristik kombinasi keseluruhan. Anda mungkin telah menggunakan prinsip ini dalam sirkuit elektronik. Misalnya, bayangkan rangkaian yang terdiri dari dua tahap, satu untuk amplifikasi, dan satu untuk penyaringan</p>
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

IF



THEN

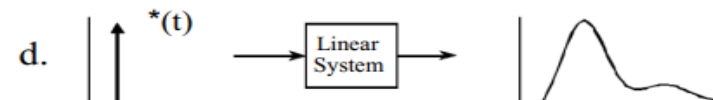
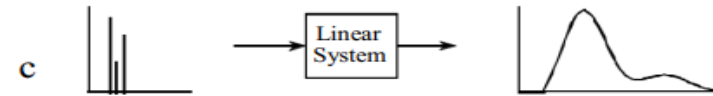
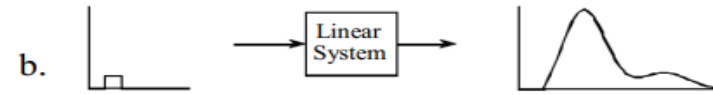
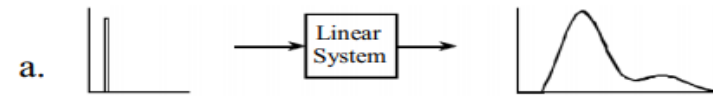


Gambar 4.5 Properti Komutatif

Properti komutatif untuk linier sistem. Ketika dua atau lebih sistem linier disusun dalam kaskade, urutan sistem tidak mempengaruhi karakteristik dari kombinasi keseluruhan.

Untuk memahami apa arti linearitas untuk sistem dengan banyak input dan / atau output, pertimbangkan eksperimen pikiran berikut. Mulailah dengan menempatkan sinyal pada satu input - input lainnya ditahan di nol. Ini akan menyebabkan beberapa keluaran merespons beberapa pola sinyal. Selanjutnya, ulangi prosedur dengan menempatkan sinyal lain pada masukan yang berbeda. Sama seperti sebelumnya, pertahankan semua masukan lainnya di nol. Masukan kedua ini sinyal akan menghasilkan pola sinyal lain yang muncul pada beberapa keluaran. Menyelesaikan percobaan, tempatkan kedua sinyal pada inputnya masing-masing secara bersamaan. Sinyalnya yang muncul pada keluaran hanyalah superposisi (jumlah) dari sinyal keluaran diproduksi ketika sinyal input diterapkan secara terpisah

	Sub-CPMK5	<p>Pemrosesan sinyal kontinu adalah bidang paralel ke DSP, dan sebagian besar tekniknya adalah hampir identik. Misalnya, DSP dan pemrosesan sinyal berkelanjutan didasarkan pada linearitas, dekomposisi, konvolusi dan analisis Fourier. Sejak sinyal kontinu tidak dapat langsung direpresentasikan dalam komputer digital, jangan berharap menemukan computer program dalam bab ini. Pemrosesan sinyal berkelanjutan didasarkan pada matematika; sinyal direpresentasikan sebagai persamaan, dan sistem mengubah satu persamaan menjadi persamaan lainnya. Sama seperti komputer digital adalah alat utama yang digunakan dalam DSP, kalkulus adalah alat utama yang digunakan pemrosesan sinyal terus menerus. Teknik-teknik ini telah digunakan selama berabad-abad, jauh sebelumnya komputer dikembangkan</p> <p>Fungsi Delta</p> <p>Karena fungsi delta didefinisikan sangat sempit dan memiliki luas tetap, maka amplitudo tersirat tidak terbatas. Jangan biarkan ini mengganggu Anda; itu sama sekali tidak penting. Karena amplitudo adalah bagian dari bentuk impuls, Anda tidak akan pernah menemukan suatu masalah di mana amplitudo membuat perbedaan, tak terbatas atau tidak. Fungsi delta adalah sebuah konstruksi matematika, bukan sinyal dunia nyata. Sinyal di dunia nyata yang bertindak sebagai delta fungsi akan selalu memiliki durasi dan amplitudo yang terbatas</p> <p>Respons sistem linear terhadap fungsi delta impuls. yaitu $h(t)$.</p>
--	------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



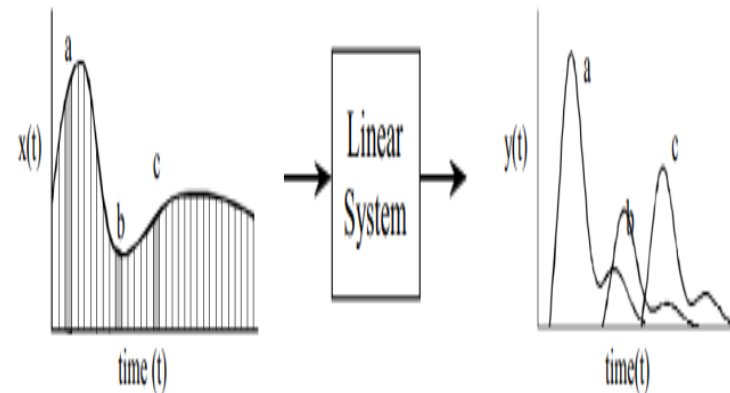
Gambar 5.1 Respons sistem Linear sistem terhadap Impuls delta.

Fungsi delta kontinu. Jika masukan ke sistem linier singkat dibandingkan dengan yang dihasilkan keluaran, bentuk keluaran hanya bergantung pada karakteristik sistem, dan bukan bentuknya dari masukan. Sinyal input pendek seperti itu disebut impuls. Gambar a, b & c mengilustrasikan contoh masukan sinyal yang merupakan impuls untuk sistem khusus ini. Istilah fungsi delta digunakan untuk menggambarkan suatu impuls yang dinormalisasi, yaitu impuls yang terjadi pada $t = 0$ dan memiliki luas satu. Simbol matematika untuk fungsi delta ditunjukkan pada (d), panah vertikal dan $\delta(t)$.

Konvolusi

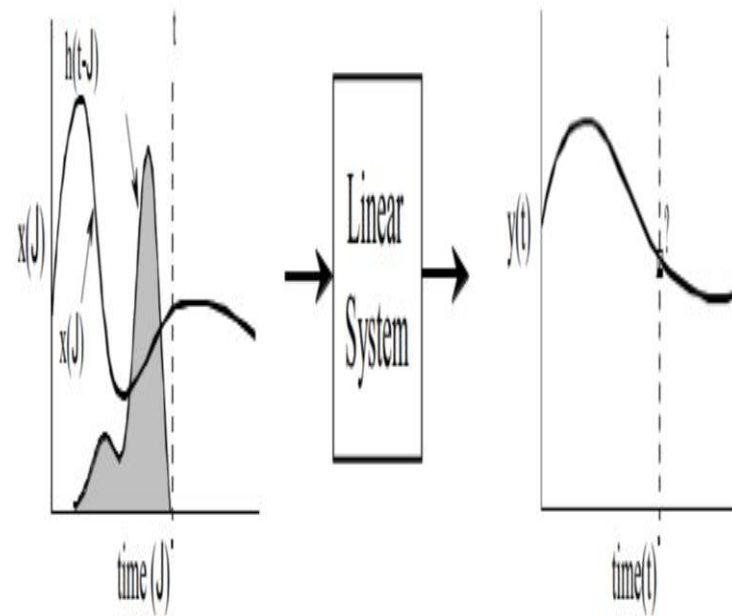
Gambar 5.2 menunjukkan bagaimana konvolusi dilihat dari sisi masukan. Sinyal masukan, $x(t)$, dilewatkan melalui sistem yang ditandai dengan respons impuls, $h(t)$, untuk menghasilkan sinyal keluaran, $y(t)$. Ini dapat ditulis dalam persamaan matematika yang sudah dikenal, $y(t) = x(t) * h(t)$. Sinyal masukan dibagi menjadi kolom sempit, masing-masing cukup pendek untuk bertindak sebagai impuls ke sistem. Dengan kata lain, sinyal input diuraikan menjadi tak terbatas jumlah fungsi delta yang diskalakan dan bergeser. Masing-masing impuls ini menghasilkan skala dan versi

bergeser dari respons impuls dalam sinyal keluaran. Sinyal keluaran akhir adalah kemudian sama dengan efek gabungan, yaitu jumlah semua respons individu. Agar skema ini berfungsi, lebar kolom harus jauh lebih pendek daripada respon sistem. Tentu saja, ahli matematika mengambil ini secara ekstrim dengan membuat segmen masukan sangat sempit, mengubah situasi menjadi masalah kalkulus. Dengan cara ini, sudut pandang masukan menjelaskan bagaimana satu titik (atau wilayah sempit) di sinyal input mempengaruhi porsi sinyal output yang lebih besar



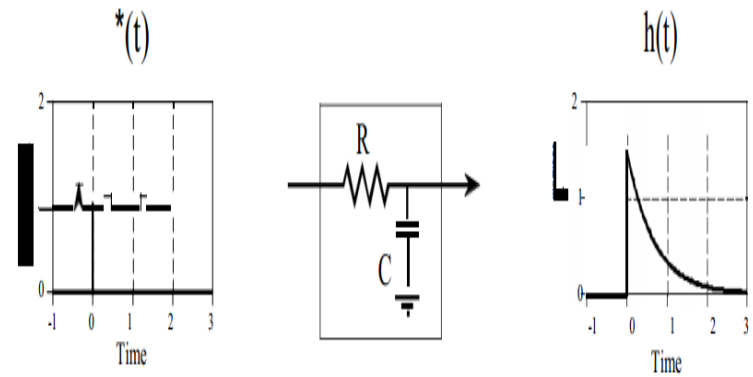
Gambar 5.3 Kovolusi pada Sistem linear

Konvolusi dilihat dari sisi masukan. Sinyal input, $x(t)$, dibagi menjadi segmen-segmen sempit, masing-masing bertindak sebagai dorongan untuk sistem. Sinyal keluaran, $y(t)$, adalah jumlah dari skala yang dihasilkan dan respons impuls yang bergeser. Ilustrasi ini menunjukkan bagaimana tiga titik dalam sinyal input berkontribusi pada sinyal keluaran



Gambar 5.4 Konvolusi dilihat dari sisi keluaran

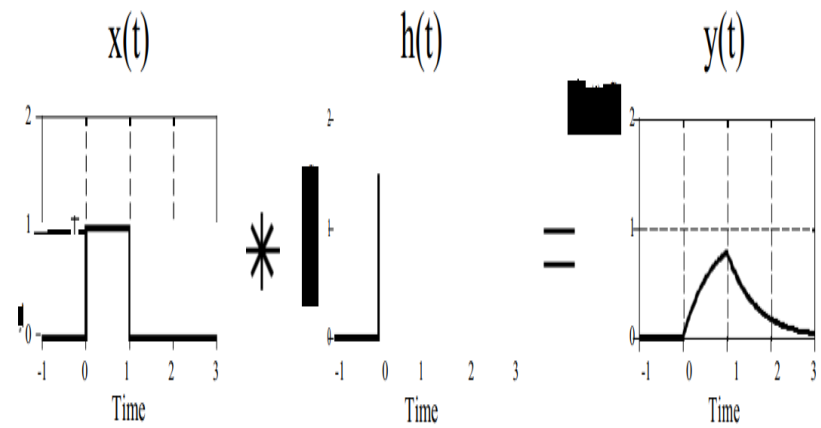
Konvolusi dilihat dari sisi keluaran. Setiap nilai dalam sinyal keluaran dipengaruhi oleh banyak poin dari sinyal input. Pada gambar ini, sinyal keluaran pada waktu t sedang dihitung. Masukan sinyal, $x(t)$, diberi bobot (dikalikan) dengan respon impuls yang dibalik dan digeser, diberikan oleh $h(t & \tau)$. Mengintegrasikan sinyal masukan berbobot menghasilkan nilai titik keluaran, $y(t)$.



Gambar 5.5

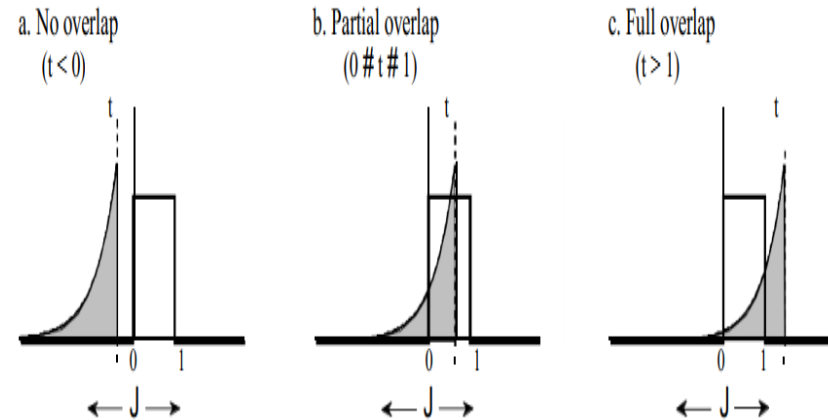
Contoh sistem linier kontinu. Sirkuit elektronik ini adalah filter low-pass yang terdiri dari satu resistor dan kapasitor. Respons

		<p style="text-align: center;">impuls sistem ini adalah eksponensial satu sisi</p> <p>respon impuls sistem ini dipecah menjadi dua bagian, masing-masing diwakili oleh persamaan:</p> $h(t) = 0 \text{ untuk } t < 0$ $h(t) = e^{-t/RC} \text{ untuk } t \geq 0$ <p>di mana $1/RC$ (R dalam ohm, C dalam farad, dan t dalam detik). Sama seperti pada diskrit kasus, respons impuls berkelanjutan berisi informasi lengkap tentang sistem, yaitu, bagaimana ia akan bereaksi terhadap semua sinyal yang mungkin. Untuk melanjutkan contoh ini lebih jauh, Gambar 13-5 menunjukkan pulsa persegi memasuki sistem, yang secara matematis dinyatakan dengan:</p> $x(t) = 1 \text{ untuk } 0 \leq t \leq 1$ $x(t) = 0 \text{ jika tidak}$ <p>Karena sinyal input dan respons impuls sepenuhnya dikenal sebagai ekspresi matematika, sinyal keluaran, $y(t)$, dapat dihitung dengan mengevaluasi konvolusi integral dari Persamaan. 5.1. Ini diperumit oleh fakta bahwa kedua sinyal tersebut ditentukan oleh bukan satu wilayah.</p>
--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Gambar 5.6

Contoh konvolusi terus menerus. Gambar ini mengilustrasikan pulsa persegi yang memasuki filter low-pass RC (Gambar. 5.5). Pulsa persegi berbelit-belit dengan respons impuls sistem untuk menghasilkan keluaran.

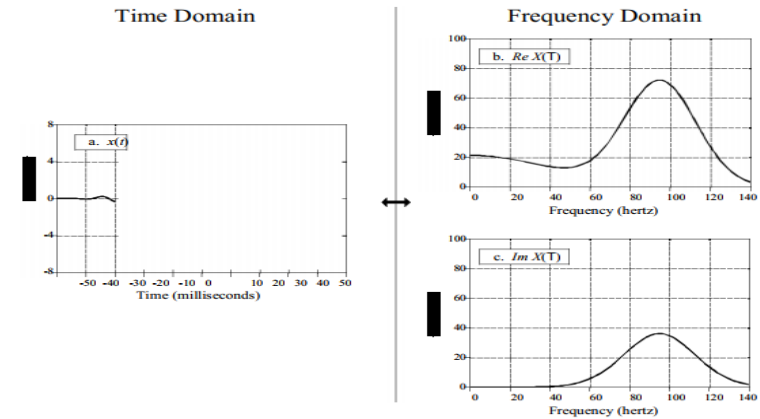


Gambar 5.7

Menghitung lilitan berdasarkan segmen. Karena banyak sinyal kontinu ditentukan oleh wilayah, maka Perhitungan konvolusi harus dilakukan wilayah demi wilayah. Dalam contoh ini, perhitungan sinyal keluaran dipecah menjadi tiga bagian: (a) tidak ada tumpang tindih, (b) tumpang tindih sebagian, dan (c) tumpang tindih total, dari sinyal input dan respon impuls bergeser-membalik Transformasi Fourier

Transformasi Fourier untuk sinyal kontinu dibagi menjadi dua kategori, satu untuk sinyal yang periodik, dan satu untuk sinyal yang bersifat aperiodik. Sinyal periodik menggunakan suatu versi

Transformasi Fourier disebut Seri Fourier, dan dibahas selanjutnya bagian. Transformasi Fourier yang digunakan dengan sinyal aperiodik disebut Fourier Mengubah. Bab ini menjelaskan teknik Fourier ini hanya dengan menggunakan matematika nyata, seperti yang telah dilakukan beberapa bab terakhir untuk sinyal diskrit.



Gambar 5.8 Contoh Transformasi Fourier. Sinyal domain waktu

Contoh Transformasi Fourier. Sinyal domain waktu, $x(t)$, meluas dari negatif ke positif tak terhingga. Domain frekuensi terdiri dari bagian nyata, $Re X(T)$, dan bagian imajiner, $Im X(T)$, masing-masing memanjang. dari nol hingga tak terhingga positif. Sumbu frekuensi dalam ilustrasi ini diberi label dalam siklus per detik (hertz). Untuk ubah ke frekuensi natural, kalikan angka pada sumbu frekuensi dengan $2B$.

Contoh Transformasi Fourier. Sinyal domain waktu, $x(t)$, meluas dari negatif ke positif tak terhingga. Domain frekuensi terdiri dari bagian nyata, $Re X(T)$, dan bagian imajiner, $Im X(T)$, masing-masing memanjang. dari nol hingga tak terhingga positif. Sumbu frekuensi dalam ilustrasi ini diberi label dalam siklus per detik (hertz). Untuk ubah ke frekuensi natural, kalikan angka pada sumbu frekuensi dengan $2B$.

		$Re X(T) = \int_{-\infty}^{\infty} x(t) \cos(Tt) dt$ $Im X(T) = \int_{-\infty}^{\infty} x(t) \sin(Tt) dt$ <p>Persamaan: Analisis transformasi Fourier persamaan. Dalam persamaan ini, $Re X(T)$ & $Im X(T)$ adalah yang nyata dan bagian imajiner dari frekuensi spektrum, masing-masing, dan $x(t)$ adalah sinyal domain waktu dianalisis.</p>
	Sub -CMK6	Sistem Diskrit Teori Sampling

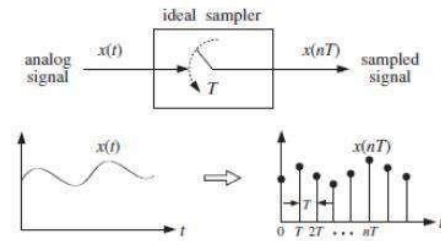


Fig. 1.3.1 Ideal sampler.

Gambar 7.1 Sampler ideal

DTFT adalah kasus khusus dari transformasi Z. Transformasi Z bilateral didefinisikan sebagai:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

Jadi kasus khususnya adalah $z = e^{i\omega}$. Since $|e^{i\omega}| = 1$, itu adalah evaluasi transformasi Z di sekitar lingkaran satuan dalam bidang kompleks

Tabel Pasangan Transformasi Z umum

	Signal, $x[n]$	Z-transform, $X(z)$	ROC
1	$\delta[n]$	1	all z
2	$\delta[n - n_0]$	z^{-n_0}	$z \neq 0$
3	$u[n]$	$\frac{1}{1 - z^{-1}}$	$ z > 1$
4	$-u[-n - 1]$	$\frac{1}{1 - z^{-1}}$	$ z < 1$

5	$nu[n]$	$\frac{z^{-1}}{(1 - z^{-1})^2}$	$ z > 1$
6	$-nu[-n - 1]$	$\frac{z^{-1}}{(1 - z^{-1})^2}$	$ z < 1$
7	$n^2u[n]$	$\frac{z^{-1}(1 + z^{-1})}{(1 - z^{-1})^3}$	$ z > 1$
8	$-n^2u[-n - 1]$	$\frac{z^{-1}(1 + z^{-1})}{(1 - z^{-1})^3}$	$ z < 1$

9	$n^3 u[n]$	$\frac{z^{-1}(1 + 4z^{-1} + z^{-2})}{(1 - z^{-1})^4}$	$ z > 1$
10	$-n^3 u[-n - 1]$	$\frac{z^{-1}(1 + 4z^{-1} + z^{-2})}{(1 - z^{-1})^4}$	$ z < 1$
11	$a^n u[n]$	$\frac{1}{1 - az^{-1}}$	$ z > a $
12	$-a^n u[-n - 1]$	$\frac{1}{1 - az^{-1}}$	$ z < a $
13	$na^n u[n]$	$\frac{az^{-1}}{(1 - az^{-1})^2}$	$ z > a $

--	--	--

14	$-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z < a $
15	$n^2 a^n u[n]$	$\frac{az^{-1}(1+az^{-1})}{(1-az^{-1})^3}$	$ z > a $
16	$-n^2 a^n u[-n-1]$	$\frac{az^{-1}(1+az^{-1})}{(1-az^{-1})^3}$	$ z < a $
17	$\cos(\omega_0 n) u[n]$	$\frac{1-z^{-1}\cos(\omega_0)}{1-2z^{-1}\cos(\omega_0)+z^{-2}}$	$ z > 1$

18	$\sin(\omega_0 n)u[n]$	$\frac{z^{-1} \sin(\omega_0)}{1 - 2z^{-1} \cos(\omega_0) + z^{-2}}$	$ z > 1$
19	$a^n \cos(\omega_0 n)u[n]$	$\frac{1 - az^{-1} \cos(\omega_0)}{1 - 2az^{-1} \cos(\omega_0) + a^2 z^{-2}}$	$ z > a $
20	$a^n \sin(\omega_0 n)u[n]$	$\frac{az^{-1} \sin(\omega_0)}{1 - 2az^{-1} \cos(\omega_0) + a^2 z^{-2}}$	$ z > a $

Properti dari Transformasi Z

Mengingat

$$\mathcal{Z}[x_n] = \mathcal{Z}[x[n]] = X(z) \quad \text{and} \quad \mathcal{Z}[y_n] = \mathcal{Z}[y[n]] = Y(z),$$

bahwa

maka

transformasi Z memiliki property seperti berikut

(i) Linearity. $Z\{c_1 x_n + c_2 y_n\} = Z\{c_1 x[n] + c_2 y[n]\} = c_1 X(z) + c_2 Y(z)$,

(ii) Delay Shift. $Z\{x[n-N] u[n-N]\} = X(z) z^{-N}$,

(iii) Advance

Shift. $Z\{x[n+N]\} = z^N (X(z) - x[0] - x[1] z^{-1} - x[2] z^{-2} - \dots - x[N-1] z^{-(N-1)})$, or

$$Z\{x_{n+N}\} = z^N (X(z) - x_0 - x_1 z^{-1} - x_2 z^{-2} - \dots - x_{N-1} z^{-(N-1)})$$

(iv) Multiplication by n . $Z\{n x_n\} = Z\{n x[n]\} = -z \frac{d}{dz} X(z)$,

--	--	--

	Sequence	z - transform
1	$\delta[n]$	1
2	$u[n]$	$\frac{z}{z-1}$
3	b^n	$\frac{z}{z-b}$
4	$b^{n-1}u[n-1]$	$\frac{1}{z-b}$
5	e^{an}	$\frac{z}{z-e^a}$
6	n	$\frac{z}{(z-1)^2}$
7	n^2	$\frac{z(z+1)}{(z-1)^3}$
8	$b^n n$	$\frac{bz}{(z-b)^2}$
9	$e^{an} n$	$\frac{ze^a}{(z-e^a)^2}$
10	$\sin(an)$	$\frac{\sin(a)z}{z^2-2\cos(a)z+1}$
11	$b^n \sin(an)$	$\frac{\sin(a)bz}{z^2-2\cos(a)bz+b^2}$
12	$\cos(an)$	$\frac{z(z-\cos(a))}{z^2-2\cos(a)z+1}$
13	$b^n \cos(an)$	$\frac{z(z-b\cos(a))}{z^2-2\cos(a)bz+b^2}$

Tabel 7.1
Transformasi Z dari beberapa urutan umum

Ujian UTS Pengolahan Sinyal Digital

Sub –CMK 8

Invers Transformasi Z

Ketika menggunakan transformasi Z

$$X(z) = \int_{n=-\infty}^{n=\infty} x[n]z^{-n}$$

(1)

seringkali berguna untuk dapat menemukan $x[n]$ diberikan $X(z)$.
Setidaknya ada 4 metode berbeda untuk melakukan ini:

1. Inspeksi (Bagian 1)
2. Ekspansi Fraksi Parsial (Bagian 2)
3. Perluasan Seri Daya (Bagian 3)
4. Integrasi Kontur (Bagian 4)

1. Metode Inspeksi

"Metode" ini pada dasarnya menjadi terbiasa dengan tabel pasangan z-transform 2 dan kemudian "merekayasa balik".

Contoh 1:

Saat diberikan

$$X(z) = \frac{z}{z - \alpha}$$

Dengan ROC³ dari

$$|z| > \alpha$$

Kita bisa menentukan "dengan inspeksi" itu

$$X[n] = \alpha^n u[n]$$

2. Metode Ekspansi Sebagian-Pecahan

Ketika berhadapan dengan sistem invarian waktu linier, transformasi-z sering dalam bentuk

$$X(z) = \frac{B(z)}{A(z)} \tag{2}$$

$$\frac{k^M = 0 (bkz^{-k})}{P_k N = 0 (akz^{-k})}$$

Ini juga bisa dinyatakan sebagai

$$X(z) = \frac{k^M = 1 - cz^{-1}}{Q^K = 1 (1 - dz^{-1})}$$

(3)

Dimana ck mewakili nol bukan nol dari $X(z)$ dan dk mewakili kutub bukan nol.

Jika $M < N$ maka $X(z)$ dapat direpresentasikan sebagai

$$X(z) = \frac{k^{N-1} A_k}{k^{-1} d_k z^{-1}} \quad (4)$$

Formulir ini memungkinkan inversi yang mudah dari setiap suku penjumlahan menggunakan metode inspeksi (Bagian 1) dan tabel transformasi 4. Jadi jika pembilangnya adalah polynomial

Maka perlu menggunakan ekspansi pecahan-parsial 5 untuk meletakkan $X(z)$ dalam bentuk di atas. Jika $M \geq N$ maka $X(z)$

3. Metode Perluasan Seri Daya

Saat transformasi- z diartikan sebagai deret pangkat dalam bentuk

$$X(z) = \int_{n=-\infty}^{n=\infty} x[n]z^{-n}$$

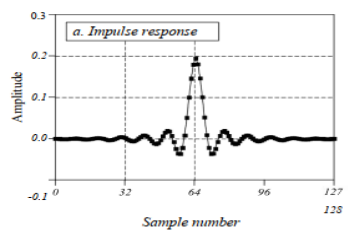
maka setiap suku dari barisan $x[n]$ dapat ditentukan dengan melihat koefisien dari kekuatan masing-masing z^{-n}

4. Metode Integrasi Kontur

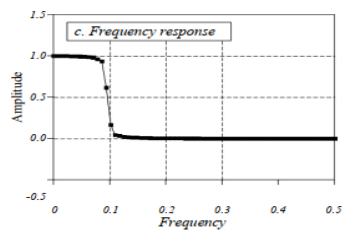
$$x[n] = \frac{1}{2\pi i} \int_{\Gamma} X(z) z^{n-1} dz$$

Dimana Γ adalah kontur berlawanan arah jarum jam di ROC dari $X(z)$ yang melingkari titik awal bidang- z . Untuk lebih memperluas metode menemukan invers ini membutuhkan pengetahuan tentang teori variabel kompleks dan dengan

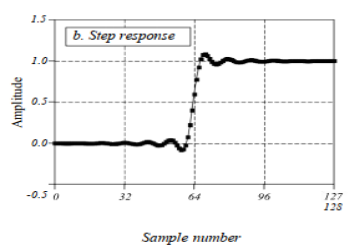
		demikian tidak akan dibahas dalam modul ini.
	Sub- CMK 9	<p>Filter Digital:</p> <p>Filter digital adalah bagian yang sangat penting dari DSP. Nyatanya, kinerja luar biasa mereka adalah salah satu alasan utama DSP menjadi begitu populer. Seperti yang disebutkan dalam pendahuluan, filter memiliki dua kegunaan: pemisahan sinyal dan pemulihan sinyal. Pemisahan sinyal diperlukan jika sinyal telah terkontaminasi oleh interferensi, derau, atau sinyal lainnya. Misalnya, bayangkan alat untuk mengukur aktivitas listrik jantung bayi (EKG) saat masih dalam kandungan. Sinyal mentah kemungkinan akan rusak oleh pernapasan dan detak jantung ibu. Filter dapat digunakan untuk memisahkan sinyal ini sehingga dapat dianalisis satu per satu.</p> <p>Seperti yang ditunjukkan pada Gambar 14-1, setiap filter linier memiliki respons impuls, respons langkah, dan respons frekuensi. Masing-masing respons ini berisi informasi lengkap tentang filter, tetapi dalam bentuk yang berbeda. Jika salah satu dari ketiganya ditentukan, dua lainnya ditetapkan dan dapat dihitung secara langsung. Ketiga representasi ini penting, karena menggambarkan bagaimana filter akan bereaksi dalam keadaan yang berbeda.</p> <p>Ada juga cara lain untuk membuat filter digital yang disebut rekursi. Ketika filter diimplementasikan dengan konvolusi, setiap sampel dalam output dihitung dengan memberi bobot pada sampel di input, dan menambahkannya bersama-sama. Filter rekursif adalah perpanjangan dari ini, menggunakan nilai yang dihitung sebelumnya dari keluaran, selain titik dari masukan. Alih-alih menggunakan kernel filter, filter rekursif ditentukan oleh sekumpulan koefisien rekursi.</p>



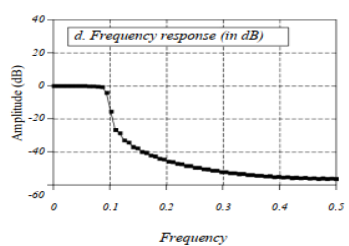
FFT



Integrate



20 Log()



Gambar 9.1

Parameter filter. Setiap filter linier memiliki respons impuls, respons langkah, dan respons frekuensi. Respon langkah, (b), dapat ditemukan dengan integrasi diskrit dari respon impuls, (a). Respon frekuensi dapat ditemukan dari respon impuls dengan menggunakan Fast Fourier Transform (FFT), dan dapat ditampilkan baik pada skala linier, (c), atau dalam desibel, (d)

Misalnya, rangkaian elektronik yang memiliki 3 bels amplifikasi menghasilkan sinyal keluaran dengan $10 \times 10 \times 10 = 1000$ kali kekuatan masukan. Desibel (dB) adalah sepersepuluh bel. Oleh karena itu, nilai desibel: -20dB, -10dB, 0dB, 10dB & 20dB, berarti rasio daya masing-masing: 0,01, 0,1, 1, 10, & 100. Dengan kata lain, setiap sepuluh desibel berarti daya telah berubah dengan faktor sepuluh.

Misalnya, bayangkan penguat dengan penguatan 20dB. Menurut definisi, ini berarti bahwa daya dalam sinyal telah meningkat dengan faktor 100. Karena amplitudo sebanding dengan akar kuadrat daya, amplitudo keluaran adalah 10 kali amplitudo masukan. Sementara 20dB berarti faktor 100 dalam kekuatan, itu hanya berarti faktor 10 dalam amplitudo. Setiap dua puluh desibel berarti amplitudo telah berubah dengan faktor sepuluh. Dalam bentuk persamaan

$$dB = 10 \log_{10} \frac{P_2}{P_1}$$

$$dB = 20 \log_{10} \frac{A_2}{A_1}$$

PERSAMAAN 9.1

Definisi desibel. Desibel adalah cara untuk menyatakan rasio antara dua sinyal. Rasio daya (P1 & P2) menggunakan persamaan yang berbeda dari rasio amplitudo (A1 & A2).

Persamaan di atas menggunakan logaritma basis 10; namun, banyak bahasa komputer hanya menyediakan fungsi untuk logaritma basis (log natural, tertulis loge x atau ln x). Log natural dapat digunakan dengan memodifikasi persamaan di atas: $dB = 4.342945 \log_e (P_2 / P_1)$ dan $dB = 8.685890 \log_e (A_2 / A_1)$.

Pertama, -3dB berarti amplitudo dikurangi menjadi 0,707 (dan dayanya oleh karena itu dikurangi menjadi 0,5). Kedua, hafalkan konversi berikut antara desibel dan rasio amplitudo:

60Db	=	1000
40Db	=	100
20Db	=	10
0Db	=	1
-20Db	=	0.1
-40Db	=	0.01
-60Db	=	0.001

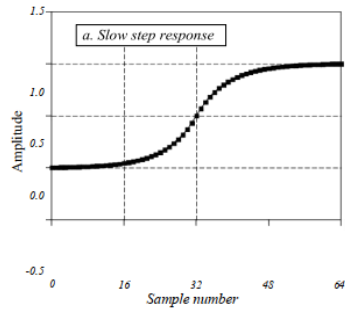
Bagaimana Informasi Diwakili dalam Sinyal

Informasi yang direpresentasikan dalam domain waktu menggambarkan kapan sesuatu terjadi dan berapa amplitudo kejadiannya. Misalnya, bayangkan sebuah eksperimen untuk mempelajari keluaran cahaya dari matahari. Output cahaya diukur dan direkam sekali setiap detik. Setiap sampel dalam sinyal menunjukkan apa yang terjadi pada saat itu, dan tingkat kejadiannya. Jika suar matahari terjadi, sinyal secara langsung memberikan informasi tentang waktu terjadinya, durasi, perkembangan dari waktu ke waktu, dll. Setiap sampel berisi informasi yang dapat diinterpretasikan tanpa mengacu pada sampel lain. Sekalipun Anda hanya memiliki satu sampel dari sinyal ini, Anda masih tahu sesuatu tentang apa yang Anda ukur. Ini adalah cara termudah agar informasi terkandung dalam sinyal.

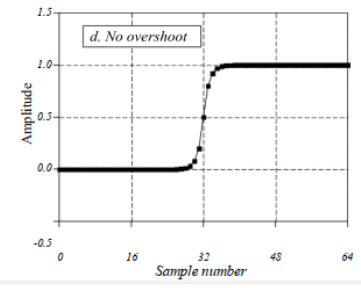
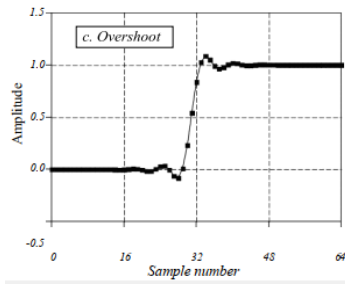
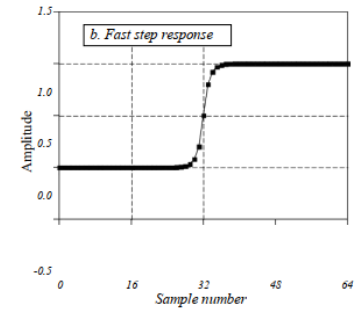
Informasi yang direpresentasikan dalam domain frekuensi lebih tidak langsung. Banyak hal di alam semesta kita yang menunjukkan gerakan periodik. Misalnya, gelas anggur yang dipukul dengan kuku akan bergetar, menghasilkan suara dering; bandul jam kakek berayun maju mundur; bintang dan planet berputar pada porosnya dan berputar mengelilingi satu sama lain, dan seterusnya. Dengan mengukur frekuensi, fase, dan amplitudo dari gerakan periodik ini, informasi tentang sistem yang menghasilkan gerakan seringkali dapat diperoleh. Misalkan kita mengambil sampel suara yang dihasilkan oleh gelas anggur yang berdering. Frekuensi fundamental dan harmonik dari getaran periodik berhubungan dengan massa dan elastisitas material. Sampel tunggal, dengan sendirinya, tidak mengandung informasi tentang gerakan periodik, dan oleh karena itu tidak ada informasi tentang gelas anggur. Informasi tersebut terkandung dalam hubungan antara banyak titik dalam sinyal.

		<p>Parameter Domain Waktu</p> <p>Parameter respons langkah yang penting dalam desain filter ditunjukkan pada Gambar. 8.2. Untuk membedakan kejadian dalam sebuah sinyal, durasi respons langkah harus lebih pendek daripada jarak kejadian. Ini menentukan bahwa respons langkah harus secepat mungkin (jargon DSP). Ini ditunjukkan pada Gambar. (a) & (b). Cara yang paling umum untuk menentukan waktu penelitian (lebih banyak jargon) adalah dengan mengutip jumlah sampel antara tingkat amplitudo 10% dan 90%. Mengapa risetime yang sangat cepat tidak selalu memungkinkan? Ada banyak alasan, pengurangan noise, batasan inheren dari sistem akuisisi data, menghindari aliasing, dll.</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

POOR



GOOD



Gambar 9.2

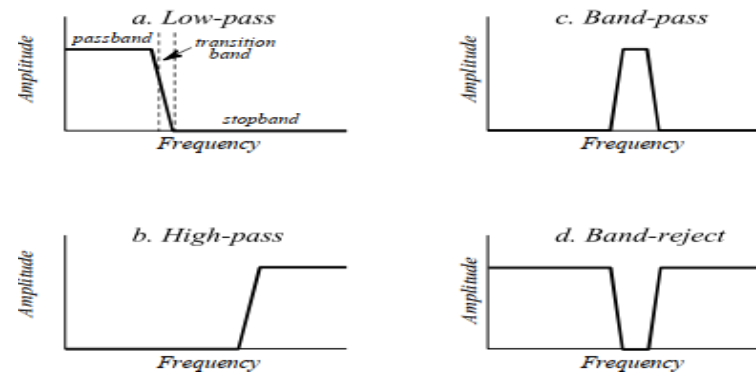
Parameter untuk mengevaluasi kinerja domain waktu. Respons langkah digunakan untuk mengukur seberapa baik kinerja filter didomain waktu. Tiga parameter penting: (1) kecepatan transisi (waktu riset), ditunjukkan pada (a) dan (b), (2) overshoot, ditunjukkan pada b. Respon langkah cepat Amplitudo Amplitudo Amplitudo Amplitudo (c) dan (d), dan (3) linieritas fase (simetri antara bagian atas dan bawah anak tangga), ditunjukkan pada (e) dan (f).

Parameter Domain Frekuensi

Gambar 9.3 menunjukkan empat respons frekuensi dasar. Tujuan dari filter ini adalah untuk memungkinkan beberapa frekuensi lewat tanpa perubahan, sementara sepenuhnya memblokir frekuensi lain. Passband mengacu pada frekuensi-frekuensi yang dilewatkan, sedangkan stopband berisi frekuensi-frekuensi yang diblokir. Dalam desain filter analog, frekuensi cutoff biasanya didefinisikan di mana amplitudo dikurangi menjadi 0,707 (yaitu, -3dB). Filter digital kurang terstandarisasi, dan adalah umum untuk melihat tingkat amplitudo 99%, 90%, 70,7%, dan 50% ditetapkan sebagai frekuensi cutoff.

Gambar 8.4 menunjukkan tiga parameter yang mengukur seberapa baik kinerja filter dalam domain frekuensi. Untuk memisahkan

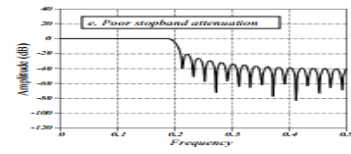
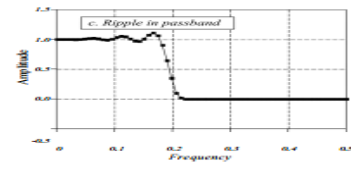
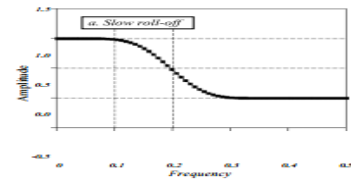
frekuensi yang jaraknya berdekatan, filter harus memiliki roll-off yang cepat, seperti yang diilustrasikan dalam (a) dan (b). Agar frekuensi passband dapat bergerak melalui filter tanpa perubahan, tidak boleh ada riak passband, seperti yang ditunjukkan pada (c) dan (d). Terakhir, untuk memblokir frekuensi stopband secara memadai, diperlukan redaman stopband yang baik, yang ditampilkan di (e) dan (f).



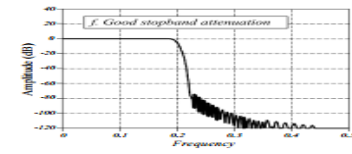
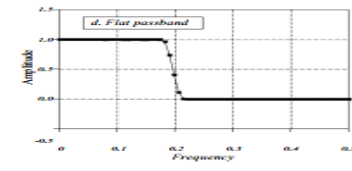
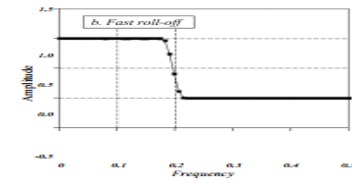
Gambar 9.3

Empat tanggapan frekuensi umum. Filter domain frekuensi umumnya digunakan untuk melewati frekuensi tertentu (passband), sementara memblokir yang lain (stopband). Empat respons yang paling umum: low-pass, high-pass, band-pass, dan band-reject

POOR



GOOD



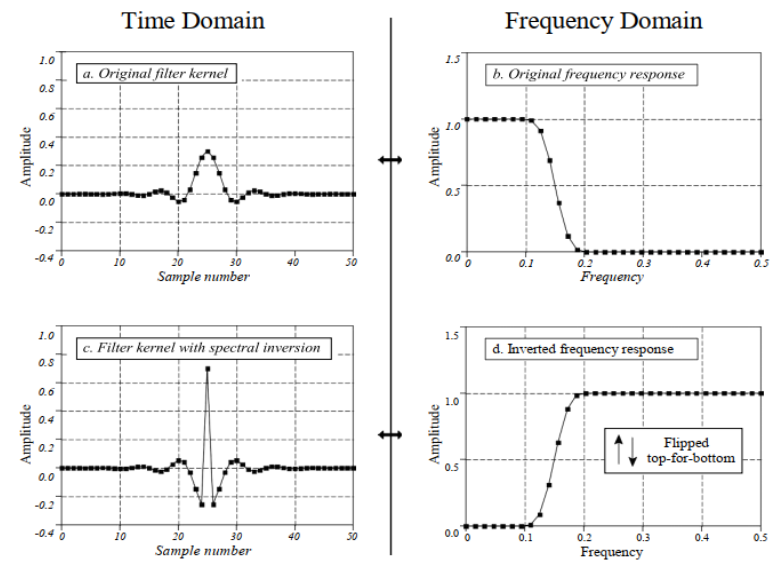
Gambar 9.4

Parameter untuk mengevaluasi kinerja domain frekuensi. Respons frekuensi yang ditampilkan adalah untuk filter low-pass. Tiga parameter penting: (1) ketajaman roll-off, ditunjukkan pada (a) dan (b), (2) riak passband, ditunjukkan pada (c) dan (d), dan (3) atenuasi stopband, ditunjukkan pada (e) dan (f).

Mengambil langkah ini lebih jauh, Anda dapat menambahkan banyak angka nol ke respons impuls untuk membuatnya, katakanlah, 256, 512, atau 1024 poin. Ide pentingnya adalah bahwa respons impuls yang lebih lama menghasilkan jarak yang lebih dekat dari titik data dalam respons frekuensi. Artinya, ada lebih banyak sampel yang tersebar antara DC dan setengah dari laju pengambilan sampel. Mengambil ini secara ekstrem, jika respons impuls diisi dengan angka nol yang tak terbatas, titik data dalam respons frekuensi sangat berdekatan, yaitu garis kontinu. Dengan kata lain, respons frekuensi filter benar-benar merupakan sinyal kontinu antara DC dan setengah dari laju pengambilan sampel. Output dari DFT adalah pengambilan sampel dari jalur kontinu ini. Berapa lama respons impuls yang harus Anda gunakan saat menghitung respons frekuensi filter? Sebagai pemikiran pertama, coba $N = 1024$, tetapi jangan takut untuk mengubahnya jika diperlukan (seperti resolusi yang tidak memadai atau waktu komputasi yang berlebihan).

Ingatlah bahwa parameter "baik" dan "buruk" yang dibahas dalam bab ini hanyalah generalisasi. Banyak sinyal yang tidak bisa dikategorikan dengan baik. Misalnya, pertimbangkan sinyal EKG yang terkontaminasi dengan interferensi 60 hertz. Informasi tersebut dikodekan dalam domain waktu, tetapi gangguan paling baik ditangani dalam domain frekuensi. Desain terbaik untuk aplikasi ini adalah terikat untuk memiliki trade-off, dan mungkin

bertentangan dengan kebijaksanaan konvensional bab ini. Ingat aturan pendidikan nomor satu: *Paragraf dalam buku tidak memberi Anda izin untuk berhenti berpikir.*



Gambar 9.5

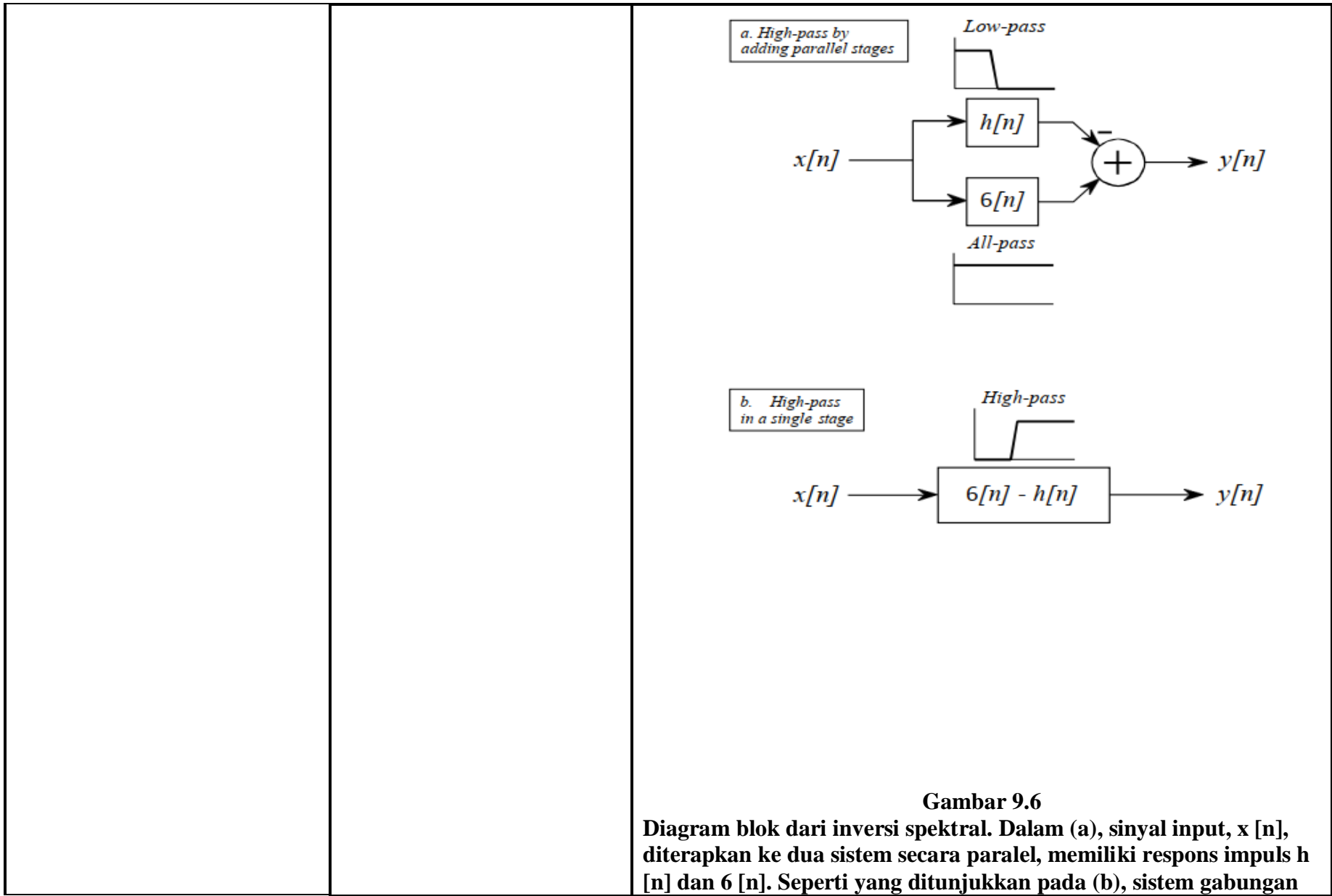
Contoh inversi spektral. Kernel filter low-pass di (a) memiliki respons frekuensi yang ditunjukkan di (b). Kernel filter high-pass, (c), dibentuk dengan mengubah tanda setiap sampel di (a), dan menambahkan satu ke sampel di pusat simetri. Tindakan ini dalam domain waktu membalikkan spektrum frekuensi (yaitu, membalikinya dari atas ke bawah), seperti yang ditunjukkan oleh respons frekuensi lintasan tinggi di (d).

High-Pass, Band-Pass and Band-Reject Filters

Filter high-pass, band-pass dan band-reject dirancang dengan memulai dengan filter low-pass, dan kemudian mengubahnya menjadi respons yang diinginkan. Oleh karena itu, sebagian besar diskusi tentang desain filter hanya memberikan contoh filter akses rendah. Ada dua metode untuk konversi low-pass ke high-pass: inversi spektral dan pembalikan spektral.

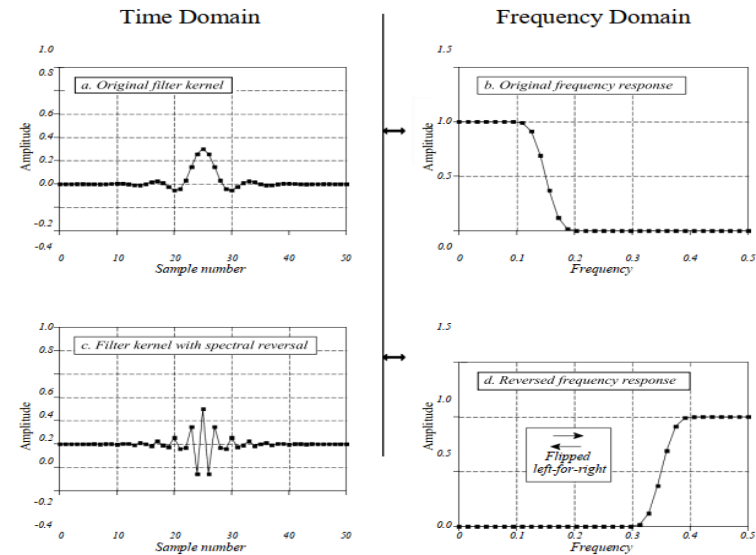
Gambar 9.6 menunjukkan mengapa modifikasi dua langkah pada domain waktu ini menghasilkan spektrum frekuensi yang terbalik. Dalam (a), sinyal input, $x[n]$, diterapkan ke dua sistem secara paralel. Salah satu sistem ini adalah filter low-pass, dengan respons impuls yang diberikan oleh $h[n]$. Sistem lain tidak melakukan apa pun terhadap sinyal, dan oleh karena itu memiliki respons impuls yang merupakan fungsi delta, $\delta[n]$. Output keseluruhan, $y[n]$, sama dengan output dari sistem all-pass dikurangi output dari sistem low-pass. Karena komponen frekuensi rendah dikurangkan dari sinyal asli, hanya komponen frekuensi tinggi yang muncul di keluaran. Dengan demikian, filter high-pass terbentuk, yang dapat dilakukan sebagai operasi dua langkah dalam program komputer: jalankan sinyal melalui filter low-pass, dan kemudian kurangi sinyal yang disaring dari aslinya. Namun, seluruh operasi dapat dilakukan dalam tahap sinyal dengan menggabungkan dua kernel filter. Seperti yang dijelaskan di Bab. Seperti yang ditunjukkan pada (b), kernel filter untuk filter high-pass diberikan oleh: $\delta[n] - h[n]$. Artinya, ubah tanda semua sampel, lalu tambahkan satu ke sampel di pusat simetri. Agar teknik ini bekerja, komponen

		<p>frekuensi rendah yang keluar dari filter jalur rendah harus memiliki fase yang sama dengan komponen frekuensi rendah yang keluar dari sistem semua jalur. Jika tidak, pengurangan total tidak dapat dilakukan. Ini menempatkan dua batasan pada metode: (1) kernel filter asli harus memiliki simetri kiri-kanan (yaitu, fase nol atau linier), dan (2) impuls harus ditambahkan di pusat simetri.</p>
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Gambar 9.6
Diagram blok dari inversi spektral. Dalam (a), sinyal input, $x[n]$, diterapkan ke dua sistem secara paralel, memiliki respons impuls $h[n]$ dan $6[n]$. Seperti yang ditunjukkan pada (b), sistem gabungan

memiliki respons impuls $6[n] - h[n]$. Ini berarti bahwa respons frekuensi dari sistem gabungan adalah inversi dari respons frekuensi $h[n]$.

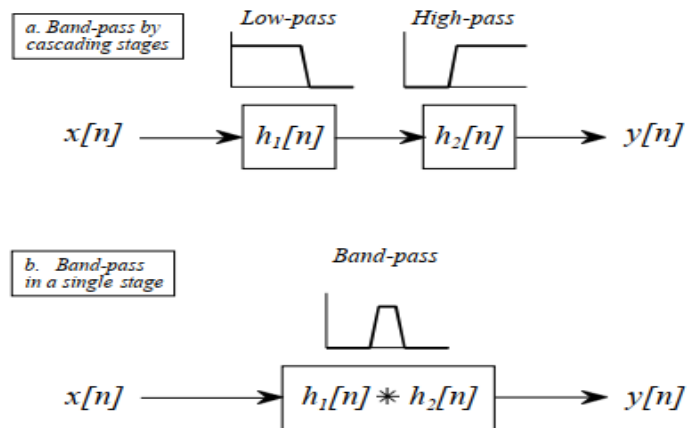


Gambar 9.7

Contoh pembalikan spektral. Kernel filter low-pass di (a) memiliki respons frekuensi yang ditunjukkan di (b). Kernel filter high-pass, (c), dibentuk dengan mengubah tanda dari setiap sampel lain di (a). Tindakan dalam domain waktu ini menghasilkan domain frekuensi yang dibalik dari kiri ke kanan, menghasilkan respons frekuensi lintasan tinggi yang ditunjukkan pada (d). sistem paralel dengan keluaran tambahan dapat digabungkan menjadi satu tahap dengan menambahkan tanggapan impulsnya.

Metode kedua untuk konversi jalur rendah ke jalur tinggi, pembalikan spektral, diilustrasikan pada Gambar 9.7. Sama seperti

sebelumnya, kernel filter low-pass di (a) sesuai dengan respons frekuensi di (b). Kernel filter high-pass, (c), dibentuk dengan mengubah tanda dari setiap sampel lain di (a). Seperti yang ditunjukkan pada (d), ini membalik domain frekuensi kiri-kekanan: 0 menjadi 0,5 dan 0,5 menjadi 0. Frekuensi cutoff dari contoh low-pass filter adalah 0,15, menghasilkan frekuensi cutoff dari filter high-pass menjadi 0,35. Mengubah tanda setiap sampel lain sama dengan mengalikan kernel filter dengan sinusoid dengan frekuensi dari 0,5. Seperti dibahas dalam Bab 10, ini memiliki efek menggeser domain frekuensi sebesar 0,5. Lihatlah (b) dan bayangkan frekuensi negatif antara -0.5 dan 0 yang merupakan bayangan cermin dari frekuensi antara 0 dan 0.5. Frekuensi yang muncul di (d) adalah frekuensi negatif dari (b) digeser 0,5. Terakhir, Gambar. 9.8 dan 9.9 menunjukkan bagaimana kernel filter jalur rendah dan jalur tinggi dapat digabungkan untuk membentuk filter band-pass dan band-reject



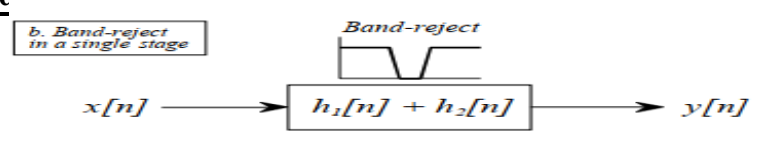
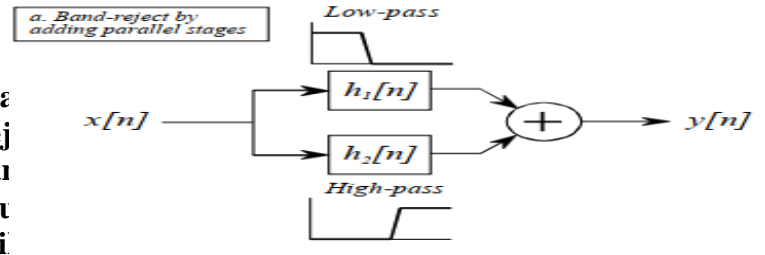
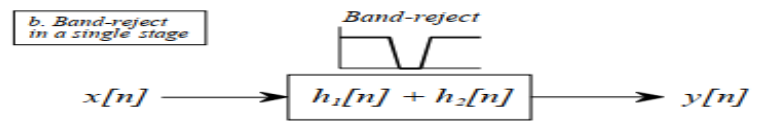
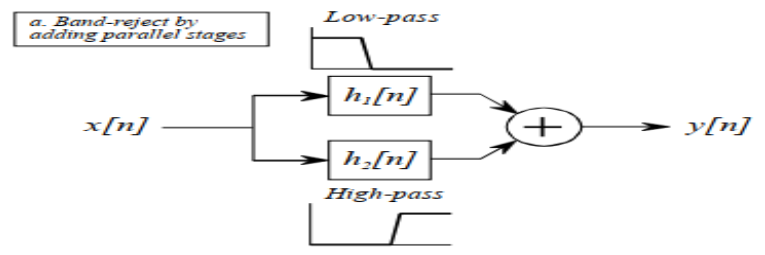
Gambar 9.8

Mendesain filter band-pass. Seperti yang ditunjukkan pada (a), band-pass filter dapat dibentuk dengan mengalirkan low-pass filter dan high-pass filter. Ini dapat direduksi menjadi satu tahap, ditunjukkan pada (b). Kernel filter dari tahap tunggal sama dengan konvolusi dari kernel filter jalur-rendah dan jalur-tinggi

Klasifikasi Filter

Tabel 9.1 merangkum bagaimana filter digital diklasifikasikan menurut penggunaannya dan implementasinya. Penggunaan filter digital dapat dibagi menjadi tiga kategori: domain waktu, domain frekuensi, dan kustom. Seperti dijelaskan sebelumnya, filter domain waktu digunakan ketika informasi dikodekan dalam bentuk gelombang sinyal. Penyaringan domain waktu digunakan untuk tindakan seperti: penghalusan, penghilangan DC, pembentukan bentuk gelombang, dll. Sebaliknya, filter domain frekuensi digunakan ketika informasi terdapat dalam amplitudo, frekuensi, dan fase dari komponen sinusoid. Tujuan dari filter ini adalah untuk memisahkan satu pita frekuensi dari yang lain. Filter kustom digunakan saat tindakan khusus diperlukan oleh filter, sesuatu yang lebih rumit daripada empat respons dasar (high-pass, low-pass,

band-pass dan band-reject



Mendesain band-reject filter dan (b) menentukan kernel filter rendah c

), bar r

		<p style="text-align: center;"><i>FILTER DITERAPKAN OLEH:</i></p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;"><i>Konvolusi</i></td> <td style="text-align: center;"><i>Rekursi</i></td> </tr> <tr> <td></td> <td style="text-align: center;"><i>Finite Impulse Response (FIR)</i></td> <td style="text-align: center;"><i>Infinite Impulse Response (IIR)</i></td> </tr> </table> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="vertical-align: middle;"> FILTER DIGUNAKAN UNTUK { <i>Time Domain</i> (<i>smoothing, DC removal</i>) <i>Frequency Domain</i> (<i>separating frequencies</i>) <i>Custom</i> (<i>Deconvolution</i>) </td> <td style="border: 1px solid black; padding: 5px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><i>Moving average (Ch. 15)</i></td> <td style="padding: 5px;"><i>Single pole (Ch. 19)</i></td> </tr> <tr> <td style="padding: 5px;"><i>Windowed-sinc (Ch. 16)</i></td> <td style="padding: 5px;"><i>Chebyshev (Ch. 20)</i></td> </tr> <tr> <td style="padding: 5px;"><i>FIR custom (Ch. 17)</i></td> <td style="padding: 5px;"><i>Iterative design (Ch. 26)</i></td> </tr> </table> </td> </tr> </table> <p style="text-align: center;">TABEL 9.1 Klasifikasi filter. Filter dapat dibagi berdasarkan penggunaannya, dan bagaimana penerapannya.</p> <p>Filter digital dapat diimplementasikan dalam dua cara, dengan konvolusi (juga disebut <i>Finite Impulse Response</i> atau FIR) dan dengan rekursi (juga disebut <i>Infinite Impulse Response</i> atau IIR). Filter yang dilakukan dengan konvolusi dapat memiliki kinerja yang jauh lebih baik daripada filter yang menggunakan rekursi, tetapi dijalankan jauh lebih lambat. Seperti yang ditunjukkan pada Tabel 8.1, konvolusi dan rekursi adalah teknik saingan; Anda harus menggunakan salah satunya untuk aplikasi tertentu.</p>		<i>Konvolusi</i>	<i>Rekursi</i>		<i>Finite Impulse Response (FIR)</i>	<i>Infinite Impulse Response (IIR)</i>	FILTER DIGUNAKAN UNTUK { <i>Time Domain</i> (<i>smoothing, DC removal</i>) <i>Frequency Domain</i> (<i>separating frequencies</i>) <i>Custom</i> (<i>Deconvolution</i>)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><i>Moving average (Ch. 15)</i></td> <td style="padding: 5px;"><i>Single pole (Ch. 19)</i></td> </tr> <tr> <td style="padding: 5px;"><i>Windowed-sinc (Ch. 16)</i></td> <td style="padding: 5px;"><i>Chebyshev (Ch. 20)</i></td> </tr> <tr> <td style="padding: 5px;"><i>FIR custom (Ch. 17)</i></td> <td style="padding: 5px;"><i>Iterative design (Ch. 26)</i></td> </tr> </table>	<i>Moving average (Ch. 15)</i>	<i>Single pole (Ch. 19)</i>	<i>Windowed-sinc (Ch. 16)</i>	<i>Chebyshev (Ch. 20)</i>	<i>FIR custom (Ch. 17)</i>	<i>Iterative design (Ch. 26)</i>
	<i>Konvolusi</i>	<i>Rekursi</i>														
	<i>Finite Impulse Response (FIR)</i>	<i>Infinite Impulse Response (IIR)</i>														
FILTER DIGUNAKAN UNTUK { <i>Time Domain</i> (<i>smoothing, DC removal</i>) <i>Frequency Domain</i> (<i>separating frequencies</i>) <i>Custom</i> (<i>Deconvolution</i>)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><i>Moving average (Ch. 15)</i></td> <td style="padding: 5px;"><i>Single pole (Ch. 19)</i></td> </tr> <tr> <td style="padding: 5px;"><i>Windowed-sinc (Ch. 16)</i></td> <td style="padding: 5px;"><i>Chebyshev (Ch. 20)</i></td> </tr> <tr> <td style="padding: 5px;"><i>FIR custom (Ch. 17)</i></td> <td style="padding: 5px;"><i>Iterative design (Ch. 26)</i></td> </tr> </table>	<i>Moving average (Ch. 15)</i>	<i>Single pole (Ch. 19)</i>	<i>Windowed-sinc (Ch. 16)</i>	<i>Chebyshev (Ch. 20)</i>	<i>FIR custom (Ch. 17)</i>	<i>Iterative design (Ch. 26)</i>									
<i>Moving average (Ch. 15)</i>	<i>Single pole (Ch. 19)</i>															
<i>Windowed-sinc (Ch. 16)</i>	<i>Chebyshev (Ch. 20)</i>															
<i>FIR custom (Ch. 17)</i>	<i>Iterative design (Ch. 26)</i>															
	Sub- CMK10	<p>Filter Rekursif Filter rekursif adalah cara yang efisien untuk mencapai respons impuls yang panjang, tanpa harus melakukan konvolusi yang lama.</p>														

Filter ini bekerja sangat cepat, tetapi memiliki kinerja dan fleksibilitas yang lebih rendah dibandingkan filter digital lainnya. Filter rekursif juga disebut filter *Infinite Impulse Response* (IIR), karena respons impulsnya terdiri dari eksponensial yang membusuk. Ini membedakannya dari filter digital yang dilakukan dengan konvolusi, yang disebut filter *Finite Impulse Response* (FIR).
Metode Rekursif

Untuk memulai diskusi tentang filter rekursif, bayangkan Anda perlu mengekstrak informasi dari beberapa sinyal $x[n]$. Kebutuhan Anda begitu besar sehingga Anda menyewa seorang profesor matematika tua untuk memproses data untuk Anda. Tugas profesor adalah menyaring $x[n]$ untuk menghasilkan $y[n]$, yang diharapkan berisi informasi yang Anda minati. Profesor memulai pekerjaannya menghitung setiap titik dalam $y[n]$ menurut beberapa algoritme yang terkunci rapat di over-nya-otak berkembang.

$x[n-1]$ adalah poin 27, dll. Untuk memahami algoritma yang digunakan, kami bertanya

diri kita sendiri: "Informasi apa yang tersedia bagi profesor untuk menghitung $y[n]$, sampel yang sedang dikerjakan?" Sumber informasi yang paling jelas adalah sinyal input, yaitu nilai: $x[n]$, $x[n-1]$, $x[n-2]$,.... Profesor itu bisa saja mengalikan setiap titik dalam sinyal masukan dengan koefisien, dan menjumlahkan hasil perkaliannya:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3x[n-3] + \dots$$

Anda harus menyadari bahwa ini tidak lebih dari konvolusi sederhana, dengan koefisien: a_0 , a_1 , a_2 , membentuk kernel konvolusi. Jika hanya ini yang dilakukan profesor, tidak akan banyak

kebutuhan untuk cerita ini, atau bab ini. Namun, ada sumber informasi lain yang profesor memiliki akses ke: nilai sinyal output yang dihitung sebelumnya, diadakan di: $y[n-1]$, $y[n-2]$, $y[n-$

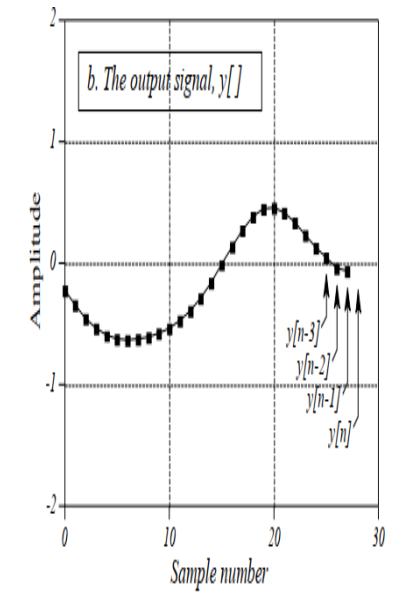
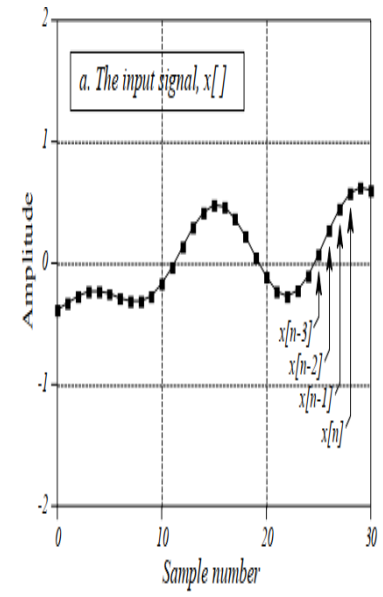
3),.... Dengan menggunakan informasi tambahan ini, algoritme akan berbentuk:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3x[n-3] + \dots \\ + b_1y[n-1] + b_2y[n-2] + b_3y[n-3] + \dots$$

Persamaan 10.1

Persamaan rekursi. Dalam persamaan ini, $x[n]$ adalah sinyal masukan, $y[n]$ adalah sinyal keluaran, dan a dan b adalah koefisien.

Dengan kata lain, setiap titik dalam sinyal keluaran ditemukan dengan mengalikan nilai dari sinyal masukan dengan koefisien "a", mengalikan nilai yang dihitung sebelumnya dari sinyal keluaran dengan koefisien "b", dan menjumlahkan produknya. Perhatikan bahwa tidak ada nilai untuk b_0 , karena ini sesuai dengan sampel yang sedang dihitung. Persamaan 10-1 disebut persamaan rekursi, dan filter yang menggunakannya disebut filter rekursif. Nilai "a" dan "b" yang menentukan filter disebut koefisien rekursi. Dalam praktik aktual, tidak lebih dari selusin koefisien rekursi dapat digunakan atau filter menjadi tidak stabil (yaitu, keluaran terus meningkat atau berosilasi). Tabel 10.1 menunjukkan contoh program filter rekursif. Filter rekursif berguna karena melewati konvolusi yang lebih panjang. Misalnya, pertimbangkan apa yang terjadi ketika fungsi delta diteruskan melalui filter rekursif. Keluarannya adalah respons impuls filter, dan biasanya berupa osilasi sinusoidal yang meluruh secara eksponensial. Karena respons impuls ini dalam panjang tak terhingga, filter rekursif sering disebut *Infinite Impulse Response (IIR)*. Akibatnya, filter rekursif menggabungkan sinyal input dengan kernel filter yang sangat panjang, meskipun hanya beberapa koefisien yang terlibat.



Gambar 10.1

Notasi filter rekursif. Sampel keluaran yang dihitung, $y[n]$, ditentukan oleh nilai dari sinyal masukan, $x[n]$, $x[n-1]$, $x[n-2]$,..., serta nilai yang dihitung sebelumnya di sinyal keluaran, $y[n-1]$, $y[n-2]$, $y[n-3]$,.... Angka-angka ini ditunjukkan untuk $n = 28$

Ini adalah bidang mereka yang berspesialisasi dalam DSP. Ada tiga cara untuk mencari koefisien rekursi tanpa harus memahami transformasi-z. Pertama, bab ini memberikan persamaan desain untuk beberapa jenis filter rekursif sederhana. Kedua, Bab 20 menyediakan program komputer "buku masak" untuk mendesain filter low-pass dan high-pass Chebyshev yang lebih canggih

Ontoh Program Recursif:

```
100 'FILTER PENYELAMAT
```

```
110 '
```

```
120 DIM X[499]           Menahan sinyal input
```

```
130 DIM Y[499]           Menahan sinyal keluaran filterd
```

```
140 '
```

```
150 GOSUB XXXX           'subrutin mitos untuk menghitung  
rekursi
```

```
160 '                   'koefisien: A0, A1, A2, B1, B2
```

```
170 '
```

```
180 GOSUB XXXX           'subrutin mitos untuk memuat X []  
dengan data masukan
```

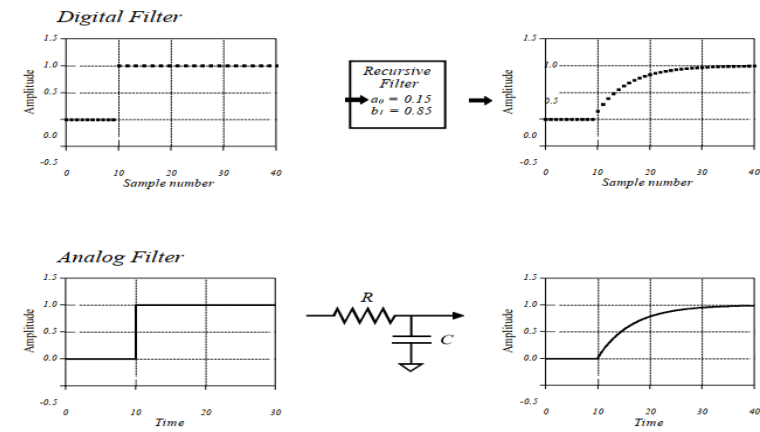
```
190 '
```

```
200 FOR I% = 2 TO 499
```

```
210 Y[I%]+A1*X[I%-1]+A2*X[I%-2]+B1*Y[I%-1]+B2*Y[I%-2]
```

```
220 NEXT I%
```

230 ‘
240 END



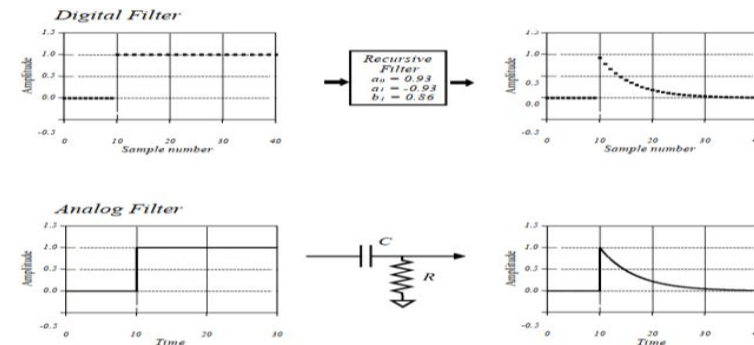
Gambar 10.2

Filter low-pass tiang tunggal. Filter rekursif digital dapat meniru filter analog yang terdiri dari resistor dan kapasitor. Seperti yang

ditunjukkan dalam contoh ini, filter rekursif low-pass kutub tunggal menghaluskan tepi input langkah, seperti filter RC elektronik

Filter Rekursif Tiang Tunggal Gambar 10.2 menunjukkan contoh dari apa yang disebut filter low-pass kutub tunggal. Filter rekursif ini hanya menggunakan dua koefisien, $a_0 = 0.15$ dan $b_1 = 0.85$. Untuk contoh ini, sinyal input adalah fungsi langkah. Seperti yang Anda harapkan untuk filter alow-pass, hasilnya adalah peningkatan yang mulus ke tingkat kondisi mapan. Ini

Gambar 10.3 juga menunjukkan sesuatu yang terkait dengan pengetahuan Anda tentang elektronik. Filter rekursif low-pass ini sepenuhnya dianalogikan dengan filter low-pass elektronik yang terdiri dari resistor dan kapasitor tunggal. Keindahan dari metode rekursif adalah kemampuannya untuk membuat berbagai macam respon dengan hanya mengubah beberapa parameter. Misalnya, Gambar 10.3 menunjukkan filter dengan tiga koefisien: $a_0 = 0.93$, $a_1 = -0.93$ dan $b_1 = 0.86$. Sebagai ditunjukkan oleh respons langkah yang sama, filter digital ini meniru filter high-pass RC elektronik. Filter rekursif kutub tunggal ini pasti sesuatu yang ingin Anda simpan di kotak peralatan DSP Anda.



Gambar 10.3

Filter high-pass tiang tunggal. Pemilihan koefisien yang tepat juga dapat membuat filter rekursif meniru filter high-pass RC elektronik. Filter rekursif kutub tunggal ini dapat digunakan di DSP sama seperti Anda menggunakan sirkuit RC dalam elektronik analog

Untuk mengeksekusi, dan menghasilkan sedikit kejutan. Koefisien ditemukan dari persamaan sederhana ini:

$$\begin{aligned}a_0 &= 1 - x \\ b_1 &= x\end{aligned}$$

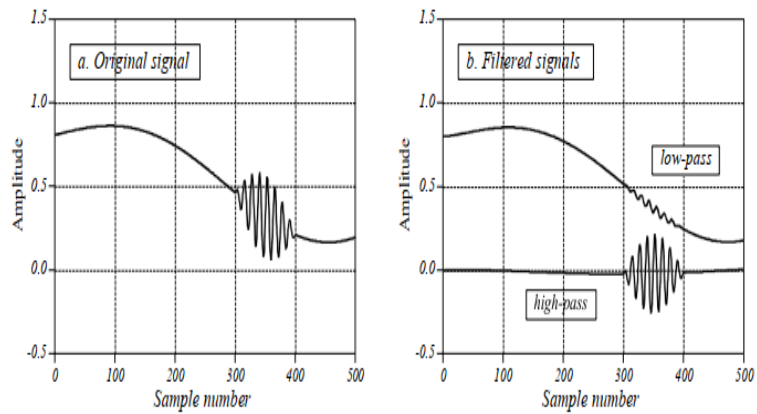
PERSAMAAN 10.2

Filter low-pass tiang tunggal. Respons filter dikontrol oleh parameter, x , nilai antara nol dan satu.

$$\begin{aligned}a_0 &= \frac{(1 + x)}{2} \\ a_1 &= -\frac{(1 + x)}{2} \\ b_1 &= x\end{aligned}$$

Persamaan 10.3

Filter high-pass tiang tunggal.



Gambar 10.4

Contoh filter rekursif kutub tunggal. Dalam (a), ledakan frekuensi tinggi terjadi pada sinyal yang bervariasi secara perlahan. Dalam (b), filter jalur-rendah dan jalur-tinggi kutub tunggal digunakan untuk memisahkan dua komponen. Filter low-pass menggunakan $\alpha = 0.95$, sedangkan high-pass filter untuk $\alpha = 0.86$.

diinginkan. Sama seperti $R \times C$ adalah jumlah detik yang dibutuhkan rangkaian RC untuk meluruh menjadi 36,8% dari nilai akhirnya, d adalah jumlah sampel yang diperlukan untuk filter rekursif untuk meluruh ke tingkat yang sama ini:

$$x = e^{-1/d}$$

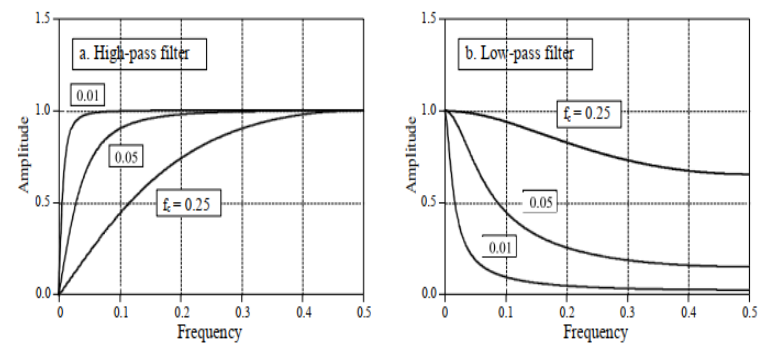
Persamaan (10.4)

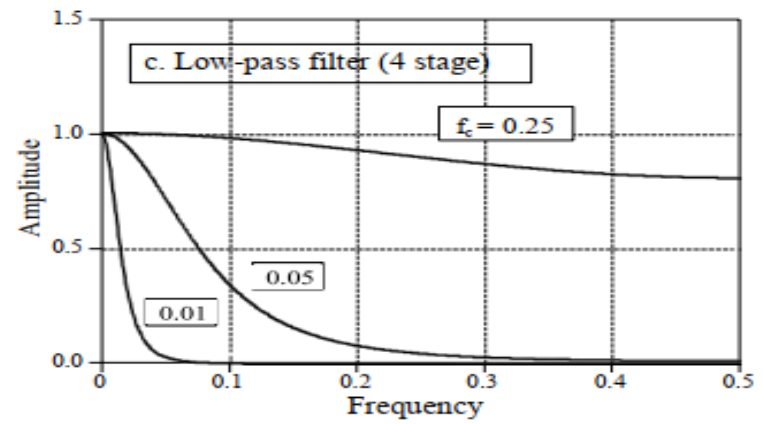
Konstanta waktu filter kutub tunggal. Persamaan ini menghubungkan jumlah peluruhan antar sampel, x , dengan konstanta waktu filter, d , jumlah sampel untuk peluruhan filter menjadi 36,8%.

$$x = e^{-2\pi f c}$$

Persamaan (10.5)

Frekuensi cutoff filter kutub tunggal. Besarnya peluruhan antar sampel, x , terkait dengan frekuensi cutoff filter, f_c , nilai antara 0 dan 0,5. Ini memberikan tiga cara untuk mencari koefisien "a" dan "b", dimulai dengan konstanta waktu, frekuensi cutoff, atau langsung memilih x . Gambar 19-4 menunjukkan contoh penggunaan filter rekursif kutub tunggal. Dalam (a), sinyal asli adalah kurva halus, kecuali ledakan gelombang sinus frekuensi tinggi. Gambar (b) menunjukkan sinyal setelah melewati filter jalur rendah dan jalur tinggi. Sinyal telah dipisahkan dengan cukup baik, tetapi tidak sempurna, seperti halnya rangkaian RC sederhana digunakan pada sinyal analog.





Gambar 10.5

Respons frekuensi kutub tunggal. Gambar (a) dan (b) menunjukkan respons frekuensi dari filter rekursif kutub tunggal jalur-tinggi dan jalur-rendah. Gambar (c) menunjukkan respons frekuensi dari kaskade empat filter low-pass. Respons frekuensi filter rekursif tidak selalu seperti yang Anda harapkan, terutama jika filter didorong hingga batas ekstrem. Misalnya, kurva $f_c = 0,25$ pada (c) tidak berguna. Banyak

faktor yang harus disalahkan, termasuk: aliasing, round-off noise, dan respons fase nonlinier.

Filter lolos-rendah empat tahap sebanding dengan filter Blackman dan tetapi dengan kecepatan eksekusi yang jauh lebih cepat.

Persamaan desain untuk low-pass filter empat tahap adalah:

$$a_0 = (1 - x)^4$$

$$b_1 = 4x$$

$$b_2 = -6x^2$$

$$b_3 = 4x^3$$

$$b_4 = -x^4$$

Filter Pita Sempit

Kebutuhan umum dalam elektronik dan DSP adalah mengisolasi pita frekuensi yang sempit dari sinyal bandwidth yang lebih luas.

Misalnya, Anda mungkin ingin menghilangkan interferensi 60 hertz dalam sistem instrumentasi, atau mengisolasi nada sinyal dalam jaringan telepon. Tersedia dua jenis respons frekuensi: band-pass dan band-reject (juga disebut filter takik). Gambar 9.6

menunjukkan respons frekuensi dari filter ini, dengan koefisien rekursi yang disediakan oleh persamaan berikut

$$\begin{aligned}
 a_0 &= 1 - K \\
 a_1 &= 2(K - R) \cos(2\pi f) \\
 a_2 &= R^2 - K \\
 b_1 &= 2R \cos(2\pi f) \\
 b_2 &= -R^2
 \end{aligned}$$

Persamaan (10.7)

Filter band-pass. Contoh respon frekuensi ditunjukkan pada Gambar 9.6a. Untuk menggunakan persamaan ini, pertama-tama pilih frekuensi tengah, f , dan bandwidth, BW. Keduanya dinyatakan sebagai sebagian kecil dari laju pengambilan sampel, dan oleh karena itu dalam kisaran 0 hingga 0,5. Selanjutnya, hitung R , lalu K , lalu koefisien rekursi

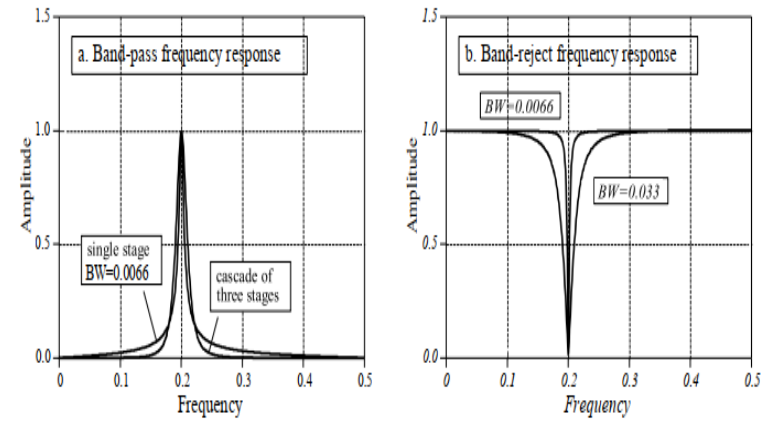
$$\begin{aligned}a_0 &= K \\a_1 &= -2K \cos(2\nu f) \\a_2 &= K \\b_1 &= 2R \cos(2\nu f) \\b_2 &= -R^2\end{aligned}$$

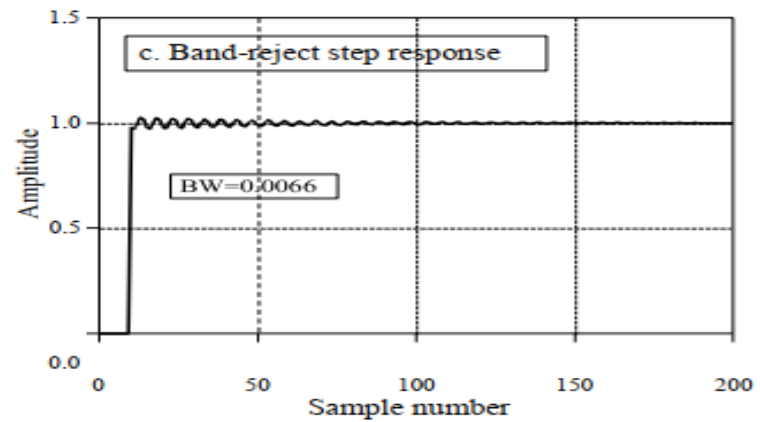
where:

$$K = \frac{1 - 2R \cos(2\nu f) + R^2}{2 - 2 \cos(2\nu f)}$$

$$R = 1 - 3 BW$$

Persamaan (10.8)
Filter penolakan pita. Filter ini biasa disebut filter takik. Contoh respons frekuensi ditunjukkan pada Gambar 9.6b





Gambar 10.6

Karakteristik filter pita sempit. Gambar (a) dan (b) menunjukkan respons frekuensi dari berbagai filter band-pass dan band-reject. Respon langkah dari band-reject filter ditunjukkan pada (c). Filter band-reject (notch) berguna untuk menghilangkan 60 Hz dan interferensi serupa dari bentuk gelombang yang dikodekan dalam domain waktu

Dua parameter harus dipilih sebelum menggunakan persamaan ini: f , frekuensi tengah, dan BW, bandwidth (diukur pada amplitudo 0,707). Keduanya dinyatakan sebagai fraksi frekuensi sampling, dan oleh karena itu harus antara 0 dan 0,5. Dari dua nilai yang

ditentukan ini, hitung variabel perantara: R dan K, dan kemudian koefisien rekursi. Seperti yang ditunjukkan pada (a), filter band-pass memiliki ekor yang relatif besar yang memanjang dari puncak utama. Ini dapat ditingkatkan dengan berjenjang beberapa tahap. Karena persamaan desain cukup panjang, lebih mudah untuk mengimplementasikan kaskade ini dengan memfilter sinyal beberapa kali, daripada mencoba mencari koefisien yang diperlukan untuk filter tunggal. Gambar (b) menunjukkan contoh filter penolakan pita. Bandwidth tersempit yang dapat diperoleh dengan presisi tunggal adalah sekitar 0,0003 dari frekuensi sampling. Saat didorong melebihi batas ini, redaman takik akan menurun. Angka (c) menunjukkan respon langkah dari band-reject filter. Ada overshoot dan dering yang terlihat, tetapi amplitudonya cukup kecil. Hal ini memungkinkan filter untuk menghilangkan gangguan pita sempit (60 Hz dan sejenisnya) dengan hanya distorsi kecil pada bentuk gelombang domain waktu.

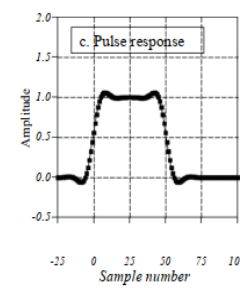
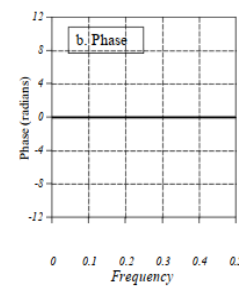
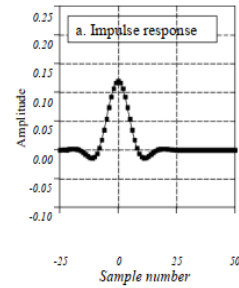
Respon Fase

Ada tiga jenis respons fase yang dapat dimiliki filter: fase nol, fase linier, dan fase nonlinier. Contoh dari masing-masing ini ditunjukkan pada Gambar 10-7. Seperti yang ditunjukkan pada (a), filter fase nol dicirikan oleh respons impuls yang simetris di sekitar sampel nol. Bentuk sebenarnya tidak masalah, hanya sampel bernomor negatif yang merupakan bayangan cermin dari sampel bernomor positif. Ketika transformasi Fourier diambil dari bentuk gelombang simetris ini, fasa akan seluruhnya nol, seperti yang ditunjukkan pada (b). Kerugian dari filter fasa nol adalah membutuhkan penggunaan indeks negatif, yang dapat merepotkan untuk dikerjakan. Filter fase linier adalah jalan keluarnya. Respons impuls dalam (d) identik dengan yang ditunjukkan pada (a), kecuali itu telah digeser untuk hanya menggunakan sampel bernomor positif. Respon impuls masih simetris antara kiri dan kanan; namun, lokasi simetri telah bergeser dari nol. Pergeseran ini

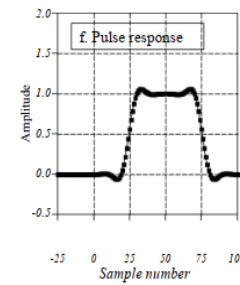
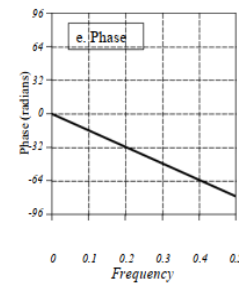
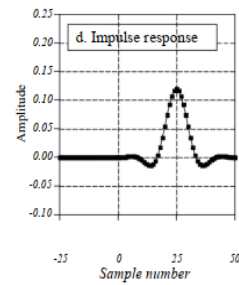
menghasilkan fase, (e), menjadi garis lurus, terhitung namanya: fase linier. Kemiringan garis lurus ini berbanding lurus dengan besarnya pergeseran. Karena pergeseran dalam respons impuls tidak melakukan apa pun selain menghasilkan pergeseran yang identik pada sinyal keluaran, filter fase linier setara dengan filter fase nol untuk sebagian besar tujuan. (g) menunjukkan respon impuls yang tidak simetris antara kiri dan kanan. Sejalan dengan itu, fase, (h), bukanlah garis lurus. Dengan kata lain, ia memiliki fase nonlinier. Jangan mengacaukan istilah: fase nonlinier dan fase linier dengan konsep linieritas sistem yang dibahas pada Bab 5. Meskipun keduanya menggunakan kata linier, keduanya tidak terkait. Mengapa ada yang peduli jika fase linier atau tidak? Gambar (c), (f), dan (i) menunjukkan jawabannya. Ini adalah respons denyut nadi masing-masing dari tiga filter. Respon denyut nadi tidak lebih dari respon langkah berjalan positif diikuti dengan respon langkah berjalan negatif. Respons pulsa digunakan di sini karena ini menampilkan apa yang terjadi pada tepi naik dan turun dalam sebuah sinyal. Inilah bagian penting: Filter fase nol dan linier memiliki tepi kiri dan kanan yang terlihat sama, sedangkan filter fase nonlinier memiliki tepi kiri dan kanan yang terlihat berbeda. Banyak aplikasi yang tidak dapat mentolerir tampilan tepi kiri dan kanan yang berbeda. Salah satu contohnya adalah tampilan osiloskop, di mana perbedaan ini dapat disalahartikan sebagai fitur sinyal yang sedang diukur. Contoh lainnya adalah dalam pengolahan video. Dapatkah Anda membayangkan menyalakan TV untuk menemukan telinga kiri aktor favorit Anda terlihat berbeda dari telinga kanannya? Mudah untuk membuat filter FIR (respons impuls terbatas) memiliki fase linier. Ini karena respons impuls (kernel filter) secara langsung ditentukan dalam proses desain. Membuat kernel filter memiliki simetri kiri-kanan adalah semua yang diperlukan. Ini tidak terjadi pada filter IIR (rekursif), karena koefisien rekursi adalah yang ditentukan, bukan respons impuls. Respon impuls dari filter

rekursif tidak simetris antara kiri dan kanan, dan oleh karena itu mempunyai fase nonlinier.

Zero Phase Filter

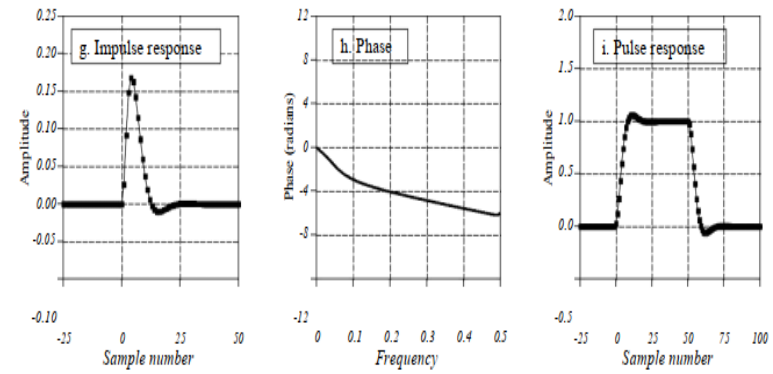


Linear Phase Filter



--	--	--

Nonlinear Phase Filter



Gambar 10.7

Filter fase nol, linier, dan nonlinear. Filter fase nol memiliki respons impuls yang memiliki simetri kiri-kanan di sekitar nomor sampel nol, seperti pada (a). Ini menghasilkan respons frekuensi yang

memiliki fase yang seluruhnya terdiri dari nol, seperti pada (b). Respons impuls fase nol diinginkan karena respons langkahnya simetris antara atas dan bawah, membuat tepi kiri dan kanan pulsa terlihat sama, seperti yang ditunjukkan pada (c). Filter fase linier memiliki simetri kiri-kanan, tetapi tidak di sekitar sampel nol, 331 seperti yang diilustrasikan dalam (d). Ini menghasilkan fase yang linier, yaitu garis lurus, seperti yang ditunjukkan pada (e). Respons pulsa fase linier, ditunjukkan pada (f), memiliki semua keuntungan dari respons pulsa fase nol. Sebagai perbandingan, respons impuls filter fase nonlinier tidak simetris antara kiri dan kanan, seperti pada (g), dan fase bukan garis lurus, seperti pada (h). Bagian terburuknya adalah tepi kiri dan kanan dari respons denyut nadi tidak sama, seperti yang ditunjukkan pada (i).

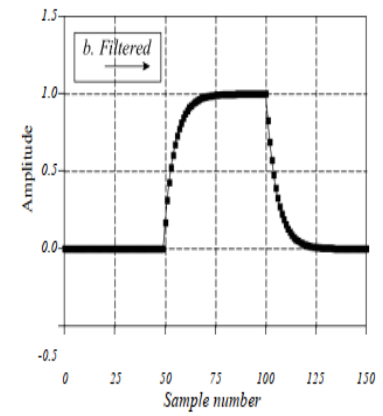
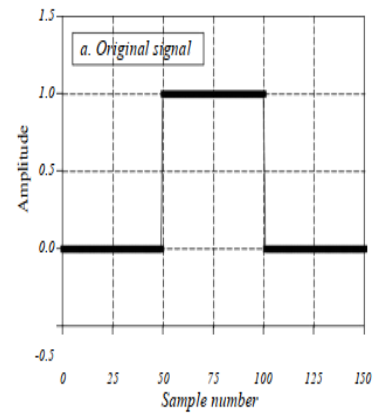
Gambar 10.8 menunjukkan contoh cara kerjanya. Sinyal input yang akan difilter ditunjukkan pada (a). Gambar (b) menunjukkan sinyal setelah disaring oleh filter low-pass kutub tunggal. Karena ini adalah filter fase nonlinier, tepi kiri dan kanan tidak terlihat sama; mereka adalah versi terbalik satu sama lain. Seperti yang telah dijelaskan sebelumnya, filter rekursif ini diimplementasikan dengan memulai dari sampel 0 dan bekerja menuju sampel 150, dengan menghitung setiap sampel. Sekarang, misalkan alih-alih berpindah dari sampel 0 menuju sampel 150, kita mulai dari sampel 150 dan bergerak menuju sampel 0. Dengan kata lain, setiap sampel pada sinyal keluaran dihitung dari sampel masukan dan keluaran di sebelah kanan sampel yang sedang dikerjakan. Artinya persamaan rekursi, Persamaan 9.1, diubah menjadi:

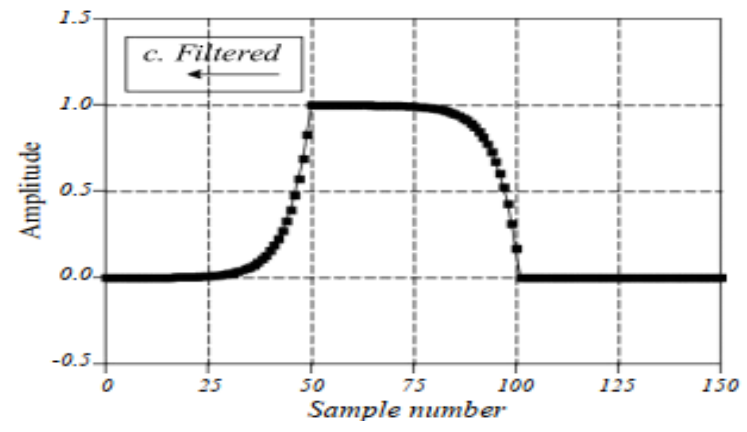
$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3x[n-3] + \dots \\ + b_1y[n-1] + b_2y[n-2] + b_3y[n-3] + \dots$$

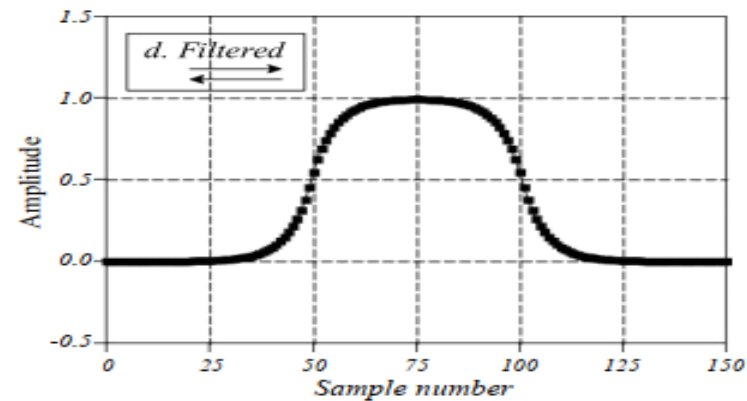
PERSAMAAN 10.9

Persamaan rekursi terbalik. Ini sama dengan Persamaan. 10.1,

kecuali sinyal disaring dari kiri ke kanan, bukan dari kanan ke kiri.







Gambar 10.8

Pemfilteran rekursif dua arah. Sinyal input pulsa persegi panjang ditunjukkan pada (a). Gambar (b) menunjukkan sinyal setelah disaring dengan filter low-pass rekursif kutub tunggal, lewat dari kiri ke kanan. Dalam (c), sinyal telah diproses dengan cara yang sama, kecuali dengan filter yang bergerak dari kanan ke kiri. Gambar (d) menunjukkan sinyal setelah disaring dari kiri ke kanan dan kemudian dari kanan ke kiri. Filter rekursif apa pun dapat

		<p>dibuat fase nol dengan menggunakan teknik ini.</p> <p>Menggunakan Bilangan bulat Titik mengambang presisi tunggal sangat ideal untuk menerapkan filter rekursif sederhana ini. Penggunaan bilangan bulat dimungkinkan, tetapi jauh lebih sulit. Ada dua masalah utama. Pertama, kesalahan pembulatan dari jumlah bit yang terbatas dapat menurunkan respons filter, atau bahkan membuatnya tidak stabil. Kedua, nilai pecahan dari koefisien rekursi harus ditangani dengan matematika integer. Salah satu cara untuk mengatasi masalah ini adalah dengan menyatakan setiap koefisien sebagai pecahan. Misalnya, 0,15 menjadi 19/128. Alih-alih mengalikan dengan 0,15, Anda mengalikannya terlebih dahulu dengan 19 dan kemudian membagi dengan 128. Cara lain adalah mengganti perkalian dengan tabel pemeta. Misalnya, ADC 12 bit menghasilkan sampel dengan nilai antara 0 dan 4095. Alih-alih mengalikan setiap sampel dengan 0,15, Anda meneruskan sampel melalui tabel pencarian yang panjangnya 4096 entri. Nilai yang diperoleh dari tabel pemeta sama dengan 0,15 kali nilai yang masuk ke tabel pemeta. Metode ini sangat cepat, tetapi membutuhkan memori ekstra; tabel pencarian terpisah diperlukan untuk setiap koefisien. Sebelum Anda mencoba salah satu metode bilangan bulat ini, pastikan algoritme rekursif untuk filter rata-rata bergerak tidak sesuai dengan kebutuhan Anda. Itu menyukai bilangan bulat.</p>
	Sub-CMK 11	Latihan membuat Filter Recursif
	Sub-CMK 12	Latihan membuat Filter Recursif untuk Filter Chebichef
	Sub- CMK13	Lanjutan membuat filter recursif Chebichef

	Sub-CMK 14	Evaluasi Filter recursif				
	Sub-CPMK15	Latihan membuat Filter chebichef				
	Sub-CMK 16	UAS Pengolahan Sinyal Digital				
	Korelasi CPMK terhadap Sub CPMK					
	Sub-CPMK1	Sub-CPMK2	Sub-CPMK3	Sub-CPMK4	Sub-CPMK5	Sub-CPMK6
CPMK1	v					
CPMK2		v	v			
CPMK3				v		
BPMK4					v	v
Deskripsi singkat MK	<p>Orientasi dari pengolahan sinyal processing ini di utamakan untuk mahasiswa undergraduate kualifikasi S2. Praktisi Engineer mahasiswa S1 sebagai pemula. Pada saat ini Digital siyal processing di ajarkan juga untuk kualifikasi college karena perkembangan dari CD player, terutama untuk music synthesizer, sound cards dalam personal computer. Dengan adanya A/D dan D/A akan memberikan effects pada processor digital audio. Akhirnya dengan belajar DSP ini banyak mamfaat yang didapat dalam aplikasi praktis nya berupa Mp3 dan MP4 utk pengolahan kualitas gambar video. Pembelajaran dilaksanakan dengan, berupaya melibatkan mahasiswa secara optimal melalui penugasan dan diskusi kelas.</p>					
Bahan Kajian : Materi Pembekajaran	<ol style="list-style-type: none"> 1. 2. 3. 4. dst					
Pustaka	<ol style="list-style-type: none"> 1. B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, New York, 1969. 2. A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, Prentice Hall, 					

Englewood Cliffs, NJ, 1989.

3. A. V. Oppenheim and R. W. Schaffer, **Digital Signal Processing**, Prentice Hall, Englewood Cliffs, NJ, 1975.
4. L. R. Rabiner and B. Gold, **Theory and Application of Digital Signal Processing**, Prentice Hall, Englewood Cliffs, NJ, 1975.
5. S. K. Mitra and J. F. Kaiser, eds., **Handbook of Digital Signal Processing**, Wiley, New York, 1993.
6. T. W. Parks and C. S. Burrus, **Digital Filter Design**, Wiley, New York, 1987.
7. A. Antoniou, **Digital Filters: Analysis and Design**, 2nd ed., McGraw-Hill, New York, 1993.
8. D. F. Elliott, ed., **Handbook of Digital Signal Processing**, Academic Press, New York, 1987.
9. L. R. Rabiner and C. M. Rader, eds., **Digital Signal Processing**, IEEE Press, New York, 1972.
10. **Selected Papers in Digital Signal Processing, II**, edited by the Digital Signal Processing Committee and IEEE ASSP, IEEE Press, New York, 1976.
11. **Programs for Digital Signal Processing**, edited by the Digital Signal Processing Committee, IEEE ASSP Society, IEEE Press, New York, 1979.
12. A. V. Oppenheim, ed., **Applications of Digital Signal Processing**, Prentice Hall, Englewood Cliffs, NJ, 1978.
13. J. S. Lim and A. V. Oppenheim, eds., **Advanced Topics in Signal Processing**, Prentice Hall, Englewood Cliffs, NJ, 1988.
14. R. A. Roberts and C. T. Mullis, **Digital Signal Processing**, Addison-Wesley, Reading, MA, 1987.

	<p>15. P. A. Lynn and W. Fuerst, Introductory Digital Signal Processing with Computer Applications,</p> <p>16. Pendukung :</p> <p>1.</p> <p>2.</p> <p>dst</p>
Dosen Pengampu	Ir. Herry Satria Utama, MT
Mata Kuliah Prasyarat	Matematika Teknik I, Matematika Teknik II, RL II, Dasar Sistem kendali

Minggu ke	Kemampuan Akhir tiap tahapan belajar (Sub CPMK)	Penilaian		Bentuk Pembelajaran; Metode Pembelajaran; Penugasan Mahasiswa; [Estimasi Waktu]		Materi Pembelajaran [Pustaka]	Bobot Penilaian (%)
		Indikator	Kriteria & teknik	Luring	Daring		
1	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan jenis fungsi dan jenis besaran kelistrikan Menjelaskan sumber tegangan listrik dan bagaimana menghasilkan sumber tegangan.	150 menit		v	v		

	Secara kimia, gerakan fluida, dan gerakan air, dan angin.						
2	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan proses kwantisasi, kinerja analog to digital, dan Digital to analog.						
3	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : System linear, time variant dan time invariant						
4	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menghitung output respons menggunakan teori convolusi.						

5	<p>Setelah mahasiswa mengikuti kuliah bahan kajian dari pertemuan 1 s/d 4 secara tuntas, mahasiswa diharapkan dapat :</p> <p>a. Memahami, menguasai, dan mampu menjelaskan bahan kajian yang telah diikutinya secara tuntas dari pertemuan 1 s/d 4</p> <p>b. Mengerjakan tugas, baik tugas mandiri maupun tugas kelompok dengan materi yang berisi pokok – pokok bahan kajian dari pertemuan 1 s/d 4 ;</p> <p>c. Menguasai dan mampu menjelaskan hasil</p>						
---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--	--	--

	<p>kerja tugasnya;</p> <p>d. Menjelaskan konsep dasar komponen – komponen simetri pada sistem rangkaian listrik;</p> <p>e. Berkomunikasi dalam memberikan informasi mengenai hasil tugasnya;</p> <p>Mempertanggungjawabkan terhadap hasil kerja tugasnya</p>						
6	<p>Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat :</p> <p>Membuat transfer function.</p> <p>Menghitung output respons untuk</p>						

	sinusoidal input . Mahasiswa dapat menghitung transient respons menggunakan pole- zero.						
7	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan bentuk model blok diagram cascade, single blok diagram, dan menggambar kan output respons bila diberikan input impuls, sinusoida, dan unit step function						
8	Ujian Tengah Semester (UTS) Tes tertulis untuk materi I s/d VII						
9	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan perhitungan besar bentuk sinyal digital						
10	Setelah mahasiswa mengikuti kuliah						

	bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Discrete transfer function						
11	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan perhitungan besar arus dan tegangan, untuk komponen pasif reactance, dan capacitance.						
12	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Menjelaskan perhitungan besar arus dan tegangan AC.						
13	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa						

	diharapkan dapat : Membuat filter butterworth						
14	Mahasiswa dapat merealisasikan filter low pass, band pass, dan high pass menggunakan metoda chebyshev						
15	Setelah mahasiswa mengikuti kuliah bahan kajian ini secara tuntas, mahasiswa diharapkan dapat : Dapat membuat filter dengan menggunakan recursif, dan menggunakan cara chebishev						
16	Ujian Akhir Semester (UAS) Tes tertulis untuk materi IX s/d XV						

