

IIR AND FIR FILTERS DESIGN

MATLAB code & Analysis

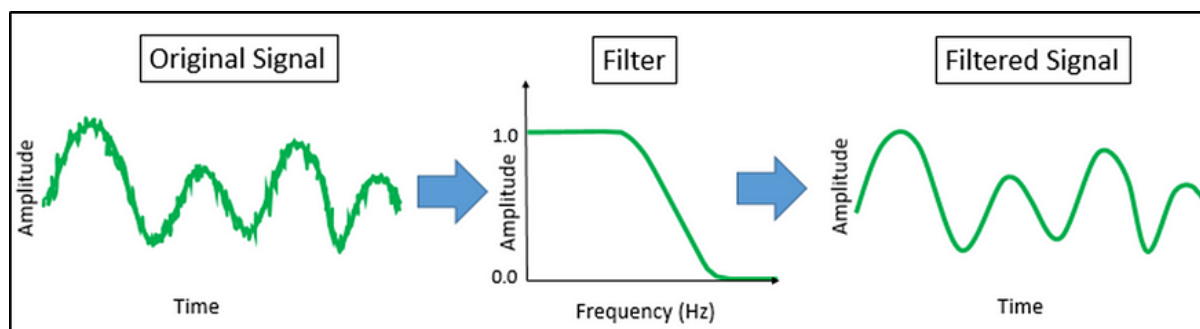


Nariman Ziaie
June 2024

فهرست مطالب

صفحه	عنوان
	[فیلترهای دیجیتال و انواع آنها]
2	۱-۱ مقدمه
3	۱-۲ فیلترهای پاسخ ضربه نامحدود (IIR)
3	۱-۳ فیلترهای پاسخ ضربه محدود (FIR)
	[فیلترهای پاسخ ضربه نامحدود "IIR"]
4	۲-۱ تعریف کلی
5	۲-۲ انواع فیلترهای IIR
	[فیلترهای پاسخ ضربه محدود "FIR"]
7	۳-۱ تعریف کلی
8	۳-۲ انواع فیلترهای FIR
	[برنامه نویسی و کدهای متلب]
9	۴-۱ بررسی کدهای مربوط به فیلتر "IIR" و "FIR"
11	۴-۱ چند قدم فراتر
14	[منابع]

فیلترهای دیجیتال و انواع آنها



۱-۱ مقدمه

فیلترهای دیجیتال ابزارهای اساسی در پردازش سیگنال‌های دیجیتال هستند که به منظور تغییر یا استخراج اطلاعات از سیگنال‌ها به کار می‌روند. این فیلترها نقش مهمی در طیف گسترده‌ای از کاربردها، از جمله پردازش صوت، تصویر، و ارتباطات دیجیتال دارند. با استفاده از فیلترهای دیجیتال، می‌توان نویز را کاهش داد، سیگنال‌ها را تقویت کرد، و ویژگی‌های خاصی را از داده‌ها استخراج نمود.

فیلترهای دیجیتال به دو دسته اصلی تقسیم می‌شوند:

۱. فیلترهای پاسخ ضربه نامحدود (IIR)

۲. فیلترهای پاسخ ضربه محدود (FIR)

هر کدام از این دو نوع فیلتر، ویژگی‌ها و کاربردهای خاص خود را دارند.

۱-۲ فیلترهای پاسخ ضربه نامحدود (IIR)

فیلترهای IIR دارای پاسخ ضربه‌ای هستند که به صورت نظری به طور نامحدود ادامه دارد. این فیلترها معمولاً از الگوریتم‌های بازگشتی برای پردازش سیگنال استفاده می‌کنند، به این معنی که خروجی آنها به خروجی‌های قبلی وابسته است. فیلترهای IIR به دلیل کارایی محاسباتی بالاتر و نیاز به تعداد کمتری از ضرایب، به ویژه در کاربردهای زمان واقعی، محبوب هستند. از جمله مهم‌ترین انواع فیلترهای IIR می‌توان به فیلترهای Butterworth، Chebyshev، Bessel و Elliptic اشاره کرد. هر کدام از این فیلترها ویژگی‌های خاصی دارند که آنها را برای کاربردهای مختلف مناسب می‌سازد.

۱-۳ فیلترهای پاسخ ضربه محدود (FIR)

فیلترهای FIR نیز دارای پاسخ ضربه‌ای هستند که در یک تعداد محدود گام به صفر می‌رسد. این فیلترها به دلیل پایداری ذاتی و قابلیت طراحی با پاسخ فاز خطی، مورد توجه قرار می‌گیرند. فیلترهای FIR برای کاربردهایی که نیاز به حداقل اعوجاج فاز دارند، مانند پردازش صوت و تصویر، بسیار مناسب هستند. روش‌های مختلفی برای طراحی فیلترهای FIR وجود دارد، از جمله روش‌های پنجره‌ای، الگوریتم Parks-McClellan و روش‌های کمینه‌سازی خطا. این روش‌ها به طراحان اجازه می‌دهند تا فیلترهایی با ویژگی‌های دقیق و کنترل شده ایجاد کنند.

در مجموع، انتخاب بین فیلترهای IIR و FIR به نیازهای خاص کاربرد بستگی دارد. در حالی که فیلترهای IIR برای کاربردهایی که نیاز به کارایی محاسباتی بالا دارند مناسب هستند، فیلترهای FIR برای کاربردهایی که نیاز به پاسخ فاز خطی و پایداری دارند ترجیح داده می‌شوند.

شناخت دقیق ویژگی‌ها و محدودیت‌های هر کدام از این فیلترها به طراحان کمک می‌کند تا بهترین انتخاب را برای نیازهای خود داشته باشند.

فیلترهای پاسخ ضربه نامحدود “IIR”

$$y[n] = - \sum_{k=1}^M a_k y[n-k] + \sum_{k=0}^N b_k x[n-k]$$

$y[n]$ is the output.

$x[n]$ is the input.

a_k and b_k are the filter coefficients.

M is the order of the feedback part (denominator).

N is the order of the feedforward part (numerator).

۲-۱ تعریف کلی

فیلترهای IIR به دلیل داشتن پاسخ ضربه‌ای که به صورت نظری به طور نامحدود ادامه دارد، شناخته می‌شوند. این فیلترها معمولاً از بازگشتی‌ها برای پردازش سیگنال استفاده می‌کنند، به این معنی که خروجی آنها به خروجی‌های قبلی وابسته است.

فیلترهای IIR به طور گسترده‌ای در کاربردهای زمان واقعی به دلیل کارایی محاسباتی بالاتر و نیاز به تعداد کمتری از ضرایب استفاده می‌شوند.

۲-۲ انواع فیلترهای IIR

• فیلترهای Butterworth

خصوصیات: این فیلترها دارای پاسخ دامنه‌ای کاملاً صاف در باند عبور هستند و هیچ گونه رپیلی در باند عبور ندارند. هر چه مرتبه فیلتر بالاتر باشد، شیب باند گذر تیزتر خواهد بود.

کاربردها: مناسب برای کاربردهایی که نیاز به پاسخ باند عبور صاف دارند، مانند پردازش صوتی.

• فیلترهای Chebyshev

خصوصیات نوع I: این نوع فیلترها رپیل در باند عبور دارند که منجر به شیب باند گذر تیزتر می‌شود.

خصوصیات نوع II: این نوع فیلترها رپیل در باند توقف دارند و باند عبور صاف‌تری ارائه می‌دهند.

کاربردها: مناسب برای سیستم‌های ارتباطی که نیاز به شیب باند گذر تیزتر دارند.

• فیلترهای Bessel

خصوصیات: این فیلترها دارای پاسخ فاز خطی هستند که منجر به حداقل اعوجاج فاز می‌شود. شیب باند گذر آنها نسبت به فیلترهای Butterworth و Chebyshev کندتر است.

کاربردها: استفاده در کاربردهایی که خطی بودن فاز مهم است، مانند کراس‌اوورهای صوتی و ارتباطات داده‌ای.

• فیلترهای Elliptic (Cauer)

خصوصیات: این فیلترها تیزترین شیب باند گذر را برای یک مرتبه مشخص دارند و دارای رپیل در باند عبور و باند توقف هستند. طراحی آنها پیچیده‌تر است ولی کارایی بالایی دارند.

کاربردها: استفاده در کاربردهایی که نیاز به حداقل باند گذر دارند، مانند ارتباطات باند باریک.

- **فیلترهای Comb**

خصوصیات: این فیلترها دارای پاسخ فرکانسی با لبه‌های فاصله‌دار هستند و ساختار ساده‌ای دارند.
کاربردها: حذف اکو و حذف هارمونیک‌ها در پردازش صوت.

- **فیلترهای Notch (Band-Stop)**

خصوصیات: این فیلترها برای حذف یک باند فرکانسی باریک طراحی شده‌اند.
کاربردها: حذف مؤلفه‌های فرکانسی خاص، مانند نویز خطوط برق.

- **فیلترهای All-Pass**

خصوصیات: این فیلترها همه مؤلفه‌های فرکانسی را به طور مساوی عبور می‌دهند ولی روابط فاز را تغییر می‌دهند.
کاربردها: تصحیح فاز و هم‌فازی.

- **فیلترهای State Variable**

خصوصیات: این فیلترها می‌توانند پاسخ‌های فیلتر متعدد از یک ساختار تولید کنند.
کاربردها: اکولایزرهای صوتی که نیاز به پاسخ‌های فیلتر متعدد دارند.

- **فیلترهای Ladder و Lattice**

خصوصیات: این فیلترها بر اساس ساختارهای شبکه‌ای یا نردبانی هستند.
کاربردها: فیلترهای تطبیقی و پیش‌بینی خطی در سیستم‌های ارتباطات دیجیتال.

فیلترهای پاسخ ضربه محدود “FIR”

$$y[n] = \sum_{k=0}^N b_k x[n - k]$$

$y[n]$ is the output.

$x[n]$ is the input.

b_k is the filter coefficients.

N is the order of the filter, which is the number of taps minus one.

۳-۱ تعریف کلی

فیلترهای FIR به خاطر پاسخ ضربه‌ای که در یک تعداد محدود گام به صفر می‌رسد، شناخته می‌شوند. این فیلترها ذاتاً پایدار هستند و می‌توانند با پاسخ فاز خطی طراحی شوند.

فیلترهای FIR برای کاربردهایی که نیاز به حداقل اعوجاج فاز دارند، مانند پردازش صوت و تصویر، بسیار مناسب هستند.

۳-۲ انواع فیلترهای FIR

• فیلترهای Windowed-Sinc

خصوصیات: این فیلترها از تابع پنجره برای قطع پاسخ ضربه ایده آل استفاده می کنند. پنجره های معمول شامل مستطیلی، Hamming، Hanning، Blackman، و Kaiser هستند. کاربردها: طراحی های ساده و آسان برای استفاده کلی.

• فیلترهای Parks-McClellan (Equiripple)

خصوصیات: این فیلترها بیشینه خطا بین پاسخ های فرکانسی مطلوب و واقعی را به حداقل می رسانند و رفتار equiripple در باند عبور و توقف ارائه می دهند. کاربردها: کنترل دقیق بر پاسخ فرکانسی در سیستم های ارتباطی.

• فیلترهای Least Squares

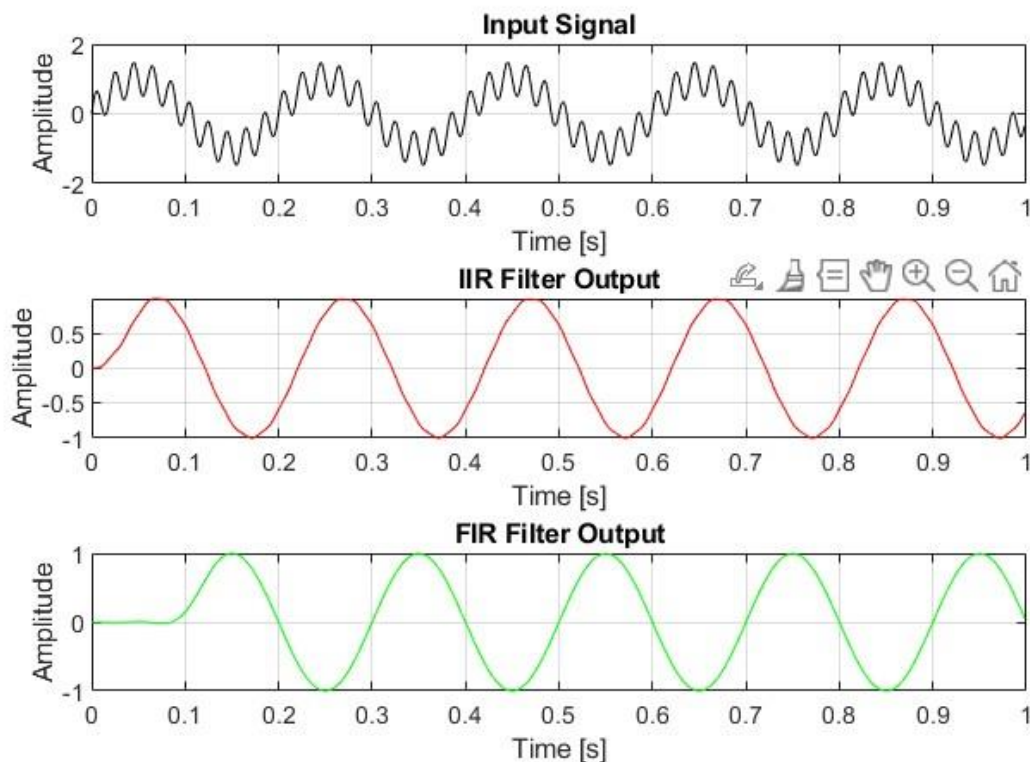
خصوصیات: این فیلترها خطای میانگین مربعی بین پاسخ های مطلوب و واقعی را به حداقل می رسانند و ریبیل کمتر در باند عبور و توقف دارند ولی باند گذر گسترده تری دارند. کاربردها: کاربردهایی که کمینه سازی خطای کلی اولویت دارد.

• فیلترهای Frequency Sampling

خصوصیات: این فیلترها با نمونه برداری از پاسخ فرکانسی مطلوب در نقاط خاص طراحی می شوند. کاربردها: پاسخ های فرکانسی سفارشی برای کاربردهای خاص.

برنامه نویسی و کدهای متلب

Input Signal and Filter Outputs



۴-۱ بررسی کدهای مربوط به فیلتر “FIR” و “IIR”

در اینجا ما با تعریف سیگنال ورودی که ترکیبی از دو موج سینوسی با فرکانس‌های ۵ هرتز و ۵۰ هرتز است و فرکانس نمونه‌برداری f_s که برابر با ۵۰۰ هرتز تنظیم شده است، می‌خواهیم یک فیلتر IIR پایین‌گذر Butterworth و یک فیلتر FIR پایین‌گذر با استفاده از پنجره Hamming طراحی کنیم.

در ادامه کدهای متلب مربوطه را مشاهده خواهید کرد:

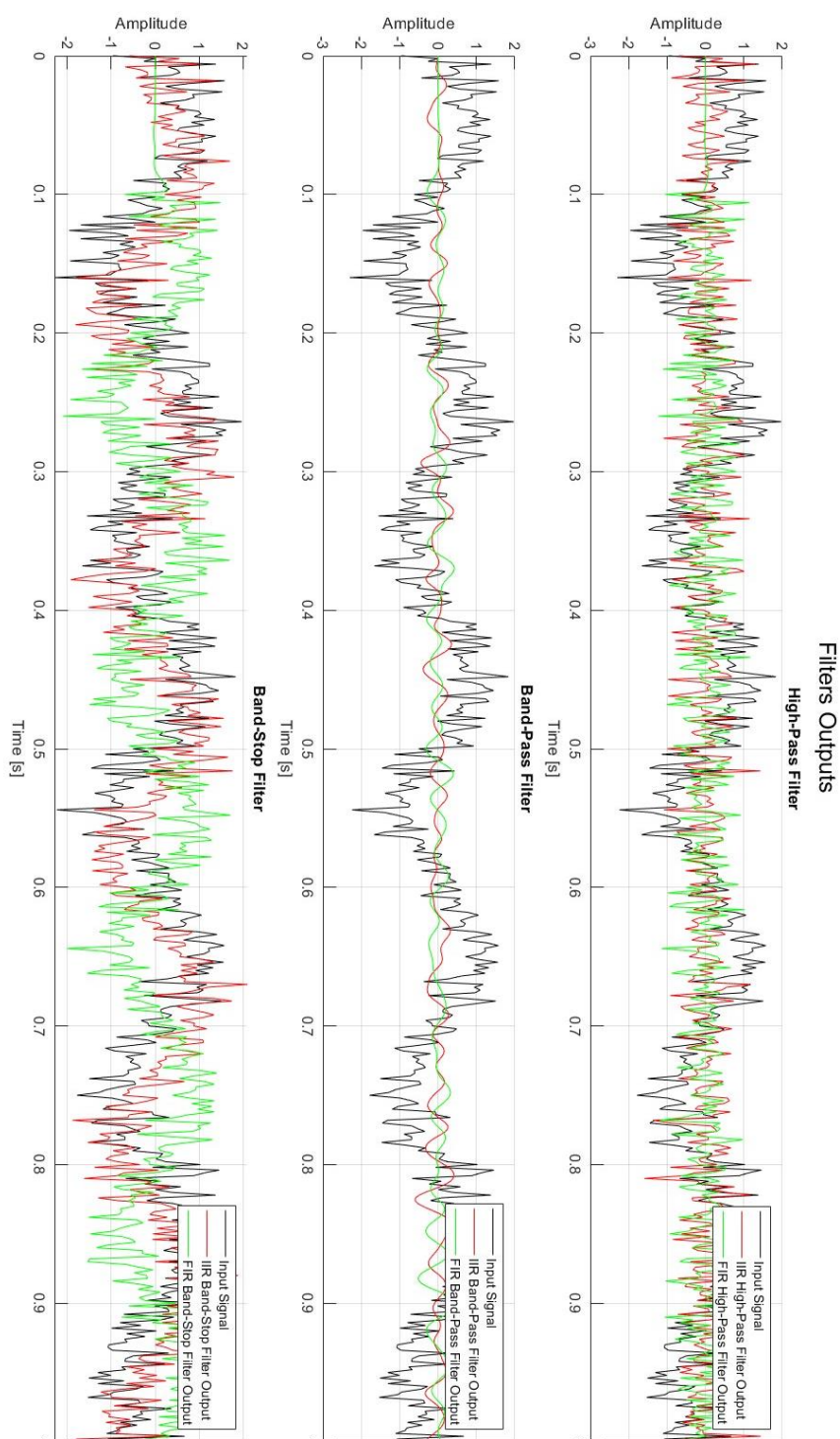
```

1. % Define the sampling frequency and time vector
2. fs = 500; % Sampling frequency in Hz
3. t = 0:1/fs:1-1/fs; % Time vector from 0 to 1 second with a step of 1/fs.
4.
5. % Create a sample input signal
6. % Combination of two sine waves with frequencies 5 Hz and 50 Hz.
7. input_signal = sin(2 * pi * 5 * t) + 0.5 * sin(2 * pi * 50 * t);
8.
9. % Design an IIR Butterworth filter
10. order = 4; % A 4th - order Butterworth low - pass filter
11. cutoff = 20; % Cutoff frequency of 20 Hz
12. [b_iir,a_iir] = butter(order,cutoff/(fs/2),'low');
13.
14. % Apply the IIR filter to the input signal
15. iir_output = filter(b_iir,a_iir,input_signal);
16.
17. % Design an FIR filter using the Hamming window method
18. numtaps = 101; % Number of taps in FIR filter
19. b_fir = fir1(numtaps-1,cutoff/(fs/2),'low',hamming(numtaps));
20.
21. % Apply the FIR filter to the input signal
22. fir_output = filter(b_fir,1,input_signal);
23.
24. % Plot the input and output signals in separate subplots
25. figure;
26.
27. % Input signal
28. subplot(3,1,1);
29. plot(t,input_signal,'k','DisplayName','Input Signal'); % Input signal in black
30. title('Input Signal');
31. xlabel('Time [s]');
32. ylabel('Amplitude');
33. grid on;
34.
35. % IIR filter output
36. subplot(3,1,2);
37. plot(t,iir_output,'r','DisplayName','IIR Filter Output'); % IIR output in red
38. title('IIR Filter Output');
39. xlabel('Time [s]');
40. ylabel('Amplitude');
41. grid on;
42.
43. % FIR filter output
44. subplot(3,1,3);
45. plot(t,fir_output,'g','DisplayName','FIR Filter Output'); % FIR output in green
46. title('FIR Filter Output');
47. xlabel('Time [s]');
48. ylabel('Amplitude');
49. grid on;
50.
51. % Adjust layout
52. sgtitle('Input Signal and Filter Outputs');

```

۴-۲ چند قدم فراتر

در ادامه چند گام فراتر میگذاریم و به بررسی کد بالا برای نویز تصادفی و فیلترهای “high-pass, band-pass, band-stop” میپردازیم.



```

1. % Define the sampling frequency and time vector
2. fs = 500; % Sampling frequency in Hz
3. t = 0:1/fs:1-1/fs; % Time vector
4.
5. % Create a sample input signal: a sine wave + random noise
6. input_signal = sin(2 * pi * 5 * t) + 0.5 * randn(size(t));
7.
8. % ----- Filters Variables ----- %
9. order = 4; % 4th - order Butterworth IIR filter
10. numtaps = 101; % Number of taps in FIR filter
11.
12. % ----- High - Pass ----- %
13. % Design an IIR Butterworth high - pass filter
14. cutoff = 20; % Cutoff frequency in Hz
15. [b_iir_hp,a_iir_hp] = butter(order,cutoff/(fs/2),'high');
16.
17. % Apply the IIR high - pass filter to the input signal
18. iir_hp_output = filter(b_iir_hp,a_iir_hp,input_signal);
19.
20. % Design an FIR low - pass filter using the Hamming window method
21. b_fir_hp = fir1(numtaps-1,cutoff/(fs/2),'high',hamming(numtaps));
22.
23. % Apply the FIR low - pass filter to the input signal
24. fir_hp_output = filter(b_fir_hp,1,input_signal);
25.
26. % ----- Band - Pass ----- %
27. % Design an IIR Butterworth band - pass filter
28. low_cutoff = 10; % Lower cutoff frequency in Hz
29. high_cutoff = 40; % Upper cutoff frequency in Hz
30. [b_iir_bp,a_iir_bp] = butter(order,[low_cutoff high_cutoff]/(fs/2),'bandpass');
31.
32. % Apply the IIR band - pass filter to the input signal
33. iir_bp_output = filter(b_iir_bp,a_iir_bp,input_signal);
34.
35. % Design an FIR band - pass filter using the Hamming window method
36. b_fir_bp = fir1(numtaps-1,[low_cutoff high_cutoff]/(fs/2),'bandpass',hamming(numtaps));
37.
38. % Apply the FIR band - pass filter to the input signal
39. fir_bp_output = filter(b_fir_bp,1,input_signal);
40.
41. % ----- Band - Stop ----- %
42. % Design an IIR Butterworth band - stop filter
43. [b_iir_bs,a_iir_bs] = butter(order,[low_cutoff high_cutoff]/(fs/2),'stop');
44.
45. % Apply the IIR band - stop filter to the input signal
46. iir_bs_output = filter(b_iir_bs,a_iir_bs,input_signal);
47.
48. % Design an FIR band - stop filter using the Hamming window method
49. b_fir_bs = fir1(numtaps-1,[low_cutoff high_cutoff]/(fs/2),'stop',hamming(numtaps));
50.
51. % Apply the FIR band - stop filter to the input signal
52. fir_bs_output = filter(b_fir_bs,1,input_signal);
53.

```

```

54. % ----- Plotting ----- %
55. % Plot the input and output signals in separate subplots
56. figure;
57.
58. % High – Pass Filter
59. subplot(3,1,1);
60. hold on;
61. plot(t,input_signal,'k','DisplayName','Input Signal'); % Input signal in black
62. plot(t,iir_hp_output,'r','DisplayName','IIR High – Pass Filter Output'); % IIR output in red
63. plot(t,fir_hp_output,'g','DisplayName','FIR High –
    Pass Filter Output'); % FIR output in green
64. hold off;
65. title('High – Pass Filter');
66. xlabel('Time [s]');
67. ylabel('Amplitude');
68. grid on;
69. legend show;
70.
71. % Band – Pass Filter
72. subplot(3,1,2);
73. hold on;
74. plot(t,input_signal,'k','DisplayName','Input Signal'); % Input signal in black
75. plot(t,iir_bp_output,'r','DisplayName','IIR Band – Pass Filter Output'); % IIR output in red
76. plot(t,fir_bp_output,'g','DisplayName','FIR Band –
    Pass Filter Output'); % FIR output in green
77. hold off;
78. title('Band – Pass Filter');
79. xlabel('Time [s]');
80. ylabel('Amplitude');
81. grid on;
82. legend show;
83.
84. % Band – Stop Filter
85. subplot(3,1,3);
86. hold on;
87. plot(t,input_signal,'k','DisplayName','Input Signal'); % Input signal in black
88. plot(t,iir_bs_output,'r','DisplayName','IIR Band – Stop Filter Output'); % IIR output in red
89. plot(t,fir_bs_output,'g','DisplayName','FIR Band –
    Stop Filter Output'); % FIR output in green
90. hold off;
91. title('Band – Stop Filter');
92. xlabel('Time [s]');
93. ylabel('Amplitude');
94. grid on;
95. legend show;
96.
97. % Adjust layout
98. sgtitle('Filters Outputs');

```

منابع

- [MATLAB Documentation - IIR Filter Design](#)
- [MATLAB Documentation - FIR Filter Design](#)
- [National Instruments Documentation - IIR Filters and FIR Filters](#)

برای مشاهده فایل‌های پروژه روی تصویر زیر کلیک کنید یا آن را اسکن نمایید.

