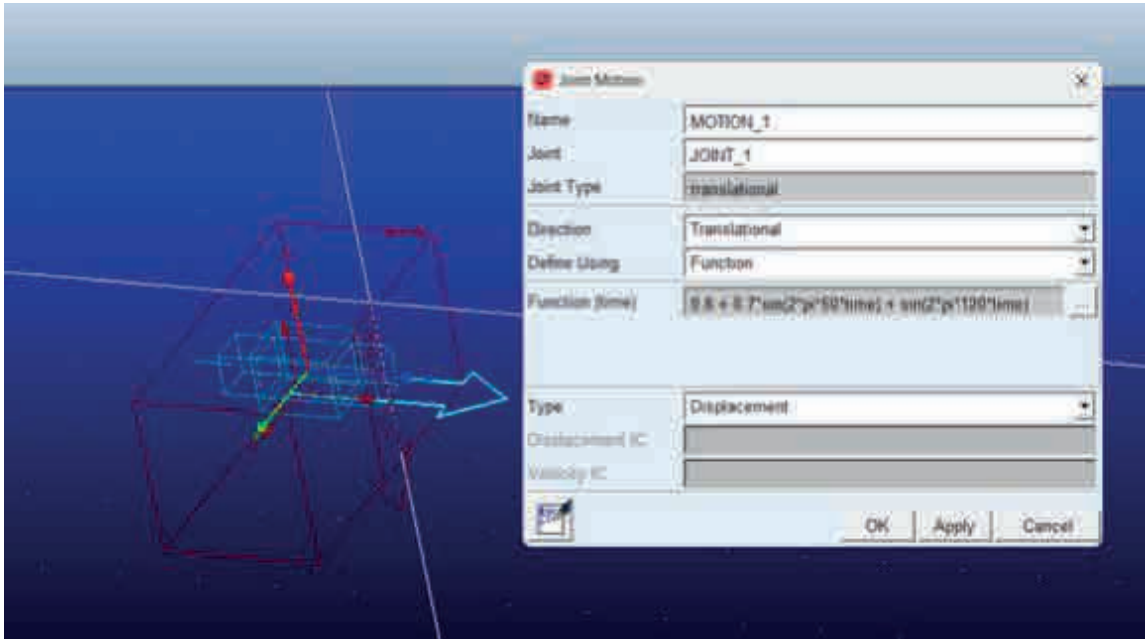


TEKNİK DOKÜMAN



Adams

Bu doküman, Adams View içerisindeki sinyal analizi-filtreleme araçlarını ele alarak FFT ve filtreleme uygulamalarının nasıl yapıldığını anlatmakla birlikte sinyal analizinin simülasyon süreçlerine nasıl yansıdığını aktarmaktadır.



BİAS
MÜHENDİSLİK



HEXAGON

Authorised Hexagon reseller

ADAMS VIEW İLE SİNYAL ANALİZİ VE FİLTRELEME

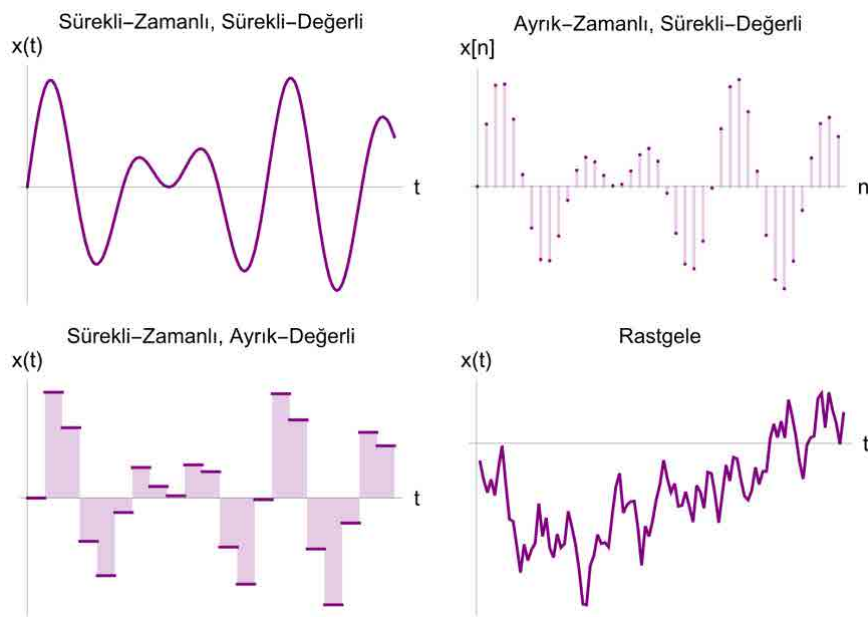
HAZIRLAYAN	ONAYLAYAN
Sefa Çiçek Mekanik Simülasyon Aday Mühendisi	Eren Morgil Kıdemli Mekanik Simülasyon Mühendisi

Tarih: 11/09/2025

1. GİRİŞ

Bilgi iletmek amacıyla kullanılan zamana bağlı herhangi bir fiziksel olay **sinyal** olarak adlandırılır. Sinyallere örnek olarak insan sesi, Mors alfabesi, deprem dalgaları, fotoğraf veya video görüntüleri verilebilir. **Gürültü** ise sinyal gibi zamana bağlı fiziksel bir olaydır; ancak genellikle faydalı bilgi taşımaz ve istenmeyen bir unsur olarak kabul edilir.

Sinyaller, birkaç genel sınıfa ayrılır. Bunlar; sürekli-zamanlı (*continuous-time*), ayrık-zamanlı (*discrete-time*), sürekli-değerli (*continuous-value*), ayrık-değerli (*discrete-value*), rastgele (*random*) ve rastgele olmayan (*non-random*) sinyallerdir. Şekil 1’de belirtilen sinyal türlerinin grafikleri görülmektedir.



Şekil 1 – Sinyal Türleri

Sürekli-zamanlı (continuous time) bir sinyal, belirli bir zaman aralığı boyunca zamanın her anında tanımlıdır. Bazı sürekli-zamanlı sinyaller için yaygın olarak kullanılan bir diğer ad analog sinyaldir. Tüm analog sinyaller sürekli-zamanlı sinyallerdir; ancak her sürekli-zamanlı sinyal analog sinyal değildir.

Bir sinyalin örneklenmesi, sürekli-zamanlı bir sinyalden belirli zaman noktalarında değer alınması işlemidir. Bu işlem sonucunda elde edilen örnekler kümesi, bir ayrık-zamanlı bir sinyal oluşturur. Ayrık-zamanlı bir sinyal, yalnızca belirli zamanlarda sinyal değerleri üreten yapısı gereği ayrık-zamanlı bir sistem tarafından da oluşturulabilir. Ayrık-zamanlı sinyaller genellikle dijital sinyaller olarak iletilir. Bu, ayrık-zamanlı bir sinyalin değerlerinin bazı kodlanmış biçimlerde rakamlar dizisi şeklinde iletilmesidir. “Dijital” terimi bazen yalnızca iki olası değeri olan ayrık-değerli sinyalleri tanımlamak için de kullanılır. Bu tür dijital sinyallerdeki rakamlar, sürekli-zamanlı sinyaller aracılığıyla iletilir.

Rastgele (random) bir sinyal tam olarak önceden tahmin edilemez ve herhangi bir kesin matematiksel fonksiyonla tanımlanamaz. Buna karşın, belirli (deterministik) sinyaller matematiksel olarak tanımlanabilir ve zaman içindeki davranışları önceden tahmin edilebilir. Gürültü (noise), rastgele sinyallerin en bilinen örneklerinden biridir.

Birçok sürekli-zamanlı sinyalin matematiksel olarak tanımlanması kolay değildir. Ancak bazı temel sinyal fonksiyonları (Impulse, Step, Real Exponential, Complex Exponential, Sinusoidal) yardımıyla ve konvolüsyon adı verilen işlem sayesinde, bu tür sinyaller matematiksel olarak özlü bir biçimde tanımlanabilir, analiz edilebilir ve işlenebilir. Matematiksel fonksiyonlarla tanımlanabilen sürekli-zamanlı sinyaller, Fourier dönüşümü aracılığıyla frekans düzlemi adı verilen başka bir alana dönüştürülebilir. Bu dönüşüm, sinyalin frekans bileşenlerinin analiz edilmesini sağlar.

2. FOURIER DÖNÜŞÜMÜ (FOURIER TRANSFORM)

Sinyal ve sistem analizinde yaygın bir durum, zamanla değişmeyen doğrusal bir sistemin (LTI – Linear Time Invariant) periyodik bir sinyalle uyarılmasıdır. Eğer bir LTI sistem, bir sinüzoidal sinyal ile uyarılırsa, sistemin tepkisi de yine bir sinüzoidal sinyal olur. Ancak bu tepki, genellikle aynı frekansta fakat farklı genlikte ve fazdadır.

Bir LTI sistem birden fazla sinyalin toplamıyla uyarıldığında, sistemin toplam tepkisi, her bir sinyale ayrı ayrı verilen tepkilerin toplamı şeklinde bulunur. Bu durum, süperpozisyon ilkesi olarak adlandırılır. Eğer bir sinyal sinüzoidallerin lineer kombinasyonu olarak ifade edilebiliyorsa, bu sinyalin bir LTI sistemden geçerken oluşturduğu çıktı, her bir sinüzoidal bileşene verilen tepkilerin toplamı şeklinde hesaplanabilir. Bu prensip, karmaşık sinyallerin analizinde büyük kolaylık sağlar.

Bir periyodik sinyalin, sinüzoidallerin lineer kombinasyonu şeklinde ifade edilmesine Fourier serisi denir. Fourier serisi, periyodik sinyalleri frekans bileşenleri cinsinden temsil etmenin bir yöntemidir. Periyodik olmayan, süreksiz ya da genel bir sürekli-zamanlı sinyalin frekans alanına dönüştürülmesi için ise Fourier dönüşümü kullanılır.

2.1. Fourier Serisi

Bir fonksiyonun **Fourier serisi**, bu fonksiyonun belirli bir aralıkta (a, b) tanımlı ve uygun koşulları sağlayan (örneğin süreklilik, periyodiklik) olması durumunda, sinüs ve kosinüs fonksiyonlarının lineer kombinasyonu şeklinde ifade edilmesidir. Bu temsilde kullanılan sinüs ve kosinüs fonksiyonları, temel frekansın tam sayı katları olacak şekilde düzenlenir. Yani, bu harmonik bileşenler, fonksiyonun **frekans içeriğini** yansıtır.

Fourier analizi, $f(x)$ fonksiyonu periyodik ve parça parça düzgün (piecewise smooth) ise, bu fonksiyonun artan frekanslı kosinüs ve sinüs terimlerinin sonsuz toplamı olan bir Fourier serisi şeklinde yazılabileceğini belirtir. Özellikle, eğer $f(x)$ 2π -periyodik ise, şu şekilde yazılabilir:

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

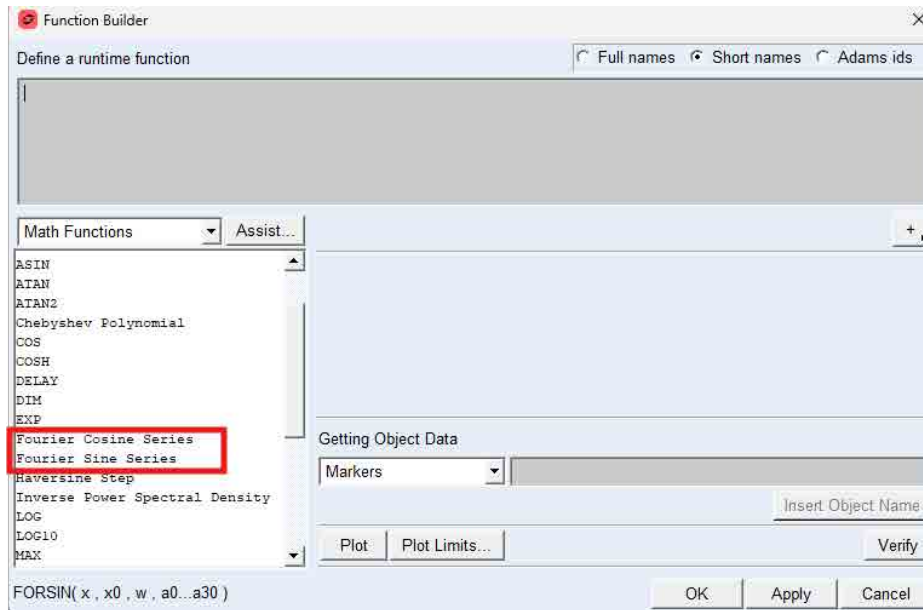
Buradaki a_0, a_n ve b_n katsayılarının değerleri aşağıdaki eşitlikler ile hesaplanır:

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx,$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \cos(nx) f(x) dx,$$

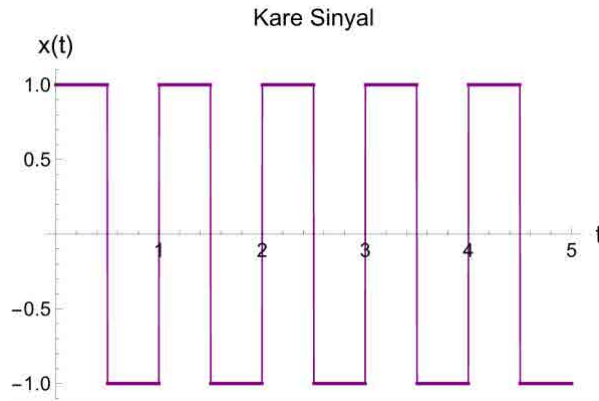
$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \sin(nx) f(x) dx.$$

Adams View programında “Function Builder” sekmesi altında Fourier serileri yardımıyla matematiksel olarak tanımlı, çok karmaşık fonksiyonları üretilebilir ve herhangi bir sisteme girdi olarak (kuvvet, deplasman, hız vb.) tanımlanabilir.



Şekil 2 – Fourier Serisi Fonksiyonları

Örnek uygulama olarak Adams View programında Fourier serisi ile kare fonksiyonu oluşturulup sisteme deplasman girdisi olarak tanımlanmıştır. Kare fonksiyonunun grafiği aşağıdaki gibidir.



Şekil 3 – Kare Fonksiyonu Sinyali

Kare fonksiyonun Fourier seri açılımı ise aşağıdaki denklemde verilmiştir.

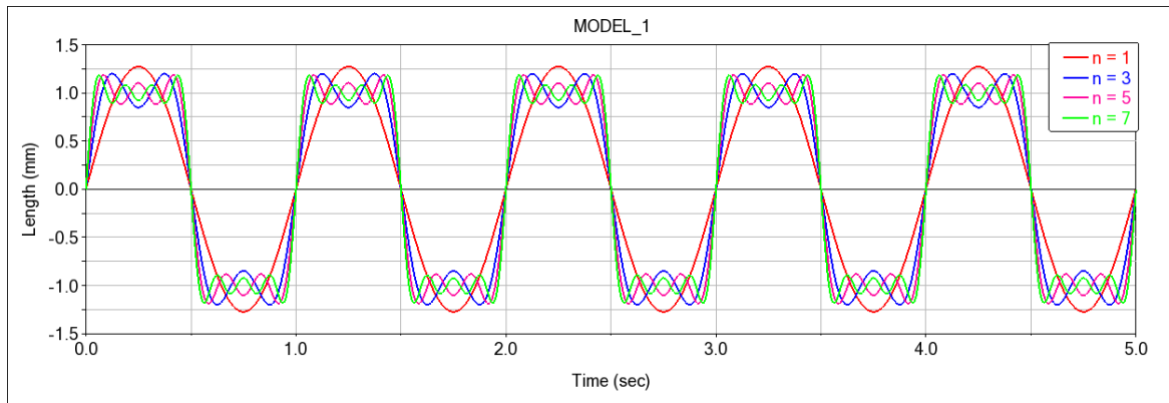
$$f(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4}{n\pi} \sin(n\omega_0 t)$$

Bu fonksiyon incelendiğinde sadece sinus fonksiyonu bulunmaktadır. Adams View’ de bulunan Fourier Sine Series (FORSIN) fonksiyon yapısı kullanılarak kare sinyali oluşturulabilir. FORSIN fonksiyon formatı FORSIN(x, Shift, Frequency, Coefficients) şeklindedir. Bu fonksiyona girilen parametreler aşağıdaki gibidir:

- **x** : Bağımsız değişken,
- **Shift** : Fourier seride bulunan faz açısı,
- **Frequency** : Fourier serisinin temel frekansı,
- **Coefficients** : Fourier seriyi tanımlayan katsayılar.

$n = 7$ ve $\omega = 2\pi$ için oluşturulması gereken fonksiyon;

“FORSIN(TIME, 0, 360D, 0, 4/pi, 0, 4/(3*pi), 0, 4/(5*pi), 0, 4/(7*pi))” şeklindedir. Aşağıdaki görselde n 0’dan, 7’ye kadar olan Fourier seri açılımlarının sırasıyla grafikleri görülmektedir.



Şekil 4 – Kare Sinyali Fourier Serisi Grafikleri

Grafikten görüldüğü gibi n değeri büyüdükçe fonksiyon gerçek değerine yaklaşmaktadır. Ayrıca “ n ” değeri arttıkça sinüs sinyallerinin frekansı artmaktadır.

2.2. Fourier Dönüşümü

Fourier serisi, periyodik fonksiyonlar için tanımlıdır. Bu nedenle tanımlı olduğu aralığın dışındaki bölgelerde fonksiyonun kendini sonsuz kez tekrar ettiği varsayılır. Fourier dönüşüm integrali ise, temel olarak Fourier serisinin tanım aralığının uzunluğu sonsuza giderken elde edilen limitidir. Bu sayede, $(-\infty, \infty)$ aralığında tanımlı ve kendini tekrar etmeyen bir fonksiyon tanımlamak mümkün olur. Burada Fourier serisini $x \in [-L, L]$ aralığında ele alınmıştır ve ardından $L \rightarrow \infty$ limitini alınmıştır. Bu tanım aralığında, Fourier serisi şu şekildedir:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right) \right] = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x/L}$$

Burada,

$$c_n = \frac{1}{2L} \int_{-L}^L f(x) e^{-in\pi x/L} dx$$

şeklinde dir.

Önceki sonuçlar yeniden ifade edilirse, $f(x)$ artık $\omega_n = \frac{n\pi}{L}$ ile verilen ayrık bir frekans kümesine sahip sinüs ve kosinüslerin toplamı şeklinde temsil edilmektedir. $L \rightarrow \infty$ limiti alındığında, bu ayrık frekanslar sürekli bir frekans aralığına dönüşür.

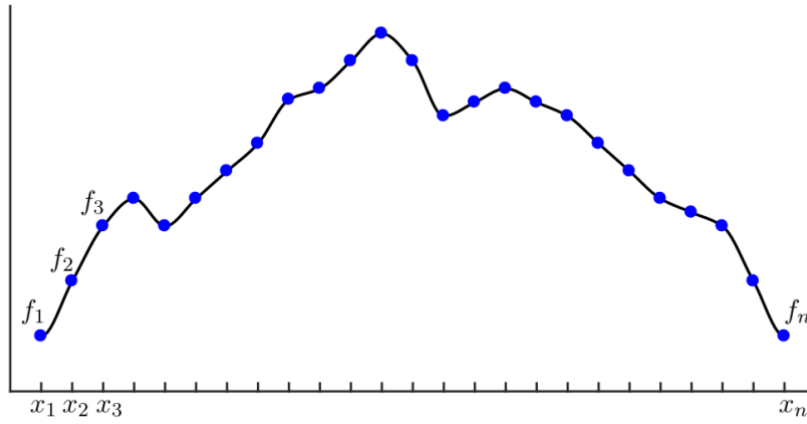
Gerekli işlemler yapıldığında $f(x)$ fonksiyonun Fourier dönüşümü aşağıda verilen eşitlik ile ifade edilir.

$$Fourier[f(x)] = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

Fourier dönüşümü, özellikle bazı özellikleri nedeniyle oldukça kullanışlıdır; bunlar arasında doğrusallık (lineerlik) özelliği ve fonksiyonların türevlerinin Fourier dönüşüm uzayında nasıl davrandığı yer alır.

2.3. Ayrık Fourier Dönüşümü (DTF – Discrete Fourier Transform)

Fourier serileri ve Fourier dönüşümü, $f(x)$ sürekli fonksiyonları içindir. Ancak, gerçek verilerle hesaplama yapılırken veya çalışılırken, Fourier dönüşümünü veri kümeleri üzerinde yaklaşık olarak hesaplamak gerekir. Ortaya çıkan ayrık Fourier dönüşümü (DFT), esasen, Şekil 5’te gösterildiği gibi $f(x)$ fonksiyonunun düzenli bir aralıkla (Δx) ayrıklaştırılmasıyla elde edilen $\mathbf{f} = [f_1 \ f_2 \ f_3 \ \dots \ f_n]^T$ veri vektörleri için Fourier serisinin ayrıklaştırılmış bir versiyonudur.



Şekil 5 – Ayırık Fourier Dönüşümü İçin Ayırık Veri Kümesi

Ayrık Fourier dönüşümü aşağıdaki formülasyon ile tanımlanır:

$$\hat{f}_k = \sum_{j=0}^{n-1} f_j e^{-i2\pi jk/n}$$

Ters ayrık Fourier dönüşümü (IDFT) ise şu şekilde tanımlanır:

$$f_k = \frac{1}{n} \sum_{j=0}^{n-1} \hat{f}_j e^{i2\pi jk/n}$$

Böylece, DFT bir lineer bir matristir ve f içindeki veri noktalarını frekans düzlemine, yani $\hat{f}$ 'ye dönüştürür:

$$\{f_1, f_2, \dots, f_n\} \xrightarrow{DFT} \{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_n\}$$

Belirli bir nokta sayısı n için, DFT veriyi temel bir frekansın tamsayı katları olan sinüs ve kosinüs fonksiyonlarıyla ifade eder. Bu temel frekans $\omega_n = e^{-2\pi i/n}$ şeklindedir. DFT, matris çarpımı yoluyla hesaplanabilir:

$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \\ \vdots \\ \hat{f}_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix}$$

Çıktı vektörü $\hat{f}$, giriş vektörü f için Fourier katsayılarını içerir ve DFT matrisi F , $n \times n$ boyutunda bir matrisidir. Matris F karmaşık değerli olduğundan, $\hat{f}$ çıktısı hem genlik hem de faz içerir; bu iki bileşenin de fiziksel olarak anlamlı yorumları vardır.

2.4. Fast Fourier Transform (FFT)

Ayrık Fourier dönüşümü (DFT), sayısal yaklaşım ve hesaplama açısından son derece kullanışlıdır; ancak çok büyük $n \gg 1$ değerleri için iyi ölçeklenmez. Çünkü bu basit formülasyon, yoğun bir $n \times n$ matris ile çarpım gerektirir ve bu da $O(n^2)$ işlem maliyeti anlamına gelir.

1965 yılında James W. Cooley (IBM) ve John W. Tukey (Princeton), günümüzde son derece önemli bir algoritma olan Hızlı Fourier Dönüşümünü (FFT) geliştirmişlerdir. Bu algoritma $O(n \log(n))$ karmaşıklığıyla çalışır. n çok büyük hale geldikçe, $\log(n)$ bileşeni yavaş artar ve algoritma neredeyse doğrusal bir ölçeklenmeye yaklaşır. Geliştirdikleri algoritma, Fourier dönüşümündeki fraktal bir simetriye dayanır ve bu simetri sayesinde, $n \times n$ boyutlu bir DFT, daha küçük boyutlu birçok DFT hesaplamasıyla çözülebilir. DFT ile FFT arasındaki bu farklı işlem karmaşıklığı küçük gibi görünse de, hızlı $O(n \log(n))$ ölçeklenme, FFT'nin gerçek zamanlı iletişimde yaygın bir şekilde kullanılabilmesini mümkün kılmıştır.

FFT'nin (Hızlı Fourier Dönüşümü) arkasındaki temel fikir, eğer veri noktalarının sayısı n , 2'nin bir kuvveti ise, DFT'nin (Ayrık Fourier Dönüşümü) çok daha verimli bir şekilde uygulanabileceğidir. Örneğin, $n = 1024 = 2^{10}$ olduğunu düşünelim. Bu durumda, DFT matrisi F_{1024} şu şekilde yazılabilir:

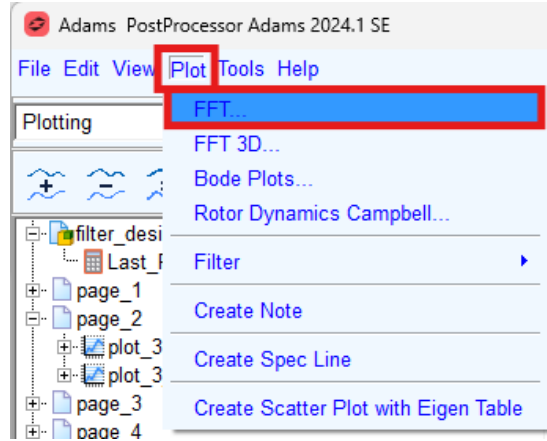
$$\hat{f} = F_{1024} f = \begin{bmatrix} I_{512} & -D_{512} \\ I_{512} & -D_{512} \end{bmatrix} \begin{bmatrix} F_{512} & 0 \\ 0 & F_{512} \end{bmatrix} \begin{bmatrix} f_{even} \\ f_{odd} \end{bmatrix}$$

Burada f_{even} , f vektörünün çift indeksli elemanlarını; f_{odd} , f vektörünün tek indeksli elemanlarını; I_{512} , 512×512 boyutunda birim matrisi temsil eder. D_{512} matrisi ise aşağıdaki gibidir:

$$D_{512} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \omega & 0 & \dots & 0 \\ 0 & 0 & \omega^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \omega^{511} \end{bmatrix}$$

Bu ifade, ayrık Fourier dönüşümünde kullanılan denklemlerdeki terimlerin dikkatli bir şekilde hesaplanması ve yeniden düzenlenmesiyle elde edilebilir. Eğer $n = 2^p$ ise, bu işlem tekrarlanabilir ve F_{512} , önce F_{256} ile, ardından sırasıyla $F_{128} \rightarrow F_{64} \rightarrow F_{32} \rightarrow \dots$ şeklinde temsil edilebilir. Eğer $n \neq 2^p$ ise, vektör, boyutu 2'nin kuvveti olana kadar sıfırlarla doldurulabilir. FFT (Hızlı Fourier Dönüşümü), daha sonra vektör f 'nin alt vektörlerinin çift ve tek indekslerinin verimli bir şekilde iç içe yerleştirilmesini ve birçok küçük 2×2 boyutlu DFT hesaplamasının yapılmasını içerir.

Adams View programında bir sinyalin FFT grafiği, Adams PostProcessor arayüzünde 'Plot' sekmesi altına bulunan 'FFT' seçeneği kullanılarak çizilebilir. Adams PostProcessor ile FFT grafiği çizimine ilişkin tüm ayrıntılı bilgilere, referanslar kısmında bulunan [5] numaralı kaynaktan ulaşılabilir.

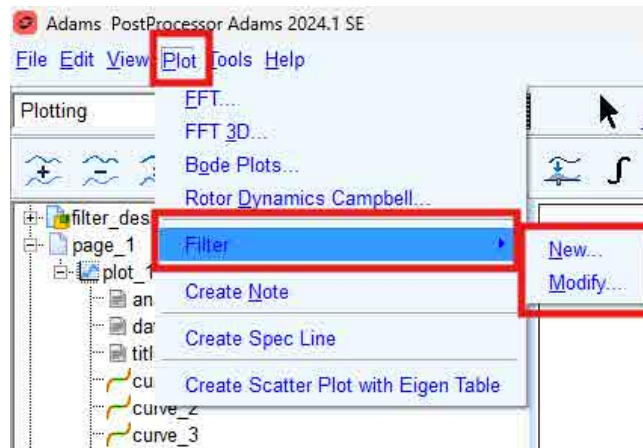


Şekil 6 – FFT Grafiği Çizimi

3. FİLTRELEME

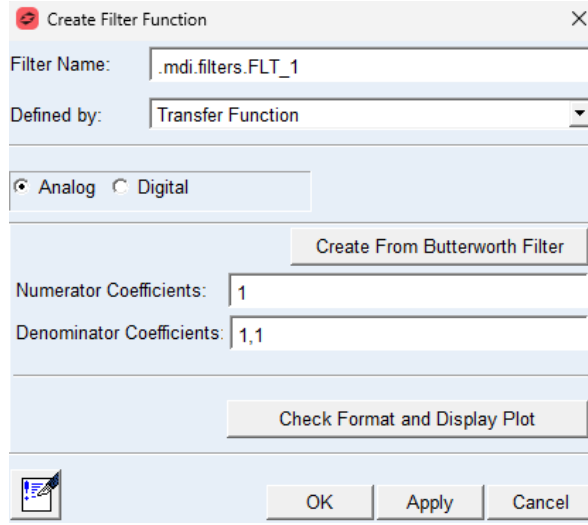
Filtreleme yapmaktaki ana amaç, bir sistemin frekans cevabını şekillendirmektir. Sinyal ve sistem analizinde, bir filtre; bir sinyalin istenilen kısmını istenmeyen kısmından ayırır. Filtre, sinyal ve sistem analizinde geleneksel olarak, bir frekans aralığındaki sinyalin gücünü vurgularken başka bir frekans aralığındaki gücü azaltan bir düzenek olarak tanımlanır.

Adams PostProcessor arayüzünde bulunan 'Plot' sekmesi altındaki 'Filter' seçeneği kullanılarak yeni bir filtre oluşturulabilir (New) veya daha önce oluşturulmuş bir filtre düzenleyebilir (Modify).



Şekil 7 – Filtre Oluşturma ve Düzenleme

Yeni bir filtre oluşturulduğunda ekranda Şekil 8’de gösterildiği gibi pencere açılacaktır.



Şekil 8 – Yeni Filtre Oluşturulması

Şekil 8’de gösterilen “Filter Name” kısmına, oluşturulan filtrenin ismi yazılır. “Defined by” kısmında ise ‘Transfer Function’ ve ‘Butterworth Filter’ olmak üzere iki seçenek bulunmaktadır. Bu filtre yapıları sırasıyla incelenmiştir.

Adams PostProcessor, Fast Fourier Transform (FFT) için iki filtreleme yöntemi sunar:

- **Analog filtering** - Analog filtreleme için sayısal işlem prosedürü şu işlemlere denktir:
 - Zaman sinyali Fast Fourier Transform (FFT) ile frekans bölgesine dönüştürmek.
 - Elde edilen fonksiyonu filtre fonksiyonu ile çarpmak.
 - Ters Fast Fourier Transform (IFFT) uygulamak.
- **Digital filtering** - Dijital filtreleme, doğrudan zaman sinyali üzerinde çalışır. Belirli bir zaman adımıdaki filtrelenmiş sinyal, önceki giriş ve çıkış sinyallerinin lineer bir kombinasyonudur. Bu kombinasyondaki katsayılar ayrık geçiş fonksiyonu tarafından tanımlanır.

Bu teknik doküman, analog filtreler özelinde hazırlanmıştır ve yapılan hesaplamalar analog filtreleme yöntemine dayanmaktadır.

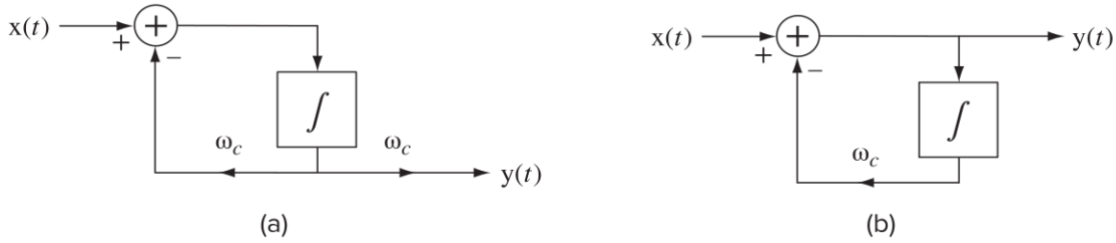
3.1. Transfer Fonksiyonu (Transfer Function)

Filtrelerin geçiş bantları (passband) ve durdurma bantları (stopband) vardır. Geçiş bandı, filtrenin sinyal gücünün nispeten etkilenmeden geçmesine izin verdiği bir frekans aralığıdır. Durdurma bandı ise, filtrenin sinyal gücünü önemli ölçüde zayıflattığı ve yalnızca çok azının geçmesine izin verdiği bir frekans aralığıdır.

Dört temel filtre türü vardır: alçak geçiren (lowpass), yüksek geçiren (highpass), bant geçiren (bandpass) ve bant durduran (bandstop) filtrelerdir.

- Alçak geçiren bir filtrede geçiş bandı, düşük frekans bölgesidir; durdurma bandı ise yüksek frekans bölgesidir.
- Yüksek geçiren bir filtrede ise bu bantlar tersine çevrilmiştir; düşük frekanslar zayıflatılır, yüksek frekanslar ise zayıflatılmaz.
- Bant geçiren bir filtre, orta frekans aralığında bir geçiş bandına ve hem düşük hem de yüksek frekanslarda durdurma bantlarına sahiptir.
- Bant durduran bir filtre ise, bant geçiren filtrenin geçiş ve durdurma bantlarını tersine çevirir.

Standart sürekli-zamanlı (continuous-time) sistem bloklarını (örneğin integratör, toplama noktaları) kullanarak filtre tasarımı yapılabilir.



Şekil 9 - (a) Alçak Geçiren (Lowpass) Filtre, (b) Yüksek Geçiren (Highpass) Filtre

Şekil 9 (a)'daki sistem, köşe frekansı ω_c (radyan/saniye cinsinden) olan bir alçak geçiren filtre olup, frekans cevabının genliği (dB), düşük frekanslarda sıfıra yaklaşır. Filtrenin transfer fonksiyonu şu şekildedir:

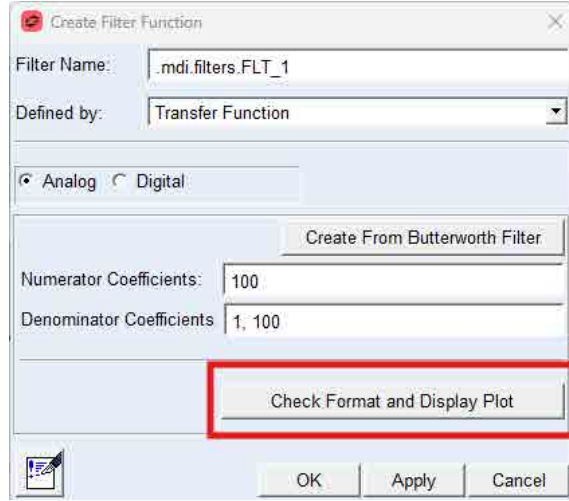
$$H(s) = \frac{\omega_c}{s + \omega_c}$$

Burada dikkat edilmesi gereken nokta, ω_c 'nin radyan/saniye cinsinden verilmiş olmasıdır. Adams View ortamında Hertz biriminde çalışılıyorsa, dönüşüm $\omega_c = 2\pi f_c$ eşitliğiyle yapılmalıdır. Örnek olarak köşe frekansı 100 radyan/saniye olan alçak geçiren filtrenin transfer fonksiyonu;

$$H(s) = \frac{100}{s + 100}$$

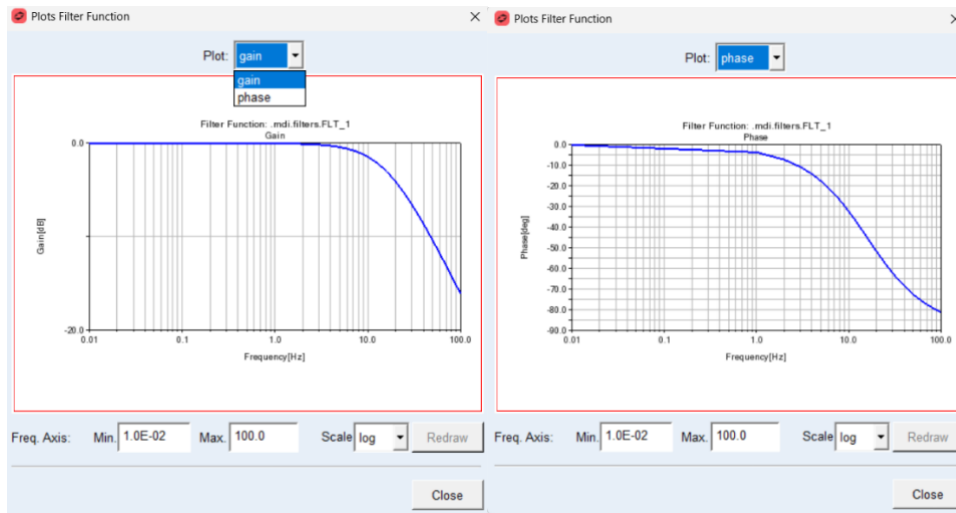
şeklinde olur.

Şekil 8’de belirtilen ‘*Numerator/Denominator Coefficient*’ kısımlarına sırasıyla filtrenin transfer fonksiyonunun pay kısmının katsayıları ve payda kısmının katsayıları yazılır.



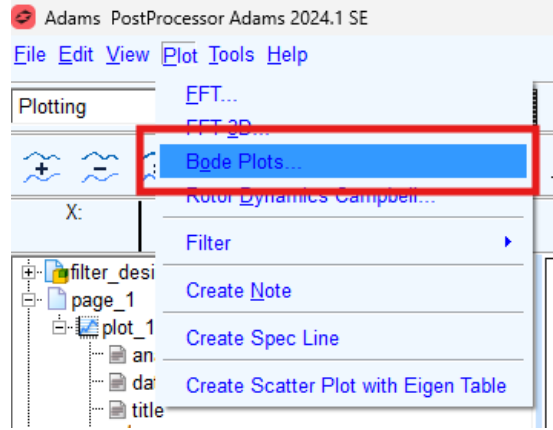
Şekil 10 – Bode Diyagramı Çizimi

Şekil 10’da gösterilen ‘*Check Format and Display Plot*’ kısmından tasarlanan filtrenin Bode diyagramı ve frekans cevabı görüntülenmektedir.



Şekil 11 – Alçak Geçiren Filtre Bode Diyagramı

Tasarlanan filtrenin Bode diyagramını daha detaylı incelemek için Adams PostProcessor arayüzünde yer alan ‘*Plot*’ sekmesi altındaki ‘*Bode Plots*’ aracı da kullanılabilir. Bu araç, Bode diyagramını incelemek için daha kullanışlı olabilmektedir.



Şekil 12 – Bode Plot Aracı

Adams PostProcessor ile Bode diyagramı çizimine ilişkin tüm ayrıntılı bilgilere, referanslar kısmında yer alan [6] numaralı kaynaktan ulaşılabilir.

Filtrenin Bode diyagramı, incelendiğinde, köşe frekansı 100 radyan/saniye'den (yaklaşık 15.92 Hertz) olan bir alçak geçiren filtre için, bu frekanstan daha küçük frekanslar sinyali hiçbir değişime uğratmadan geçirir. Ancak frekans değeri arttıkça, sinyalin geçirgenliği azalır ve sinyal filtre tarafından bastırılmaya başlar.

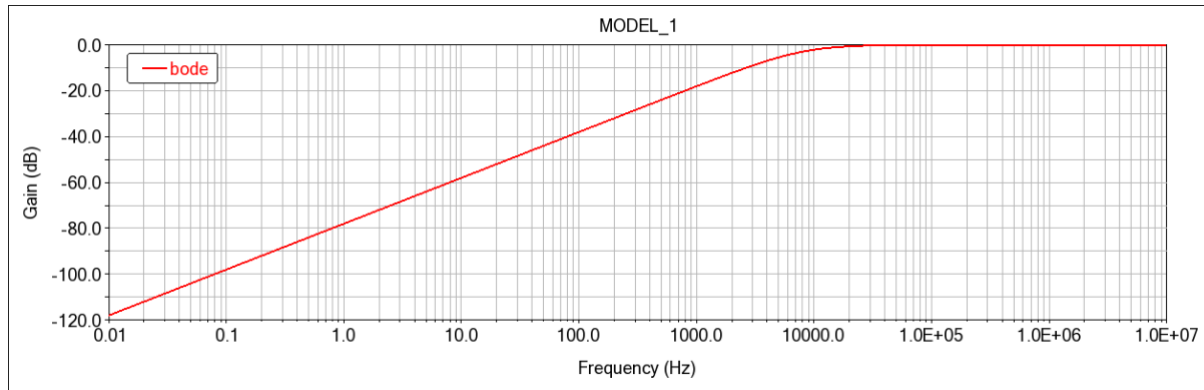
Şekil 9 (b)'deki sistem ise köşe frekansı ω_c olan bir yüksek geçiren filtredir. Bu sistemin transfer fonksiyonu ise şöyledir:

$$H(s) = \frac{s}{s + \omega_c}$$

Örnek olarak köşe frekansı 50,000 rad/saniye olan yüksek geçiren filtrenin transfer fonksiyonu;

$$H(s) = \frac{s}{s + 50000}$$

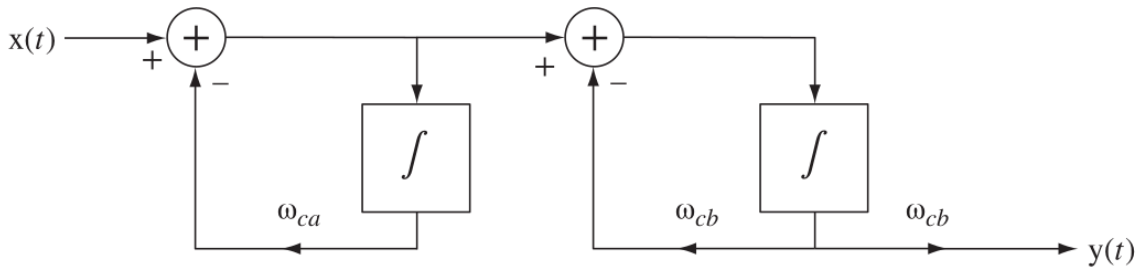
şeklindedir. Filtrenin Bode diyagramı ise Şekil 13'de gösterildiği gibidir.



Şekil 13 – Yüksek Geçiren Filtre Bode Diyagramı

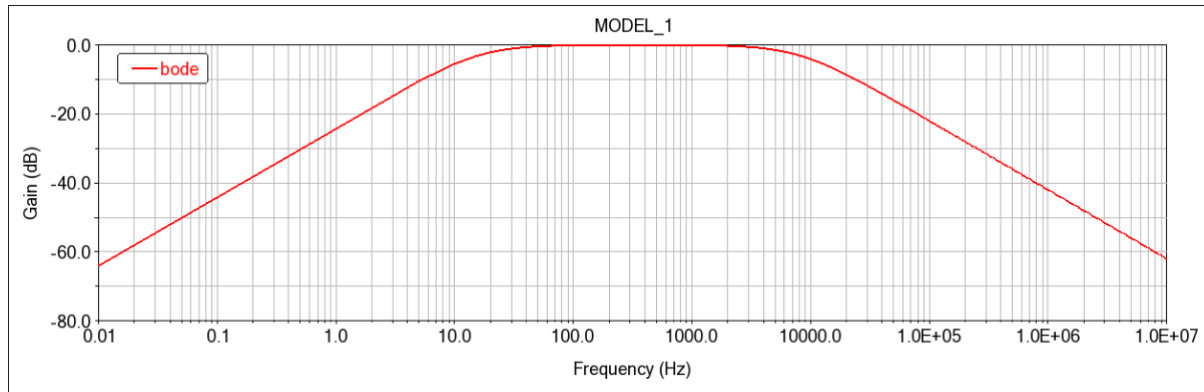
Her iki filtrede de eğer ω_c (köşe frekansı) ayarlanabiliyorsa, düşük ve yüksek frekanstaki sinyallerin göreceli güçleri kontrol edilebilir. Bu iki sistem ardışık (cascade) olarak bağlandığında, bir bant geçiren filtre elde edilir (Şekil 14). Bant geçiren filtrenin transfer fonksiyonu şu şekilde ifade edilir:

$$H(s) = \frac{\omega_{cb} s}{s^2 + (\omega_{ca} + \omega_{cb})s + \omega_{ca}\omega_{cb}}$$



Şekil 14 – Bant Geçiren Filtre

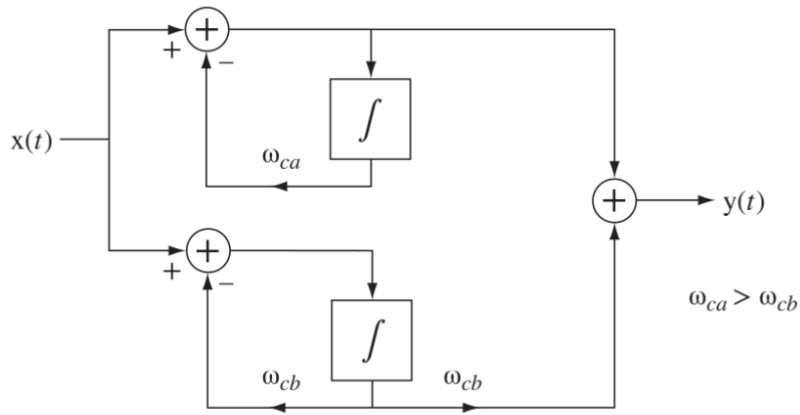
Örnek olarak $\omega_{ca} = 100$, $\omega_{cb} = 50,000$ değerleri için filtrenin frekans cevabı Şekil 15'te görüldüğü gibi olmaktadır.



Şekil 15 – Bant Geçiren Filtre Bode Diyagramı

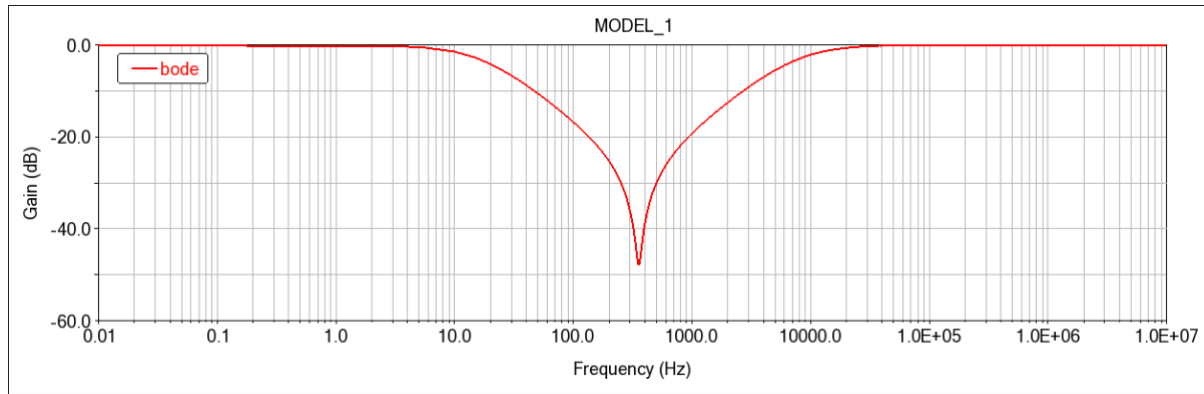
Bir bant durdurucu filtre, bir alçak geçiren filtre ile bir yüksek geçiren filtrenin paralel bağlanmasıyla oluşturulabilir. Ancak bu durumda alçak geçiren filtrenin köşe frekansı, yüksek geçiren filtreninkinden daha düşük olmalıdır (Şekil 16). Bant durdurucu filtrenin transfer fonksiyonu ise şu şekildedir:

$$H(s) = \frac{s^2 + 2\omega_{cb}s + \omega_{ca}\omega_{cb}}{s^2 + (\omega_{ca} + \omega_{cb})s + \omega_{ca}\omega_{cb}}$$



Şekil 16 – Bant Durduran Filtre

Örnek olarak $\omega_{ca} = 50000$, $\omega_{cb} = 100$ değerleri için filtrenin frekans cevabı Şekil 17’de görüldüğü gibi olmaktadır.



Şekil 17 - Bant Durduran Filtre Bode Diyagramı

3.2. Butterworth Filtrelemesi

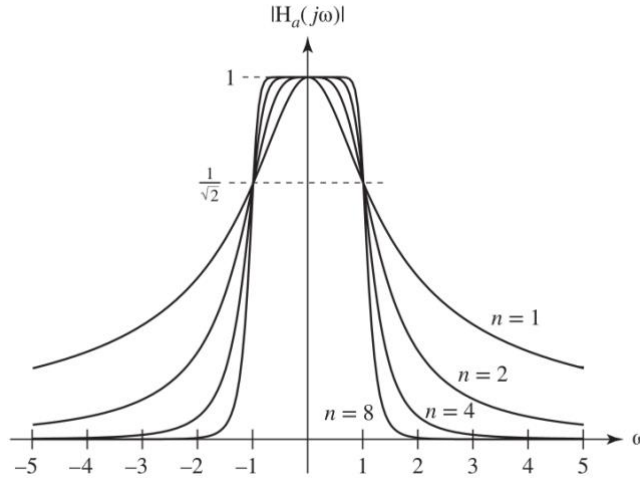
Çok yaygın kullanılan bir analog filtre türlerinden biri, İngiliz uygulamalı fizikçi Stephen Butterworth tarafından icat edilen Butterworth filtresidir.

n . dereceden bir alçak geçiren Butterworth filtresinin, frekans cevabının genliğinin karesi (squared magnitude of the frequency response) aşağıdaki şekilde ifade edilir:

$$|H_a(j\omega)|^2 = \frac{1}{1 + \left(\frac{j\omega}{j\omega_c}\right)^{2n}}$$

Düşük geçiren Butterworth filtresi, geçiş bandındaki frekanslarda ($\omega = \omega_c$) maksimum düzlüğe sahip olacak şekilde tasarlanmıştır. Yani, frekans cevabı geçiş bandında monotonik olarak değişir ve frekans sıfıra yaklaştıkça, eğrinin türevi sıfıra yaklaşır.

Şekil 18’de, $\omega_c = 1$ köşe frekansına sahip bir Butterworth filtresinin dört farklı mertebedeki frekans tepkisini göstermektedir. Filtrenin mertebesi arttıkça, genlik frekans tepkisi ideal bir düşük geçiren filtrenin tepkisine yaklaşır.



Şekil 18 – Butterworth Filtresi Frekans Cevapları [2]

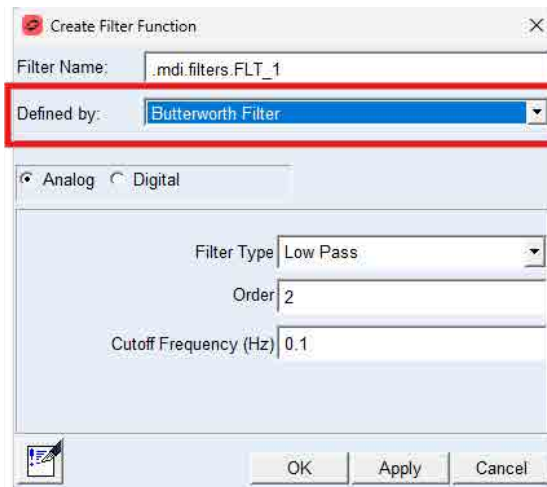
Buradaki bazı n değerleri için Butterworth filtresinin transfer fonksiyonları aşağıda verilmiştir:

$$n = 1 : \quad H(s) = \frac{\omega_c}{s + \omega_c};$$

$$n = 2 : \quad H(s) = \frac{\omega_c^2}{s^2 + \sqrt{2}\omega_c s + \omega_c^2};$$

$$n = 3 : \quad H(s) = \frac{\omega_c^3}{s^3 + 2\omega_c s^2 + 2\omega_c^2 s + \omega_c^3};$$

Adams View programında Butterworth filtre oluşturmak için iki farklı yöntem bulunmaktadır. Birinci yöntem, ‘Create Filter Function’ penceresinde yer alan ‘Defined by:’ kısmından “Butterworth Filter” seçeneğini seçmektir.

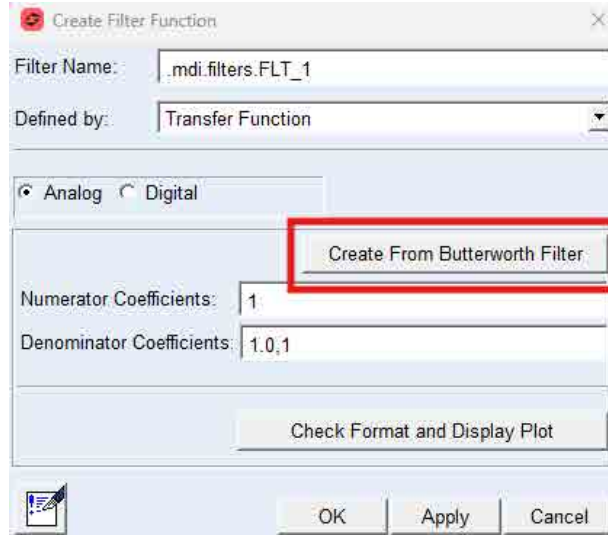


Şekil 19 - Butterworth Filtresi Oluşturma (1)

Bu pencerede bulunan:

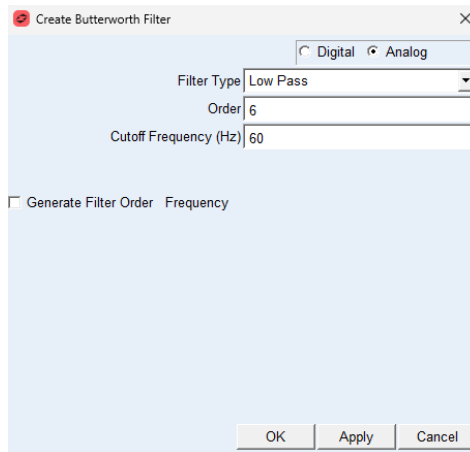
- 'Filter Type' kısmından, filtrenin tipi seçilebilir (Low Pass, High Pass, Band Pass, Band Stop.)
- 'Order' kısmından filtrenin kaçinci mertebeden olacağı ayarlanabilir.
- 'Cutoff Frequency' kısmından filtrenin köşe frekansı ayarlanabilir.

Butterworth filtresi oluşturmak için ikinci yöntem ise, 'Create Filter Function' penceresinde 'Defined by:' seçeneğinden Transfer Function seçildikten sonra, aynı pencerede bulunan 'Create From Butterworth Filter' seçeneğinin kullanılmasıdır.



Şekil 20 – Butterworth Filtresi Oluşturma (2)

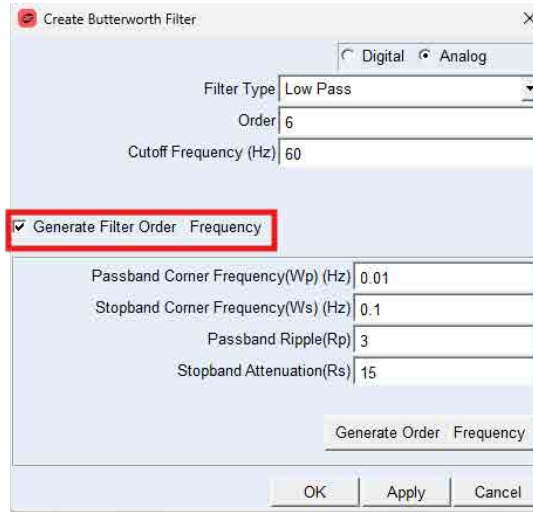
Bu seçeneği seçtikten sonra, Şekil 21’de gösterildiği gibi bir pencere açılır.



Şekil 21 – Butterworth Filtresi Oluşturma (3)

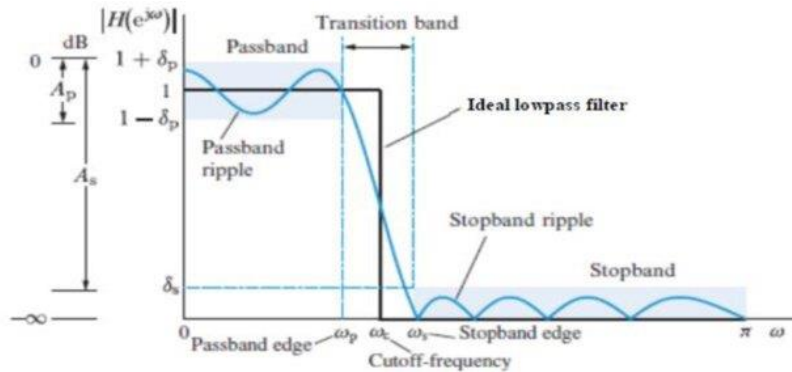
Bu kısımdan, yine aynı şekilde filtrenin türü, mertebesi ve köşe frekansı ayarlanarak filtre oluşturulabilir.

Bu pencerede bulunan ‘Generate Filter Order Frequency’ seçeneği seçilerek, filtre parametreleri daha spesifik değerler için detaylı biçimde ayarlanabilir.



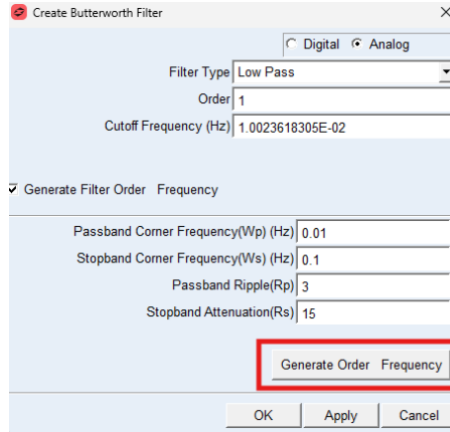
Şekil 22 – Butterworth Filtresi Oluşturma (4)

Bu kısımda bulunan parametrelerin, filtrenin frekans cevabı üzerindeki etkisi Şekil 23’de gösterilmiştir.



Şekil 23 – Butterworth Filtresi Frekans Cevabı

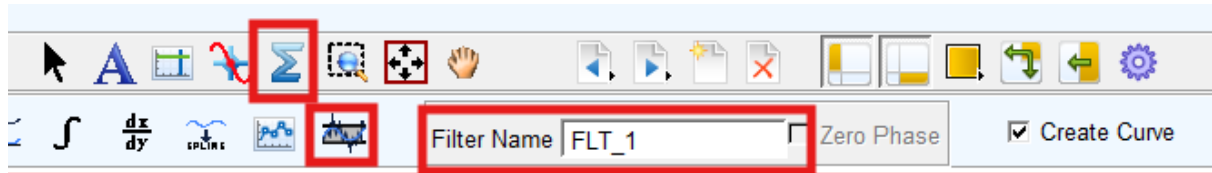
Belirtilen parametreleri ilgili alanlara tanımlandıktan sonra, ‘Generate Order Frequency’ seçeneği seçilerek, filtrenin mertebesi ve köşe frekansı Adams tarafından otomatik olarak hesaplanacaktır.



Şekil 24 – Generate Order Frequency

Tüm parametreleri ayarlandıktan sonra, sırasıyla 'Apply' ve 'Ok' seçeneklerini seçerek filtre tasarımı tamamlanır.

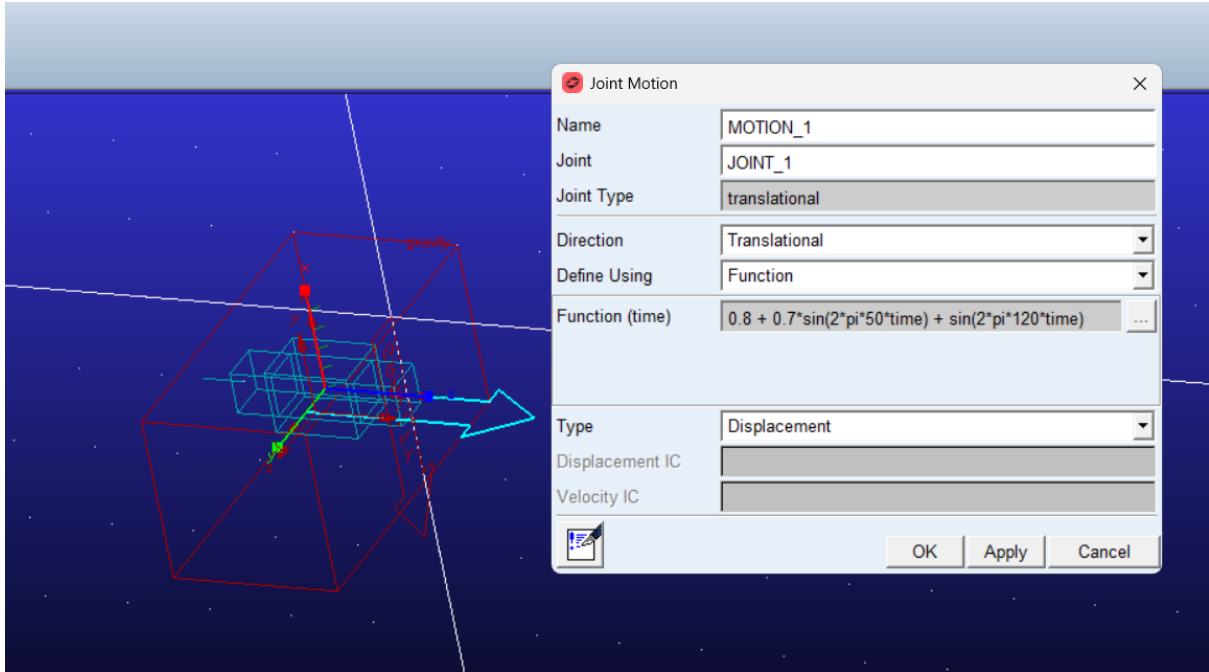
Tasarlanan filtrenin bir sinyale uygulanması için Adams PostProcessor içerisinde bulunan 'Curve Edit Toolbar' sekmesi altında yer alan 'Filter a curve' seçeneği seçilebilir ve 'Filter Name' kısmına tasarlanan filtrenin ismi yazılabilir. Bu işlemler tamamlandıktan sonra filtrelenmek istenen sinyal seçilmelidir.



Şekil 25 - Filtrenin Sinyale Uygulanması

4. ÖRNEK SİNYAL VE FİLTRELEME ÇALIŞMASI

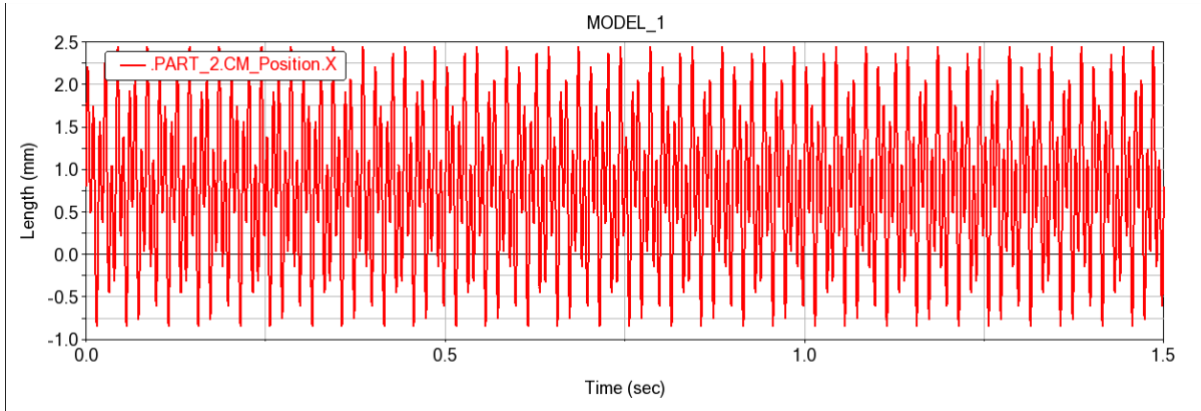
Adams View programında, örnek bir çalışma kapsamında manuel olarak oluşturulan bir sinyalin FFT analizi ve filtrelemesi gerçekleştirilmiştir. Oluşturulan bir 'Translational Joint' e Şekil 26'te gösterildiği gibi deplasman hareketi tanımlanmıştır.



Şekil 26 – Örnek Modeldeki Hareket Fonksiyonu

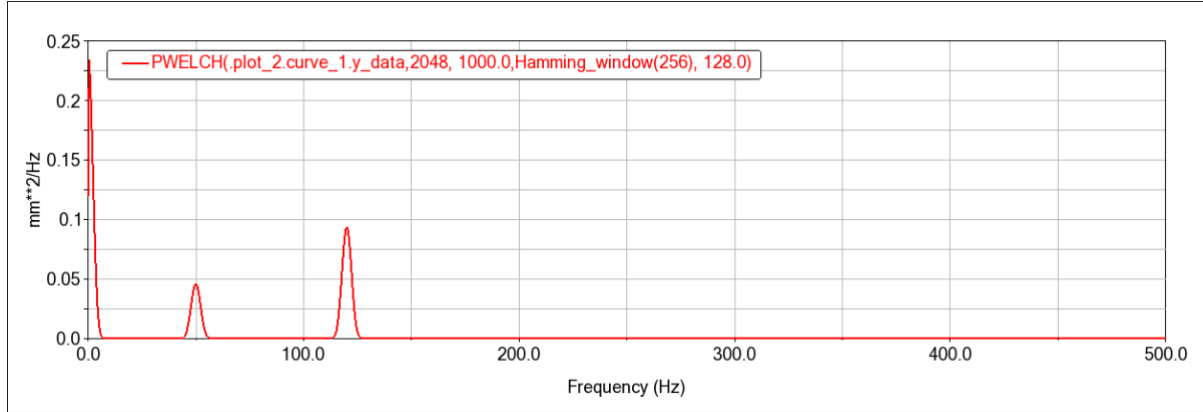
Oluşturulan modeli simülasyon süresi 1.5 saniye ve adım boyutu 0.001 olarak ayarlanmıştır.

Oluşturulan hareketin grafiği Şekil 27’de gösterildiği gibidir.



Şekil 27 – Hareket Grafiği

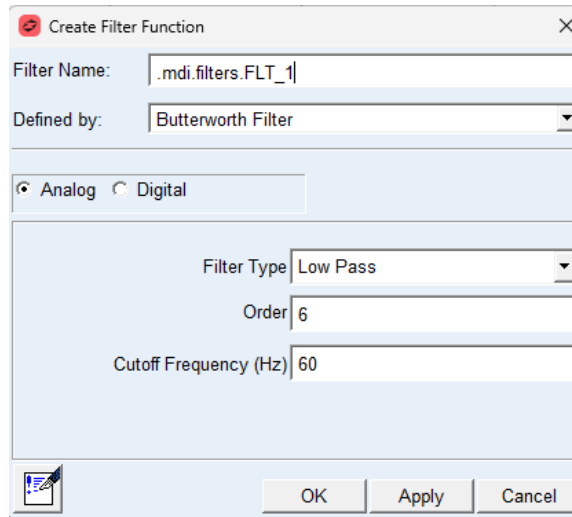
Bu sinyalin Fast Fourier Transform (FFT) grafiği şekil 28’de gösterildiği gibidir.



Şekil 28 – Hareket Sinyalinin FFT Grafiği

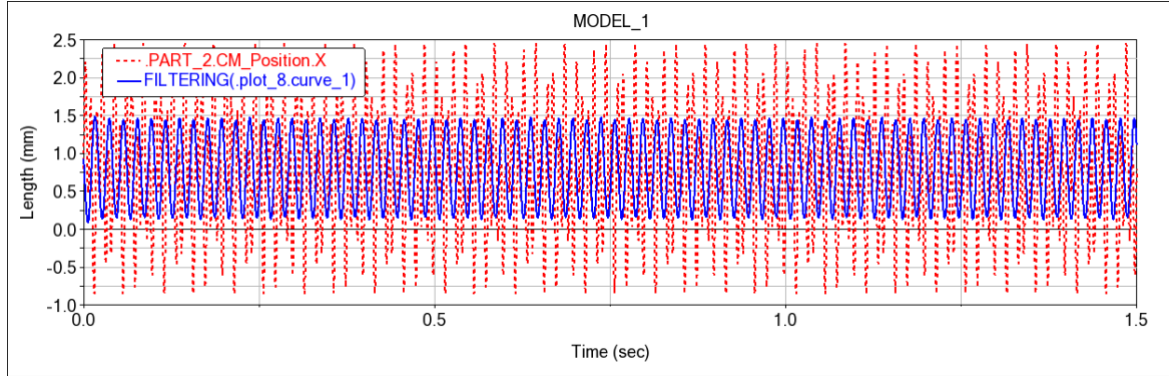
Sinyalin FFT grafiği incelendiğinde, üç frekans değeri için sinyalin büyüklüğü gözlemlenmektedir. Bu değerler 0 Hz, 50 Hz ve 120 Hz’dir.

Bu sinyalin filtrelenmesi için alçak geçiren Butterworth filtresi tasarlanmıştır. Tasarlanan filtre, 0 Hz ve 50 Hz frekanslarını geçirirken, 120 Hz frekansını geçirmemesi amaçlanmıştır. Butterworth filtresinin mertebesi 6 olarak belirlenmiş ve köşe frekansı, 50 Hz ‘den büyük, 120 Hz’den küçük olacak şekilde 60 Hz olarak seçilmiştir.



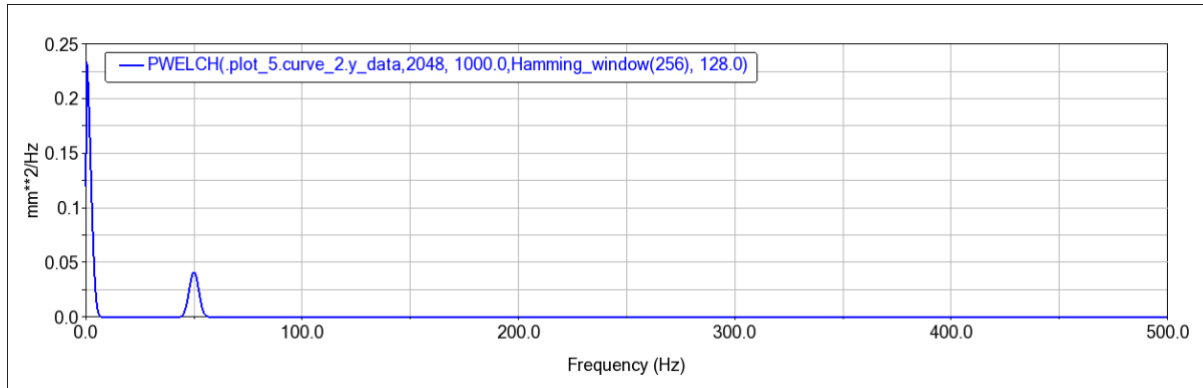
Şekil 29 – Örnek Filtre Tasarımı

Hareket sinyaline tasarlanan filtre uygulandığında, Şekil 30’da gösterildiği gibi hem gürültülü sinyal hem de filtrelenmiş sinyalin grafiği oluşturulmuştur.



Şekil 30 – Gürültülü ve Filtrelenmiş Sinyal Grafiği

Şekil 31’de filtrelenmiş sinyalin FFT grafiği gösterilmektedir. Grafik incelendiğinde, 0 Hz ve 50 Hz frekansların filtreden tam olarak geçtiği, ancak 120 Hz frekansının filtreden geçmediği gözlenmektedir.



Şekil 31 – Filtrelenmiş Sinyalin FFT Grafiği

5. REFERANSLAR

- i. Adams 2024.1 Help Documentation
- ii. Alan V. Oppenheim, Alan S. Willsky, with S. Hamid - Signals and Systems-Prentice Hall (1996)
- iii. Dr. Michael J. Roberts - Signals and systems _ analysis using transform methods and MATLAB (2018)
- iv. Data Driven Science & Engineering, Machine Learning, Dynamical Systems, and Control, Steven L. Brunton, J. Nathan Kutz
- v. ADAMS POSTPROCESSOR – FAST FOURIER TRANSFORM (FFT), Deniz Aslan
- vi. ADAMS POSTPROCESSOR – BODE PLOT KULLANIMI, Hüseyin Umutcan Şanlı