

LMS Adaptif Dizi Anten Algoritmasının Sabit Noktalı Sayısal İşaret İşleyici Uygulaması

M. Gökhan YILDIZ¹, Haydar KAYA¹, Şevket GÖĞÜSDERE²

Karadeniz Teknik Üniversitesi

Elektrik-Elektronik Mühendisliği Bölümü¹, Enformatik Bölümü²

61080 Trabzon, TÜRKİYE

mgyildiz@ktu.edu.tr, hkaya@ktu.edu.tr, sevket@ktu.edu.tr

Özet: Adaptif dizi antenler, gezgin haberleşme sisteminin kapasite problemini çözebilecek, önemli birkaç yöntemden biridir. Bu çalışmada adaptif antenlerin, sayısal işaret işleyiciler (DSP) ile gerçekleştirilmesi üzerine En Küçük Ortalama Kareler (LMS) algoritmasının kullanıldığı bir uygulama yapılmıştır. Yapılan uygulamada Texas Instruments'a ait sabit noktalı (fixed point) bir işlemci olan TMX320C6211 sayısal işaret işlemcisi üzerinde LMS Adaptif Anten algoritmasının performansı incelenmiştir.

1. Giriş

Adaptif anten sistemiyle, istenen işaretlerin ve girişim işaretlerinin varış doğrultularının tespit edilebilmesi için ileri düzeyde adaptif işaret işleme algoritmaları sürekli bir şekilde uygulanır. Bu çalışmada, istenen işaretin girişim ve gürültüye oranını (SINR) artıracak optimum ağırlık katsayılarını bulan, adaptif LMS algoritması kullanılmıştır. LMS algoritmasının optimum noktalara yakınsama hızı, bilinen diğer algoritmalarinkine (RLS, SMI) oranla daha yavaştır. Fakat bu algoritmanın hesaplama karmaşıklığı diğerlerinininkine göre çok daha azdır [1]. Adaptif anten sistemlerinde işlem yoğunluğunun yükünü çeken ve hızı belirleyen modül DSP modülüdür. Sabit noktalı işlemcilerin kayan noktalı işlemcilere göre duyarlılığı çok düşük olduğundan hesaplamalarda hata payı daha fazladır. Fakat çoğu endüstriyel uygulamalarda sabit noktalı DSP işlemciler, kayan noktalı DSP işlemcilere tercih edilirler. Bunun temel sebepleri ise; aynı CPU saat hızında, sabit noktalı işlemlerin kayan noktalı (floating point) işlemlere göre daha hızlı olması, sabit noktalı işlemcilerin kayan noktalı işlemcilere göre daha az güç harcamaları ve daha az yarı iletken gerektirmeleridir [2]. Burada sabit noktalı işlemcilerdeki duyarlılıktan kaynaklanan hataların sistem başarımına olan etkisi incelenmiştir.

2. Sabit Noktalı LMS Algoritmasının Adaptif Dizi Antenlerde Kullanımı

En küçük ortalama kareler algoritması, ağırlık vektörünü gelen veriye göre adapte eden bir algoritmadır. LMS algoritması aşağıdaki üç denklemle tanımlanabilir;

$$y(k) = w^H(k)x(k) \quad (\text{süzgeç çıkışı}) \quad (1)$$

$$e(k) = r(k) - y(k) \quad (\text{kestirim hatası}) \quad (2)$$

$$w(k+1) = w(k) + \mu x(k)e^*(k) \quad (\text{ağırlık adaptasyonu}) \quad (3)$$

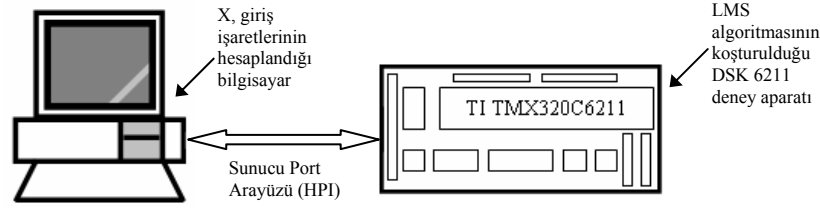
Burada $r(k)$, referans işaretini ve μ , adım büyüklüğünü belirtir. LMS algoritmasında adım büyüklüğünün seçimi çok önemlidir. Eğer μ çok küçük seçilirse, her bir adım ağırlık vektörü üzerinde küçük değişimlere sebep olacak ve algoritma yavaş çalışacaktır. Bununla beraber eğer μ çok büyük seçilecek olursa adaptif dizi kararsız hale gelebilir. Bundan dolayı μ , $0 < \mu < 1/\lambda_{\max}$ aralığında seçilmelidir. Burada $\lambda_{\max}$, R özilişki matrisinin en büyük özdeğeridir [1].

LMS algoritmasının sabit noktalı DSP üzerindeki başarımını inceleyebilmek için öncelikle sabit noktalı işlemlerde karşılaşılabilecek problemlerin çözümleri ve bu çözümlerin LMS algoritmasında nasıl uygulanacağı göz önünde bulundurulmalıdır. Sabit noktalı işlemlerde, N bitlik iki sayının çarpımı $2 \times N$ bitlik, toplamı ise $N+1$ bitlik bir sonuç verecektir. N bitlik sabit noktalı bir işlemcide çarpma ve toplama işlemleri sonucunda N bitten büyük değerlerde sonuçların meydana geldiği taşma (overflow) durumu veya işaret bitinin beklenmedik bir şekilde değiştiği sarma (wrap-around) durumu ile karşılaşılabılır. Sabit noktalı LMS algoritmasında da çarpma ve toplama işlemleri sıkça kullanıldığından karşılaşılabilecek sorunların üstesinden gelebilmek için doyum (saturation) yöntemi kullanılabilir [3]. Doyum yönteminde, N bitlik sistemde kullanılabilecek en büyük

$$\mu = \alpha_{fix} 2^{-k_2} = 2^{-k_3-k_2} \Leftrightarrow \alpha_{fix} = 2^{-k_3} \quad (9)$$

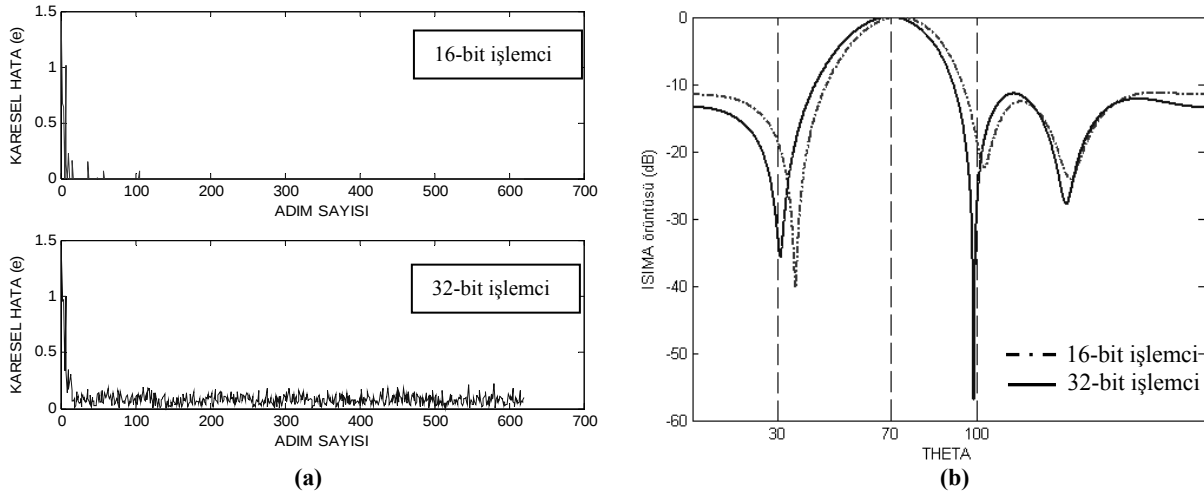
292

adaptif dizi anten algoritmasında kullanıp en uygun ağırlık katsayılarını hesaplamakta ve çıkış işaretini oluşturmaktadır. Giriş ve çıkışlar eş zamanlı olup sistem gerçek zamanlı olarak çalışmaktadır.



Şekil 2. Uygulamanın genel yapısı

Bu çalışmada sabit noktalı LMS algoritması iki ayrı durum için koşurulmuş ve başarımı incelenmiştir. Bu iki durum işlemcinin 32 bitlik veya 16 bitlik olarak koşurulmasıdır. Her iki durum için de bilgisayar tarafından sırasıyla 16 ve 8 bitlik giriş işaretleri hesaplanmış ve bu veriler HPI üzerinden DSP'nin belleğine gönderilmiştir. Uygulamada ortamın 0,01 varyanslı Gauss gürültüsüne sahip olduğu, 4 elemanlı doğrusal anten dizisinin $0,5 \lambda$ aralık ile dizildiği, istenen işaretin 70° 'den 0 dBW gücünde, girişim işaretlerinin ise 30° ve 100° 'lerden ise yine 0 dBW gücünde gelmekte olduğu varsayılmıştır.



Şekil 3. 16 ve 32 bit işlemcilerin, (a) Hesapladığı karesel hata. (b) Hesapladığı ışıma örüntüleri

Şekil 3.'den de görüldüğü gibi 32 bit işlemci yaklaşık 31 dB SINR oranı, 16 bit işlemci ise yaklaşık 18 dB SINR oranı sağlamış ve 32-bit işlemcide sıfırlar girişim işaretlerine, 16-bit işlemcidekine göre daha iyi yönlendirilmiştir.

4. Sonuç

Yapılan çalışmada 32 bit işlemcinin duyarlılığının 16-bit işlemcinin duyarlılığından daha fazla olduğu görülmüştür. Karesel hatalar dikkate alındığında 16 bitlik işlemcide hesaplanan hata değeri belli bir adımdan sonra sıfır olmakta, bunun sonucunda da bu adımdan sonra ağırlık katsayıları güncellenememektedir. Burada hata değeri gerçekte sıfır değil çok küçük bir değere eşittir. Ancak bu küçük değer 16-bit işlemcinin duyarlılığının çok daha altında kaldığından işlemci bu değeri sıfır olarak algılamakta ve ağırlık katsayılarını güncellenememektedir. 32-bit işlemcide ise duyarlılık daha yüksek olduğundan daha küçük olan hata değerlerini de göz önünde bulundurabilmekte ve en uygun ağırlık katsayılarına yakın değerde ağırlık katsayıları hesaplayabilmektedir. Sonuçta, 32 bit işlemci ile sıfırlar girişim işaretlerine daha doğru yönlendirilebilmekte ve daha yüksek SINR değerleri elde edilebilmektedir.

Kaynaklar

- [1]. Haykin, S., Adaptive Filter Theory, Prentice Hall, Englewood, New Jersey, 1991.
- [2]. Kuo, S.M., ve Lee, B.H., Real Time Digital Signal Processing, Wiley, England, 2001.
- [3]. Yıldız, M.G., Adaptif Antenlerde Sayısal İşaret İşleyici Uygulamaları, Yüksek Lisans Tezi, K.T.Ü, Fen Bilimleri Enstitüsü, Trabzon, 2004.