

OPEN SOUND CONTROL İLE SES SENTEZLEYİCİ DENETLEME: “PURE DATA” ORTAMINDA EKLEMELİ SENTEZLEME UYGULAMASI

Arda EDEN*

İnönü Üniversitesi, Güzel Sanatlar ve Tasarım Fakültesi,
Müzik Bölümü, Müzik Teknolojisi Ana Bilim Dalı, Malatya

ÖZET

OSC (Open Sound Control) 90'lı yılların sonlarında elektronik çalgılar arasında iletişim sağlamak amacıyla geliştirilmiş bir protokoldür. OSC mesajları özellikle bilgisayar ağları başta olmak üzere çeşitli yollardan iletilebilmektedir. Oldukça esnek bir yapıya sahip olan OSC, mesajların farklı uygulamalara göre şekillendirilebilmesine olanak sağlamaktadır. Çalışmada Android 2.2 yüklü bir tablet PC üzerinde çalışan FingerPlay MIDI yazılımı kullanılarak, OSC protokolü ile Pure Data üzerinde gerçekleştirilmiş eklemeli sentezleyicinin kontrol edilmesine yönelik bir uygulama sunulmuştur..

Anahtar Kelimeler: OSC, open sound control, pure data, eklemeli sentezleme

CONTROLLING SOUND SYNTHESIZER WITH OPEN SOUND CONTROL: AN APPLICATION OF ADDITIVE SYNTHESIS IN “PURE DATA”

ABSTRACT

OSC (Open Sound Control) is a protocol build for communicating data among electronic musical instruments. OSC messages can be transmitted via several ways especially over computer networks. Having a flexible structure, OSC lets shaping messages in a unique way for any application. In this study, an application of controlling an additive synthesizer implemented on Pure Data by using FingerPlay MIDI, a control interface software running on Android 2.2 tablet PC is introduced.

Keywords: OSC, open sound control, pure data, additive synthesis

1. GİRİŞ

1980'li yıllarda ortaya çıkan MIDI (Musical Instruments Digital Interface), elektronik çalgılar ve çeşitli ses, görüntü araçları ile bilgisayar sistemleri arasında iletişim kurmak amacıyla tasarlanmıştır. Bugün yaklaşık 30 yıllık geçmişe sahip bir protokol olan MIDI halen geçerliliğini korumaktaysa da, sahip olduğu bir takım sınırlılıklar ve esnek olmayan yapısı ile özellikle jestsel (gestural) müzik uygulamaları gibi performansların sergilenmesinde yetersiz kalabilmektedir.

90'lı yılların sonlarına doğru ortaya çıkan OSC (Open Sound Control), müzikal kontrol verilerinin iletilmesine yeni bir bakış açısı getirmiştir. OSC, özel bir donanıma gerek duymadan, varolan ağ alt yapılarını kullanarak bir ortamdan diğerine veri iletimi sağlayabilmektedir. Kullanıldığı uygulamanın ihtiyaçları dahilinde şekillendirilebilen OSC veri içeriği, iletişim protokolünün tasarım üzerindeki sınırlandırmasını ortadan kaldırmaktadır.

Çalışmada kontrol arayüzü olarak, Android 2.2 işletim sistemi üzerinde çalışan “FingerPlay MIDI” yazılımından faydalanılmıştır. FingerPlay MIDI, WIFI üzerinden OSC mesajları iletilebilmektedir. Tasarlanan eklemeli sentezleyici Pure Data ortamında gerçekleştirilmiştir.

* Yazışma yapılacak yazar: ardaeden@gmail.com

Makale metni 29.05.2011 tarihinde dergiye ulaştırılmış, 24.06.2011 tarihinde basım kararı alınmıştır.

2. OSC

OSC (Open Sound Control), UC Berkeley Center for New Music and Audio Technology (CNMAT) bünyesinde Adrian Freed ve Matt Wright tarafından, bilgisayarlar, ses sentezleyiciler ve modern ağ teknolojisine uygun biçimde düzenlenmiş diğer çokluortam araçları arasında iletişim kurmak amacıyla geliştirilmiş bir protokol, bir içerik biçimidir (content format). Oldukça esnek bir yapıya sahip olan OSC, gerçek zamanlı ses kontrolü ve pek çok ortam (media) işleme uygulamasına yönelik geniş iletişim ve kontrol imkanı sunmaktadır. Geliştirilmiş ve halen geliştirilmekte olan pek çok ağ etkileşimli araç, yazılım sentezleyici, çeşitli programlama dilleri ve algılayıcı ölçüm araçları gibi sayısız uygulamada OSC protokolünden faydalanılmaktadır (OSC, 2011).

OSC’de iletişim, istemci / sunucu ilişkisi içerisinde gerçekleşmektedir. OSC verileri paket adı verilen birimlerden meydana gelir. OSC paketi gönderen uygulama (yazılım, donanım vs.) istemci, paket alan uygulama ise sunucu rolü üstlenir. OSC yüksek seviyeli (high-level), uygulama düzeyinde (application-layer) çalışan bir protokoldür. Bu nedenle verinin iletiildiği ağ mekanizmasına yönelik herhangi bir tanımlama içermez (Wright, vd., 2003). OSC her ne kadar bir protokol olarak tanımlansa da aslında bir içerik biçimini ifade eder. Bir protokolden beklenen tüm nitelikler OSC’nin yapısında bulunmamaktadır (Freed & Schmeder, 2009).

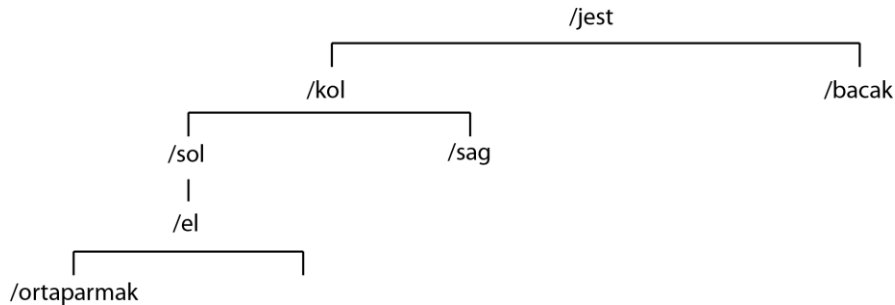
Bir OSC paketi ikilik veri bloğundan meydana gelen içerik (contents) ve içerik bölümünü de kapsayan 8-bit uzunluğundaki boyut (size) bölümlerinden meydana gelir. OSC paket içeriği, bir OSC mesajı veya OSC yığıdır (OSC bundle) (Wright & Freed, 1997). Paket içeriğinin ilk baytı içeriğin hangisi olduğunu belirler.

2.1. OSC Mesajı

OSC verisi bir adres örüntüsü (address pattern), bir tip etiket metni (type tag string) ve argümanlardan oluşan bir mesajdır. OSC ile iletilecek olan argümanlar, etiket tipi metnini takip eden bir dizi ikili (binary) veri biçimindedir. Tipik bir OSC mesajı “Adres örüntüsü - Tip Etiket Metni- Argüman” şeklinde düzenlenmiştir.

2.1.1. OSC Adres Alanı

OSC sunucusu içerisinde tanımlanmış tüm kontrol noktaları sunucunun adres alanı (address space) olarak adlandırılan ağaç yapılı hiyerarşik bir düzen içerisinde tanımlanır. Adres alanına ait her düğümün (node) bir sembolik ismi bulunur (Wright, vd., 2003). OSC mesajı içerisinde bulunan adres örüntüsü sunucunun adres alanı içerisindeki herhangi bir düğümü işaret eder. OSC, MIDI ve ZPI gibi protokollerin aksine elektronik çalgıların ne şekilde tasarlanacağı ve hangi mesajlar ile haberleşeceği gibi konularda bağlayıcı değildir. Geliştirici, yazılımın veya donanımın işleyiş biçimine ve ihtiyaçlara bağlı olarak adres alanını dilediği biçimde oluşturabilir. OSC bu anlamda oldukça esnek bir yapıya sahiptir.



Şekil 1 – OSC sunucu adres alanı yapısı

Open Sound Control ile Ses Sentezleyici Denetleme: "Pure Data" Ortamında Eklemeli Sentezleme Uygulaması

Şekil 1’de bir jestsel (gestural) müzik uygulamasına yönelik bedensel hareket algılayıcılarından gelen verilerin değerlendirilebilmesi amacıyla bir OSC sunucu üzerinde oluşturulmuş adres alanı yapısı görülmektedir. Bu adres alanı içerisinde bulunan “/ortaparmak” adlı düğüme bir OSC mesajı içerisinde “/jest/kol/sol/el/ortaparmak” adres örüntüsüyle ulaşılabilir. Bu anlamda OSC’de adres örüntüsü UNIX dosya sisteminin hiyerarşik ağaç yapısına benzerlik göstermektedir. OSC’de adres alanı dinamiktir, çalışma esnasında yapısı ve içeriği değiştirilebilir.

OSC adres alanının içerisinde bulunan dalların (branches) ayrıştığı düğümler OSC hazneleri (OSC Container), en uçta bulunan düğüm ise OSC metodu (OSC Method) olarak adlandırılır. Şekil 1’deki yapıya göre düşünüldüğünde “/jest/kol/sol/el/ortaparmak” adresi, “jest” haznesi içerisindeki, “kol” haznesi içerisindeki, “sol” haznesi içerisindeki, “el” haznesi içerisindeki “ortaparmak” metodunu işaret etmektedir. Bir adres içeren mesaj sunucuya ulaştığında ilgili adresin gösterdiği metod beraberinde gelen argümanlarla birlikte çağrılır. Bir metodun çağırılması programcılıkta kullanılan prosedür çağrısına benzetilebilir (OSC, 2011). OSC’de metod ve hazne adları Tablo 1’de görülen karakterler dışında kalan yazdırılabilir tüm ASCII karakterlerden meydana gelebilir.

Tablo 1 - OSC Metod ve OSC Container isimlendirilmesinde kullanılmayan karakterler

| Karakter | Ad | ASCII Kodu (ondalık) |
|----------|-------------------------|----------------------|
| | boşluk | 32 |
| # | sayı işareti | 35 |
| * | asteriks | 42 |
| , | virgöl | 44 |
| / | slash | 47 |
| ? | soru işareti | 63 |
| [| köşeli parantez açma | 91 |
|] | köşeli parantez kapatma | 93 |
| { | süslü parantez açma | 123 |
| } | süslü parantez kapatma | 125 |

2.1.2. OSC Tip Etiket Metni (Type Tag String)

OSC tip etiket metni “,” (virgöl) ile başlayan ve mesaj içerisinde bulunan her bir argümana karşılık gelen bir dizi karakterden meydana gelir. Virgülden sonra gelen her karakter OSC Tip Etiket (OSC Type Tag) olarak adlandırılır. OSC içerisinde kullanılan temel tip etiketleri Tablo 2’de görülmektedir.

Tablo 2 - OSC Tip Etiketleri ve karşılığı olan argüman tipi

| OSC Tip Etiketi | İlgili argümanın tipi |
|-----------------|--|
| i | int32 - 32-bit, big-endian, ikinin tümleyeni, tamsayı |
| f | float32 - 32-bit, big-endian, IEEE 754, ondalıklı (floating point) sayı |
| s | OSC-string - NULL ile biten bir dizi ASCII karakter. Veriyi 32 bitin tam katı haline getirmek üzere NULL karakterler ile tamamlanır. |
| b | OSC-blob - int32 büyüklüğünde ikili veri. 0 - 3 sıfır değerli byte eklenerek 32 bitin tam katı haline getirilebilir. |

Örnek olarak; bir tane “float32” argüman için “f” tip etiketi, iki tane “int32”, bir tane “OSC-string” ve 3 tane “float32” argüman tanımlamak için “,iisfff” tip etiketi kullanılır. Tek başına “,” tip etiketi hiçbir argüman olmayacağını belirtir. Bazı OSC uygulamaları kendi aralarında haberleşmek amacıyla yukarıda belirtilen argüman tipleri dışında başka argüman tiplerinden de yararlanırlar. Bu argüman tipleri Tablo 3’de verilmiştir.

Tablo 3 - Diğer OSC Tip Etiketleri ve karşılıkları olan argüman tipleri

| OSC Tip Etiketi | İlgili argüman tipi |
|-----------------|---|
| h | 64 bit, big-endian, ikinin tümleyeni tamsayı |
| t | OSC-timetag |
| d | 64 bit (double) IEEE 754 ondalıklı sayı |
| S | Alternatif bir OSC-string tipi |
| c | ASCII karakter, 32 bit olarak gönderilir |
| r | 32 bit RGBA rengi |
| m | 4 bayt MIDI mesajı. MSB’den LSB’ye doğru baytlar: port id, status byte, data1, data2 |
| T | True. Argüman içermez. |
| F | False. Argüman içermez. |
| N | Nil. Argüman içermez. |
| I | Infinitum. Argüman içermez. |
| [| Dizi (array) başlangıcı belirtir. Takip eden etiketler dizi argümanlarının tipini belirler. |
|] | Dizi bitişini belirtir. |

2.2. OSC Yığını (OSC Bundle)

OSC yığını, “#bundle” metnini takip eden bir OSC zaman etiketi (OSC Time Tag) ile sıfır veya daha fazla OSC yığın öğesinden (OSC Bundle Element) meydana gelir. Her OSC yığın öğesi ise boyut ve içerik bilgisinden oluşur (Tablo 4). İçerik bir mesaj olabileceği gibi bir yığın da olabilir. Bir OSC yığını başka OSC yığınları içerebilir.

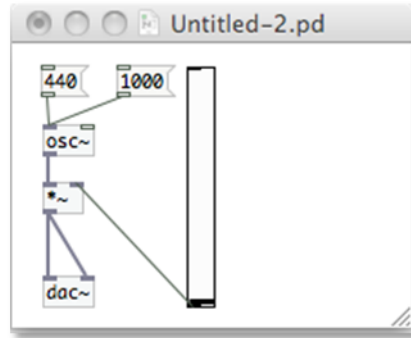
Tablo 4 - OSC Yığınına meydana getiren bölümler

| Veri | Boyut | İşlev |
|-----------------------------|---|-------------------------------------|
| OSC-String “#bundle” | 8 bayt | Verinin bir yığın olduğunu belirtir |
| OSC-timetag | 8 bayt | Tüm yığına etki eden zaman etiketi |
| Birinci yığın öğesi boyutu | int32 = 4 bayt | Birinci yığın öğesi |
| Birinci yığın öğesi içeriği | Birinci yığın öğesi boyutunda belirtilen kadar. | |
| İkinci yığın öğesi boyutu | int32 = 4 bayt | İkinci yığın öğesi |
| İkinci yığın öğesi içeriği | İkinci yığın öğesi boyutunda belirtilen kadar. | |
| Vb. | | |

3. PURE DATA

Pure Data (Pd), ses, video ve grafik işlemek amacıyla IRCAM (Institut de Recherche et Coordination Acoustique/Musique) bünyesinde Miller Puckette tarafından tasarlanmış bir gerçek zamanlı grafiksel programlama ortamıdır. Yazılımın çekirdeği her ne kadar Puckette tarafından programlanmış olsa da, genel olarak tüm Pd paketi pek çok programcının katılımıyla geliştirilmektedir. Bununla beraber Pure Data açık kaynak kodlu bir özgür yazılımdır (Pure Data, 2011).

Pura Data dosyaları "patch" olarak adlandırılır. "Patch" lerin oluşturulabilmesi için Pure Data grafiksel bir arayüz sunar. Pd, "dataflow (veri akışı)" olarak adlandırılan bir programlama yöntemi kullanır. Tüm fonksiyonlar nesnelere (objects) halinde düzenlenmiştir. Bu nesnelere grafiksel ortamda giriş ve çıkış soketleri bulunan kutular biçiminde gösterilirler. Nesnelere arasındaki bağlantılar kablolar (wires) ile gerçekleştirilir (Farnel, 2010). Benzer biçimde bu kablolar grafiksel olarak çizgiler ile ifade edilmektedir.



Şekil 2 - Basit bir Pura Data patch dosyası grafiksel görünümü

Basit bir Pure Data patch dosyası Şekil 2'de görülmektedir. Aynı dosyanın ".pd" uzantılı metin tabanlı içeriği aşağıda verilmiştir:

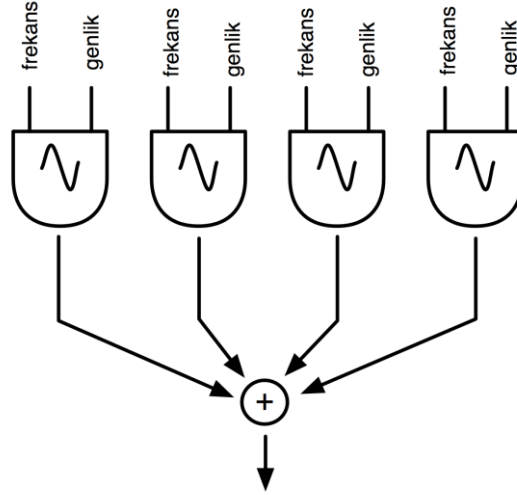
```
#N canvas 659 244 450 300 10;  
#X msg 55 40 440;  
#X msg 105 43 1000;  
#X obj 55 94 osc~;  
#X obj 55 152 *~;  
#X obj 165 55 vsl 15 128 0 1 0 0 empty empty empty 0 -9 0 10 -262144 -1 -1 0 1;  
#X obj 53 226 dac~;  
#X connect 0 0 2 0;  
#X connect 1 0 2 0;  
#X connect 2 0 3 0;  
#X connect 3 0 5 0;  
#X connect 3 0 5 1;  
#X connect 4 0 3 1;
```

4. EKLEMELİ SENTEZLEME (ADDITIVE SYNTHESIS)

Eklemeli sentezleme yöntemi temel olarak Fourier analizi teorisine dayanmaktadır. Bu nedenle eklemeli sentezleme kimi zaman Fourier sentezleme olarak da adlandırılır. Teknik, karmaşık periyodik dalgaların çeşitli genlik ve frekanslardaki sinüsoidlerin birleştirilmesi yöntemi üzerine kuruludur (Miranda, 2002:50). Bu sinüsoidlerin birleştirilmesiyle meydana gelen bileşenler doğrudan algılanmasa da, sonuçta meydana gelen sesin tınısı üzerinde büyük etkiye sahiptirler (Moore, 1990:207).

Ekleme yönteminde her bileşen için bir osilatöre gerek duyulmaktadır. Osilatör sayısı arttıkça üretilen bileşen sayısı da artar. Ekleme yönteminde üretilen tınının harmonik yapısının önceden biliniyor olması gerekir. Bu nedenle sentezlenmek istenen sesin önceden analiz edilerek bileşen içeriğinin belirlenmesi gerekmektedir (Önen & Pasinlioğlu: 78-79). Şekil 3’de 4 osilatörlü bir ekleme sentezleyici şeması görülmektedir.

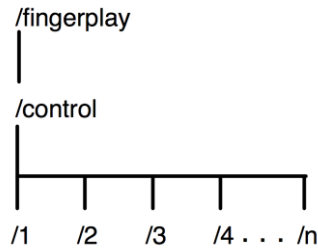
Ekleme yönteminde her bir osilatörün frekansı ve genliği ayrı ayrı belirlenir. Frekanslar elde edilecek tınının niteliğine bağlı olarak temel frekansın tam katları (harmonik) veya uyumsuz doğuşkanlar meydana getirecek biçimde belirlenebilir.



Şekil 3 - 4 adet sinüs osilatörü ile ekleme sentezleme

5. PURE DATA İLE OSC KONTROLLÜ EKLEMELİ SENTEZLEYİCİ TASARIMI

FingerPlay MIDI tarafından gönderilen OSC adres örüntüsü Şekil 4’deki biçimde düzenlenmiştir. Burada “/fingerplay” ve “/control” hazneleri yazılım tarafından belirlenmiştir. “/control” haznesi içerisinde bulunan “/1, /2, ..., /n” biçiminde düzenlenmiş metodlar ise FingerPlay MIDI “layout” dosyası içerisinde her bir kontrole istenildiği biçimde atanabilmektedir.

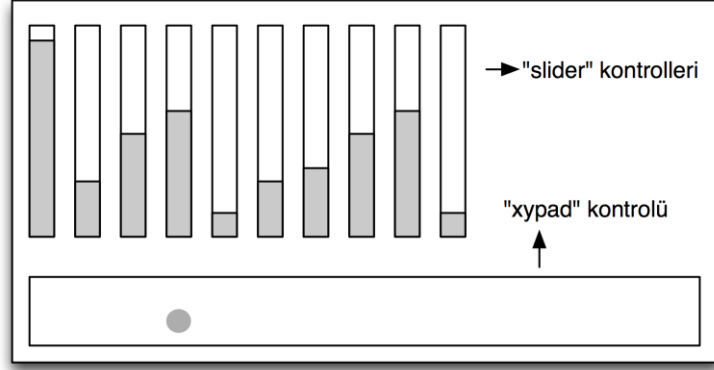


Şekil 4 - FingerPlay MIDI Adres örüntüsü

FingerPlay MIDI “layout” dosyası yeniden düzenlenerek oluşturulan ekran tasarımı Şekil 5’de görülmektedir. Ekleme yönteminde kullanılacak 10 adet osilatörün genliklerini kontrol edecek “slider” kontrollerinin “/control” haznesi altındaki metodları “/1, /3, /5, ... , /19” biçiminde atanmıştır. Her bir “Slider” kontrolü dokunma ve değer üretme biçiminde iki olay meydana getirmektedir. Tasarımdaki birinci “slider”, dokunma olayı gerçekleştiğinde “/1”, değer

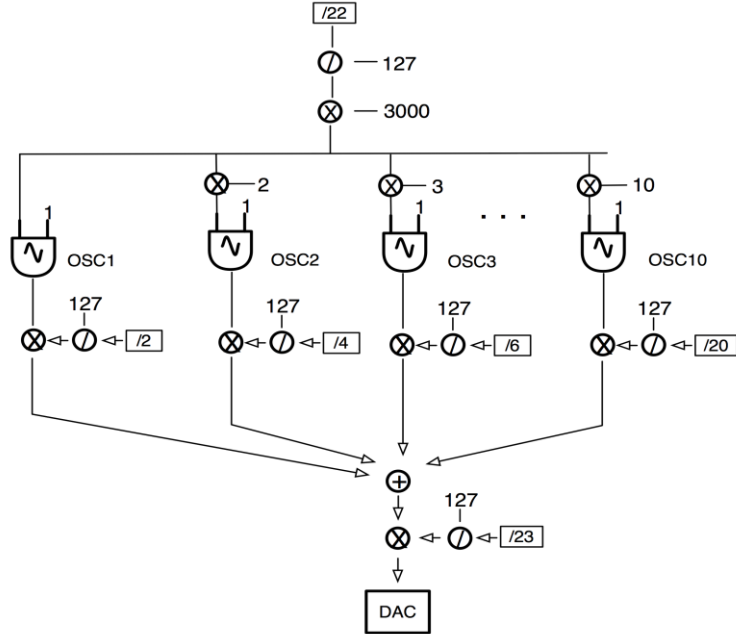
Open Sound Control ile Ses Sentezleyici Denetleme: "Pure Data" Ortamında Eklemeli Sentezleme Uygulaması

üretme gerçekleştiğinde ise "/2" metodunu meydana getirmektedir. Benzer biçimde tüm "slider" kontrolleri dokunma olayı için tek sayılı, değer üretme olayı için çift sayılı metodları meydana getirecektir.



Şekil 5 - FingerPlay MIDI ekran tasarımı ve metodların belirlenmesi

Tasarımda kullanılan "xypad" kontrolü eklemeli sentezleyicinin temel frekansını ve çıkış genliğini belirlemek amacıyla tercih edilmiştir. Bu kontrole "/21" metodu atanmıştır. "xypad" kontrolü "slider" kontrollerine benzer biçimde dokunma olayı için "/21", "x" eksenindeki değer değişimi için "/22", "y" eksenindeki değer değişimi için "/23" metodlarını meydana getirmektedir. FingerPlay MIDI her metod için 0 - 127 arasında değer alan argüman üretmektedir.



Şekil 6 - Tasarlanan 10 osilatörlü eklemeli sentezleyicinin sinyal akış diyagramı

Tasarlanan 10 osilatörlü eklemeli sentezleyicinin sinyal akış şeması Şekil 6'de görülmektedir. Pure Data'da analog sayısal dönüştürücüye gönderilen ses sinyal genliği 0 - 1 arasında değişmektedir. Bu nedenle FingerPlay MIDI tarafından gönderilen 0 - 127 arasında değişen argümanlar 127'ye bölünerek gerekli dönüşüm gerçekleştirilmiştir. "xypad" kontrolü tarafından gönderilen "/22" metodu sentezleyicinin temel frekansını belirlemektedir. Maksimum

En alttaki "OSCroute" nesnesi tarafından gönderilen argümanlar yapılan bağlantılar aracılığı ile ilgili nesnelere giriş soketlerine aktarılmaktadır. 0 - 127 arasında değişen bu argüman değerleri 127 ile bölünerek 0 - 1 aralığına indirgenmiş istenilen maksimum frekans değeri ile çarpılarak (bu uygulamada 3000 Hz) "osc~" nesnelere girişlerine gönderilmiştir. İlk osilatör 1. harmoniği yani temeli meydana getireceğinden hiçbir işleme tutulmayan bu frekans değeri üst harmonikleri meydana getirecek diğer osilatörlerin girişlerine gönderilmeden önce "2, 3, ... ve 10" katsayıları ile çarpılmıştır. Osilatör çıkış genliklerini kontrol eden "slider" kontrolleri tarafından meydana getirilen "1/2, 1/4, ... ve 1/20" metodlarına ait argümanlar 127 ile bölünerek 0 - 1 aralığına indirgenmiş ve her bir osilatörün çıkış genliği ile çarpılarak seviyelendirmelerin gerçekleştirilmesi sağlanmıştır. "xypad" kontrolü tarafından meydana getirilen "1/23" metodunun argümanı, 127 ile bölünüp oranlanılarak tüm osilatörlerin toplam genliği ile çarpılmış böylelikle genel çıkış seviyesinin "xypad" kontrolünün düşey hareketi ile kontrol edilmesi mümkün hale gelmiştir.

6. SONUÇLAR

Harmonik seviyelerinin ayarlanması sırasında "slider" kontrollerinin meydana getirdiği görsellik, sentezleme sonucunda oluşan dalgaya ait bileşenlerin frekans domenindeki grafiksel görüntüsünü sergilemektedir. Bu anlamda arayüz spectrumun belirlenmesine görsel destek sağlamaktadır. Seslendirme amacıyla tercih edilen "xypad" kontrolü, frekans ve genliği tek bir kontrol altında toplayabilmesi açısından kullanışlı görünmekle beraber melodi seslendirmeye uygun bir arayüz olmaktan uzaktır. Bununla beraber kontrollerin tepki süresi gerçek zamanlı bir performans sergileyebilmek için oldukça yeterli görünmektedir.

KAYNAKLAR

- Farnel, A. 2010. *Designing Sound*. MIT Press.
- Freed, A., Schmeder, A. 2009. Features and Future of Open Sound Control Version 1.1 for NIME, USA. *Proceeding at the New Interfaces for Musical Expression (NIME-09)*.
- Miranda, E. R., 2002. *Computer Sound Design: Synthesis Techniques and Programming*. Focal Press.
- Moore, F. R., 1990. *Elements of Computer Music*. Prentice Hall, New Jersey USA.
- OSC. 2011. Introduction to OSC. Open Sound Control, <http://opensoundcontrol.org/introduction-osc> (Erişim tarihi: 25.05.2011)
- Önen, U., Pasinlioğlu, T. 2011. *Synthesizer Teknolojileri ve Programlama: Geçmişten Günümüze Popüler Sentez Teknikleri*, Çitlembik Yayınları, İstanbul.
- Pure Data. 2011. Pure Data (by IOhannes m zmoeInig), <http://puredata.info> (Erişim tarihi: 29.05.2011)
- Wright, M., Freed, A., Momeni, A. 2003. OpenSound Control: State of the Art 2003, Canada. *Proceeding at the New Interfaces for Musical Expression (NIME-03)*, 153.
- Wright, M., Freed, A. 1997. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers, Greece. *International Computer Music Conference 101-104*.