

**MÜZİK TEKNOLOJİSİNDE SENKRONİZASYON ve
ZAMAN KODLU SENKRON SORUNLARININ KÖKENİ(*)**

**Synchronization on Music Technology and The Root of the
Time-Code Sync. Problems**

DOI NO: 10.5578/amrj.57396

Cihan IŞIKHAN¹

Özet

Senkron (eşgüdüm, synchronization), donanımlar arasında uygun adım hareketi sağlayan bir iletişim, sistemin bir merkezden yönetilmesine olanak tanıyan makinalar arası bir protokoldür. Eşgüdüm sayesinde iki ya da daha fazla donanım zaman ve hız değerlerinde anlaşıp uyum içinde çalışır veya çalışması beklenir. Çünkü özellikle yoğun içerikli profesyonel ses kayıt uygulamalarında bazen teoriyle pratik her zaman örtüşmeyebilir ve eşgüdüm kaynaklı ciddi sorunlar ortaya çıkar. Bu sorunların pek çoğunun bağlantı, ayar ya da kullanım hatasından ortaya çıktığı düşünülse de aslında hiç de azımsanmayacak kadarı eşgüdüm tipinden, seçilen protokolün olağan yapısından kaynaklanır.

Bu çalışma, teorik olarak hiçbir sorun olmayacağı beklentisiyle yola çıkılan uygulamalarda bile yalnızca seçilen zaman kodlu (time code) eşgüdümün dijital yapısı gereği ortaya çıkan ciddi sorunlar, sorunların kaynağı olan protokollerin içyapıları ve çözüm önerilerini içermektedir. Sözü edilecek protokoller, uygulamalarda en çok kullanıldığı tespit edilen ve geleneksel zaman kodu SMPTE'nin MIDI protokolündeki temsilcisi MTC ve türevleriyle (tam kare, MMC vs.), özel bir otomasyon eşgüdüm sistemi olan HUI'dir. Çözüm için otomasyonun devreye girdiği HUI'yi bir tarafa bırakacak olursak, MTC'nin ana hatlarıyla bir çeyrek kare mesajı olması, MMC'nin ise içinde tam kare MTC'yi barındırmasına rağmen klasik bir SysEx yapısı taşıyor olması, zaman kodlu eşgüdüm sorunlarının en önemli kökenidir.

Anahtar Kelimeler: Müzik Teknolojisi, Senkronizasyon, MIDI.

(*) Güncellenerek yeniden yayıma hazırlanan bu makale, Audio Technology for Music and Media (ATMM'13) kongresinde bildiri olarak sunulmuş ve bildiri kitapçığında yayımlanmıştır.

¹ Prof. Dr., Dokuz Eylül Üniversitesi, Güzel Sanatlar Fakültesi, Müzik Bilimleri Bölümü, Müzik Teknolojisi Anabilimdalı.

Abstract

Synchronization (sync.) is a communication protocol that allows hardware to manage from a unique center in audio-video system. Two or more equipment are processed or expected to processing in harmony using time and speed values. Sometimes the theory and practice doesn't match in content-intensive applications of audio recording and serious problems come into existence. Most of these problems are caused by the structure of sync. type except user error or connections.

In this paper, some sync. problems caused by digital structure of selected time-code sync. such as MTC, MMC, HUI are discussed. MTC is a quarter-frame message included acting a limited area of audio system. MMC works in SysEx message that requires full compliance with hardware using full frame of MTC. Also, HUI is a part of automation more than sync. However, these entire structural properties are able to base on the problem of time coded synchronization.

Key Words: Music Technology, Synchronization, MIDI.

GİRİŞ

Müzik teknolojisi yalnızca bir bütünü içermez. Bütünü oluşturan her bir parça, kendine ait disiplini farklı olsa da mutlaka müzik teknolojisi kapsamına girer, bu kapsamda incelenir. Bu bağlamda, profesyonel müzik kayıtlarını bir bütün olarak kabul edersek, bu bütünün parçalarından biri de müzik teknolojisi kapsamına giren eşgüdümdür (*synchronization*).

Eşgüdüm, donanımlar arasında uygun adım gidişi sağlayan bir iletişim, sistemin bir merkezden yönetilmesine olanak tanıyan makineler arası bir protokoldür (Durmaz, 2000: 93). Eşgüdüm sayesinde iki ya da daha fazla donanım zaman ve hız değerlerinde anlaşarak *eşzamanlı* çalışır. Örneğin bilgisayar, çok kanallı kayıt cihazı, video oynatıcı, ses masası, sound modül, klavye gibi tek bir sistem içinde birbirine bağlı farklı donanımlar, ancak eşgüdüm sayesinde aynı anda *uygun adım* çalışırlar.

Eşgüdüm için öncelikle, eşgüdümü sağlayacak protokolün belirlenmesi gerekir. Çünkü eşgüdümün temeli olan zaman ve hız, onu içeren verileriyle bir saat dilimini, bir tempoyu, bir geri sayımı ve hatta herhangi bir sürekli tınıyı kendine kılavuz seçebilir (Huber 2007: 223). Protokol seçiminden sonra bu protokolü donanımlara aktaracak bağlantının yapılması şarttır. Bu doğrudan bir audio da olabilir, MIDI gibi akışın kendine özgü protokolle sağlandığı herhangi bir verisel bağlantı da. Bu adımdan sonra seçilecek eşgüdüm tipine göre kullanım detayları devreye girer. Bundan sonrası, kullanıcı ve

kontroller arasındaki ilişkiye bağlı olarak gelişir.

Ancak her teknolojik içerikte olduğu gibi eşgüdümde de veri akışının her zaman sorunsuz yürümesi beklenemez. Yani, eşgüdüm için her şeyin sağlandığı uygulamalarda bile daha yolun en başında ya da işe başladıktan sonra birçok sorunla karşılaşılabilir. Bu sorunların pek çoğu bağlantı, ayarlar ya da kullanıcı hatasından kaynaklıdır ancak hiç de azımsanmayacak kadarı eşgüdüm tipinin, bir başka ifadeyle protokolün kendisindedir. İşte bu olası sorunun zor olan çözümü için, eşgüdüm protokolünün yapısına, yani sorunun kökenine inmek gerekir. Bunun için ses kayıt stüdyolarında en çok kullanılan eşgüdüm protokollerini masaya yatırdık, yapısını inceledik. Çünkü incelemede amaç eşgüdümde kullanılan her bir protokol tipini tanıtmak değil; piyasada en çok kullanımda olanların sorunlarına eğilmektir. Bu da hiç kuşkusuz bizi, *zaman kodlu* çalışan eşgüdüm tipine götürdü.

Zaman kodlu eşgüdümün atası SMPTE'dir (Işıkhan, 2009). Adını, 20. yy. başlarında faaliyete geçen *Society of Motion Picture and Television* derneğinin baş harflerinden alan SMPTE, bugün zaman kodunda bir dünya standardıdır. SMPTE, eşgüdümde saat-dakika-saniye ve kare bilgilerini referans alarak üretim yapar ve bu bilgileri karşı tarafa analog ya da dijital audio sinyaliyle aktarır. Birkaç istisna hariç günümüzde neredeyse tüm ses kayıt stüdyolarında eşgüdüm için SMPTE kullanılır. SMPTE'nin dijital ortamdaki en iyi temsilcilerinden biri MIDI olduğundan, MIDI araçlı zaman kodları SMPTE için en uygunu sayılabilir. Dolayısıyla bu çalışmada zaman kodlu eşgüdüm sinyali olarak MIDI tabanlı sinyaller ele alınacaktır.

Bu çalışmada hedeflenen, MIDI'li eşgüdüm için kullanılan zaman kodlarının yapısını incelemek ve mimari yapıyı bağlantı-ayar ya da kullanıcıdan kaynaklanmayan olası eşgüdüm sorunlarıyla örtüştürecek çözüm yolu ya da yolları önermektir

MTC ve Yapısı

MTC (*MIDI Time Code*) kısaca, üzerinde zaman bilgisi taşıyan MIDI mesajıdır. Taşınan zaman bilgisi, sistemdeki diğer donanımlarla eşgüdüm sağlar. Bir başka ifadeyle MTC kullanılarak sisteme ait tüm donanımlar zaman bilgisi üzerinden eşgüdüm kurar. Burada zaman, sırasıyla kare (*frame*), saniye (*second*), dakika (*minute*) ve saat (*hour*) bilgilerinin tümüyle ve aynı anda taşınmasıyla oluşturulur. Dolayısıyla MTC, MIDI'ye özel bir zaman kodudur (Işıkhan 2009).

Aslında MTC'deki bu özel kodlama, 1987 yılında Digidesign bilgi işlem uzmanı Chris Meyer ve kurucu ortaklarından Evan Brooks

tarafından SMPTE yapısının aynen MIDI'ye uyarlanmış halidir. Dolayısıyla sistemde SMPTE ile eşgüdüm sağlayan audio donanımlar, MIDI ile eşgüdüm sağlayan MIDI'li donanımlarla bağlantı kurabilir, onlarla eşgüdümlü çalışabilir. Bunun için yalnızca SMPTE audio sinyalini MIDI verisine çeviren harici ya da dâhili donanımlara ihtiyaç duyulur. Dolayısıyla MTC, mimari yapısı gereği adı SMPTE ile birlikte anılması gereken bir MIDI mesajıdır.

MIDI'deki 241 numaralı (*Heximal*: F1, *Binary*: 1111 0001) mesaj MTC için ayrılmıştır. MIDI'li donanım, üzerindeki MTC üretici açıldığında 241 numaralı mesajdan zaman bilgisini kendine MIDI ile bağlı diğer donanıma iletir veya mesajı karşılar. Bu mesajın veri akışı soldan sağa şu şekildedir:

1111 0001	Onnn dddd	1111 0001	0zzz mmmm
-----------	-----------	-----------	-----------

Akıшта önce MTC bilgisi taşındığını belirten 241 numaralı sistem mesajı iletilir: 1111 0001 Ardından veri "byte"ı (*data byte*) olduğunu gösteren sabit 0 değeriyle hangi zaman bilgisinin taşındığını gösteren 3 "bit"lik değişken bir kod (n ve z) gelir. Burada n ve z , 8 farklı zaman bilgisinin karşılığıdır (2^3). Bunlar sırasıyla,

0	Frames Low Nibble
1	Frames High Nibble
2	Seconds Low Nibble
3	Seconds High Nibble
4	Minutes Low Nibble
5	Minutes High Nibble
6	Hours Low Nibble
7	Hours High Nibble and SMPTE Type

olarak kodlanmıştır. SMPTE tipini belirleyecek olan ve y diyebileceğimiz özel bir değişken atamasıyla kare bilgisi seçilir ve sonrasında Low Nibble'lar akışta " n ", High Nibble'lar ise " z "ye karşılık gelir. Dolayısıyla mesaj sırasına göre önce n (*low*) sonra z (*high*) zaman bilgisi ayrı-ayrı taşınır. Bunun nedeni, d veya m ile gösterilen zaman değerinde gizlidir.

Zaman değeri, bilgisine göre şekillenen zamanın sayısal

karşılığıdır. Örneğin zaman bilgisi saat (*hour*) ise d ve m 0–23 arasında değer alır. Dakika veya saniye ise 0–59, kare ise y atama değerine göre 1–24/1–25/1–30 gibi değerler döndürür. Şimdi buraya dikkat: d ve m için akışta 4 bit kullanıldığından, örneğin 59 gibi bir değer ikili sayı sisteminde 4 bit ile doğrudan taşınamaz. Çünkü 4 "bit" in alacağı en büyük ondalık değer 16'dır (2^4). Bunu ortadan kaldırmak için tek yapılması gereken 4 "bit" in 8'e çıkarılmasıdır. Ancak, MIDI mimarisinin yapısı gereği her veri en az fazla 1 byte (8 bit) ile taşınacağından, 4 bit ile (*nibble*) tutulan tek bir verinin 8 "bit" e çıkarılması mimariye terstir. Bu nedenle sorunu çözmek için ilginç bir yöntem kullanılmış, zaman değeri tam ortadan ikiye bölünerek tersten haneler paylaştırılmıştır. Yani örneğin 59 sayısı veri akışında önce 9 sonra 5 ile iletilir. Bu bölünme sırasında doğal olarak 241 numaralı sistem mesajı tekrar araya girer. Böylece MTC veri yapısı tamamlanarak karşı tarafa iletilmiş olur. Örneğin 59. saniyenin (00:00:59:00) MTC veri yapısındaki karşılığı şu şekildedir:

1111 0001 (241)	0010 1001 (9)	1111 0001 (241)	0011 0101 (5)
-----------------	---------------	-----------------	---------------

241 numaralı MTC mesajı, *low* ve *high* olarak 8 farklı bilgiyi ikişerli gruplar halinde dört zamana böldüğünden (saat, dakika, saniye kare), MIDI jargonunda Çeyrek Kare Mesajı, (*Quarter-Frame Message*) olarak adlandırılır. Dolayısıyla 241 numaralı MTC mesajının MIDI'deki gerçek tanımı Çeyrek Kare Mesajı (ÇKM)'dir.

MMC ve Yapısı

MMC (*MIDI Machine Control*), 1992 yılında MIDI ailesine katılan gelişmiş bir sistem mesajı türüdür. MIDI'li donanımlara çalma (*play*), durma (*stop*), ileri-geri sarma (*forward- backward*) gibi komutları iletmek olan MMC sayesinde her bir donanım, herhangi bir zaman kaygısı gözetmeksizin kullanıcının eşgüdüm için ihtiyacı olan *hareket* bilgilerini otomatik olarak almış olur (Işıkhana 2009).

MMC mesajı, aslında kendi başına bir mesaj olmayıp, MIDI'de 240 numaralı mesajla başlayan (SysEx= Hex: F0, Binary: 1111 0000) ve 247 numaralı mesajla sonlanan (EOX (*End of Exclusive*= Hex: F7, Binary: 1111 0111)) SysEx (*System Exclusive*) mesajı içinde yer alır. Bu nedenle, bir kullanıcı için aslında bilinmesi gereken yapı SysEx mesajıdır. Bu mesaj kısaca, firma ve ürünlerine göre değişen özgün

bilgi gruplarını içerir. Örneğin bir MIDI uygulamasının hangi marka ve hangi model bir donanımla yapıldığını gösteren üretici bilgileri, tıpkı barkodlar gibi SysEx mesajında gizlidir. Ancak sistem otomasyon, veri yedekleme veya transfer etme gibi uygulamalarda çok daha geçerli olan üretici bilgileri, MMC gibi evrensel zaman kodlarının içine gömüldüğü durumlarda 0000 0110 sayı değeriyle sabitlenir. Yani MMC, yapısında firma kimlik bilgisi içermez. MMC bir SysEx mesajı olduğundan, üretici firmalara göre farklılık gösterebilir. Yani bazı MMC komutları bazı firmalar tarafından üretilmeyebilirler. Bu da MMC'li eşgüdüm sorunlarının en temel boyutu, bir başka ifadeyle sorunun kökenidir... MMC veri yapısı şu biçimdedir:

1111 0000	0111 1111	0ccc cccc	0000 0110	0kkk kkkk	1111 01111
-----------	-----------	-----------	-----------	-----------	------------

Bu yapıda 240 (1111 0000) ile başlayıp 247 (1111 0111) ile tamamlanan SysEx içinde konumlanır ve Sub-ID olarak adlandırılan bir durum "byte"ı ile sabitlenir (0000 0110). Yapıda, SysEx sonrası gelen "'kimlik' byte"ı daha önce belirtildiği gibi sabittir ve SysEx ile gelecek olan verinin, zamana bağımlı evrensel bir mesaj (universal real-time message) olduğunu gösterir. Bu mesajın yazılabilir olması, yani SysEx mesajının otomasyon veya yedekleme gibi tüm den veri transferine (data dump) açık olması nedeniyle, MMC için MIDI içinde global bir kanal ayrılabilir. Kanal bilgisi (c) genelde her donanım için 1 veya 16 olarak sabitlenmiştir. Mesajda k ile gösterilen değişken, MMC'nin karşı tarafa donanım kontrollerinden hangisinin yapıldığını belirtir komutu temsil eder. Daha önce MMC tanımını yaparken kullanıcı için *donanımın hareket* bilgisi olarak belirttiğimiz bu komutlar, 7 bit ile temsil edilirler. Bu komutların neler olduğu ise Tabloda gösterilmiştir. Örneğin MMC ile eşgüdümlü bir MIDI sistemde kayıt yazılımının stop tuşuna bastığımızda, mesajın komut "byte"ına 0000 0001 (Hex: 01) değeri yerleşir ve diğer donanımlara iletilir. Böylece MMC ile eşgüdümlü diğer tüm donanımlar durdurulmuş olur.

Tablo. Eşgüdüm için kullanılan MMC hareket komutları

MMC Komutu (k)	Heximal
Stop	01
Play	02
Deferred Play	03
Fast Forward	04
Rewind	05
Record Strobe (Punch In)	06
Record Exit (Punch In)	07
Record Ready	08
Pause	09
Eject	0A
Chase	0B
MMC Reset	0F
Write	40
Locate/Go To	44
Shuttle	47

MMC yapısında eşgüdüm, MMC içine gömülen MTC'de gizlidir. Bu özel yapı nedeniyle MMC içindeki MTC bilgisine *tam kare mesajı* verilir.

Bir kullanıcı, uygulamasının *synchronization* menüsünden eşgüdüm olarak MMC'yi seçtiğinde o uygulama, eşgüdüm için diğer donanımlara *hareket* bilgilerini göndermeye hazır hale gelir. MMC bir *zaman kodu* içermediğinden, aynı kullanıcı bu kez zaman kodu için tercih yapmak zorundadır ve bunun için çoğunlukla MTC tercih edilir. Eşgüdüm için asıl tercih MMC olduğundan, zaman kodunda MTC tercihi aslında MTC'nin Tam Kare mesajını çağırarak anlamına gelir. Bu yolla *Tam Kare* mesajları devreye girer ve zaman kodu SysEx üzerinden taşınır. Eğer kullanıcı MMC'yi devreye sokmadan doğrudan MTC ile çalışırsa bu durumda MTC Çeyrek Kare mesajları görevde olacaktır. Dolayısıyla, bir zaman kodu olarak MTC'nin hangi durumlarda hangi mesajla devreye girmesi gerektiğini aslında MMC belirler.

Kimi durumlarda bazı hareket komutları, MMC yerine MTC yeteneğiyle karşı tarafa iletilebilir. İşte bu iletilen özel durumlu komutlara MTC *Cueing* mesajları adı verilir. MIDI jargonunda *MTC kullanıcı bitleri (users bits)* olarak tanımlanan bu özel eşgüdüm biçimi,

tıpkı Tablo ile belirtilen MMC komutları gibi start, stop, fast forward, punch in-out vs. gibi hareket bilgilerini SysEx mesajı içine yapılandırılmış MTC zaman koduyla birlikte karşı tarafa gönderir.

SONUÇ

Artık herkes ister istemez kabul etmelidir ki günümüz teknolojik yapıda ses kayıtlarında eşgüdüm gerekliliğinin pek bir hükmü kalmadı. Şimdilerde pek çok uygulama yazılımların dar ama bir o kadar bütünsel hizmetlerinde gerçekleştiriliyor. Ancak yine de ses ile görüntünün birleştirildiği uygulamaların birçoğunda harici eşgüdüm kurulumlarının desteği şart. Ancak bu destek, geçmişte olduğu gibi sayıca fazla olan eşgüdüm tiplerinden çok, zaman kodlu olmak kaydıyla neredeyse tek bir eşgüdüm tipine kaydırılmış durumda: SMPTE.

SMPTE denildiğinde ise artık neredeyse istisnasız bir şekilde MIDI akla gelmekte. Zaman kodlu eşgüdümün en büyük yardımcısı MIDI, bünyesindeki MTC ve MMC mesajlarıyla eşgüdümde neredeyse tek başına bir kaynak durumunda.

Ancak her içerikte olduğu gibi bu tür eşgüdümelerde de kendi yapılarından kaynaklanan pek çok sorun mevcut. Önemli olan bu soruların üstesinden gelebilmek için sorunun kaynağına inebilmek. Yani kökenine ulaşmak.

MTC'deki ÇKM yapısından kaynaklı akışın doğrusal olmaması, sistemdeki olası MTC zaman kodu sorunlarının aslında en büyük sebebidir. Yani yukarıda belirtilen akışta zaman diliminin ikiye bölünmesi, örneğimizdeki 59. saniyenin veri akışında önce 9 sonra 5 olarak görülmesi ve üstelik 9 ve 5 arasına 241 numaralı mesajın tekrar yazılması, doğrusal akması gereken MTC mesajını farklı bir yapıya çekerek akışı etkiler. Bu durum maalesef MIDI veri akışında bant genişliğine yansır. Sistemde yalnızca MTC değil, aynı zamanda nota bilgileri, süreler, hiç kaybolmayan MIDI Clock vs. gibi birçok MIDI verisi hareket ettiğinden, bu verilerin arasına denk gelen MTC, çok dar bir alanda hareket etmek zorunda kalır. MTC bant genişliği, %100'lük bir dilimde ancak %8'lik bir aralığa denk gelir. Nota akışının %60 olduğu bir MIDI bağlantısında bu değer oldukça küçüktür.

Bu bir sorundur. Hatta bu sorun, MTC'de ender oluşan ve kullanıcının elinde olmayan eşgüdüm hatalarının da kökeni sayılır. Hatanın uygulamadaki görünümü, donanımların eşgüdümde birbirine eşit olmayan MTC değerleri olarak karşımıza çıkar. Örneğin bir donanım 3. saniyedeiken diğeri henüz 2. saniyeye ulaşır veya daha büyük farklarla eşgüdüm sağlanmaya çalışılır. Hatta kısmen de olsa

eşgüdümün hiç kurulmamasına veya durmasına neden olabilir.

Bu sorunun iki farklı çözümü vardır. Birincisi, MIDI bağlantısı için gerekli MIDI kablolarını kaliteli seçmek ve uzunluklarını kısa tutmaktır. Yani olası bir eşgüdüm sorununda eğer tüm seçenekler doğruysa ama halen MTC'de sıkıntı yaşıyorsa, önce kablo kalitesinden ve sonra uzunluğundan kuşkulanmak gerekir. Uzunluk mutlaka 2 metreden az tutulmalı, kablonun ve dolayısıyla soket ve jackların alaşımı, yoğunluklu yüksek bakırdan veya platinyuma kadar giden diğer maddelerden seçilmelidir. Sorunun kökeni sayısal veri akışında ve MTC yapısında olduğundan, kullanıcının ilk seçenektan başka bir müdahalesi maalesef beklenemez.

Tam kare mesajları, çeyrek kare olanların handikapı sayılan MIDI'de bantgenişliğini ortadan kaldıracak biçimde tasarlanmıştır. Tıpkı çeyrek karede olduğu gibi tam kare zaman kodu da saat-dakika-saniye-kare bilgisi taşır ancak en önemli fark, bu bilgilerin yerleştirildiği mesajdadır. Çeyrek kareler MIDI'de tek başlarına bağımsız bir mesajla giderlerken (241 numaralı mesaj), tam kareler SysEx mesajının içine gömülü bulunurlar. Böylece *alan darlığı* meselesi kendiliğinden ortadan kalkar. MMC mesajının SysEx içine gömülü bir küresel zaman kodu olduğu; genelde 1. veya 16. global MIDI kanalı üzerinden kendi adına kodlanmış özel yapısıyla saat, dakika, saniye ve kare bilgilerinden oluştuğu açıktır. Böylece her bir zaman dilimi 7 "bit"lik, yani doğal olanından da fazla bir kümeyle taşındığından çeyrek karede olan zaman sorunlarının neredeyse hiçbiri tam karede yaşanmaz dersek abartmış olmayız. Yani MIDI kablo uzunluğu veya kalitesi, fazla veri içeren uygulamalar tam kare zaman kodu için bir sorun değildir. Ancak bunun yerini, kaynağında MMC'nin olduğu başka sorunlar alır ki maalesef bu sorunların kökeni de zaman kodunun SysEx ile olan ilişkisinde, yani yapısında gizlidir.

MMC'deki gerçek yapısal sorun, MMC ile harekete geçen komut ve zaman koduna dayalı eşgüdümün SysEx mesajı ile taşınmasındadır. Eşgüdümü *hareket komutları* ve *zaman kodu* etkileşimiyle ayrı iki koldan yönetmek, her bir komut ve kod için neredeyse 1 Byte kullanmak MMC'nin eşgüdümde sağladığı en büyük avantaj gibi görülebilir. Ancak bu mesajların kendine özel yapısıyla değil de SysEx mesajı içine gömülerek gönderilmesi, donanımlar arasında uyumsuzluğu ortaya çıkarır ve işte MMC eşgüdümlü zaman koduna ait sorunların kökeninde de bu durum yatar. Aynı marka veya farklı marka ama birbirlerini desteklediği uluslararası bildirgelerde açıklanmış donanımlardan oluşan bir sistem için MMC ile sağlanan eşgüdümde neredeyse sorun çıkmazken; çok farklı marka ve

modellerin kullanıldığı sistemlerde MMC ile kurulan eşgüdümde zaman kodu sorunları hat safhaya varabilir. Çünkü her ne kadar MMC veya MTC Tam Kare mesajları SysEx içinde firma bilgisi yerine sabit bir kod taşısa da, özellikle zaman kodlarında mesajın SysEx içinde gönderilmesi, mutlaka karşı donanımlarda bir marka/model sorgulamasını tetikler. Örneğin ProTools destekli donanımlar kendi içlerinde sorun yaşamazken, ProTools tarafından açıkça desteği belirtilmeyen Tascam veya Mackie gibi dijital masalar sisteme girdiğinde MMC ile sorunlar ortaya çıkabilir.

MTC ve MMC bu yapılarıyla yukarıda sözü edilen bir takım sorunları ortaya çıkarmakla birlikte, bu sorunlara karşı üretilen diğer bir takım teknolojiler de ön plana çıkıyor. Örneğin bunlardan biri HUI'dir. Mackie ve Digidesign firmalarının başlarda otomasyon amaçlı yola çıktıkları HUI, şimdilerde MTC ve MMC kaynaklı sorunları aşmak adına eşgüdüm için uygulamaya konulan bir başka yapı. Ancak yine de HUI'yi aslında bütünüyle bir otomasyon sisteminin adı olarak düşünmek gerektiğinden, yapısıyla değil uygulamasındaki MMC amaçlı eşgüdümle bu noktada anımsamakta fayda var.

İster MTC olsun ister MMC veya HUI, bu türden eşgüdüm teknolojilerinin aslında sorunsuz üretildiklerini düşünmemek gerek. Bir kullanıcı olarak elbette sorun denildiğinde ilk akla gelen bağlantılar veya ilgili ayarlar olur ancak burada anlatılmaya çalışılan aslında pek çok eşgüdüm sorununun doğrudan eşgüdüm tipinden de kaynaklanıyor olabileceğidir. Dolayısıyla özellikle MTC ve MMC mimarisi, yapılarındaki özel durumlar nedeniyle özellikle günümüzde birçok eşgüdüm sorunlarının kökenidir.

KAYNAKÇA

- Durmaz, S. (2000), MIDI, Dokuz Eylül Yayınları, İzmir
- Huber, D. M. (2007), *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*, 3rd ed., Focal Press, Burlington, USA
- Işıkhani, C. (2009), Eşgüdüm, *Sound Dergisi*, Mart 2009 Sayısı, s. 48-51.