

Bir TTCAN Sisteminde Matris Çevrimi Tasarımı: PSA Benchmark Örneği

Süleyman Halil Temel¹, Mahmut Tenruh²

¹ Teknik ve Endüstri Meslek Lisesi, Bilişim Teknolojileri Alanı, Muğla

² Muğla Üniversitesi, Teknik Eğitim Fakültesi, Elektronik Bilgisayar Eğitimi Bölümü, Muğla
haliltemel@gmail.com, tmahmut@mu.edu.tr

Özet: Denetleyici Alan Ağı (CAN: Controller Area Network) ilk olarak araç içi, Elektronik Kontrol Üniteleri (ECU) arası haberleşme ihtiyacını gidermek ve aynı zamanda kablolama karmaşıklığını ortadan kaldırmak amacıyla geliştirilmiştir. CAN protokolü, düşük maliyeti ve yüksek performansı sayesinde geniş uygulama alanları bulmuştur. CAN protokolünde iletişim genellikle olay tetiklemelidir. Mutlak gerçek zamanlı bir işlemin gerçekleştirilmesinde verinin gerekli zaman sınırı içerisinde iletilmesi şarttır. Bu zaman şartının karşılanamaması halinde sistemde ciddi problemler ortaya çıkabileceği için, TTCAN (TTCAN: Time-triggered CAN) protokolü kullanılmaya başlanmıştır. TTCAN protokolünde, iletişim zaman tetiklemeli olarak gerçekleştirilmektedir. TTCAN periyodik olarak üretilen mesajların iletilmesinde, Matris Çevrimi (Matrix Cycle) kullanılır. Matris çevrimi tasarlanırken, kullanılan yöntemlerden birisi Azaltılmış Matris Çevrimi yöntemidir. Bu çalışmada Azaltılmış Matris Çevrimi yöntemiyle, PSA Benchmark mesaj seti için Matris Çevrimi geliştirilmesi amaçlanmıştır. Bu yöntem kullanılarak geliştirilen Matris Çevriminde, mesajlar orijinal periyot değerleriyle matris içerisine sorunsuz bir şekilde yerleştirilebilmiştir. Bu sayede mesajlar, gecikme olmadan iletebilmiştir.

Anahtar Sözcükler: Denetleyici Alan Ağı (CAN), Zaman Tetiklemeli CAN, Mutlak Gerçek Zamanlı Sistemler, PSA Mesaj Seti, Azaltılmış Matris Çevrimi

Designing a Matrix Cycle for a TTCAN System : PSA Benchmark Case

Abstract: Controller Area Network (CAN) was initially developed to provide in-vehicle communication among Electronic Control Units (ECUs) and also to remove complexity of wiring. CAN protocol has a wide application area due to its low-cost and high performance. In CAN protocol, communication is generally event-triggered. In a hard real-time task, the time requirement of a message delivery is critical since the messages have to be delivered in their deadlines. In such tasks, missing a deadline may cause a major failure in the system. Because of this, TTCAN (Time-triggered CAN) has been developed. In TTCAN, communication is realized in a time-triggered manner. TTCAN uses a Matrix Cycle in communication of periodical messages. One way of designing a Matrix Cycle is the use of Reduced Matrix Cycle method. In this study, it is aimed to develop a Matrix Cycle for PSA Benchmark with Reduced Matrix Cycle method. In the Matrix Cycle that has been developed by this method, messages can be placed with their original period values. Thus, messages can be transmitted without delays.

Keywords : CAN, TTCAN, Hard Real Time Systems, PSA Message Set, Reduced Matrix Cycle, Message Scheduling

1.Giriş

Otomobil endüstrisinde araç içi haberleşmeye artan talep CAN (CAN : Controller Area Network) ve diğer araç içi haberleşme ağlarının gelişmesinin sebeplerindedir. Burada esas amaç, araç içerisindeki elektrik tesisatındaki karmaşıklığın azaltılmasıdır. Araç içerisinde kullanılan kablo uzunluğu toplam 2 km'ye ulaşmakta, ağırlığı ise 100 kg'ı aşmaktadır [1]. Bu karmaşık yapı, tesisatın döşenmesini zorlaştırmakta ve maliyetin artmasına sebep olmaktadır. Otomobil endüstrisiyle ilgili şirketler, bir haberleşme protokolü geliştirmek için 1980'lerde araştırma çalışmalarına başlamışlardır. Değişik ihtiyaçlar göz önünde tutularak yapılan çalışmalar sonunda değişik üreticiler tarafından çok sayıda haberleşme protokolü geliştirilmiştir. Bunlardan bazıları CAN, VAN, J1850, ABUS ve CCD protokolleridir.

Bu protokollerden en önde gelenleri arasında Robert Bosch GmbH tarafından geliştirilen Denetleyici Alan Ağı (CAN) protokolü görülmektedir. Bosch, bu protokolün geliştirilmesi ve silikon entegre üretimi için üretici firma Intel ile işbirliği yapmış ve Autobus protokolüne dayalı ilk standart silikon CAN ürünü 1989 yılında Intel tarafından

üretilmiştir. Diğer oto içi haberleşme ağları arasında CAN özellikle Avrupa'da en yaygın kullanılan haberleşme ağı olmuştur ve ilk kez 1990'ların başında otomobillerde kullanılmaya başlanmıştır [2]. CAN kullanımı araç içinde bulunan bütün elektrikli ve elektronik cihazların bir tek haberleşme hattı üzerinden kontrol edilmesini ve elektrik tesisatındaki karmaşık yapının ortadan kaldırılmasını sağlamıştır. CAN protokolünün uygulama alanı, yüksek performansı, düşük maliyeti ve çok sayıda yarı-iletken üreticisi tarafından üretilmesi nedeniyle sadece otomotiv endüstrisiyle sınırlı kalmamış, diğer endüstri alanlarında da yaygın olarak kullanılmaya başlanmıştır [3]. Günümüzde CAN otomotiv endüstrisiyle birlikte aşağıdaki alanlarda da kullanılmaktadır.

Yolcu arabaları [4], kamyonlar, uçaklar [5], trenler [6], gemiler [7], zirai makineler [8], programlanabilir mantıksal kontrol (PLC) uygulamaları, robot kontrolü [9,10], akıllı motor kontrolü, akıllı algılayıcı ve uygulayıcılar, laboratuvar otomasyonu, tekstil makineleri, bina otomasyonu, asansörler, alışı-veriş makineleri, oyuncaklar, mekanik araçlar ve endüstriyel otomasyon.

2. CAN Sistem Özellikleri

CAN, taşıyıcı algılamalı çoklu erişim yöntemiyle birlikte çarpışma denetimli ve çözümlenmeli yöntemi kullanarak iletişim ortamına erişim sağlar (CSMA/CD+CR: Carrier Sense, Multiple Access/Collision Detection with Collision Resolution) [11].

Veri iletişimi bütün birimlerin bağlı bulunduğu iletişim hattı üzerinden, verinin gönderilmesi ile gerçekleşir. Bütün birimler gönderilen veriyi kontrol eder, sadece ilgili birimler veriyi kaydeder ve kullanır.

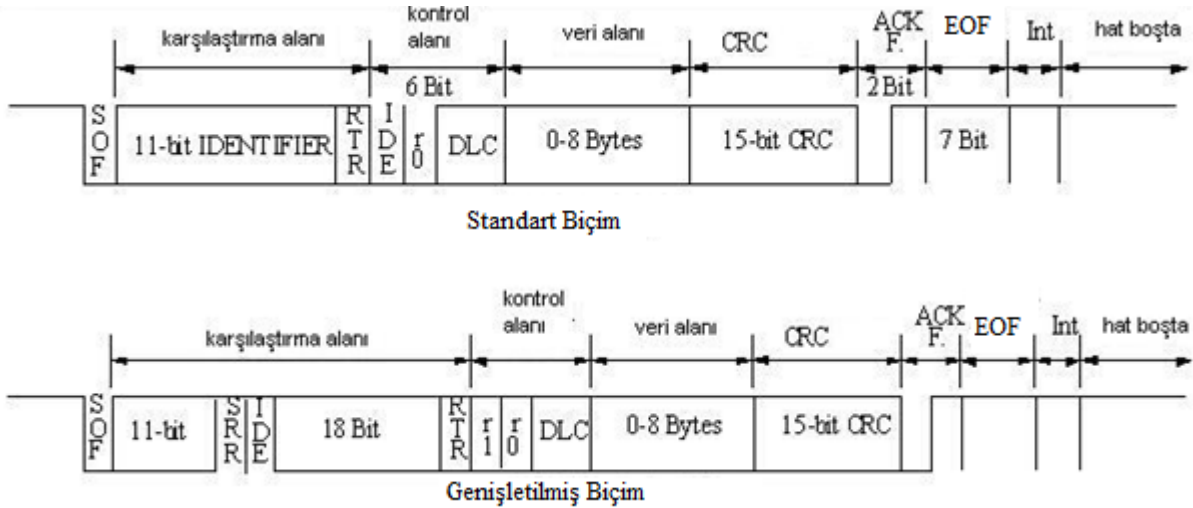
Bir kontrol birimi, yeni bir veri ortaya çıktığında, bunu haberleşme hattı üzerinden göndermeye başlar. İletişim genellikle olay tetiklemelidir (event-triggered). CAN protokolünde iletişim, öncelik eleme yöntemiyle sağlanır. Bu yöntem yüksek öncelikli mesajlar için gecikmesiz iletişim sağlar. Ancak, daha düşük öncelikli mesajlar, uzun gecikmelere maruz kalabilir. Bu nedenle gerçek zamanlı sistemlerde büyük öneme sahip gecikme sınırının aşılması riski ortaya çıkabilir. Mutlak gerçek zamanlı (Hard Real-time) işlemlerin gerçekleştirilmesinde verinin gerekli zaman sınırı içerisinde iletilmesi şarttır. Bu zaman şartının karşılanamaması halinde sistemde ciddi problemler ortaya çıkabilir. Diğer taraftan, toleranslı gerçek zamanlı (Soft Real-time) işlemlerin gerçekleştirilmesinde zaman şartına uyulması istenir, ancak bunun yerine getirilememesi halinde kritik sonuçlar ortaya çıkmaz.

CAN protokolünde olay tetiklemeli mesajların yanında, eşit zaman aralıklarında tekrarlanan ve zaman tetiklemeli olarak ortaya çıkan mesajlar da bulunabilir.

CAN bütün iletişim sistemi üzerinde tutarlılık olması esasına göre çalışır. Bir CAN sistemi içerisinde bir veri gönderildiği zaman bütün birimler tarafından kabul edilir veya reddedilir. Bunun sebebi sistemin sahip olduğu hata önleme mekanizmasıdır. Veri iletişimi sırasında her hangi bir kontrol ünitesi bir hata tespit ederse derhal bir hata mesajı gönderir ve bütün birimler, iletilen veriyi iptal eder. Veri, gönderici tarafından yeniden gönderilir. CAN toplam beş çeşit hata önleme mekanizmasına sahiptir. Bu nedenle hatalı bir verinin tespit edilememesi ihtimali $p < 4.7 * 10^{-11}$ olarak ifade edilebilir ve bu da 1 milyon saatlik çalışma sırasında 1 hatanın fark edilememesinden daha düşük bir ihtimaldir [11].

2.1 Paket Biçimleri (Frame Formats)

CAN, Veri Paketi (Data Frame), Uzak Paket (Remote Frame), Hata Paketi (Error Frame) ve Aşırıyük Paketi (Overload Frame) olmak üzere 4 farklı pakete sahiptir. CAN protokollerinin iki farklı versiyonu vardır. İlk versiyon olan CAN 1.0 da veri paketinde ve uzak pakette tanıttıcı kısmı 11 bittir. Diğer versiyon CAN 2.0 ise CAN 2.0A ve CAN 2.0B olmak üzere ikiye ayrılmıştır. CAN 2.0 A versiyonu CAN 1.0 versiyonunun yerini almıştır. CAN 2.0B versiyonunda paket tanıttıcı kısmı 29 bit olarak belirlenmiştir. CAN 2.0B versiyonu CAN 1.0 ve CAN 2.0A biçimlerini destekler. Tanıttıcı uzunluğuna göre veri ve uzak paketler Standart ve Genişletilmiş olmak üzere ikiye ayrılır [11].



Şekil 1. Standart ve Genişletilmiş CAN çerçeveleri

2.2 Zaman Tetiklemeli CAN (TTCAN)

TTCAN, CAN yapısının değişmeden, üzerine eklenen bir katman olarak tanımlanabilir [12]. Bu üst seviye protokol, global bir sistem saati sağlayarak bütün istasyonların iletişimini senkronize eder. Bu protokol yüksek güvenilirlikli uygulamalar için geliştirilmiştir. Fren, direksiyon gibi otomobillerde güvenliğin önemli olduğu ve bu gibi mekanik, hidrolik sistemlerin yerini elektronik sistemlerin aldığı uygulamalarda kullanılmak üzere tasarlanmıştır. TTCAN mutlak gerçek zamanlı sistemlerde yani mesajın doğru olarak iletilmesinin yanında tam zamanında iletilmesinin de önemli olduğu sistemlerde kullanılır. Klasik CAN ağlarında eğer mesaj iletilirken bir hata meydana gelmişse, mesaj tekrar iletilir. Bu durum TTCAN uygulamaları için geçerli değildir, böyle bir durumda kontrol uygulaması bilgilendirilir.

Olay tetiklemeli sistemden farklı olarak, zaman tetiklemeli sistemde mesaj iletimi bir zaman dizisiyle gerçekleştirilir ve bütün istasyonlar senkronize edilmiş, global saate ayarlanmıştır. Her mesajın kendine ait zamanı olduğu için herhangi bir çarpışma meydana gelmez. Bütün mesajların zamanlaması çalışma zamanından önce statik zaman planlaması olarak gerçekleştirilir. Bütün sistem mesajları zamanlarının planlanması, sorunsuz bir sistem meydana getirir. Ağdaki bütün istasyonların ortak zaman bilgisi vardır ve her istasyon ağ zaman planlamasını bilir.

CAN dünya çapında yaygın bir otomobil ağ yapısı olmasına rağmen güvenliğin kritik olduğu uygulamalarda, yani zaman tetikleme ihtiyacı olan durumlarda TTCAN tercih edilebilir.

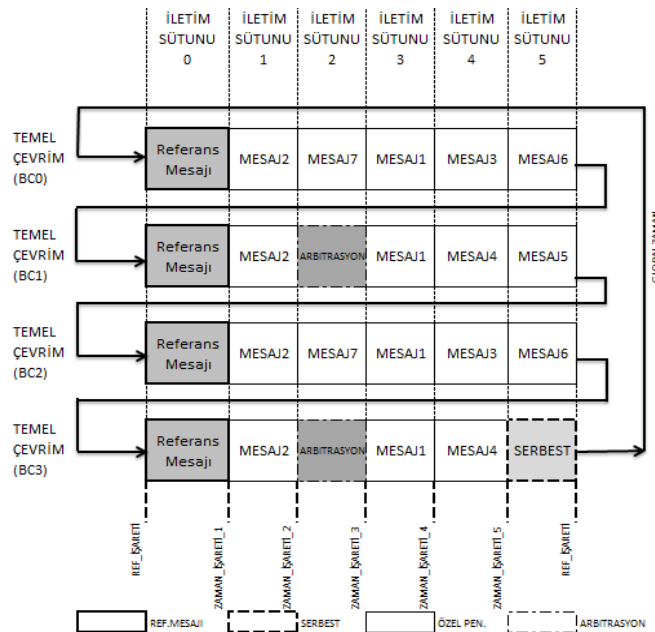
CAN mimarisinde OSI katmanlarından fiziksel katman ve veri iletim katmanı bulunur. TTCAN klasik CAN

katmanlarına, zamanlamayı gerçekleştirmek için oturum (Session) katmanı ekler.

Dağıtılmış kontrol uygulamalarında TTCAN, CAN ağına, oturum katmanının biçimsel zamanlama ortamını kullanarak erişir. Böylece uygulama mesajlarının önceliğine göre yerleştirilmiş zaman pencereleri kullanılır. Zaman tetiklemeli CAN sisteminde mesaj gönderme ve G/Ç servisleri gibi sistem olayları periyodik olarak gerçekleştirilir. Sistemin bütün durum geçişleri önceden tanımlı ve matris çevrimi (Matrix Cycle) olarak bilinen bir zamanlama çizelgesi ile ilişkilendirilir. Olay tetiklemeli sistemde, sistem durum geçişlerinin önceden tanımlı zamanları yoktur, dolayısıyla bütün mesajlar en kötü ihtimalle kararlaştırma (arbitrasyon) durumunda kalabilir ve sonuç olarak uzun beklemler olabilir.

TTCAN güvenilir mesaj zamanlamasının yanında, mevcut bant genişliğinin etkili kullanımını da sağlar. Güncel olay tetiklemeli CAN uygulamalarında ağdan yararlanma göreceli olarak düşüktür. Kritik olmayan uygulamalar için ağdan yararlanma %50 seviyesindedir ve mutlak gerçek zamanlı sistemler için ağdan yararlanma, %20- %30 civarındadır. Bu oran mutlak gerçek zamanlı sistemler için uygundur, çünkü hata olması durumunda mesajın yeniden gönderilmesi gerekir. TTCAN ağ bant genişliğinden %90 seviyesinde yararlanır çünkü hata durumunda mesaj tekrar iletilmez, bir sonraki zaman penceresinde iletilir.

TTCAN hata kontrolünü iki ayrı basamakta gerçekleştirir. Birincisi CAN standardında olduğu gibi, fiziksel ve veri iletim katmanında yapılır, ikincisi ise zaman planlama hatalarına odaklanan oturum katmanında yapılır.



Şekil 2. TTCAN Matris Çevrimi

2.2.1 TTCAN Matris Çevrimi

Zaman tetiklemeli çizelge belirli sırada zaman pencerelerinden oluşur. Her mesaj penceresinde bir mesaj iletilir. Bütün olarak geçiş sırası matris çevrimi olarak bilinir ve ağ normal olarak çalışırken periyodik olarak tekrar eder. Matris çevrimi, bir referans mesajıyla başlayan ve bir sonraki referans mesajıyla son bulan temel çevrimlerden (BC: Basic Cycle) meydana gelir. Şekil 2'de 4 temel çevrimden oluşan örnek bir matris çevrimi görülmektedir. Matris çevrimi, bütün ağ için geçerli olan bir zamanlama yapısını temsil eder.

TTCAN protokol terminolojisinde zaman işareti (time-mark) zaman içerisinde pencerelerin başlangıç ve bitişindeki belirli bir noktayı işaret eder.

Ağda bulunan istasyonlar, bir pencere süresince tam iletim kapasitesine sahip olarak zaman pencerelerine atanırlar [13]. Her istasyon, mesaj paketini temel çevrim biçiminde gönderir. Bütün temel çevrimler zaman noktasında eşit büyüklüktedir, ancak değer noktasında gönderilen paketlerin içeriğine ve uzunluğuna bağlı olarak farklılık olabilir. Bir temel çevrim tamamlandığında, aynı erişim biçimine fakat farklı paket sırasına sahip bir sonraki temel çevrim başlar. Temel çevrimlerin sayısı (0-63 arasında) matris çevriminin uzunluğunu belirler. Matris çevrimi tamamlandığında, iletim matris çevriminin tekrarı biçiminde devam eder.

Matris çevrimini oluşturan zaman pencereleri üç farklı tipte olabilir. Özel zaman pencereleri (Exclusive Time Windows) belirli bir mesaja atanırlar. Kararlaştırma zaman pencereleri (Arbitration Time Windows) birden fazla mesaja atanırlar ve hatta meydana gelen karşılaşmalar, CAN öncelik kararlaştırma yöntemiyle çözülür. İki veya daha fazla kararlaştırma penceresi peş peşe açılabilir, bu durum kararlaştırmaya tabi mesajların iletiminde geniş zaman aralığı sağlar. Serbest Zaman Pencereleri (Free Time Windows) zaman aralıklarında herhangi bir mesaj içermeyen pencerelerdir. Ağın gelecekteki genişlemesi düşünülerek ayrılmışlardır.

3. PSA Benchmark için, Azaltılmış matris yöntemiyle Matris Çevriminin geliştirilmesi

PSA benchmark 1997 yılında Peugeot-Citroen tarafından araç içi ağlar düşünülerek geliştirilmiştir. Bu benchmarkta CAN ağ yapısında 5 birim bulunur.

Bu birimler:

Motor Denetleyici, Otomatik Vites Kutusu Denetleyici (AGB), Kilitlemesiz Fren Sistemi ve Araç Hareket Denetleyici (ABS/VDC), Askılama Denetleyici, Teker Açık Algılayıcı ve Hareketli Far Doğrultucudur (WAS / DHC) [14].

3.1 Matris Tasarımı

PSA benchmark mesaj setinde 12 adet mesaj bulunmaktadır. Mesajların periyodları ve veri uzunlukları Tablo 1'de gösterilmiştir.

Tablo 1. PSA Benchmark Mesaj Seti

Öncelik (ID)	İleten Nokta	Veri Uzunluk Kodu (DLC) Byte	Periyot (ms)
1	Motor Denetleyici (Engine Controller)	8	10
2	Teker Açık Algılayıcı (Wheel Angle Sensor)	3	14
3	Motor Denetleyici (Engine Controller)	3	20
4	Otomatik Vites Kutusu (AGB)	2	15
5	Anti Blok Sistemi (ABS)	5	20
6	Anti Blok Sistemi (ABS)	5	40
7	Anti Blok Sistemi (ABS)	4	15
8	Bodywork Gateway	5	50
9	Aygıt y	4	20
10	Motor Denetleyici (Engine Controller)	7	100
11	Otomatik Vites Kutusu (AGB)	5	50
12	Otomatik Vites Kutusu (AGB)	1	100

Matris Çevrimi (MC) tasarımı yapılırken, 4 adet Temel Çevrimden (BC) oluşan Matris Çevriminin toplam süresi 40 ms olarak ve her Temel Çevrim süresi 10 ms olarak planlanmıştır.

Azaltılmış Matris Çevrimi yöntemi uygulanırken, Matris Çevriminin (MC) toplam süresi ile her mesajın periyodunun ayrı ayrı Ortak Katlarının En Küçüğü (OKEK) hesaplanır. Daha sonra her mesaj için tekrar sayısı (T_s), her mesaj için bulunan OKEK'in, mesaj periyoduna (P_m) bölünmesiyle bulunur [15]. Bu konuda farklı yöntemler kullanılarak tasarlanan matris çevrimleri de bulunmaktadır [17, 18, 19, 20].

Mesajların matris çevrimindeki tekrar sayıları hesaplanırken, Matris çevriminin toplam süresi olan 40 ms ile mesajın periyodunun OKEK'i hesaplanmıştır. Bulunan değer (OKEK) mesajın periyoduna (P_m) bölünerek, tekrar sayısı (T_s) hesaplanmıştır. Buna göre hesaplanan değerler Tablo 2'de gösterilmektedir.

Tablo 2. Matris Çevrim Değerleri

Mesaj No	P_m	OKEK ($P_m ; 40$)	T_s (OKEK / P_m)
M1	10	40	4
M2	14	280	20
M3	20	40	2
M4	15	120	8
M5	20	40	2
M6	40	40	1
M7	15	120	8
M8	50	200	4
M9	20	40	2
M10	100	200	2
M11	50	200	4
M12	100	200	2

Mesajların tekrar sayıları (Ts) hesaplandıktan sonra, toplam tekrar sayısı 63 olarak bulunmuş ve Matris Çevriminin 64 pencereden oluşması planlanmıştır. Her temel çevrimin de 16 pencereden oluşması planlanmıştır. Altmış dördüncü (64.) pencerede tanımlı bir mesaj olmadığı için bir pencere Serbest (Free) olarak bırakılmıştır. Daha sonra mesajların bulunan sayıda, Matris Çevrimine yerleştirme planlaması yapılmıştır. Tasarımı yapılan Matris Çevrimi Şekil 3'de görülmektedir. Mesaj pencerelerinin uzunluklarının hesaplanmasında oluşabilecek maksimum mesaj boyutu dikkate alınmıştır [14].

CAN çerçevesinin iletilmesinde senkronizasyon, bit değerinin değişimleri sırasında sağlanır ve senkronizasyonun korunması için peş peşe en fazla 5 adet bitin aynı seviyede iletilmesine izin verilir. Aynı seviyede daha fazla bit olması durumunda, zıt yönde bir altıncı (6.) bit eklenir. Bu durum mesaj boyutunun uzamasına neden olur. Alıcı tarafta bu fazla bitler mesajdan atılır. Bu durumda maksimum mesaj boyutu (C_m) aşağıdaki şekilde hesaplanır [14,16].

$$C_m = \left(\left(\frac{34+8sm}{4} \right) + 47 + 8sm \right) Tbit$$

Burada ;

34 bit: Senkronizasyon için bit ekleme yapılabilecek, sabit bit sayısını göstermektedir;

8sm: Byte olarak uzunluğu verilen veri boyutunun bit olarak hesaplanmasını sağlar.

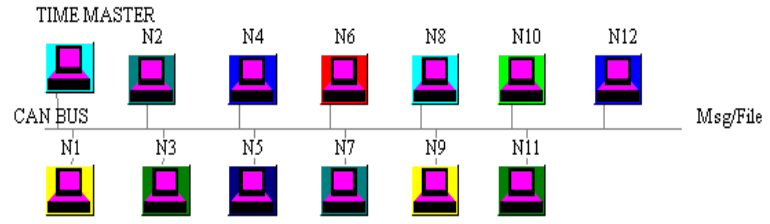
4 bölen değeri: Senkronizasyon için eklenebilecek maksimum bit sayısının elde edilmesini sağlar.

47 sabiti: Standart CAN çerçevesinde veri alanı olmadan bulunan sabit bit uzunluğunu ifade eder.

Tbit: Hattın bit zamanını ifade eder. Örneğin; 1 Mbps için $1\mu s$, 500 Kbps için $2\mu s$

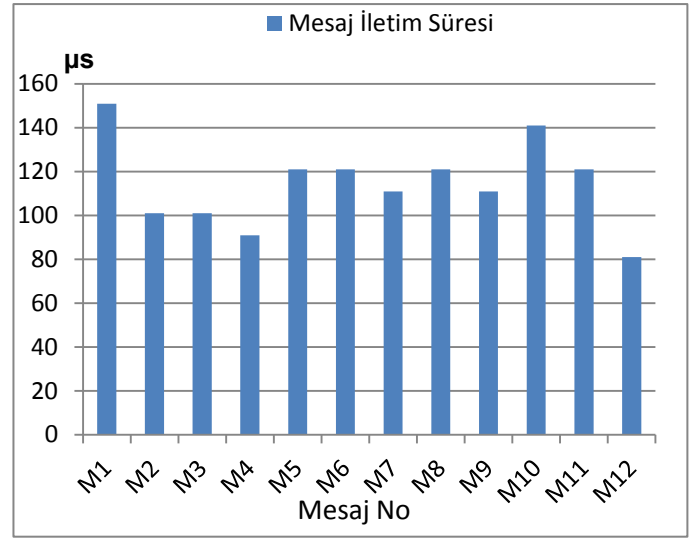
Her mesaj penceresinin başlangıcında 16 bit uzunluğunda bir Tx aralığı, mesajın başlama aralığı olarak bulunur. Pencere boyutunun hesaplanmasında Maksimum mesaj uzunluğuna (C_m) Tx değeri eklenir.

Matris Çevriminde (Şekil 3), mesajların dağılımı yapılırken pencereler arasında, eşit boşluklar olması planlanmıştır. Daha sonra, mesaj yerleşimleri yapılırken, pencere boyutlarındaki taşmaların önlenmesinde, bu boşluklar tasarım esnekliği sağlamaktadır.



Şekil 4. Simülasyon Modeli

Şekil 4'de Network II.5 simülasyon programında oluşturulan ve geliştirilen matris çevriminin (Şekil 3) uygulandığı model görülmektedir.



Şekil 5. Mesaj İletim Süreleri

Yapılan tasarım sonucunda elde edilen matris çevriminin simülasyon modeline uygulanması sonucunda mesaj iletim sürelerinde herhangi bir bekleme veya sıralama gecikmesinin olmadığı görülmüştür. Her mesaj kendi penceresi içerisinde, kendi periyodu ile ortaya çıktığı anda gecikmeksizin iletilebilmiştir.

İletim süresinde sadece mesaj boyutu ve hat üzerindeki iletim süresi belirleyici etkindir. Bununla ilgili değerler Şekil 5. de görülmektedir.

Ref	M2	M4	M7	M2	M1	M5	M2	M8	M3	M4	M2	M7	M11	M2	M10
Ref	M2	M4	M7	M2	M1	M6	M2	M8	M12	M4	M2	M7	M11	M2	M9
Ref	M2	M4	M7	M2	M1	M5	M2	M8	M3	M4	M2	M7	M11	M2	M10
Ref	M2	M4	M7	M2	M1	Free	M2	M8	M12	M4	M2	M7	M11	M2	M9

Şekil 3. Tasarlanan Matris Çevrimi

4. Sonuç ve Öneriler

CAN protokolü öncelik eleme esaslı ortam erişim yöntemi kullandığı için düşük öncelikli mesajların iletiminde büyük gecikmelere sebep olabilir. Gerçek zamanlı sistemlerde mesajların verilen süre içerisinde iletilmesi gerekir. Bu çalışmada periyodik mesajların gecikmesiz iletimini sağlamayı amaçlayan bir TTCAN modeli geliştirilmiştir.

TTCAN protokolünün zaman tetiklemeli iletişimde kullanılması ve matris çevrimi kullanması dolayısıyla periyodik mesajların gecikmesiz iletilmesi sağlanmaktadır. Bu çalışmada PSA Benchmark mesaj seti için matris çevrimi geliştirilmiştir. TTCAN protokolünde matris çevrimi yapılırken azaltılmış matris çevrimi yöntemi kullanılmış ve bir matris çevrimi tasarlanmıştır. Tasarlanan bu matris çevrimi ile mesajlar pencere sürelerinde çakışma/taşma olmadan ve herhangi bir gecikme olmadan iletebilmiştir. Matris çevriminin simülasyonu ile çakışma olmadan mesajların gecikmesiz iletebildiği görülmüştür.

5. Kaynaklar

- [1] Lawrenz, W., **CAN System Engineering From Theory to Practical Applications**, Springer-Verlag Inc., New York, 1997.
- [2] "Introduction to In-Vehicle Networking", **MCS96 papers**, Intel Corporation, 1998.
- [3] "CAN Sales Figures", **Press Releases, CAN in Automation (CiA)**, Germany, 1999. www.can-cia.de/NP.htm
- [4] Wenkebach, U., Reckels, B., "System Concepts for Serial Data Communication in Cars", **Proceeding of the Institution of Mechanical Engineer, Seventh International Conference Automotive Electronics**, pp. 203-212, 9-13 Oct. 1989, London.
- [6] M. Stock, "Higher-Layer Protocol for Avionics", **CAN Newsletter**, pp 28, 30, 32, 34, September 1999.
- [7] "CAN-based Computer System for Railways" **CAN Newsletter**, p. 66, September 1998.
- [8] "NMEA 2000: CAN in Marine Electronics", **CAN Newsletter**, pp. 52-53, June 1999.
- [9] Hofstee, J. W., Goense, D., "Simulation of a CAN-based Tractor-Implement Field Bus According to DIN9684", **J. Agric. Engng. Res.**, pp. 89-100, 1997.
- [10] Fredriksson, L. B., "Distributed Embedded Control Systems in Robotics", **KVASER AB**. Sweden. www.kwaser.se/dcs/decs.htm
- [11] **CAN Specification, Version 2.0**, Robert Bosch GmbH, Germany, 1991.
- [12] Leen G., Heffernan, D., TTCAN: a new time-triggered controller area network. **Microprocessors and Microsystems** 2002;26(2):77-94
- [13] Leen, G., Heffernan, D., A time-triggered control network for industrial automation, *Assembly Automation*; 2002; 22, 1; **ABI/INFORM Global** pp. 60-68
- [14] Navet, N., Song, Y.Q., Simonot, F., "Worst-case deadline failure probability in real-time applications distributed over controller area network". **Journal of System Architecture** 2000;46(7):607-17
- [15] Tenruh, M., "Message Scheduling with Reduced Matrix Cycle and Evenly Distributed Sparse Allocation for Time-Triggered CAN," **Journal of Network and Computer Applications**, Vol. 34, pp. 1240-1251, 2011.
- [16] Tindell, K., Burns, A., "Guaranteed message latencies for distributed safety-critical hard real-time control networks". **Technical report YCS229**, Department of Computer Science, University of York, UK, 1994.
- [17] Ding, S., Xie, Z., Yin, X., "A GA-based systematic message scheduling method for time-triggered CAN". **IEEE/IFIP international conference on embedded and ubiquitous computing**, 2008.
- [18] Xiao, T., Li, X., Tan, X., Zhou, X., "Real-time dynamic scheduling algorithm for TTCAN and its realization". **3rd international conference on advanced computer theory and engineering (ICACTE)**, 2010
- [19] Zhu, Z., Sui, J., Yang, L., "Bin-packing algorithms for periodic task scheduling". **2010 WASE international conference on information engineering (ICIE)**, 2010
- [20] Schmidt, K., Schmidt, E.G., "Systematic message schedule construction for time-triggered CAN". **IEEE Transaction on Vehicular Technology** 2007;56(6): 3431-41.