

# Yazılım Tabanlı MIL-STD-1553B Veri Yolu Simülasyonu Üzerinden Sensör Verilerinin Aktarılması Transmitting Sensor Data Over MIL-STD-1553B Data Bus Interface Simulation

Ozan Özgür<sup>1</sup>, Esra Açıan<sup>1</sup>, Yetkin Aksoy<sup>1</sup>, Halit Oğuztüzün<sup>1</sup>

<sup>1</sup>Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği

e177703@metu.edu.tr, e177607@metu.edu.tr, e189212@metu.edu.tr, oguztuzn@ceng.metu.edu.tr

## Özet

Bu çalışmada Data Distribution System (DDS) üzerinde çalışan MIL-STD-1553B Veri Yolu simülasyonu için gömülü sistem geliştirilmiştir. Ayrıca sistemin geliştirilme, doğrulanma ve performans analizi aşamaları kapsamında kısıtlamaları ve değerlendirmeleri içeren bilgi sunulmaktadır. MIL-STD-1553B veri yolu genellikle aviyonik sistemlerde, uçak ve uzay sistemlerinde veri ve bilgi akışını sağlamak amacıyla kullanılmaktadır. Bu kapsamda gerçekleştirilen veri yolu simülasyonunda gömülü hız-hareket sensörleri kullanılmış, bu sensörleri veri yoluna bağlayacak arayüzler ve veri akışını izleyen gözlem uygulamaları geliştirilmiştir. Ayrıca sistemlerin geliştirilme sürecinde her aşamada doğrulama ve geçirme etkinlikleri yürütülmüştür. Yürütülen bu etkinlikler sonucunda performans testleri gerçekleştirilmiştir.

## Abstract

This paper presents an experimental embedded system set up which is proposed to run on a simulation of MIL-STD-1553B Serial Data Bus interface that uses a protocol based on the Data Distribution Service (DDS) middleware standard. In addition, this paper describes the implementation and verification details, constraints and assessment of the system by evaluating the performance results. The MIL-STD-1553B serial data bus is widely used for data and information exchange in avionics, aircraft and space systems. Therefore, monitoring interfaces and implementation of embedded accelerometer and gyroscope sensors are adapted to the embedded system to demonstrate the actual use of the bus. Verification activities were conducted in every step and the implemented system was evaluated.

## 1. Giriş

MIL-STD-1553B standardı, 1970'lerde Amerika Savunma Bakanlığı tarafından yayımlanmış, askeri araçlarda kullanılacak seri veri yolunun mekanik, elektronik ve fonksiyonel karakteristiklerini belirleyen bir askeri standarttır. Başta askeri havacılık kapsamında kullanılan bu veri yolu, sivil havacılık sektörlerinde de kullanılan bir standart haline gelmiştir.

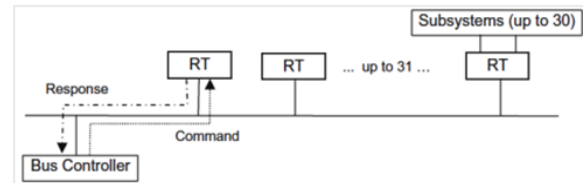
Veri yolu standardının yayımlanmasının ardından havacılık sektöründe bu veri yolunun kullanımı kısa sürede yayılmış, veri yolunu kullanan sistemler hızla geliştirilmiş ve zamanla daha karmaşık hale gelmiştir. MIL-STD-1553B veri yolunu kullanan karmaşık bütünleşmiş sistemlerin ve bağlı alt sistemlerin testleri kapsamında özel gereksinimleri olan (kablolama, gömülü kodlar, vb.) bilgisayar donanımları kullanılmaktadır. Bu donanımların kullanımı, veri yolunun kullanılması ve veri yolu üzerinde sistem geliştirme işlemlerinin yapılmasında yüksek maliyetlere neden olmaktadır [4]. DDS Tabanlı MIL-STD-1553B Veri Yolu arayüzünün geliştirilmesindeki asıl amaç da bu geliştirme maliyetlerinin düşürülmesidir [5].

DDS Tabanlı MIL-STD-1553B Veri Yolu arayüzü, MIL-STD-1553B için kullanılan bilgisayar donanımlarının rolünü, entegrasyonun sistem entegrasyon laboratuvarı öncesi aşaması kapsamında üstlenmektedir.

Çalışmamızın amacı, daha önce gerçekleştirilmiş, DDS üzerinde çalışan MIL-STD-1553B Veri Yolu arayüzü simülasyonu ile çalışabilecek gömülü sistem donanımları geliştirmek, veri yolu üzerinde gömülü sistem aracılığıyla alınan sensör verilerinin aktarımını sağlamak, sınıma ve performans ölçüm faaliyetlerini yapmaktır. Çalışma kapsamında HIL (hardware-in-the-loop, döngüde donanım) simülasyonu denemeleri yapılacak, simülasyonun verimli ve kullanılabilir olup olmadığını irdelenirken, gelecekteki geliştirmelere ışık tutacak öneriler ve karşılaşılan eksiklikler de belirlenecektir.

## 2. Teknolojiler

### 2.1. MIL-STD-1553B Altyapısı: Genel Bakış



Şekil 1 – MIL-STD-1553B Veri Yolu altyapısı

MIL-STD-1553B Veri Yolu sistem yapısı Şekil 1’de gösterilmekte olup, aralarında ana/bağımlı ilişkisi bulunan birçok alt sistemden oluşmaktadır. Sistemde, Veri Yolu Denetçisi (Bus Controller, BC) ana alt sistem olarak her zaman bulunmalıdır. Sistemde birden fazla BC olmasına izin varken bu durum yalnızca yedekleme amaçlı olup, yalnızca bir BC aktif olarak yönetici pozisyonunda olabilir. En fazla 31 adet olmak üzere başka bağımlı alt sistemler Uzak Terminaller (Remote Terminal, RT) adı altında veri yolu kullanılarak birbirine bağlanabilmektedir [3]. BC ve RT’ler dışında sistemde, yalnızca günlük tutmakla görevli Veri Yolu Gözleyicileri (Bus Monitor, BM) bulunabilir.

MIL-STD-1553B veri yolu üzerindeki iletişim, sözcük paketleri aracılığıyla yapılmakta olup, standartta tanımlanmış –Komut Sözcüğü (Command Word), Durum Sözcüğü (Status Word) ve Veri Sözcüğü (Data Word) olmak üzere– üç adet sözcük yapısı bulunmaktadır [6].

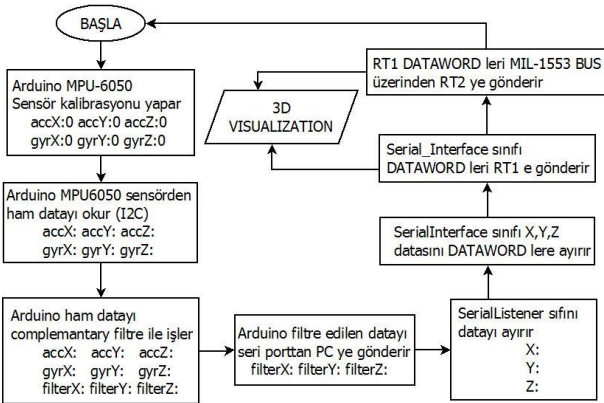
### 2.2. Veri Dağıtım Sistemi (DDS) Yapısı

DDS, Object Management Group (OMG) tarafından yayımlanmış orta katman yazılım standardıdır. Yayımcı/abone yapısında çalışan DDS, yüksek performanslı ve gerçek zamanlı veri iletişimini sağlama teknolojisidir. Genellikle dağıtık yapıdaki sistemlerin geliştirilmesinde ve tümleştirilmesinde kullanılır.

## 3. Tasarım ve Gerçekleştirme

### 3.1. Çalışma Kapsamı

Geliştirilen gömülü sistem MIL-STD-1553B arayüzünü kullanarak, MIL-STD-1553B standardı kapsamında oluşturulan veri yolu simülasyonu üzerinde çalışmaktadır. Algılayıcılar içeren bir gömülü donanım veri yoluna Uzak Terminal (RT) olarak bağlanmakta ve diğer tarafta yine RT olarak bağlı bir izleyici uygulamaya veri sağlamaktadır. Veri akışı, MIL-STD-1553B standardına uygun sözcükler kullanılarak gerçekleştirilmekte olup, sisteme Veri Yolu Denetçisi (BC) olarak bağlı bir yazılım ile yönetilmektedir.



Şekil 2 – Sistemde veri akışı

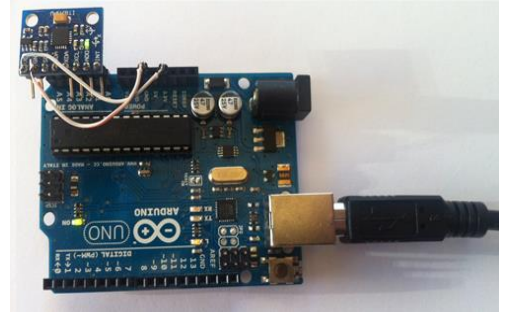
### 3.2. Donanım Altyapısı

#### 3.2.1. Geliştirme Donanımı ve Sensörler

RT cihazı olarak sistemde, üzerinde hız ve jiroskop sensörleri bulunan gömülü sistem kullanılmıştır. Bu kapsamda üzerinde

küçük bir bilgisayar bulunan Arduino UNO-R3 model uygulama geliştirme donanımı tercih edilmiştir [7]. Tercihin sebebi ise Arduino IDE ile gömülü yazılımların işletim sisteminden ve bilgisayar özelliklerinden bağımsız olarak geliştirilebilmesidir [8].

Arduino geliştirme donanımı ile birlikte, hareket verisini oluşturmak üzere MPU-6050 yongası kullanılmıştır. Bu yonga, üzerinde hem jiroskop hem de hız sensörleri bulunan ve I2C standardında veri akışı sağlayan bir yonga olarak yaygın şekilde kullanılmaktadır.

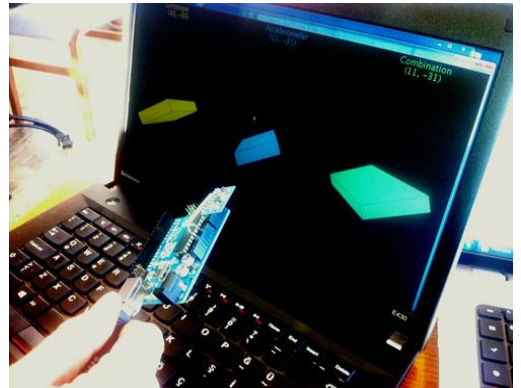


Şekil 3 – Arduino ve MPU-6050

#### 3.2.2. Gömülü Yazılım

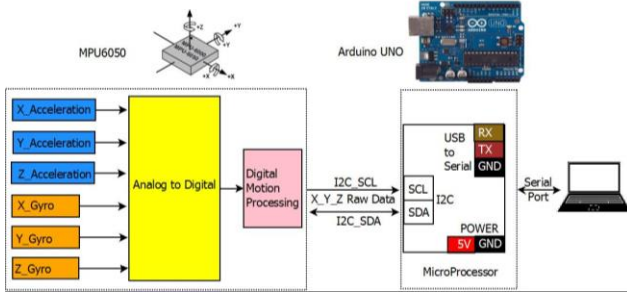
Arduino geliştirme donanımı üzerinde, Krodal tarafından geliştirilmiş MPU-6050 yongasından I2C kullanarak ham veriyi okuyan açık kodlu bir yazılım kullanılmıştır [9]. Bu kodla veri okunarak anlamlı açı değerlerine dönüştürülmektedir.

Jiroskop ve hız sensörlerinden alınan veriler, her zaman donanımsal eksiklikler nedeniyle hatalar içermektedir. Hataları ayıklamak, doğru ve kesin bilgiye ulaşabilmek için genellikle Kalman Filtresi veya Tümler Filtre (Complementary Filter) kullanılmaktadır. Bu çalışmada daha basit ve yine de yeterince doğru veri sağlayan Tümler Filtre kullanılmıştır [10].



Şekil 4 – Sensör verisinin “Processing Tool” kullanılarak görselleştirilmesi

Çalışmanın donanım bölümünde hız ve jiroskop sensörlerinden alınan veriler filtrelenerek döngüsel açı değerlerinde çevrilmekte ve bu veriler de CSV formatında seri port kullanılarak 19200 baud hızında simülasyona aktarılmaktadır.



Şekil 5 – Donanım Altapısı

### 3.3. Yazılım Tasarımı ve Gerçekleştirimi

MIL-STD-1553B simülasyon arayüzünün (API) Java arayüzü olması nedeniyle çalışmanın yazılım bölümü tamamıyla Java programlama dili kullanılarak geliştirilmiştir [2].

Çalışma kapsamında geliştirilen araç ve sınıflar şunlardır;

#### 3.3.1. Seri Dinleyici

Seri Dinleyici (*SerialListener*) sınıfı, oluşturulan simülasyon ortamının veri giriş noktasıdır. JAVA RXTX Kütüphanesi kullanılarak oluşturulan bu sınıf, seri porttan aldığı açılı verilerini kontrol ederek –PI ve PI aralığında kayan noktalı sayılara dönüştürür ve Serial Interface sınıfında bir fonksiyon çağırarak bu sayıları aktarır.

#### 3.3.2. Seri Arayüz

Seri Arayüz (*SerialInterface*) sınıfı, veri işleme ve veri sözcüklerini (Data Word) oluşturmakla görevlidir. Veri, MIL-STD-1553B veri yolu üzerinde yalnızca veri sözcüğü içerisinde var olabilir. Aynı şekilde DDS'te de veriler, paketler halinde daha önceden belirlenmiş veri tiplerinde yer alabilmektedir. Serial Interface sınıfı Serial Listener sınıfından aldığı verileri, simülasyonun giriş noktası olan RT'ye aktarmakla görevlidir. Bu amaçla X, Y, Z eksenleri üzerinde dönüş açıları olarak alınan veriler MIL-STD-1553B standardında veri sözcüğü olarak paketlenir ve RT sınıflarına iletilir.

#### 3.3.3. Uzak Terminaller

Uzak Terminaller (Remote Terminal, RT) veri yolu ile alt sistemlerin veya diğer veri yollarının bağlantısını sağlayan arayüzler sağlamaktadır. Bu çalışmada, veri yolu simülasyonuna bağlı olan iki adet RT bulunmaktadır.

MIL-STD-1553B standardı kapsamında her türlü mesaj akışı Veri Yolu Denetleyici (Bus Controller, BC) tarafından başlatılmaktadır. BC tarafından başlatılan RT'den RT'ye transferler; Kaynağı BC olan, RT'ye bağlı bir alt sistem olan veya Veri sözlüğünde BC'nin isteğine karşılık gelen her türlü veriyi taşıyabilmektedir [11].

Kullanılan simülasyon API'sinin en büyük eksikliği burada ortaya çıkmaktadır. Şöyle ki; kullanılan API versiyonunda bir RT, yalnızca kendisine BC tarafından gönderilen veriyi, veri yoluna tekrar gönderebilmektedir. Bağlı bir sistemin veya diğer veri yolunun veri akışına katılması mümkün olmamaktadır.

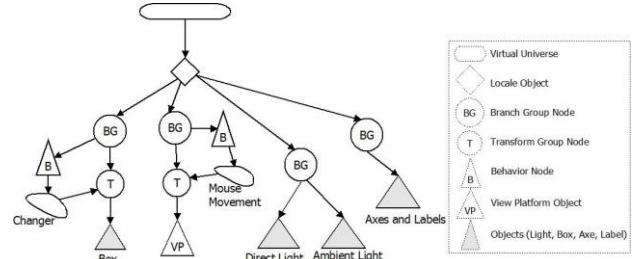
Çalışmanın devamı adına, API'de karşılaşılan bu probleme geçici bir çözüm getirilmiş, bu problemin giderilebilmesi için her veri akışının sanal bir BC tarafından başlatıldığı varsayılmış, RT'lerin dışsal değişkenler alabilen içsel bir metodu olan *sendDataWordToRT* sonsuz bir döngü içerisinde çağırılarak dışarıdan alınan verinin veri yoluna sürekli aktarılması sağlanmıştır.

Bu kapsamda, veri sözcükleri *RT\_1SerialInterface* sınıfına ait *sendDataWordToRT* metodu çağırılarak veri akışına dahil edilmiştir.

#### 3.3.4. 3D Görselleştirme Aracı

3D Görselleştirme Aracı (3D Visualization Tool) X, Y, Z eksenlerinde dönüş açıları olarak bunların görselleştirilerek okunabilirliğini arttırmak amacıyla geliştirilmiştir. Kullanılan animasyonlar Java 3D API kullanılarak oluşturulmuştur. Bu araç, 3 eksen ve merkezlerinde bir kutu ile aldığı verileri ekrana yansıtmaktadır.

Bu araç geliştirilirken kullanılan sahne çizgesi (Scene Graph) Şekil 6'da gösterildiği gibidir.



Şekil 6 – 3D Görselleştirme Aracı'nın sahne çizgesi

### 3.4. Doğrulama ve Geçerleme

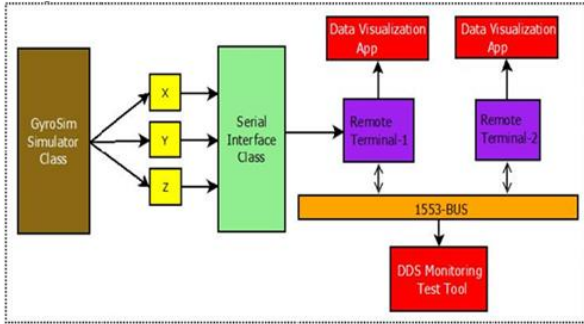
Gömülü sistemleri içeren sistemlerin yapıları gereği, gözlemleyici uygulamalar tarafından gerçek zamanda izlenmeleri veya hata ayıklama işlemleri kolay değildir. Bu nedenle tümleşik sistemlerin ve parçaların sınanması çok önemlidir.

Daha önce yapılan çalışmalarda SIL (Software-in-the-loop) ve PIL (Processor-in-the-loop) simülasyonları yapılmış olup, bu çalışmada HIL (Hardware-in-the-loop) simülasyonu denlenmektedir. Çalışma kapsamında, sistemin beklendiği gibi çalıştığının izlenebilmesi amacıyla GYROSIM, DDS İzleme Aracı (DDS Monitoring Tool) ve 3D Görselleştirme Sınama Aracı (3D Visualization Test Tool) isimli araçlar geliştirilmiştir.

3D Görselleştirme Sınama Aracı, 3D Görselleştirme Aracı'nın kalibrasyonunun yapıp doğruluğunu sınanması, ayrıca geliştirilme aşamasında kullanılan API ve diğer sınıfların etkilerinin saptanması amacıyla kullanılmaktadır. Bu amaçla önceden ardışık dönüş açısı değerleri kullanılarak elle oluşturulmuş bir metin dosyasından alınan veriler 3D Görselleştirme Aracı'na gönderilmiş ve aracın davranışları gözlemlenmiştir. Araç içerisindeki göstergenin her ekseninde yaptığı tam tur dönüş hareketi izlenerek doğrulama tamamlanmıştır.

GYROSim, gömülü sistemi taklit eden bir yazılım olup, sistemin donanım ve yazılım bölümlerinin ayrılıp, yazılım bölümünün bağımsız sınanması amacıyla kullanılmıştır. Bu sayede yazılım geliştirme aşamaları kapsamında sına ve geri besleme işlemleri, hayli vakit alan donanım geliştirme işlemleri ile eş zamanlı olarak yürütülmüştür. GYROSim, 3D Görşelleştirme Sınama Aracı'na benzer şekilde, ardışık dönüş açısı değerleriyle önceden oluşturulmuş bir metin dosyasından aldığı verileri *RTI\_SerialInterface* sınıfına göndermekte ve bu sayede veri yolu üzerindeki veri akışı gözlemlenebilmektedir.

DDS İzleme Aracı, simülasyon sisteminden bağımsız olarak çalışmakta olup, direkt olarak DDS sisteminde bağılı olarak çalışmaktadır. Bu araç DDS üzerinden geçen her türlü veriyi gözlemlenmek amacıyla geliştirilmiş, bu sayede MIL-STD-1553B standardına uygun sözcükler DDS üzerinde buldukları gibi ham haliyle gözlemlenebilmektedir. GYROSim tarafından oluşturulmuş verilerin de yardımı ile tüm veri akışı incelenmiş, zamanlama ve performans ölçümleri yapılmıştır. Sınama altyapısı Şekil 7'de gösterildiği gibidir.



Şekil 7 – Sınama Altyapısı

#### 4. Performans

MIL-STD-1553B veri yolu, gerçek zamanlı bir veri yolu olup, veri sözcüklerinin zamanlaması konusunda çok hassastır. Bu kapsamda, gerçekleştirilen simülasyonun zamanlama performansını ölçmek ve bazı ölçüm değerleri toplamak amacıyla sınama çalışmaları yürütülmüştür. Ayrıca, simülasyon, aynı konfigürasyona sahip bilgisayarlarda, gerçek zamanlı ve normal çekirdeklere sahip iki işletim sisteminde çalıştırılarak performans farklılıkları incelenmiştir.

##### Bilgisayar 1

İşlemci	Intel i5 M480 2.67GHz
RAM	8GB DDR3
İşletim Sistemi	Ubuntu 12.04 LTS 32 Bit 3.2.0-23-realtime kernel

##### Bilgisayar 2

İşlemci	Intel i5 M480 2.67GHz
RAM	8GB DDR3
İşletim Sistemi	Ubuntu 12.04 LTS 32 Bit 3.8.0-29-generic kernel

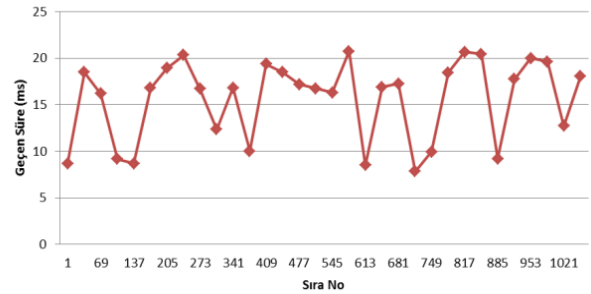
Sınama işlemleri kapsamında, sensörlerin yel aldığı gömülü sistem, oluşturduğu sensör verilerini seri port üzerinden simülasyon sistemine aktarmaktadır. Bu sırada simülasyon sistemine bağılı RT'ler, kabul ettikleri ve gönderdikleri her mesajın zamanlama bilgilerini de dahil ederek günlüklerini tutmaktadır. Aynı zamanda DDS İzleme Aracı, altyapıda kullanılan DDS yapısı üzerindeki tüm mesaj akışının dışsal olarak günlüğünü tutmaktadır. Her sınama, yaklaşık 1000 adet veri sözcüğünün veri yolu üzerinde gözlemlenmesi ile tamamlanmış, veri sözcüklerinin gönderilme ve alınma zamanları arasındaki farklar kaydedilmiştir.

Farklı zamanlarda yapılan sınamalar kapsamında, mesajların veri yolu üzerindeki iletim sürelerinin tutarlı olmadığı gözlemlenmiştir. Bu durumun simülasyonun çalıştığı işletim sistemlerinde arka planda çalışan uygulamalar olduğu değerlendirilmektedir.

Gerçek zamanlı çekirdeğe sahip işletim sisteminde çalıştırılan sınamada 1056 veri sözcüğü zamanlama bilgileri ile kaydedilmiş olup, zamanlama bilgileri aşağıda özetlenmiştir:

Veri Sayısı	Verinin Ortalama İletilme Süresi (ms)	Varyans(ms)	standart Sapma (ms)
1056	15,49966939	34,64836108	5,886285848

Zamanlama verilerinin değişim grafiği aşağıda sunulmuş olup, "Sıra No" mesajın sayısını, "Geçen Süre" ise verinin veri yolu üzerinde geçirdiği süreyi milisaniye cinsinden ifade etmektedir.

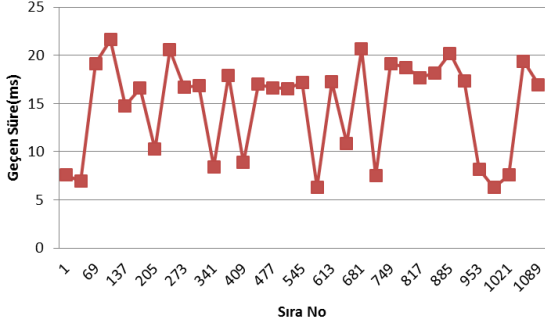


Şekil 8 – Zamanlama Verileri

Gerçek zamanlı olmayan çekirdeğe sahip işletim sisteminde çalıştırılan sınamada 1092 veri sözcüğü zamanlama bilgileri ile kaydedilmiş olup, zamanlama bilgileri aşağıda özetlenmiştir:

Veri Sayısı	Verinin Ortalama İletilme Süresi (ms)	Varyans(ms)	andart Sapma (ms)
1092	15,12949342	24,14216683	4,913467902

Zamanlama verilerinin değişim grafiği aşağıda sunulmuş olup, "Sıra No" mesajın sayısını, "Geçen Süre" ise verinin veri yolu üzerinde geçirdiği süreyi milisaniye cinsinden ifade etmektedir.



Şekil 9 – Zamanlama Verileri

Grafiklerde görüldüğü üzere, gerçekleştirilen iki sınamada da zamanlamaların varyansının oldukça yüksek olduğu gözlemlenmiştir. Gerçek bir MIL-STD-1553B veri yolu üzerinde gerçekleştirilen bir sınamada 32 bit veri sözcüğünün veri yolu üzerindeki transfer süresi 0,68 ms. olarak gözlemlenmiş, bu çalışmada ise bu süre ortalama olarak 15~ms. olmuştur. Bu veriler kapsamında, DDS üzerinde çalışan MIL-STD-1553B veri yolu simülasyonunun gerçek zamanlı olarak kullanılamayacağı değerlendirilmektedir.

## 5. Gelecek Çalışmalar

Bu çalışmanın asıl amacı, daha önce gerçekleştirilmiş, DDS üzerinde çalışan, MIL-STD-1553B simülasyon sistemi aracılığıyla gömülü bir sistemden alınan sensör verilerinin aktarımı ve böylece simülasyonun performansının değerlendirilmesidir. Bu kapsamda kullanılan MIL-STD-1553B arayüzünde karşılaşılan problemler ve eksikliklerin giderilmesi gelecek çalışmalar olarak ele alınabilir. Bu geliştirmelerin ardından aynı çalışmanın tekrar yapılması uygun görülmektedir.

## 6. Sonuçlar

Bu çalışmada, Gömülü Sistem kullanılarak DDS Tabanlı MIL-STD-1553B Veri Yolu Simülasyonu üzerinde sensör verisi aktarma denemeleri gerçekleştirilmiştir.

Bu kapsamda, yazılım ve donanım birimleri hazırlanmış, tümeleştirme ve sınamaya çalışmaları yapılmıştır.

Daha önce geliştirilmiş olan MIL-STD-1553B veri yolu simülasyonu, MIL-STD-1553B arayüz donanımlarıyla yapılan sistem entegrasyon testlerinin maliyetlerini düşürmek amacıyla hazırlanmıştır. Simülasyonun gerçek zamanlı çalışmaması nedeniyle bu hedefin yalnızca mantıksal boyutta yakalanabileceği düşünülmektedir.

Sonuç olarak, MIL-STD-1553B veri yolu simülasyonu, farklı giriş çıkış noktalarından bağlanan gömülü sistemlerin veya diğer veri yollarının gerçek zamanlı olmayan sınamalarında rahatlıkla kullanılabilecek yapıda olup, yapısının esnekliği sayesinde MIL-STD-1553B standardı için gerçekleştirilen yazılım ve donanım sınamalarının hazırlık aşamalarında kullanılabileceği değerlendirilmiştir.

## 7. Kaynaklar

- [1] Object Management Group (OMG), “Data Distribution Services for Real Time Systems”, Version 1.2, 2007.
- [2] Durak U., Deniz E., Güçlü K., Çam S. ve Oğuztüzün H., “DDS Based MIL -STD- 1553B Serial Data Bus Virtual Integration Testbed”, Journal of Defense Modeling and Simulation, basım sürecinde
- [3] Condor Engineering Inc., MIL-STD-1553B, Tutorial, 2000
- [4] Downing, N., “Virtual MIL-STD-1553B”, 25th Digital Avionics Systems Conference, 2006
- [5] Ertan, D., “DDS Based MIL-STD-1553B Data Bus Interface Simulation,” yüksek lisans tezi, ODTÜ Bilgisayar Mühendisliği Bölümü , 2012
- [6] Aziz S. M., “A cycle-accurate transaction level system model for a serial communication”, Journal of Computers and Electrical Engineering Vol. 35, September 2009.
- [7] Arduino Uno, hardware: <http://arduino.cc/en/Main/arduinoBoardUno>
- [8] Arduino API Reference, Wire Library: <http://www.arduino.cc/en/Reference/Wire>
- [9] Krodal’ın kodu: <http://playground.arduino.cc/Main/MPU-6050>
- [10] Tümler Filtre ve Kalman Filtresi: <http://robottini.altevista.org/kalman-filter-vs-complementary-filter>
- [11] MIL-STD-1553B Designers Guide, 6th Edition, 2003.