TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOĞRUSAL KONTROLCÜ DONANIMLARINDA SİNUS SİNYALİYLE AKTİF HATA TESPİTİ

YÜKSEK LİSANS TEZİ

Mutluhan Özkan

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Çoşku Kasnakoğlu

EYLÜL 2020

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

.

Mutluhan Özkan

ÖZET

Yüksek Lisans Tezi

DOĞRUSAL KONTROLCÜ DONANIMLARINDA SİNÜS SİNYALİYLE

AKTİF HATA TESPİTİ

Mutluhan Özkan

TOBB Ekonomi ve Teknoloji Üniveritesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof.Dr.Çoşku Kasnakoğlu

Tarih: Eylül 2020

Doğrusal kontrolcüler dinamik sistemlerin kontrolünde yaygın olarak kullanılmaktadır. Kontrolcü donanımında oluşabilecek bir hata dinamik sistemin yanlış sürülmesine neden olabilir. Bu durum iş güvenliği riski oluşturabilir. Kontrolcü donanımlarında aşırı ısınma, güç kaynağı bozuklukları, radyasyon gibi etkilerden dolayı hata meydana gelebilir. Hali hazırda kontrolcü donanımlarında hata tespit metotları kullanılmaktadır. Bu metotlar pasif hata tespiti metodu olan karşılaştırma yöntemine dayanmaktadır. Aktif hata tespiti ise dinamik sistemlerde hata tespiti yapmak için yaygın olarak kullanılmaktadır. Aktif hata tespitinin en önemli iki avantajı daha hızlı ve güvenilir olmasıdır. Bu çalışma kapsamında doğrusal kontrolcü donanımlarında aktif hata tespiti yapılması amaçlanmıştır. Aktif hata tespitindeki yardımcı sinyal olarak sinüs sinyali seçilmiştir. Doğrusal kontrolcüde aktif hata tespiti yapılmasının en büyük zorluğu hata tespitinin kontrolcü cevabını etkilemeyecek şekilde yapılmasıdır. Çünkü kontrolcü cevabındaki en ufak bir değişiklik dinamik sistemin yanlış sürülmesine neden olabilir. Bu çalışma kapsamında kontrolcü cevabıyla yardımcı sinyal çıkışının birbirlerinden veri kaybı olmadan ayrılması için ikizlenmiş kontrolcü tasarımı önerilmiştir. İkizlenmiş kontrolcü tasarımında kontrolcü donanımı içerisindeki doğrusal kontrolcü ikizlenir.

Doğrusal kontrolcülerden ilki pozitif sinüs sinyaliyle ikincisi negatif sinüs sinyaliyle hata tespiti yapar. İkizlenmiş kontrolcülerin çıkışları toplanarak kontrolcü cevabı elde edilir. Aynı şekilde ikizlenmiş kontrolcü çıkışları birbirlerinden çıkarılarak yardımcı çıkış sinyali elde edilir. İkizlenmiş kontrolcü tasarımı yapıldıktan sonra yardımcı sinüs sinyalinin seçilme kriterleri belirlenmiştir. Yardımcı sinüs sinyalinin frekansı doğrusal kontrolcünün frekans cevabı incelenerek seçilir. Yardımcı sinyal için belirlenen frekansta doğrusal kontrolcünün frekans cevabı büyüklüğü gözlemlenebilir olmalıdır. Yardımcı sinyalin büyüklüğü doğrusal kontrolcü donanımın dijital&analog giriş-çıkış limitlerine göre seçilmelidir. Yardımcı sinyal ve yardımcı çıkış sinyali kontrolcü donanımının giriş-çıkış limitlerine uygun olmalıdır. Yardımcı sinyalin seçilme kıstasları belirlendikten sonra aktif hata tespitinin simülasyon ortamında gerçeklenmesi yapılmıştır. Simülasyon ortamında aktif hata tespitinin gerçeklenmesi için manyetik yükseltici sistemi ve bu sistem için tasarlanmış doğrusal kontrolcü kullanılmıştır. Simülasyon sonucunda aktif hata tespitinin daha önceden tasarlanmış doğrusal kontrolcünün performansını etkilemediği gözlemlenmiştir. Simülasyon ortamında aktif hata tespiti tasarımı dış etmen hatası, örnekleme zamanı hatası ve bit değişimi hatası senaryolarında test edilmiştir ve sonuçlar kaydedilmiştir. Simülasyon testleri yapıldıktan sonra aktif hata tespiti tasarımının deneysel testi yapılmıştır. Deneysel test kapsamında aktif hata tespiti tasarımı Atmega328P kontrolcü donanımına gömülmüştür. Gömülü tasarım dış etmen , örnekleme zamanı ve bit değişimi hatası senaryolarında test edilmiştir ve sonuçlar kaydedilmiştir. Simülasyon ve deneysel testlerden sonra aktif hata tespitiyle hata maskelenmesi yapılmıştır. Hata maskelenmesi için sıcak bekleme dinamik yedekleme tasarımı önerilmiştir ve hata tespit mekanizması oluşturulmuştur. Hata tespit mekanizması yardımcı sinyalin rms değerini kontrol ederek rms değerinin belirtilen sınırların dışarı çıkması durumunda hata tespiti yapar. Hata tespiti yapıldığında hatalı kontrolcüyü basit anahtarlama mekanizmasıyla sistemden izole edilir ve sistem yedek kontrolcüyle sürülür. Hata maskelemesi tasarımı simülasyon ortamında dış etmen, örnekleme zamanı ve bit değişimi hatası senaryolarında test edilmiştir ve sonuçlar kaydedilmiştir.

Anahtar Kelimeler: Aktif hata tespiti,Kontrolcü hata Maskelenmesi, Kontrolcülerde hata tespiti, Yardımcı sinyal çıkışı ayrıştırılması, Yedekli kontrolcü tasarımı, Sinüs sinyaliyle hata tespiti

ABSTRACT

Master of Science Thesis

ACTIVE FAULT DETECTION in LINEAR CONTROLLER HARDWARE with SINE SIGNAL

Mutluhan Özkan

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Electrical and Electronics Science Programme

Supervisor: Prof.Dr.Çoşku Kasnakoğlu

Date: September 2020

Linear controller is used commonly to control dynamic system. In case of any failure in controller hardware, dynamic system could be misdirect. This case would be risky for human and work safety. The failure could occur in controller hardware due to over heating, power source malfunction, radiation and other effects. There are currently fault detection methods for controller hardware. This fault detection methods are passive and based on comparison approach. Active fault detection methods are used in dynamic system already. Active fault detection has two main advantage such as reliability and high speed. In this thesis, active fault detection is used in linear controller hardware and auxiliary signal is selected as sine signal. Active fault detection in controller hardware has challenge. The challenge is that controller output which goes dynamic system must not been affected by fault detection process. Because any small change in controller output could mislead dynamic system and cause risky for work safety. This these propose duplicated controller in the same hardware design. The design aim to separate controller output and auxiliary output signal without losing any signal data. In dublicated controller design, there are two duplicated controller. First controller use positive sine signal and second controller use negative sine signal for fault detection. Controller output is

obtained by adding two duplicated controller output and auxiliary output signal is obtained by subtracting controller outputs. After constructing duplicated control design, auxiliary signal selections criteria are declared. Frequency of auxiliary signal is selected according to bode plot of linear controller. The criteria for frequency is that magnitude of linear controller response at selected frequency must be observable. The magnitude of auxiliary signal is chosen according to controller hardware restriction. Controller hardware has digital&analog input-output limit such as 5 volt. Thus magnitude of auxiliary signal and auxiliary output signal must be proper to hardware input-output limit. After declaring auxiliary signal selection criteria, active fault detection method is constructed in simulation. Active fault detection method is applied on magnetic levitation system and its predesigned linear controller in simulation. Simulation result show that active fault detection process does not affect predesigned controller performance. Also active fault detection is tested in simulation with external factor, loop cycle time and bit change faults scenarios. After saving simulation result, active fault detection methods is embed into Atmega328P microcontroller. Embedded fault detection software is also tested with external factor , loop cycle time, bit change fault scenarios. After saving experimental test results, thesis aim to make fault masking based on active fault detection. Hot standby dynamic redundancy design is proposed to fault masking process. Fault detection mechanism use rms value of auxiliary output signal. Mechanism check rms value. If rms value exceed faulty boundary values, mechanism activate fault indicator signal. After activating fault indicator, simple switch mechanism isolate faulty controller and drive dynamic system by redundant controller. Fault masking design is constructed in simulation. The fault masking design is tested with external factor fault, loop cycle time fault and bit change fault scenarios.

Keywords: Active fault detectiom, Controller fault massking, Simulation, Fault detection in controller, Seperation of axuxillary output, Redundant Controller Design, Fault detection with sine signal

TEŞEKKÜR

Yüksek lisans sürecimde bana güvenen ve her daim destek veren değerli hocam Prof.Dr Çoşku Kasnakoğlu'na , biricik aile üyelerim Meral Özkan , Ahmet Özkan ve Oğuzhan Özkan'a , TOBB ETÜ'nün değerli öğretim üyeleri Tolga Girici ve İmam Şamil Yetik'e, Mercedes-Benz Türk bünyesindeki çalışma arkadaşlarım Tolga Ocaktaner ve Mert Altun'a, değerli yöneticim Davut Belen'e teşekkür ederim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

KISALTMALAR

PID **:** Proportional Integral Derivative
RMS **:** Root Mean Square
EDDI **:** Error Dedection by Duplicated Instruction
FOH **:** First Order Hold
IDE **:** Integrated Development Environtment
SIL **:** Safety Integrity Level

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
sn	saniye
msn	milisaniye
v	volt
mv	milivolt
K_P	Proportional katsayısı
K_I	Integrator katsayısı
K_D	Derivative katsayısı
N	Derivative filter katsayısı

RESİM LİSTESİ

Sayfa

1. GİRİŞ

Doğrusal kontrolcüler doğrusal sistemlerin ve doğrusallaştırılmış sistemlerin kontrolünde yaygın olarak kullanılırlar. PID kontrolcüsü, doğrusal durum geribesleme kontrolcüsü, doğrusal gürbüz kontrolcüsü, döngü şekillendirme kontrolcüsü doğrusal kontrolcülerden ilk akla gelenlerdir. Doğrusal Kontrolcüler dijital ve analog donanımlarla gerçeklenebilir. Kontrolcü donanımlarında aşırı ısınmadan[1,2], elektromanyetik[3], radyasyon[4] ve kozmik ışık[5] etkilerinden, bileşen[6], güç kaynağı[7] ve transistor seviyesindeki[8] bozukluklardan dolayı arıza meydana gelebilir. Bu yüzden kontrolcü donanım hatasının insan hayatı için tehlikeli olduğu uçuş ve otomotiv elektroniği, uydu sistemleri, nükleer santral gibi sistemlerde yedekli kontrolcü tasarımı kullanılır[7].Yedekli kontrolcü tasarımlarında hatalı kontrolcünün cevabı izole edilir ve dinamik sistem düzgün çalışan kontrolcüyle sürülür. Yedekli kontrol tasarımındaki en zor görevlerden birisi hatalı kontrolcünün tespit edilmesidir. Kontrolcüdeki hatayı tespit etmek için yazılım ve donanımsal hata tespit yöntemleri uygulanmaktadır. Hata tespit yönteminin hızlı ve güvenilir olması dinamik sistemler ve dinamik sistemlerle etkileşimde olan insanların güvenliği için kritiktir.

Bu çalışmada sinüs sinyali kullanılarak doğrusal kontrolcüde aktif hata tespiti yapılmıştır. Bu metotla şu anda kullanımda olan hata tespit uygularından daha güvenli ve daha az maliyetli bir hata tespit uygulamasına ulaşılması amaçlanmıştır. Ayrıca çalışmada önerilen hata tespit yöntemin gerçek bir doğrusal kontrolcüde gerçeklenmesi ve performansının analiz edilmesi de hedeflenmiştir.

1.1. Hata Tespiti Çalışmaları

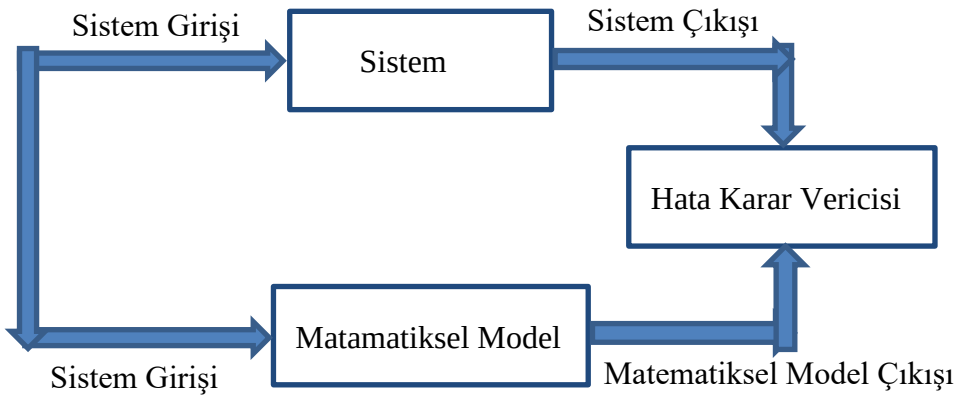
Hata tespiti için aktif hata tespiti ve pasif hata tespiti olmak üzere iki temel yaklaşım kullanılır. Pasif hata tespiti metodunda sistemin giriş ve çıkışı izlenerek sistemde hata olup olmadığı anlaşılır[9].Pasif hata tespiti model tabanlı, derin öğrenme ya da sinyal izleme metotlarıyla yapılabilir[10]. Aktif hata tespiti ise yardımcı bir test sinyalinin tasarlanıp çalışır durumdaki dinamik sisteme enjekte edilerek dinamik sistemin çıkışıyla yardımcı sinyalin ilişkisinin analiz edilerek hata tespiti yapılması olarak tanımlanır[11].

1.2 Pasif Hata Tespiti Çalışmaları

Pasif hata tespitinin temel ilkesi sistemin giriş çıkış sinyallerini inceleyerek sistem performansını etkilemeden sistem çalışırken hata tespiti yapmaktır. Pasif hata tespiti çalışmaları genel olarak model tabanlı, derin öğrenme tabanlı ya da sinyal analizi tabanlı olarak sınıflandırılabilir.

1.2.1 Model Tabanlı Pasif Hata Tespiti

Model tabanlı hata tespitinde hata tespiti yapılan sistemin matematiksel modeli çıkartılır. Sistemin girişi matematiksel modele giriş olarak verilir ve matematiksel model sistemle paralel olacak şekilde çalıştırılır.

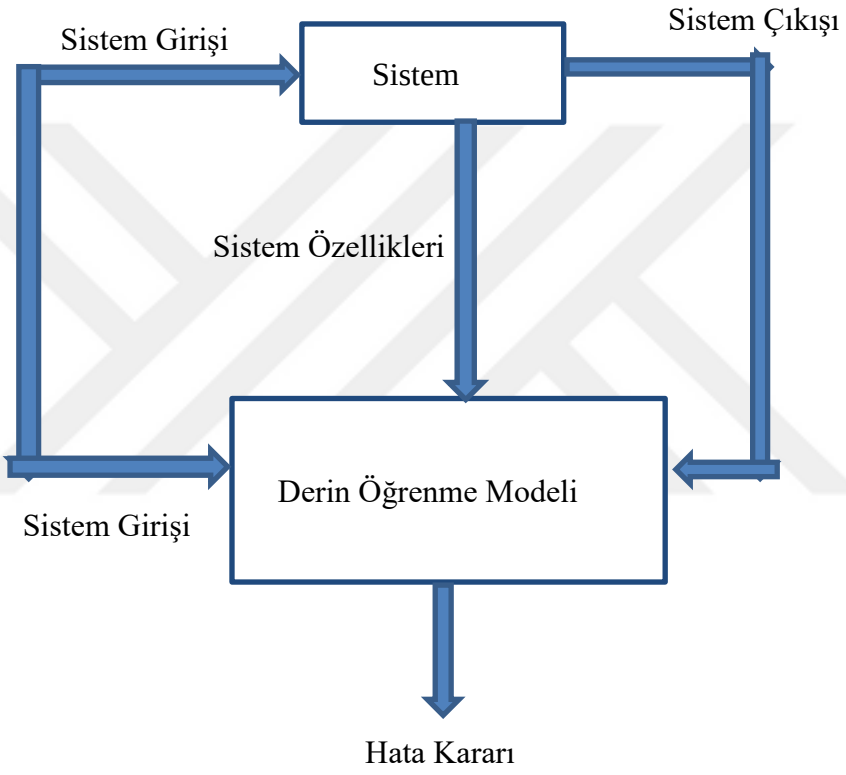


Şekil 1.1: Model tabanlı hata tespit yapısı

Model tabanlı hata tespit yapısı bir pasif hata tespit metodudur ayrıca daha yüksek bir matematiksel model ve durum tahmin edici kullanılarak sistem parametrelerindeki değişiklikler de tespit edilebilir[12]

1.2.2 Derin Öğrenme Tabanlı Hata Tespiti

Derin öğrenme tabanlı hata tespit uygulamalarında giriş sinyali, çıkış sinyali, çalışma durumu, titreşim sinyali, sıcaklık ve kullanılan akım gibi sistemin karakteristik özellikleri kaydedilerek derin öğrenme verileri oluşturulur[13]. Derin öğrenme verileri üzerinde derin öğrenme algoritmaları uygulanarak derin öğrenme sistemi belirlenir. Derin öğrenme sistemi dinamik sistemin karakteristik özelliklerini değerlendirerek hata kararını verir.

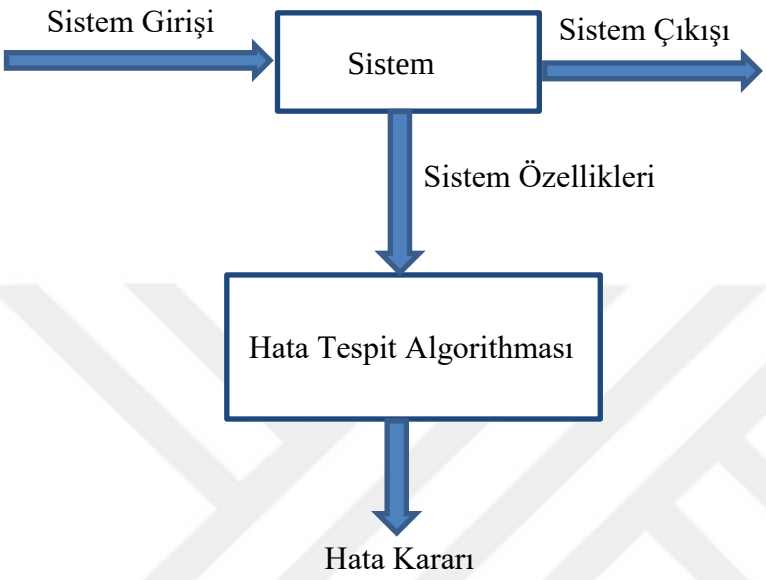


Şekil 1.2 :Derin öğrenme tabanlı hata tespit methodu gösterimi

Derin öğrenme tabanlı hata tespiti bir pasif hata tespiti metodudur. Özellikle matematiksel modellemenin çok zor olduğu ya da mümkün olmadığı sistemlerde hata tespiti yapmak için kullanılabilir.

1.2.3 Sinyal İzleme Methoduyla Hata Tespiti

Hata tespiti sistemin giriş ve çıkış sinyallerini incelerek yapılabileceği gibi sistemin sıcaklık titreşim, devre elemanları akımı gibi karakteristik özelliklerini izleyerek de yapılabilir[14]



Şekil 1.3: Sinyal İzleme Methoduyla Hata Tespiti Gösterimi

Sinyal izleme metoduyla hata tespiti bir pasif hata tespit metodudur. Diğer pasif hata tespit metotlarından daha basit bir yapıya sahiptir. Örnek olarak referans[14]'de motor sisteminde stator akımı incelenerek dişli hatası tespiti yapılmıştır.

1.3 Mikrokontrolcü Donanımlarındaki Hata Tespiti Çalışmaları

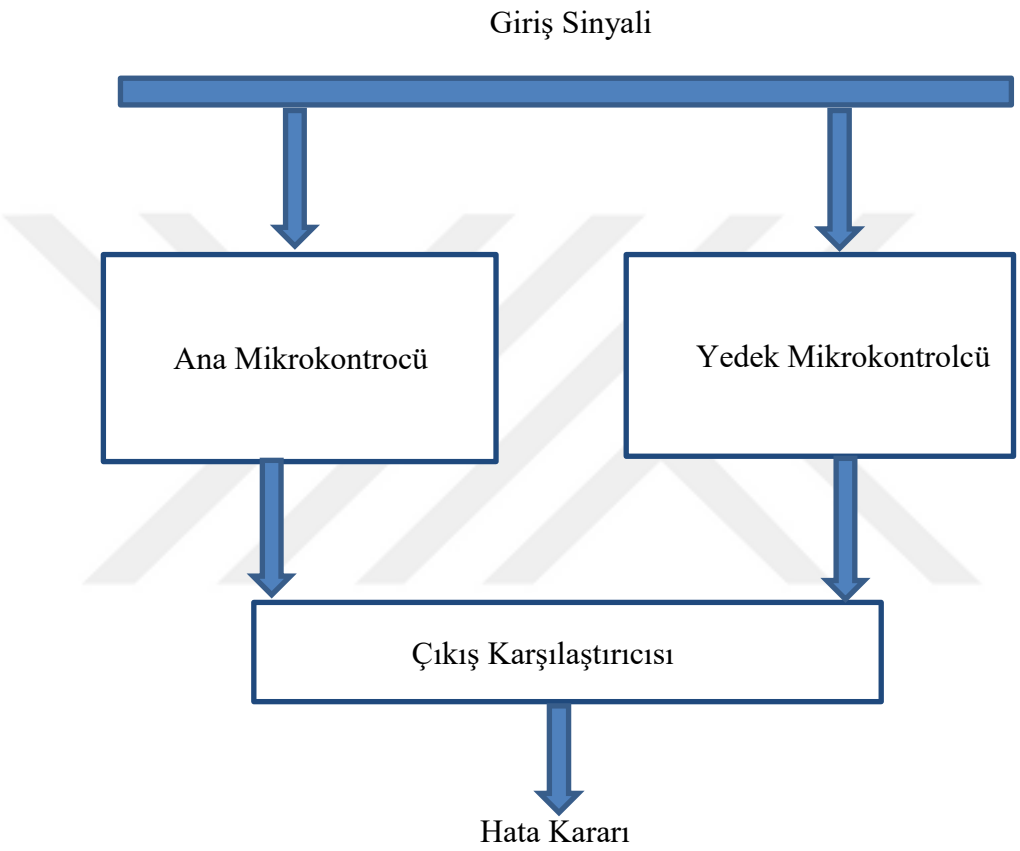
Mikrokontrolcü donanımlarının düzgün çalışması insan ve dinamik sistem güvenliği için hayatı öneme sahiptir. Örneğin, ilk Apollo uzay programlarındaki mikrokontrolcülerdeki transistor kapıları 4 tane transistörün paralel olarak bağlanmasıyla oluşturulmuştur. 4 tane transistorün paralel olarak bağlanması herhangi bir transistordeki oluşan hata durumunun paralel bağlanan transistorlerle maskelenmesini sağlamıştır.[7]

Pasif hata tespiti çalışmaları ayrıntılı olarak incelendikten sonra mikrokontrontrolcü donanımlarındaki hata tespit çalışmaları mercek altına alınmıştır. Öncelikle hata tespiti mikrokontrolcülerin kendileri tarafından yazılımsal ya da donanımsal olarak yapılabilir. Watchdog zamanlayıcısı metodu mikrodenetçilerdeki hata tespit yöntemlerinin en bilinenlerindendir. Watchdog zamanlayıcısı metodunda mikrodenetçiye ayrık olarak Watchdog zamanlayıcısı bağlanır ve mikrodenetçinin zamanlayıcıyı periyodik olarak sıfırlanması beklenmektedir[15].Zamanlayıcının sıfırlanmasında gecikme ya da hızlanma olması durumunda mikrodenetçide hata olduğu çıkarımı yapılır. Watchdog zamanlayıcısı özellikle mikrodenetçilerdeki clock hatalarını tespit etmek için kullanılır. Benzer olarak EDDI yöntemi mikrodenetçideki transistör seviyesindeki hataları tespit etmek için kullanılır. EDDI yönteminde mikrokontrolcünün içindeki yazılım özdeş iki yazılım olarak ikizlenir. İki özdeş yazılımın farklı registerler kullanması garanti edilir. Özdeş iki yazılımın sonuçları karşılaştırılır, iki yazılım sonucu arasında farklılık olması durumunda mikrodenetçide transistor seviyesinde hata olduğu çıkarımı yapılır[16].

Mikrokontrolcülerde daha gelişmiş hata tespiti çalışmaları için yedek mikrokontrolcü içeren pasif hata tespit metotları kullanılır. Bu metotlardan en çok bilinenleri karşılaştırma metodu üçlü modüler yedekli kontrolcü metodudur.

1.3.1 Mikrokontrolcülerde Karşılaştırma Methoduyla Hata Tespiti

Karşılaştırma metodunda yedek mikrokontroncü ana mikrokontrüye paralel bağlanır ve ana mikrokontrolcünün çıkış sinyalleri ile yedek kontrolcünün çıkış sinyalleriyle karşılaştırılır. Herhangi bir tutarsızlık durumunda hata tespit edildiği çıkarımı yapılır ve sistem için güvenli mod prosedürleri uygulanır[7].

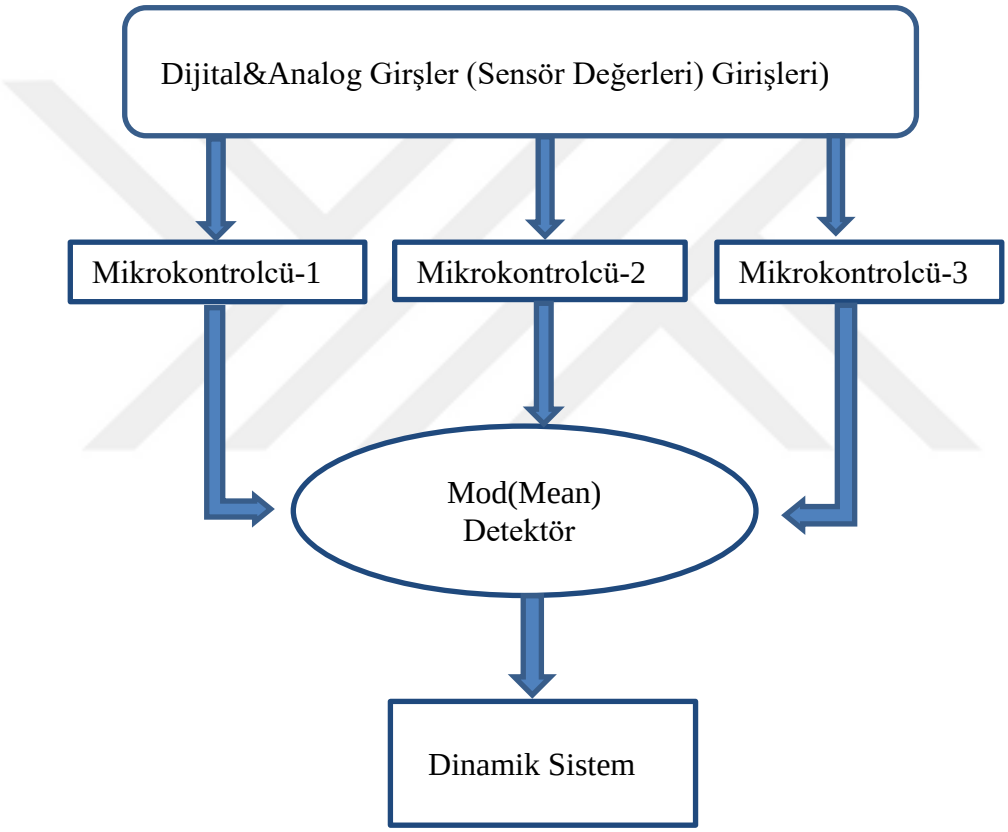


Şekil 1.2: Mikrokontrolcüde karşılaştırma yöntemiyle hata tespiti

Mikrokontrolcülerdeki karşılaştırma metoduyla hata tespiti bir pasif hata tespit etme metodudur ve yedek mikrokontrolcü kullanır bu da daha fazla maliyet ve güç tüketimi anlamına gelir. Mikrokontrolcülerdeki karşılaştırma yöntemi şekil-1.1'de açıklanan model tabanlı hata tespit etme yöntemine oldukça benzemektedir. Karşılaştırma metodunun daha gelişmiş versiyonlarında yedek kontrolcü aynı zamanda sistem verilerini kaydetmek için de kullanılır. Ana kontrolcüde oluşabilecek veri kaybında yedekli kontrolcünün verileri kullanılır[7].

1.3.2 Üçlü Modüler Yedekli Kontrolcü Sistemi

Üçlü modüler yedekli kontrolcü tasarımı en çok kullanılan donanımsal yedekli tasarımlardan birisidir[17 ,18] .Üçlü modüler yedekli kontrolcü tasarımında üç tane özdeş mikrokontrolcü ve mod(mean) detektörü bulunur. Özdeş kontrolcüler aynı sensör verilerini alırlar ve aynı kontrol algoritmalarına sahiptirler. Özdeş kontrolcüler çıkışlarını mod(mean) detektörüne iletirler. Mod(mean) detektörü 3 özdeş kontrolcünün çıkışının modunu tespit eder ve mod değerini dinamik sisteme gönderir.



Şekil 1.5: Üçlü modüler yedekli kontrolcü tasarımı

Üçlü yedekli modüler kontrolcü uzay ve askeri uygulamalarda kontrolcü sisteminin güvenliğini artırmak için tasarlanmıştır[19]. Üçlü yedekli modüler kontrolcünün matematiksel olarak güvenirlik denklemi [19]'den alınmıştır.

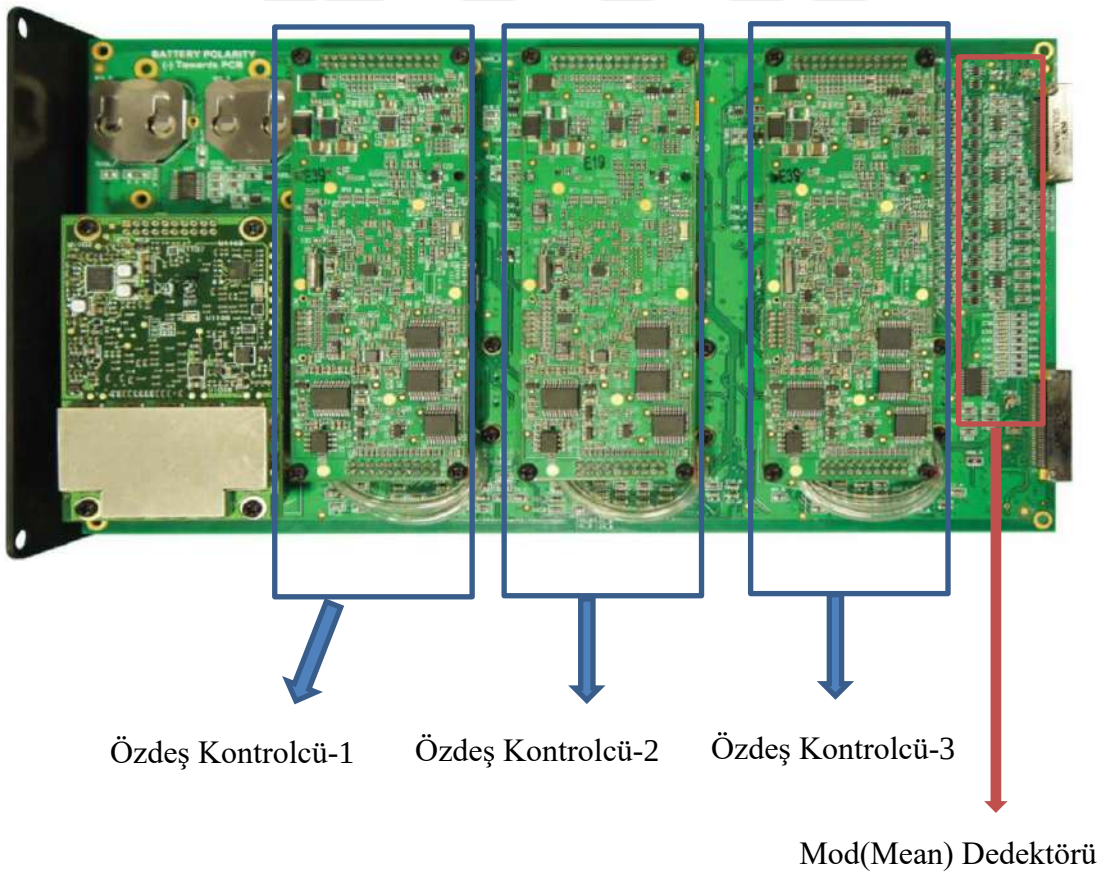
$$R_1 = bir\ kontrolcünün\ güvenirlik\ yüzdesi$$

$$R_{\text{üçlü}} = \text{Üçlü}\ modüler\ yedekli\ kontrolcü\ güvenirlik\ yüzdesi$$

$$R_{\text{üçlü}} = 3R_1^2 - 2R_1^3 \qquad (1.1)$$

Eşitlik-1'e göre %90 ile çalışan bir kontrolcünün üçlü yedekli modüler kontrolcü tasarımında kullanılmasıyla üçlü modüler yedekli kontrolcünün güvenirliği %97.2 değerine ulaşır. Üçlü modular yedekli kontrolcü sadece güvenirliği %50 üzeri olan mikrokontrolcülerin güvenirliğini artırır. Örneğin %40 güvenirlikle çalışan bir mikrokontrolcüyle üçlü modüler yedekli kontrolcü tasarımı yapılması durumunda %35.2 güvenirliğine ulaşılır.

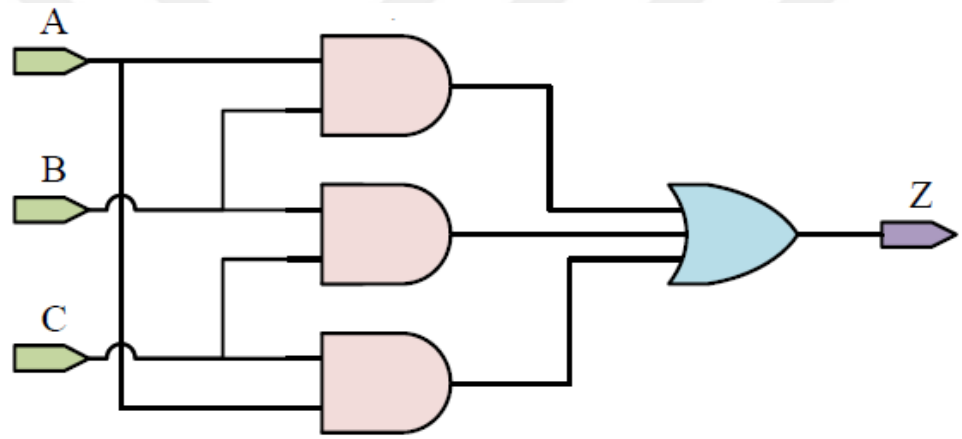
Üçlü yedekli modüler kontrolcü sistemi özellikle kontrolcü hatasının ciddi sonuçlara yol açtığı otopilot sistemlerinde kullanılmaktadır[20].



Resim 1.1: MP21283X Üçlü modüler yedekli kontrolcüsü[20]

Resim-1.1'de insansız hava araçları için dizayn edilmiş üçlü modüler yedekli kontrolcü otopilotu gösterilmiştir. Otopilot içerisinde 3 tane özdeş MP2128 kontrolcüsü ve mod mean dedektörü bulunur[20]. Üçlü modüler yedekli kontrolcü otopilotu orman ve okyanus gibi güvenli iniş imkanı olmayan insansız hava araçları görevleri için tavsiye edilmiştir. Ayrıca çok değerli elektronik yük taşıyan insansız hava araçlarında kaza risklerinin azaltılması için de üçlü modüler yedekli kontrolcü otopilotu kullanılması tavsiye edilmiştir[20].

Mod(Mean) detektörü transistor düzeyindeki lojik kapıları ile gerçeklenir, bu sayede mod(Mean) detektöründe oluşabilecek bir hata olasılığı transistörlerin paralel bağlanmasıyla en aza indirilir[7]



Şekil 1.6 :Mod(Mean) detektörünün lojik kapılarıyla gerçeklenmesi[21]

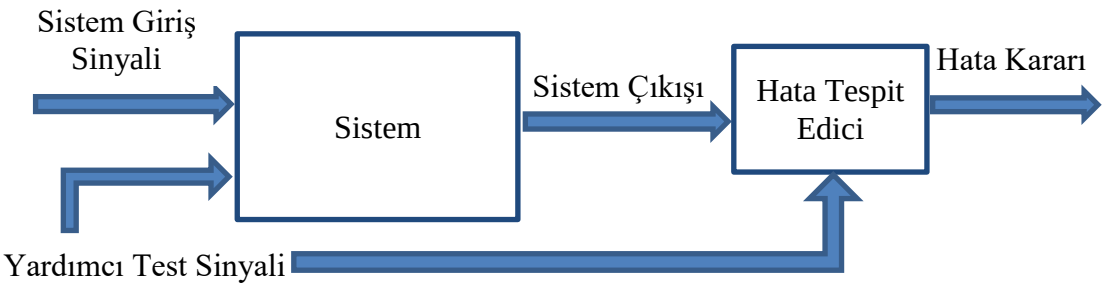
Şekil-1.6'da Mod(Mean) detektörünün lojik kapılarıyla gerçeklenmesi gösterilmiştir. A,B,C sembolleri özdeş mikrokontrolcülerin çıkış sinyalini Z sembolü ise üçlü modüler yedekli kontrolcü çıkışını temsil etmiştir. Mod(Mean) detektörü 3 tane VE kapısı ve 1 tane VEYA kapısı ile gerçeklenmiştir

Üçlü modüler yedekli kontrolcü tasarımı en çok tercih edilen yedekli kontrolcü tasarımına olmasına karşı bazı zayıflıklara sahiptir. Öncelikle üç tane mikrokontrolcü tasarım maliyetini üç katına çıkartır. Ayrıca üç mikrokontrolcü güç tüketimini artırmaktadır. Üçlü modüler yedekli kontrolcü tasarımının en önemli zayıflıklarından birisi de senkronizasyon hatasıdır. Üç özdeş kontrolcünün çıkış sinyalleri mod(mean) detektöre aynı zamanda ulaşmalıdır[7].

Senkronizasyon hatası durumunda mod(mean) detektörü hatalı değeri tespit eder ve dinamik sistem hatalı değerle sürülür.
Sonuç olarak, üçlü modüler yedekli kontrolcü bir pasif hata tespit uygulamasıdır ve temel karşılaştırma yöntemini kullanılır. Üçlü modüler yedekli kontrolcünün maliyet, güç harcaması, senkronizasyon hatası gibi zayıflıkları vardır. Bunlara ek olarak üçlü modüler yedekli kontrolcü sadece bir mikrokontrolcüdeki hata durumunu maskeleyebilir. Diğer bir değişle iki mikrokontrolcünün hata yapması durumunda mod(mean) detektörü hatalı değeri tespit edecektir ve dinamik sistemi yanlış yönlendirecektir.

1.4 Aktif Hata Tespiti Çalışmaları

Aktif hata tespiti, yardımcı test sinyalin tasarlanıp çalışır durumdaki dinamik sisteme enjekte edilerek dinamik sistemin çıkışıyla yardımcı sinyalin ilişkisinin analiz edilerek hata tespiti yapılması olarak tanımlanır[11].Aktif hata tespiti pasif hata tespitinden daha hızlı ve güvenilirdir[11].Aktif hata tespiti rüzgar tribünleri[22], kimyasal süreçler[23],insansız hava araçları[24],elektrikli araçlar[25] gibi dinamik sistemlerde yaygın olarak kullanılır.



Şekil 1.7 : Aktif hata tespiti blok diagramı

Aktif hata tespitinde dikkat edilmesi gereken en temel unsurlardan ilki yardımcı test sinyalinin sistemin çıkışını çok fazla etkilemeyecek şekilde seçilmesidir. Sistem çıkışının aktif hata tespiti yüzünden tasarlanan değerinden çok uzaklaşması sistemin performansını azaltır[11].

Aktif hata tespitinde dikkat edilmesi gereken diğer bir temel unsur da hata tespit algoritmasının seçilmesidir. Hata tespit algoritması normal sistem ile hatalı sistemin ayrımını yapmalı ve ortamdaki belirsizliklerden etkilenmemelidir[11]

2. DOĞRUSAL KONTROLCÜLERDE AKTİF HATA TESPİTİ

Aktif hata tespiti, yardımcı test sinyalin tasarlanıp çalışır durumdaki dinamik sisteme enjekte edilerek dinamik sistemin çıkışıyla yardımcı sinyalin ilişkisinin analiz edilerek hata tespiti yapılması olarak tanımlanır[11].

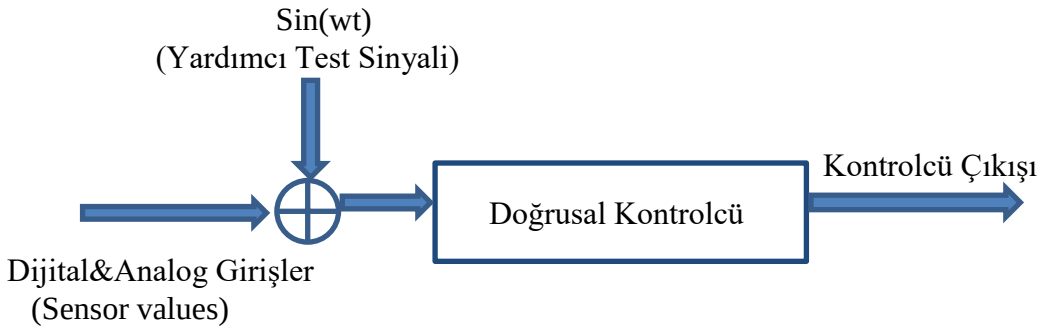
Sinüs sinyali büyüklük ve frekans özelliklerinden dolayı doğrusal sistemlerde yardımcı test sinyali olarak yaygın olarak kullanılır. Doğrusal Sistemin girişi sinüs sinyali olarak ayarlanırsa, doğrusal sistemin çıkışı da sinüs sinyali olur. Doğrusal sistemin çıkışı sistemin frekans cevabından tahmin edilir.

Şekil 2.1: Doğrusal sistemin sinüs cevabı

$A = Doğrusal\ sistemin\ w\ frekansındaki\ büyüklüğü$

$Q = Doğrusal\ sistemin\ w\ frekansındaki\ fazı$

2. 1 Doğrusal Kontrolcüde Klasik Aktif Hata Tespiti



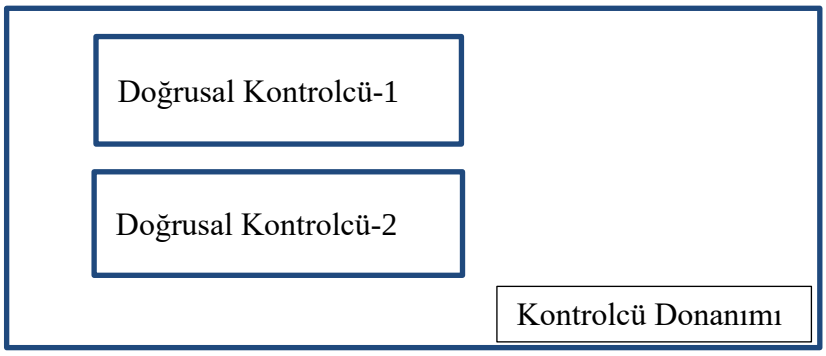
Şekil 2.2 : Doğrusal kontrolcüde aktif hata tespiti gösterimi

Doğrusal Kontrolcüdeki aktif hata tespitinin ana zorluğu kontrolcü çıkışının yardımcı test sinyalinin çıkışının ve dijital&analog çıkışının birleşiminden oluşmasıdır. Kontrolcü çıkışındaki bu iki sinyal birbirinden veri kaybı olmadan ayrılmalıdırlar. Sinyalleri birbirinden ayırmak için kullanılan klasik filtreleme yöntemleri veri kaybına yol açmaktadır.

Bu çalışmada doğrusal kontrolcüde ikizlenme yöntemi test sinyal çıkışının ve dijital&analog çıkışının bir birinden veri kaybı olmadan ayrılması için kullanılmıştır.

2.2 İkizlenmiş Doğrusal Kontrolcü Tasarımı

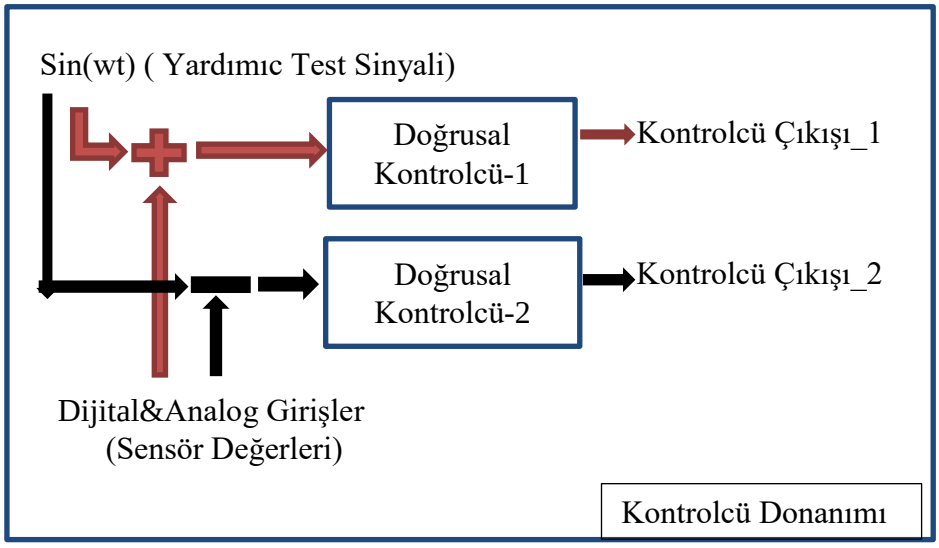
.



Şekil 2.3: İkizlenmiş doğrusal kontrolcü tasarımı

Doğrusal kontrolcü-1 ve doğrusal kontrolcü-2 özdeştir, diğer bir deyişle iki kontrolcü aynı algoritmaya sahiptirler ama farklı registerları(devre elamanları) kullanırlar.

Yardımcı test sinyalinin çıkışıyla kontrolcü cevabının birbirinden veri kaybı olmadan ayırmak için süperpozisyon özelliğinden faydalanılmıştır. Doğrusal kontrolcü-1'de pozitif sinüs sinyali kullanılarak ve doğrusal kontrolcü-2'de negatif sinüs sinyali kullanılarak aktif hata tespiti yapılır.



Şekil 2.4: İkizlenmiş doğrusal kontrolcü tasarımında sinüs sinyal kullanımı

$$Kontrolcü\ Çıkışı_1 = Kontrolcü\ Cevabı + Yardımcı\ Sinyal\ Çıkışı \quad (2.1)$$

$$Kontrolcü\ Çıkışı_2 = Kontrolcü\ Cevabı - Yardımcı\ Sinyal\ Çıkışı \quad (2.2)$$

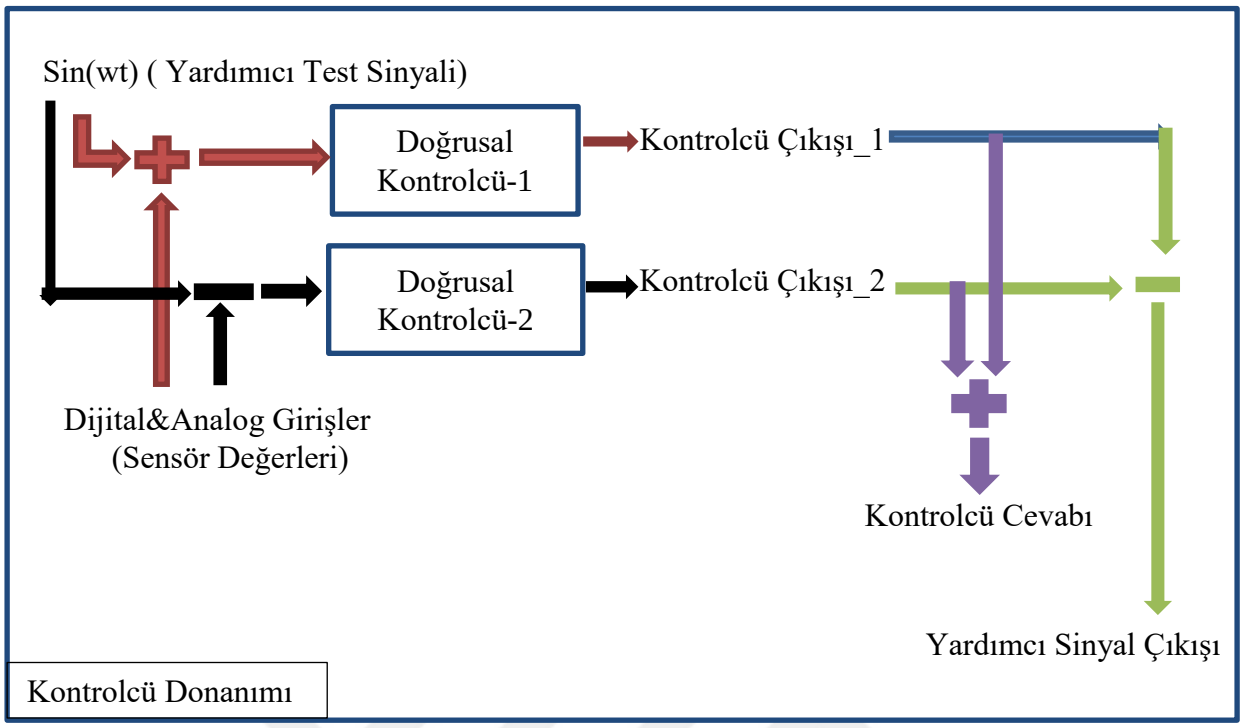
Yardımcı sinyal çıkışıyla kontrolcü cevabını birbirlerinden veri kaybı olmadan ayırmak için $Kontrolcü\ Çıkışı_1$ ve $Kontrolcü\ Çıkışı_2$ üzerlerinde işlemler yapılabilir.

$$\frac{Kontrolcü\ Çıkışı_1 + Kontrolcü\ Çıkışı_2}{2} = Kontrolcü\ Cevabı \quad (2.3)$$

$$\frac{Kontrolcü\ Çıkışı_1 - Kontrolcü\ Çıkışı_2}{2} = Yardımcı\ Sinyal\ Çıkışı \quad (2.4)$$

$Kontrolcü\ Çıkışı_1$ ve $Kontrolcü\ Çıkışı_2$ üzerinde basit toplama ve çıkarma manipülasyonuyla kontrolcü cevabı ve yardımcı sinyal çıkışı birbirinden ayrılmıştır. Bu sayede kontrolcü cevabını etkilemeyecek şekilde bir aktif hata tespitinin yapılmasının önü açılmıştır. Ayrıca yardımcı sinyal çıkışı incelenerek kontrolcü donanımında hata tespiti yapılabilir.

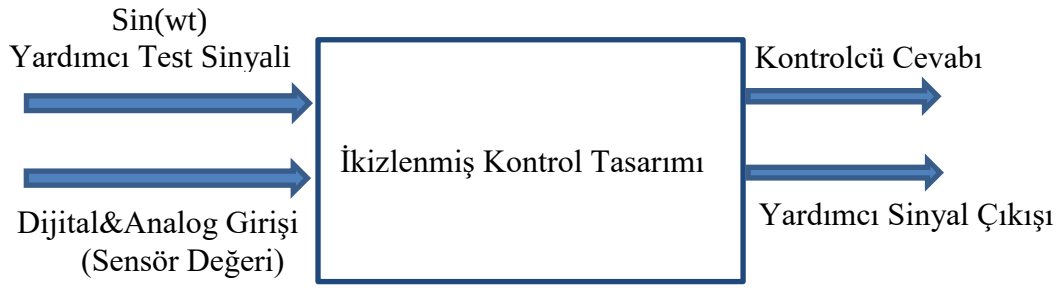
2.3 İkizlenmiş Kontrol Tasarımda Aktif Hata Tespiti Gösterimi



Şekil 2.5: İkizlenmiş kontrol tasarımında aktif hata tespiti gösterimi

Şekil-2.5'de İkizlenmiş kontrol tasarımında aktif hata tespiti dizaynı gösterilmiştir. Bu dizayn dijital ve analog doğrusal kontrolcülerde kullanılabilir ama bu çalışmada geliştirilen dizaynın dijital kontrolcülerde gerçeklenmesi üzerinde odaklanılmıştır.

Kontrolcü cevabı ve yardımcı çıkış sinyali ayrıldıktan sonra dinamik sistem kontrolcü cevabıyla sürülür. Yardımcı Sinyal çıkışı ise hata tespiti yapmak için kullanılır.



Şekil 2.6: İkizlenmiş kontrol tasarımı blok diyagramı

İkizlenmiş kontrol tasarımını daha basit ifade etmek ve gelecek bölümlerde daha kolay kullanmak için Şekil-2.6'da ikizlenmiş kontrol tasarımının blok diyagramı gösterilmiştir.

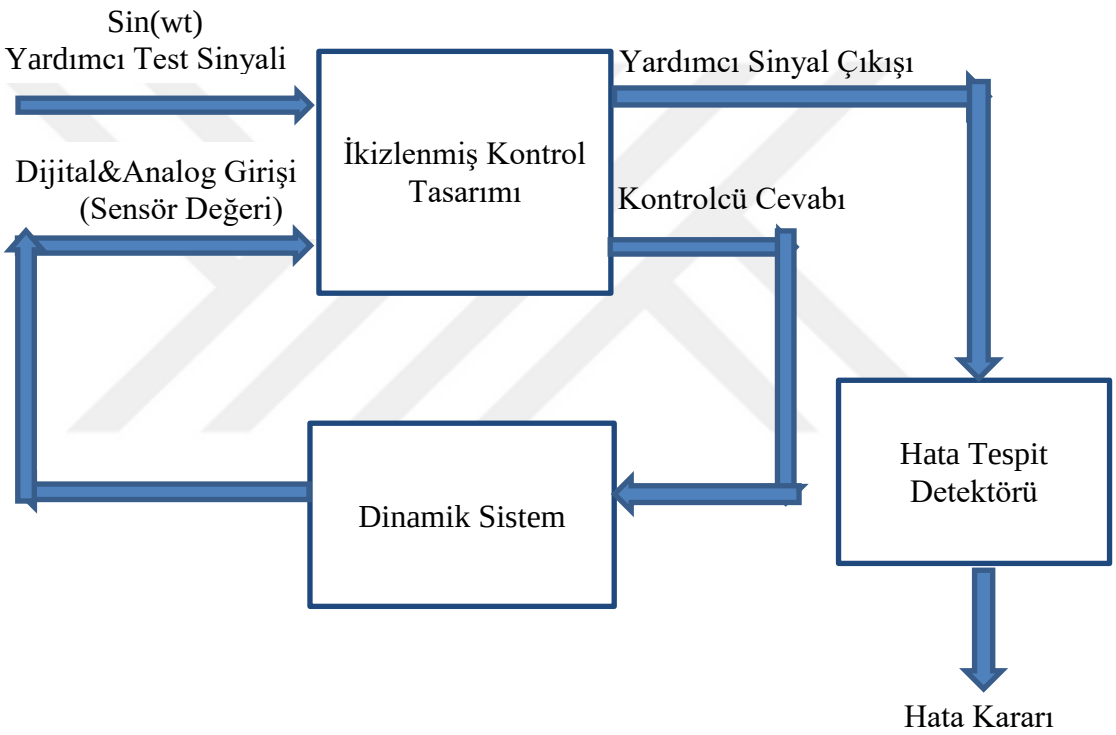
Yardımcı Sinyal Çıkışı, doğrusal kontrolcünün sinüs(yardımcı test) cevabına eşittir.

$$Yardımcı\ Sinyal\ Çıkışı = \mathrm{A}\sin(\mathrm{wt} + \mathrm{Q}) \quad (2.5)$$

$$A = Doğrusal\ Kontrolcünün\ w\ frekansındaki\ büyüklüğü$$

$$Q = Doğrusal\ Kontrolcünün\ w\ frekansındaki\ fazı$$

2. 4 Hata Tespit Düzeneği



Şekil 2.7 : Hata tespit mekanizması

İkizlenmiş Kontrol Tasarımının kontrolcü cevabı daha önceden tasarlanan doğrusal kontrolcünün kontrol cevabına eşittir. Diğer bir deyişle ikizlenmiş kontrol dizaynının daha önceden tasarlanmış kontrolcü ve dinamik sistem cevabını değiştirmeyeceğini ön görülmüştür. İlerdeki simülasyon ve deney sonuçlarda bu çıkarım test edilecektir.

2.5 Hata Tespit Methodu

Hata tespiti yardımcı sinyal çıkışı analiz edilerek yapılır. Yardımcı sinyal çıkışı büyüklük, faz, frekans olmak üzere 3 ayırt edici özelliğe sahiptir.

$$Yardımcı\ Sinyal\ Çıkışı = \mathrm{Asin}(\mathrm{wt} + \mathrm{Q}) \qquad (2.6)$$

$$A = Doğrusal\ Kontrolcünün\ w\ frekansındaki\ büyüklüğü$$

$$Q = Doğrusal\ Kontrolcünün\ w\ frekansındaki\ fazı$$

$$w = Yardımcı\ Test\ Sinyalinin\ frekansı$$

Hata tespiti A(sinyal büyüklüğü) ,Q(sinyal fazı),w(sinyal frekansı) değerlerini inceleyerek yapılabilir. Bu çalışmada hata tespiti A(sinyal büyüklüğü) üzerinde yapılmıştır.

2.6 Yardımcı Test Sinyalinin Büyüklük ve Frekansının Ayarlanması

Yardımcı test sinyali sinüs sinyali olarak ayarlanmıştır. Sinüs sinyalinin büyüklük ve frekansının seçilmesi için belirli kıstaslar oluşturulmuştur. Kıstaslar doğrusal kontrolcünün frekans cevabına göre kontrolcü donanımının dijital&analog giriş çıkışlarına göre belirlenmiştir.

2.6.1 Frekans Kıstası

Sinüs frekansı doğrusal kontrolcünün frekans cevabına göre seçilir. Seçilen frekansta doğrusal kontrolcü cevabının gözlemlenebilir olması gerekir. Sinüs frekansının kontrolcü cevabının çok büyük ya da küçük değerlerde seçilmesi yardımcı çıkış sinyalinin analiz edilmesini güçleştirir.

2.6.2 Büyüklük Kıstası

Yardımcı test sinyalinin büyüklüğü kontrolcü donanımın dijital&analog giriş-çıkış limitlerine göre yapılır. Çoğu dijital kontrolcü 5 volt dijital&analog çıkışına sahiptir, bu yüzden yardımcı test sinyali girişi ve yardımcı çıkış sinyalinin büyüklüğü 5v'dan küçük olmalıdır.

2.6.3 Hata Tespit Detektörü

Hata tespit detektörü beklenen yardımcı sinyal çıkışıyla gerçek yardımcı sinyal çıkışını karşılaştırır. Yardımcı sinyal çıkışının beklenen değerden sapması sonucunda tespit detektörü hata olduğu çıkarımı yapılır. Hata tespit detektörü analog ve basit devre elemanlarıyla gerçeklenebileceği gibi yardımcı çıkış sinyali gerçek zamanlı bir monitöre aktarılarak eş zamanlı olarak da takip edilebilir.

3. HATA TESPİT ALGORİTMASI

Hata tespit algoritmasında öncelikle Matlab ortamında örnekleme zamanı göz önünde bulundurularak doğrusal kontrolcünün frekans cevabı(bode grafiği) çizdirilir. Frekans cevabı üzerinde daha önceden belirlenen kıstaslar göz önünde bulunarak yardımcı test sinyalin büyüklüğü ve frekansı seçilir. Seçilen yardımcı sinyalin doğrusal kontrolcü üzerindeki cevabı beklenen yardımcı test çıkışı olarak ayarlanır.

Şekil 3.1: Beklenen yardımcı sinyal hesaplanması

3.1 Beklenen Yardımcı Sinyalinin Hesaplaması

Öncelikle doğrusal kontrolcünün transfer fonksiyonu S sürekli dönüşümüyle elde edilir, daha sonra dijital sistemin örnekleme zamanı belirlenir. FOH(First Order Hold) yönteminin dinamik sistemler için diğer ayrıklaştırma yöntemlerinden daha iyi olduğu literatür araştırmasında gözlenmiştir[26]. Bu yüzden sürekli doğrusal kontrolcü FOH yaklaşımıyla ayrık doğrusal kontrolcüye dönüştürülmüştür. FOH dönüşümü eşitlik-3.1'de belirtilmiştir. Ayrık kontrolcünün frekans cevabı(bode grafiği) çizdirilir. Yardımcı test sinyalin büyüklüğü ve frekansı ayrık kontrolcünün frekans cevabına göre seçilir. Ayrık kontrolcünün seçilen yardımcı test sinyaline verdiği cevap beklenen yardımcı sinyal çıkışı olarak ayarlanır.

$$h(kT+\tau) = x(kT) + \frac{x(kT) - x\big((k-1)T\big)}{T} \tag{3.1}$$

$$0 < \tau < T \tag{3.2}$$

$$T = Örnekleme\ Zamanı\ ,\ \ k = Ayrık\ Zaman\ Adımı$$

$$\tau = Ayrık\ Zaman\ Ara\ Adımı$$

3.2 Hata Tespitindeki Belirsizlikler

Hata tespiti yaparken genel olarak matematiksel modellemelerden, simülasyon yakınsamalarından, elektronik gürültülerden kaynaklı belirsizlikler olabilir. Çalışmamızda hata tespiti aşamalarındaki oluşabilecek belirsizlikler araştırılmıştır.

3.2.1 Örnekleme Zamanı Belirsizliği

Hata tespit algoritmasındaki ilk belirsizlik örnekleme zamanının belirlenmesidir. Mikrokontrolcülerde clock iletimindeki çarpıklıklardan, seğirmelerden ve güç kaynağındaki gürültülerden dolayı döngü süresinde belirsizlikler meydana gelebilir[27].Diğer bir değişle, mikrokontrolcünün döngü süresini tam olarak hesaplamak mümkün değildir. Bu nedenle belirlenen örnekleme zamanıyla mikrokontrolcüdeki gerçek örnekleme zamanı arasında ufak farklılıklar olabilir

.

3.2.2 Sürekli Sistemin Ayrık Sisteme Dönüştürülmesi Belirsizliği

Sürekli sistemde tasarlanan kontrolcünün ayrık zamana dönüşümünde kesin bir eşleşme yakalanamaz. Çünkü sürekli sistemi ayrık zamana dönüştüren metotlar yaklaşımlara dayanmaktadır. Bu çalışmada kullanılan FOH metodu da doğrusal yaklaşıma dayanmaktadır. Bu yüzden doğrusal sistemin ayrık sisteme dönüştürülmesinde elde edilen modeller gerçek sisteme çok yakın olsalar da ufak belirsizlikler içerebilirler.

3.2.3 Yardımcı Sinyaldeki Belirsizlikler

Yardımcı sinyal mikrokontrolcü dışındaki analog yada dijital bir kaynaktan analog ve dijital formlarda elde edilebilir .Ayrıca yardımcı sinyal mikrokontrolcünün kendi zamanlayıcısı kullanılarak da oluşturulabilir[28]. Yardımcı sinyalin oluşturulması sırasında clock ya da gürültü kaynaklı bozukluklar olabilir ve bu bozukluklar yardımcı sinyalin frekansında ufak belirsizliklere yol açabilir.

3.2.4 Hata Tespit Payının Belirlenmesi

Hata tespitindeki belirsizliklerden dolayı hesaplanan yardımcı sinyalle gerçekte ölçülen yardımcı sinyalin bire bir aynı olamayacağı açıktır. Bu nedenle $\pm\%10$ hata payı belirlenmiştir. Diğer bir değişle gerçek yardımcı sinyalle hesaplanan yardımcı sinyalin büyüklüğü arasındaki sapma miktarı $\pm\%10$ dan fazla olması durumunda sistemde hata olduğu çıkarımı yapılır.

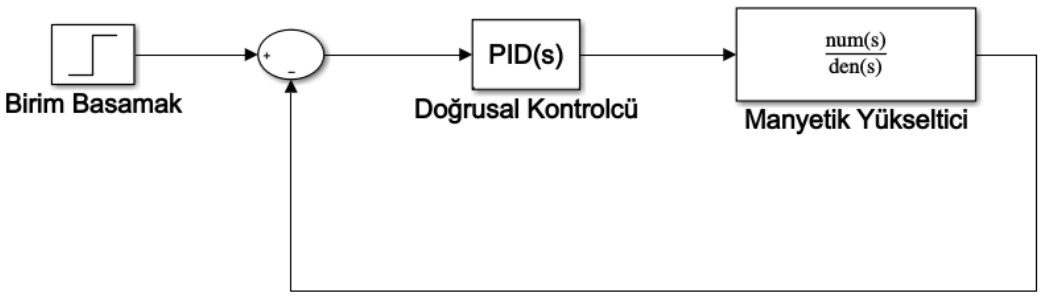
4. HATA TESPİTİNİN SİMÜLASYON ORTAMINDA GERÇEKLENMESİ

Doğrusal kontrolcüde aktif hata tespiti yapmak için ilk olarak bir dinamik sistem ve dinamik sistem için tasarlanmış kontrolcü literatür araştırmasında seçilmiştir. Manyetik yükseltici sistemi ve bu sistem için tasarlanan doğrusal PID kontrolcü [29]'den alınmıştır.

Simülasyonda önceden tasarlanmış olan doğrusal kontrolcü üzerinde hata tespiti yapılmıştır. Öncelikle hata tespit metodunun tasarlanan dinamik sistemin ve doğrusal kontrolcünün performansları üzerinde etkisi olmadığının gösterilmesi amaçlanmıştır. Ayrıca simülasyon üzerinde doğrusal kontrolcüye çeşitli hatalar eklenip hata tespit performansı ölçülmüştür.

4.1 Tasarlanan Dinamik Sistem ve Kontrolcü Performansları

Dinamik sistem ve kontrolcü değerleri [29]'den alınmıştır. Öncelikle tasarlanan modelin doğrusal kontrolcü performansı ve dinamik sistem birim cevabı analiz edilmiştir.

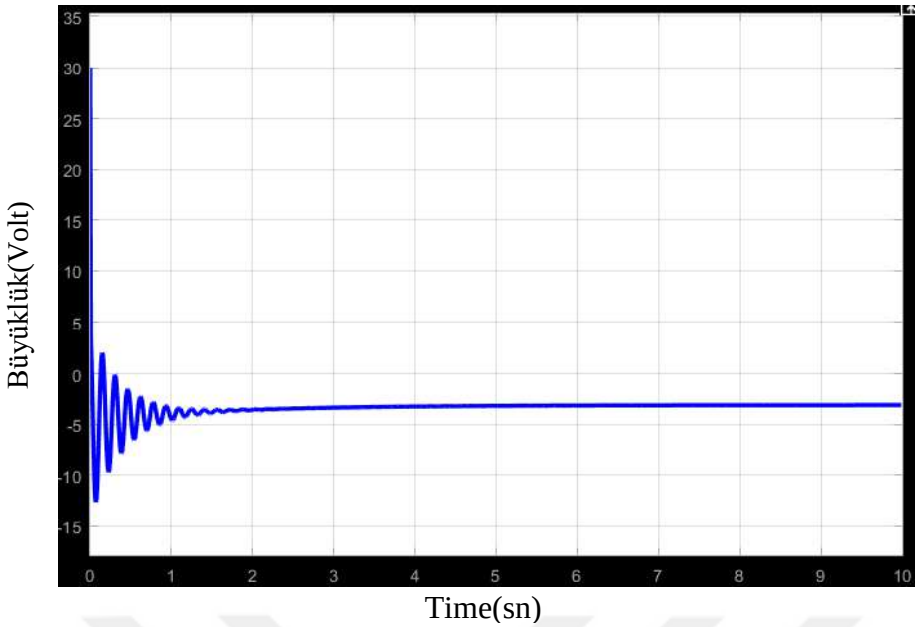


Şekil 4.1: Tasarlanan modelin kapalı döngü birim basamak gösterimi

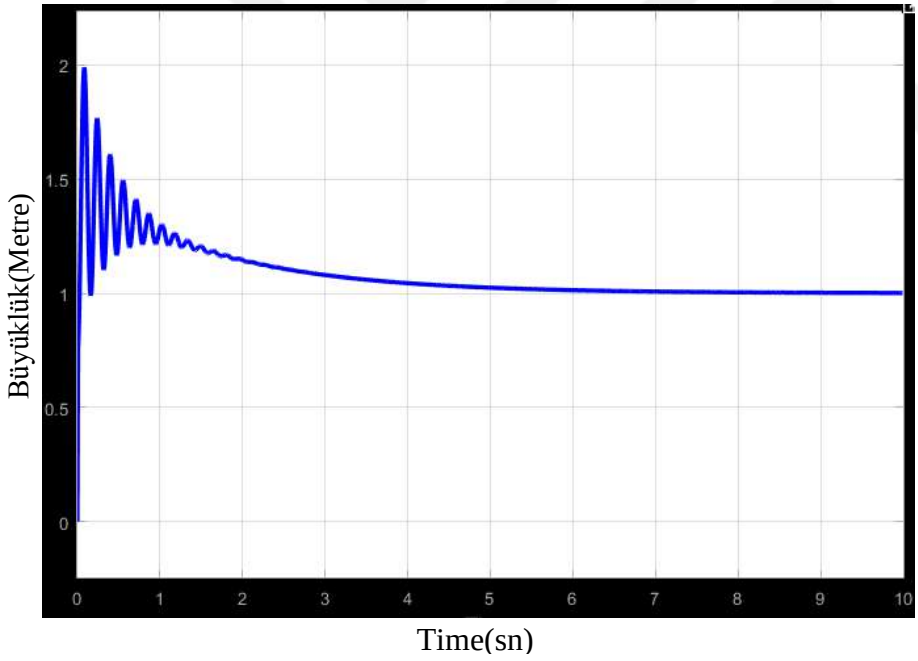
Simülasyon modelinin başka çalışmalarda da tekrarlanabilmesi için manyetik yükselticinin transfer fonksiyonu ve PID kontrolcünün parametreleri paylaşılmıştır

$$Manyetik\ Yükseltici = \frac{20.66s^2 + 61803}{s^4 + 132.5s^3 - 1471s^2 - 194900}$$

$$Douğrusal\ Kontrolcü\ Parametreleri \rightarrow K_P = 10\ K_I = 4\ \ K_D = 0.2\ \ N = 100$$



Şekil 4.2: Doğrusal kontrolcü kapalı döngü birim basamak performansı

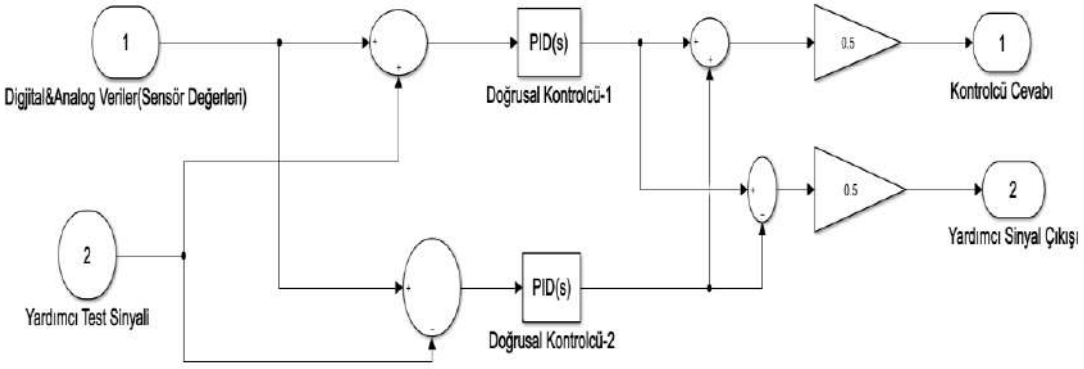


Şekil 4.3 :Dinamik sistemin kapalı döngü birim basamak cevabı

Tasarlanmış doğrusal kontrolcünün kapalı döngü birim basamak performansı ve dinamik sistemin kapalı döngü birim basamak cevabı yukarıda belirtilmiştir. Bu çalışmada hata tespit uygulamasının önceden tasarlanmış kontrolcünün performansını ve birim basamak cevabı üzerinde etkisi olmadığı gösterilecektir.

4.2 Hata Tespit Yapısının Kurulması

İlk olarak doğrusal kontrolcü ikizlenir ve Şekil-2.5'deki hata tespit yapısı kurulur.



Şekil 4.4: Simalasyon ortamında hata tespit yapısı

Kontrolcü cevabı dinamik sisteme iletilir ve yardımcı sinyal çıkışı hata tespiti yapmak için kullanılır.

4.3 Yardımcı Test Sinyalinin Seçilmesi

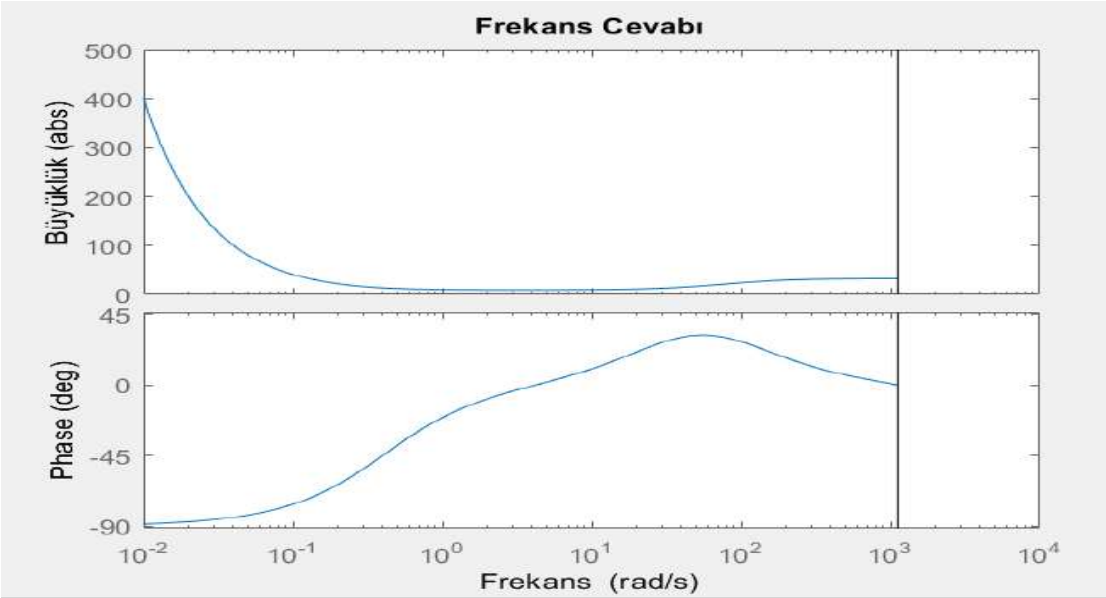
Yardımcı test sinyali doğrusal kontrolcünün frekans cevabına göre seçilir. İlk olarak doğrusal kontrolcünün sürekli transfer fonksiyonu elde edilir. Sürekli transfer fonksiyonu foh yöntemiyle ayrık sisteme dönüştürülür.

$$\frac{30s^2 + 1004s + 400}{s^2 + 100s} \rightarrow foh\ dönüşümü \rightarrow \frac{29.03z^2 - 57.11z + 28.08}{z^2 - 1905z + 0.9048} \quad (4.1)$$

Simülasyon ortamında örnekleme zamanı 0.001 sn olarak ayarlanmıştır ama deneysel ortamda microkontrolcünün örnekleme zamanı ölçülmelidir.

Doğrusal kontrolcünün ayrık sistem transfer fonksiyonu elde edildikten sonra ayrık sistemin frekans cevabı çizdirilir. Yardımcı test sinyalinin frekansı ayrık sistemin frekans(bode) cevabı üzerinde seçilir.

4.4 Ayrık Kontrolcünün Frekans Cevabı



Şekil 4.5 Ayrık kontrolcünün frekans cevabı

Ayrık kontrolcünün frekans cevabı şekil-4.5'de gösterilmiştir. Grafikte görüldüğü gibi kontrolcü düşük frekanslarda yüksek cevaplara ulaşır, bu yüzden yardımcı sinyal düşük frekanslarda seçilmemelidir. Örnekleme zamanı 0.001sn olduğu için Nyquist teoreminden dolayı yardımcı sinyalin büyük frekanslarda seçilmesi güvenli olmayabilir. Bu nedenlerden dolayı yardımcı sinyal frekansı 10 rad/sn seçilmiştir ama yardımcı sinyalin frekansı bu frekanslara uyan başka bir değer de seçilebilir.

4.5 Yardımcı Sinyal Büyüklüğünün Seçilmesi

İlk olarak seçilen frekanstaki kontrolcü kazancı bulunur, daha sonra dijital kontrolcü giriş-çıkış limitlerine göre hesaplanan yardımcı test sinyali çıkışı ve yardımcı sinyal büyüklüğü belirlenir

$$10 \text{ rad/s frekansında kontrolcü kazancı} = 10.32$$

$$Dijital\ giriş\&çıkış\ limiti = 5\ volt$$

$$Hata\ tespit\ payı = \pm\%10$$

$$Belirlenen\ yardımcı\ sinyal\ çıkışı \rightarrow 4.5\sin(10t)$$

Yardımcı sinyal çıkışının frekansı yardımcı test sinyalinin frekansına eşittir. Ama hesaplanan yardımcı sinyalin büyüklüğü dijital giriş-çıkış limitine göre ayarlanır.

$$Belirlenen\ yardımcı\ sinyal\ çıkışı\ \pm\ \%10\ hata\ tespiti < 5 \qquad (4.2)$$

Belirlenen yardımcı sinyal belirtildikten sonra yardımcı test sinyalin büyüklüğü hesaplanan yardımcı sinyalin büyüklüğünün kontrol kazancına bölünmesiyle bulunur

$$Yardımcı\ Test\ Sinyal\ Girişi = \frac{Belirtilen\ Yardımcı\ Sinyal\ Çıkşı}{Kontrolcü\ Kazancı} \qquad (4.3)$$

$$Yardımcı\ Test\ Sinyal\ Büyüklüğü = \frac{4.5}{10.32} \cong 0.44 \qquad (4.4)$$

$$Yardımcı\ Test\ Sinyali = 0.44\sin(10t)$$

$$Hesaplanan\ Test\ Sinyali = 4.5\sin(10t)$$

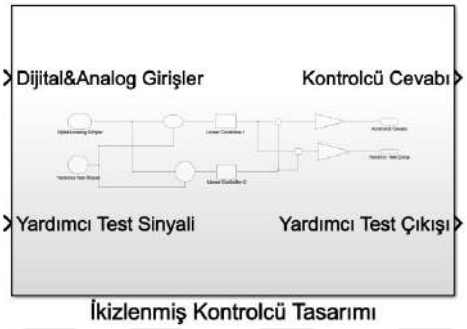
4.5.1 İdeal Doğrusal Kontrolcünün Yardımcı Sinyal Tepkisi

Şekil 4.6 : İdeal kontrolcünün yardımcı test sinyal cevabı

İdeal çalışan bir kontrolcünün 0.44sin(10t) yardımcı test sinyaline 4.5sin(10t) çıkış vermesi beklenir. Matematiksel modelleme ve elektronik belirsizliklerden dolayı hata payı $\pm\%\ 10$ olarak ayarlanmıştır. Diğer bir deyişle yardımcı sinyal çıkışı 4.05sin(10t) ile 4.95sin(10t) arasında ise kontrolcünün hatasız çalıştığı çıkarımı yapılır.

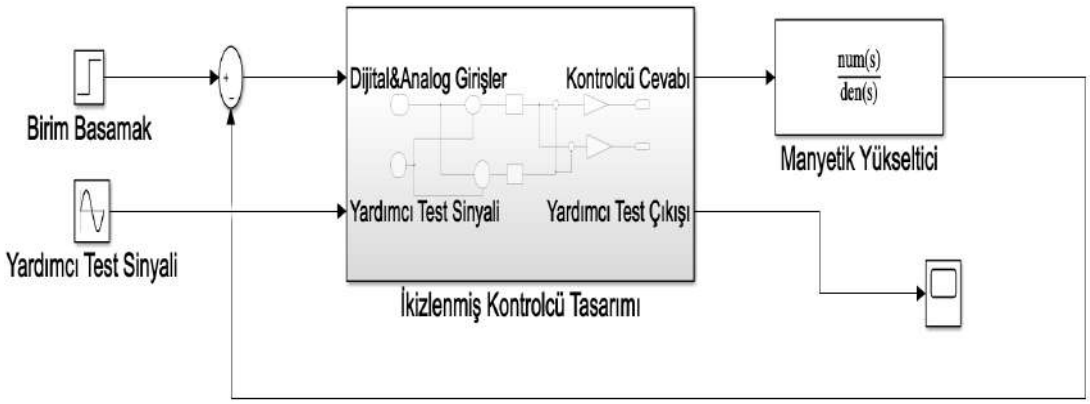
4.6 İkizlenmiş Kontrolcü Bloğu

Şekil-4.4'deki hata tespit yapısı simülasyonun daha kolay anlaşılması şekil-4.7'deki blok diyagramına indirgenmiştir.



Şekil 4.7: İkizlenmiş kontrolcü bloğu

4.7 Hata Tespit Düzeneği

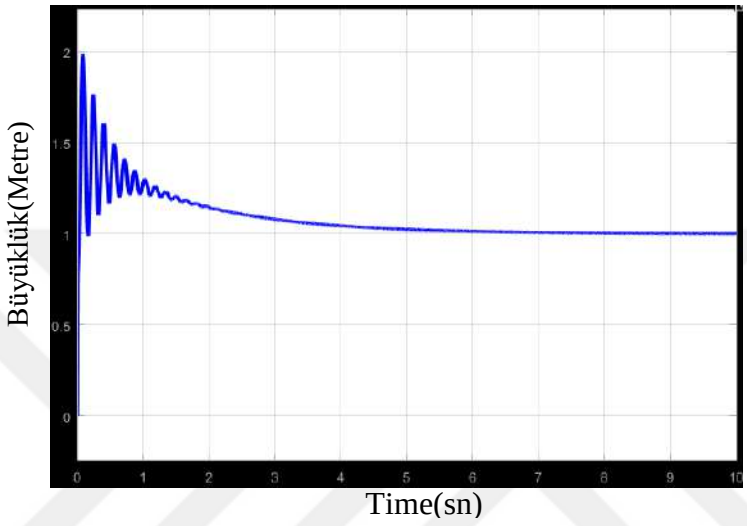


Şekil 4.8: Hata tespit düzeneği

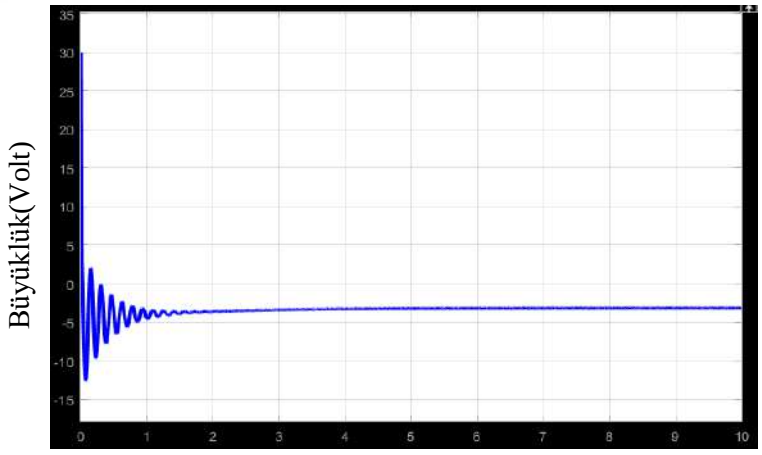
Şekil-2.7'deki hata tespit mekanizması simülasyon ortamında şekil-4.8'daki gibi gerçeklenmiştir. Dijital&Analog değerlerini temsil etmek için birim basamak fonksiyonu kullanılmıştır. Hata tespit düzeneği olarak sinyal monitörü kullanılmıştır. Monitörle yardımcı sinyal çıkışı izlenmiştir.

4.8 Aktif Hata Tespiti Simülasyon Sonuçları

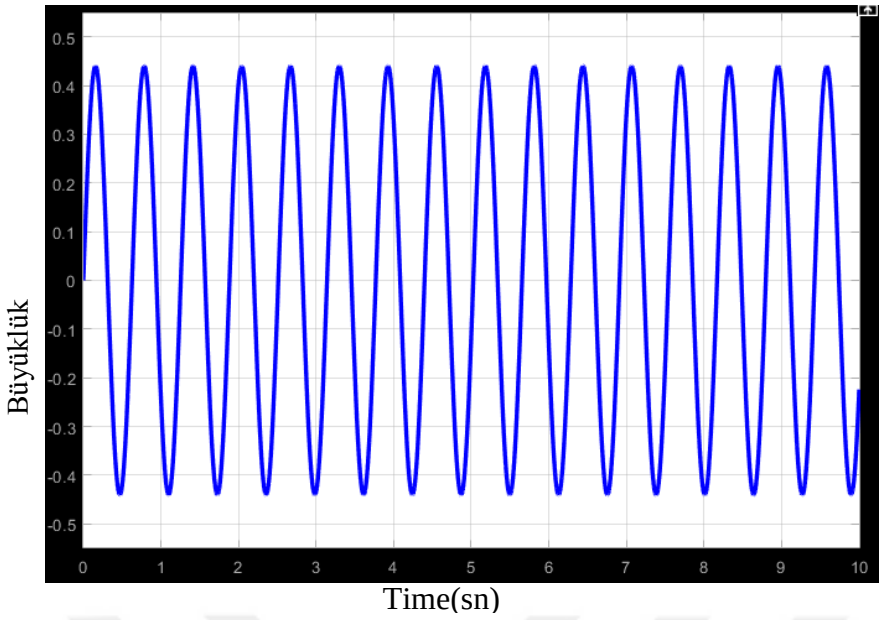
Hata tespit düzeneği simülasyon ortamında kurulduktan sonra dinamik sistemin kapalı döngü birim basamak cevabı, kontrolcünün kapalı döngü birim basamak cevabı, yardımcı test sinyal girişi ve yardımcı test sinyal çıkışları simülasyon ortamında çizdirilmiştir.



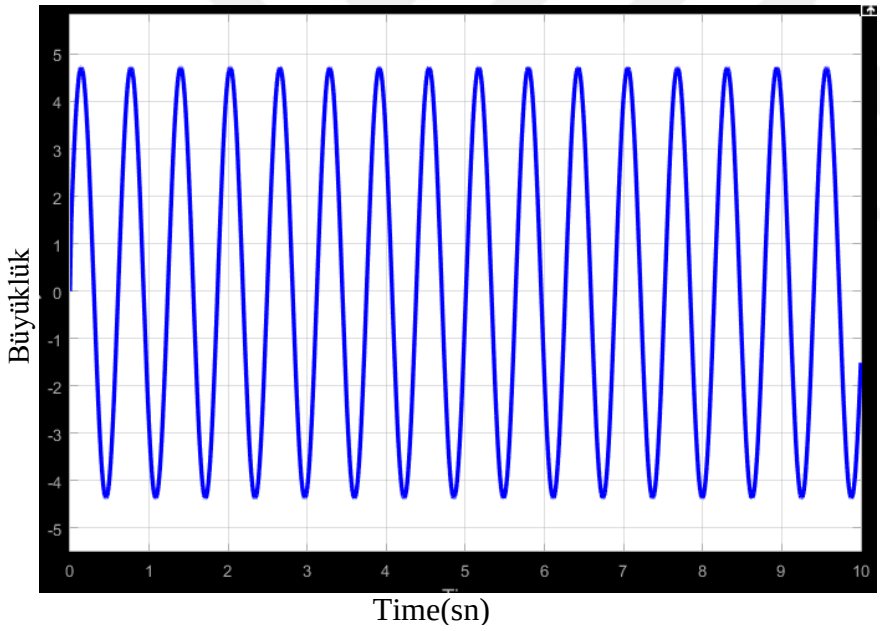
Şekil 4.9 :Aktif hata tespiti metodunun kapalı döngü birim basamak cevabı



Şekil 4.10 :Aktif hata tespit metodunun kontrolcü kapalı döngü birim basamak performansı



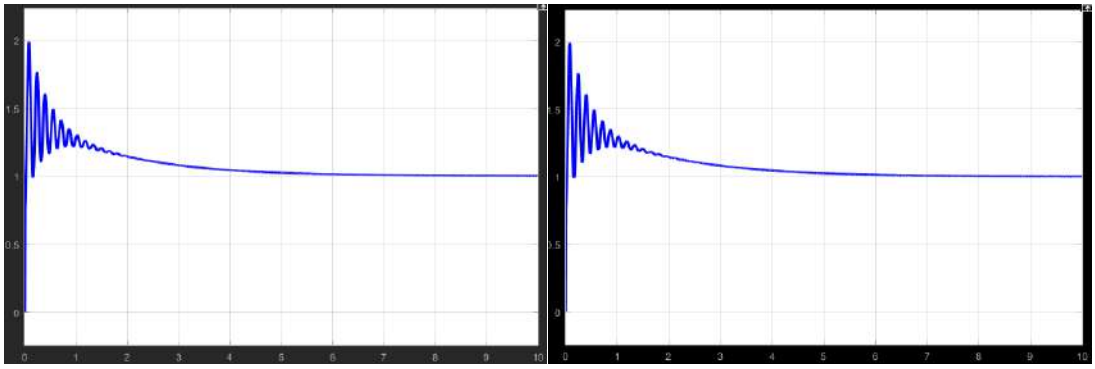
Şekil 4.11: Yardımcı test sinyali girişi



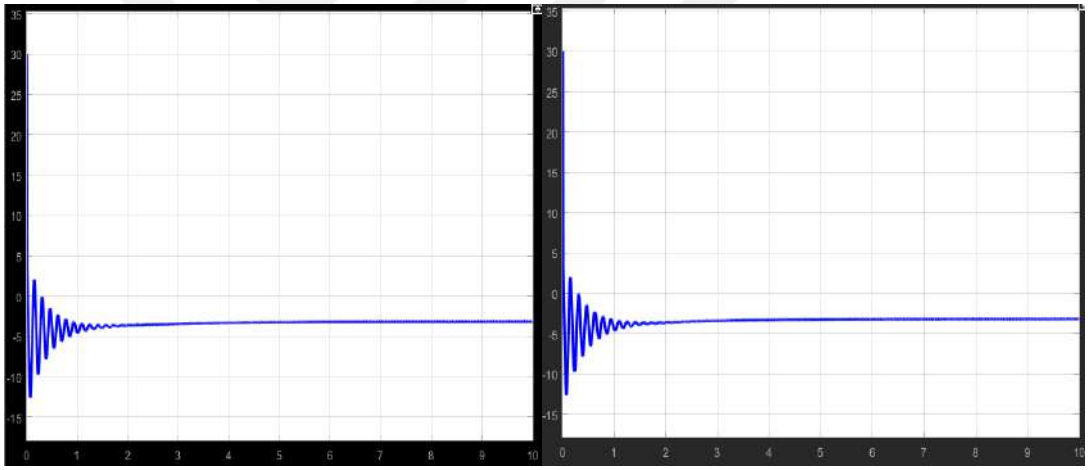
Şekil 4.12: Yardımcı test sinyali çıkışı

4.9 Simulasyon Çıkarımları

Simülasyon sonuçları incelendiğinde hata tespit metodunun dinamik sistem ve doğrusal kontrolcünün performanslarını değiştirmediği görülür.



Şekil 4.13: Sırasıyla tasarlanmış sistem ve hata tespiti yapılmış sistem performansları



Şekil 4.14: Sırasıyla tasarlanmış kontrolcü ve hata tespiti yapılmış kontrolcü performansları

Hata tespiti ise yardımcı sinyal çıkışının analiz edilerek yapılır. Şekil-4.12'de yardımcı sinyal çıkışı gösterilmiştir.

$$Simulasyondaki\ Yardımcı\ Sinyal\ Çıkışı = 4.73\sin(10t)$$

$$Hesaplanan\ Yardımcı\ Sinyal\ Çıkışı = 4.5\sin(10t)$$

$$Sapma\ Oranı = \frac{0.23}{4.5} = \%5.1 \qquad (4.5)$$

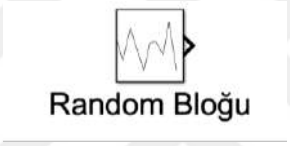
Daha önce bahsedilen belirsizliklerden dolayı simülasyonda yardımcı sinyal çıkışında %5.1 sapma payı gerçeklenmeştir. Sapma payı $\pm\%10$ hata payı sınırları içerisinde kalır, bundan dolayı doğrusal kontrolcünün doğru çalıştığı çıkarımı yapılır.

5. HATA SENARYOLARININ SİMÜLASYONDA GERÇEKLENMESİ

Simülasyon ortamındaki hata modellemeleriyle hata tespit algoritmasının performansının ölçülmesi hedeflenmiştir. Gerçek hatalar dış etmenlere, elektronik belirsizliklere bağlıdır. Hata modelleriyle gerçek hataların simülasyon ortamında benzetimleri yapılmaya çalışılmıştır.

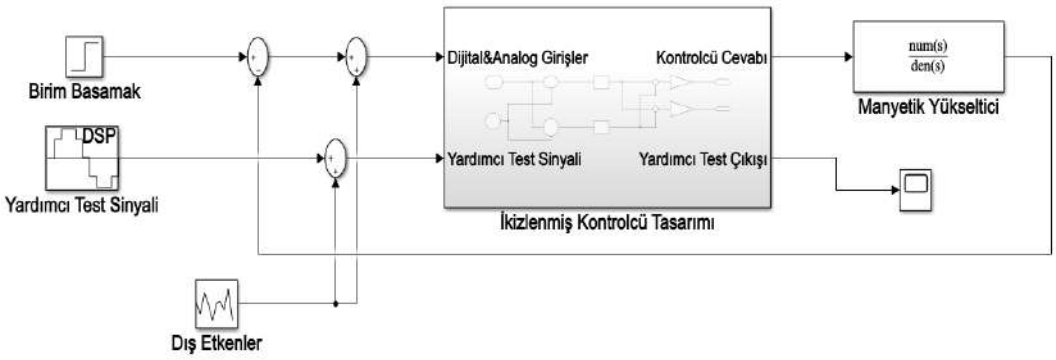
5.1 Dış Etmen Hatası Modellemesi

Kontrolcü donanımlarında elektromanyetik, radyasyon, kozmik ışık etkilerinden ve güç kaynağındaki bozukluklardan dolayı arıza meydana gelebilir. Simülasyon ortamında bu hataların modellenmesi random bloğu kullanılarak modellenmiştir.



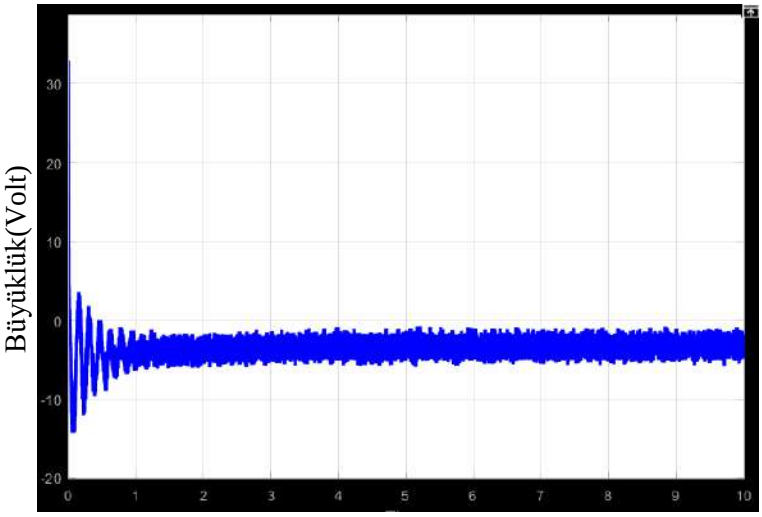
Şekil 5.1: Random bloğu

Random bloğu dış etmenleri ve gürültüleri modellemek için kullanılmıştır. Gürültünün büyüklüğü 0.1v olarak ayarlanmıştır ve gürültü doğrusal kontrolcünün girişine eklenmiştir.

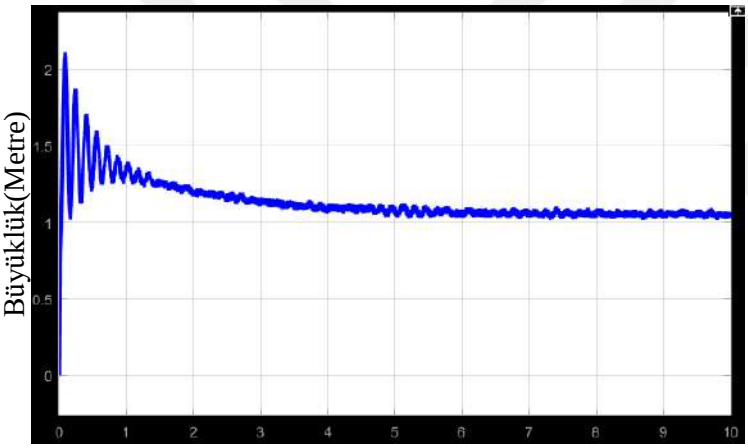


Şekil 5.2: Dış etkenler hatası modellemesi

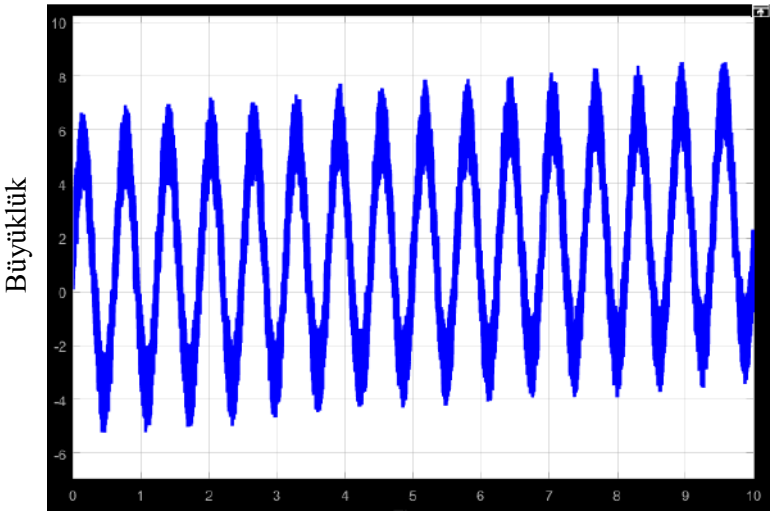
5.2 Dış Etmen Hatası Modellemesi Sonuçları



Şekil 5.3 Dış etmen hatası durumunda kontrolcü performansı



Şekil 5.4 Dış etmen hatası durumunda kapalı döngü cevabı



Şekil 5.5 :Dış etmen hatası durumunda yardımcı sinyal çıkışı

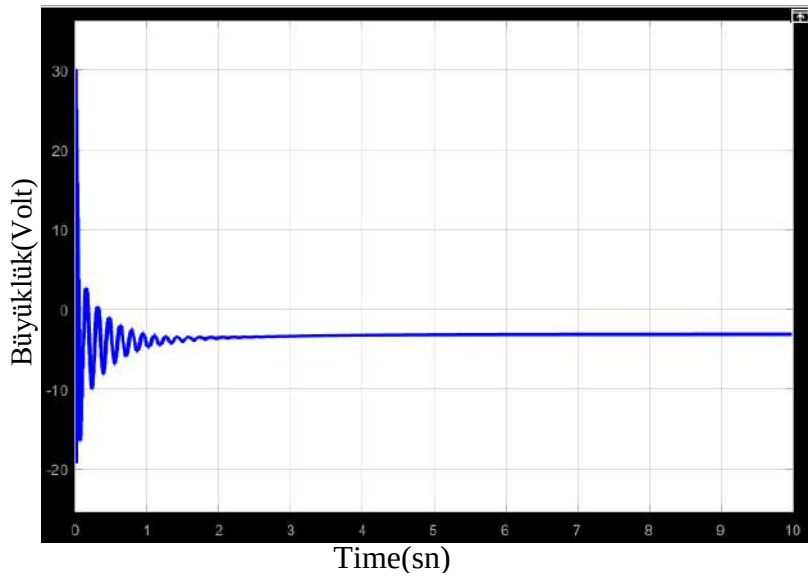
Dış etmen hatası 0.1 volt olarak ayarlanmıştır. Dış etmen hatası Şekil-5.3'de görüldüğü gibi kontrolcü performansını ve Şekil-5.4'de görüldüğü gibi dinamik sistemin kapalı döngü cevabını etkiler. Dış etmen hatası Şekil-5.5'de yardımcı sinyal çıkışı analiz edilerek kolayca tespit edilebilir. Doğru çalışan kontrolcü için yardımcı sinyalin büyüklüğü 4.05-4.95 arasında olmalıdır. Dış etmen hatasında yardımcı sinyal büyüklüğü 6.4'den başlamıştır. Yardımcı sinyal büyüklüğü belirtilen aralığın dışında kalmıştır, bu sayede kolayca hata tespiti yapılır.

5.3 Zamanlayıcı Hatası Modellenmesi

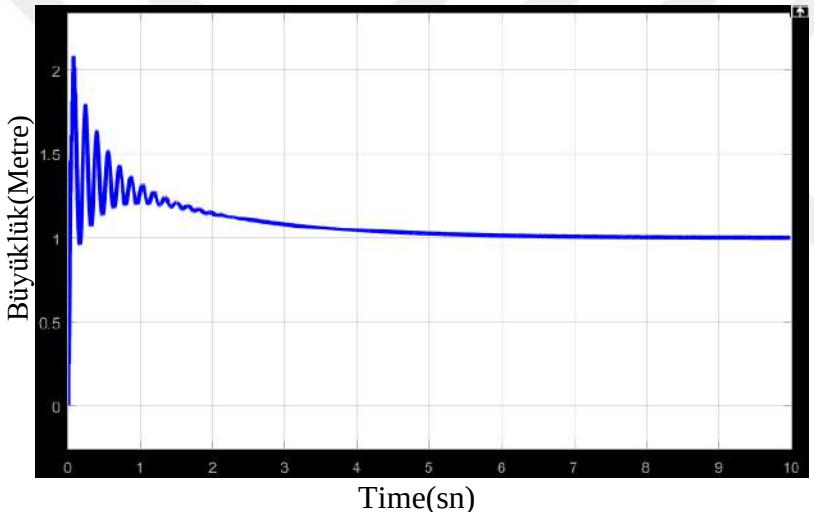
Dijital kontrolcülerde gerçekleşebilecek hatalardan biri de clock(zamanlayıcı) hatasıdır. Zamanlayıcıdan kaynaklı hatalardan dolayı kontrolcü daha hızlı ya da yavaş çalışabilir. Simülasyon ortamında zamanlayıcı hatası örnekleme zamanının değiştirilmesiyle modellenmiştir. Beklenen yardımcı sinyal hesaplanırken kontrolcünün örnekleme zamanı 0.001sn olarak belirtilmiştir. Kontrolcünün örnekleme zamanı değiştirilerek zamanlayıcı hatası modellenmiştir.

5.4 Zamanlayıcı Hatası Modellenmesi Sonuçları

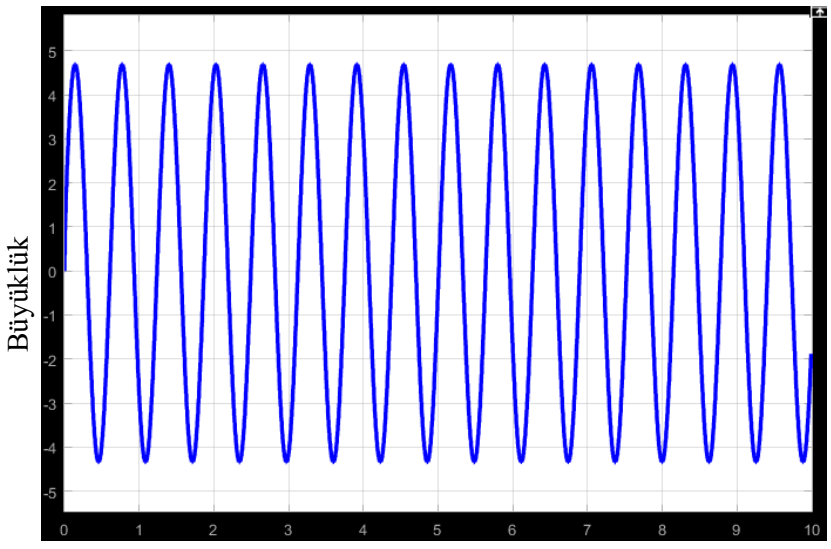
Kontrolcünün örnekleme zamanı simulasyon ortamında kademeli olarak 0.0028sn'ye kadar artırılmıştır. 0.0028sn'kadar örnekleme zamanının artırılması kontrolcü performansını ve dinamik sistemin performansını etkilememiştir.



Şekil 5.6 : Kontrolcü performansı örnekleme zamanı 0.0028sn

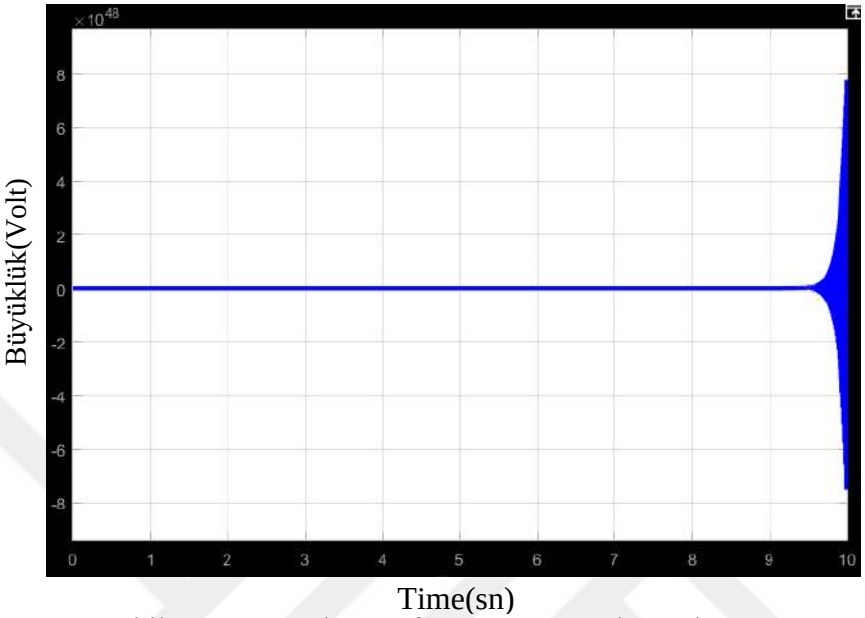


Şekil 5.7 :Dinamik sistem cevabı örnekleme zamanı 0.0028sn

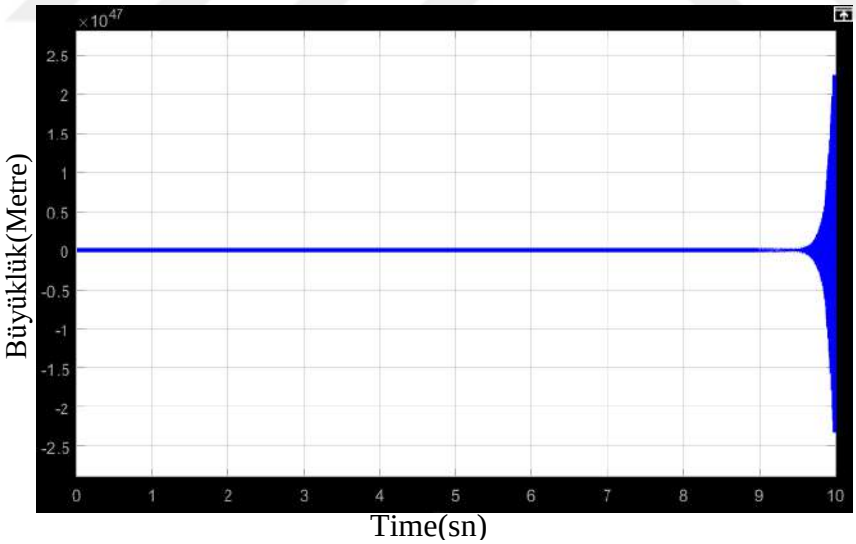


Şekil 5.8 :Yardımcı sinyal çıkışı örnekleme zamanı 0.0028sn

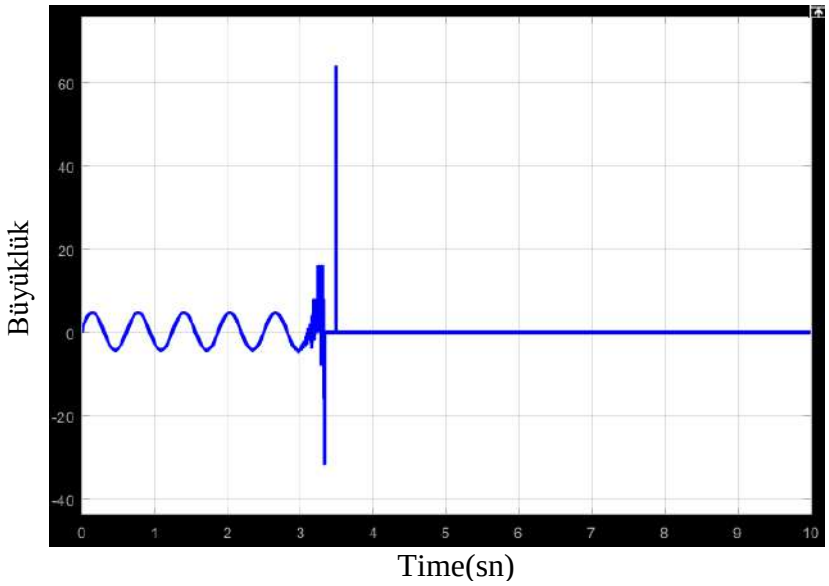
Örnekleme zamanı 0.003sn olduğunda tasarlanmış sistem kararsız davranmaya başlar kontrolcü performansı ve dinamik sistem cevabı kararsız olur. Yardımcı sinyal çıkışı analiz edilerek örnekleme zaman hatası tespit edilebilir.



Şekil 5.9: Kontrolcü performansı zamanlayıcı hatası 0.003sn



Şekil 5.10: Dinamik sistem cevabı zamanlayıcı hatası 0.003sn



Şekil 5.11: Yardımcı sinyal çıkışı zamanlayıcı hatası 0.003sn

Şekil-5.11'de görüldüğü gibi yardımcı sinyal çıkışı analiz edilerek hata tespiti yapılabilir. 3.saniyeden sonra yardımcı sinyal çıkışı doğru çalışma aralığı olan 4.05-4.95 büyüklüğünden sapmıştır.

Simülasyon ortamında dış etmen hatası ve zamanlayıcı hatası modellenmiştir. Yardımcı çıkış sinyali analiz edilerek 0.1v dış etken hatasını ve 0.003sn örnekleme zamanı hatası tespit edilmiştir. Zamanlayıcı hatasında kontrolcünün örnekleme zamanı 0.001sn başlayarak kademeli olarak artırılmıştır ve yardımcı sinyal çıkışı 0.003sn örnekleme zaman gecikmesini tespit etmiştir.

6. AKTİF HATA TESPİTİNİN DENEYSEL OLARAK GERÇEKLENMESİ

Simülasyonda elde edilen başarılı sonuçların ardından aktif hata tespitinin doğrusal kontrolcü donanımında gerçeklenmesi hedeflenmiştir. İlk olarak doğrusal kontrolcü donanımı seçilmiştir.

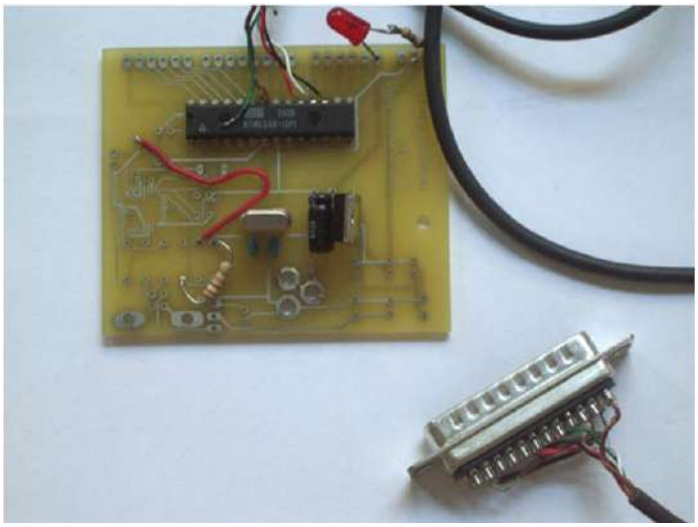
6.1 Doğrusal Kontrolcü Donanımının Seçilmesi

Kontrolcü donanımı seçimindeki temel kriterler kontrolcü donanımının kolay programlanabilmesi ve donanımın hazır iletişim protokollerine sahip olması olarak belirlenmiştir. Ayrıca aktif hata tespitinin diğer çalışmalarda da birebir olarak gerçeklenebilmesi için deneyde kullanılan kontrolcünün ucuz ve yaygın olarak bulunabilmesi de bir kriter olarak belirlenmiştir. Belirtilen kıstaslar göz önüne alındığında ATmega328P kontrolcü donanımı deneyde kullanılmıştır.

Şekil 6.1: ATmega328P kontrolcü donanımı[30]

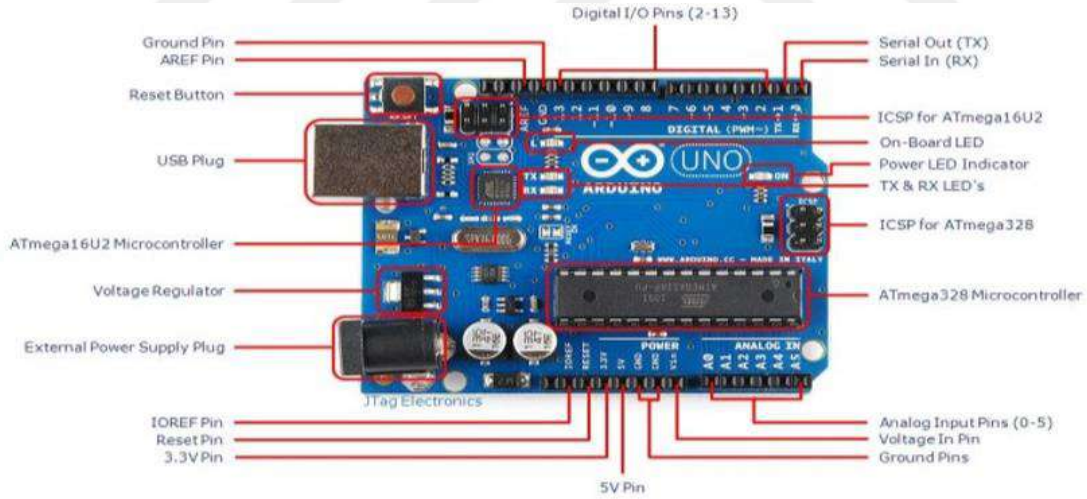
6.2 Kontrolcü Donanımının Kurulması

Kontrolcü donanımı seçildikten sonra kontrolcünün çalışması için clock sinyalinin, güç kaynağının, USB portunun kurulumu yapılmalıdır. Ayrıca güç kaynağındaki dalgalanmaların kontrolcüye etkisini azaltmak için kapasitor devresi de kontrolcü donanımına eklenmelidir.



Şekil 6.2 :Temsili Kontrolcü Kurulumu Devresi[31]

Kontrolcü donanımın kurulmasını sadeleştirmek, temassızlık ve lehim hatalarından sakınmak için Şekil 6.2'deki gibi kontrolcü donanımını kurmak yerine hazır kurulmuş ATmega328P mikrokontrolcüsünü içeren Arduino Uno donanımı tercih edilmiştir.



Şekil 6.3: Arduino Uno Elektronik Yapısı[32]

Şekil-6.3'de görüldüğü gibi Arduino Uno voltaj düzenleyici, güvenlik kapasitorü, USB portu, USB UART çevirici, sistem clock kristali ve reset butonu gibi donanımları içerir. Bu donanımlar sayesinde Arduino kartının daha istikrarlı çalışmasını sağlar.

6.3 Doğrusal Kontrolcü Yazılımın Kontrolcü Donanımına Yüklenmesi

Kontrolcü donanımı seçildikten sonra simülasyonda kullanılan doğrusal kontrolcü kontrolcü donanımına yüklenerek hata tespit metodunun deneysel olarak performans ölçümünün yapılması hedeflenmiştir.

6.3.1 İkizlenmiş Kontrolcünün Kontrolcü Donanımına Yüklenmesi

Şekil-2.7'de anlatılan ikizlenmiş kontrolcü tasarımı kontrolcü donanımına yüklenecektir. İlk olarak belirtilen kontrolcünün transfer fonksiyonu durum-uzay formuna çevrilmiştir. Durum uzay formundaki kontrolcünün donanımda gerçeklenmesi daha kolay olacaktır.

6.3.2 Doğrusal Kontrolcünün Durum-uzay Gösterimi

$$\text{Doğrusal kontrolcü tranfer fonksyonu} = \frac{30s^2 + 1004s + 400}{s^2 + 100s}$$

$$\text{Doğrusal Kontrolcu durum} - \text{uzay gösterimi}$$

$$\downarrow$$

$$\dot{X} = AX + BU \quad Y = CX + DU$$

$$X = \text{Sistemin Durumu} \quad U = \text{sistemin girişi} \quad Y = \text{Sistem Çıkışı}$$

$$A, B, C, D = \text{durum uzay matrixler}$$

Doğrusal Kontrolcünün Durum-Uzay Gösterimi

$$\downarrow$$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -100 \end{bmatrix} X + \begin{bmatrix} 1 \\ -100 \end{bmatrix} U \quad Y = [4 \;\; 20]X + [30]U \qquad (6.1)$$

6.3.3 Ayrık Zaman Durum-Uzay Gösterimi

Sürekli zamandaki durum-uzay gösterimi kontrolcü donanımına yüklenebilmesi için ayrık zaman durum-uzay gösterimi kullanılmıştır.

$$X[k+1] = AX[k] + BU[k] \quad (6.2)$$

$$Y[k] = CX[k] + DU[k] \quad (6.3)$$

$$k = Ayrık\ zaman\ adımı \quad X[k] = Ayrık\ sistem\ durumu$$

$$U[k] = Ayrık\ sistem\ girişi \quad Y[k] = Ayrık\ sistem\ çıkışı$$

$$A, B, C, D = Ayrık\ Sistem\ Matrixleri$$

Kontrolcüye ilk yüklenen ayrık durum-uzay sisteminin amacı kontrolcünün örnekleme zamanını ölçmektir. Bu nedenle kontrolcüye ilk yüklenen ayrık durum-uzay gösteriminde kullanılan sayısal parametreler önemli olmadığı için sürekli durum-uzay parametrelerinin aynısı kullanılmıştır.

6.3.4 Ayrık Durum-Uzay Kontrolcüsünün Kodlanması

Programlama dili olarak Arduino IDE(Entegre Geliştirme Ortamı) kullanılmıştır. Arduino IDE bulundurduğu hazır kütüphanelerle kontrolcünün programlanmasını ve analizini kolaylaştırmaktadır.

```
//Ayrık Zaman Matrixlerin Tanımlanması
mtx_type a[2][2]={{0 ,0},{0 ,-100 },}
mtx_type b[2][1]={{1},{-100}};
mtx_type c[1][2]={{4 , 20}};
mtx_type d[1][1]={{30}};
//Sistem Girişi Birim Basamak olarak Tanımlanmıştır.
mtx_type u[1][1]={{1}};
```

```
// Sistem Döngüsü
void loop() {

// Ayrık Sistem Gerçeklenmesi

Matrix.Multiply((mtx_type*)a, (mtx_type*)x, 2, 2, 1, (mtx_type*)f); // Ax[k]

Matrix.Multiply((mtx_type*)b, (mtx_type*)u, 2, 1, 1, (mtx_type*)f1); // Bu[k]

Matrix.Multiply((mtx_type*)c, (mtx_type*)x, 1, 2, 1, (mtx_type*)t); // Cx[k]

Matrix.Multiply((mtx_type*)d, (mtx_type*)u, 1, 1, 1, (mtx_type*)t1); // Du[k]

Matrix.Add((mtx_type*)f, (mtx_type*)f1, 2, 1, (mtx_type*)xn); // x[k+1]=Ax[k]+Bu[k]

 x[0][0] = xn[0][0]; // x[n+1]=x[n]

 x[1][0] = xn[1][0]; // x[n+1]=x[n]

Matrix.Add((mtx_type*)t, (mtx_type*)t1, 1, 1, (mtx_type*)y); // y[k]=Cx[k]+Du[k]

}
```

6.3.5 Ayrık Durum-Uzay Kontrolcüsünün İkizlenmesi

Şekil-2.3'deki ikizlenmiş kontrolcü tasarımı iki ayrı özdeş durum-uzay gösterimi kontrolcü donanımına yüklenerek dizayn edilmiştir.

```
// Durum-Uzay Kontrolcüsü-1

Matrix.Multiply((mtx_type*)a, (mtx_type*)x, 2, 2, 1, (mtx_type*)f); // Ax[k]

Matrix.Multiply((mtx_type*)b, (mtx_type*)u, 2, 1, 1, (mtx_type*)f1); // Bu[k]

Matrix.Multiply((mtx_type*)c, (mtx_type*)x, 1, 2, 1, (mtx_type*)t); // Cx[k]

Matrix.Multiply((mtx_type*)d, (mtx_type*)u, 1, 1, 1, (mtx_type*)t1); // Du[k]

Matrix.Add((mtx_type*)f, (mtx_type*)f1, 2, 1, (mtx_type*)xn); // x[k+1]=Ax[k]+Bu[k]

 x[0][0] = xn[0][0]; // x[n+1]=x[n]

 x[1][0] = xn[1][0]; // x[n+1]=x[n]
```

Matrix.Add((mtx_type)t, (mtx_type*)t1, 1, 1, (mtx_type*)y); // y[k]=Cx[k]+Du[k]*

*// **Durum-Uzay Kontrolcüsü-2***

Matrix.Multiply((mtx_type)ay, (mtx_type*)xy, 2, 2, 1, (mtx_type*)fy); // Ax[k]*

Matrix.Multiply((mtx_type)by, (mtx_type*)uy, 2, 1, 1, (mtx_type*)f1y); // Bu[k]*

Matrix.Multiply((mtx_type)cy, (mtx_type*)xy, 1, 2, 1, (mtx_type*)ty); // Cx[k]*

Matrix.Multiply((mtx_type)dy, (mtx_type*)uy, 1, 1, 1, (mtx_type*)t1y); // Du[k]*

Matrix.Add((mtx_type)fy,(mtx_type*)f1y,2,1,(mtx_type*)xny);//*
x[k+1]=Ax[k]+Bu[k]

xy[0][0]=xny[0][0];// x[n+1]=x[n]

xy[1][0]=xny[1][0]; // x[n+1]=x[n]

Matrix.Add((mtx_type)ty,(mtx_type*)t1y, 1, 1,(mtx_type*)yy); // y[k]=Cx[k]+Du[k]*

6.4 Kontrolcü Donanımın Örnekleme Zamanının Ölçülmesi

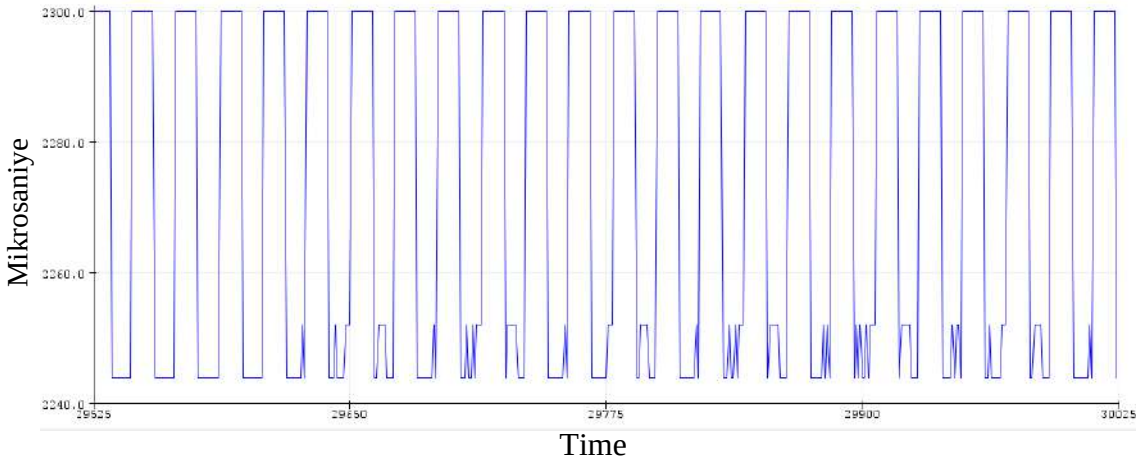
İkizlenmiş kontrol tasarımı kontrolcü donanımına yüklendikten sonra kontrolcü donanımının örnekleme zamanı ölçülmüştür. Örnekleme zamanı ölçümünde *micros()* komutu kullanılmıştır.

6.4.1 Micros() Komutu

Micros Komutu ATmega328P kontrolcüsündeki Timer0 zamanlayıcısını kullanarak kontrolcünün performansını etkilemeden microsaniye birimiyle zaman ölçümü yapar.

başlangıç = micros();
//
Kontrolcü Gerçeklenmesi
//
bitiş=micros();
Süre=başlangıç-bitiş;

Kontrolcünün ikizlenmiş kontrolcü tasarımı gerçekleştirme süresi seri port arayüzüyle bilgisayara aktarılmıştır.



Şekil 6.4: Kontrolcü donanımının ikizlenmiş kontrolcü tasarımını gerçekleme süresi

Şekil-6.4'de görüldüğü gibi kontrolcü donanımı ikizlenmiş kontrolcü tasarımını gerçeklerken kesin istikrarlı bir örnekleme zamanı yoktur.Bu durumun nedeni mikrokontrolcülerdeki clock iletimindeki çarpıklıklar, seğirmeler ve güç kaynağındaki gürültüler olabilir[27].

Kontrolcü donanımın örnekleme zamanı 2300-2245 microsaniye arasında değismektedir. Diğer bir deyişle kontrolcü donanımı saniyede yaklaşık 430 kez ayrık kontrolcü gerçeklemesi yapar.

Kontrolcü donanımın örnekleme zamanı 2300-2245 aralığının ortanca değeri olmasından 2270 microsaniye olarak belirlenmiştir.

6.5 Sürekli Kontrolcünün Ayrık Kontrolcüye Çevirimi

Kontrolcü donanımın örnekleme zamanı 2270 microsaniye olarak belirlenmiştir. Bu nedenle sürekli kontrolcü örnekleme zamanı 2270 microsaniye olarak ayrık kontrolcüye dönüştürülmüştür.

$$\text{Sürekli kontrolcü tranfer fonksyonu} = \frac{30s^2 + 1004s + 400}{s^2 + 100s} \quad (6.2)$$

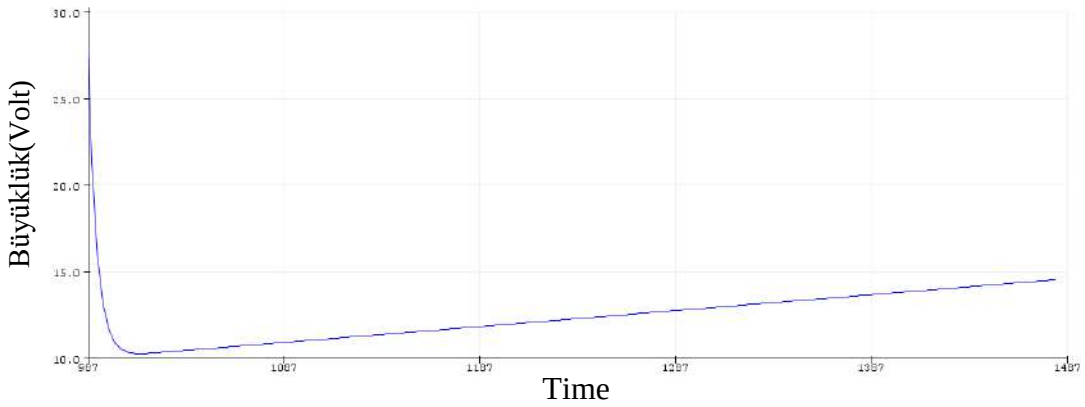
Foh Dönüşümü
Örnekleme Zamanı(2270 ms)

$$Ayrık\ Kontrolcü\ transfer\ fonksyonu = \frac{27.897z^2 - 53.7532z + 25.858}{z^2 - 1.7969z + 0.7969} \quad (6.3)$$

Ayrık Kontrolcü Durum-Uzay Gösterimi

$$\dot{X} = \begin{bmatrix} 1 & 0 \\ 0 & 0.7969 \end{bmatrix} X + \begin{bmatrix} 0.0023 \\ -0.2031 \end{bmatrix} U \quad Y = [4 \;\; 17.8924]X + [27.897]U$$

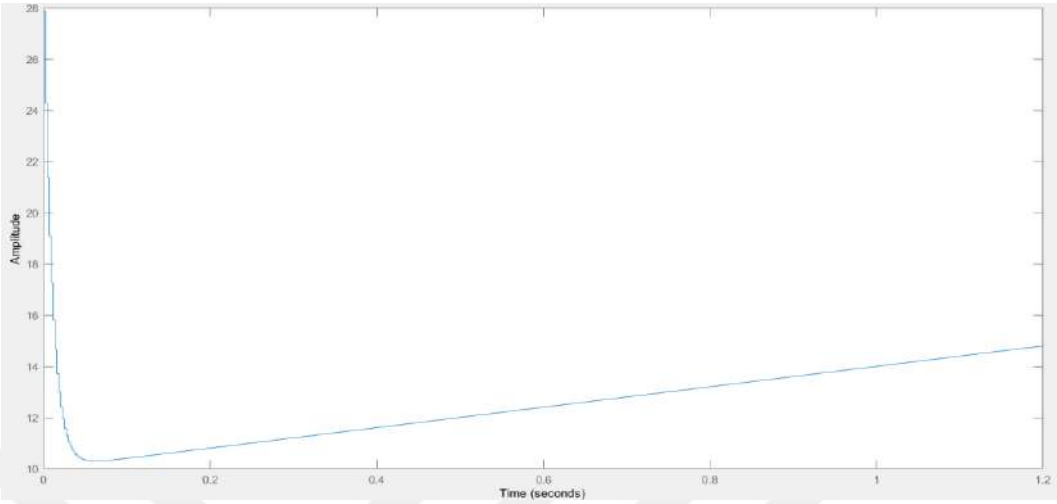
6.6 Kontrolcü Donanımın Açık Döngü Birim Basamak Cevabı



Şekil 6.5 : Kontrolcü donanımın birim basamak cevabı

Şekil-6.5'de kontrolcü donanımın birim basamak cevabı gösterilmiştir. Kontrolcü donanımın birim basamak cevabı seri port arayüzüyle bilgisayara aktarılıp çizdirilmiştir.

Simülasyon ortamıyla ile kontrolcü donanımının paralel çalıştığını göstermek için Şekil-6.6'de simülasyon ortamında ayrık kontrolcünün birim basamak cevabı gösterilmiştir.



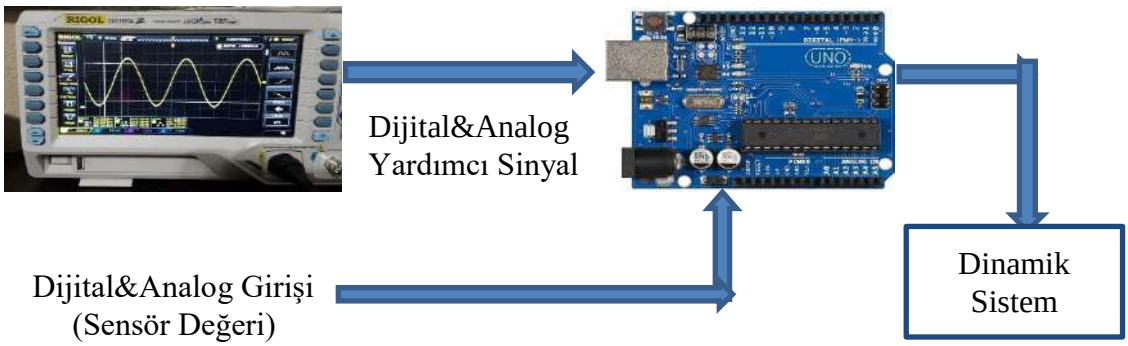
Şekil 6.6 :Simulasyon ortamında ayrık kontrolcünün açık döngü birim basamak cevabı

Kontrolcü donanımıyla simülasyon ortamının sonuçları aynıdır. Bu eşlik sayesinde kontrolcü donanımına ayrık kontrolcünün düzgün olarak yüklendiği ve başarılı bir simülasyon modeli oluşturulduğu çıkarımı yapılabilir.

6.7 Yardımcı Sinyalin Gerçeklenmesi

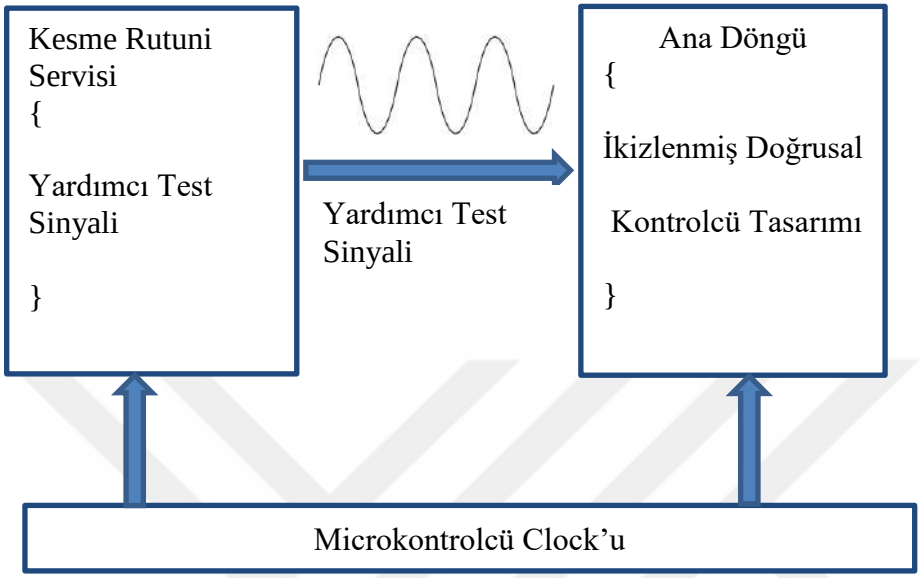
Yardımcı sinyal kontrolcü donanımına dışardaki bir kaynaktan verilebileceği gibi kontrolcü donanımındaki zamanlayıcı kullanarak da oluşturulabilir.

Yardımcı sinyalin dışardaki kaynaktan elde edilmesi fazladan maliyete ve donanım karmaşıklığına neden olabilir ama istikrarlı ve güvenli bir dış kaynak seçilmesi durumunda yardımcı sinyalden kaynaklı belirsizlikler azaltılabilir.



Şekil 6.7: Yardımcı sinyalin dış kaynakla gerçeklenmesi

Yardımcı sinyal dışardaki bir kaynaktan elde edilebileceği gibi doğrusal kontrolcünün clock'nu kullanarak kontrolcü performansını etkilemeyecek şekilde ISR(Kesme Rutini Servisi) ile elde edilebilir.



Şekil 6.8: Yardımcı sinyalin mikrokontrolcü clock'uyla gerçeklenmesi

Kesme rutini servisi mikrokontrolcüye kesme(interrupt) gelmesi durumunda ana döngüyü bölerek kesme rutini kodunu çalıştırır. Kesme rutin kodunun kesme durumunda gerekli cevabı verecek şekilde ve mümkün olduğu kadar optimal olmalıdır. Diğer bir deyişle, kesme rutin kodunun microkontrolcünün performansını etkilememesi için çok kısa sürede gerçeklenmesi gerekir[33]

6.7.1 Kesme Rutini Servisi Yapılandırılması

Atmega328P mikrokontrolcüsü 3 tane zamanlayıcı(Timer) içerir. Kesme rutini servis kodu Timer1 zamanlayıcına eklenmiştir.
Timer1 zamanlayıcısı periyodik olarak kesme rutini servis kodunu çalıştırılacak şekilde yapılandırılmıştır.

TCCR1A → Zamanlayıcı Sayıcısı Kontrol Registerı(Sayma Modu)

TCCR1B → Zamanlayıcı Sayıcısı Kontrol Registerı(Sayma Hızı)

OCR1A → Çıkış Karşılaştırma Registerı

TCCR1A = 0; // Zamanlayıcı kontrol registerı artan şekilde sayma yapmaya ayarlanmıştır

TCCR1B = 0; // Zamanlayıcı kontrol registarı resetlenir.

TCNT1 = 0; // Başlangıç Zamanının Resetlenmesi

OCR1A = 15; // Kesme Rutuni Servis Kodunun Çalışma Frekansı

TCCR1B |= (1 << WGM12); // Zamanlayıcının OCR1A değerine kadar çalışması ayarlanır.

TCCR1B |= (1 << CS12) | (1 << CS10); // Zamanlayıcının Sayma Hızının ayarlanması

TIMSK1 |= (1 << OCIE1A); // Sayma bitince TIMER1_COMPA vectöründeki kodu çalıştır.

Atmega32 serisindeki timer(Zamanlayıcı) registerları ve yapılandırılması [34]'den alınmıştır, daha detaylı bir timer(zamanlayıcı) yapılandırılması için referans [34] incelenebilir.

6.7.2 Kesme Rutini Servisi Frekansının Ayarlanması

Atmega328P microkontrolcüsünde clock frekansı 16000000 Hz'dir, bu frekans istikrarlı bir işlem için çok yüksektir. Bu nedenle clock frekansı 1024'te birine indirilmiştir.

TCCR1B |= (1 << CS12) | (1 << CS10); Clock Frekansı 1024'de birine indirgenmiştir

$$\frac{16000000Hz}{1024} = 15625\ Hz \rightarrow 64\ Microsaniye \qquad (6.4)$$

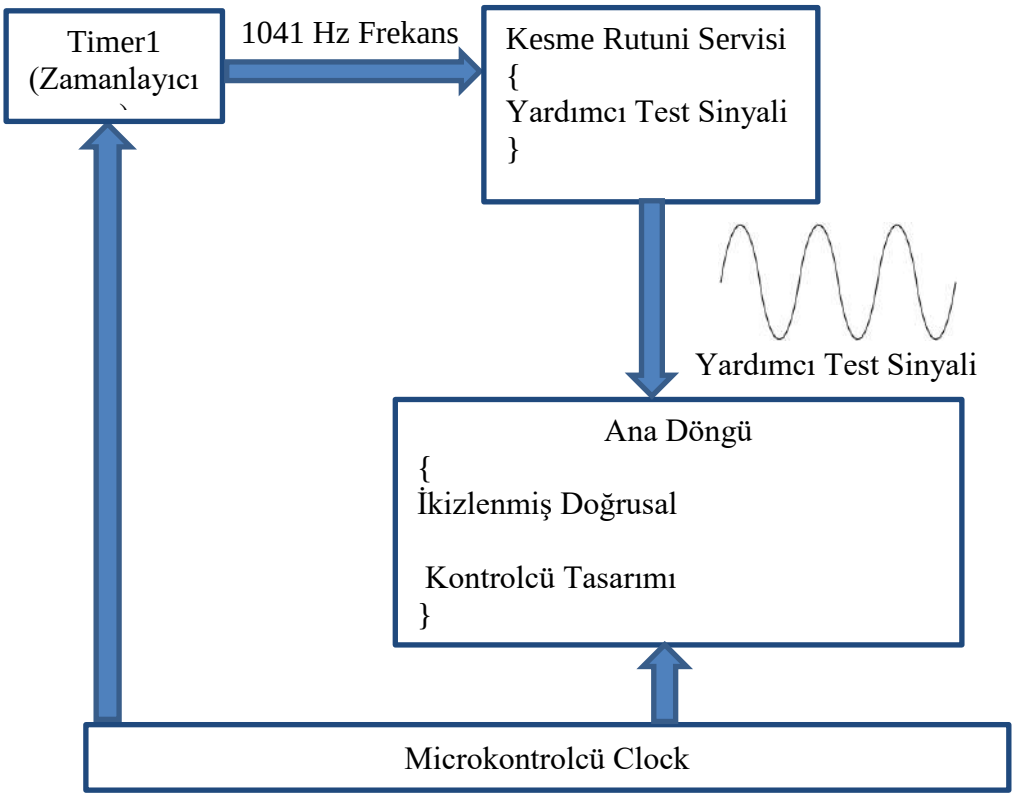
OCR1A = 15 // Kesme Rutuni Frekansı belirlenmiştir

$$15 \times 64Ms = 960\ Ms \rightarrow 1041\ Hz \qquad (6.5)$$

Kesme rutini frekansı 1041 Hz olarak belirlenmiştir, bu frekans değeri *OCR1A* registerı değiştirilerek ya da clock frekansı indirgenme değeri değiştirerek ayarlanabilir.

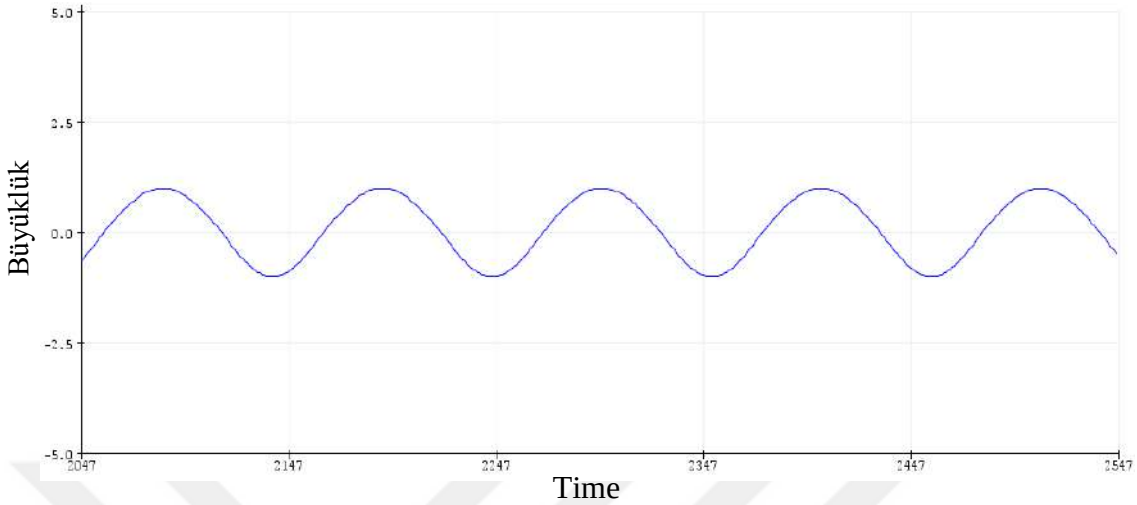
6.7.3 Kesme Rutini Servisinin Yardımcı Sinyal Gerçeklenme Kodu

```
ISR(TIMER1_COMPA_vect) // Timer1 ile Kesme Rutuni İlişkilendirilir
{
yardımcı_sinyal= sin(2 * 3.14 * Fc * ts); // Yardımcı Sinus sinyalinin gerçeklenmesi
Fc=Yardımcı Sinyal Frekansı  ts=Örnekleme Zamanı
ts = ts + 0.001;// Örnekleme Adımı her döngüde 0.001 saniye artar.
 return Yardımcı_sinyal; // Kesme Rutuni kodu bitiminde yardımcı sinyal değerine dönülür.
}
```



Şekil 6.9 : Yardımcı sinyal gerçeklenmesi diagramı

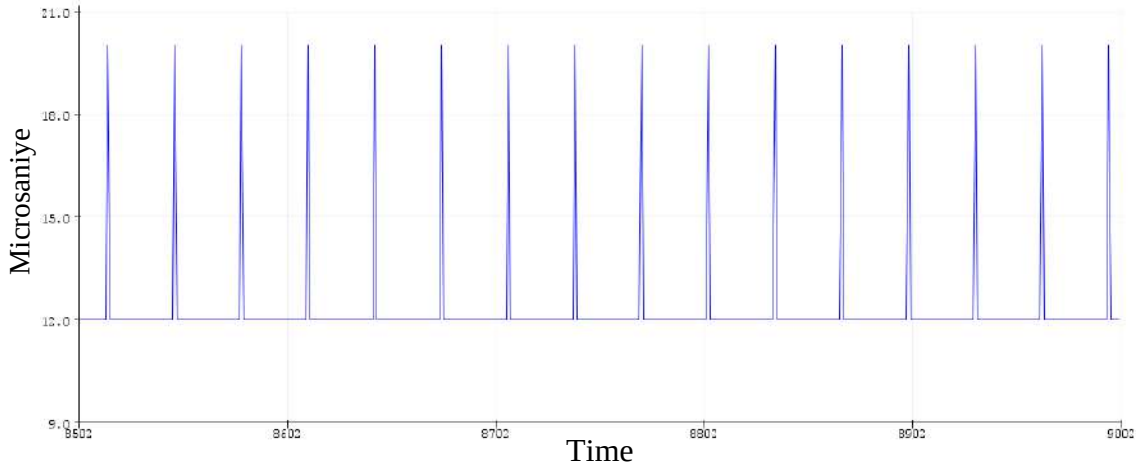
Şekil-6.9'deki diyagram Atmega328P mikrokontrolcüsünde belirtilen kodları kullanarak gerçeklenmiştir.



Şekil 6.10 : Yardımcı test sinyali sin(20t)

Belirtilen diagramın Atmega328P mikrokontrolcüsüne yüklendikten sonra yardımcı test sinyal girişi serial port aracılığı ile bilgisayar ekranına aktarılmıştır ve Şekil-6.10'da çizdirilmiştir.

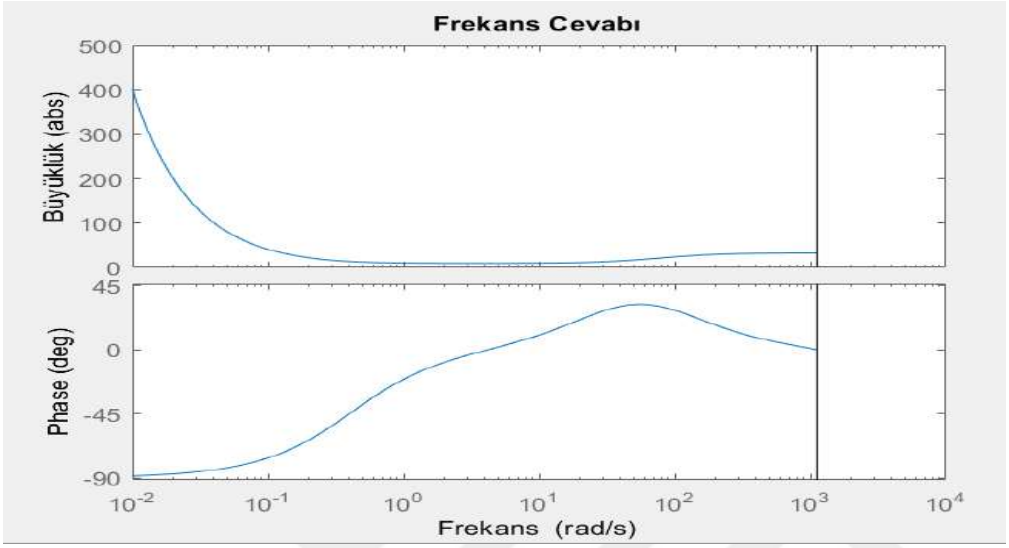
Daha önceden belirtildiği gibi kesme rutini servisi kodu mikrokontrolcü performansını etkilememek için olabildiğince kısa sürede gerçeklenmelidir. Şekil-6.11'da kesme rutini servis kodunun gerçeklenme süresi ölçülmüştür ve kesme rutini servis kodunun 12-20 microsaniye arasında gerçeklendiği görülmüştür. Bu süre mikrokontrolcü performansını etkilemeyecek kadar kısadır.



Şekil 6.11: Kesme rutuni servisi gerçeklenme süresi

6.8 Yardımcı Sinyal Çıkışı

Öncelikle, ayrık kontrolcünün frekans cevabı simülasyon ortamında çizdirilerek beklenen yardımcı sinyal analiz belirtilir.



Şekil 6.12: Ayrık kontrolcünün frekans cevabı

Ayrık kontrolcü frekans cevabı analiz edildiğinde 20rad/s frekansının 11.5 büyüklüğüne karşılık geldiği görülür. Diğer bir deyişle ayrık kontrolcüye 20rad/s frekansına sahip birim sinüs sinyali gönderildiğinde kontrolcünün 11.5 büyüklüğünde bir cevap vermesi beklenir. Bu nedenle beklenen yardımcı sinyal 11.5sin(20t) olarak belirlenir

6.9 Aktif Hata Tespiti Yapısının Mikrokontrolcüye Kodlanması

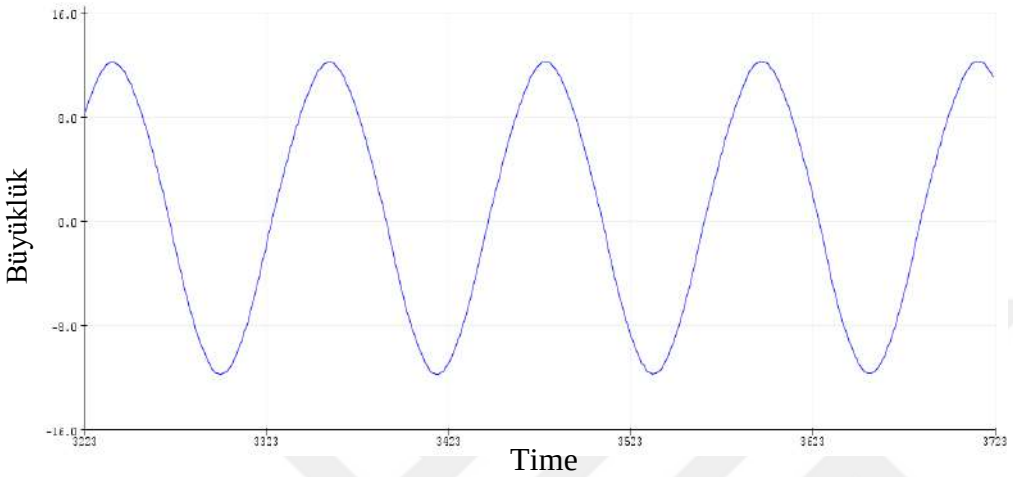
Mikrokontrolcüdeki aktif hata tespit yapısı diyagramı şekil-6.9'de gösterilmiştir. Bu diyagramın kodlaması çalışmanın diğer araştırmacılar tarafından da gerçeklenebilmesi için detaylı olarak açıklanmıştır.

```
mtx_type u[1][1] = {{1+Yardımcı_sinyal }};
 // Doğrusal Kontrolcü-1'e positif sinus sinyali eklenmesi
 mtx_type uy[1][1] = {{1-Yardımcı_sinyal}};
 // Doğrusal Kontrolcü-2'e negatif sinus sinyali eklenmesi
```

```
// Durum-Uzay Kontrolcüsü-1

Matrix.Multiply((mtx_type*)a, (mtx_type*)x, 2, 2, 1, (mtx_type*)f); // Ax[k]

Matrix.Multiply((mtx_type*)b, (mtx_type*)u, 2, 1, 1, (mtx_type*)f1); // Bu[k]

Matrix.Multiply((mtx_type*)c, (mtx_type*)x, 1, 2, 1, (mtx_type*)t); // Cx[k]

Matrix.Multiply((mtx_type*)d, (mtx_type*)u, 1, 1, 1, (mtx_type*)t1); // Du[k]

Matrix.Add((mtx_type*)f, (mtx_type*)f1, 2, 1, (mtx_type*)xn);

// x[k+1]=Ax[k]+Bu[k]

 x[0][0] = xn[0][0]; // x[n+1]=x[n]

 x[1][0] = xn[1][0]; // x[n+1]=x[n]

Matrix.Add((mtx_type*)t, (mtx_type*)t1, 1, 1, (mtx_type*)y); // y[k]=Cx[k]+Du[k]


// Durum-Uzay Kontrolcüsü-2

Matrix.Multiply((mtx_type*)ay, (mtx_type*)xy, 2, 2, 1, (mtx_type*)fy); // Ax[k]

Matrix.Multiply((mtx_type*)by, (mtx_type*)uy, 2, 1, 1, (mtx_type*)f1y); // Bu[k]

Matrix.Multiply((mtx_type*)cy, (mtx_type*)xy, 1, 2, 1, (mtx_type*)ty); // Cx[k]

Matrix.Multiply((mtx_type*)dy, (mtx_type*)uy, 1, 1, 1, (mtx_type*)t1y); // Du[k]

Matrix.Add((mtx_type*)fy, (mtx_type*)f1y, 2, 1,(mtx_type*)xny); //
x[k+1]=Ax[k]+Bu[k]

 xy[0][0]=xny[0][0];// x[n+1]=x[n]

 xy[1][0]=xny[1][0]; // x[n+1]=x[n]

Matrix.Add((mtx_type*)ty, (mtx_type*)t1y, 1, 1,(mtx_type*)yy); //
y[k]=Cx[k]+Du[k]

Matrix.Subtract((mtx_type*)y, (mtx_type*)yy, 1, 1,(mtx_type*)out);

//  Kontrolcü-1 ile Kontrolcü-2'nin çıkışlarının farkının alınması

Matrix.Add((mtx_type*)y, (mtx_type*)yy, 1, 1,(mtx_type*)rout);

// Kontrolcü-1 ile Kontrolcü-2'nin çıkışlarının toplanması
```

rout[0][0]=rout[0][0]/2;//Kontrolcü Çıkışı(Dinamik sisteme aktarılmak için hazır)

out[0][0] = out[0][0]/2;//Yardımcı Sinyal Çıkışı(Hata Tespiti Yapmak için Kullanılır)



Şekil 6.13: Yardımcı sinyal çıkışı 12sin(20t)

Deneysel olarak ölçülen yardımcı sinyal 12sin(20t)'dir

$$Sapma\ Miktarı = \frac{12 - 11.5}{11.5} = \%4.35 \qquad (6.6)$$

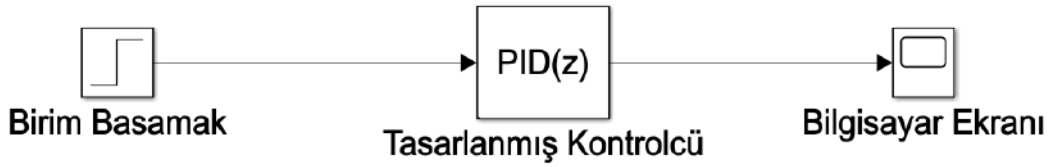
Sapma miktarı %4.35 olarak hesaplanmıştır. Sapma miktarı hata sınırı olarak belirlenen $\pm\%10$ değerinin altındadır, kontrolcü hatasız çalışıyor çıkarımı yapılabilir.

6.10 Hata Tespit Methodunun Kontrolcü Performansına Etkisi

Hata Tespit yapısı Atmega328P mikrokontrolcüsünde gerçeklendirilmiştir. Yardımcı çıkış sinyali analiz edilerek mikrokontrolcünün hata durumu tespit edilmiştir. Bu başlık altında hata tespit metodunun mikrokontrolcü performansına etkisi analiz edilmiştir.

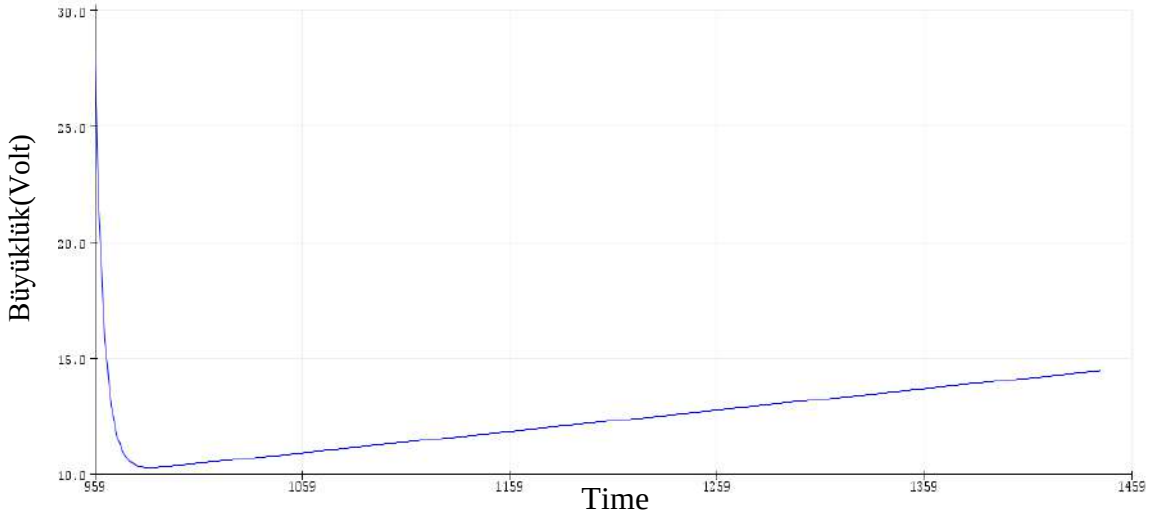
Simülasyon ortamında hata tespit metodunun kontrolcü performansına etkisi olmadığı gösterilmiştir. Bu başlık altında deneysel olarak da hata tespit metodunun kontrolcü performansına etkisi olmadığının gösterilmesi hedeflenmiştir.

İlk olarak Şekil-6.14'de gösterildiği gibi tasarlanmış kontrolcünün açık döngü diyagramı Atmega328P kontrolcüsüne gömülmüştür



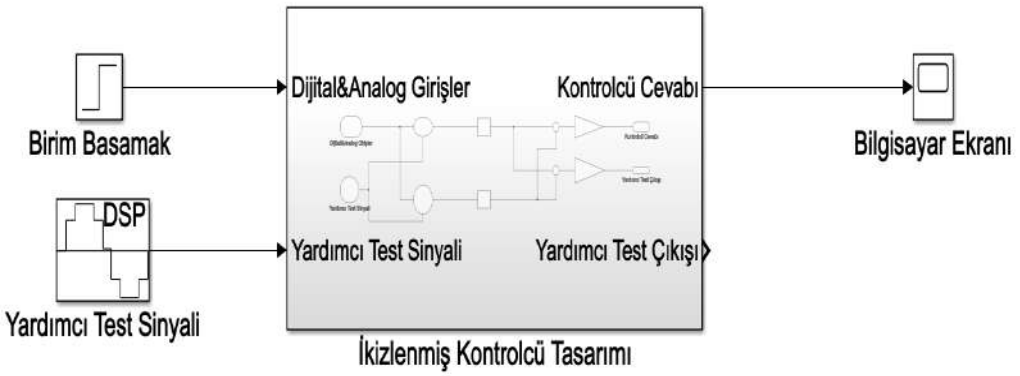
Şekil 6.14 :Tasarlanmış kontrolcü açık döngü diyagramı

Tasarlanmış kontrolcünün cevabı seri port arayüzüyle bilgisayar ekranına aktarılmış ve çizimi yapılmıştır.



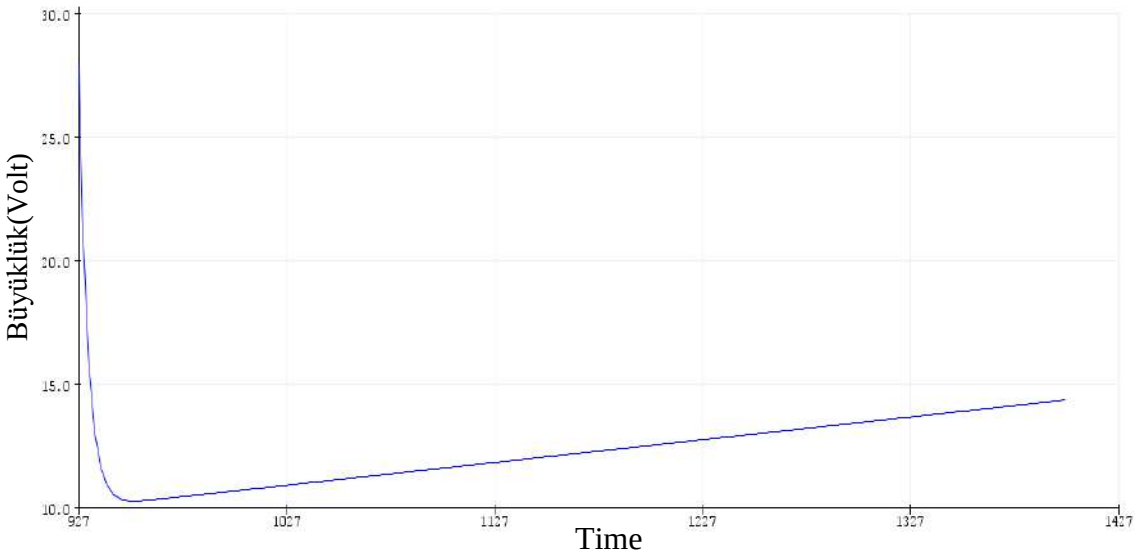
Şekil 6.15: Tasarlanmış kontrolcü açık döngü cevabı

Tasarlanmış kontrolcünün açık döngü cevabı elde edildikten sonra hata tespit mekanizmasının da açık döngü cevabı analiz edilmiştir. Öncelikle hata tespit mekanizmasının açık döngü cevabı Şekil-6.9'da gösterildiği gibi Atmega328P kontrolcüsüne gömülmüştür.



Şekil 6.16: Hata Tespit Mekanizmasının Açık Döngü Gösterimi

Hata tespit mekanizmasının açık döngü cevabı şekil-6.17'de gösterilmiştir. Şekil-6.15'de de tasarlanmış kontrolcünün açık döngü cevabı gösterilmiştir. Tasarlanmış kontrolcü ve hata tespit mekanizmasının açık döngü cevabı birebir aynıdır. Bu sayede hata tespit mekanizmasının kontrolcü performansını etkilemediği gösterilmiştir.



Şekil 6.17 :Hata tespit mekanizmasının açık döngü cevabı

7. DENEYSEL ORTAMDA HATA TESPİT PERFORMANSININ ANALİZİ

İlk olarak hata tespitinin mikrokontrolcüdeki döngü süresi hatası durumunda performansı analiz edilmiştir. Mikrokontrolcüde hata olmasını beklemek çok uzun süre alacağı için mikrokontrolcüye hata enjeksiyonu yapılmıştır.

7.1 Zaman Hatası Enjeksiyonu

Mikrokontrolcüde zaman enjeksiyonu ana döngüye delay() komutu eklenerek yapılmıştır. Delay komutu mikrokontrolcünün ana döngüsünde basit sayma işlemi yaptırarak mikrocontrolcünün belirtilen sürede ana döngüsünün beklemesini sağlar[34]

7.1.1. Delay() Fonksiyonu

```
void delay(unsigned long ms)  // Fonksiyon Tanımı
{
      uint32_t start = micros();// İşlemci çalıştırılmasından beri geçen zaman
      while (ms > 0)// Bekleme zamanı 0'dan büyükse saymayı başla
{
            while ( ms > 0 && (micros() - start) >= 1000)
// Sayma işlemi bitene kadar saymaya devam et
{
                  ms--;
                  start += 1000;
            }
      }
}
```

Delay fonksiyonu [35]'den alınmıştır.

7.2 Zaman Hatası Performans Ölçümü

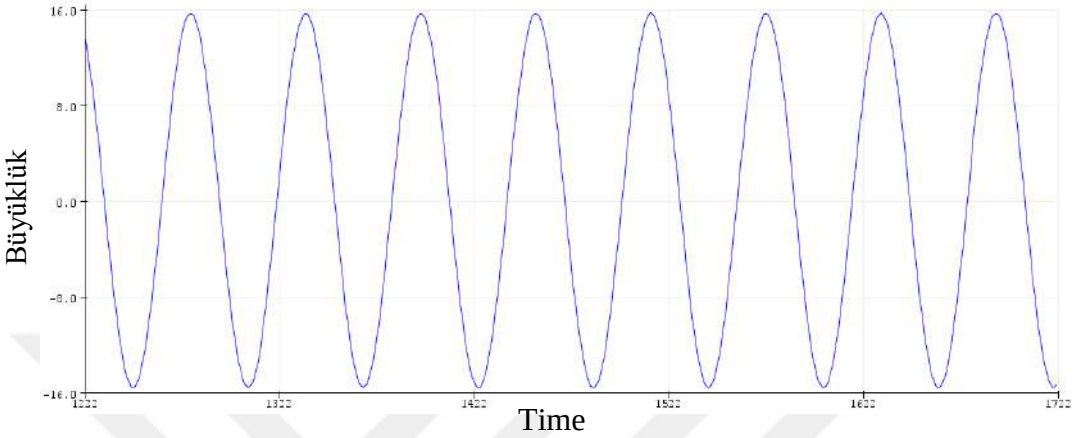
Zaman hatası enjeksiyonu kademeli olarak yapılmıştır. İlk olarak 1 milisaniye zaman hatası enjekte edilmiştir ve yardımcı sinyal çıkışı büyüklüğü kaydedilmiştir. Aynı işlem enjekte edilen zaman hatasının büyüklüğü artırılarak tekrarlanmıştır ve değerler kaydedilmiştir.

Tablo 7.1 : Hata Tespit Metodunun Zaman Hatası Performansı.

Zaman Hatası (milisaniye)	Yardımcı Sinyal Çıkışı Büyüklüğü	Hata Sapma Payı Yüzdesi
1	12	4.35
2	12	4.35
3	12	4.35
4	12	4.35
5	12	4.35
6	12	4.35
7	12.42	8
8	12.97	12.78
9	13.4	16.52
10	14	21.74
11	14.53	26.35
12	15.06	30.96
13	15.61	35.74
14	16.07	39.74
15	16.66	44.87
30	22.48	95.48

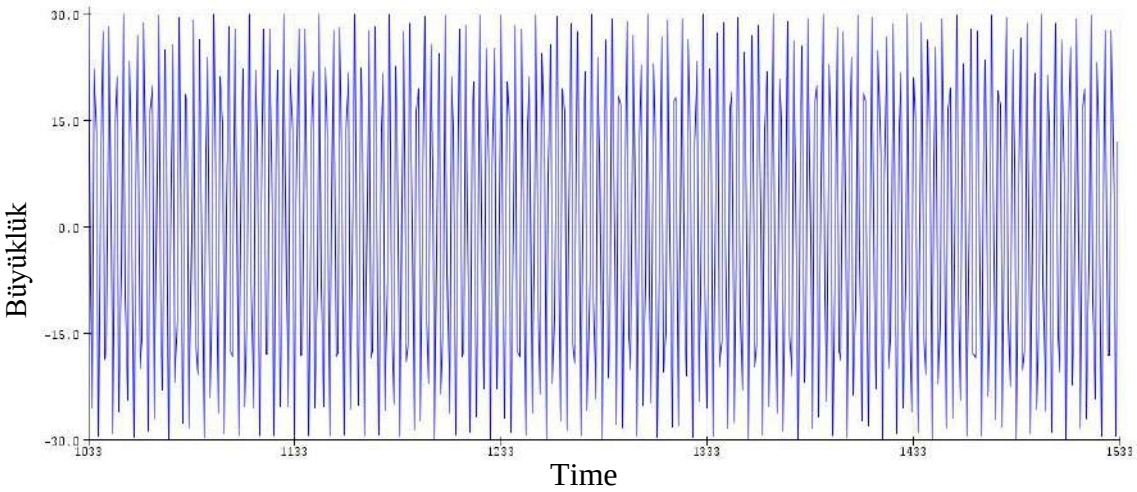
Tablo-1'de hata tespit metodunun zaman hatası performansı gösterilmiştir. Hata tespit metodu 8 milisaniye' den büyük zaman hatalarını tespit edebilmektedir. Diğer bir deyişle 0.008 saniyeden büyük zaman hataların yardımcı sinyal çıkışı $\pm\%10$ hata sınırının üstündedir. Ayrıca zaman hatasının büyüklüğü artıkça yardımcı sinyalin sapma yüzdesi de artmaktadır.

Şekil-7.1’de 13 milisaniyelik zaman hatası durumunda yardımcı test sinyalinin çıkışı gösterilmiştir. Yardımcı test sinyali zaman hatasına rağmen hala sinüzoidal özellik gösterir ama büyüklüğü 15.61 olarak ölçülmüştür.



Şekil 7.1 :13 milisaniye zaman hata durumundaki yardımcı test çıkışı

Ayrıca zaman hatasının büyük değerlere ulaşması durumunda yardımcı sinyal sinüzoidal özelliğini kaybetmiştir. Şekil-7.2’de zamanlama hatasının 200 milisaniye olması durumundaki yardımcı sinyal çıkışı gösterilmiştir.



Şekil 7.2: 250 milisaniye zaman hatası durumunda yardımcı sinyal çıkışı

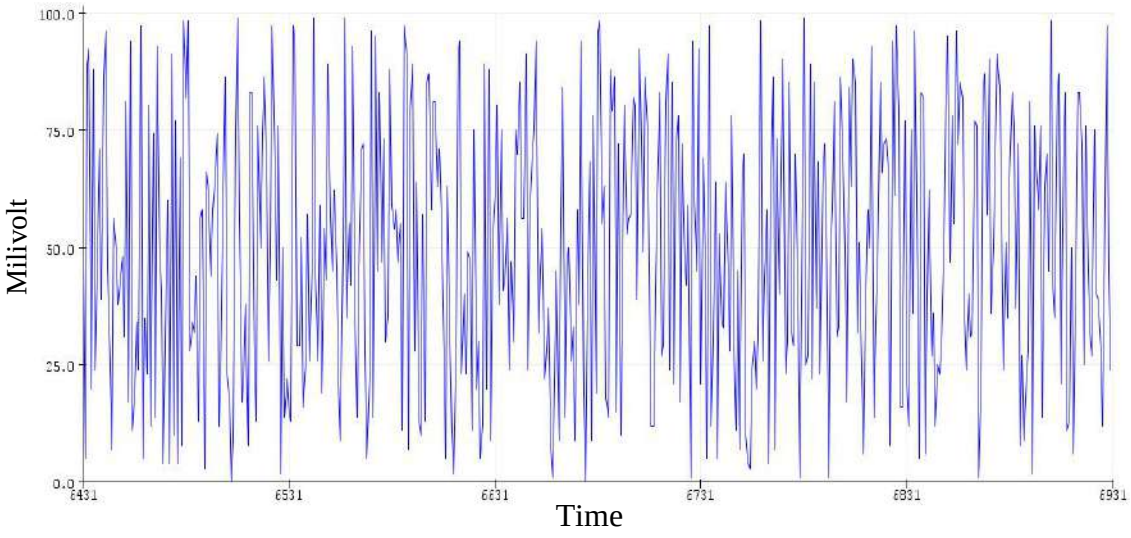
Zamanlama hatası büyüklüğü artıkça yardımcı sinyal çıkışı sinüzoidal özelliğini kaybetmeye başlamıştır. Bu sayede yardımcı sinyal çıkışının frekansı izlenerek hata yapılabileceği önerisi de desteklenmiştir.

7.3 Dış Etmen Hatası Enjeksiyonu

Simülasyon ortamında dış etmen hatası enjeksiyonu şekil-5.2'de detaylı olarak hali hazırda gösterilmiştir. Dış etmen hatası deneysel olarak Atmega328P'ye enjekte edilmiştir. Dış etmen hatasının benzetimini yapmak için mikrokontrolcü içerisine gürültü sinyali enjekte edilmiştir

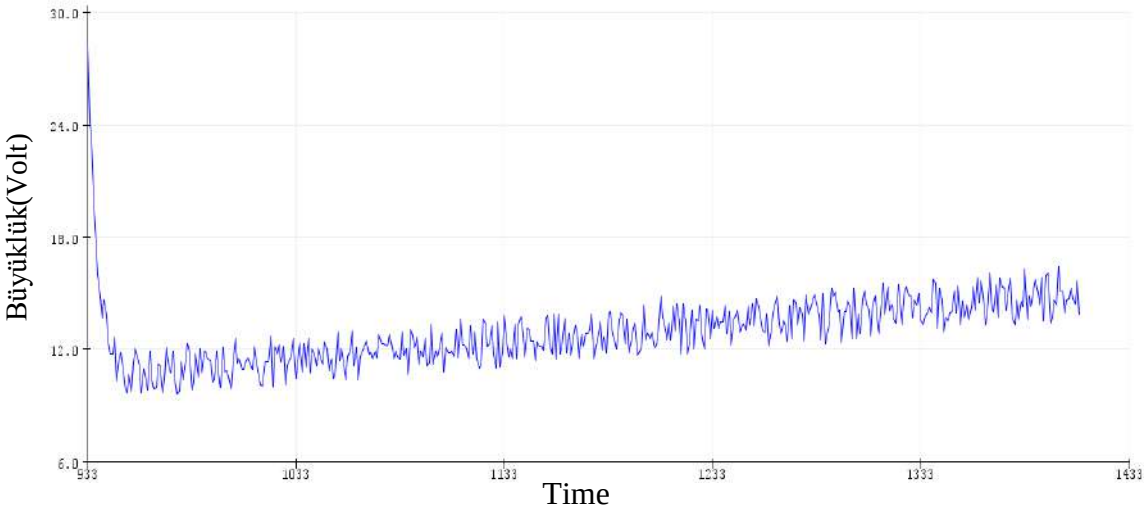
7.3.1 Gürültü Sinyalinin Gerçeklenmesi

Gürültü Sinyalinin gerçeklenmesi random() komutuyla yapılmıştır. Random() komutuyla en fazla 0.1v büyüklüğe ulaşacak şekilde rastgele oluşturan sinyal dizileri elde edilmiştir. Şekil-7.3'de 0-100 mv arasında oluşturan gürültü sinyali gösterilmiştir.



Şekil 7.3 : Gürültü sinyali 0-100 mv

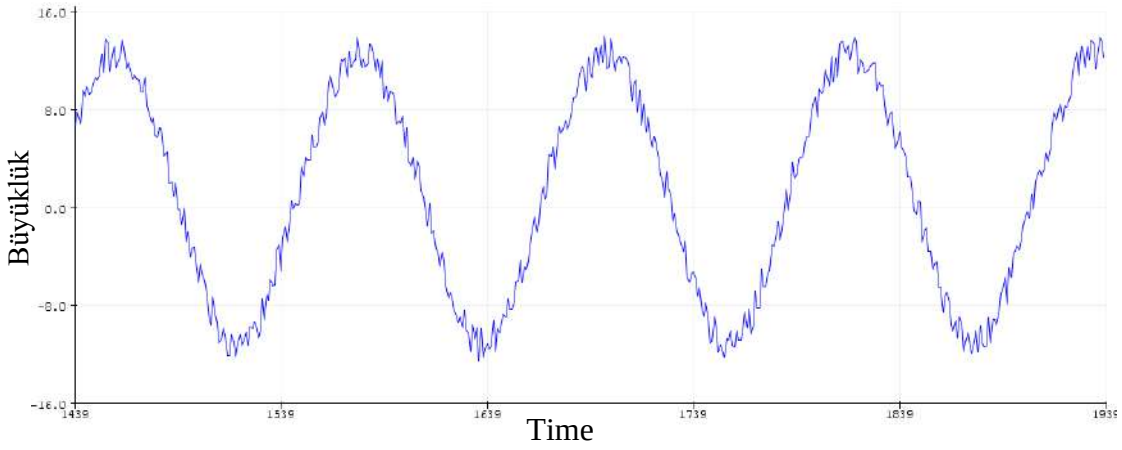
Gürültü sinyali elde edildikten sonra Şekil-5.2'daki gösterildiği gibi gürültü sinyali yardımcı test sinyaline ve doğrusal kontrolcü girişine eklenir. Gürültünün kontrolcünün açık döngü birim cevabına ve yardımcı test sinyali çıkışına olan etkisi analiz edilir. Şekil-7.4'da gürültünün kontrolcünün açık döngü birim cevabına etkisi görülmektedir.



Şekil 7.4 : Gürültü ortamında kontrolcünün açık döngü birim cevabı

Gürültü kontrolcünün açık döngü birim basamak cevabında görülür bir bozulmaya neden oluyor. Şekil-6.15'de gürültüsüz ortamdaki kontrolcünün açık döngü birim basamak cevabı gösterilmiştir. Daha detaylı bir analiz için Şekil-6.15 ve Şekil-7.4 karşılaştırılabilir.

Doğrusal kontrolcü donanımındaki gürültünün yardımcı test sinyal çıkışına etkisinin gösterilmesi için Şekil-7.5'de gürültülü ortamdaki yardımcı test sinyali çıkışı gösterilmiştir.



Şekil 7.5 : Gürültülü ortamdaki yardımcı test sinyali çıkışı

Dış etmen hatası durumunda yardımcı test çıkışının büyüklüğü yaklaşık 12.85 olarak ölçülmüştür.

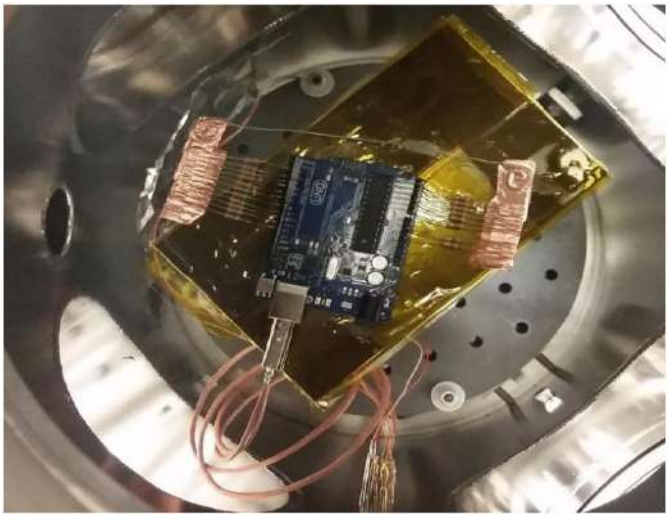
$$Hata\ Sapma\ Yüzdesi = \frac{12.85 - 11.5}{11.5} = \%11.74 \qquad (7.1)$$

Aktif hata tespiti metodu deneysel ortamda Atmega328P kontrolcüsündeki 0.1v büyüklüğündeki dış etmen hatasını tespit etmiştir.

7.4 Bit Durumu Değişikliği Hatası

Mikrokontrolcünun yüksek şiddetli radyasyona fazla sürede maruz kalması durumunda Flash hafızasındaki bitlerin durumu değişebilir[36].[36]'de Atmega328P mikrokontrolcüsünü kullanan Ardiuno Uno kartı uzay şartlarının sağlandığı özel kapsülde test edilmiştir. Test sonucunda kapsülde uygulanan şiddetli radyasyondan sonra mikrokontrolcünün flash hafızasındaki iki bitin durumunun değiştiği gözlemlenmiştir.

Şekil 7.6 : Uzay şartlarının sağlandığı özel kapsül[36]



Şekil 7.7 : Mikrokontrolcü dayanaklığını ölçen deney düzeneği[36]

7.4.1 Bit Durum Değişikliği Hatasının Gerçeklenmesi

Bit durum değişikliği hatası referans[36]'de belirtildiği üzere flash hafızasında gerçeklenmiştir. Flash hafızası program kodunun saklandığı yerdir, bu yüzden bit durum değişikliği mikrokontrocünün çalışmasında ciddi sonuçlara neden olabilir.

Bit durum hatası değişiklinin benzetimi için program başında tanımlanan sistem matris parametresinin değeri bit düzeyinde değiştirilmiştir.

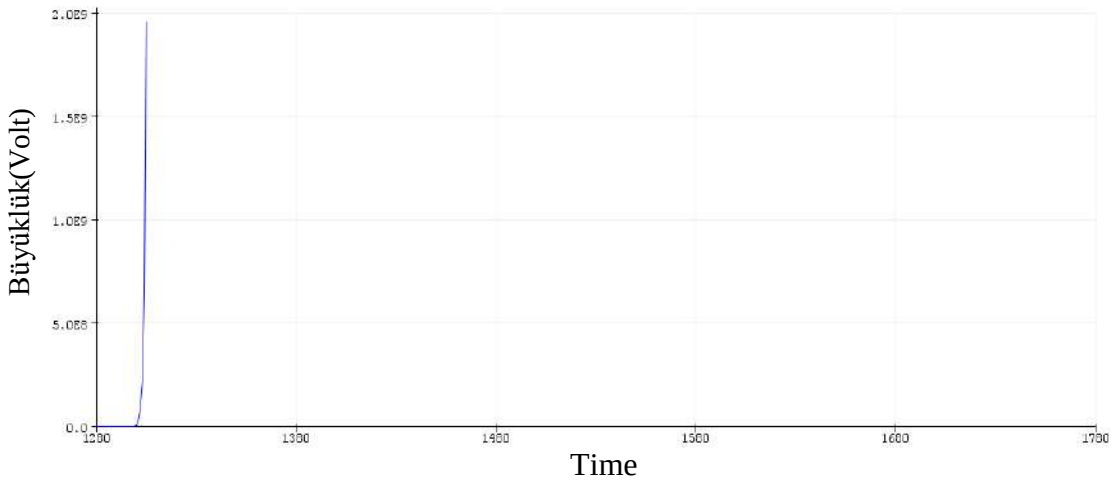
$$X = \begin{bmatrix} 1 & 0 \\ 0 & 0.7969 \end{bmatrix} \dot{X} + \begin{bmatrix} 0.0023 \\ -0.2031 \end{bmatrix} U \quad Y = [4 \;\; 17.8924]X + [27.897]U$$

00000001 (8 bitlik gösterimi) → Bit Değişimi Hatası → 00000011

Bit değişimi hatası $A\ sistem$ matrixindeki 1 değerinin 8 bitlik depolama alanında sondan ikinci bitinin değişerek 3 değerine dönüşümüyle gerçeklenmiştir.

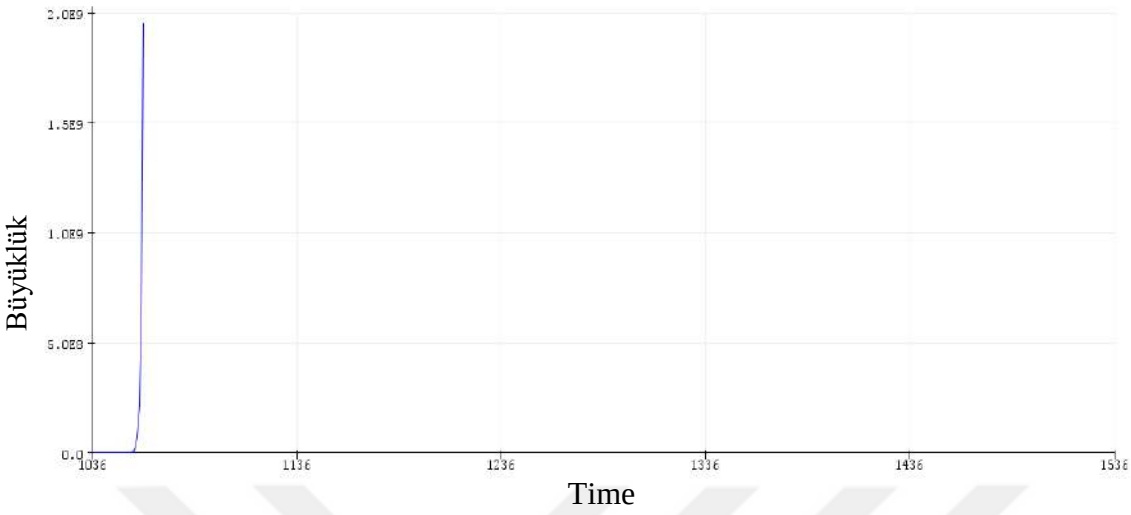
7.4.2 Bit Durum Değişikliği Hatasının Deneysel Sonuçları

Bit değişimi hatasının gerçeklenmesinden sonra yeni A sistem matrixi Atmega328P mikrokontrolcüsüne yüklenmiştir ve yardımcı sinyal çıkışıyla kontrolcünün açık döngü birim basamak cevabı kaydedilmiştir.



Şekil 7.8: Bit durum hatası durumunda kontrolcünün açık döngü birim cevabı

Bit durum hatasından sonra mikrokontrolcüdeki kontrolcünün karakteri tamamıyla değişmiştir. Hatalı kontrolcüyle dinamik sistemin sürülmesi çok risklidir.



Şekil 7.9 : Bit hatası durumunda yardımcı test sinyali çıkışı

Şekil-7.9'de bit hatası durumunda yardımcı sinyalin test sinyal çıkışı gösterilmiştir. Yardımcı sinyal çıkışı izlenerek kontrolcü donanımındaki bit durumu hatası kolayca tespit edilebilir. Bit durum hatası durumunda kontrolcü cevabı diğer hatalardan daha agresif bir çıkışa sahiptir, bu nedenle bit durum hatasının erken tespit edilmesi dinamik sistemin güvenliği için kritik bir öneme sahiptir.
Şekil-7.9'de de görüldüğü gibi aktif hata tespiti bit durum hatasını mikrokontrolcü çalışmaya başlar başlamaz tespit etmiştir.

7.5 Hata Tespit Metodunun Performansının Değerlendirilmesi

Deneysel olarak üç farklı hata durumu Atmega328P mikrokontrolcüsünde gerçeklenmiştir. İlk olarak örnekleme zamanı hatası mikrokontrolcünün ana döngüsüne zaman hatası enjekte edilerek gerçeklenmiştir. Aktif hata tespit etme metodu 0.008 saniye büyüklüğündeki zaman hatasını tespit etmiştir.
Dış etmen kaynaklı hataların mikrokontrolcüde gerçeklenmesi için 0.1 v büyüklüğündeki gürültü sinyali yardımcı test sinyal girişine ve dijital&analog sensor girişine eklenmiştir. Yardımcı test sinyal çıkışı 0.1v büyüklüğündeki dış etmen hatasını tespit etmiştir

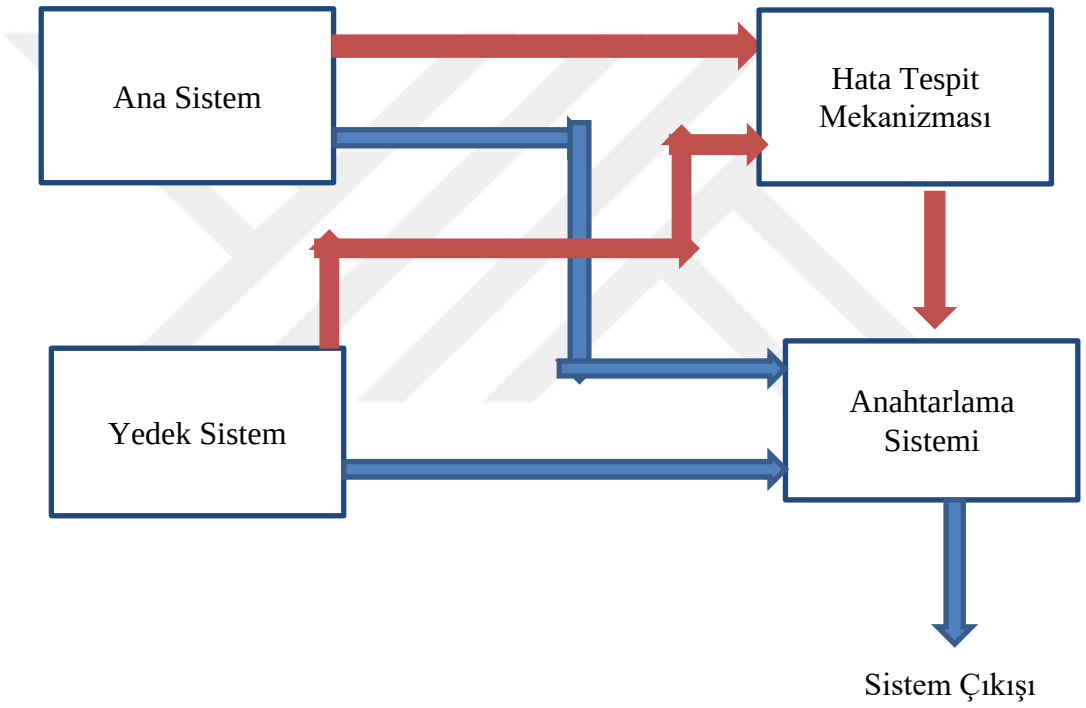
Literatür araştırması yapıldığında yüksek radyasyonun Atmega328P kontrolcünün Flash Hafızasında bit değişim hatasına neden olduğu gözlemlenmiştir. Bit değişim hatası mikrokontrolcüye kontrolcünün sistem matrisi parametrelerinin değiştirilmesiyle gerçeklenmiştir. Hata tespit metodu 1 bitlik durum değişiklik hatasını tespit etmiştir.

Tablo 7.2 : Hata tespit mekanizmasının deneysel ortamda performans değerlendirilmesi.

Hata Durumu	Hata Gerçeklenme Yöntemi	Tespit Edilme Durumu
Örnekleme Zamanı Hatası	Mikrokontrolcünün ana döngüsüne zaman hatası enjekte edilerek	0.008 saniyelik zaman hatası tespit edildi.
Dış etmen hatası	Dijital&Analog sensor değerlerine ve yardımcı test sinyal girişine gürültü sinyali eklenerek	0.1 Volt büyüklükteki dış etmen hatası tespit edildi.
Bit Durumu Değişikliği	Kontrolcünün sistem matrislerin parametre değerleri değiştirerek	$A\ sistem$ matrisindeki 1 bitlik durum değişikliği tespit edildi

8. AKTİF HATA TESPİTİYLE HATA MASKELENMESİ

Aktif hata tespitinin performansı daha önceki başlıkta analiz edilmiştir. Bu başlık altında ise aktif hata tespitiyle hata maskelenmesi üzerinde çalışılmıştır. Hata maskelenmesi için sıcak bekleme dinamik yedekleme dizaynı önerilmiştir.
Sıcak bekleme dinamik yedekleme dizaynında iki tane sistem paralel olarak eş zamanlı çalıştırılır, hata tespit düzeneği kontrolcülerin hata durumunu takip eder ve kontrolcülerden birinde hata olması durumunda anahtarlama mekanizmasıyla dinamik sistemin hatasız çalışan kontrolcüyle sürülmesi sağlar[37].



Şekil 8.1: Sıcak bekleme dinamik yedekleme dizaynı[37]

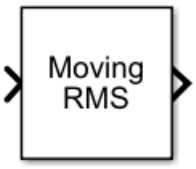
Sıcak bekleme dinamik yedekleme dizaynında hata tespit mekanizması anahtarlama sistemini yönlendirir ve ana sistemde hata meydana gelmesi durumunda yedek sistemi devreye alır.

8.1 Sıcak Bekleme Dinamik Yedekleme Dizaynının Simülasyon Ortamında Gerçeklenmesi

Aktif hata tespitiyle hata maskelemesi yapmak için sıcak bekleme dinamik yedekleme dizaynı simülasyon ortamında gerçeklenmiştir. Öncelikle hata tespit mekanizması tasarlanmıştır.
Daha önceden yardımcı sinyal çıkışı izlenerek hata tespiti yapılmıştır. Hata tespit mekanizmasında yardımcı sinyalin rms değeri kullanılmıştır.

$$Yardımcı_{rms} = \frac{A}{\sqrt{2}} \tag{8.1}$$

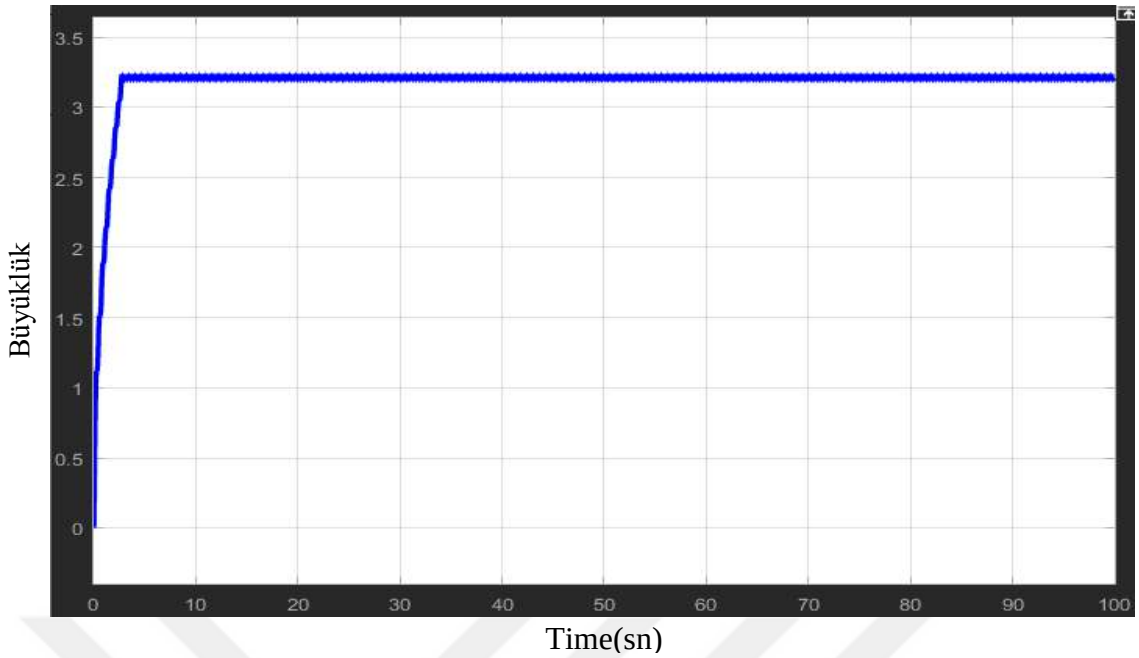
Eşitlik-8.1'de gösterildiği gibi sinus sinyalinin rms değeri sinyal büyüklüğünün $\sqrt{2}$'ye bölümüne eşittir.Rms değeri sayesinde sinus sinyalininin büyüklüğü skalar olarak izlenebilir.



Şekil 8.2: Simulasyon ortamındaki rms bloğu

Rms bloğu kendi içinde 1000 sayılık bir dizin oluşturarak bu dizindeki sayıların karesini alıp topladıktan sonra toplam değeri 1000'ne bölüp karekökü alır.

Yardımcı sinyal çıkışı Şekil-4.12'de gösterilmiştir. Yardımcı sinyal çıkışı rms bloğuna bağlanarak rms cevabı analiz edilmiştir.



Şekil 8.3 :Yardımcı sinyal çıkışının rms cevabı

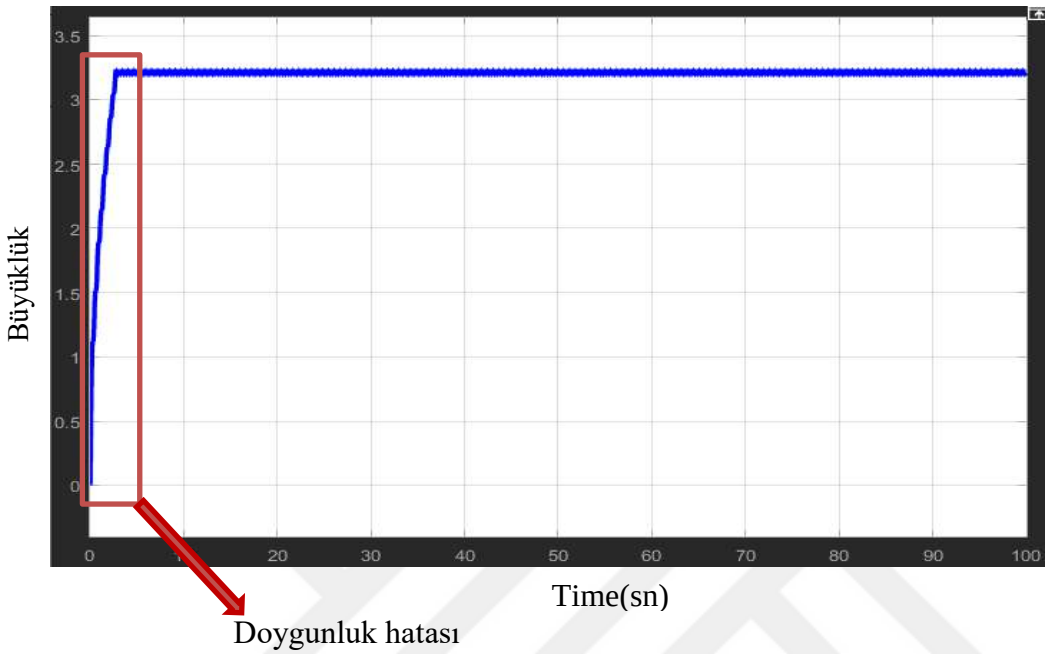
Şekil-8.3'de yardımcı sinyal çıkışının rms değeri gösterilmiştir. Rms bloğu 1000 sayılık bir dizinden oluştuğu için dizinin doygunluğa ulaşması için 2 saniye süre geçmiştir. Rms bloğu doygunluğa ulaştıktan sonra istikrarlı çalışmıştır.
Şekil-8.3'de rms değeri 3.3 olarak ölçülmüştür. Bu değer ölçülen yardımcı sinyal çıkışı büyüklüğüyle uyumludur.

$$Yardımcı\ Sinyal\ Çıkışı = 4.73\sin(10t)$$

$$Yardımcı\ Sinyal\ rms'i = \frac{4.73}{\sqrt{2}} \cong 3.3 \qquad (8.2)$$

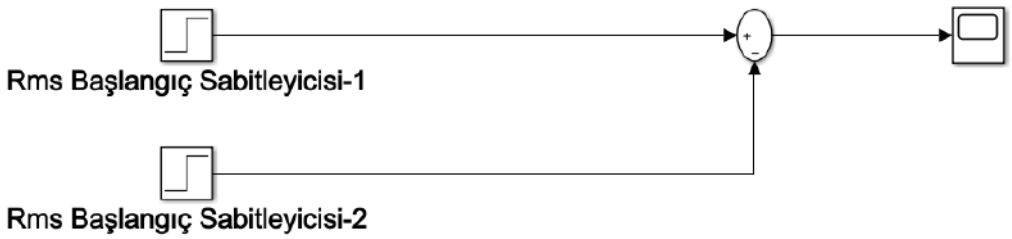
8.1.2 Rms Bloğu Doygunluk Hatası Çözümü

Rms bloğu 1000 sayılık bir dizine sahip olduğu için rms bloğunun yaklaşık 2 saniye kadar doygunluğa ulaşması gerekir.



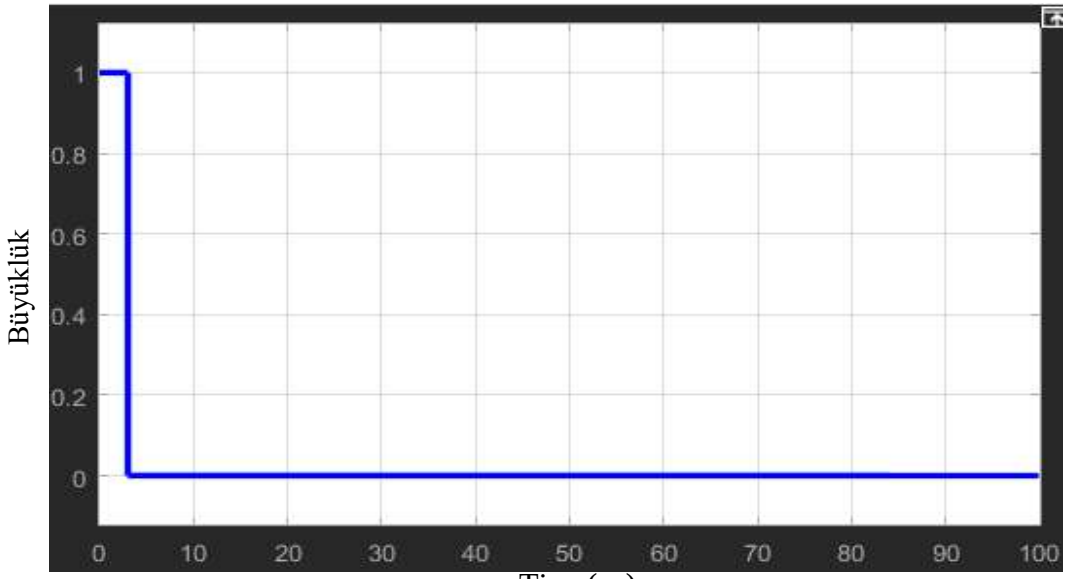
Şekil 8.4: Doygunluk hatası gösterimi

Doygunluk hatasını gidermek için hata tespit mekanizmasına 0.2 saniyelik birim sinyal eklenir, bu sayede hata mekanizmasının başlangıçtan 0.2 saniyelik kısma kadar geçen sürede hata mekanizmasının hata vermesi önlenir.



Şekil 8.5: Rms doygunluk hatası çözücü

Doygunluk hatasını gidermek için iki tane birim basamak fonksiyonu kullanılmıştır. İkinci birim basamak fonksiyonu 2. saniyeden başlayarak çalışmaya başlatılmıştır. Bu sayede süresi 2 saniye olan bir birim fonksiyon elde edilmiştir.



Şekil 8.6: Rms düzenleyicisi cevabı

Şekil-8.6'de rms düzenleyicisi cevabı belirtilmiştir. Rms düzenleyicisi 2 saniyelik birim sinyal oluşturarak hata tespit mekanizmasının rms doygunluğu süresince hata vermesini engeller.

8.1.3 Hata Üst vs Alt Sınırının Rms Değerlerinin Belirlenmesi

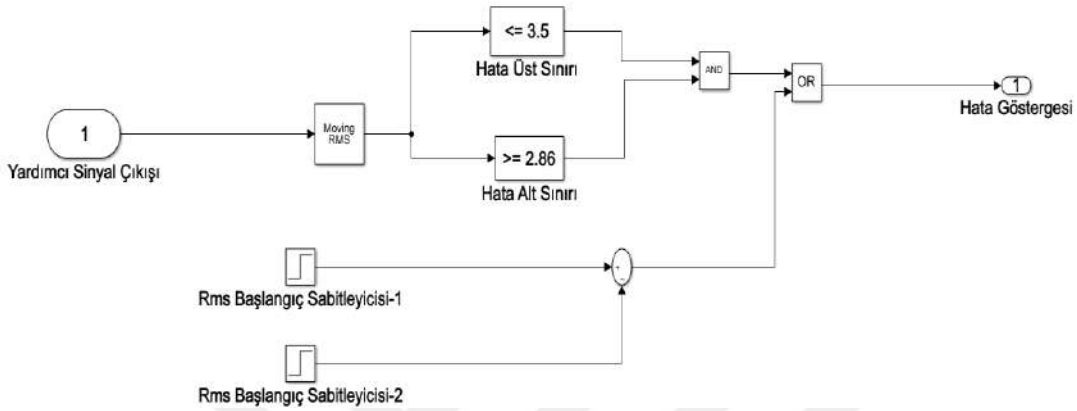
Çalışmadan daha önceden belirtildiği gibi düzgün çalışan bir mikrokontrolcü için çıkış sinyal büyüklüğü 4.05 ile 4.95 arasında olmalı, bu nedenle düzgün çalışma rms değeri de bu değerlerin $\sqrt{2}$'e bölerek elde edilir.

$$\frac{4.05}{\sqrt{2}} < düzgün_{rms} < \frac{4.95}{\sqrt{2}} \rightarrow \quad 2.86 < düzgün_{rms} < 3.5 \qquad (8.3)$$

Rms değerinin 2.86'nın altına düşmesi durumunda ya da 3.5'in üstüne çıkması durumunda hata olduğu çıkarımı yapılır.

8.1.4 Hata Tespit Mekanizmasının Kurulması

Hata tespit mekanizması mikrokontrolcünün yardımcı sinyal çıkışının 3.5'in üstüne ve 2.86'nın altına düşme durumunda 0 çıkışı verecek şekilde dizayn edilmiştir. Ayrıca rms doygunluk hatası da hata tespit mekanizmasında çözülmüştür.



Şekil 8.7 : Hata tespit mekanizması

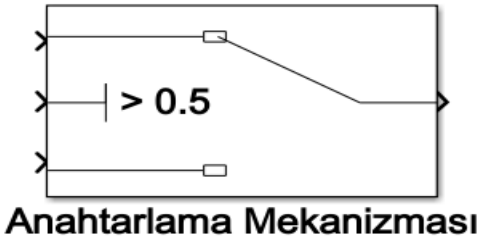
Hata tespit mekanizmasında lojik kapıları karar vermek için kullanılmıştır. Hata üst sınırı ve hata alt sınırı kontrol edilerek rms değerinin bu değerlerin dışında kalması durumunda hata göstergesinin uyarılması sağlanır.

Şekil 8.8 : Hata tespit mekanizması bloğu

Hata tespit mekanizmasının daha sade gösterimi için şekil-8.8'deki hata tespit mekanizması bloğu kullanılmıştır.

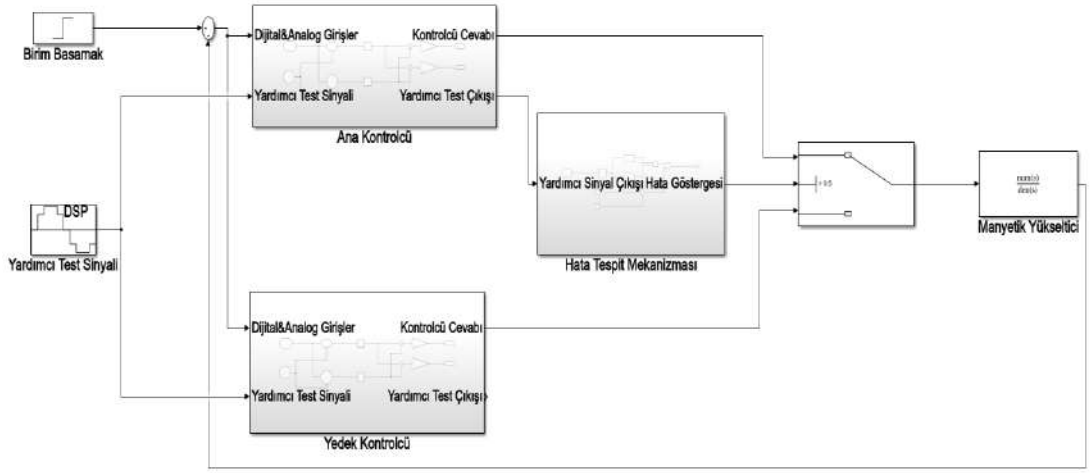
8.1.5 Anahtarlama Mekanizması

Anahtarlama mekanizması ana mikrokontrolcüyle yedekli mikrokontrolcü arasındaki geçişi sağlamak için kullanılmıştır.



Şekil 8.9 : Anahtarlama bloğu

Hata tespit mekanizması çıkışı anahtarlama sisteminin eşik değerine bağlanacaktır. Bu sayede hata tespit mekanizması ana mikrokontrolcüde hata tespit ettiğinde ana mikrokontrolcüyü izole ederek dinamik sistemin yedek kontrolcüyle sürülmesini sağlar.

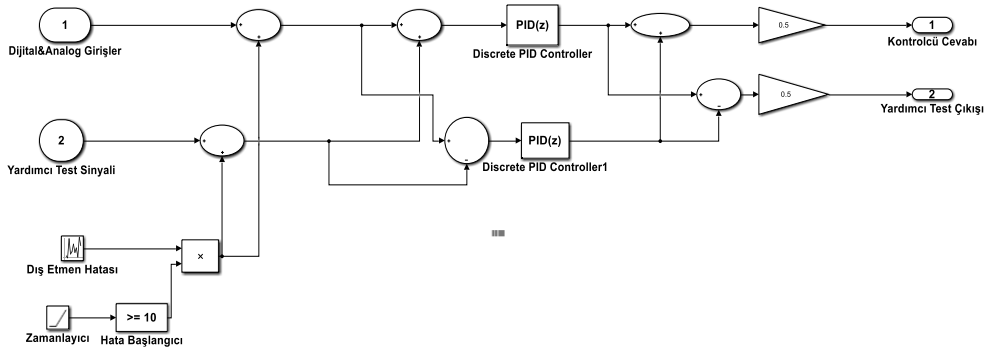


Şekil 8.10 : Sıcak bekleme dinamik yedekleme dizaynı

Hata tespit mekanizması ve anahtarlama sistemi gerçeklendikten sonra Şekil-8.10'daki sıcak bekleme dinamik yedekleme sistemi dizayn edilmiştir. Sistemde yedek kontrolcü çalışır vaziyette beklemektedir ve ana kontrolcüde hata tespit edilmesi durumunda ana kontrolcü sistemden izole edilerek sistemin yedekli kontrolcüyle sürülmesi sağlanır.

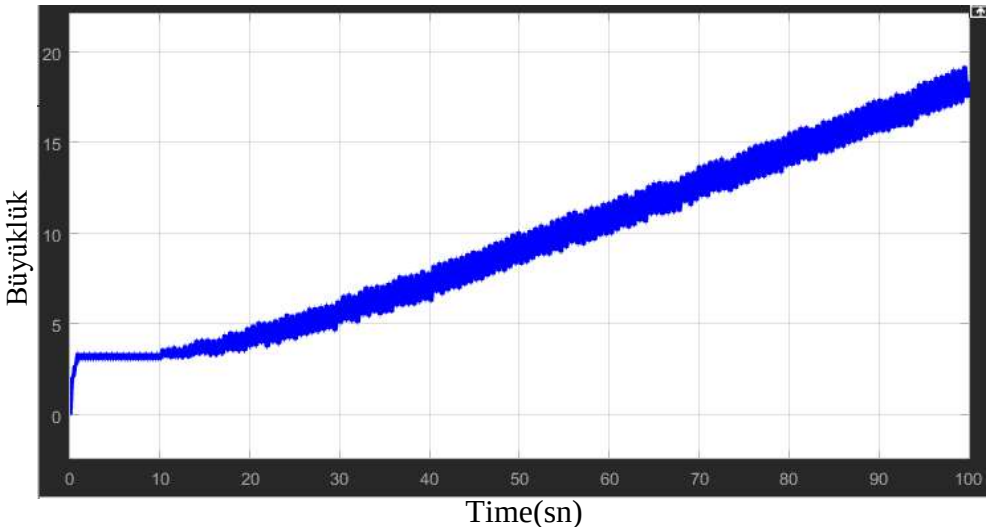
8.2 Sıcak Bekleme Dinamik Yedekleme Tasarımının Simülasyon Sonuçları

Simülasyon ortamında sıcak bekleme dinamik yedekleme dizaynı Şekil-8.10'daki gibi gerçeklenmiştir. Dizayn performansını ölçmek için dış etmen hatası ana kontrolcüye gürültü şeklinde eklenmiştir.



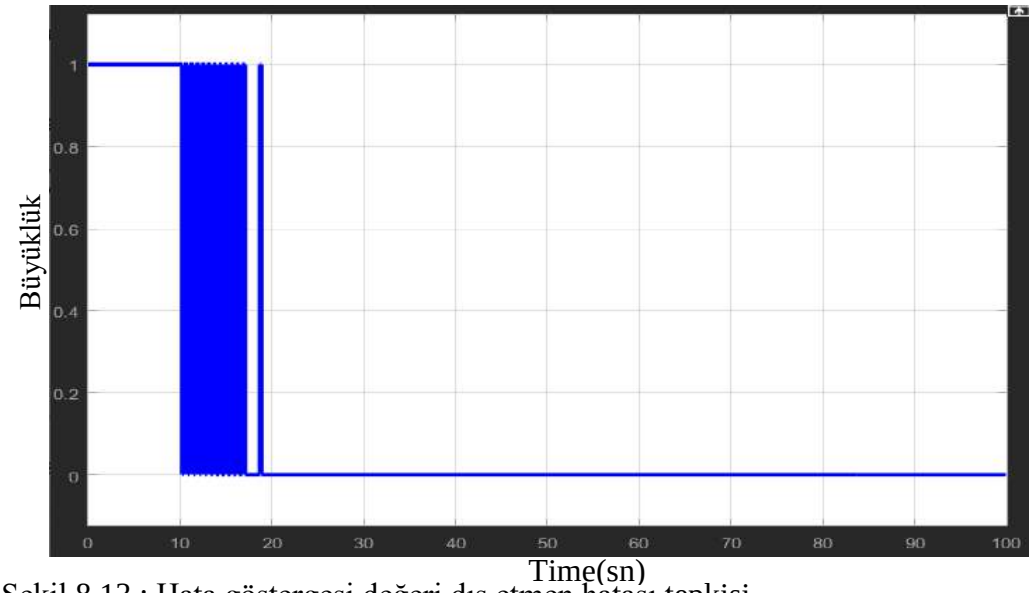
Şekil 8.11: Sıcak bekleme dinamik yedekleme dizaynı dış etmen hatası gösterimi

0.1v büyüklüğündeki dış etmen hatası ana kontrolcüye şekil-8.11'de gösterildiği gibi eklenmiştir. Dış etmen hatası 10. Saniyeden sonra aktif hale gelir. Yardımcı sinyalin rms değerinin, hata göstergesinin ve dinamik sistemin kapalı döngü cevabının dış etmen hatasına karşı tepkileri ölçülmüştür.



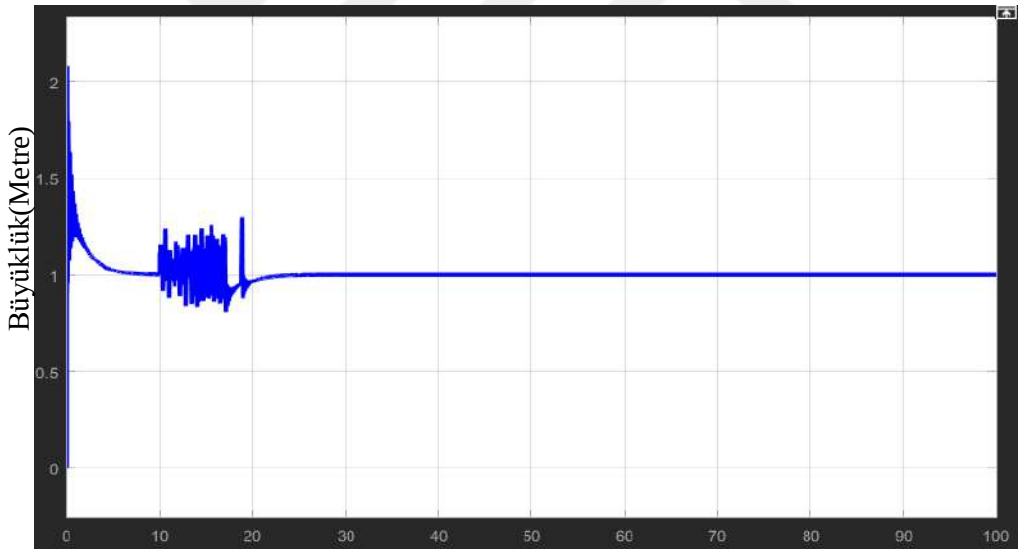
Şekil 8.12: Yardımcı sinyalin rms değeri

Yardımcı Sinyalin rms değeri 10.saniyeden sonra dış etmen hatasına tepki olarak sürekli artmaya başlamıştır.



Şekil 8.13 : Hata göstergesi değeri dış etmen hatası tepkisi

Şekil-8.13'de hata göstergesi değerinin dış etmen hatası tepkisi gösterilmiştir. Hata mekanizmasının hata tespiti yapması yaklaşık 8 saniye sürmektedir. Bu süre kaybının nedeni hata mekanizmasının rms algoritması kullanmasıdır.

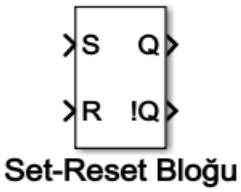


Şekil 8.14 : Dinamik sistem kapalı döngü birim basamak cevabının dış etmen hatası Tepkisi

Şekil-8.14'de görüldüğü gibi 10.saniye ve 20.saniye arasında dış etmen hatası dinamik sistemin kapalı döngü cevabını etkilemektedir. Bu durum nedeni rms algoritmasının 1000 sayılık bir dizin oluşturup bu dizini dinamik olarak yenilemesidir. Bu nedenle dış etmen hatasının rms bloğunda gözlenmesi için belirli bir süre geçmesi gerekir. Dış etmen hatasının dinamik üzerindeki etkisinin ortadan kaldırılması yaklaşık 8 saniye sürmüştür.

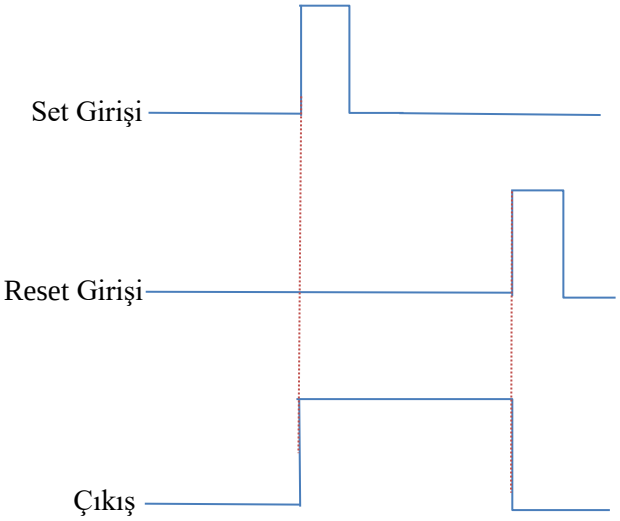
8.3 Set-Reset Bloğunun Hata Tespit Mekanizmasında Kullanılması

Şekil-8.13'deki hata tespit mekanizmasının dış etmen hatasını tespit etmesi yaklaşık 8 saniye sürmüştür. Bu durumun nedeni rms algoritmasının 1000 sayılık bir dizin oluşturması ve dış etmen hatasının bu dizinin üzerindeki etkisinin gözlemlenmesinin zaman almasıdır. Dış etmen hatasının daha hızlı tespit edilmesi için Set-Reset Bloğu önerilmiştir.



Şekil 8.15: Set-Reset bloğu

Set-Reset bloğunun temel çalışma prensibi set değerine herhangi bir zaman içerisinde 1 girişi geldiğinde çıkış sinyali 1 olarak ayarlanır, set değeri 0 olsa bile blok resetlenene kadar çıkış değeri 1 olarak kalır.



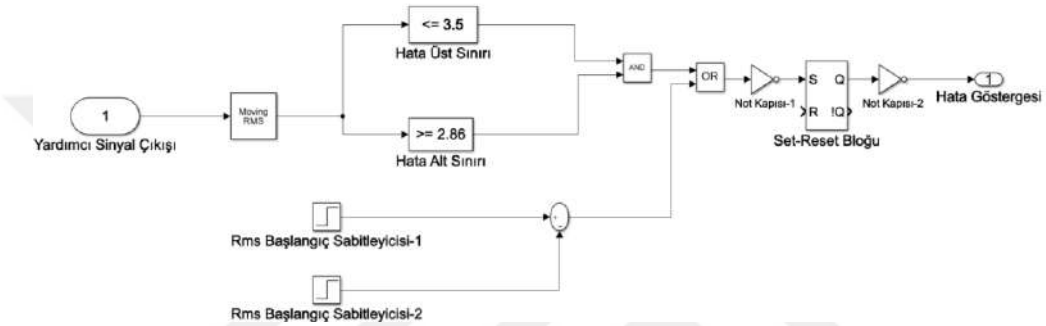
Şekil 8.16: Set-Reset bloğu sinyal diagramı

Set-Reset bloğunda set değeri 1 sinyaliyle tetiklendiği zaman blok çıkış sinyali 1 olarak ayarlanır ve reset sinyali gelene kadar 1 çıkışı verir.

Hata tespit mekanizmasına daha hızlı hata tespiti yapılabilmesi için set-reset bloğu ve not(değil) kapıları hata mekanizmasına eklenmiştir.

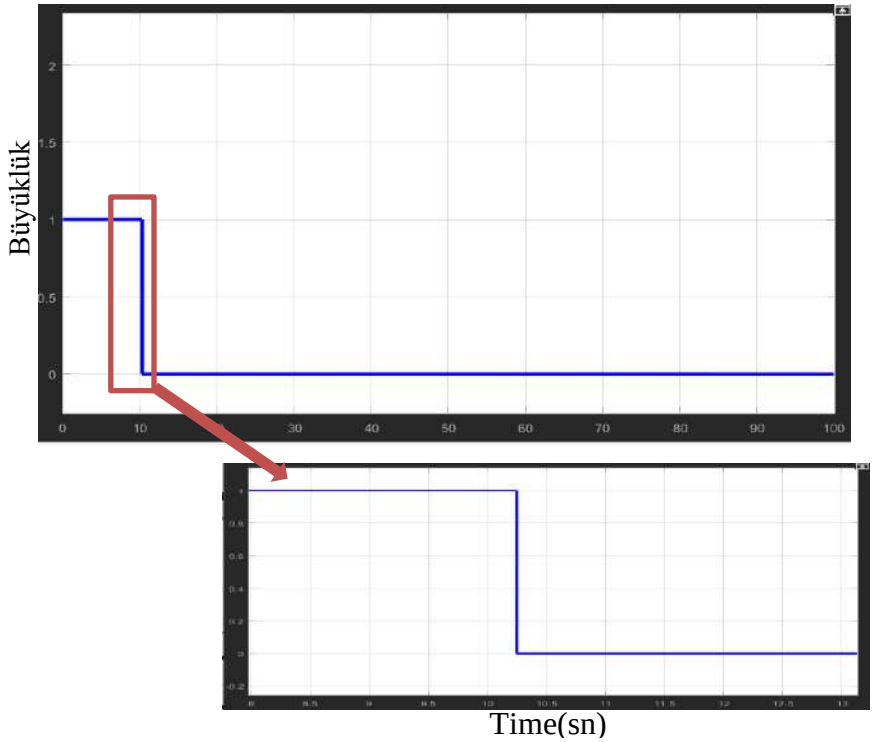
Şekil 8.17: Not(Değil) kapısı bloğu

Hata tespit mekanizması düzgün çalışma durumunda 1 hata durumunda 0 çıkışı verecek şekilde dizayn edilmiştir.



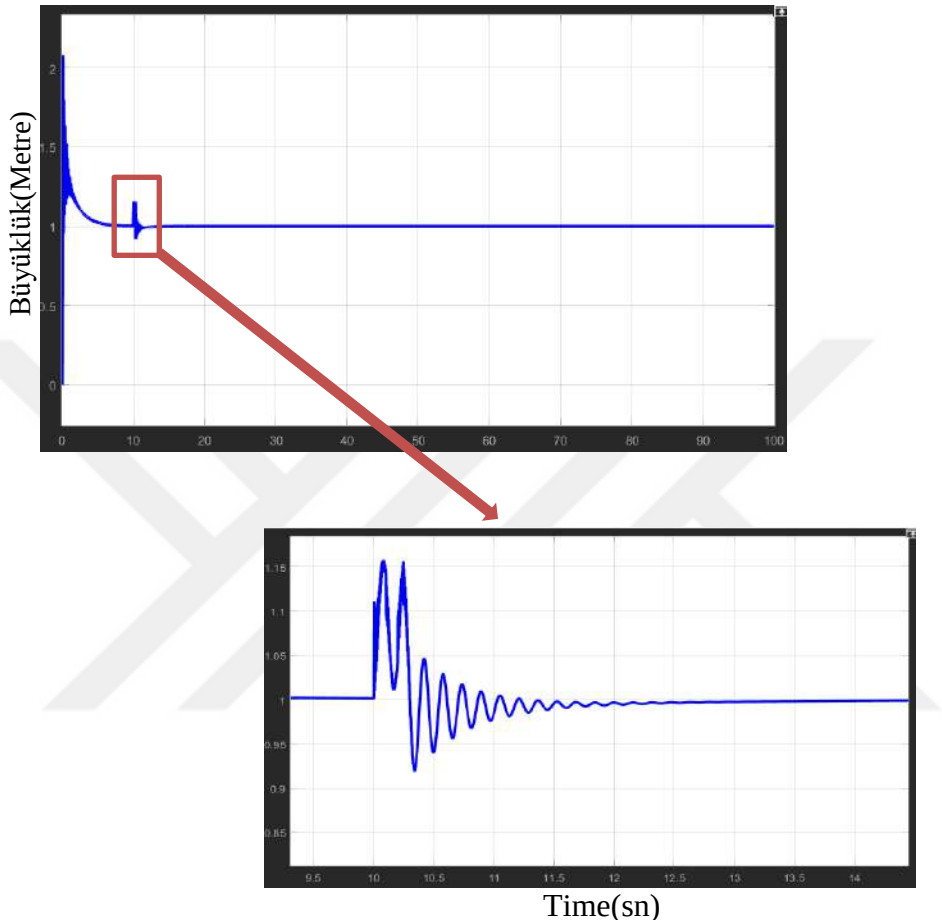
Şekil 8.18 : Hata tespit mekanizması Set-Reset bloğu

Set-reset bloğu sayesinde yardımcı sinyal çıkışının rms değeri belirtilen değerlerin dışına çıkar çıkmaz hata tespiti yapılır.



Şekil 8.19 : Hata tespit sinyalinin detaylı gösterimi dış etmen hatası

Set-reset bloğundan sonra hata tespit süresi şekil-8.19'de görüldüğü gibi yaklaşık 0.24 saniye düşmüştür. Diğer bir değişle hatalı kontrolcüyle yedek kontrolcünün yer değiştirmesi 0.24 saniye sürmüştür, bu nedenle 0.24 saniye boyunca sistem hatalı kontrolcüyle sürülmüştür.



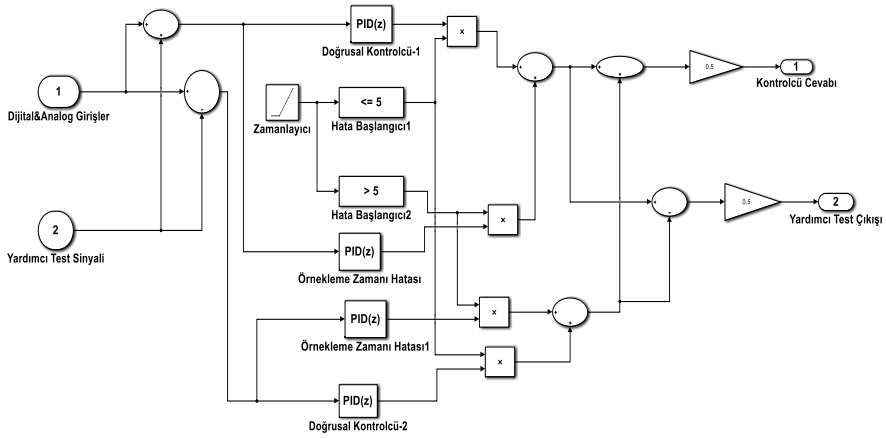
Şekil 8.20 : Dinamik sistem kapalı döngü birim basamak cevabının dış etmen hatasına tepkisi

Şekil-8.20'da dinamik sistemin kapalı döngü birim basamak cevabının dış etmen hatasına verdiği tepki gösterilmiştir. Hatalı kontrolcünün sistem üzerindeki etkisi yaklaşık 2.5 saniye sürmüştür 2.5 saniyeden sonra hatalı kontrolcünün dinamik sistem üzerindeki etkisi giderilmiştir.

Set-Reset bloğuyla yapılan iyileştirme sayesinde hata tespit mekanizmasının hata tespiti yapması 0.24 saniye sürmüştür. Bu nedenle dinamik sistem 0.24 saniye boyunca hatalı sistemle sürülmüştür. Yedekli kontrolcünün hatalı sürülen dinamik sistemi düzeltmesi is 2.5 saniye sürmektedir.

8.4 Sıcak Bekleme Dinamik Yedekleme Tasarımının Örnekleme Zamanı Cevabı

Sıcak bekleme dinamik yedekleme tasarımı dış etmen hatasıyla test edildikten sonra bu başlık altında tasarımın örnekleme zamanı hatasıyla test edilmesi amaçlanmıştır. Örnekleme zamanı hatasının sıcak bekleme dinamik yedekleme tasarımı üzerinde test yapılması için öncelikle test senaryosu hazırlanılmıştır. Test senaryosu kapsamında 5.saniyede ana mikrokontrolcünün içindeki kontrolcünün örnekleme zamanı 0.0028 saniyeden 0.05 saniyeye değiştirilmiştir.

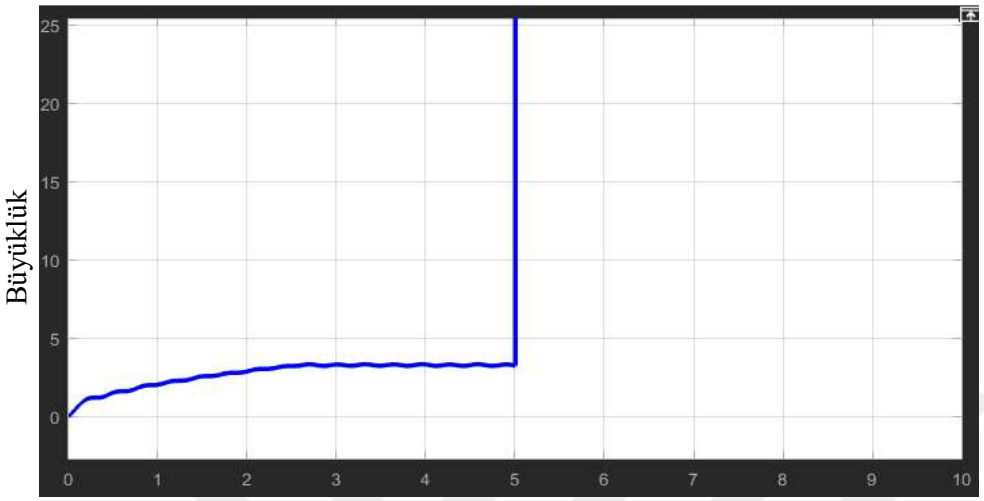


Şekil 8.21 : Örnekleme zamanı hatası senaryosunun simülasyon ortamında gerçeklenmesi

Şekil-8.21'de Örnekleme zamanı hatasının simülasyon ortamında gerçeklenmesi gösterilmiştir. Test kapsamında düzgün çalışan kontrolcü 5.saniyeden sonra örnekleme zamanı hatalı olan kontrolcüyle değiştirilmiştir.

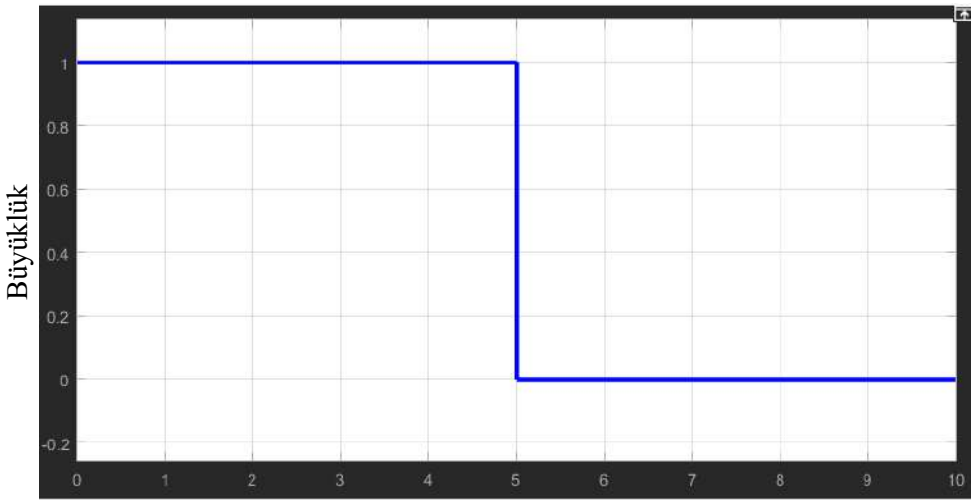
Örnekleme zamanı 0.05 saniye olan bir kontrolcünün dinamik sistemi düzgün süremeyeceği ve sistemin kararsız cevap vereceği çalışma kapsamında daha önceden belirtilmiştir. Bu yüzden hata tespit mekanizmasının örnekleme zamanı hatalı kontrolcüyü tespit etmesi ve ana mikrokontrolcüyü yedek mikrokontrolcüyle değiştirmesi beklenmektedir.

Test kapsamında yardımcı çıkış sinyalin rms değeri, hata tespit sinyali ve dinamik sistem kapalı döngü birim basamak cevabı analiz edilmiştir.



Şekil 8.22: Yardımcı sinyal rms değeri

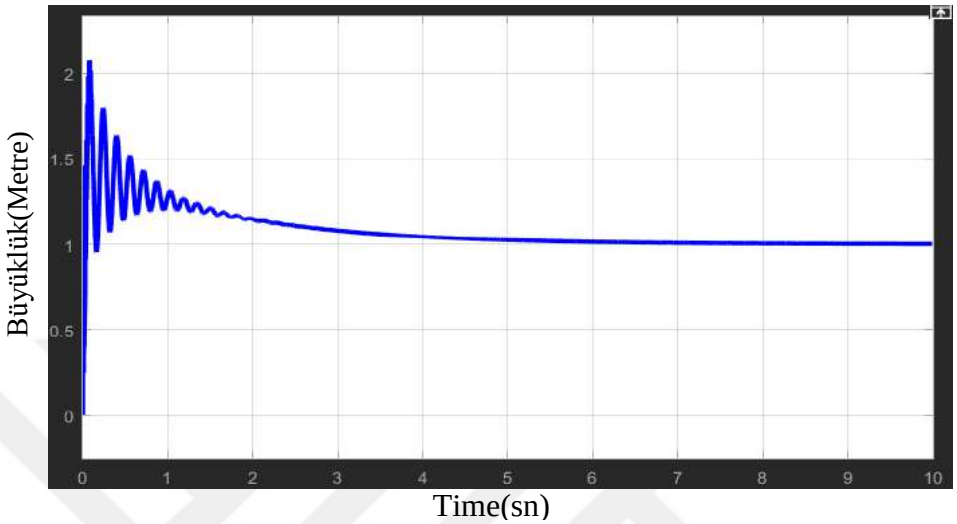
Yardımcı sinyal rms değeri Şekil-8.22'de gösterilmiştir. Simülasyon kapsamında 5.saniyede hata meydana gelmiştir ve bu hata yardımcı sinyalin rms değerine direkt yansımıştır.



Şekil 8.23 : Hata tespit sinyali örnekleme zamanı hatası cevabı

Hata tespit mekanizması örnekleme zamanı hatasını hata olur olmaz 5.saniyede tespit etmiştir. Bu sayede hatalı ana kontrolcüyle yedek kontrolcüyü hata olur olmaz değiştirilebilir. Ayrıca hata tespit mekanizmasının dış etmen hatasını tespit etmesi

yaklaşık 2.5 saniye sürerken zaman döngüsü hatası direkt tespit edilmiştir. Bu durumun nedeni örnekleme zamanı hatasının çok şiddetli olmasıdır. Hata çok şiddetli olduğu için yardımcı sinyalin rms değerine direkt etki etmiştir bu yüzden süre kaybı olmadan hata tespiti yapılmıştır.



Şekil 8.24: Dinamik sistem kapalı döngü birim basamak cevabının örnekleme hatası cevabı

Hata mekanizması örnekleme zamanı hatasını zaman kaybetmeden tespit ettiği için şekil-8.24'de görüldüğü gibi dinamik sistemin kapalı döngü birim basamak cevabı zaman döngüsü hatasından etkilenmemiştir.

8.5 Sıcak Bekleme Dinamik Yedekleme Tasarımının Bit Değişim Hatası Cevabı

Sıcak bekleme dinamik yedekleme tasarımı dış etmen hatası ve örnekleme zamanı hatasıyla test edildikten sonra bit değişimi hatasıyla test edilmiştir. Bit değişimi hatasının simülasyonda gerçeklenmesi için bit değişimi hatasıyla çalışan hatalı kontrolcü oluşturulmuştur. Bit değişimi hatasının doğrusal kontrolcünün durum-uzay gösterimindeki A matrisinde meydana geldiği varsayılmıştır.

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 0.7969 \end{bmatrix} \dot{X} + \begin{bmatrix} 0.0023 \\ -0.2031 \end{bmatrix} U \quad Y = [4 \;\; 17.8924]X + [27.897]U$$

00000000
(8 bitlik gösterimi)

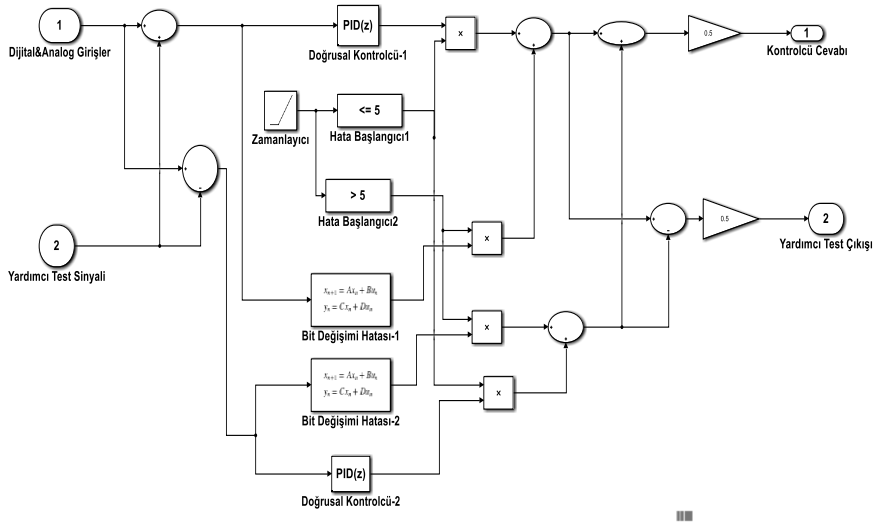
00000001

Bit Değişimi Hatası

Hatalı Kontrolcü

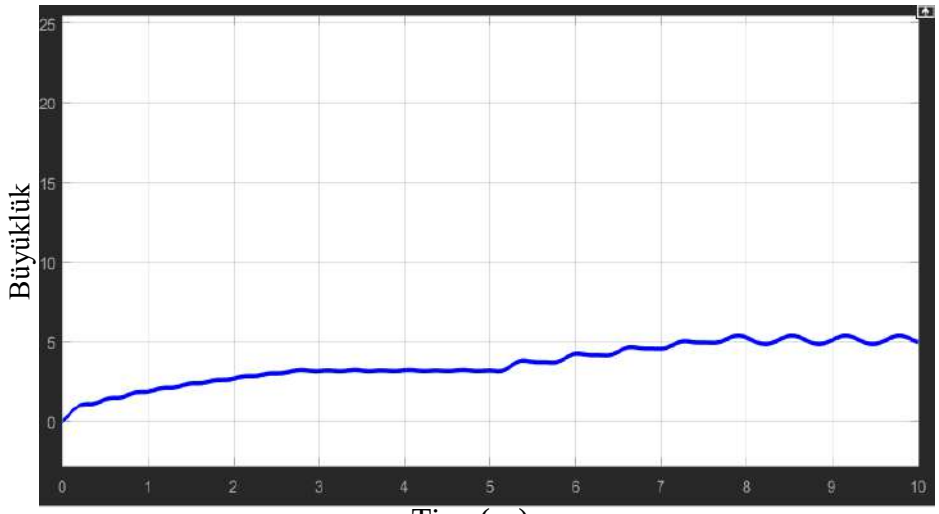
$$X = \begin{bmatrix} 1 & 1 \\ 0 & 0.7969 \end{bmatrix} \dot{X} + \begin{bmatrix} 0.0023 \\ -0.2031 \end{bmatrix} U \quad Y = [4 \;\; 17.8924]X + [27.897]U$$

Test senaryosu kapsamında 5.saniyede düzgün çalışan ana kontrolcü hatalı kontrolcüyle değiştirilmiştir. Yardımcı çıkış sinyalin rms değeri, hata tespit sinyali ve dinamik sistemin kapalı döngü birim basamak cevabı incelenerek hata maskeleme mekanizmasının performansı ölçülmüştür.



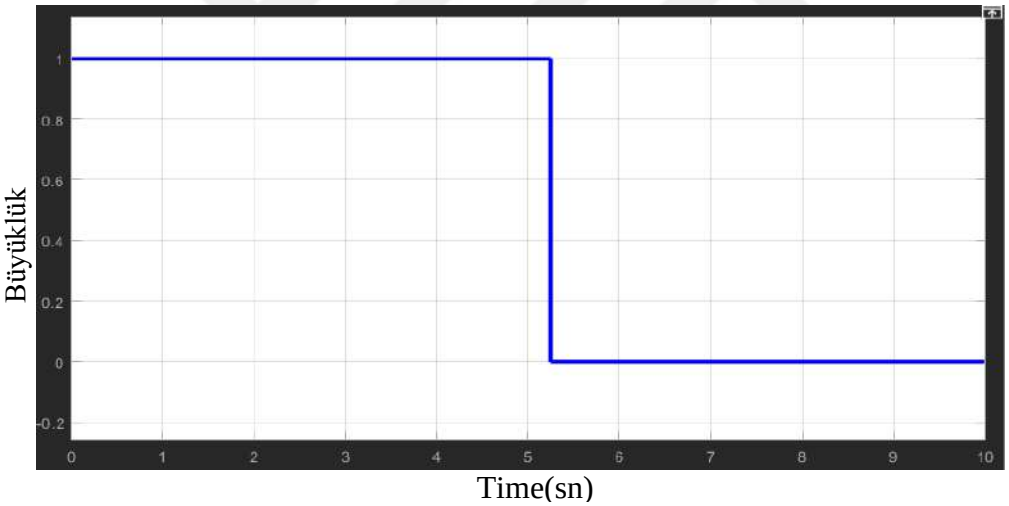
Şekil 8.25: Bit değişimi hatasının simülasyon ortamında gerçeklenmesi

Şekil-8.25'de bit değişimi hatasının simülasyon ortamında gerçeklenmesi gösterilmiştir. Bit değişimi hatası bloğuna hatalı kontrolcü parametreleri gömülmüştür.



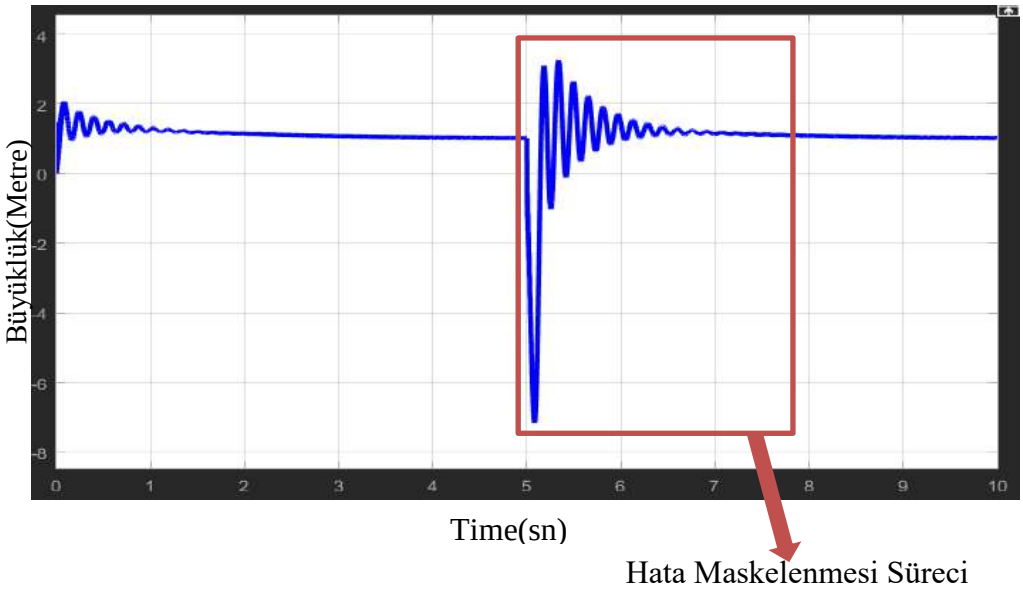
Şekil 8.26 : Yardımcı sinyalin rms değeri

Şekil-8.26'da görüldüğü gibi bit değişimi hatası 5.saniyeden sonra yardımcı sinyalin rms değerini etkilemeye başlamıştır.



Şekil 8.27 : Hata tespit sinyali bit değişimi hatası

Şekil-8.27'de hata tespit sinyalinin bit değişimi hatası cevabı gösterilmiştir. Hata tespit mekanizmasının hata tespiti yapması yaklaşık 0.2 saniye sürmektedir. Bu durumun nedeni rms bloğunun 1000 sayılık bir dizin oluşturmasıdır. Bit değişimi hatasının rms değeri üzerindeki etkisinin gözlenmesi zaman almıştır.



Şekil 8.28: Dinamik sistem kapalı döngü birim basamak cevabı bit hatası tepkisi

Şekil-8.28'de dinamik sistemin kapalı döngü birim basamak cevabı gösterilmiştir. 5.saniyede gerçekleşen bit değişimi hatasının dinamik sistem üzerinde maskelenmesi yaklaşık 3 saniye sürmüştür. Bu sürenin temel nedeni hata mekanizmasının hatayı 0.2 saniyede tespit etmesidir. Bu nedenle dinamik sistem 0.2 saniye hatalı kontrolcüyle sürülür. Yedek kontrolcünün dinamik sistemi düzeltmesi 3 saniye sürmüştür.

8.6 Hata Maskeleme Sonuçlarının Değerlendirilmesi

Çalışma kapsamında hata maskelenmesi için hata mekanizması önerilmiştir. Hata mekanizmasında öncelikle yardımcı çıkış sinyalinin rms değeri alınarak rms değerinin belirlenen hata sınırları içerisinde olup olmadığı kontrol edilmiştir. Set-Reset bloğuyla rms değeri hata sınırları dışına çıkar çıkmaz hata tespiti yapılmıştır.

Hata maskelenmesi için sıcak bekleme dinamik yedekleme sistemi önerilmiştir. Bu tasarımda ana kontrolcü ve yedek kontrolcü eş zamanlı olarak çalıştırılmıştır. Basit anahtarlama mekanizmasıyla hata tespit sinyali gözetilmiştir, hata tespiti yapılması durumunda anahtarlama mekanizması ana kontrolcüyü izole ederek sistemin yedek kontrolcüyle sürülmesini sağlamıştır.

Simülasyon ortamında hata senaryosu oluşturularak hata maskelenmesinin dış etmen hatasına , örnekleme zamanı hatasına ve bit değişimi hatasına tepkisi ölçülmüştür. Test senaryosu kapsamında 5.saniyede düzgün çalışan kontrolcü hatalı kontrolcüyle değiştirilmiştir. Yardımcı sinyalin rms değeri, hata tespit sinyali ve dinamik sistemin kapalı döngü birim basamak cevabı analiz edilmiştir.

Simülasyon sonucunda hata tespit mekanizmasının örnekleme zamanı hatasını dinamik sistemi etkilemeden maskelediği ama dış etmen hatası ve bit değişimi hatasının maskelenmesinin zaman aldığı görülmüştür. Bu durumun nedeninin rms algoritması olduğu belirtilmiştir. Çünkü rms algoritması 1000 sayılık bir dizin oluşturduğu için dış etmen hatasının ve bit değişimi hatasının rms sinyalinde gözlenmesi zaman almaktadır. Bu nedenle hata tespit mekanizmasının dış etmen hatasını ve bit değişimi hatasını tespit etmesi zaman almaktadır ve bu süre boyunca dinamik sistem hatalı kontrolcüyle sürülmektedir.

Tablo 8.1 Hata maskeleme sonuçlarının detaylı gösterimi.

Hata Durumu	Hata Tespit Süresi (Saniye)	Dinamik Sistem Üzerinde Hata Maskelenme Süresi (Saniye)
Dış Etmen Hatası	0.24	2.5
Bit Değişimi Hatası	0.2	3
Örnekleme Zamanı Hatası	0	0

9. AKTİF HATA TESPİTİ METODUYLA ÜÇLÜ MODÜLER YEDEKLİ KONTROLCÜ TASARIMININ KARŞILAŞTIRILMASI

Çalışma kapsamında aktif hata tespiti yapısı ve aktif hata tespiti yapısında hata maskelenmesi detaylı incelenmiştir. Ayrıca üçlü modüler yedekli kontrolcü tasarımı da literatür araştırmasında detaylı olarak açıklanmıştır. Bu başlıkta aktif hata tespiti ile üçlü modüler yedekli kontrolcü tasarımı karşılaştırılmıştır. Karşılaştırma kriterleri olarak güvenirlik, uygulanabilirlik, maliyet, güç tüketimi ve hata maskelenmesi seçilmiştir.

9.1 Maliyet Karşılaştırılması

Üçlü modüler yedekli kontrolcü tasarımının zayıf noktalarından biri de getirdiği yüksek maliyettir. Tasarım hata tespiti ve maskelemesi yapmak için üç tane özdeş mikrokontrolcü kullanır bu yüzden tasarım maliyeti üç katına çıkar. Aktif hata tespitinde ise tek mikrokontrolcüde hata tespiti yapılabilmektedir. Aktif hata tespiti metodunda hata maskelenmesi için ana mikrokontrolcü ve yedek mikrokontrolcü olmak üzere iki mikrokontrolcü yeterlidir. Sonuç olarak aktif hata tespiti metodu üçlü modüler kontrolcüden daha az maliyetle hata tespiti ve hata yedeklemesi yapabilmektedir.

9.2 Güç Tüketimi Karşılaştırılması

Üçlü modüler yedekli kontrolcü tasarımı üç tane mikrokontrolcü kullandığı için hata maskelemesi için toplam güç tüketimini üç katına çıkarmıştır. Buna karşılık aktif hata tespitinde tek mikrokontrolcü ve sinüs sinyal kaynağı kullanılmıştır. Sinüs kaynağının mikrokontrolcü dışında bir elektronik yapı ile gerçeklenmesi güç tüketimini artırabilir. Bu nedenle güç tüketimi karşılaştırılmasında deneysel bir çalışma yapılmadan karar verilmesi uygun değildir.

9.3 Uygulanabilirlik Karşılaştırılması

Üçlü modüler yedekli kontrolcü tasarımı hali hazırda kullanılan bir tasarımdır. Ama tasarımın en büyük zorluğu senkronizasyon ayarlanmasıdır. Diğer bir değişle üç özdeş kontrolcüdeki bütün elektronik bileşenlerin birebir aynı olması gerekir. Üç kontrolcünün tam olarak aynı anda çalışmaya başlatılması gerekir. Örneğin güç devresindeki kapasitör farklılığından dolayı bir kontrolcünün çalışmaya önce başlaması üçlü modüler yedekli kontrolcü tasarımına zarar verecektir.

Aktif hata tespitinin en büyük avantajlarından birisi kolay uygulanabilmesidir. Çalışma kapsamında Atmega328P mikrokontrolcüsünde hata tespitinin pratik uygulaması yapılmıştır.

9.4 Güvenirlik Karşılaştırılması

Üçlü modüler yedekli kontrolcü pasif hata tespiti metotlarından olan karşılaştırma yöntemini kullanır. Bu nedenle eş zamanlı çalışan özdeş üç kontrolcüden ikisinin hatalı çalışması durumunda dinamik sistem hatalı kontrolcü cevabıyla sürülür. Eşitlik-1'de üçlü modüler kontrolcünün güvenirlik yüzdesi eşitliği açıklanmıştır.

Aktif hata tespitinde ise hata tespiti kontrolcülerden bağımsız somut kriterler göre yapılır. Bu nedenle hata maskelemesinde kullanılan iki kontrolcünün güvenirlik yüzdesi birbirinden bağımsızdır.

$$R_1 = bir\ kontrolc\text{ünün}\ g\ddot{u}venirlik\ y\text{üzdesi}$$

$$R_{Aktif} = iki\ kontrolc\text{üden}\ olu\text{şan}\ hata\ maskeleme\ sistemi$$

$$R_{Aktif} = 1 - \big((1 - R_1) \times (1 - R_1)\big) \rightarrow \ 2R_1 - R_1^2 \qquad (9.1)$$

Eşitlik-9.1'de çalışma kapsamında incelenen hata maskeleme sisteminin güvenirlik yüzdesi açıklanmıştır. Eşitlik-1.1'de ise üçlü modüler yedekleme sisteminin güvenirlik yüzdesi açıklanmıştır. Bu eşitliklere dayanarak %90 güvenirlikle çalışan kontrolcüyle oluşturulan üçlü modüler hata tespit mekanizması %97.2 güvenirliğe sahipken , aynı güvenirlikle çalışan aktif hata maskeleme sistemi %99 güvenirliğe sahiptir.

9.5 Hata Maskelemesi Karşılaştırılması

Üçlü modüler yedekli kontrolcü tasarımının en önemli avantajlarından biri hata maskelemesidir. Tasarım basit lojik kapıları ile sistem performansını etkilemen hata maskelemesi yapabilmektedir.

Bu çalışmada aktif hata tespiti tasarımında hata maskelemesi yapılabilmesi için rms metodu önerilmiştir ama rms metodu kümülatif bir algoritma olduğu için bazı hata senaryolarında hata maskelemesi biraz zaman almıştır. Gelecek çalışmalarda aktif hata tespiti için hata maskelemesi üzerinde yoğunlaşabilir.

10. GÜVENLİK BÜTÜNLÜK TESTLERİ

Güvenlik bütünlük seviyesi(SIL), güvenlik sistemlerinin performansının somut olarak ölçüldüğü değerlendirme sistemidir[38]. Sürekli çalışan bir güvenlik sistemi için saat başına düşen hata olasılığı(PFH) değerlendirme birimi olarak tanımlanır[38].

Tablo 10.1 IEC61508 Standartlarına göre Güvenlik Bütünü Seviyesi Derelecelendirilmesi[38]

Güvenlik Bütünü Seviyesi(SIL)	Saat Başına Düşen Hata Olasılığı(PFH)
SIL 4	$10^{-9} - 10^{-8}$
SIL 3	$10^{-8} - 10^{-7}$
SIL 2	$10^{-7} - 10^{-6}$
SIL 1	$10^{-6} - 10^{-5}$

Tablo 10.1'de PFH değerine göre yapılan güvenlik bütünü seviyesi(SIL) derecelendirilmesi gösterilmiştir. Sistemin PFH değerini belirlemek için Petri Ağı, Markov Modeli, Monte Carlo Simülasyonu, güvenirlik blok diyagramı gibi yöntemler kullanılmıştır[39].

Tez çalışması kapsamında belirtilen belirsizliklerden dolayı belirli bir matematiksel model oluşturmak için Monte Carlo metodu kullanılmıştır. Monte Carlo metodu içerdiği rastgele süreçler sayesinde belirsizlik içeren sistemlerin SIL testlerinde yaygın olarak kullanılır[39].

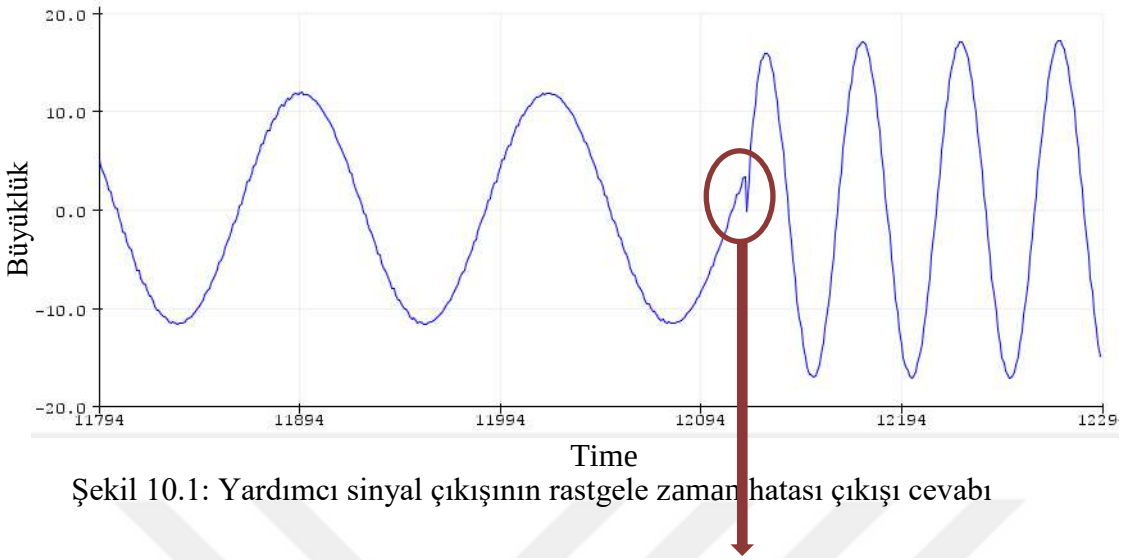
10.1 Monte Carlo Yapısının Kurulması

Tez kapsamında zaman hatası, dış etmen hatası, bit durumu hatası durumları incelenmiştir. Çalışmada sadece zaman hatası üzerinde Monte Carlo simülasyonu yapılmış, üç hatayı kapsayan bir simülasyon yapılması gelecek çalışmalara bırakılmıştır.

Tablo 7.1'de zaman hatası durumunda yardımcı sinyalin büyüklük değerleri gösterilmiştir. Zaman hatası Atmaga328P mikrokontrolcüsüne Pseudo-Random üretim metodunu kullanarak rastgele enjekte edilmiştir. Rastgele seçilen zaman hatasının üst sınırı 30 milisaniye olarak belirlenmiştir.

randomhata=random(0,30);// Pseudo-Random metoduyla rastgele sinyal üretimi.

Kontrolcüye enjekte edilen zaman hatası Pseudo-Random yöntemiyle elde edildikten sonra her 10 saniyede enjekte edilen hata yenileniler ve yardımcı sinyal çıkışı izlenir.

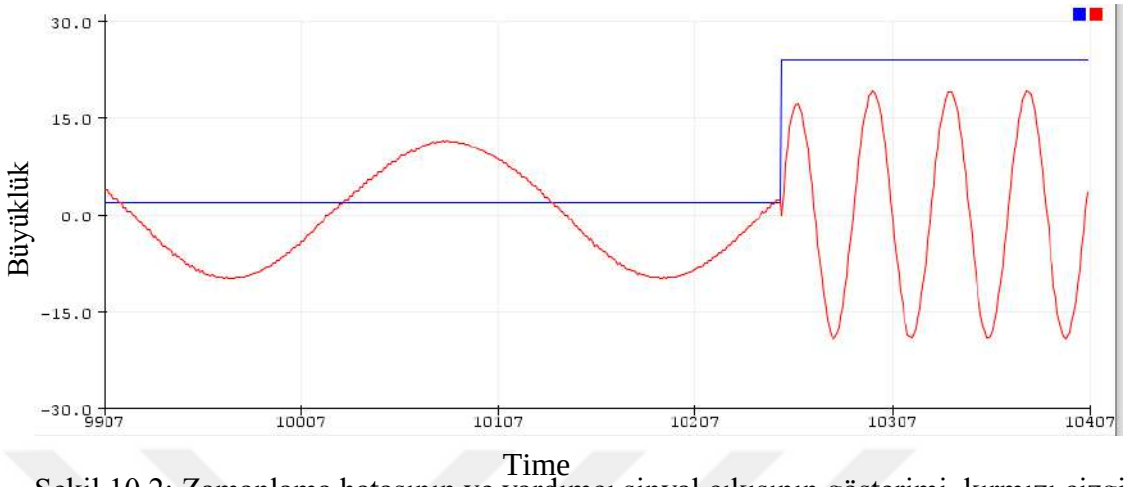


Şekil 10.1: Yardımcı sinyal çıkışının rastgele zaman hatası çıkışı cevabı

Tablo 7.1' de ikizlenmiş kontrolcü tasarımının 8 milisaniyeden sonraki hataları tespit ettiği gösterilmiştir. Bu nedenle enjekte edilen zaman hatası 8 milisaniyeden büyük eşit olması durumunda hata tespitinin yapılması gerekir. Zaman hatasının 8 milisaniyeden büyük olduğu ve hata tespitinin yapılmadığı durumda sistemin hatalı çalıştığı çıkarımı yapılır.

```
if(randomhata>=8 && (yardımcı_çıkış >12.65))
 // Zaman hatasının 8 milisaniyeden büyük olduğu ve hata tespitinin yapıldığı
durumda sistemin doğru çalıştığı çıkarımı yapılır
{
  sistem_durumu=true
  }
if(randomhata>=8 && (yardımcı_çıkış <12.65))
Zaman hatasının 8 milisaniyeden büyük olduğu ve hata tespitinin yapılmadığı
durumda sistemin yanlış çalıştığı çıkarımı yapılır
{
  sistem_durumu=false
  }
```

Rastgele seçilen ve 10 saniyede bir yenilenen zamanlama hatasının büyüklüğü ve yardımcı sinyal çıkışı bilgisayar ortamında kaydedilir.



Şekil 10.2: Zamanlama hatasının ve yardımcı sinyal çıkışının gösterimi, kırmızı çizgi yardımcı sinyal çıkışını mavi çizgi zamanlama hatasını göstermektedir

10.2 Monte Carlo Simülasyon Sonuçları

Monte Carlo simülasyonunda zaman hatası 0-30 milisaniye aralığında Pseudo-Random yöntemiyle oluşturulmuştur ve her 10 saniyede güncellenmiştir. Simülasyon süresince yardımcı sinyal çıkışı ve zamanlama hatası verileri kaydedilmiştir. Simülasyon 15 saat çalıştırılmış ve simülasyon boyunca zaman hatası 5400 defa güncellenmiştir. 15 saatlik simülasyon boyunca ikizlenmiş kontrolcü tasarımının çalışmasında hata gözlemlenmemiştir. Ama simülasyon süresi SIL-4 seviyesini sağlamak için yeterli değildir. SIL-4 seviyesinin elde edilmesi için simülasyonun 10000 saat çalıştırılması gerekir. İkizlenmiş kontrolcü tasarımının SIL-4 sertifikasyonun alması gelecek çalışmalara bırakılmıştır.

11. SONUÇLAR VE GELECEK ÇALIŞMA ÖNERİLERİ

Tez çalışması kapsamında dinamik sistemlerin üzerinde yaygın olarak kullanılan aktif hata tespiti metodu doğrusal kontrolcü donanımları üzerinde uygulanmıştır. Doğrusal kontrolcü donanımları üzerinde aktif hata tespitinin yapılmasındaki en büyük zorluk hata tespit sürecinin kontrolcü cevabını etkilemeyecek şekilde yapılmasıdır. Çünkü kontrolcü cevabı üzerindeki oluşacak en ufak bir değişiklik dinamik sistemin yanlış sürülmesine ve iş güvenliği risklerine yol açar. Bu nedenle aktif hata tespitinin kontrolcü performansını etkilemeyecek şekilde yapılmasını sağlamak için ikizlenmiş kontrolcü tasarımı önerilmiştir ve yardımcı sinyal olarak sinüs sinyali seçilmiştir.. İkizlenmiş kontrolcü tasarımda kontrolcü donanımı içerisindeki doğrusal kontrolcü ikizlenir. İlk kontrolcü pozitif sinüs sinyalini , ikinci kontrolcü negatif sinüs sinyalini kullanarak hata tespiti yapar. İkiz kontrolcülerin çıkışları toplanarak kontrolcü cevabı ve ikiz kontrolcüler birbirinden çıkarılarak yardımcı sinyal çıkışı elde edilir. Yardımcı sinyal çıkışı doğrusal kontrolcünün yardımcı sinyale olan tepkisine eşittir. İkizlenmiş kontrolcü tasarımı açıklandıktan sonra yardımcı sinüs sinyalinin seçilme kriterleri belirtilmiş. Yardımcı sinüs sinyali için belirtilen frekansta doğrusal kontrolcünün frekans cevabı gözlenebilir olmalıdır. Yardımcı sinüs sinyalinin büyüklüğü ise kontrolcü donanımının dijital&analog giriş-çıkış limitlerine göre belirlenmiştir. Örneğin giriş-çıkış limiti 5v olan bir kontrolcü donanımı için yardımcı sinyal ve yardımcı çıkış sinyali büyüklüğü 5v'dan daha küçük olmalıdır. Yardımcı sinyalin frekans ve büyüklük kriterleri belirlendikten sonra aktif hata tespitinin simülasyon ortamında gerçeklenmesi yapılmıştır. Simülasyon ortamında manyetik yükseltici için tasarlanmış doğrusal kontrolcü üzerinde aktif hata tespiti yapılmıştır. Öncelikle simülasyon ortamında aktif hata tespitinin kontrolcü cevabını etkilemediği gözlemlenmiştir. Daha sonra dış etmen, örnekleme zamanı ve bit değişimi hataları kontrolcüye eklenerek hata tespitinin performansı ölçülmüştür ve hata tespiti üç hatayı da tespit etmiştir. Simülasyon ortamından sonra aktif hata tespiti deneysel ortamda gerçeklenmiştir. Aktif hata tespiti Atmega328P kontrolcüsüne gömülmüştür ve gömülü sistemin dış etmen, örnekleme ve bit değişikliği hatalarına karşı tepkileri ölçülmüştür. Deneysel testlerin sonucunda gömülü sistemin dış etmen, örnekleme zamanı ve bit değişimi hatalarını tespit ettiği görülmüştür. Aktif hata tespitinin deneysel olarak gerçeklenmesinden sonra hata maskelemesi tasarımı yapılmıştır. Hata maskelemesi için sıcak bekleme

dinamik yedekleme sistemi önerilmiştir. Bu sistemde iki kontrolcü eş zamanlı olarak çalıştırılır ve bir kontrolcüde hata tespit edilmesi durumunda kontrolcü izole edilerek, dinamik sistemin düzgün çalışan kontrolcüyle sürülmesi sağlanır. Sıcak bekleme dinamik yedekleme sisteminde kullanılmak üzere hata tespit mekanizması dizayn edilmiştir. Hata tespit mekanizması yardımcı sinyal çıkışının rms değerini gözetleyerek rms değerinin belirlenen hata sınırları dışına çıkması durumunda hata kararı verir. Ayrıca set-reset bloğu hata tespit mekanizmasının hata verme kararını hızlandırmak için kullanılmıştır. Hata kararı verildikten sonra basit anahtarlama mekanizması hatalı kontrolcüyü sistemden izole eder ve dinamik sistemin düzgün çalışan kontrolcüyle sürülmesini sağlar. Hata maskeleme tasarımı yapıldıktan sonra tasarım simülasyon ortamında gerçeklenir. Simülasyon ortamında tasarım dış etmen, örnekleme zamanı ve bit değişimi hatalarına karşı test edilir. Test sonucunda tasarım örnekleme zamanı hatasını dinamik sistemin cevabını etkilemeden maskalerken , dış etmen hatasının maskelenmesi 2.5sn ve bit değişimi hatasının maskelenmesi 3sn sürmüştür. Son kısımda da en çok tercih edilen yedekli kontrolcü tasarımı olan üçlü modüler yedekli kontrolcü tasarımı ile aktif hata tespiti metodu karşılaştırılmıştır. Karşılaştırma sonuncunda aktif hata tespiti tasarımın maliyet , uygulanabilirlik ve güvenirlik kıstaslarında üçlü yedekli modular kontrolcüye üstünlük kurduğu belirtilmiştir.

Hata mekanizmasının dış etmen hatasını tespit etmesi 0.24sn ve bit değişimi hatasını tespit etmesi ise 0.2sn sürmüştür. Bundan dolayı dinamik sistem 0.24sn dış etmen hatalı ve 0.2sn bit değişim hatalı kontrolcüyle sürülmüştür. Yedek kontrolcünün dinamik sistem üzerinde hatalı kontrolcünün etkisini düzeltmesi dış etmen hatası için 2.5sn ve bit değişimi hatası için 3sn sürmüştür. Bu çalışmada hata tespit mekanizması rms bloğu kullanılarak dizayn edilmiştir. Rms algoritması kümülatif bir yapı olduğu için kontrolcüdeki bazı hataların rms çıkışına etki etmesi süre alabilir bu da daha geç hata tespiti yapılmasına neden olur. Gelecek çalışmalarda hata tespit yapısında rms bloğu yerine daha anlık değişimlerin daha kolay yansıyacağı daha dinamik bir yapı üzerinde çalışılabilir.

Sonuç olarak tez kapsamında doğrusal kontrolcü donanımları için aktif hata tespit metodu önerilmiştir. Aktif hata tespit metodunun kurulması için ikizlenmiş kontrolcü tasarımı dizayn edilmiştir. Aktif hata tespitini kullanan ikizlenmiş kontrolcü tasarımının hali hazırda yaygın olarak kullanılan ve pasif hata tespitine dayanan üçlü

modüler kontrolcü tasarımına maliyet, güç tüketimi kıstaslarında üstünlük kurduğu gösterilmiştir. Çalışma kapsamında doğrusal kontrolcü donanımlarında aktif hata tespiti ve hata maskelenmesi tasarımları yapılmıştır. Doğrusal kontrolcüde hata tespiti yapmak için doğrusal sistemlerin frekans cevabından ve süperpozisyon ilkesinden yararlanılmıştır. Ama Doğrusal olmayan bir kontrolcü donanımında aktif hata tespiti yapmak için yardımcı sinyalin seçilmesi, frekans cevabının elde edilmesi ve süperpozisyon ilkesinin uygulanması doğrusal sistemlerde olduğu kadar kolay değildir. Bu nedenle gelecek çalışmalarda doğrusal olmayan kontrolcü donanımlarında aktif hata tespiti üzerinde çalışılabilir.

KAYNAKLAR

[1] **Michael Hutter, J.-M. S**. (2013). The Temperature Side Channel and Heating Fault Attacks. *CARDIS: International Conference on Smart Card Research and Advanced Applications.* Berlin.

[2]**Konstantin O. Petrosyants, I. A.** (2016). Hardware-Software Subsystem for Multilevel Thermal Fault Detection and Analysis of Electronic Components . *International Siberian Conference on Control and Communications (SIBCON)* . Siberian.

[3] **O'Flynn, C. P.** (2017, March 07). *United States Patent No. US 2017/0067961 A.*

[4] **Heather Quinn, Z. B**. (2017). Robust Duplication With Comparison Methods in Microcontrollers. *TRANSACTIONS ON NUCLEAR SCIENCE, 64*(1), 338-345.

[5]**Paulo R. C. Villa, R. T**. (2018). Processor checkpoint recovery for transient faults in critical applications. *Latin-American Test Symposium (LATS).* Sao Paulo.

[6] **Felix Möhre, K. B.** (2017). A Formal Approach for Automating Compositional Safety Analysis Using Flow Type Annotations In Component Fault Trees. *Safety and Reliabilty - Theory and Application.* Florida.

[7] **György Györök, B. B.** (2017). Duplicated Control Unit Based Embedded Faultmasking Systems . *International Symposium on Intelligent Systems and Informatics* . Subotica.

[8] **Baumann, R.** (2005). Soft Errors in Advanced Computer Systems. *IEEE Design & Test of Computers, 22*(3), 258-266.

[9] **L.Campbell, S., & Nikoukhah, R.** (2004). *Auxiliary Signal Design for Failure Detection.* Oxfordshire: Princeton University Press.

[10] **T.Sorsa, Koivo, H., & Koivisto, H.** (1991). Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 815-825.

[11] **Tor Aksel N.Heirung, A. M.** (2019). Input design for active fault diagonis. *Annual Review in Control, 47*(1), 35-50.

[12] **Moseler, O., & Isermann, R.** (2000). Application of model-based fault detection to a brushless DC motor. *IEEE Transactions on Industrial Electronics*, 1015-1020.

[13] **Luo, B., Wang, H., Liu, H., Li, B., & Peng, F.** (2018). Early Fault Detection of Machine Tools Based on Deep Learning and Dynamic Identification. *IEEE Transactions on Industrial Electronics*, 509-518.

[14] **Schoen, R., Habetler, T., Kamran, F., & Bartfield, R.** (1995). Motor bearing damage detection using stator current monitoring. *IEEE Transactions on Industry Applications*, 1274-1279.

[15] **Ravi Krishnan Unni, V. P.** (2018). FPGA Implementation of an Improved Watchdog Timer for Safety-critical Applications. *31th International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems.*

[16] **Venu Babu Thati, J. V.** (2017). Data Error Detection and Recovery in Embedded Systems: a Literature Review. *Advances in Science, Technology and Engineering Systems Journal, 2*(3), 623-633.

[17] **Mitra, S., & McCluskey, E.** (2000). Word-voter: a new voter design for triple modular redundant systems. *Proceedings 18th IEEE VLSI Test Symposium.* Montreal: IEEE.

[18] **Şinca, R., & Szász, C.** (2017). Fault-tolerant digital systems development using triple modular redundancy. *International Review of Applied Sciences and Engineering* , 3-7.

[19] **Lyons, R. E., & Vanderkulk, W.** (1962). The Use of Triple-Modular Redundancy to Improve Computer Reliability. *IBM Journal of Research and Development*, 200 - 209.

[20] ***micropilot.com/white-papers*.** (2020, July 29). micropilot: https://www.micropilot.com/pdf/white-papers/mp21283x.pdf adresinden alındı

[21] **Lodhi, F. K., Hasan, S. R., Hasan, O., & Awwad, F.** (2014). Low Power Soft Error Tolerant Macro Synchronous Micro Asynchronous (MSMA) Pipeline. *2014 IEEE Computer Society Annual Symposium on VLSI* (s. 601-606). Tampa: IEEE.

[22] **Sandra Vásquez, M. K.** (2019). Active Fault Diagnosis on a Hydraulic Pitch System Based on Frequency-Domain Identificatio. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, 27*(2), 663-678.

[23] **G. Roberto Marseglia, D. M**. (2017). Active fault diagnosis: A multi-parametric approach. *Automatica, 79*(223-230), 223-230.

[24] **François Bateman, H. N.** (2018). Active fault detection and isolation strategy for an unmanned aerial vehicle with redundant flight control surfaces. *Mediterranean Conference on Control and Automation.* Corsica.

[25] **Stoustrup, J.** (2009). AN OBSERVER PARAMETERIZATION APPROACH TO ACTIVE FAULT DIAGNOSIS WITH APPLICATIONS TO A DRAG RACING VEHICLE. *Fault Detection, Supervision and Safety of Technical Processes* . 2009.

[26] **Khairudin, M.** (2016). Comparison Methods for Converting a Spindle Plant to Discrete System. *Indonesian Journal of Electrical Engineering and Computer Science* , 575-582.

[27] **Franch, R., Restle, P., James, N., Huott, W., Friedrich, J., Dixon, R., . . . Salem, G.** (2007). On-chip timing uncertainty measurements on IBM microprocessors . *2007 IEEE International Test Conference* (s. 1-7). Santa Clara: IEEE.

[28] **Edwards, L.** (2005). *Open-Source Robotics and Process Control Cookbook Designing and Building Robust, Dependable Real-time Systems.* Burlington: Elsevier.

[29] **Mundher H.A.Yaseen, H. J**. (2018). Modeling and control for a magnetic levitation system based on SIMLAB platform in real time. *Results in Physics, 8*, 153-159.

[30] ***Amazon.com/ electronics-store*.** (2020, June 10). Amazon.com: https://www.amazon.com/ATMEGA328P-PU-with-Arduino-Bootloader-Uno/dp/B007SH0D0A adresinden alınmıştır

[31] **Kushner, D**. (2020, June 10). *http://web.eecs.umich.edu/~prabal/teaching/eecs582/*. umich.edu: http://web.eecs.umich.edu/~prabal/teaching/resources/eecs582/kushner11arduino.pdf adresinden alınmıştır

[32] **Sarkar, S., Ghosh, G., Mohanta, A., Ghosh, A., & Mitra, S.** (2017). Arduino based foot pressure sensitive smart safety system for industrial robots. *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (s. 1-6). Coimbatore: IEEE.

[33] **Zhang, P.** (2010). *Advanced Industrial Control Technology.* Oxford: Elsevier.

[34] ***http://medesign.seas.upenn.edu/*.** (2020, June 10). upenn.edu: http://medesign.seas.upenn.edu/index.php/Guides/MaEvArM-timer1 adresinden alınmıştır

[35] **Norman Dunbar.** (2020). *Arduino Software Internals A Complete Guide to How Your Arduino Language and Hardware Work Together.* New York: Springer.

[36] **Olsen, W.** (2020, June 10). *https://digitalcommons.usu.edu/phys_capstoneproject/49/*. usu.edu: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1052&context=phys_capstoneproject adresinden alınmıştır

[37] **Kim, M. H., Lee, S., & Lee, K. C.** (2009). Kalman Predictive Redundancy System for Fault Tolerance of Safety-Critical Systems. *IEEE Transactions on Industrial Informatics* , 46 - 53.

[38] **Kaczor, G., Młynarski, S., & Szkoda, M.** (2016). Verification of safety integrity level with the application of Monte Carlo simulation and reliability block diagrams. *Journal of Loss Prevention in the Process Industries*, 31-39.

[39] **Zhao, X., Malasse, O., & Buchheit, G**. (2019). Verification of safety integrity level of high demand system based on Stochastic Petri Nets and Monte Carlo Simulation. *Reliability Engineering and System Safety*, 258-265.

ÖZGEÇMİŞ

Ad-Soyad : **Mutluhan Özkan**

Uyruğu : **Türk**

Doğum Tarihi ve Yeri : **03.01.1995 Uşak/Banaz**

E-posta : **mutluhan.12@windowslive.com**

ÖĞRENİM DURUMU:

- **Lisans** : 2018, Orta Doğu Teknik Üniversitesi, Mühendislik Fakültesi, Elektrik-Eloktronik Mühendisliği
- **Yükseklisans** : 2020, TOBB ETÜ, Elektrik-Eloktronik Mühendisliği, Tezli Yüksek Lisans Ptogramı

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2019-2020	**Mercedes-Benz Türk**	**PEP Trainee**
2017-2018	**TAİ-TUSAŞ**	**CO-OP Intern**
2016-2017	**Arçelik**	**Stajer**

- **TOBB ETU Araştırma Bursu-2020**

YABANCI DİL: İngilizce, Almanca

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER

- **Özkan, M.**, & Kasnakoğlu, Ç. (2020). Active Fault Detection in Linear Controller Hardware with Sine Signal. *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)* (s. 90-94). Antalya: IEEE.