

Bilişsel Robotlarda Eylem Gözetimi

Action Monitoring in Cognitive Robots

Melis Kapotoglu, Cagatay Koc, Sanem Sariel, Gökhan Ince
Yapay Zeka ve Robotik Laboratuvarı
Bilgisayar Mühendisliği Bölümü,
İstanbul Teknik Üniversitesi, İstanbul, Türkiye
{kapotoglu, kocca, sariel, gokhan.ince}@itu.edu.tr

Özetçe —Bilişsel robotların eylemlerini yürütürken içinde bulunduğu ortama ve ortamdaki nesnelere zarar vermemesi için yürütme esnasında oluşan hataları sezmesi gerekir. Bu nedenle, robotlar planlarındaki eylemleri yürütürken aynı zamanda bir yürütme gözlem birimine ihtiyaç duyarlar. Bu çalışmada, eylem yürütme aşamasında oluşabilecek hata durumlarını, farklı sensör tipleri kullanarak sezilebilecek modele dayalı ve modelden bağımsız hata sezme yöntemlerini bir arada kullanan bir yürütme gözetim sistemi önerilmektedir. Önerilen sistemde, takip edilecek eylem tipine özgü Metrik Zamansal Mantık formülleri oluşturulmakta ve bu formüller zamansal olarak analiz edilerek hatalar sezilmektedir.

Anahtar Kelimeler—Yürütme Gözlemeleme, Hata sezme, Metrik Zamansal Mantık.

Abstract—Cognitive robots need to detect execution failures in runtime to prevent potential damages to their environments or objects in their interest. For this reason, robots monitor the execution to detect any inconsistencies. In this paper, we propose a hybrid monitoring system that processes different sensory data from different sources by using both model-based and model-free methods. Our proposed system continually monitors formulas, represented in Metric Temporal Logic (MTL), related to each action in order to detect failures. These formulas are analysed temporally according to the sensory data gathered during the execution to decide on a failure.

Keywords—Execution Monitoring, Failure Detection, Metric Temporal Logic.

I. GİRİŞ

Robotların fiziksel dünyada eylemlerini yürütürken hatalarla karşılaşmaları kaçınılmazdır. Ortamda herhangi bir zarara yol açmadan güvenli bir şekilde plan yürütebilmek için oluşan hataların sezilmesi gerekir. Bu sebeple, bilişsel bir robotun, planındaki eylemleri yürütürken çalıştığı ortamda sürekli olarak gözlem yapması gerekmektedir. Önceki çalışmamızda, bilişsel robotlar için planlama, çıkarım ve yaşam boyu deneysel öğrenme sistemi [1] sunulmuştur. Bu çalışmada ise, robotların eylem yürütme hatalarını sezmek için gerekli olan yürütme gözlemeleme birimlerinin tasarımı sunulmaktadır.

Robotlar kendi donanımlarından kaynaklanan hataları sezmek için sahip oldukları dahili sensörlerin yanında, dış

dünyada oluşan hataları ve beklenmedik durumları algılayabilmek için de ek sensörlere ihtiyaç duyarlar. Robotların tüm sensörlerinden gelen verileri yorumlayarak herhangi bir hata durumunda, bu hatayı sezebilmesi gerekir. Robot sistemlerinde hata gözlemeleme ile ilgili kapsamlı araştırmalar mevcuttur [2]. Robotlarda hata sezme sistemlerinde sıkça kullanılan model tabanlı yaklaşımlardan en yaygın olanı, gözlemlenen çıktılar beklenen sonuçlarla karşılaştırılarak tutarsızlıkların belirlendiği gözlem tabanlı yöntemdir [3]. Modelden bağımsız yöntem ise, toplanan verilerin önceden belirlenmiş bir modele bağlı olmadan kontrol edilmesi prensibine dayanır. Diğer sistemlerden farklı olarak, bu çalışmada modele dayalı ve modelden bağımsız hata sezme yöntemlerini bir araya getiren ve hata sezme aşamasında ses verisini de kullanan bir hata gözlem sistemi geliştirilmiştir. Önerilen sistem, modele dayalı hata sezme için Metrik Zamansal Mantık formüllerinin gözlenmesi ve hedef takibi algoritması [4] yöntemlerini uygular. Buna ek olarak, sonar ve ses verisini değerlendirme aşamasında limit değer kontrolüne dayanan [3] modelden bağımsız bir yaklaşım kullanılmaktadır.

Bu çalışmada model tabanlı yürütme gözlemeleme yöntemi, ses ve sonar verisinin modelden bağımsız bir yöntemle işlenmesi sonucu oluşturulan algılama eylemleri ile desteklenmiş ve gerçek robot sistemleri üzerinde yürütme hatalarını sezilen bir sistem tasarlanmıştır. Önerilen sistemin çalışması, Pioneer 3-AT robotumuzun ortamda bulunan nesnelere yerlerini belirleyerek bu nesnelere etkileşimini içeren görevleri yürütmesi esnasında yapılan analiz ile sunulacaktır.

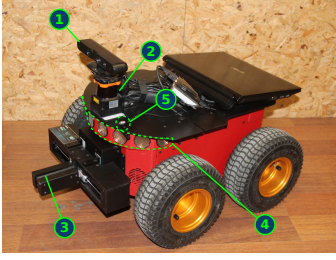
II. ROBOT EYLEM HATALARINI SEZME

Bilişsel robotlarda sürekli planlama, çıkarım ve yaşam boyu öğrenme üzerine sunulan sistem, *Sensör Arayüzü, Sahne Yorumlama, Bilgi Tabanı, Planlama, Öğrenme, Yürütme, Yürütme Gözlemeleme* ve *Motor Arayüzü* isimli sekiz ana modülden oluşmaktadır. Bu modüller arasındaki senkronizasyon ve haberleşme altyapısını sağlamak için sistem, ROS (Robot Operating System) üzerine kurulmuştur. Öğrenme ve öğrenme güdümlü planlama bileşenleri önceki çalışmalarda sunulmuştur [1]. Bu çalışmadaki asıl amaç *Sensör Arayüzü, Sahne Yorumlama, Yürütme* ve *Yürütme Gözlemeleme* modüllerini içeren hata sezme sisteminin tasarımıdır. Gerçeklenen tasarım, nesne etkileşim senaryoları üzerinde analiz edilmiştir.

A. Robot Donanımı

Pioneer 3-AT gezgin robotu, farklı tipte sensörler ile donatılarak gelişmiş bir bilişsel robot sistemi oluşturulmuştur.

Bu çalışma 111E286 no'lu proje kapsamında TÜBİTAK tarafından desteklenmektedir.



Şekil 1: Pioneer 3-AT robotu ve sensörleri: (1) ASUS Xtion PRO RGB-D Kamera (2) Hokuyo UTM-30LX Lazer Uzaklık Algılayıcı (3) Tutucu Pedallarındaki Dokunma Sensörleri (4) 8 adet Sonar Sensör (5) SONY ECMC115 Mandal Mikrofon

Böylelikle robot, ortamdan yeterince veri toplayabilecek ve beklenmedik durumları algılayabilecek hale gelmiştir. Şekil 1'de görüldüğü gibi Pioneer 3-AT, ön tarafında bulunan ve menzili 15 cm'den 7 m'ye kadar uzanan sekiz sonar sensöre sahiptir. Nesnelere etkileşimini sağlamak için ön tarafına 2 eksenli (2 DOF) tutucu monte edilmiştir. Tutucunun pedallarında dokunma sensörü bulunmaktadır. Robota özgü olan donanım parçaları ve sensörlerine ek olarak, 30 m menzile ve 270° tarama açısına sahip Hokuyo UTM-30LX lazer uzaklık algılayıcı robotun üzerine, sonar sensörler gibi ileri bakacak şekilde yerleştirilmiştir. Böylelikle haritalama ve konumlama yapılabilmektedir. Ayrıca, lazer uzaklık algılayıcı üzerinde, üç boyutlu nesne saptama ve tanıma işlemlerinde kullanılacak olan ASUS Xtion PRO RGB-D kamera sensörü bulunmaktadır. SONY ECMC115 mandal mikrofon, ortamdaki ses verisini değerlendirmek için kullanılmak amacıyla robot üzerine yerleştirilmiştir. Robotun kontrolü için Ubuntu 12.04 işletim sistemine sahip bir Intel i5 dizüstü bilgisayarı bulunmaktadır.

B. Sensör Arayüzü

Sensör Arayüzü, robot sisteminde bulunan sensörleri kullanarak ortamın algılanması, toplanan verilerin işlenmesi ve çıktılarının *Sahne Yorumlama* modülüne iletilmesini sağlar. Kullanılan sensörler farklı tipte ve özellikte veri topladığından, herbir sensör çıktısına farklı işlemler uygulanmaktadır.

1) *Haritalama ve Konumlama*: Robot üzerine yerleştirilen lazer uzaklık algılayıcı ile, RaoBlackwellized parçacık filtreleme [5] algoritması kullanılarak ortamın 2-B hücreli engel haritası oluşturulur ve Monte Carlo Konumlama (Monte Carlo Localization) [6] algoritması kullanılarak, robotun bilinen bir haritada kendi konumunu ve yönünü bulması sağlanır.

2) *3-B Nesne Saptama ve Tanıma*: RGB-D Kamera sensörü ile alınan üç boyutlu veri, LINE-MOD [7] algoritması ile işlenerek ortamdaki nesnelere konumları bulunmaktadır. LINE-MOD çok modellenmiş şablon eşleştirme yöntemine dayanan bir nesne tanıma yaklaşımıdır. Sistemi çalıştırmadan önce, deneylerde kullanılacak her nesne için nesne etrafındaki renk gradyanlarını ve üzerindeki yüzey normallerini içeren şablon dosyası oluşturulur. Elde edilen bu şablon dosyasına ek olarak, LINE-MOD algoritmasını nesnelere renklerini de değerlendirecek şekilde kullanmak için, şablonların renk histogramları da kaydedilir. Böylelikle, nesnelere hem şekilleri hem de renkleri baz alınarak tanımlanmaktadır [8]. Bir nesne sadece LINE-MOD ile tanındıysa, renk bilgisi yoktur ve renk histogramlarını

da değerlendiren geliştirilmiş LINE-MOD ile tanımadıkça rengi bilinemez. LINE-MOD ve renk histogramlı LINE-MOD çıktıları Şekil 2'de görülmektedir. Şablon tabanlı tanıma metoduna ek olarak, RGB-D kamera sensöründen alınan nokta bulutu üzerinde derinlik tabanlı bölütleme algoritması uygulanır. Bu yöntemle algılanan nesnelere, bilgi tabanında üç boyutlu nokta bulutları olarak temsil edilmektedir. Örnek bir segman çıktısı, Şekil 2'de görülmektedir. Bu nedenle, bölütleme algoritması ile nesnelere tipleri belirlenemez, sadece konum ve boyut bilgisine ulaşılabilir. Hedef nesnelere ortamdan ayırt edilebilmek için, yer zemin üzerinde bulunmayan veya belirli bir eşik değerinden daha büyük olan nokta bulutları elenmektedir.

3) *Ses İşleme*: Robot, bir nesneyi tutma veya yere bırakma eylemlerini gerçekleştirirken, kendisi nesneye çarpıp nesneyi düşürebilir veya nesne dışarıdan gelen bir etki ile de düşebilir. Nesnenin malzemesi ve şekline bağlı olarak, nesne yere çarptığında farklı bir ses çıkartabilir. Bu çalışmada robotun ses ile hata sezebilmesi için gerekli olan hata sezme yöntemi de geliştirilmiştir. Robotun üzerine yerleştirilen mikrofon ile ortamdan alınan ses verisi, nesnenin düşebileceği belirli zaman aralıklarında (ör. nesneyi tutma eylemi yürütülürken) değerlendirilir. Bu sayede, dinamik ortamda diğer zamanlarda oluşabilecek gürültülü veriler gözardı edilmiş olur.

Ortamdaki sesi algılama ve dalga formundaki sinyali Fourier dönüşümünden geçirerek zamansal hale getirme işlemleri, HARK robot işitme yazılım sistemi kullanılarak gerçekleştirilmektedir [9]. Dönüşüm sonucu elde edilen ses enerjisi değerlerinin ortalaması alınır ve verinin daha küçük ölçeklendirilmesi için logaritması hesaplanır. Bulunan değer, deneysel olarak belirlenen bir limit ile karşılaştırılarak ortamda oluşan beklenmedik ses yükselişleri Limit-Değer Kontrolü (Limit-Value Checking) ile zamansal olarak sezilir [3].

4) *Tutucu Pedalı Dokunma Sensörü Arayüzü*: Tutucu pedallarında bulunan dokunma sensörü, bir nesne tutuluyorsa veya pedallar tamamen kapalıysa tetiklenir. Pedalların yarım veya tamamen açık olduğu durumlarda ise sensör üzerinde basınç olmadığından farklı bir değer döner. Dokunma sensöründen elde edilen ve lojik değer alabilen bu veri, önerilen sistemde iki farklı şekilde kullanılmaktadır. Robot nesneyi aldıktan sonra, nesne tutucu pedallarından kayıp düşerse veya dışarıdan bir etkiyle alınırsa, başta tetikleniyor durumundaki sensörler boşa kalır ve robot tutucusunda olmasını beklediği nesnenin orada olmadığını sezer. Tutucu pedallarının arada nesne yokken, tamamen kapanma süresi sabittir ve önceden ölçülmüştür.

5) *Sonar Sensör Arayüzü*: Ses yayılımı ve yankı prensibine dayanarak çalışan sonar sensörler, robot üzerinde yerden 23 cm yükseklikte yer aldıklarından, bu sensörlerin çıktıları yeterince



Şekil 2: LINE-MOD (solda) ve renk histogramlı LINE-MOD (ortada) ile nesne tanıma, bölütleme algoritması ile (sağda) nesne yerini saptama.

uzun nesnelere yapılan deneylerde değerlendirilir. Robot ilgili nesneyi tutmak için yaklaştığında, nesnenin, sonar sensör dizisinin ortasındaki iki sensörden en az biri tarafından belirlenebilmesi gerekir.

C. Sahne Yorumlama

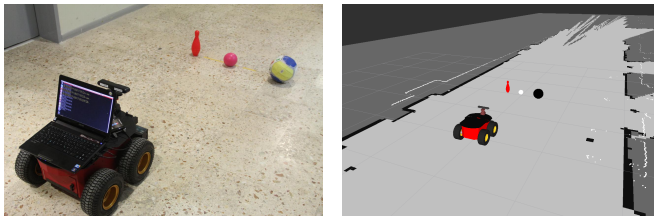
Robotun sensörleriyle ortamdaki topladığı veri gürültü içerir. Bu nedenle, toplanan ve işlenen veriyi, doğrudan bilgi tabanına kaydetmekten önce yorumlama sürecinden geçirip anlamlandırıldıktan sonra depolamak, tutarlı bir dünya modeli elde edebilmek için gerekli ve önemli bir adımdır. Sensörlerden alınan veriler *Sensör Arayüzü* modülünde, her sensör için farklılık gösteren işlemlerden geçtikten ve anlamlı hale getirildikten sonra *Sahne Yorumlama* modülüne gelir ve bir araya getirilerek çıkarımlar yapılır. Böylelikle bilgi tabanına kaydedilecek verinin güvenilirliği, birden fazla kaynak değerlendirilerek artırılmış olur. Toplanan veriler üzerinde yapılan çıkarımlar bilgi tabanına kaydedildikten sonra, sistemin diğer modüllerine bilgi tabanı üzerinden aktarılır.

Tanıyan ve/veya saptanan nesnelere konum bilgilerini kaydederken, şablon eşleştirme tabanlı tanıma metodu ile bölütleme sonucu elde edilen bilgi birleştirilir ve kaydedilen her nesne hesaplanan güvenilirlik değeri ile birlikte tutulur. LINE-MOD, renk histogramlı LINE-MOD ve bölütleme algoritmaları güvenilirlikleri birbirinden farklı olan kaynaklar olduklarından, her birine deneysel olarak belirlenen ağırlıklar atanır ve nesnelere güvenilirlik değerleri, veri alınan kaynağın ağırlığına göre hesaplanarak güncellenir [10]. Robot ortamda yeni nesnelere saptadıkça veya tanıdıkça bu nesnelere bilgileri bilgi tabanına eklenir. Robot başka bir yöne döndüğünde veya ilerlediğinde önceden kaydedilmiş nesnelere bilgi tabanında depolanmaya devam eder. Örnek bir sahnede, robotun haritadaki konumu ve bilgi tabanında depolanan nesnelere Şekil 3'te görülmektedir.

D. Yürütme Gözlemlene ve Hata Sezme

Yürütme gözlemlene sisteminin çalışabilmesi için, dünya durumunun sürekli olarak takip edilmesi gerekir. Sensörlerden alınan verinin işlenmesi ve yorumlanıp birleştirilmesi sonucu, ortamda bulunan nesnelere, robotun sensörleriyle veya doğrudan o anki dünya durumu ile ilgili yüklemeler üretilerek bilgi tabanına kaydedilir. Üretilen yüklemeler robot, ortamı algıladığı sürece güncellenir ve yürütme gözlemlene sistemine gönderilir.

1) *Bilgi Temsili için Oluşturulan Yüklemeler*: Bilgi tabanına depolanan her nesneye tekil bir numara atanır ve nesne bilgisi tabanında varolduğu sürece i numaralı nesne için $nesne_i$



Şekil 3: Gerçek robot (solda) ve robotun bilgi tabanı (sağda).

yüklemeli oluşturulur. Robot konumunun veya yönünün belirli bir zaman aralığında değişmesini ifade eden yüklem $robotHareketli$ dir. $tutucuYukari$ yüklem, tutucu yukarı doğru hareket ediyorken, $tutucuKapali$ yüklem ise tutucunun pedalları kapatılıyorken yüklem listesine eklenir. Bu işlemler sona erdiğinde silinir.

Tutucu pedallarındaki dokunma sensörlerini kontrol ederek oluşturulan $elinde_i$ yüklem robotun i numaralı nesneyi tutması ile oluşturulur ve nesne bırakıldığında silinir. $nesneyeYakin_i$ yüklem ise robot, i numaralı nesneye belirli bir yakınlığa ulaştığında üretilir. Benzer şekilde, $nesneTutmaPozisyonunda_i$ yüklem, i numaralı nesne tutucu pedalları arasına girmeye başladıktan itibaren oluşturulur ve tutucu pedalları kapanmaya başlayana kadar saklanır. Sonar sensörle ilgili iki yüklem yaratılmaktadır. Bunlardan ilki, $uzunNesne_i$, sonar sensörler ile farkedilebilecek kadar uzun olan nesnelere için üretilir ve $sonarlaSaptananNesne_i$ yüklem i numaralı nesne, sonar sensörle saptandığında aktive edilir. Ortamda beklenmedik bir ses algılandığında, $sesAsim$ yüklem oluşturulur.

2) *Eylem Gözetim Kuralları*: Önerilen sistemde, eylem gözetim kurallarını ifade etmek için Metrik Zamansal Mantık yöntemi kullanılmaktadır. Metrik Zamansal Mantık, ikili ve zamansal bağlaçları kullanarak mantıksal önermelerin ifade edilebileceği bir dildir. Bu dildeki ikili bağlaçlar; \wedge (ve), \vee (veya) ve \neg (değil); zamansal bağlaçlar ise; \square_τ (her zaman), U_τ (kadar), \diamond_τ (sonunda) ve \bigcirc_τ (sonraki)'dir. τ zamansal bağlaçlar ile \geq_t , \leq_t , $<_t$ ve $>_t$ şeklinde (t pozitif bir tam sayı olmak üzere) zaman aralıkları tanımlamak için kullanılır. $\square_\tau f$ ifadesi f kuralının, τ zaman aralığında tüm dünya durumları için doğru olduğunu belirtir. $\diamond_\tau f$, f kuralının τ zaman aralığındaki en az bir durumda doğru olacağını ifade eder. $\bigcirc_\tau f$ ise f kuralının τ zaman aralığında gerçekleşen bir sonraki durumda doğru olacağını tanımlar. $f_1 U_\tau f_2$, τ zaman aralığında f_1 kuralının, f_2 kuralı doğru olana kadar doğru olduğunu belirtir. MTL formüllerini değerlendirmek için aşamalı ilerleme algoritması (goal progression algorithm) kullanılmıştır.

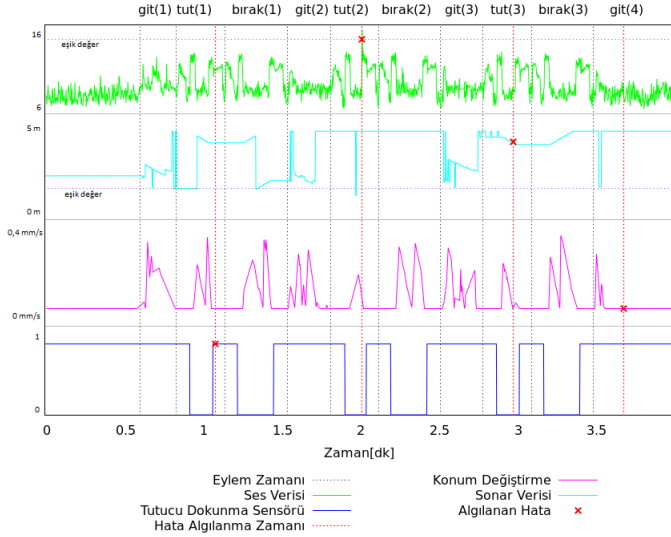
$git(nesne_i)$ eylemi için üç farklı formül kullanılmaktadır. Denklem (1), robot nesneye ilerlerken, nesnenin bilgi tabanındaki varlığını kontrol eder. Robotun yol planlaması yaptıktan sonra, nesneye yaklaşıp hareket edip etmediği, Denklem (2) ile kontrol edilir. Belirlenen zaman periyodu zarfında robotun nesneye istenen yakınlığa ulaşması Denklem (3) ile takip edilir. t_b süresi eylemin başladığı anı, $t_{planlama}$ yol planlaması yapmak için gereken süreyi ve $t_{nesneyeGitme}$ ise robotun nesneye ulaşabilmesi için gerekli olan süreyi ifade eder.

$$\square_{\geq t_b}(nesne_i) \quad (1)$$

$$\square_{\geq t_{planlama}}(robotHareketli U_{\geq t_b}(nesneyeYakin_i)) \quad (2)$$

$$\diamond_{\geq t_{nesneyeGitme}} nesneyeYakin_i \quad (3)$$

$tut(nesne_i)$ eylemi için altı farklı formül eklenmiştir. Denklem (4), git fonksiyonunda olduğu gibi nesnenin durumunu eylem yürütülürken kontrol eder. Denklem (5), robotun tutucusunu yukarı kaldırma aşamasında nesneyi tutuyor olması gerektiğini belirtmektedir. Denklem (6), robot nesneye yeterince yakınsa ve ses değişimi varsa (nesnenin düştüğü varsayımıyla) hatanın sezilmesini sağlar. Denklem (7) ile, robot nesneyi tutma pozisyonuna gelmeden önce tutucuyu kapatırsa hata sezilir. Denklem (8), nesne tutma pozisyonuna gelene kadar robotun hareketini takip eder. Sonar sensörlerle



Şekil 4: Robotun sensör verileri ve eylem yürütmesi anında sezilen hatalar (kırmızı işaretler ile) gösterilmektedir.

saptanacak kadar uzun nesnelere üzerinde sonar kontrolünün yapılması Denklem (9) ile gerçekleştirilir.

$$\square_{\geq t_b}(nesne_i) \quad (4)$$

$$\square_{\geq t_b}(\neg tutucuYukari U_{\geq t_b}(elinde_i)) \quad (5)$$

$$\square_{\geq t_b}(\neg(nesneyeYakin_i \wedge sesAsim)) \quad (6)$$

$$(\neg tutucuKapali) U_{\geq t_b}(nesneTutmaPozisyonunda_i) \quad (7)$$

$$robotHareketli U_{\geq t_b}(nesneTutmaPozisyonunda_i) \quad (8)$$

$$(\neg uzunNesne_i) \vee (\neg nesneTutmaPozisyonunda_i U_{\geq t_b} sonarlaSaptananNesne_i) \quad (9)$$

$birak(nesne_i)$ eyleminde Denklem (10) kullanılarak, nesneyi bırakırken nesnenin düşmesi ses ile algılanmaktadır. Denklem (11), robotun $t_{birakma}$ süresi zarfında nesneyi bırakması gerektiğini belirtir. $t_{birakma}$ süresi, hatasız durumda nesneyi bırakma süresi ölçülerek belirlenmiştir. Denklem (12) ise, uzun nesnelere bırakılma anlarında sonar tarafından algılanması gerekliliğini belirtmektedir.

$$\square_{\geq t_b}(\neg(nesneTutmaPozisyonunda_i \wedge sesAsim)) \quad (10)$$

$$\square_{\geq t_{birakma}} \neg elinde_i \quad (11)$$

$$(\neg uzunNesne_i) \vee (\neg nesneTutmaPozisyonunda_i U_{\geq t_b} sonarlaSaptananNesne_i) \quad (12)$$

E. Analiz

Önerilen sistem fiziksel dünyada gerçek robot üzerinde analiz edilmiştir. Robot, bilgi tabanında bulunan nesnelere sırayla gidip, tutmayı ve bırakmayı denemektedir. Yapılan analizin sonuçları Şekil 4'te görülmektedir. Deneyde iki farklı büyüklükte top, bir lobut ve sonar sensör ile algılanabilecek uzunlukta dikdörtgenler prizması şeklinde bir kutu kullanılmıştır. Bu deneyler sırasında hatalar, sisteme el ile enjekte edilmiştir. Analizi verilen senaryoda, robot ilk olarak küçük topa ilerler ve nesneye yaklaşırken nesne önünden alınır.

Deney esnasında, robot tutucusunu kapatırken tutucu pedalındaki dokunma sensörünü kullanarak pedalları kapatma süresinin, olması gerektiğinden fazla sürdüğünü Denklem (5)'i kontrol ederek algılamış ve nesneyi tutamadığını sezmiştir. Robot ikinci olarak lobut nesnesine yönelir ve nesneye yaklaştığı anda lobut dışarıdan bir etki ile düşürülür. Lobutun düşmesiyle beraber ortamdaki ses dalgaları, belirlenen eşik değerini aştığından robot nesnenin düştüğünü tutucusunu kontrol etmeden önce Denklem (6) ile algılayabilmiştir. Ardından robot uzun, prizma şekilli kutuya yönelmiş ve nesneye yaklaşırken nesne önünden alındığından, sonar sensörleri ile nesne algılanamayıp Denklem (9) sağlanmamış ve hata yine tutucu kontrolünden daha erken sezilebilmiştir. Son olarak büyük topa gitme eylemi yürütülürken robotun motorları durdurulmuş ve sisteme konumlama hatası enjekte edilmiştir. Belirli bir süre konum değişimi algılamadığından, robot Denklem (2)'yi kullanarak hatayı başarıyla saptayabilmiştir.

III. SONUÇ

Bu çalışmada, modele dayalı hata sezme için Metrik Zamanlı Mantık formülleri kullanılarak hata durumları fiziksel ortamda tespit edilmektedir. Bunun yanında, model tabanlı hata sezme yöntemine, sonar sensörleri ve ses algılayıcı sensörlerden gelen bilgilerin değerlendirildiği modelden bağımsız bir yaklaşım entegre edilmiştir. Yapılan analiz robotun farklı tipteki hataları farklı sensörlerle sezilebildiğini göstermektedir. Sensör çeşitliliği hataların daha erken sezilebilmesini sağlamaktadır. İleri çalışma olarak, farklı senaryolarda daha kapsamlı analizlerin yapılması planlanmaktadır.

KAYNAKÇA

- [1] S. Karapinar, M. Ersen, M. Kapotoglu, P. Yıldız, S. Sariel-Talay, and H. Yalcin, "Bilisel robotlarda deneysel öğrenme," in *21. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Girne, KKTC, 2013.
- [2] O. Pettersson, L. Karlsson, and A. Saffiotti, "Model-free execution monitoring in behavior-based robotics," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 890–901, 2007.
- [3] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.
- [4] J. Kvarnström, F. Heintz, and P. Doherty, "A temporal logic-based planning and execution monitoring system," in *ICAPS*, 2008, pp. 198–205.
- [5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 425–437, 2002.
- [6] D. Fox, "Kld-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, no. 1, pp. 26–32, 2001.
- [7] S. Hinterstoisser, C. Cagniard, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012.
- [8] M. Ersen, S. Sariel-Talay, and H. Yalcin, "Extracting spatial relations among objects for failure detection," in *Proc. of the KI 2013 Workshop on Visual and Spatial Cognition*, 2013, pp. 13–20.
- [9] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Design and implementation of robot audition system 'hark'—open source software for listening to three simultaneous speakers," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2010.
- [10] M. D. Ozturk, M. Ersen, M. Kapotoglu, C. Koc, S. Sariel-Talay, and H. Yalcin, "Scene interpretation for self-aware cognitive robots," in *AAAI Spring Symposia, Qualitative Representations for Robots*, 2014.