

# Bilişsel Robotlar için Eylem Yürütme Hatalarının Tanısı

## Diagnosis of Action Execution Failures for Cognitive Robots

Dogan Altan, Sanem Sariel

Yapay Zeka ve Robotik Laboratuvarı, Bilgisayar Mühendisliği Bölümü,

İstanbul Teknik Üniversitesi, İstanbul, Türkiye

{daltan,sariel}@itu.edu.tr

**Özetçe** —Gerçek dünyanın dinamik ve kısmi gözlemlenebilir yapısı nedeniyle, robotik uygulamalarında yürütme hatalarının oluşması muhtemeldir. Bu hata durumları robot tarafından sezilmeli ve bu durumların nedenlerini belirlemek için bir çıkarsama yordamı yürütülmelidir. Bu çalışmada, robotların tespit ettikleri hataların tanısı için Hiyerarşik Saklı Markov Modellerini (HSMM) kullanan bir yöntem önerilmektedir. Önerilen yöntemde, farklı tipteki hataları izlemek için paralel olarak çalışan HSMM'lerin kullanılması uygun görülmüştür. Yöntemin başarımlarını analiz için Pioneer 3-AT gezgin robotu ile farklı tipte hatalar içeren senaryolar üzerinde yapılmıştır. Elde edilen sonuçlar, geliştirilen yöntemin hatanın birden fazla nedene bağlı olduğu durumlarda da hataların olasılıksal olarak belirlenebileceğini göstermektedir. Ayrıca, hata tipleri ve eylemler arasındaki ilişkilerin kullanılması da durum uzayını daraltarak, yöntemin bellek gereksinimini azaltmaktadır.

**Anahtar Kelimeler**—Robotlarda Hata Tanısı, Modele Dayalı Tanı, Hiyerarşik Saklı Markov Modeli.

**Abstract**—Execution failures are likely in robotic applications due to dynamic and partially observable structure of the physical world. These failures should be detected by the robot, and a reasoning procedure should take place to diagnose the causes of the failures. In this paper, we propose a Hierarchical Hidden Markov Model (HHMM) based failure diagnosis method to identify the cause of a failure. Parallel HHMMs are used in the proposed method in order to track different type of failures. The performance of the proposed method is evaluated on our Pioneer 3-AT robot in several failure scenarios. The results reveal that using a probabilistic method ensures diagnosing multiple failures when there are more than one cause of a failure. Furthermore, using relations between the failure types and actions decreases memory requirements of the method by reducing the state space.

**Keywords**—Failure Isolation for Robots, Model-Based Diagnosis, Hierarchical Hidden Markov Model.

Bu çalışma 111E286 no'lu proje kapsamında TÜBİTAK tarafından desteklenmektedir. Yazarlar görüntü işleme algoritmasındaki katkılarından dolayı Mustafa Ersen'e ve robot deneylerindeki katkılarından dolayı Melis Kapotoğlu, Çağatay Koç, Melodi Deniz Öztürk ve Mehmet Biberçi'ye teşekkür eder.

### I. GİRİŞ

Robotlar, planlarındaki eylemleri yürütürken çeşitli hata durumları ile karşılaşabilirler [1]. Bu hataların sezilebilmesi için yürütme hem eylem hem de plan seviyesinde gözlemlenmelidir. Gürbüz eylem yürütme için her ne kadar hata sezme önemli bir problem olsa da [2], oluşan hatanın nedeni de (hata tanısı) belirlenebilmelidir. Hata tanısı yapılabilmesi için de bir çıkarım mekanizmasının yürütülmesi gerekmektedir. Bu çalışmada, olasılıksal bir hata tanı yöntemi önerilmektedir. Çalışma kapsamında, donanım/duyurga kısıtları, kısıtlı bilgi veya çevresel dış etkenler nedeniyle oluşabilecek eylem yürütme hataları ele alınmaktadır.

Yürütme gözlemlenebilir ve hata durumları hakkında çıkarım yapma bir veya daha fazla duyurgadan gelen bilgiyi yorumlamak ile mümkün olabilmektedir. Bu nedenle, robotun hata sezme için ortamı yorumlaması ve çıkarsama yordamı yürütmesi gerekir. Benzer bir yordam, robotun hata tanısı yapabilmesi için de gereklidir. Bazı durumlarda, hatanın nedeni o anda yürütülmekte olan eylem ile doğrudan ilintili olmayabilir. Hatanın nedeni, robot tarafından daha önce yürütülmüş olan bir eylemin istenmeyen bir sonucuna veya çevresel bir etmene ilişkin olabilir. Bu gibi durumlarda hatanın nedenini belirleyebilmek için olasılıksal ve zamansal bir modele ihtiyaç duyulmaktadır. Bu bildiride, hata durumlarının nedenlerini belirleyebilmek için Hiyerarşik Saklı Markov Modellerine (HSMM) dayalı bir hata tanı yöntemi önerilmektedir. Önerilen yöntem, paralel olarak işlenen ve farklı tipte hata durumlarını temsil eden HSMM'lerini içermektedir. HSMM'lere dayalı olan model, durumların zamansal boyutta analizinin yapılmasına olanak vermekte ve beklenen durumlardan bir sapma gerçekleştiğinde zamansal hata bilgisinin durumlar arasında yayılmasını sağlamaktadır. Birden çok hipoteze göre hata durumunun olasılıksal olarak tanısını yapabilmek için farklı durumların aynı anda takip edilmesi gerekir. Bu çalışmanın ana katkısı, her bir hata tipinin eylem-hata tipi ilişkilerini de kullanarak ayrı bir hiyerarşik SMM olarak temsil edilmesidir. Bu temsil, bir nesne veya çevresel olay üzerine kalıcı olan hata durumları da dahil olmak üzere birçok hata durumunun tanısını mümkün kılmaktadır.

Literatürde farklı uygulama alanlarında hata tanısı için SMM'ler kullanılmıştır [3], [4]. Ayrıca, Kısmi Gözlemlenebilir

Markov Karar Süreçleri (KGMKS) ve Parçacık Filtreleri (PF) de otonom sistemlerde hataları modellemek ve tanımak için kullanılmıştır [5], [6]. Bir çalışmada, robotun görüntü işleme, konum belirleme ve dış etkenler nedeniyle karşılaştığı hatalara, mantıksal ifadeler olarak verilen arka plan modelleri üzerinde yürütülen çıkarım algoritması ile tanı konulmuştur [7]. Başka bir çalışmada [8], yürütme sırasındaki beklenmedik durumlar, üretilen hipotezlerin maliyetlerini ve tutarlılıklarını göz önüne alan bir mantıksal programlama yöntemi ile açıklanmıştır.

## II. OLASILIKSAL HATA TANISI

Robot, istenen bir hedef durumuna ulaşmak için sembolik bir plan üretmelidir. Üretilen plandaki ardışıl eylemlerin yürütülmesinin sonucunda robotun istenen hedef durumuna ulaşması beklenir. Robotun ortamda bulunan nesnelere başlangıç konumlarından istenen konumlara taşınmasının gerektiği bir senaryoyu ele alalım. Başlangıçta bütün nesnelere ilk konumlarında olsunlar. Robot *tut*, *nesneye-git*, *konuma-git* ve *birak* eylemlerini yürütebilmektedir. *konuma-git* eylemi robotu bir noktadan başka bir noktaya hareket ettirmek için yürütülürken *nesneye-git* eylemi robotu bir nesnenin yanına konumlandırmaktadır. *tut* eylemi robotun tutucusuyla bir nesneyi tutmasını sağlarken, *birak* eylemi robotun bir nesneyi tutucusundan ayırmasını sağlar. Bu eylem tanımları göz önüne alındığında, verilen örnekteki hedefi yerine getirmek için robotun planlayıcısı şu şekilde bir plan üretir: [*nesneye-git(nesne)*, *tut(nesne)*, *konuma-git(hedef)*, *birak(nesne)*]. Robot, üretilen bu sembolik planın yürütülmesiyle hedefine ulaşmış olur.

Verilen plandaki eylemlerin yürütülmesi sırasında robot, çeşitli hata durumları ile karşılaşabilir. Örneğin, robot ilgili nesneyi yanlış yerinden kavradığı için veya boyutundan dolayı kaldıramayabilir. Başka oluşabilecek bir hata ise robotun görüntü işleme modülünün nesneyi tanıyamamasıdır. Başka bir senaryoda ise nesne, robot dışında başka bir dış etmen tarafından manipüle edilebilir. Bu gibi beklenmedik durumlarda robotun hata tanıma birimi, oluşan hata durumlarının nedenlerini belirten açıklamalar getirebilmelidir.

### A. SMM Tabanlı Hata Tanısı

Saklı Markov Modeli (SMM) [9] Markov süreçlerinin modellenmesini sağlayan zamansal yapılarıdır. Planların yürütülmesi sırasında oluşan hatalar genellikle zamanda yayıldığı için, hata tanıma işlemi de zamansal boyutta analiz edilmelidir. Ayrıca, belirsizlik ve eylemlerin gerekirci olmayan yapısı, SMM tabanlı yöntemleri hata tanıma işlemi için uygun kılmaktadır.

Bir SMM modeli, gizli durumlar, gözlemler, geçiş modeli, gözlem modeli ve ilk durum dağılımı olmak üzere beş bileşenden oluşur. Bir SMM'de  $N$  tane gizli durum vardır ve  $\mathcal{S}$  gizli durumların kümesi olmak üzere her bir gizli durum  $s_i \in \mathcal{S}$  ile temsil edilir. Geçiş modeli  $\mathcal{A} = a_{ij}$  ile temsil edilir ve  $s_i, s_j \in \mathcal{S}$  olmak üzere  $s_i$  durumundan  $s_j$  durumuna geçme olasılığını belirtir. Duyargalardan  $t$  zamanında alınan gözlemler  $y_t$  ile gösterilir. Gözlem modeli ise  $\mathcal{B} = b_{s_i}(y)$  ile temsil edilir ve  $s_i$  durumunda  $y$  gözleminin gözlemlenme olasılığını verir. İlk durum dağılımı ise,  $\pi = \pi_i$  ile gösterilir ve ilk olarak sistemin verilen durumlardan hangisinde olacağı olasılığını temsil eder.

Önerilen yöntemde [10], klasik SMM modelleri hiyerarşik bir yapıda [11] kullanılmaktadır. Bu hiyerarşide, SMM modeli

iki katmanlıdır ve üst katmandaki her durum, ayrıca alt katmandaki durumları içerir. Üst katmandaki bir SMM düğümü etkinleştiği zaman, bu düğümün alt katmanındaki SMM aktive edilir ve model güncellemesi bu katmanda devam eder. Üst katmandaki modelin güncellemesi alt katmandaki modelin bir son duruma gelmesine kadar askıya alınır. Alt katmandaki SMM bir son duruma geldiğinde, üst katmana haber gönderilir ve model güncellemesi üst katmandan devam eder. Önerilen yöntemde her bir hata tipi, ayrı bir hiyerarşik SMM olarak temsil edilmiştir. Hata tiplerinin bu şekilde hiyerarşik olarak temsil edilmesinin nedeni plan, eylemler ve eylem durumları arasında bir hiyerarşi kurmaktır. Kurulan bu hiyerarşi ile kalıcı hataların tanınması da mümkün olacaktır. Örneğin, robotun görüntü işleme birimi belirli bir nesneyi hiçbir zaman tanıyamıyor olabilir. Bu durumda, kurulan hiyerarşi sayesinde bu tipteki hataların tanısı mümkün olacaktır.

Paralel olarak yürütülen her bir hiyerarşik SMM modeli  $M_i$  ile gösterilir. Bu temsilde  $i$ , hatanın indisidir.  $M_i$  hata modelindeki her bir durum ise  $M_i[ws_j]$  ile temsil edilir.  $ws_j$ , planda  $t$  anındaki ayrı zaman adımındaki dünya durumudur. Hiyerarşik modellerdeki her  $M_i[ws_j]$ , başarı ve hata şeklinde gizli değişkenlere sahiptir. Yöntemin genel amacı, gizli değişkenleri hata ile işaretlenmiş ve hesaplanan hata nedeni olma olasılığı değeri belli bir eşik değerin üstündeki modelleri bulmaktır.

Bu çalışmada incelenen eylem yürütme hataları şunlardır: Bütün nesnelere için *Görüntü işleme*, *Konumlama*, *Donanım kısıtları* (eyleyici/efektör) ve *Dış olay* hatalarıdır. *Görüntü(nesne)* hata modeli, görüntü işleme algoritmasının belirli bir nesneyi doğru olarak tanıyamadığı durumu temsil eder. *Donanım kısıtları* (eyleyici/efektör) hata tipi, robotun fiziksel kapasitesi dışındaki durumları temsil eder. *Dış olay* hata tipi ise robotun kontrolü dışında, dış dünyada oluşan değişiklikler nedeniyle oluşan hataları temsil eder. *Konumlama* hata tipi robotun ortamda konumunu belirleyemediği durumları temsil eder. Yürütülen eylem ve parametrelerine bağlı olarak, paralel olarak yürütülen hata tiplerinden ilgili modeller aktif hale geçer. Aksi durumda, yani hata modeli ve yürütülen eylem ilgili değil ise, bu model pasif hale geçer. Yürütme anında bir modelin, yürütülen eylem ile ilgili olup olmadığına *Eylem-Hata Tipi* ilişkilerine (Tablo I) bakılarak karar verilir.

Önerilen sistem, planlayıcı tarafından üretilen ve eylem dizisi içeren sembolik planı ve ayrı ayrı paralel olarak temsil edilen hata tipi modellerini girdi olarak kabul etmektedir. Sistemin çıktısı ise, hata durumunun nedeninden sorumlu olan olası hata modelleri listesidir. Robot planında yer alan bir eylem üç durumda olabilir: *başarı*, *hata* ve *yürütmede*. *başarı* durumu robotun bir eylemi beklenen sonuçları ile sonlandırdığını, *hata* durumu eylemin beklenmedik etkilerle

Eylem	Hata Tipi
<i>nesneye-git(nesne)</i>	Görüntü(nesne), Dış Olay Konumlama(Robot)
<i>konuma-git(hedef)</i>	Konumlama(Robot)
<i>tut(nesne)</i>	Dış Olay, Tutucu(Robot) Görüntü(nesne), Konumlama(Robot)
<i>birak(nesne)</i>	Tutucu(Robot)

Tablo I: Eylem-Hata Tipi İlişkileri

sonlandığını ve *yürütmede* ise robotun eylemi yürütmeyi henüz bitirmediğini temsil eder. Her eylemin yürütülmesi sırasında, Hiyerarşik SMM'lerin alt katmanında o eyleme karşı düşen modeller Tablo I'deki ilişkiler göz önüne alınarak güncellenir. Eğer Tablo I'e göre hata tipi modeli ve eylem ilgili ise model güncellenir, değil ise o eylem süresince ilgili hata modeli pasif kalır. Yürütme aşamasında robotun duyargaları aracılığı ile alınan gözlemler, duyargaların istatistiksel analizine bağlı olan bir olasılık dağılımına karşılık düşürülür ve bu değere göre yeni durum oluşturulur.

İlgili eylemin yürütülmesi sona erince, alt katman modelinde Viterbi Algoritması [12] yürütülür ve alt katmandaki durumlardan hata nedeni olma olasılığı en yüksek olan durumun değeri üst katmana aktarılır. Üst katmandaki model, yürütülen eylem ile ilgili ise güncellenir. Aksi takdirde model pasif olarak işaretlenir ve ilgili modelin şimdiki zaman düğümü yaratılmadan bir önceki durumun zaman vektörü genişletilir. Bu durum, robot planındaki bütün eylemleri başarılı bir şekilde yürütüncüye veya hata durumu ile karşılaşınca kadar devam eder. Hataların, gözlemlenmeye dayalı bir yöntem ile sezildiği varsayılmaktadır. Hata durumu oluştuğunda, üst katman modellerinde Viterbi Algoritması yürütülerek hataya neden olabilecek modeller belirlenir. Bunun için, modellerdeki durumlar incelenir ve *hata* ile işaretlenmiş durumlar saptanarak olasılıklarına bakılır. Bu olasılıklardan belli bir eşik değerinin üzerinde olan modeller aday listesine eklenir.

### III. DENEYLER

Önerilen yöntem, RGB-D kamera kullanan Pioneer 3-AT robotu (Şekil 1a) üzerinde çeşitli hata durumları sisteme enjekte edilerek test edilmiştir. Deneylerde bir lobut, bir küçük top ve bir voleybol topu (Şekil 1b) kullanılmıştır. Robotun dünya durumu hakkındaki algısı bir sahne yorumlama birimi [13] tarafından duyargalardan gelen verilere göre yapılmaktadır. Tablo II'de robotun bilgi tabanında tutulan bazı yüklemeler verilmiştir. *nesne* yüklemi, robotun ilgili nesneyi görüntü işleme birimi ile tespit ettiğini, *yerde* yüklemi nesnenin zemin üzerinde bulunduğunu, *bölüt* yüklemi nesnenin bölütleme algoritması ile tespit edildiğini, *eli-boş* yüklemi robotun tutucusunun boş olduğunu ve *eli-dolu* yüklemi robotun tutucusuyla bir nesne tutmakta olduğunu belirtir.

Modeller için geçiş ve gözlem olasılıkları duyargaların istatistiksel analizleriyle belirlenmektedir. Görüntü işleme hataları için kullanılan Koşullu Olasılık Tabloları (KOT), görüntü işleme algoritmasının çıktısı olan benzerlik oranına göre ve kullanılan bölütleme algoritmasının başarımına göre belirlenmektedir. Konumlama hatası modeli için odometri ve konumlama algoritmasındaki sapmalar dikkate alınmaktadır. Dış olay hata modelinde sahne yorumlama biriminden gelen bilgiler doğrultusunda KOT oluşturulur. Donanım hataları için ise, tutucudaki basınç duyargası yorumlanır.

Yüklem	Kaynak
nesne	RGB-D Kamera
yerde	RGB-D Kamera
bölüt	RGB-D Kamera
eli-boş	Basınç Duyargası
eli-dolu	Basınç Duyargası

Tablo II: Yüklemeler ve Yüklemelerin Alındığı İlgili Duyargalar



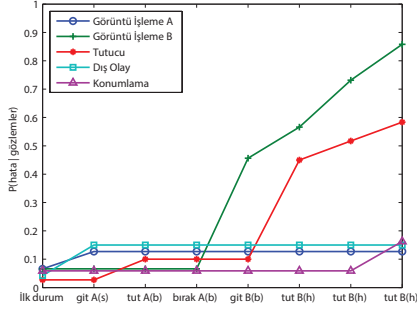
Şekil 1: (a) Pioneer 3-AT (b) Deneylerde kullanılan nesnelere

1) *Senaryo I*: Bu senaryoda, robotun görevi bir plastik lobutu (A) ve voleybol topunu (B) belirlenen yerlere taşımaktır. Robot, lobutun yerini doğru bir şekilde saptayabilmesine rağmen, voleybol topunu tanıyamamıştır. Görüntü işleme algoritması nesnelere yüzey normallerini göz önüne alarak arama yaptığından, hata olarak voleybol topunun sol kenarında bir küçük top tanımlanmıştır. Bu durum *tut* eyleminin parametrelerini etkilemektedir. Dolayısıyla, robot voleybol topunu tutucusu ile kavraması sırasında hata durumuyla karşılaşmaktadır. Hata durumu oluştuğundan sonra robot aynı eylemi iki kez daha denemektedir. Lobut doğru bir şekilde tanıdığı için robot, ilk denemesinde bu nesneyi kaldırmıştır.

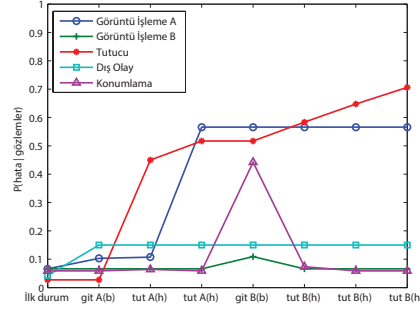
Şekil 2a'da eylem yürütülmesi sırasında beş adet hata tipi için olasılık değerleri görülmektedir. Şekilde görüldüğü gibi, *nesneye-git(A)* ve *tut(A)* eylemlerinin yürütülmesi sırasında herhangi bir hata belirtisi gözlemlenmemiştir. Dolayısıyla, bu zaman aralığında hata olasılıkları durağanlık göstermektedir. *konuma-git(hedef)* eylemi, daha karmaşık eylemler üzerinde hata tanısı yapabilmek için plana dahil olduğu halde yürütülme, doğrudan *birak(nesne)* eylemi yürütülmüştür. Grafiklerde, x eksenindeki eylemlerde parantez içinde verilen *b* ve *h*, sırasıyla ilgili eylemin *başarı* veya *hata* durumu ile sonlandığını belirtmektedir. *B* nesnesi için görüntü işleme hatası olasılığı, görüntü işleme algoritması ve bölütleme algoritması arasındaki nesnenin boyutları üzerindeki uyumsuzluk nedeniyle yükselmektedir. Bu tip bir uyumsuzluk olduğu durumlar için de robotun nesneyi tutucusu ile kavrayabilmesi mümkündür. Fakat verilen senaryoda bu durum geçerli değildir. Ayrıca grafikten görüldüğü üzere, *B* nesnesi için tutucu hatası olması olasılığı hatanın yayılması özelliği nedeniyle artmaktadır.

2) *Senaryo II*: Önerilen yöntemin tutucu arızalarını tespit edebilmesi için, robotun tutucusuna dışarıdan hata enjekte edilmiştir. Bu durum, robotun kontrolörü değiştirilerek tutucunun tamamen kapanmasını ve robotun tutucunun durum bilgisini almasını yazılımsak olarak önleyerek sağlanmıştır.

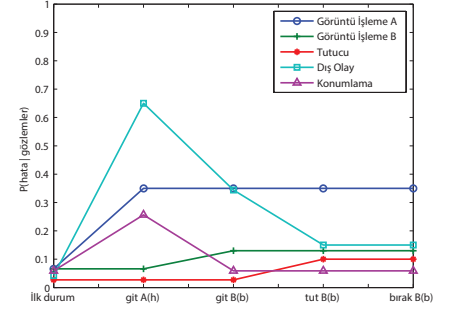
Bu senaryoda robot bir plastik lobutu (A) ve bir küçük topu (B) tutucusu ile tutmaya çalışmaktadır, fakat iki nesneyi de sürekli alamamaktadır. Şekil 2b'den de görülebileceği gibi her kaldırma eylemi denemesi sonrasında, robotun tutucu hatası inancı artmaktadır. Ayrıca, robotun A nesnesi üzerinde görüntü işleme hatası da, görüntü işleme algoritmasının aynı yerde iki farklı nesne saptaması nedeniyle artmıştır. Bunun yanında, *nesneye-git(B)* eyleminin yürütülmesi sırasında robotun konumlama hatası inancı, odometri ve özellik tabanlı konumlama algoritmasındaki fark nedeniyle artmıştır. Fakat ilerleyen zaman dilimlerinde bu durum düzelmiştir. Aynı eylem



(a) Bir nesne üzerindeki tutarlı bir görüntü işleme hatası, ilgili olasılıkta artışa neden olmaktadır.



(b) Sisteme enjekte edilen tutucu hatası, ilgili olasılık değerlerinde değişime neden olmaktadır.



(c) Ortamda dış etmenlere dayalı hata oluştuğunda, hesaplanan olasılıklar bu durumu açıklamaktadır.

Şekil 2: Her bir grafik, bütün hata durumları için eylem yürütme sırasında hesaplanan olasılık değerlerini göstermektedir.

sırasında  $B$  nesnesi üzerinde bir görüntü işleme hatası olasılığı da, görüntü işleme algoritmasının ilgili nesneyi şablonu ile düşük oranda benzetmesi nedeniyle artmıştır.

3) *Senaryo III*: Bu senaryoda, dış etkenlere bağlı oluşan hata durumları analiz edilmiştir. Bu amaçla, ortamdaki nesnelere biri robot planını yürütürken elle ortamdan alınmıştır. Bu senaryoda bir lobut ( $A$ ) ve bir küçük top ( $B$ ) kullanılmıştır. Robot lobutu almak üzere nesneye doğru ilerlerken lobut ortamdaki nesnelere ortamdan yok olduktan itibaren artmaktadır. Bölütleme algoritması da ayrıca bu hipotezi desteklemektedir. Fakat  $A$  nesnesi üzerinde bir görüntü işleme hatası olması da olası bir durumdur. Ayrıca, dış olay hata tipi temsilde her nesne için farklı bir model verilmemesinden bu hata tipinin olasılığı, diğer nesne ile ilgili eylemler yürütülürken, diğer nesne robotun görüş alanında olduğundan ve robotun kamerasıyla saptanabildiğinden azalmaktadır.

#### A. Bellek Kullanımı Analizi

Bu kısımda, önerilen yöntemin Tablo I’de verilen ilişkiler yardımıyla bellek kullanımını ne şekilde etkilediği analiz edilmiştir. Kullanılan ilişkiler yardımıyla HSM’ler üzerinde üretilen durum sayısı azaltılmıştır. İlişkilerin kullanıldığı ve kullanılmadığı senaryolardaki durum sayıları, Tablo III’te verilmiştir. Tablonun son sütunu yüzde olarak bellek kazanım oranını vermektedir. Tablodan da görüldüğü gibi, ilişkilerin kullanılması bellek kullanımını ortalama %37.7 azaltmaktadır.

### IV. SONUÇLAR

Bu çalışmada, zamansal ve hiyerarşik SMM’leri barındıran bir hata tanısı yöntemi önerilmiştir. Yöntemde, her bir hata tipi ayrı birer hiyerarşik SMM olarak temsil edilmiştir. Ayrıca bu temsil, birden fazla eş zamanlı oluşabilecek hataların nedenlerini açıklamaya olanak vermektedir. SMM’lerin hiyerarşik yapısı, bir nesne veya dış olay üzerinde oluşabilecek

Senaryo	İlişkiler Kullanılmadan ü.d.	İlişkiler Kullanılarak ü.d.	Kazanım
Senaryo I	242	146	39.66%
Senaryo II	258	173	32.94%
Senaryo III	153	91	40.52%

Tablo III: SMM Durum Üretimi Analizi

kalıcı hataların da tanınmasını sağlamaktadır. Eylem-hata tipi ilişkilerinin kullanılması ise durum uzayını daraltmış olup, yöntemin kullandığı bellek alanını azaltmıştır.

#### KAYNAKÇA

- [1] S. Karapinar, D. Altan, and S. Sarel-Talay, “A robust planning framework for cognitive robots,” in *Proceedings of the AAAI-12 Workshop on Cognitive Robotics (CogRob)*, 2012.
- [2] O. Pettersson, “Execution monitoring in robotics: A survey,” *Robotics and Autonomous Systems*, vol. 53, pp. 73–88, 2005.
- [3] J. M. Lee, S.-J. Kim, Y. Hwang, and C.-S. Song, “Diagnosis of mechanical fault signals using continuous hidden markov model,” *Journal of Sound and Vibration*, vol. 276, no. 3–5, pp. 1065 – 1080, 2004.
- [4] Z. Li, Z. Wu, Y. He, and C. Fulei, “Hidden markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery,” *Mechanical Systems and Signal Processing*, vol. 19, no. 2, pp. 329 – 339, 2005.
- [5] J. Flores-Quintanilla, R. Morales-Menendez, R. Ramirez-Mendoza, L. Garza-Castanon, and F. Cantu-Ortiz, “Towards a new fault diagnosis system for electric machines based on dynamic probabilistic models,” in *American Control Conference, 2005. Proceedings of the 2005*, 2005.
- [6] V. Verma, J. Fernandez, R. Simmons, and R. Chatila, “Probabilistic models for monitoring and fault diagnosis,” in *The Second IARP and IEEE/RAS Joint Workshop on Technical Challenges for Dependable Robots in Human Environments*, 2002.
- [7] G. Steinbauer and F. Wotawa, “Robust plan execution using model-based reasoning,” *Advanced Robotics*, vol. 23, no. 10, pp. 1315–1326, 2009.
- [8] S. Gspandl, S. Podesser, M. Reip, G. Steinbauer, and M. Wolfram, “A dependable perception-decision-execution cycle for autonomous robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2992–2998.
- [9] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. pp. 1554–1563, 1966.
- [10] D. Altan and S. Sarel-Talay, “Probabilistic failure isolation for cognitive robots,” in *Proceedings of the The Twenty-Seventh International FLAIRS Conference*, Pensacola Beach, Florida, USA, 2014.
- [11] S. Fine, Y. Singer, and N. Tishby, “The hierarchical hidden markov model: Analysis and applications,” *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.
- [12] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.
- [13] M. D. Ozturk, M. Ersen, M. Kapotoglu, C. Koc, S. Sarel-Talay, and H. Yalcin, “Scene interpretation for self-aware cognitive robots,” in *AAAI-14 Spring Symposium on Qualitative Representations for Robots*, 2014.