

ROBOTLAR VE PLANLAMA

1 Giriş

1959 yılında ilk endüstriyel robotun tanıtılmasından bu yana robotlar teknolojik toplumların vazgeçilmez araçları durumuna gelmişlerdir. Yeni kullanım alanları ile birlikte, robotların çok daha hassas, daha kararlı ve doğru hareket edebilmeleri hedeflenmiştir. Bu denli "zeki" robotların inşa edilebilmesi ise *robotik bilimindeki* gelişmelere bağlıdır. James S. Albus robotik bilimini "yapay zekayı mekanik araçların geribesleme kontrolü ile entegre etmeye çalışan bir sistem bilimi" olarak tanımlamaktadır. Bu alandaki çalışmalar mekanik ve donanım konularının yanında *örüntü tanıma, görüntü analizi, geometrik çıkarım, yazı ve konuşma anlama, fiziksel dünya modelleme, yol bulma, iş bölümü, planlama*, v.b. gibi alanlarda yoğunlaşmaktadır [21, 22, 1, 29, 14].

Dinamik olarak değişen çevre şartlarına uygun hareket etmek canlıların yaşamlarını sürdürebilmelerinde önemli bir rol oynar. Geleceği tahmin edebilmek ve amaçlarını gerçekleştirmek için hareket planları yapabilmek yüksek seviyeli canlıların karakteristik özelliklerindedir. Bu sebeple hareketler ve planlar üzerine çıkarım yapmak gerçek dünyada karşılaştıkları problemleri çözebilecek zeki robotların geliştirilmesinde en önemli faktörlerden birini oluşturmaktadır [6]. Yazımızda robot planlamasında karşılaşılan problemleri ve bunların felsefi yönlerini ele alacağız. Matematiksel mantığı bu alandaki çalışmalar için en uygun çatı olarak görmekteyiz [9]. Formel mantık, bilgi gösterimi, sağduyusal çıkarım gibi yapay zekanın öğeleri sayılabilecek yapılar bu doğrultudaki çalışmalara katkıda bulunmaktadır.

2 Hareketlerin ve Olayların Gösterimi

2.1 Durum Kalküsü

Gerçek dünya ortamlarında meydana gelen olaylar, bu olayların özellikleri ve aralarındaki bağlantıların gösterimi zeki bir robotun işlevlerini yerine getirmesi açısından büyük önem taşımaktadır. Bu sebeple olayların tamamlanması ve onlar hakkında çıkarım yapılabilmesi için formel bir dile ihtiyaç vardır.

McCarthy bu probleme artık klasik bir yaklaşım haline gelmiş formel bir *durum kalküsü (situation calculus)* önermiştir [24]. Bu bölümde durumların, olayların ve bağlantıların durum kalkülüsünde gösterimi için kullanılan mantıksal terimleri tanımlayacağız. Örneğin, *üzerinekoy* (A, B) terimi A bloğunu B bloğunun üzerine koyma hare-

Varol Akman, Erkan Tın()*

()Bilgisayar ve Enformatik Mühendisliği Bölümü, Bilkent Üniversitesi.*

ketinin gösterimi için kullanılabilir. Benzer olarak, bağıntı terimi üzerinde (A, B), A bloğunun B bloğu üzerinde olduğunu belirtir. Bileşik olayların ve daha karmaşık bağıntıların gösterimi de mümkündür.

Durum kalkülüsünde *yüklem* (*predicates*) ise esas olarak bağıntıların özel durumlardaki değerleri hakkında önermelerde bulunmak amacı ile kullanılırlar. Örneğin, *tutarlı* (*b, s*) terimi *b* bağıntısının *s* durumunda *doğru* olduğu anlamına gelmektedir. Ya da *tutarlı* (*üzerinde* (*A, B*), *s*) terimi ancak *üzerinde* (*A, B*) bağıntısının *s* durumunda *doğru* olması halinde *doğru* olacaktır. Bağıntılar daha başka yüklem ve fonksiyon sembolleri ile de gösterilebilir [23, 24, 27].

Problem alanında olan herhangi bir olayla ilgili durum geçişlerinin tanımlanması gerekmektedir. Bunun için *e* olayının *s* durumunda oluşmasından ortaya çıkan yeni durumu gösteren *sonuç* (*e, s*) terimi kulla-

nılabilir. Örneğin, *sonuç* (*üzerine koy*(*A, B*), *s*) robotun *s* durumunda *A* bloğunu *S* bloğu üzerine koymasının sonucunda ortaya çıkan yeni durumu gösterir. Bu *sonuç* fonksiyonu bir robotun verilen bir durumdan ulaşabileceği durumları karakterize etmek için kullanılabilir.

Durum kalkülüsünün formüllerinde mantıksal bağlaçlar (*A, V, o*) ve niceleyiciler (*V, 3*) kullanılarak robotun hareketlerinin etkileri ve olaylar hakkında çıkarım yapılabiliriz. *A* bloğunu *B* bloğu üzerine koyma hareketinin sonucunu aşağıdaki aksiyom ile ifade edebiliriz:

$V s [tutarlı(üstüboş(A)s) A tutarlı (üstüboş(B), s)^$

$tutari(üzerinde(A, B), sonuçüzerine koy(A, B),s)]$

Bu aksiyomla, *A* ve *B* bloklarının üstleri başlangıçta boş ise *üzerine koy* (*A, B*) hareketinin sonucunda *A* bloğunun β bloğu üzerinde olacağı söylenmektedir.

Dinamik alanlarda bilgi gösterimi ve çıkarım yapılabilmesi için değişik mantıksal yöntemler geliştirilmiştir. Bunlardan en çok bilineni *modal mantık* (*modal logic*) [19]. Bu mantıkta durumları gösteren terimlerin açıkça kullanımından kaçınılır. Modal mantığın zaman boyutunu ele alan bir türü ise *temporal mantık* (*temporal logic*) olarak bilinir [31]. Örneğin, gelecekte yağmur yağacak önerisi *O yağmur yağacak* formülü ile anlatılır. (*O* *0* formülü gelecekte *0* önermesinin doğru olacağı bir durumun var olduğunu söyler.) Bu gibi zamansal işlemciler eldeki uygulama alanlarının özelliklerinin tanımlanması için doğal araçlar olarak görülmektedirler.

2.2 STRIPS

Durum kalkülüsüne alternatif olarak Fikes ve Nilsson hareketler için *STRIPS gösterimini* (*representation*) önermişlerdir [10]. Bu gösterimin amacı durum kalkülüsünü kullanarak plan yapmada karşılaşılan zorlukları ortadan kaldırmaktır. STRIPS gösteriminde bir dünya durumu, bir mantıksal formüller kümesi ve bu kümedeki formüllerin birleşmeleri ile ifade edilmektedir. Hareketler ve olaylar ise *operatörler* ile gösterilir. Bir *operatör önşart* (*precondition*), *ekleme listesi* (*addlist*) ve *silme listesi* (*deletelist*) kısımlarından oluşmaktadır. Verilen bir *s* durumunun tanımlanmasında, önşart, hareketin bu *s* durumunda yapılabileceğini belirtir. Ekleme ve silme listeleri ise hareketin *s* durumunda gerçe-



kleşmesi halinde ortaya çıkan yeni durumun gösteriminin nasıl elde edileceğini belirtir. Bir sonraki durumun belirlenmesini sağlayan bu şemaya *STRIPS kuralı (rule)* adı verilir.

Aşağıdaki STRIPS operatörü bir robotun *A* bloğunu *0* pozisyonundan *1* pozisyonuna kaydırması hareketinin gösterimine bir örnek olarak verilebilir:

Önşart: *pozisyon (A, 0) A üzeriboş (A)*

Ekleme listesi: *{pozisyon (A, 1)}*

Silme listesi: *{pozisyon (A, 0)}*

Başlangıçta *A* bloğunu kırmızı renk, üzeri boş ve *0* pozisyonunda duran bir blok olarak varsayarsak, bu durum *{kırmızı(A), üzeriboş (A), pozisyon(A, 0)}* formüller kümesi ile ifade edilebilir. Bu küme önşart formülünü doğrular ve operatör uygulanabilir. Operatörün uygulanması sonucundaki durum, başlangıç durumunu gösteren kümeden *pozisyon (A, 0)* çıkarılarak *pozisyon (A, 1)* eklenmesi ile elde edilir: *{kırmızı (A), üzeriboş (A), pozisyon (A, 1)}*. Sonuç durumunda, silme listesindeki formüller her zaman yanlış olarak değerlendirilmemelidirler. Aksi ispatlanmadıkça bu formüllerin doğruluk değerleri "bilinmiyor" şeklinde yorumlanır. Oldukça yaygın olarak kullanılmasına rağmen STRIPS gösterimi durum kalkülüsü kadar ifade gücüne sahip değildir [40]. Özellikle, STRIPS gösteriminin bir hareketin etkileyebileceği bütün formüllerin, hareketi tanımlayan operatörün silme listesine yazılmasını gerektirmesi, onu gösterim gücü ve genelleme açısından zayıf kılmaktadır [32].

3 Plan Sentezi

Plan sentezi sınırlı bir alan içerisinde işlemlerini sürdüren robotların daha önceden saptanmış amaçları gerçekleştirmeleri için gerekli hareketlerin planlanmasıdır. Sentezin başarısında robotların karşılaştığı risklerin ve onların çıkarım yapmalarını zorlaştıran çevresel koşulların aşılması önemli rol oynar. Planların başarısız olması durumunda yapılacak stratejik hareketleri ve bununla ilgili akıl yürütme, v.b. gibi problemleri yazımızın konusu dışında tutacağız-

3.1 Planların Kullanımı

Bir planın en gerekli bölümünü, robota uygulaması için verildiğinde başarılı bir davranış göstermesini sağlayan kısım oluşturur. Örneğin, bir fincanı hareket ettirmesi için yazılan program, robotun mutfaktaki çalışma planının özel bir parçasıdır. So-

nuçta ne tür bir davranışın ortaya çıkacağı ise planın gerçekleştirildiği ortama ve bunu yapan robota bağlıdır. Birden fazla robotun bulunduğu ortamlarda her robotun kendine ait planları olabileceği gibi işbirliği içinde çalışmaları için tek bir planları da olabilir. Bu tür planları *çok robotlu planlar* olarak adlandıracağız.

Bir plan başarı ile uygulanabileceği gibi bazı kısımları başarısızlıkla sonuçlanabilir. Bu durumlarda robotların alternatif planları ve bunların sonuçlarını göz önüne almaları zorunludur [13]. Planlar genelde belirli, basit temel yapılardan oluşan daha büyük yapılardır. Bir planın yaratılmasında *zincirleme (sequential)*, *şartlı (choice)*, *iteratif (iterative)* ve *yineleyen (recursive)* planlama gibi standart yöntemler kullanılır. Bileşik planları oluşturan *altplanların* bölünemeyen en ufak yapılarına *atomik veya basit planlar* adı verilir. Planlar sadece atomik planlardan oluşmazlar, iki veya daha fazla plan biraraya gelerek daha büyük planları meydana getirebilirler. Örneğin, bir alış-veriş planı markete gitme, bisküvi ve meyva suyu alma ve geri dönme altplanlarından oluşabilir. Plandaki her adım detaylı olarak tanımlanmalıdır, fakat bu adımların sırası belirtilmeyebilir. Bu tür planlara *kısmi planlar adı* verilir.

Planlar üzerinde akıl yürütebilmek için kullanılan formel dile *plan terimleri* eklenmesi gerekir. Plan terimleri bir planın ifade edilmesini ve planın oluşturulması için gereken operatörlerin uygulanmasını sağlarlar.

Daha önce de belirttiğimiz gibi bir planın en önemli özelliği ortaya çıkardığı başarılı davranışlardır. Aynı anda birden fazla hareketin yapılamadığı durumlarda, bir plan başlangıç durumunun bitiş durumuna dönüştürülmesi olarak, başka bir deyişle bir seri *transformasyon* olarak düşünülebilir. Şimdi, *p* planının *M* robotu tarafından s_1 durumunda uygulanarak s_2 durumuna neden olmasını *üretir (M, p, Sr, sz)* ile gösterelim. Bu şekilde *p* planının ve bu planı oluşturan operatörlerin uygulanmasının etkilerini tanımlamak mümkün olabilmektedir. Örneğin, $s >$ durumunda *0* formülü doğru iken *M* robotu *p* planını uygularsa sonuçta oluşan s_2 durumunda formülünün doğru olacağını aşağıdaki aksiyom şeması ile anlatılabilir:

Vs_1Vs_2 [Tutarlı ($0, S_1$) *A* üretir (M, p, s_1, s_2) tutarlı (y, s_2)].

Görüldüğü gibi planlar *hareket (action)* değildirler. Planların hareketlerle olan ilişkileri *üret/r* gibi önermeler ile ifade edilir. Bu

"Bir fincanı hareket ettirmesi için yazılan program, robotun mutfaktaki çalışma planının özel bir parçasıdır. Sonuçta ne tür bir davranışın ortaya çıkacağı ise planın gerçekleştirildiği ortama ve bunu yapan robota bağlıdır."



Bir robotun mutfak masası etrafında bir kez dolanması olayı mutfaktaki diğer robotların davranışlarını ve hatta etrafta uçuşan bir kuşun hareketlerini de içermelidir.

doğrultuda *süreç mantığı (process logic)* ve dinamik *mantık (dynamic logic)* bu ilişkilerin daha üst seviyede formel olarak tanımlanmasında kullanılabilirler [28, 17].

3.2 Genel Tümevarım Yöntemleri

Verilen hareketlerin ve formülasyonların ışığında, planlamaya en basit yaklaşım amaca ulaşmayı sağlayacak bir dizi hareketin varlığını ispatlamaktır. Ulaşılması gereken amacı 0 formülü ile gösterelim ve başlangıç durumunun (diyelim s) y formülünü doğruladığını varsayalım. Bu başlangıç durumundan ulaşılabilir ve 0 şartını sağlayacak bir z durumunun varlığını göstermek için aşağıdaki teoremi ispatlamak yeterli olacaktır:

Vs [tutarlı (y, s) z] z [tutarlı(0, z) A ulaşılabilir (z, s)].

Örneğin, B bloğunun A bloğu ve C bloğunun B bloğu üstünde olduğu bir başlangıç durumu verildiğini düşünelim. A bloğunun üzerinin boşaltılması için gerekli plan aşağıdaki teoremi ispatlayarak elde edilebilir:

Vs [tutarlı(üzerinde(C, B), s) A tutarlı (üzerinde(B, A), s) z]

$3z$ [tutarlı(üzeriboş(A), z) A ulaşılabilir (z, s)]-

Bu teoremin ispatı bize, A bloğunun üzerinin boşaltılmasının ilk önce C bloğunun, daha sonra B bloğunun indirilerek masaya konması ile olacağını söyleyebilir:

Vs [tutarlı(üzerinde(C, B), s) tutarlı (üzerinde(B, A), s)]^

tutarlı (üzeriboş(A), sonuç(üzerine koy(B, masa), sonuç (üzerine koy (C, masa), s)))]].

Green bu düşünceyi ilk uygulayan kişidir [16]. Daha sonraki araştırmacılar tümevarım ve teorem ispatlama yöntemlerini *şartlı ve yineleyen* planlama metodlarında karşılaşılan problemleri çözmek için kullanmış ve geliştirmişlerdir [23].

4. Çok Robotlu Alanlar

Gerçek dünya bir robotun kontrolü dışında bir çok süreci içerir. Dahası, birden fazla robotun çalıştığı alanlarda, robotlar gerek bağımsız gerekse işbirliği içerisinde hareket etmek zorundadırlar. Böyle bir ortamda tek bir robot planlayıcısı yeterli değildir. Robotun kendi kontrolü dışındaki hareketler hakkında akıl yürütmesi adeta imkansızlaşır. Bu nedenle yeni hareket ve planlama modellerinin geliştirilmesi gerekir. Bu bir hareketin diğerine olan müdahalesi, iki olay veya hareketin aynı anda olması, beraber plan yapma, rekabet durumunda plan yapma, vb. gibi konuların gözönüne alınmasını zorunlu kılmaktadır. Bunlara ek olarak, diğer robotların düşünceleri hakkında akıl yürütebilecek ve onların davranışlarını gözlemleyerek onların amaç ve eğilimlerini anlayabilecek zeki sistemlerin geliştirilmesi gerekmektedir.

4.1 Hareketin Gösterimi

Tek bir robotun işlevini sürdürdüğü alanlar için önerilen yapılar, formalizmler ve basitleştirici varsayımlar, çok robotlu alanlarda yeterli değildirler. En azından, hareketleri durumlar arasında bağıntılar olarak görmek imkansızlaşır. Bunun nedeni aynı anda (birden fazla robot tarafından) yapılan hareketlerin etkilerinin bu hareketlerin oluş sürecinde neler meydana geldiğine bağlı olmasıdır [11, 30, 37]

Daha önce olduğu gibi, bir olayı onun bütün olası durumlarda meydana gelişini gösteren bir durumlar zinciri kümesi olarak tanımlamak mümkündür [4, 14, 15, 30, 26]. Tek robotlu alanlardan farklı olarak, verilen bir olayın aynı anda veya zamanda meydana gelen diğer olayları da kapsamaması gereklidir. Örneğin, bir robotun mutfak masası etrafında bir kez dolanması olayı mutfaktaki diğer robotların davranışlarını ve hatta etrafta uçuşan bir kuşun hareketlerini de içermelidir. Bu doğrultuda hareketlerin gösterimi ve çıkarım yapılabilmesi için yeni formel sistemler geliştirilmektedir.

Durum kalkülüsünün genişletilmesi bu tür yaklaşımlardan sadece biridir [14, 15].

McDermott konuya değişik bir formalizm önererek yaklaşmıştır [26]. Bu sistemde verilen her iki durum arasında mutlaka farklı bir durumun varlığı kabul edilir ve bununla *sürekli (continuous)* süreçler üzerinde çıkarım yapabilmek amaçlanır. Benzer bir yaklaşım ise Ailen ve Pelavin tarafından izlenmiştir [4, 30]. McDermott'un bir önermeyi durumlar üzerinden bir fonksiyon olarak görmesine karşın, Ailen ve Pelavin bir önermeyi *durum aralıkları (Intervals)* üzerinde bir fonksiyon olarak almaktadırlar. Örneğin, *tutarlı (üzeriboş (A), At)* bu formalizmde A bloğunun At zaman aralığında üzerinin boş olduğunu gösterir. Bu şekildeki aralıklı gösterim, durumsal gösterimden daha karmaşıktır. Bir durum, ya diğerinden önce ya da sonra zaman üzerinde sıralanmıştır. Fakat bir zaman aralığı değişik şekillerde diğer bir zaman aralığı ile karşılaşabilir, onu kapsayabilir ve onunla kesişebilir. *Aralık kalkülüsü (interval calculus)* adını verdiğimiz bu formalizmin araştırmacılar tarafından halen üzerinde çalışılmaktadır [3, 39, 30].

4.2 Nedensellik

Hareketlerin sonuçlarını ifade etmek için yazılan aksiyomlar genelde karmaşıktır, ö bloğunun pozisyonunun A bloğunun pozisyonundan bağımsız olduğunu söyleyebilmek, ffnin A'nın önünde, Anın üzerinde, Anın üzerinde fakat kapıya bağlı olup olmadığı gibi bir çok şarta bağlıdır. *Nedensellik (causality)* nosyonu bu kaotik durumu çözmek için kullanılabilir. Bir olayın aynı anda bir diğerine neden olması ve bir olayın ondan sonra meydana gelen başka bir olaya neden olması nedenselliğin modellenmesinde kullanılan iki tür yaklaşımdır. Bunlar sürekli olay ve süreçlerden ziyade, ayırık olay ve süreçlerin modellenmesinde uygulanırlar.

Bir örnek verecek olursak, "ne zaman bir blok hareket ettirilirse üzerinde bulunan blok da (engellenmedikçe) hareket eder" cümlesi nedensel bir kural olarak alınabilir. Burada nedenselliğe sadece olayların meydana geldiği ortama bağlı atomik olaylar arasındaki bir ilişki olarak bakılmaktadır. Halbuki nedensellikte olayların zaman bo-



yutu da önemli bir rol oynar. Bir olayın kendisinden önce meydana gelen başka bir olaya neden olamayacağı ve daima kendisinden sonra meydana gelen olaylar üzerinde etkisi olduğu nedensel kuramların başlıca varsayımları arasındadır. Yapay zeka literatüründe nedenselliği değişik yönleri ile ele alan bir çok çalışma bulmak mümkündür [4, 26, 34, 38, 36, 2].

4.3 Çok Robotlu Planlama

Çok robotlu alanlarda başarılı çıkarımlar yapabilmek için bir çok formal sistem geliştirilmesine karşın, tamamlanmış ve uygulanmaya konmuş planlama sistemlerinin sayısı oldukça azdır. Ailen ve Koomen [5] tarafından tanımlan ve aralık mantığı üzerine kurulmuş planlayıcı en dikkate değer sistemdir. Ne var ki bu tür sistemlerin gerçek dünyanın kompleks alanlarında ne kadar başarılı olabilecekleri tartışma konusudur. Bu konuda çözülmesi hedeflenen problemler arasında, farklı planlara sahip robotların haberleşmeleri ve koordine edilmiş bir biçimde, birbirlerini engellemeden işbirliği içinde çalışmalarını bulunmaktadır.

Yapılan çalışmalar, her robotun kendi planındaki hangi hareketlerin diğer robotların hareketleri ile çakışabileceğinin bulunması ve koordine edilmiş planlarda bu çakışma-



ların önlenmesi yönünde yoğunlaşmıştır [11, 12, 35]. Bazı araştırmacılar robotların riskli durumlarda sağlıklı çıkarımlar yaparak nasıl işbirliği içerisinde çalışabileceklerini incelemişlerdir [33]. Bir kısmı ise asenkron çalışan robotların koordinasyonu ve kullanılması gereken organizasyon stratejilerini araştırmışlardır [8]. Nedensel bağımsızlıkları kullanan bir planlayıcı Lanksy tarafından geliştirilmiştir [20].

5 Felsefi Problemler

5.1 Çerçeve Problemi

Robotların hareketleri ve planları hakkında çıkarım yapabilmek temelde bazı felsefi sorunları beraberinde getirmektedir. Bunların ilki *çerçeve problemi* (*frame problem*) olarak bilinmektedir [24]. Bir masanın üzerinde biri kırmızı diğeri mavi iki blok bulunduğunu düşünelim. Bu ortamda robotumuzun kırmızı bloğu mavi bloğun üzerine

koyduğunu varsayalım. Yeni durumda blokların renklerinin ne olduğu sorusuna doğal olarak "değişmediler" cevabını veririz. Aynı cevabı robotumuzun verebilmesi için yaptığı bir hareketin neleri değiştirebileceğinin yanısıra neleri değiştirmeyeceğini de bilmesi gerekmektedir. Bu da robotun her hareketinin içinde bulunduğu dünyanın hangi özelliklerini değiştirmedeğini öneren aksiyomlar (*çerçeve aksiyomları*) yazmak anlamına gelir [24]. Gerçek dünya ortamları için yazılması gereken aksiyomların sayısı düşünülecek olursa problemin boyutu açıklık kazanır. Başka bir deyişle, sorun bütün çerçeve aksiyomlarını her özellik-hareket çifti için açıkça yaz-

mak ve bunlarla çıkarım yapmanın zorluğundan kaynaklanmaktadır. Önerilen çözümler kısıtlı dünyalarda vasayımların formal hale getirilmesinden öteye gidememiştir. Bazı araştırmacılar problemi genelde bir *kombinatorik* (*combinatorics*) sorunu [24, 32], diğerleri ise *tamamlanmamış bilgi* (*incomplete knowledge*) ile çıkarım yapma problemi olarak görmektedirler [26].

5.2 Kalifiye Olma Problemi

McCarthy tarafından gözlemlenen diğer bir felsefi problem ise *kalifiye olma* problemi (*qualification problem*) [25]. Bu problem, bir hareketin etkisinin spesifikasyonunun kalifiye olma durumuna bağlı olmasından kaynaklanmaktadır. Bu da gerek hareketin ortamda yarattığı değişikliklerin gerekse hareketin değiştirmediklerinin aksiyomlarla ifade edilmesinde zorluklar yaratmaktadır [25, 34, 26]. Tekrar blok örneğimize dönelim. Robotun kırmızı bloğu bulunduğu yerden kaldırarak mavi bloğun üzerine koyabilmesi için bir dizi hareket yapması gerekmektedir. Zeki bir robotun kırmızı bloğu hareket ettirebilmesi için bir çok durumun sağlanıp sağlanmadığından emin olması gerekir. Örneğin, bloğun masaya yapışık olmaması, üzerinde başka nesnelerin olmaması, bloğun robotun mekanik gücü ile kaldırılabilecek ağırlığa ve yapıya sahip olması, bloğun bir iple herhangi bir yere bağlı olmaması, v.b. Ayrıca robotun yaptığı her hareketin ne ile sonuçlanabileceğini önceden tahmin edebilmesi gerekir. Burada kalifiye olma sorunu ortaya çıkmaktadır. Başarılı bir dizi hareketin sonunda kırmızı bloğun mavi blok üzerinde olması gerekir. Fakat, kırmızı blok bir yay ile masadaki eski yerine bağlıysa, mavi blok kırmızı bloğu üzerinde tutabilecek nitelikte değilse (örneğin, küre şeklinde ise), kırmızı blok bırakıldığı anda yönü oradan geçen bir başka robotun çarpması ile değiştirilmişse, v.b. bu yönde bir çıkarım yanlış olacaktır.

6. Sonuç

Matematiksel mantık, ispat kuramı ve çeşitli mantık dizgeleri zeki robotların tasarımında karşılaşılan felsefi problemlerin çözümüne temel katkılarda bulunabilecek doğal alanlar olarak görülmektedir. Robotların (yukarıda sözünü ettiğimiz biz insanların kontrolü altındaki fonksiyonları dışında) günlük yaşamın fiziğine ait sağduyuya sahip olmaları da gerekmektedir. "Su dolu bir kap ateşe yerleştirilir ve yeterince ısıtılsa su kaynar" veya "bir iple bir şeyi çekebilir ama itemezsiniz" gibi herkesçe bilinen şeylerin robotlara öğretilmesi *naif fizik* (*naive physics*) [18] ve *nitel fizik* (*qualitative physics*) [7] diye adlandırdığımız çalışmaların esas amaçlarını oluşturur, inanıyoruz ki, bir çok problemin temelinde yatan felsefi sorunların çözümü ifade gücü yüksek formal sistemlerin geliştirilmesine bağlıdır ve bu yönde yapılacak çalışmalar daha yetenekli robotların inşa edilmesine öncülük edeceklerdir.

KAYNAKLAR

- [1] Akman, V., ten Hagen, P.J.W. The power of physical representations, *AI Magazine*, 10: 49-65, 1989.
- [2] Akman, V., Tın, E. What is in a context.? *Proceedings of 1990 Bilkent International Conference in Communication, Control, and Signal Processing*, Vol. 2, E. Arkan, ed., Elsevier, pp. 1670, 1676, 1990.
- [3] Ailen, J.F. Maintaining knowledge about temporal intervals, *Communications of the ACM*, 26: 832, 843, 1982.
- [4] Ailen, J.F. Towards a general theory of action and time, *Artificial Intelligence*, 23: 123-154, 1984.
- [5] Ailen, J.F., Koomen, J.A. Planning using a temporal world model, *Proceedings of Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 741, 747, 1983.
- [6] Ailen, J., Hendler, J., Tate, A., eds. *Readings in Planning*, San Mateo, Calif.: Morgan Kaufmann, 1990.
- [7] Bobrow, D.G., ed *Qualitative Reasoning about Physical Systems*, Cambridge, Mass.: MIT Press, 1985.
- [8] Corkill, D.D., Lesser, V.R. The use of meta-level control for coordination in a distributed problem solving network, *Proceedings of Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 748-756, 1983.
- [9] Curry, H. *Foundations of Mathematical Logic*, New York: Dover, 1977.
- [10] Fikes, R.E., Nilsson, N. J. STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence*, 2: 189-208, 1971.
- [11] Georgeff, M.P. Communication and interaction in multiagent planning, *Proceedings of Third National Conference on Artificial Intelligence*, Washington, D.C., pp. 125-129, 1983.
- [12] Georgeff, M.P. A theory of action for multiagent planning, *Proceedings of Fourth National Conference on Artificial Intelligence*, Austin, Texas pp. 121-125, 1984.
- [13] Georgeff, M. P., Lansky, A. L. Procedural knowledge, *Proceedings of the IEEE Special Issue on Knowledge Representation*, 74: 1383-1398, 1986.
- [14] Georgeff, M. P. Actions, processes, and causality, *Reasoning about Actions and Plans: Proceedings of 1986 Workshop*, Los Altos, Calif.: Morgan Kaufmann, pp. 99-122, 1987.
- [15] Georgeff, M. P. Many agents are better than one. *The Frame Problem in Artificial Intelligence: Proceedings of 1987 Workshop*, Los Altos, Calif.: Morgan Kaufmann, pp. 59-75, 1987.
- [16] Green, C. C. Application of theorem proving to problem solving, *Proceedings of First International Joint Conference on Artificial Intelligence*, Washington, D.C., pp. 219-239, 1969.
- [17] Harel, D. *First Order Dynamic Logic*, Lecture Notes in Computer Science, No. 68, Springer-Verlag, 1979.
- [18] Hayes, P.J. *The second naive physics manifesto*, *Formal Theories of the Commonsense World*, J. Hobbs and R.C. Moore, eds., New Jersey: Ablex, pp. 1-36, 1985.
- [19] Hughes, G.E., Cresswell, M.J. *An Introduction to Modal Logic*, New York: Routledge, 1989.
- [20] Lansky, A. L., Fogelson, D.S. Localized representation and planning methods for parallel domains, *Proceedings of Sixth National Conference on Artificial Intelligence*, Seattle, Washington, pp. 240, 245, 1987.
- [21] Loranzo-Perez. T. Robotics, *Artificial Intelligence*, 19: 137, 143, 1982.
- [22] Loranzo-Perez. T. Spatial planning: a configuration space approach, *IEEE Transactions on Computers*, 32: 108, 120, 1983.
- [23] Manna, Z., Waldinger, R.J.A. *Theory of Plans, Reasoning about Actions and Plans: Proceedings of 1986 Workshop*, Los Altos, Calif.: Morgan Kaufmann, pp. 11-45, 1987.
- [24] McCarthy, J., Hayes, P.J. Some philosophical problems from the standpoint of artificial intelligence, *Machine Intelligence*, 4: 463, 502, 1969.
- [25] McCarthy, J. Circumscription a form of nonmonotonic reasoning, *Artificial Intelligence*, 13: 27-39, 1980.
- [26] McDermott, D.A. Temporal logic for reasoning about processes and plans, *Cognitive Science*, 6: 101-155, 1982.
- [27] Montague, R. Deterministic theories, Formal Philosophy: *Selected Papers of Richard Montague*, R.H. Thomason, ed., New Haven, Conn.: Yale University Press, 1974.
- [28] Nishimura, H. Descriptively complete process logic, *Açta Informatica*, 14: 359-369, 1980.
- [29] Paul, R.P. Robots, models, and automation, *Computer*, 12: 19-27, 1979.
- [30] Pelavin, R., Ailen, J.F. A formal logic of plans in a temporally rich domain. *Proceedings of the IEEE Special Issue on Knowledge Representation*, 74: 1364-1382, 1986.
- [31] Prior, A. N. Past, *Present and Future*, Oxford: Clarendon Press, 1967.
- [32] Reiter, R.A. Logic for default reasoning. *Artificial Intelligence*, 13: 81-132, 1980.
- [33] Rosenschein, J.S. *Rational Interaction: Cooperation Among Intelligent Agents*, PhD thesis, Stanford University, Stanford, Calif., 1986.
- [34] Shoham, Y. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*, Cambridge, Mass.: MIT Press, 1988.
- [35] Stuart, C.J. *Synchronization of Multiagent Plans Using a Temporal Logic Theorem Prover*, Technical Note 350, SRI International, Menlo Park, Calif.: Artificial Intelligence Center, 1985.
- [36] Tın, E. Nedensel formal sistemlerle çıkarım, *Bilkent Üniversitesi Elektrik-Elektronik ve Bilgisayar Mühendisliği Konferansı Bildiriler Kitabı*, Ankara, pp. 355-358, 1991.
- [37] Tın, E., Akman, V. Representing concurrent actions, *Proceedings of Sixth International Symposium on Computer and Information Sciences*, Vol. 1, M. Baray and B. Özgüç, eds., Elsevier, pp. 641-650, 1991.
- [38] Tın, E., Akman, V. Computing with causal theories, *International Journal of Pattern Recognition and Artificial Intelligence*, çıkacak.
- [39] van Benthem, J. Tense logic and time, *Nötre Dame Journal of Formal Logic*, 25: 1-16, 1984.
- [40] Waldinger, R. Achieving several goals simultaneously, *Machine Intelligence*, 8: 94, 136, 1977.