YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAREKET TEMELLİ YAPAY YAŞAM FORMLARININ FARKLI ÖĞRENME YÖNTEMLERİ KULLANILARAK UYARLANABİLMEDEKİ BAŞARILARININ ÖLÇÜLMESİ

Bilgisayar Müh. Ekin Su UĞURLU

FBE Bilgisayar Mühendisliği Anabilim Dalında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. A. Coşkun SÖNMEZ

İSTANBUL, 2007

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

σ	etkinleştirme fonksiyonu
w	Yapay Sinir Ağı'nın ağırlıkları
η	öğrenme katsayısı
γ	indirim oranı
d	içinde bulunulan durum
h	yapılan hareket
r	erişilmesi istenen ödül
τ	yaşam formunun eklemlerine uygulanan tork değeri
π	politika parametre vektörü
R_i	hareket politikası
ε	fark değeri

KISALTMA LİSTESİ

YZ	Yapay Zeka
Y-Yaşam	Yapay Yaşam
GA	Genetik Algoritma
YSA	Yapay Sinir Ağı
İBSA	İleri Beslemeli Sinir Ağı
GYÖ	Geri Yayılımlı Öğrenme
YYSA	Yinelenen Yapay Sinir Ağı
GZYÖ	Gerçek Zamanlı Yinelenen Öğrenme
PGDÖ	Politika Gradyanlı Destekli Öğrenme

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖNSÖZ

Tez danışmanım Prof. Dr. A. Coşkun Sönmez' e tez çalışmam boyunca bilgi ve deneyimlerini aktardığından ve yardımlarından dolayı teşekkür ederim.

Tez çalışması süresince manevi desteğini eksik etmemiş olan değerli hocam Prof. Dr. Oya Kalıpsız' a teşekkür ederim.

Masa komşum iş arkadaşım Dr. Ömer Özgür Bozkurt'a sorularıma büyük bir sabırla cevap verdiğinden ve yardımını hiç esirgemediğinden dolayı teşekkür ederim.

Her zaman tüm konularda desteğini veren ve bunu sonuna kadar hissettiren, moralimin azalmasına bir an olsun izin vermeyen nişanlım Göksel Biricik' e çok teşekkür ederim.

Sevgili ailem, annem ve babama bütün hayatım boyunca bana güvenip arkamda olduklarından, sonsuz sevgilerini ve desteklerini eksik etmediklerinden, başka kaygılar taşımadan çalışmalarıma odaklanabildiğim bir ortam sunduklarından, şu anda olduğum kişi olmamı ve yaptığım işleri yapmamı sağladıklarından dolayı çok teşekkür ederim.

ÖZET

Hareket etme problemi öğrenme yöntemleri üzerinde çalışmak için çok elverişli bir problemdir. Bu durum hareket probleminin zor bir öğrenme probleminin bütün parçalarını içermesinden kaynaklanmaktadır. Bu bilgiden yola çıkarak tezde eklemli yapıya sahip bir yapay yaşam formunun hareketlerini kontrol edebilmeyi öğrenmesini sağlayacak çeşitli öğrenme yöntemlerinin uygulanması ve karşılaştırılması gerçekleştirilmiştir. Öğrenme doğanın fizik kurallarının benzetildiği gerçekçi bir yapay ortamda sağlanmıştır.

Hareket etme, uyarlanabilirlik isteyen bir problemdir. Bu nedenle hareket kontrol mekanizmasının önceden belirlenmiş veya dış kaynaklı olmasındansa nasıl hareket etmesi gerektiğini yapay yaşam formunun kendi kendine öğrenmesi daha tercih edilebilir bir durumdur. Dolayısıyla yapay yaşam formunun hareket etmeyi öğrenmesini gerçekleştirebilmek amacıyla organizmaların işleyiş şekillerinden etkilenerek oluşturulan üç öğrenme alanından faydalanılmıştır.

Temel alınan alanlar evrimsel öğrenme, denetimli öğrenme ve destekli öğrenmedir. Evrimsel öğrenmeyi gerçekleştirmek için Genetik Algoritmalar; denetimli öğrenmeyi gerçekleştirmek için Yapay Sinir Ağlarının eğitilmesine yarayan Geri Yayılımlı Öğrenme ve Yinelenen Yapay Sinir Ağlarının eğitilmesinde kullanılan Gerçek Zamanlı Yinelenen Öğrenme; destekli öğrenmeyi sağlamak için ise Q-Öğrenme ve Politika Gradyanlı Destekli Öğrenme yöntemlerinden faydalanılmıştır.

Genetik Algoritma ile öğrenmenin gerçekleşmesi diğer yöntemlere göre çok yavaş olmuştur; öğrenme sonucunda da belirgin ve emin adımlarla ilerlediği görünümünü sağlayan yavaş adımlar gözlemlenmiştir. Denetimli öğrenmeyi gerçekleştirmek amacı ile kullanılan yöntemler arasında Gerçek Zamanlı Yinelenen Öğrenme daha iyi sonuçlar vermiştir. Bu yöntemle öğrenme çabuk gerçekleşmiştir ve yaşam formu çok hızlı adımlarla ilerlemiştir fakat çoğunlukla ilerlediği doğrultuda sapmalar yaşanmıştır. Destekli öğrenme için kullanılan yöntemlerden Q-Öğrenme problem uzayının boyutuna ayak uyduramadığından başarısız sonuçlar vermiştir. Politika Gradyanlı Destekli Öğrenme ile eğitilen yapay yaşam formları ise büyük ve belirgin adımlar ile ilerlemişlerdir. Öğrenme çabuk gerçekleşmiştir, bu öğrenme yöntemi ile eğitilen yapay yaşam formlarının ilerleme hızı yavaştır fakat hareketler düzgündür, genelde sapma göstermemişlerdir.

Sonuç olarak üç öğrenme alanının kullanımından da görsel olarak belli bir gerçekçilik eşiğini tutturan hareketler elde edilmiştir. Yöntemlerin sonuçlarındaki farklılık daha çok öğrenme hızı, hareket hızı ve doğrultu değişiminden kaynaklanmaktadır. Gerçek Zamanlı Yinelenen Öğrenmedeki sapmalarının ileriki çalışmalarda azaltılması hedeflenmelidir. Ayrıca daha sonraki çalışmalarda Genetik Algoritmalar ile Politika Gradyanlı Destekli Öğrenme yöntemleri birleştirilerek daha iyi sonuçlar elde edilmesi beklenmektedir.

Anahtar Kelimeler: Yapay Yaşam, Dinamik Hareket, Evrimsel Öğrenme, Denetimli Öğrenme, Destekli Öğrenme

ABSTRACT

Locomotion is a favorable area to work on learning techniques. This situation arises from its capability to include all the elements of a difficult learning problem. Due to this fact, this study aims to apply and compare the effect of the usages of various learning techniques on an articulated artificial life form. This way the articulated structure tries to control its actions autonomously. Learning is implemented on an artificial environment which simulates nature's physics laws in their basic form.

Locomotion is a problem that requires adaptability. Thus instead of building a fixed locomotion control structure or providing control from outside of the structure, it is more desirable for the artificial life form to learn how to control its own actions. Therefore, this study makes use of three learning areas in order for the artificial life form to learn to act autonomously. The commonality of these three learning areas is that they have all been influenced by the way organisms operate.

The learning areas worked on are evolutionary learning, supervised learning and reinforcement learning. Genetic Algorithms are used to realize evolutionary learning. Backpropagation Learning is applied on Artificial Neural Networks and Real Time Recurrent Learning is applied on Recurrent Neural Networks in order to realize supervised learning. Lastly, Q-Learning and Policy Gradient Reinforcement Learning are implemented to realize reinforcement learning.

The conclusion of learning with Genetic Algorithms took much more time with respect to other learning techniques and, slow and distinctive movements were observed upon completion of the learning process. Real Time Recurrent Learning, which is one of the tried supervised learning techniques, produced better results than learning by Backpropagation. This technique eventuated fast. In addition, the artificial life form moved very quickly but directional deviations were observed frequently. Due to its inefficiency to deal with the defined large problem space, Q-learning turned out to be unsuccessful. The other reinforcement learning technique Policy Gradient Reinforcement Learning on the other hand, converged fast and resulted in large and distinctive movements. Only the pace of the locomotion turned out to be low.

In conclusion, the implementations of all the learning areas resulted in some kind of locomotion that was visually above the threshold. The results of the learning techniques mostly differed on speed of learning, pace of locomotion and change in direction. The directional diversions of Real Time Recurrent Learning are aimed to be reduced in future studies. Additionally, the combination of Genetic Algorithms and Policy Gradient Reinforcement Learning is offered as another future work.

Keywords: Artificial Life, Locomotion, Evolutionary Learning, Supervised Learning, Reinforcement Learning

1. GİRİŞ

Bu tezde zeki davranışların yaratılması, uyarlanabilirliğin sağlanması ve basit etkileşimlerden kaynaklanan karmaşık davranışların geliştirilmesi üzerine çalışılmıştır. Bu bölümde, bahsi geçen çalışma alanlarının geçmişine genel bir bakış yapılmıştır (Wolfram, 2002).

Yapay Zeka (Artificial Intelligence) için önerilmiş birçok tanım insansal davranışlara olan benzerliklere dayanmaktadır. Bu anlamda, bu tezde ele alınan yapay zeka kavramı karmaşık problemlere doğada rastlanılan türde çözümler getirebilen etmenler (agent) tasarlamak üzerine kurulmuştur.

Geleneksel yapay zeka yöntemleri yukardan aşağıya yaklaşım sergilerler. Bu yaklaşıma göre kavrama yetisi yüksek seviyeli bir kavramdır ve düzeneğin uygulamasından ayrı bir düzlemde ele alınmalıdır. Bu yaklaşımın başlangıcı 1950 yılına, Alan Turing'in makinelerin düşünme eyleminde bulunup bulunamayacağı üzerine ortaya attığı tartışmaya dayanır. Alan Turing bu problemin üzerinde çalışmaya Turing testini tasarlayarak başlamıştır. Bu testin aktörleri soru soran bir kişi, bir erkek ve bir kadından oluşmaktadır. Soru soran kişi diğer ikisinden ayrı bir odaya yerleştirilir ve soracağı sorulara verilen yanıtlara göre yanıt verenin kadın mı erkek mi olduğunu anlamaya çalışır. Erkeğin özellikle yanıltıcı cevaplar vermeye hakkı vardır, kadın ise soru soran kişiye olabildiğince yardımcı olacaktır. Turing'in yapay zeka alanındaki sorusu burada devreye girer; kadın veya erkek katılımcıdan biri yerini bir makineye devrettiği takdirde makinenin yeterince iyi bir iş çıkarıp çıkaramayacağı anlaşılmaya çalışılır. Bu testin sonunda makinenin testi geçtiği kabul edilmiştir fakat bu testle makinenin düşünüp düşünemeyeceği konusu cevaplanmış olmaz.

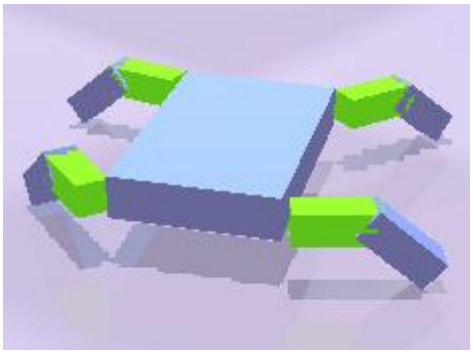
Yapay zeka çalışmalarının baştaki amacı genel amaçlı problem çözme programları yaratmak olmuştur. İlk zamanlardaki girişimler sonuç vermeyince yapay zeka çalışmalarının ilgi alanı 1960'lardan 1980'lerin başına kadar gitgide daralma göstermiştir. Sadece kimi uzmanlık alanlarındaki dar kullanımı olan bazı problemlere belirli arama algoritmaları uygulanmıştır. Bunun yapılabilmesinin nedeni belli bir alandaki bilgilerin bilgisayarların üzerlerinde mantık yürütebileceği şekilde gösterimlerinin yapılmasının mümkün oluşunun fark edilmesidir. Daha sonra yapay zekanın klasik yukardan aşağı yaklaşımı kullanılarak uzman sistemler geliştirilmiş; bu tür sistemlerin insanlarla karşılaştırıldığında süreklilik ve kullanım kolaylığı gibi pek çok avantajı olduğu öne sürülmüştür.

1980'lerin sonlarına doğru yapay zeka çalışmaları sınıflandırılması kolay olmayan birkaç yöne ayrılır. Bu yönlerden ikisi giderek daha çok ilgi kazanmaya başlamıştır. Her iki tutum da

aşağıdan yukarı bir yaklaşım sergiler. Aşağıdan yukarı yaklaşım fikri biyolojik evrimsel sistemler esasına dayanarak ortaya atılmıştır. Bu tutumlar başlıca Makine Öğrenmesi ve Arama/Eniyileme temeline dayanırlar. Makine Öğrenmesi Sinir Ağları (Neural Networks), Veri Madenciliği (Data Mining), Bayes Ağları (Bayesian Networks) ve Karar Ağacı Öğrenme (Decision Tree Learning) yöntemlerini içerir. Arama ve Eniyileme ise planlama ve fonksiyon eniyilemesine dayanır. Bu tezde her iki yaklaşımdan da yararlanılmıştır.

Dinamik hareket etme problemi öğrenme yöntemleri üzerinde çalışmak için çok elverişli bir problemdir. Bu durum hareket probleminin zor bir öğrenme probleminin bütün parçalarını içermesinden kaynaklanmaktadır. Zorluğun öncelikli nedeni hareket eden etmenin bünyesinde birçok bağımsızlık derecesi bulundurmasıdır; bu yapı, etmenin her olası durumuna göre performans en iyileştirmesi yapmak istemesini beraberinde getirir. Bir diğer neden ise bir hareket problemini öğrenmeye çalışan yöntemlerin aynı zamanda yeryüzü ile yaşanan çarpışmalardan kaynaklanan süreksiz durumlarla baş edebilmesi gerekliliğidir. Önemli nedenlerden bir başkası da öğrenme yöntemlerinin aynı zamanda ertelenen ödül sorunuyla karşılaşacak olmasıdır; bu sorun belli bir anda etmene uygulanan torkların daha ileriki adımlarda etmenin performansına etki edebilecek olmasıdır.

Otomatik hareket etme problemi bir etmenin görsel olarak da inandırıcı olabilecek şekilde mesafe kat edebilmesi olarak tanımlanabilir (Tedrake vd., 2005). Tezde birçok kere etmen olarak bahsedilecek varlık aslında yapay yaşam formunun kendisidir. Bu etmen bir animasyon karakteri gibi düşünülebilir. Birçok animasyon karakteri gibi ve daha sonra nedeni anlatılacağı üzere, etmen eklemli bir yapıya sahip olacak şekilde tasarlanmıştır. Öğrenme de bu yapı üzerinden gerçekleştirilmiştir. Öğrenmenin gerçekleştirilebilmesi için başarı ölçütü etmenin kat ettiği mesafe olarak tanımlanmıştır. Belirlenen başarı ölçüsüyle öğrenme gerçekleştirildikten sonra elde edilmek istenen sonuç yaratılmış olan etmen için fiziksel olarak gerçekçi görünen hareketler elde etmenin yanında etmenin bu hareketleri yaptığında bulunduğu konumdan olabildiğince uzaklaşmasıdır. Şekil 1.1'de bu tez için yaratılmış olan yapay yaşam formunun genel yapısı gösterilmiştir.



Şekil 1.1 Yapay yaşam formu

1.1 Yapay Yaşam

Yapay yaşam (Artificial Life), gerçek yaşamın bağımsız örneklerini veya başlıca öğelerini yaratma olanaklarını araştırır. Gerçek yaşamı anlama işlemi onu parçalara ayırarak değil inşa ederek gerçekleştirilir. Bu nedenle bu tutum indirgemeci veya analitik olmaktan çok sentetik olarak değerlendirilmektedir (Ray, 2001).

Bizler evrimin dünyadaki yaşama özgü bir olay olduğunu düşünmeye alışmışızdır. Biyoloji, bu anlamda, uygulamaları bölgesel olarak sınırlanmış bir bilimdir. Elimizdeki tek yaşam örneği dünyadaki karbon temelli yaşamdır. Dünyadaki bütün yaşam formları aynı temel düzenekleri gerektirir. Bütün canlılar protein ve DNA'dan oluşmuş mekanizmaların kontrolü altında çoğalır ve gelişir. Fakat bunun tek olası yaşam temeli olup olmadığı tam olarak bilinmemektedir (Farmer ve Belin, 1991).

O zaman, başka gezegenlerdeki yaşam formlarına rastlayacak olursak biyoloji bilimi onları anlamamızda bize yardımcı olabilecek midir? Bu soruyu sorduğumuzda yaşamın ne olduğunu düşünmeye motive oluruz. Yapay bir şeyin aynı zamanda nasıl yaşayabileceğini anlamak için de yaşamın ne olduğu sorusu mümkün olduğu kadar cevaplandırılmaya çalışılmalıdır.

Yaşam için üzerinde genel olarak anlaşılmış bir tanım henüz mevcut değildir. Yaşamı tanımladığını düşündüğümüz herhangi bir özellik ya yanında birçok cansız sistemi de tanımlayacak kadar geniş, ya da kimi yaşıyor olarak kabul ettiğimiz varlıkları tanımlayamayacak kadar dar bir anlam içerecektir. Bu nedenle yaşamın yaratılmasının ne demek olduğu hakkında da kesinlik ifade eden sözcükler kullanılamaz. Genelde, yaşamı tanımlama çabaları belli başlı özelliklerin sağlanıp sağlanmadığını test etmekten geçer. Tanımlamadaki problemlerden biri de bu özellik listesine nelerin ekleneceği konusundaki anlaşmazlıktır. Çoğalabilme, evrim geçirebilme, metabolize olma, uyarıcılara tepki verme ve hasarı tamir etme gibi nitelikler birçok listenin ortak özellikleridir. Yapay yaşamın birçok

örneği özellik listesi çok kısa olmadıkça bu tür bir testi geçemeyecektir (Ray, 2001). Bu yaşam-testi yaklaşımı pek yeterli değildir. Bunun nedenini anlayabilmek için bir makinenin bir tartışma arenasında oradaki insanlar kadar akıllı bir katılımcılık sergilediğini düşünelim. Bu makine çoğalamasa, evrim geçiremese veya birçok listedeki çoğu özelliği gösteremese bile onun bir bakıma yaşıyor olduğunu inkar etmek zordur (Ray, 2001).

Bu tür düşünceler probleme alternatif bakış açıları üretilmesine önayak olur. Bu nedenle bir başka anlayış daha geliştirilmiştir. Bu sefer, sadece canlı sistemlerde olabileceği bilinen, cansızlarda rastlanması olanaklı olmadığı düşünülen, özelliklerin uzun bir listesi yapılır. Artık Yapay Yaşam'ın bir örneği sayılmak için bu listedeki her özelliği sergileyebilmek yerine listedeki herhangi bir özelliği temsil edebilmek yeterlidir (Ray, 2001).

Y-yaşam çalışmalarının çoğunda bu tutum kabul edilmiştir. Genelde yaşamın evrim, zeka, dil, toplumsal davranış, gelişme gibi tek bir durumu ile ilgilenilir. Sistem ilgi alandaki yaşam özelliğini sergileyecek şekilde yaratılmaya çalışılır. Bu sistem diğer yaşam özelliklerinin az bir kısmını sağlayabilir veya hiçbirini sağlamaz. Böylelikle bunlar birbirinden ayrı oluşturulmuş yaşam örnekleri haline gelir. Bunların canlılığını sorgulamak doğru olmayabilir çünkü bu cevaplanabilirliği olan bir soru değildir. Bunun yerine sorulması gereken ve daha kolay cevaplanabilecek olan soru bu sistemlerin yaşam özelliklerinden bir veya birkaçını gösterebiliyor olup olmadıklarıdır (Ray, 2001). Bu tezde hareket edebilme yeteneği temsil edilmeye çalışılmıştır.

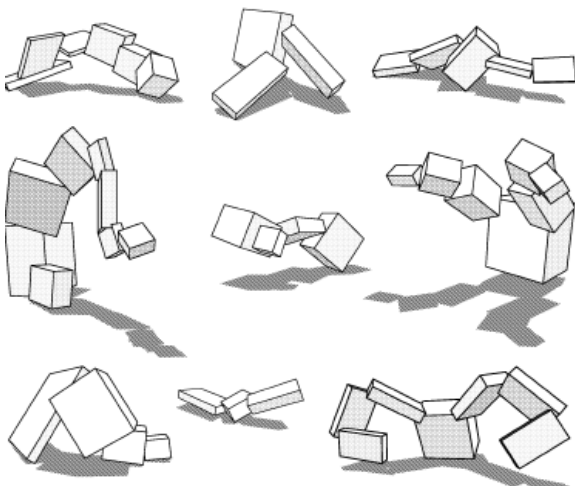
Yapay yaşam araştırmalarında bilgisayar organik yaşamı modelleyen bir araç olarak görülmek yerine karbon temelli olmayan yaşamların mesken edindiği bir evren olarak kabul edilir. Y-yaşam çalışmalarında bu evrene yaşam formlarının tohumları ekilir ve daha zengin dijital formları desteklemek için sistem geliştirilir.

Y-yaşam yaşayan biyolojik sistemleri karmaşık algoritmalar yoluyla modellemeyi hedefleyen bilimsel bir çalışma alanıdır ve karmaşık uyarlanır sistemlerde ortaya çıkan davranışlarla gelişen zekayı inceler. Y-yaşamın ana amacı etmenler ve çevreleri arasındaki etkileşim sayesinde zeka veya yaşamsal özellik yaratımıdır. Yaşamın evrimsel gelişimini izleyip, evrimsel süreci çalışmak da bir başka önemli amaçtır. Etkileşim kaynaklı davranışlar yaratma veya zeka geliştirme sayesinde ortaya çıkan yaşam kendini çoğaltabilen ve etrafına uyum sağlayabilen bir yapı sayılır. Aynı zamanda yaratılan yaşamın dışardan gelen etkilerden bağımsız hareketler üretebilmesi gerekir.

Y-yaşam çalışmalarında yaratılan çevrenin doğayı yansıtması gerekmez. Hatta birçok Y-

Yaşam uygulamasında ya basitleştirilmiş ya da kuralları doğanınkilerden tamamen farklı olan bir çevre kullanılmaktadır. Çevrenin basitleştirilmesi, araştırmacıların çevrenin karmaşıklığı yerine yaşamın özellikleri üzerine odaklanmasını sağlar. Bundan dolayı Y-Yaşam alanında doğanın fizik kurallarını modelleyen çalışmalar çok yoktur. Şimdiye kadarki en karmaşık ve gerçekçi denilebilecek simülasyonun Karl Sims'in 1994'te Siggraph'ta yayınladığı çalışmasına ait olduğu söylenebilir. Karl Sims'in yapay yaratıkları 3 boyutlu fiziksel bir dünyada hareket eder. Bu tezin çalışma alanının seçilmesindeki ana etmen de Karl Sims'in blok yaratıklarıdır.

Blok yaratıklar (Şekil 1.2) davranışları kendileri tarafından üretilmiş yaratıklardır. Bu amaçla bir çevre hazırlanmış ve yaratıklara fonksiyonlar verilmiştir. Daha sonra bu yaratıklar evrimleşme yoluyla çevreyle nasıl etkileşim haline geçeceklerini çözmeye çalışmışlardır. Yaratıkların evrimleşebilmesi için genellikle hedefler bir isteklendirme unsuru olarak kullanılmıştır.



Şekil 1.2 Yürümek için geliştirilmiş yaratıklar (Sims, 1994)

Sims blokların birleştirilmesinden oluşturulmuş bir yaratık olmanın problemlerine çözüm getirmek için esnek bir genetik sistem kullanmıştır. Bu yaratıkların blok parçaları, elektrik devreleriyle kontrol edilen "kaslarla" güçlendirilmiş esnek eklemler ile bağlanmıştır. Sims bu yaratıkları su veya yüzey gibi bilindik fiziksel ortamların simülasyonlarına yerleştirmiştir. Bu yaratıklar, daha sonra, yüzmek, bir ışık kaynağının peşi sıra yüzmek, bir yüzey üzerinde hareket etmek, bir yüzey üzerinde zıplamak ve bir kare bloğa sahip olmak için başka bir yaratıkla yarışmak gibi farklı görevleri gerçekleştirebilme becerilerine göre seçilmişlerdir. Simülasyonlarının hesaplama karmaşıklığı çok yüksek olduğundan Sims onları makinelerin birbirine bağlı olduğu bir süperbilgisayar (supercomputer) üzerinde çalıştırmıştır.

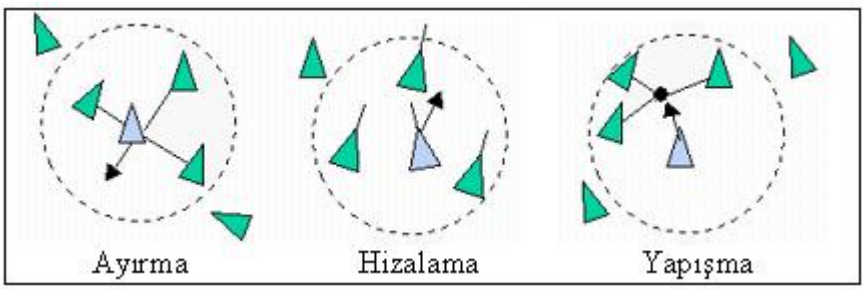
Yaratıkların morfolojileri ve kas güçlerini kontrol etmeleri için gerekli olan sinir sistemleri genetik algoritmalar kullanılarak otomatik olarak üretilmiştir. Farklı uygunluk değerlendirme fonksiyonları evrimleri belli davranışlara doğru yönlendirmek için kullanılmıştır. Bu yaratıkların beyinleri sinir ağı ile yapılandırılmamıştır, bunun yerine daha çok bir bilgisayar programının veri akışına benzemektedir.

Yaratıkların morfoloji ve sinir devrelerini tanımlamak için yönlü graflar kullanılmıştır. Bu yönlü grafları temsil etmek içinse düğüm noktalarını ve bağlantıları ilkel öğeler olarak kullanan bir genetik dil sunulmuştur. Bu genetik dil, sayısız miktarda farklı davranışlara sahip olası yaratıkları içeren bir uzay tanımlar. Bu uzay üzerinde en iyileme teknikleri ile araştırmalar yapıldığında, değişik birçok başarılı ve ilginç hareket stratejileri belirir.

Bir diğer 3-boyutlu yaşam formu simülasyon da 1986 yılında Craig Reynolds (1987) tarafından gerçekleştirilmiştir. Reynolds bir hayvan sürüsü yaratmasını sağlayacak bir program yazmıştır. Bir sürüyü oluşturan oluşumun, kuşların kendi başlarına gerçekleştirdikleri davranışların etkileşimi olduğunu varsaymıştır. Bir sürüyü simüle etmek için, tek bir kuşun hareketinin simülasyonunu (en azından kuşun davranışlarından bir sürüye katılmasını sağlayan kısmını) gerçekleştirmiştir. Bu davranışsal kontrol yapısını desteklemek üzere, kuşun algısal mekanizmalarını ve uçuş için gerekli fiziğini de simüle etmek gereklidir. Simüle edilen kuş doğru sürü üyesi özelliklerine sahip olursa yapılması gereken tek iş, kuş modelinden birkaç örnek türetmek olacaktır. Simüle edilen kuş, etkileşerek bir sürü oluşturmak için gerekli işleri kendisi yapacaktır.

Temel sürü modeli, yönlendirici üç basit davranıştan oluşur (Şekil 1.3):

- Ayırma: Kalabalık yaratan yerel sürü üyeleri birbirinden ayrılmalıdır.
- Hizalama: Uçuş yönü, yerel sürü üyelerinin uçtukları yönün ortalamasına doğru olmalıdır.
- Yapışma: Pozisyon, yerel sürü üyelerinin pozisyonlarının ortalamasında olmalıdır.

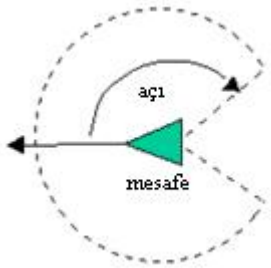


Şekil 1.3 Reynold' un yönlendirici kuralları

Şekil 1.3'teki çemberler, yerel sürü üyelerini göstermektedir. Mevcut boid'in belirli bir miktar

uzağında bulunan boid'lerin tümü, yerel sürü üyeleridir.

Daha ayrıntılı komşuluk fikri, yerelliği belirleyen çemberi boid'in mevcut ileri yönüyle sınırlar. Bu yaklaşım, gerçek dünyada sürü üyelerinin birbirleriyle olan ilişkilerini daha gerçekçi yansıtır (Şekil 1.4).

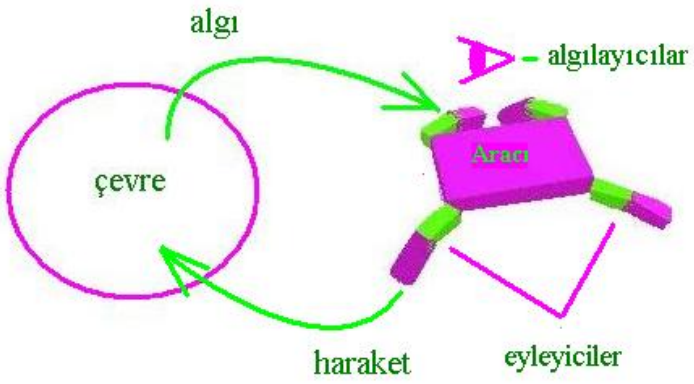


Şekil 1.4 Uzaklık ve açı temelli boid komşuluğu

Hedef arama ve engelden sakınma kuralları da eklendiğinde, yaratıklar tam bir sürü gibi yön bulabilir hale gelir. Yaratıklar sütunlar arasından düzenli bir şekilde geçebilir, bir grup halinde durabilir, gidiş hızlarını ayarlamak üzere hızlanabilir veya yavaşlayabilirler.

1.2 Etmen

Bir etmen, en genel anlamda, çevresini algılayıcılar (sensors) tarafından duyumsayabilen ve eyleyicilerini (actuators) kullanarak mantıklı bir şekilde hareket edebilen bir bilgisayar yazılım sistemidir (Şekil 1.5). Dolayısıyla, etmen, algıları birer birer elde eder ve bu algı sırasını yapacağı hareketlere eşler. Bir etmenin sahip olabileceği temel özellikler yerleşmişlik (situatedness), özerklik (autonomy), uyarlanabilirlik (adaptivity) ve toplumsallıktır (sociability). Yerleşmişlik özelliğine sahip olan bir etmen çevresinden bir takım algısal veriler alır ve çevresini bir şekilde değiştiren bir hareket üretir. Etmenin özerk olması ise insanlar veya başka etmenler doğrudan müdehale etmeksizin etmenin kendi hareketleri ve iç durumunu (internal state) kontrol edebilmesi anlamına gelir. Uyarlanabilirliği olan bir etmenin çevresindeki değişikliklere tepkilerinin esnek olması, gerektiğinde hedef güdümlü insiyatif alabilmesi ve kendi tecrübelerinden, çevresinden ve diğer şeylerle olan etkileşimlerinden bir şeyler öğrenebilmesi beklenir. Toplumsallık da bir etmenin diğer etmenlerle veya insanlarla eşler arası (peer-to-peer) şekilde etkileşebilmesidir (Siegwart ve Nourbakhsh, 2004). Bu çalışmada yaratılan etmenlerin toplumsallık dışındaki bütün özellikleri taşıması amaçlanmıştır.



Şekil 1.5 Etmenin çevresiyle etkileşimi

Etmenin tanımında da karşılaşıldığı gibi mantılılık (rationality) önemli bir kavramdır. Hedefe ulaşmanın yanında doğru hareketlerde bulunma ve gerçekçilik önemli yer tutar. Bir etmenin mantıklı olması birden fazla koşulun sağlanmasına bağlıdır. Bunlardan önemli bir tanesi performans ölçüsüdür (performance measure). Performans ölçüsü etmenlerin başarılarını ölçebilmek için şekillendirilen ölçüte verilen isimdir. Diğer koşullar ise etmenin çevreyi başarılı bir şekilde algılaması, etmenin gerçekleştirebildiği hareketlerin çeşitli olması ve algıları hareketlere eşleştiren fonksiyonun veya sistemin başarılı olmasıdır. Böylelikle, mantıklı bir etmen, bildiklerini kullanarak, her olası algı sırası için performans ölçüsünü enbüyültecek hareketlerde bulunmalıdır.

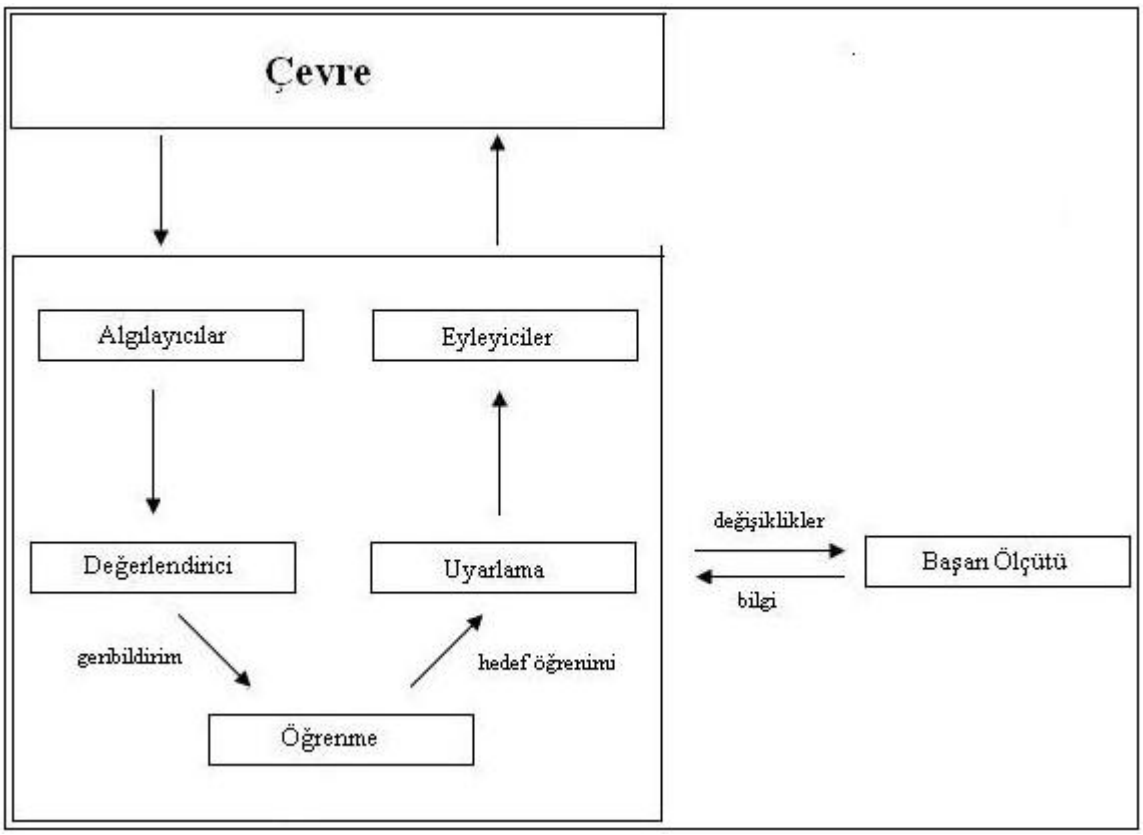
1.2.1 Etmen Tipleri

Genelde dört farklı etmen tipinden bahsedilir. Bunlar basit refleksli etmenler (simple reflex agents), durum bilgili refleksli etmenler (reflex agent with state), hedef temelli etmenler (goal-based agents) ve yarar temelli etmenlerdir (utility based agents). Bu etmenlerin her biri öğrenme etmenlerine (learning agents) çevrilebilir (Siegwart ve Nourbakhsh, 2004). Basit refleksli etmenlerin önceki algısal girdiler için bellekleri yoktur ve durum koruması yapmazlar. Durum bilgili refleksli etmenlerde önemli görülen algısal girdiler etmenin durumunda değişikliğe neden olabilirken diğer ilgisiz girdiler sisteme verildiğinde durum değişiklik göstermeyebilir. Böylelikle etmenin hareketleri etmenin hem o andaki durumu hem de algı girdilerine bağlı olarak gerçekleşecektir. Bir refleksli etmen hedefleri varmış gibi davranabilir ama onları açıkça temsil etmez. Bu onun hedeflerine nasıl varması gerektiğini muhakeme edemediği anlamına gelir (Siegwart ve Nourbakhsh, 2004). Hedef temelli etmen ise çok daha gelişmiş bir tür etmendir. Bir veya daha fazla hedef durumu tanımlıdır. Etmenin durumu içinde bulunduğu çevrenin bir gösterimini (representation) içermektedir. Bu tip bir

etmen hareketlerinin sonuçlarını muhakeme edebilme yeteneğine sahiptir. Yani D1 durumunda H1 hareketini yaptığında D2 durumuna geçeceğini bilir. Böylece etmen hedefine ulaşabilmek için tecrübelerinden yararlanabilir. Hedef temelli etmenlerde hedef durumları önceden belirlenmiştir, etmen kendi isteklerini değerlendirip yeni hedefler oluşturamaz. Buna karşın yarar temelli etmenlerde hedefler önceden belirlenmiş durumlar olmaktan çok daha soyut kavramlardır. Bunun için bir yarar fonksiyonuna (utility function) ihtiyaç vardır (Siegwart ve Nourbakhsh, 2004). Bu fonksiyon herhangi bir durumu alır ve buna karşılık bir değer üretir. Tezde etmen üzerinde uygulanan öğrenme yöntemleri değiştikçe etmenin tipi de değişkenlik göstermektedir. Buna göre gözlemlenen etmen tipleri basit refleksli, durum bilgili refleksli ve hedef temelli etmenlerdir.

1.2.2 Etmenlerde Öğrenmenin Yeri

Bahsedilen etmenlerin hepsi tamamen önceden belirlenmiş algoritmalar çalıştırabilir fakat bu tür bir hareket alma stratejisi herhangi bir durumla karşılaştığında etmenin, yararı enbüyütme amaçlı olarak, davranışını değiştirmesine olanak tanımaz. Buna karşılık öğrenme etmeni daha esnektir (Şekil 1.6). Bu etmenin uyarıcı ve durum bilgisini harekete eşleyecek bir performans unsuru mevcuttur. Etmenin hareketlerine neden olan durumların yararlılığını değerlendirebilecek bir eleştiriciden yararlanılır. Daha iyi durumlarla sonuçlanan eşlemeleri elde etmek adına performans unsurunu etkileyen bir öğrenme öğesine başvurulur. Kimi zaman da etmen yeni bir şeyler denemeye zorlanabilir. Bunun için etmenin yeni hareketler denemesi gerekir. Keşif (exploration) ve sömürü (exploitation) arasındaki ilişkinin iyi ayarlanması, dengenin sağlanması önemlidir.



Şekil 1.6 Etmenlerin öğrenme devri

Bir etmenin mantıklı davranabilmesi için öncelikle etrafını mümkün oldukça iyi bir şekilde sezinleyebilmesi gerekmektedir. Bu nedenle etmenler bir tür geri beslemeli kontrol (feedback control) mekanizmasına gerek duyarlar. Geri beslemeden kasıt etmenin algılayıcılarını sürekli olarak izlemesi ve değişikliklere tepki vermesidir. Böylelikle geri beslemeli kontrol etmenin öz denetimini (self-regulation) sağlar. Artı geri besleme ve ters geri besleme olmak üzere iki çeşit geri besleme yöntemi vardır. Bunları kullanarak etmen kendi kontrolünü sağlar. Artı geri besleme sistem durum/çıktı'sını büyütmeye yönelik olarak hareket ederken ters geri besleme sistem durum/çıktı'sını ayarlamaya yönelik olarak hareket eder.

1.2.3 Hareket Yeteneği

Bu çalışmada amaç yaratılan etmenin hareket etme becerisinin çeşitli yöntemlerle arttırılması ve sonuçlanacak hareketlerin gerçekçi görünmesidir. Bu izlenimi yaratmak için kullanılacak yöntemlerden önce hareket (locomotion) kavramının açıklanması önemlidir. Yaratılan etmen bir gezgin robot simülasyonu niteliği gösterdiğinden etmen böyle bir robotun hareket yeteneğine sahip olmalıdır. O halde simülasyonun gerçekçiliğini sağlamak amacıyla gezgin robotlardaki hareket mekanizmalarından (locomotion mechanism) faydalanmak önemlidir. Bir gezgin robotun kendi çevresi boyunca sınırsız bir şekilde hareket edebilmesi için hareket

mekanizmalarına ihtiyacı vardır. Bol miktarda değişik hareket olasılığı bulunduğundan robotun hareket kavramına yaklaşım tarzı gezgin robot tasarımında oldukça önemli bir yer tutar.

Gezgin bir robotun hareketi (locomotion) ve bir robotun kımıldaması (motion) farklı tür hareketlerdir. Kımıldama işinde robot kolu sabittir fakat güç uygulayarak çalışma alanı içindeki nesnelerin yerlerini değiştirir. Gezgin bir robotun hareketinde ise çevre sabittir ve robot çevre üzerinde güç uygulayarak kendi durumunu değiştirebilmektedir. Bu türlü bir harekette, hareketin gerçekçi olabilmesi için ve varsa hedeflere ulaşmada başarı sağlanabilmesi için önemsenmesi gereken birçok özellik mevcuttur. Bunlar denge, temas noktalarının sayısı ve geometrisi, ağırlık merkezi, statik/dinamik kararlılık, bölgenin eğimliliği, temasın özelliği, temasın açısı, çevrenin tipi, yapı ve medyadır (su, hava, yumuşak, sert vb.) . Bu çalışmada, engebesiz ve sert bir arazide yerçekimli bir ortamda çalışılmaktadır. Bu çalışma karada yaşayan hayvanlara benzer bir yaşam formu üzerinde gerçekleşmektedir. Bu yaşam formunun anlatımı daha detaylı bir şekilde üçüncü bölümün ikinci kısmında yer almaktadır.

1.2.4 Hareket Tipleri

Doğada çeşitli hareketler gözlemlemek mümkündür. Canlı varlıklarda bunlardan en çok rastlanılanları sürünme, kayma, zıplama ve yürümedir. Bu çalışmada hareketin türü hakkında bir kısıtlama yoktur. Yaratılan yapay yaşam formunun morfolojisi bu hareketlerin hangisini yapmaya daha yatkınsa ona benzer hareketlerin gözlemlenmesi beklenmektedir. Gezgin robotlar genelde iki tür hareket mekanizmasından birini kullanır. Bunlardan biri bacaklı hareket (legged locomotion) mekanizması, bir diğeri ise tekerlekli hareket (wheeled locomotion) mekanizmasıdır. Tekerlekli hareket mekanizması aslen taşıtlar için yaratılmış bilindik bir teknolojinin eseridir. Bacaklı hareket ise günlük hayatta çok daha az karşılaşılan bir tür mekanizmadır, hareketin esası eklemli (articulated) bacak kullanımına dayanır. Bacaklı hareket genellikle daha yüksek serbestlik dereceleri (degrees of freedom) gerektirdiğinden bu hareketin mekanik karmaşıklığı tekerlekli harekete göre büyüktür. Bu bir dezavantaj gibi görünüyor olsa da doğada gözlemleyebildiğimiz canlılara gerçekçi bir yaklaşım üretmemizde daha başarılı sonuçlar elde etmemiz açısından faydalı olmaktadır. Ayrıca, tekerlekli hareket mekanizmaları yer yüzeyi yumuşadıkça, yuvarlanma sürtünmesi nedeniyle, etkilerini yitirirler (Siegwart ve Nourbakhsh, 2004). Öte yandan bacaklı hareketler, bu tür ve birçok farklı yer yüzeyinden çok daha az zarar görürler. Bunun nedeni zeminle noktasal temas üzerine kurulu

bir yapılarının olmasıdır. Dolayısıyla, tekerlekli hareketin etkinliği başta zeminin düzlüğü ve sertliği olmak üzere çevresel etkenlere çokça bağlıyken bacaklı hareketin etkinliği bu tür faktörlerden pek etkilenmez. Bacaklı hareket, çevresel etkenler yerine bacak ve gövde ağırlığına bağlıdır. Bu iki hareket türünün etkinlik alanlarına bakıldığında bacaklı hareketin doğaya daha uygun olduğu görülmektedir. Bundan ötürü bu çalışmada da bacaklı hareket mekanizmasından yararlanılmaktadır.

Bacaklı hareket, robot ve yer yüzeyi arasındaki noktasal temas dizisiyle nitelendirilir. Bu hareketin en önemli avantajları araziye uyum yeteneği (adaptability) ve manevra yapma kabiliyetidir (maneuverability). Hareketi yapmakta yalnızca bir dizi temas noktasına gerek duyulduğundan, robotun gövdesiyle zemin arasında yeterli miktarda açıklık olduğu takdirde, o noktalar arasındaki zeminin özelliği pek bir önem taşımaz. Fakat daha önce de bahsedildiği gibi bacağın ağırlığı önemlidir. Bacak robotun toplam ağırlığının bir kısmına destek olmak durumundadır. Ayrıca birçok robotta, bacağın robotu yukarı kaldırıp indirebilmesi de beklenebilmektedir. Üstelik bu özelliklerin sağlanması serbestlik derecesi arttıkça baş edilmesi daha güç bir hal almakla beraber yüksek manevra kabiliyeti ancak yeterli miktarda serbestlik derecesine farklı yönlerde güç uygulandığında erişilebilmektedir. Bacaklı gezgin robotlarda genelde, bacağı yukarı kaldırıp öne doğru sallayabilmesi için, en az iki serbestlik derecesinin olması talep edilir. Daha karmaşık hareketlerin gerçekleştirilebilmesi için de üçüncü bir serbestlik derecesinden faydalanılır.

1.2.5 Hareket Düzenekleri

Hareketin yapılması için dengeleyici ve eyleyiciye ihtiyaç vardır. Dengeleyici robotun çevresi üzerinde etkisini sağlayan bir aygıttır. Bu aygıt robotun kontrolü altındadır. Dengeleyicilere örnek olarak bacaklar, tekerlekler, kol ve parmaklar verilebilir (Siegwart ve Nourbakhsh, 2004). Denetleyicilerin (controller) rolü robotun görevine bağlı olarak denetleyicilerin çevre üzerinde istenen etkilerini göstermelerini sağlamaktır. Erişim düzenekleri robotların hareketini başlatan mekanizmalardır. Bu mekanizma dengeleyicinin işini yapabilmesi için gereklidir çünkü işlem için kuvvet üretir. Böylelikle kontrollü enerji üretimi sağlanır.

Serbestlik derecesi bir robot sisteminin yapılanışını (configuration) tanımlamak için gerekli olan boyut sayısıdır. Eklem sayısı buna bir örnektir. Genelde bir alan içinde hareket edebilen serbest bir gövde altı serbestlik derecesine sahiptir. Bunlardan üçü x, y, z doğrultusundaki aktarım içinken diğer üçü yönelme (orientation) ve rotasyonu sağlamak içindir. Bu son üç serbestlik derecesi yuvarlanma (roll), atış (pitch) ve sapma (yaw) hareketlerini yaratır.

Robotun her dengeleyicisi için robotun kaç serbestlik derecesinin olduğu bilinmelidir (Siegwart ve Nourbakhsh, 2004). Her serbestlik derecesi için bir denetleyicinin bulunması bütün serbestlik derecelerinin kontrol edilir nitelikte olduğunu gösterir fakat bütün serbestlik derecelerinin kontrol edilebilirliğine sık rastlanmaz.

2. YÖNTEMLER

Robotlarla ilgili ilk çalışmalar 1980'li yılların ortalarında boy göstermeye başlar. Bu yeni ilgi alanı beraberinde araştırmacıları yeni araştırma tekniklerine yöneltmiştir. Bu tekniklerden birçoğu belli bir oranda canlı sistemlerin yapısal özelliklerinden yararlanmanın yanında davranış temelli robotik, yapay yaşam, evrimsel yöntemler ve davranış modelleme gibi araştırmaya çok açık olan konulardan da etkilenmiştir. Bu alanlara olan ilgi basitlik ve uyarlanabilirlik (adaptability) prensiplerini temel alır. En iyi doğa bilir ("nature knows best") mantığı robotik problemlerinde giderek daha çok rastlanılan bir yaklaşım haline gelmiştir (Sharkey, 1997). Bu tezde de hareket problemine bu mantıktan yola çıkarak yaklaşılmıştır. Bu amaçla, canlıların işleyişini ele alan, doğayı modellemeye çalışan üç önemli yöntem ele alınmıştır. Bunlar evrimsel öğrenme (evolutionary learning), denetimli öğrenme (supervised learning) ve destekli öğrenmedir (reinforcement learning).

Özellikle Yapay Yaşam konularında olmak üzere evrimsel öğrenme robot çalışmalarında popüler bir öğrenme yöntemi olmaya başlamıştır. Evrimsel öğrenme, özellikle bilinmeyen ortamlarda çalışıldığında, önemli bir araçtır. Genel yaklaşımı gerçek evrimsel teoriye benzer bir yapıya sahiptir; üzerinde çalışılan programın problemin amacına ne kadar uygun olduğuna karar vermek amacıyla bir uygunluk fonksiyonu vardır ve farklı çözümler sağlayabilmek amacı ile gen dizilerine benzer diziler üzerinde çaprazlama ve mutasyon işlemleri gerçekleştirilir. Robotların davranışları doğadaki canlıların davranışlarına yaklaştırılmaya çalışıldığında evrimsel öğrenmenin önemi yadsınamaz (Gahot, 2005).

Yapay Sinir Hücresi Yakınsaması (Perceptron Convergence) ve Widrow-Hoff kurallarının tanımlanmasıyla birlikte yapay sinir ağlarının temelleri 1950 ve 1960'larda denetimli öğrenme ile atılmıştır. Geri Yayılımlı Öğrenme'nin (Backpropagation Learning) keşfi ise gerek kullanıldığı problemler için olsun gerek ondan esinlenerek yazılan yeni öğrenme yöntemlerinin ortaya çıkmasına vesile teşkil etmesinden olsun yapay sinir ağları çalışmalarında çok ses getiren keşiflerden biri olmuştur (Sharkey, 1997). Yinelenen Yapay Sinir Ağları ise daha sonradan ortaya çıkarak robot çalışmalarındaki önemli yerini almıştır. Denetimli öğrenmede alınmak istenen sonuçlar tanımlıdır; öğrenmenin sonunda bu sonuca en yakın veriler elde edilmeye yani hata olabildiğince azaltılmaya çalışılır (Uğurlu ve Biricik, 2006). Algılayıcılar ve eyleyiciler arasında direk bir eşleme sağlama imkanından dolayı hareketin kontrol edilmesinde yapay sinir ağları kullanmanın faydalı olacağı düşüncesinden yola çıkılarak bu tezde denetimli öğrenme yöntemlerinden Geri Yayılımlı Öğrenme ve Gerçek Zamanlı Yinelenen Öğrenme' den yararlanılmıştır.

Günümüz robot çalışmalarında kullanılan bir diğer öğrenme yöntemi ise destekli öğrenmedir. Destekli öğrenme, bir Rus davranış psikologu olan Dimitry Pavlov'un (1927) klasik koşullandırma (classical conditioning) ve Thorndike'ın (1898) kafesten kaçma çalışmalarına dayanır. Bu, destekli öğrenme yönteminin 19. yüzyılda araştırılmaya başlanmış olduğu anlamına gelir. Pavlov bir dürtünün koşullandırılmayan bir davranışla nasıl bağdaştırıldığını göstermiştir. Bunu kanıtlanmak için bilindik köpek ve salyalama deneyini yapmıştır. Deneye göre yiyeceğin kendisini gören veya kokusunu duyduğunda ağzı sulanan bir köpek yiyecek yokken de aynı davranışı sergileyebilir. Bunun üzerine Skinner (1953) birçok deney yaptıktan sonra hayvanların davranışlarının dürtü ve tepki birleşimleri olarak incelenebileceğini ifade etmiştir. Skinner'ın bu çalışması modern destekli öğrenme tekniklerinin temelini oluşturmuştur (Sharkey, 1997). Destekli öğrenmenin temelindeki fikir istenen davranışı elde etmeye yönelik ardışık tepkileri ödüllendirilerek davranışı koşullandırmaktır. Evrimsel öğrenme gibi destekli öğrenme yöntemlerinin de önemli bir özelliği bilinmeyen ortamlarda veya görevlerde kullanılmaya elverişli olmasıdır.

2.1 Evrimsel Öğrenme

Bilgisayar animasyonlarında gerçekçi bir şekilde hareket eden yapay yaşam formları yaratmak oldukça zorlayıcı olabilir. Hayvanların eklemli organları gibi hareket eden nesneler pozisyon ve açılarını direk olarak kontrol etme çabası sonucu istenen çıktıları üretebilecek bile olsa çoğu zaman fiziksel anlamda gerçekçi bir görüntü sağlamaz. Uygun dinamiklerin sağlandığı fiziksel anlamda gerçekçi bir çevre simüle edilmesi ise gerçekçiliği arttırırken istenen davranışın elde edilmesini zorlaştırır. Özellikle, yapay varlıkların karmaşıklıkları arttığı zaman bu tür bir yaklaşımın zorluğu da bir o kadar artar. Karmaşıklıkla başa çıkabilmenin bir yolu bir uygunluk fonksiyonu tanımlayarak etmenin davranışını en iyileştirmektir. Evrimsel yaklaşım işte tam burada devreye girer. Evrimsel yaklaşımlar kullanılarak etmenlerin davranışlarını gerçeğe yaklaştırmak mümkündür. Üstelik bunu gerçekleştirirken etmenlerin davranışlarını tanımlayan yordam ve parametrelerin anlaşılması gerekmez. Yöntemin isminden de anlaşılacağı gibi ortam, doğasal benzeri bir güç rasgele davranışlar üzerinde oynayıp kendi içinde kurallar geliştiriyormuş gibi modellenir. Buna göre değişik tip hareketler için değişik uygunluk fonksiyonları tanımlanabilir. Bu yaklaşım kullanıcının elinden kontrol gücünü büyük ölçüde alır. Yine de kullanıcının hiçbir kontrol olanağının kalmadığı söylenemez. Uygunluk fonksiyonunu belirlemek de, çıktılar üzerinde etkili olacağından, bir tür kontrol mekanizmasıdır.

Daha önce de üzerinde durulduğu gibi, yaratılan etmenin kaslarını kontrol edebilmesi için bir tür sinir sistemi geliştirilmiştir. Sinir sisteminde etmenin kendi iç dinamiğini hissettiren algılayıcılar ve hareket etmesini sağlayan eyleyiciler mevcuttur. Daha gerçekçi hareketler elde edebilmesi için, etmenin sinir sistemi evrimleştirilebilir. Böylelikle daha uygun hareketler yaratılmış olur.

2.1.1 Geçmiş

Evrimsel hesaplama oldukça yeni olan bir araştırma alanıdır. Hatta "evrimsel hesaplama" terimi 1991'de ortaya atılmıştır. Farklı açılardan evrimin simülasyonunu yapmaya çalışan araştırmacılar bu terim altında birleşmişlerdir (Baeck, 2000). Evrimsel öğrenmede üreme, rasgele değişim ve nüfustaki bireylerin seçilmesi durumlarından yararlanırlar.

1950li ve 1960lı yıllarda birkaç araştırmacı birbirlerinden bağımsız olarak evrimsel sistemler üzerinde çalışmaya başlamışlardır. Bunun amacı evrim fikrinin mühendislik problemlerinde eniyileme yapmak amacı ile kullanabileceğidir (Friedberg, 1958; Bremermann, 1962). Evrimleşmeyi modellemeye çalışan bu ilk sistemlerin hepsinde en uygunun seçilmesi (selection of the fittest) kavramından biraz olsun yararlanılmıştır.

Evrim teorisini temel alan bu çalışmalar öncelikle kuşkuyla karşılanmıştır. Daha sonra, 1960ların sonlarına doğru, evrimsel yöntemler üç ana dala bölünmüştür. Bunlar genetik algoritmalar (genetic algorithms), evrim stratejileri (evolutional strategies) ve evrimsel programlamadır (evolutioanry programming).

1960'ların sonlarında ortaya atılan evrim stratejileri başta gerçek değerli parametreleri eniyilemek için tasarlanan bir yöntemdir. Daha sonra daha da geliştirilen bu yöntem evrimsel hesaplamada aktif bir alan haline gelmiştir. Yine 1960larda geliştirilen bir başka evrimsel öğrenme yöntemi de evrimsel programlamadır. Bu her iki alandaki çalışmalar yaklaşık 25 sene kadar genetik algoritmalarla paralel devam etmiştir. Bu üç farklı evrimsel yöntem araştırmacılarının aynı topluluğa katılmaları ise 1990larda gerçekleşmiştir. Evrimsel hesaplama tanımı da o zaman yapılmıştır (Mitchell ve Forrest, 1994).

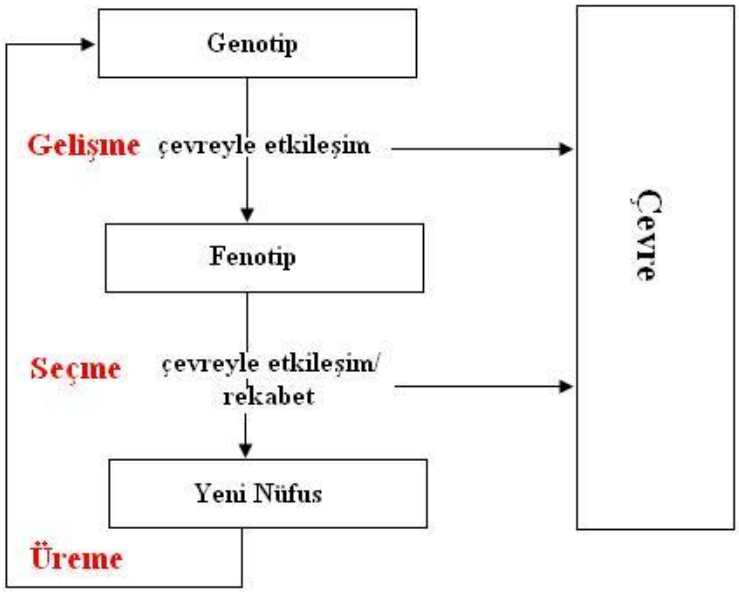
Tezde kullanılan evrimsel hesaplama yöntemi olan genetik algoritmalar ilk olarak 1960larda John Holland (1962) tarafından tanımlanmıştır. Daha sonra Michigan Üniversitesi'nde Holland ve öğrencileri tarafından geliştirilmeye devam edilmiştir. Holland'a göre uyarlanabilir sistemlerin önemli özellikleri rekabetten başarılı bir şekilde faydalanmaları ve değişen olaylara ve çevrelere dinamik bir şekilde tepki verebilmeleridir. Holland evrimin basit

modellerinin bu özellikleri sağlayabilecek kurallara sahip olduğunu görmüştür. Bu modeller en uygun olanının hayatta kalması ve nüfusa devamlı yeni bireylerin katılması prensiplerini temel alır (Baeck, 2000).

Holland'ın öğrencileri 1960'ların ortalarında bu yönteme daha da adına yakışacak bir yön vermişlerdir. Evrimleştirilmeye çalışılan nesneler genomlarla temsil edilmeye başlanmış; üreme ve kalıtım gibi mekanizmalar da bilindik genetik işleçler olan çaprazlama ve mutasyonun basit soyutlamalarıyla gerçekleştirilmiştir. 1970'lerde genetik algoritmaların davranışlarını anlamak için bir miktar daha fazla çaba gösterilmeye başlanır. Nüfus boyutu, gösterim şekli, genetik işleç seçimi ve seçilme oranları gibi konuların genetik algoritmaların davranışlarında önemli yerleri olduğu anlaşılır (Baeck, 2000). 1970'lerin ortasında genetik algoritmalar artık farklı üniversitelerin farklı araştırma laboratuarlarında araştırılmaya başlanır. Genetik algoritmanın bu genişlemesi başta yavaş da olsa ileriki zamanlarda daha da hızlanmıştır. Özellikle 1990'lardan sonra muazzam bir hızlanma yaşanmaktadır. Son yıllarda genetik algoritmalar altında geliştirilen algoritmalar birçok farklı hal almıştır. Araştırmacılar birçok farklı gösterim teknikleri, çaprazlama ve mutasyon işleçleri ve seçim yöntemleri üzerinde durmuşlardır. Yine de hepsi Holland'ın sunduğu yöntem odaklı olmayı sürdürmektedir (Mitchell ve Forrest, 1994).

2.1.2 Yapay Evrim

Evrim sayısız olasılıklar arasından sonuçlar arayan bir yöntemdir. Biyolojide bu olasılıklar kümesi olası genetik dizilere, istenen sonuçlar ise uygunluğu en yüksek olan, yani kendi ortamında yaşamını sürdürüp üremeyi gerçekleştirebilen, canlılara denk gelir. Evrimleşme süreci en basit haliyle Şekil 2.1'de verilmiştir. Evrimleşme her zaman bireylerden oluşmuş bir nüfus üzerinden yapılır. Doğada bunlar yaşam formları iken yapay evrim sürecinde bireyler problemin çözümünün bir parçasıdırlar. Bu tezde ise bunlar yapay yaşam formlarıdır, yani evrimleşme süreci görüntüsel olarak da desteklenmektedir. Her birey birtakım özellikler taşır. Bu özellikler sayesinde toplamında etmenin genomu (genome) oluşturulmuş olur. Bir genomun kapsadığı genlerin toplamı da genotipi (genotype) oluşturur. En basit durumda her gen bir özelliği temsil eder. Genotip, genetik yapı ile organizmanın en son halini, yani fenotipini (phenotype), ayırt eder. Genler genomdaki yerleşimleriyle tanınırlar. Aynı türden olan bireyler aynı sayıda gene sahiptirler ve genlerin genomdaki konumları da eştir. Buna karşın, genlerin değerleri farklılık gösterebilir. Bir bireyin tüm genlerinin değerleri yaşam döngüsü başlamadan belirlenir ve yaşam süreci boyunca değişim göstermez.



Şekil 2.1 Evrimleşme süreci

Gelişim süreci sonunda genotip fenotipe dönüşür. Bu süreç boyunca bir anlamda genler kendilerini anlatmaya çalışırcasına fenotip üzerinde etkilerini göstermeye çalışırlar. Yapay evrim uygulamalarında genelde fenotip ve genotip birbirlerinden farklı olmazlar. O zaman, fenotip kendi ekolojik ortamında diğer bireyler ile rekabet eder. Mücadele sonunda kazananlar seçilir ve yeni bir nüfus oluşturulur. Yeni nüfusun ortalama uygunluğu bir öncekine göre daha yüksektir. Gelişimin sağlanabilmesi için sıra hayatta kalabilenlerin yani seçilmiş nüfusun ortaya yeni bireyler getirmesindedir. Bu işleyiş sırasından da anlaşılacağı gibi evrimsel öğrenmede bireyler tek başlarına değil bir aradayken önem kazanırlar.

2.1.3 Genetik Algoritma

Yapay evrim 1960'ların ortasında zor problemleri çözmek amacıyla geliştirilen buluşsal bir arama algoritması olarak ortaya çıkmıştır (Fogel,1966). O zamandan beri yapay evrimi gerçekleştirmek için birçok yaklaşım geliştirilmiştir. Bu yaklaşımlardan önemli bir tanesi Genetik Algoritma'lardır. Genetik Algoritma, temelinde, klasik arama ve en iyileştirme yöntemlerinden farklı çalışan bir arama ve en iyileştirme yöntemidir. Birçok alana uygulanabilirliği ve kolay kullanımı sayesinde birçok problemde kullanılmaya başlanmıştır. Doğrusal olmayan ve karmaşık etkileşimler içeren problemlerde arama uzayı çoğunlukla birden çok uygun çözüm barındırabilmektedir. Geleneksel yöntemlerin arama uzayındaki yerel çözümlere takıldıkları andan itibaren daha iyi sonuçlar elde edebilmeleri olanaksızlaşır. Genetik Algoritmalar bu anlamda geleneksel yöntemlere üstünlük sağlarlar (Deb, 1998).

Genetik Algoritma genetik ve doğal seçim (natural selection) prensiplerine dayanan bir

yöntemdir ve çalışma yöntemi klasik yöntemlerden çok farklıdır. Genetik Algoritmalar değişkenler yerine gen dizileri ile çalışırlar. Genetik işleçlerle bir genetik dizi oluşturulduktan sonra çözümün değerlendirilmesinin yapılması gerekir. Bu amaçla, probleme özel bir uygunluk fonksiyonu (fitness function) oluşturulur.

Genetik Algoritma nüfus olarak adlandırılan bir hipotez havuzunu bir döngü içinde güncelleyerek çalışır. Arayışa rasgele bir çözüm setinden başlanır. Çözüm nüfusu rasgele oluşturulduktan hemen sonra her biri değerlendirilir. Daha sonra sona erdirme ölçütünün yerine gelip gelmediği denetlenir. Eğer bu ölçüt sağlanmamışsa, çözümler değiştirilir ve yeni bir nüfus elde edilir. Bu yeni bir kuşak yaratılmış olduğu anlamına gelir. Böylelikle, Genetik Algoritma tamamlanana (sona erdirme ölçütüne erişene) kadar birçok kuşak yaratılır. Genetik Algoritma'nın çalışabilmesi için önceden belirlenmesi gereken parametreler vardır. Bunlara Genetik Algoritma'nın girdileri de denilebilir. Girdiler şunlardır (Mitchell, 1997):

- Aday hipotezleri derecelendirebilmek için bir uygunluk fonksiyonu
- Algoritmayı sonlandırmak için yeterli uygunluk derecesine erişildiğini gösteren bir eşik
- Muhafaza edilmesi gereken nüfus sayısı
- Sonraki kuşakların nasıl üretileceğini belirleyen parametreler
- Her kuşakta değiştirilecek nüfus kesimi
- Mutasyon oranı

Genetik Algoritma'nın Temel İşleyiş Şekli (Mitchell, 1996):

1) [Başlangıç] N kromozomlu rasgele bir nüfus yaratılır.
2) [Uygunluk] Nüfustaki her *x* kromozomunun *f(x)* uygunluğu hesaplanır.
3) [Yeni nüfus] Sona erdirme ölçütü erişilene kadar aşağıdaki adımlar yinelenerek yeni bir nüfus yaratılır.
    a. [Seçme] Uygunluklarına göre eşleşmeleri için yeni üst (parent) kromozomlar seçilir. Yüksek uygunluğa sahip kromozomların seçilme şansı diğerlerine göre daha yüksektir.
    b. [Çaprazlama] Seçilen üst kromozomlar kendi aralarında çaprazlanarak yeni bireyler oluştururlar. Çaprazlanma yapılmazsa bu yeni birey, üstünün kopyası olur.
    c. [Mutasyon] Yeni bireyin kromozomu belli bir bölgesinde mutasyona uğrar.
    d. [Kabul] Yeni birey nüfustaki yerini alır.
4) [Yenileme] Genetik Algoritma yeni nesilleri de barındıran yeni nüfus ile çalıştırılır.
5) [Test] Sona erme koşulu sağlandığıysa Genetik Algoritma durdurulur ve o andaki nüfusun en uygun bireyleri şuana kadar elde edilmiş en iyi bireyler sayılır. Yoksa ikinci adıma dönülür.

Evrimleşme sürecinin gerçekleşebilmesi için her bir Genetik Algoritma'da bir seçici baskı

(selective pressure) şekli uygulanır ve farklı seçme (selection), üreme (reproduction), kodlama (encoding) yöntemlerinden yararlanılabilir. Bu yöntemler genel hatlarıyla Çizelge 2.1'de gösterilmiştir.

Çizelge 2.1 Evrimleşmede genel unsurlar

Seçici Baskı	Seçme	Üreme	Kodlama
• Yeni örnekler üzerinden • Üst örnekler üzerinden	• Turnuva Seçimi • Seçkinci Strateji • Derecelendirme Stratejisi • Rulet Tekerleği Seçimi	• Çaprazlama • Mutasyon	• İkili Kodlama • Gerçek Sayı Kodlaması • Çok karakterli kodlama

2.1.3.1 Seçici Baskı

Genetik algoritmanın gitgide daha iyi örnekler elde etme kabiliyeti nüfusa uyguladığı seçici baskıya bağlıdır. Seçici baskı iki farklı şekilde uygulanabilir. Bunlardan ilki nüfusta tutulanlardan çok yeni örnek yaratmak ve bir sonraki nesil için bu yaratılan örneklerin en iyilerini seçip sadece onları tutmaktır. Bu yönteme göre üst örnekler rasgele seçilmiş olsalar bile giderek daha iyi yeni örnekler yaratılacaktır çünkü seçici baskı yeni örnekler üzerine uygulanır. Diğer seçici baskı şekli ise iyi üst örnekler seçip yeni örnekleri bu seçilen üst örneklerden oluşturmaktır. Buna göre de yeni nesilde, nüfusta bulunan yeni örnekler kadar örneğin yaratılması yeterlidir. Seçici baskı yeni örneklere uygulanmasa bile bu yöntem iyileşen örnekler yaratmaya devam edecektir. Bunun nedeni seçici baskının üst örneklere uygulanmasıdır (Syed, 1995).

Doğadaki evrim sürecinde her iki seçici baskı türü çalışıyormuş gibi görünüyorsa da yapay evrime dayalı algoritmalarda bu yöntemlerden ikincisi daha çok tercih edilir. Böylesine bir tercihin asıl nedeni hesaplama yoğunluğu ile ilgilidir. Uygunluklarına karar vermek amacıyla yeni örneklerin değerlendirilmesi genetik algoritmanın hesapsal açıdan yoğun bir kısmını oluşturur. Hesaplama gücü göz önünde bulundurulduğunda hangi üst örneklerin çoğalmasına izin verileceği hakkında seçim yapma kararı almak, seçici baskıyı bu yönde uygulamak daha tercih edilebilir bir yöntemdir. Böylelikle nüfusta tutulacak örnekler yaratılacak ve değerlendirilecektir. İlk yöntemde olduğu gibi nüfusa katılmayacak örnekler yaratmak ise ekstra hesap yükü getireceğinden çoğu zaman tercih edilmez.

2.1.3.2 Seçme

Seçme işlemini gerçekleştirmekteki ana amaç iyi çözümleri destekleyip kötü çözümlerden kurtulmaktır. Bu işlem, doğadaki uygun bireyin kurtuluşu (survival of the fittest) prensibini temel alır. Koşullara uyan, istenen çözüme en yaklaşan bireyler nüfusta tutulur; koşullara en ayak uyduramayan, çözüme en uzak olan bireyler ise yok olur. Bütün bunlar olurken nüfus sayısı sabit kalır. Seçme işleminin üç genel adımı vardır:

- Nüfustaki uygun bireyler teşhis edilir.
- Uygun bireyler kopyalanır.
- Uygun olmayan bireyler yenileriyle değiştirilir.

Yeni nesillerin oluşmasının gerekliliği çaprazlama işleminde hangi bireylerin kullanılacağını seçmeyi gerektirir. En genel seçme işlemi üst nüfustaki tüm bireylerin uygunluk değerlerini toplayıp, herhangi bir bireyin uygunluk değerini bu toplam değere bölerek yapılır. Bu yöntemle bireylerin seçilme olasılığı hesaplanmış olur. Fakat bu türlü bir olasılık atama tekniği kullandığında genetik algoritma kimi fonksiyonla karşılaştığında garip davranabilir. Örneğin iki fonksiyon göz önüne alalım: $y = ax^2$ ve $y = ax^2 + b$. b değeri çok büyük olduğu takdirde seçimin doğruya daha yakın yapılması zorlaşır (Baeck, vd., 2000).

Bir nüfus içindeki en uygun bireyleri seçip diğerlerini tamamen gözden çıkarmak da çekici gelebilir. Fakat bu da çeşitliliğin azalmasına neden olur. Çeşitliliğin azalması ise uzun vadede işimize gelmeyecektir çünkü başka diğerleri kadar iyi sonuçlar vermeyen bireylerin uzun vadede işe yarayacak özelliklerinin daha sonraları gözlenmesi mümkündür. Daha sonra destekli öğrenmede üzerinde durulacak olan keşif – sömürme pazarlılığı bir anlamda burada da gündeme gelmektedir. Kısaca, keşif uzayı iyi sonuçlar veren bireyler bulundurduktan sonra bile aramaya devam edilmek istenebilir. Bunun nedeni yerel uygunluk gösteren bir noktaya takılmamaktır. Bu nedenle farklı seçim yöntemleri geliştirilmiştir. Dört bilindik yöntemden bahsedebiliriz:

- Turnuva Seçimi (Tournament Selection): Nüfustan küçük bir altküme kadar birey rasgele seçilir ve bu kümenin en başarılılarından biri veya ikisi eşleşme bölgesi (mating buffer) için ayrılır (Baeck, vd., 2000). Bu işlem defalarca tekrarlanır. Turnuvalar genelde iki birey arasında yapılır ama bu sayıyı arttırmak da mümkündür (Blickle, 1995).

- Seçkinci Strateji (Elitist Strategy): En başarılı bulunan bireylerin birer kopyası gelecek kuşağa aynen aktarılır (Baeck, vd., 2000).

- Derecelendirme Stratejisi (Ranking Selection): Bireyler uygunluk değerlerine göre sıralanırlar. En uygun olan bireye N, en uygun olmayan bireye ise 1 olmak üzere bireylere dereceler atanır. Bireylerin seçilme olasılıkları aldıkları derecelere göre hesaplanır

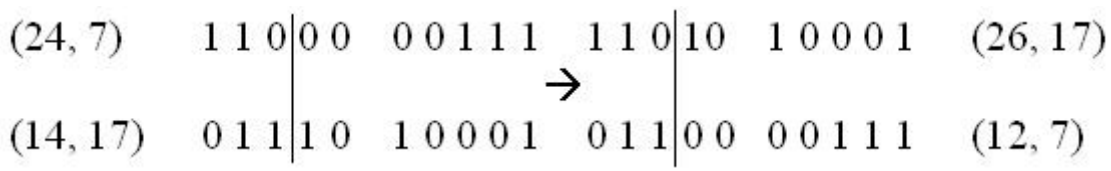
(Blickle, 1995).

- Rulet Tekerleği Seçimi (Roulette Wheel Selection): Her kromozoma dairesel bir rulet tekerleği parçası atanır. Her parçanın alanı atandığı kromozomun uygunluk oranına denktir. Hangi kromozomun eşleşeceğini seçebilmek için [0,100] aralığında rasgele bir rakam seçilir. Alanı bu rakamı içeren kromozom eşleşme için seçilmiş olur (Negnevitsky, 2002). Rulet Tekerleği seçimi genellikle tercih edilir bir yöntem de olsa uygunluk fonksiyonuna çok bağımlı olduğundan dolayı ölçeklendirme konusunu gündeme getirir. Bu nedenle onun yerine derecelendirmeye dayalı bir seçme yöntemi bu tür çalışmalara daha uygundur (Gahot, 2005).

Evrimsel yaklaşım birçok rasgele işlemden yararlanıyor da olsa sonucun rasgele olması beklenmez. Böylelikle yaratılan yaşam formlarının çevreyle ilgili herhangi bir ön bilgisi olmasa da problem verildiğinde (uygunluk fonksiyonu belirlendiğinde) olası en uygun sonuçları vermeleri beklenir.

2.1.3.3 Üreme

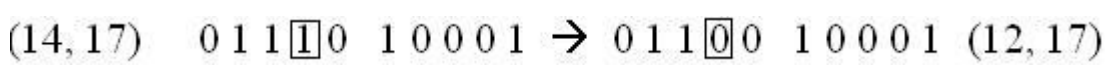
Seçme işlemi tek başına nüfusu yenileyemez, ancak o kadar uygun olmayan bireylerin yerini alacak şekilde uygun bireyleri kopyalar. Nüfusu yenilemek için ise yeni bireylerin yaratılmasına ihtiyaç vardır. Genetik Algoritma'da şimdiki örneklerden yenilerini üretebilmek için kullanılan iki temel genetik işleçten söz edilir; bunlar çaprazlama (crossover) ve mutasyon işleçleridir. Çaprazlama işleci, basitçe anlatmak gerekirse, iki örneğin dizilerini birleştirerek yeni dizili yeni bir örneğin yaratılmasını sağlar. Çoğu çaprazlama işlecinde, eşleşme havuzundan (mating pool) kromozomları temsil eden iki gen dizisi rasgele seçilir ve bu dizilerin bazı kısımları karşılıklı değiştirilir. Çaprazlama işlecinin uygulamanın çeşitli yolları vardır. Genel olarak kullanılan bir biçimi tek noktalı çaprazlamadır (1-point crossover). Tek noktalı çaprazlama işleci her iki üst diziyi ikişer alt diziye böler. Her iki üst dizi de rasgele seçilmiş olan aynı bölgeden bölünürler. Bu bölgenin sağında kalan alt diziler iki yeni örnek oluşturmak amacı ile aralarında yer değiştirirler. Bu işlem Şekil 2.2'de gösterilmiştir. Bu örneğe göre değiştirme bölgesi olarak üçüncü konum seçilmiştir. Üçüncü kısmın sağında kalan bölge karşılıklı değiştirilmiştir. İşlemin sonucunda iki yeni birey oluşmuştur.



Şekil 2.2 Çaprazlama örneği

Her çaprazlama sonucunda yeni bireyler eskilerden (çaprazlanmış olanlar) daha iyi olacak diye bir şart yoktur. Bunun tersi de zaman zaman doğru olabilmektedir. Fakat daha iyi sonuçlar elde etme ihtimali rasgele duruma göre daha fazladır. Ayrıca, yeterince uygun olmayan bireylerin daha sonraki nesillerde yok olma ihtimali yüksektir. Bunun dışında yeni üretilen bireyler yeterince iyilerse Genetik Algoritma'nın bir sonraki döngüsünde yeni nesillerin yaratılmasına yol açma olasılıkları yüksektir, böylelikle iyi bireylerinkine benzer kromozomların nüfusa katılması beklenilir. Bu prensip de zaten biyologların ve evrim kuramcılarının organizmaların basit yapılardan daha karmaşık yapılara geçiş sürecine bakış açılarını yansıtır niteliktedir. Çaprazlama, aynı zamanda, genel en iyi sonuca ulaşılması olasılığını yükseltir ve çabuk yakınsama sağlar.

Çaprazlama işlecinin dezavantajı nüfus çeşitliğini tek başına yeterince sağlayamamasıdır. Hatta tek başına kullanıldığında bir süre sonra bütün örneklerin birbirine benzediği ve bundan dolayı çaprazlamanın yeni örnekler yaratamayacağı bir nüfus oluşumuna neden olabilir. Benzer şekilde, gen dizilerindeki belli bir bölgedeki değer bütün örnekler için aynıysa sadece çaprazlama işleci kullanılarak bahsi geçen geninin değeri farklı olan bir örnek yaratmak mümkün olamayacaktır. Böyle durumlardan sakınabilmek ve nüfusun çeşitliliğini devam ettirebilmek için mutasyon işlecinin de çaprazlamaya ek olarak kullanılması gerekir. Dolayısı ile genelde çaprazlama işlemi gerçekleştirildikten sonra mutasyon işleci çalıştırılır. Bu işleç kromozomun gen dizisini yerel değişime maruz bırakır. Mutasyon işleci kısaca üst örneğin bir kopyasını yapar ve dizinin bazı elemanlarının değerlerini değiştirir. Her dizi elemanının değiştirilme olasılığı mutasyon oranı adı verilen sabit bir değer ile tanımlanır. Bu yeni dizi daha iyi sonuçlar verecek diye bir şart yoktur ama öyle olması umulur. Bu işlem sonunda da, daha önce de bahsedildiği gibi, kötü sonuçlar elde edilirse onları eleme işi seçme işlecine bırakılır. Mutasyonun amacı nüfusta çeşitlilik sağlamaktır fakat mutasyonun da daha önce bulunmuş güzel gen değeri sıralarını bozma olasılığı gibi olumsuz bir tarafı vardır. Mutasyon işlemi Şekil 2.3'te örneklendirilmiştir.



Şekil 2.3 Mutasyon örneği

Çaprazlama yeni ve benzersiz örnekler yaratabilme özelliği sayesinde Genetik Algoritma için vazgeçilmez bir işleçtir. Buna karşın, nüfustaki farklılıklar azaldıkça çaprazlama işlecinin benzersiz örnek yaratabilme yeteneği de azalmaktadır. Bunu takip eder şekilde, benzersiz örnek yaratılma ihtimali azaldıkça da nüfusun çeşitliliği azalacaktır. Bu durumdan da anlaşılabileceği gibi, Genetik Algoritma'da nüfusun çeşitliliğini mutasyon ile sağlayabilmek

meselesi ciddi bir konudur. Nüfusun boyutu ve mutasyon oranı çeşitliliğin değişimindeki oranı etkileyebilecek ana etmenlerdir. Nüfus azaldıkça bütün örneklerdeki belli bir genin aynı değerde kalma olasılığı artar (Frankham, 1996). Buna göre, çeşitliliğin azalmasını engellemek için, nüfus küçüldükçe mutasyon oranının ona bağlı olarak artması gerekmektedir. Seçici baskı ve seçme yöntemleri de nüfusun farklılığındaki değişimi etkiler fakat bu etmenler olmasa bile nüfusun farklılığı zamanla azalma eğilimi gösterir.

2.1.3.4 Kodlama

Genetik Algoritma'nın diğer evrimsel öğrenme yöntemlerinden temel farkı nüfustaki örnekleri diziler şeklinde ifade etmesidir. Bu gösterim tarzı doğada var olan organizmaların özelliklerini kodlayan DNA kıvrımlarına benzerlik gösterir. Bu tür bir kodlama, Genetik Algoritma'nın dizileri değiştirmeyi amaçlayan genetik işleçler kullanmasına olanak tanır. Böylelikle nüfusta yeni örnekler belirir. Bahsi geçen işleçler, üreme sırasında organizmaların DNA'larında rastlanılan tipteki işlemlerin bir benzerini gerçekleştirirler. Bu yaklaşımın önemli bir avantajı genelliğidir (Goldberg, 1989); dizi gösteriminin kullanılabileceği problemler geniş bir alana yayılmıştır. Bu avantajının yanında, genetik işleçlerle yeni örneklerin yaratılabilmesi arama algoritmasının, örneklerin kodlanma biçiminden ve uygulanan problemden bağımsız olmasını sağlar. Her bir Genetik Algoritma'da nüfustaki örnekler sabit uzunlukta diziler taşır. Bu diziler genelde genotipik gösterimini ifade eder.

Kromozomdaki genleri farklı şekillerde temsil etmek mümkündür. Bu temsil yöntemlerinden biri her genin ya 0 ya da 1 değerini alabileceği ikili kodlama yöntemidir. Bu yöntem en genel kodlama yöntemi olmakla birlikte kimi problemlerde kullanılması tercih edilmez; onun yerine genlere direk olarak gerçek sayı değerleri atanabilir. Bu kodlama yöntemine gerçek sayı kodlaması denir. Gerçek sayı kodlamasına göre, r 'nin herhangi bir gerçek sayıyı temsil ettiği kabul edilirse, her bir gen $[-r,r]$ aralığında bir değer alır. Bir başka kullanılan yöntem ise çok karakterli kodlamadır (many–character encoding). Bu yöntemde kromozomu oluşturmak için birden fazla karakterden oluşan bir alfabe kullanılır (Mitchell, 1996). Bunlar dışında başka kodlama yöntemlerinden de yararlanmak mümkündür.

2.2 Yapay Sinir Ağları

Denetimli öğrenmede eğitmek için bir öğretici gerekir. Öğretici, basitçe, çıktının ne olması gerektiğini söyler. Verilen girdi ile çıktı üretilir, üretilen çıktı doğru çıktı ile karşılaştırır. Sistem, hataya göre kendini yeniden ayarlar. Bu işlem, kabul edilebilir bir hata seviyesine

erişinceye dek devam eder. İleri Beslemeli Sinir Ağları'nın aksine Yinelenen Yapay Sinir Ağları eski girdilere duyarlı olabilir ve onlara uygun davranabilir. Bu bölümde İleri Beslemeli Sinir Ağlarında kullanılan Geri Yayılımlı Öğrenme yöntemi anlatılmış, yapılandırılan yaşam formuna uygunluğu denenmiş, daha sonra da yapı genişletilerek Yinelenen Yapay Sinir Ağları ve onun eğitilmesinde kullanılan Gerçek Zamanlı Yinelenen Öğrenme üzerinde durulmuştur.

2.2.1 Geçmiş

Yinelenen Yapay Sinir Ağları kontrollü robotlar kimi araştırma alanlarının ilgi konusu olmuşlardır. Yapay Sinir Ağları genel olarak robot kontrolü problemine uyumlu olmalarını sağlayacak özelliklere sahiptirler. Genelde gürültüye dayanıklı sayılırlar ve eksik verileri kullanabilirler. Sinir ağlarının bu özelliği onlara, geleneksel yöntemlerle karşılaştırıldığında, dinamik bir çevredeki belirsizliklerle ve algılayıcı ölçümlerindeki sınırlandırmalar benzeri sorunlarla daha çok baş edebilme olanağını tanır (Ziemke, 1999). Dahası, Yapay Sinir Ağları modelden bağımsız yöntemlerdir; bu yöntemle eğitilen robotlar çevreyle olan etkileşimlerinden yeni şeyler öğrenebilir. Yapay Sinir Ağları'nın esnekliği sayesinde robot tasarımcıların işlerinin kolaylaştığı söylenebilir.

Çoğu seyyar robot için çevre çok kolay bir yapı taşımaz. Çevredeki birçok bölge birbirine benzeyebilir. Öyle ki robotun önceki konumunu hatırlamadan bu yeni bölgenin ayırtına varılamaz. Ayrıca robot, bu tezde ele alındığı şekli ile etmen, problemleri daha çok, açık ve kesin girdi çıktı çiftleri yerine soyut hedefler üzerinden tanımlanırlar (Meeden, 1996). Meeden (1996) Yinelenen Yapay Sinir Ağlarını incelemiş ve bu yapının, yalnızca şimdiki girdiye cevap verebilmekten öte, belirli hareket stratejilerine denk gelen davranış dizileri icra ederek içsel durumdan yararlanabildiğini göstermiştir.

1950lerin ortalarında başlayan geleneksel yapay zeka araştırmaları Grey Walter'ınki (1950, 1953) gibi robotik alanındaki önceki çalışmaları genel olarak göz ardı etmiştir (Ziemke, 1999). Bunun yerine yapay zeka beynin modellenmesi alanına yönelmiştir. O zamanlar genel kanı vücudu olan bir beyinin gerçekleştirilmesinden pek fayda elde edilemeyeceği yönündedir. Dolayısıyla yapay zeka çalışmaları etmenin çevresiyle olan etkileşiminden büyük ölçüde ayrılmıştı (Ziemke, 1999). 1980'lerin ortalarına kadar yapay zeka alanında robotlara karşı ilgi oldukça az olmuştur. 1980'lerde ise birçok yapay zeka araştırmacısı etmenler ve çevreleri arasındaki etkileşimi incelemeye başlamışlardır. Brooks ve Wilson gibi araştırmacılar yapay zekaya aşağıdan yukarıya bir yaklaşım önermişlerdir (Ziemke, 1999).

Buna göre yapay zekanın bundan böyle otonom etmenler ve çevreleri arasındaki etkileşime yapay sinir hücreleri ve hareket vasıtasıyla yaklaşması üzerinde tartışılmıştır.

1990larda robotları daha da otonom yapabilmek için çaba sarf edilmiştir. Bu amaçla geliştirilen öğrenme teknikleri genelde etmenlerin içsel parametrelerini kontrol mekanizmalarına adapte edebilme üzerinedir. Böylelikle etmenlerin öğrenmesini veya bulundukları ortama uyumunu sağlamak amacı ile Yapay Sinir Ağlarının kullanımı yaygınlaşmaya başlamıştır. Bundan sonra ise yapay sinir ağları ile kontrol edilen etmenler mühendislik ve yapay zeka alanında önem kazanmakla kalmayıp zihinsel bilimin (cognitive science) cisimsel (embodied) ve mekansal (situated) doğasını incelemek için de kullanılabilecek bir yaklaşım olduğunu göstermiştir (Clark, 1997, Sharkey, 1998).

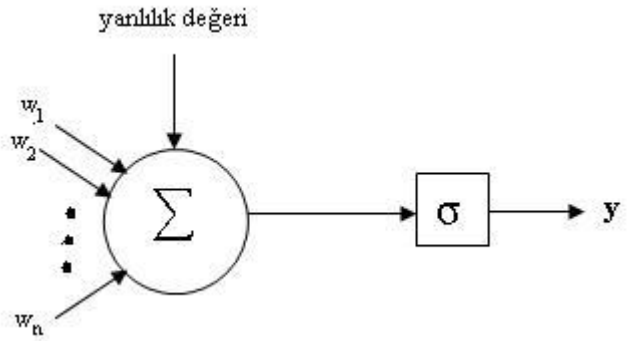
Peschl (1996) Yinelenen Yapay Sinir Ağlarının gerçek sinir sistemlerine benzer şekilde yapı kararlı (structure determined) olduğuna dikkat çekmiştir. Bunun anlamı bu tür sinir ağlarının çevresel etkilere karşı tepkilerinin her zaman sistemin o andaki durumuna bağlı olduğudur. Dolayısı ile tek başına bir girdi sistemin tepkisini tek başına belirleyemez.

2.2.2 Genel Bakış

Bir Yapay Sinir Ağı birbirine bağlı işlem öğeleri yani nöron topluluğudur (Haykin, 1994). Tarihi 1940lara kadar uzanan Yapay Sinir Ağı çalışmaları, adından da anlaşılacağı gibi, büyük ölçüde, hayvanların beyinlerinde bulunan biyolojik sinir ağlarından esinlenmiştir. Yapay Sinir Ağları karmaşık problemleri daha kolay problemlere bölerek çalışırlar. Doğru şekilde eğitildikleri takdirde de çoğu zaman doğru bir genelleştirme yapabilirler. Genelleştirmenin doğru olup olmadığının anlaşılabilmesi için YSA'nın öğrenme sırasında eğitim kümesinden olmayan bir girdiye nasıl bir sonuç verdiğine bakılabilir. Çıktılar çoğu zaman mantıklı ise genelleştirme kabul edilebilir nitelikte demektir. YSA'larının bir diğer avantajı gürültüye dayanıklı olmalarıdır. Birimleri ağırlıklı sinyallerin toplamına dayandığı için, sinyallerin bireysel değerleri sinir ağının davranışını çok etkilemez (Nolfi ve Floreano, 2000). YSA bilim, teknoloji, ekonomi gibi çeşitli alanlarda ve görüntü algılama, zaman dizisi tahmini gibi pek çok uygulamada kullanılmaktadır (Wahde, 2003). YSA robotlarda da otonom davranış sağlayabilmek amacıyla kullanılmıştır. Robotlar genelde gürültülü algılayıcılara sahiptirler ve gürültülü çevrelerle etkileşim halinde olabilirler. Bu gibi durumlarda YSA'larının dayanıklılık özelliği faydalı bir özellik olarak kendini gösterir (Nolfi ve Floreano, 2000).

Yapay Sinir Ağı, verilen eğitim kümesi ile karmaşık problemleri çözebilme özelliğinde sahip, parametrik olmayan bir kestiricidir (Uğurlu ve Biricik, 2006). Biyolojik sistemlerdeki sinir hücrelerinin çalışma prensiplerini ve birleşimlerini örnek alan bir bilgi işlem sistemi olan YSA, eğitim kümesini muhakeme ederek özelliklerini öğrenir ve aralarındaki ilişkiyi modeller (Mitchell, 1997). Bu modelleme sonucunda karmaşık problemlere, detaylarını fazla bilmeden bile çözüm üretebilir.

Genel sinir ağı modeli, biyolojik nöronları temsil eden işlem elemanları ve bunların birbirlerine bağlantılarını sağlayan ağırlıklardan oluşur. Ağın yapısına bağlı olarak, ağırlıklar belirli bir giriş serisi için verilen bir çıkış serisini en iyi oluşturacak şekilde ayarlanır. Bir nöronun ağırlığının değişmesi, o nöronun "öğrendiğini" gösterir. Nöronlar ağırlık, birleştirme fonksiyonu ve etkinleştirme fonksiyonu (activation function) şeklinde üç bileşene sahiptir. Ağırlıklar, diğer nöronlardan gelen bilginin değerini gösterir. Diğer bir deyişle, ağın sakladığı bilgi ağırlıklarda tutulur. Ağırlıklı girdileri birleştiren birleştirme fonksiyonu ve bu fonksiyonun sonucunu değerlendiren etkinleştirme fonksiyonu nöronun diğer işlem elemanlarıdır (Alpaydın, 2004).



Şekil 2.4 Bir yapay sinir hücresinin yapısı

Yapay Sinir Ağlarındaki nöronlar birbirlerine bağlıdırlar. Nöronların bağlantılarının kuvveti bağlantı ağırlığı olarak adlandırılan ve pozitif veya negatif bir değer alabilen bir değişken ile belirtilir. Her nöron diğer nöronlardan aldığı ağırlıklı girdileri ve bir de sapma (bias) değerini toplar ve sonucu etkinleştirme fonksiyonuna gönderir. Nöronun yapısı, anlatıldığı şekli ile, şekil 2.4'te gösterilmiştir.

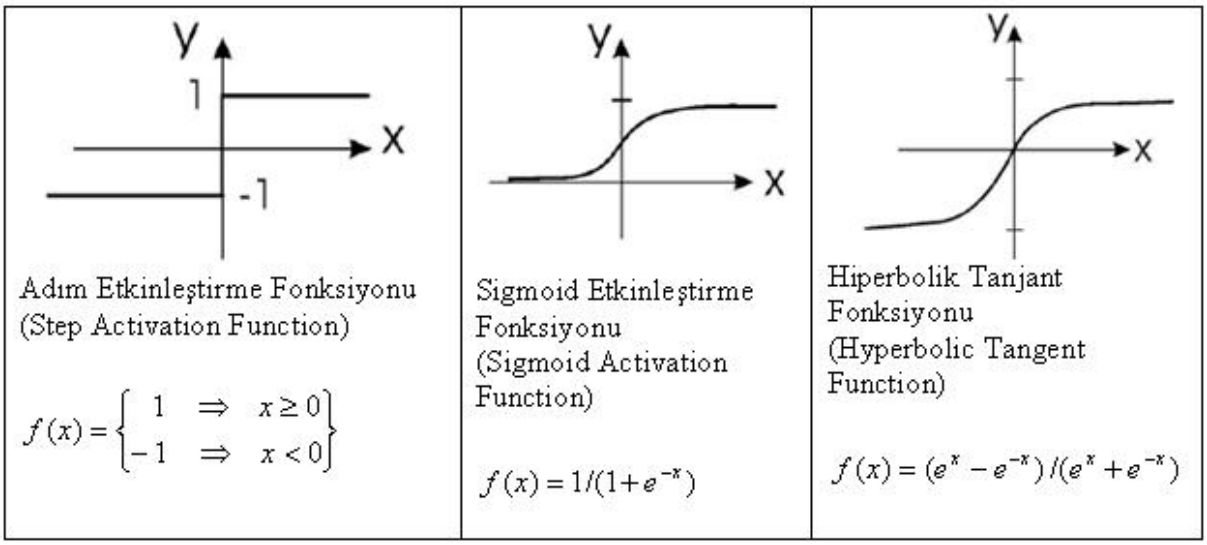
Yapay sinir hücresi girdilerini başka bir çevreden veya başka bir YSA hücresinin çıktısından alabilir. Bir YSA hücresinde her girdi bir bağlantı ağırlığı, diğer bir adıyla sinaptik (synaptic) ağırlık ile ilişkilendirilir. En basit durumda çıktı girdilerin ağırlıklı ortalamasından oluşur. Bu durum eşitlik (2.1)'de ifade edilmiştir.

$$o = \sum_{j=1}^{d} w_j x_j + w_o \tag{2.1}$$

Bu denklemde X ($X = x_1, x_2, x_3, ..., x_d$) girdiler matrisini, W ($W= w_1,w_2, w_3, ..., w_d$) ağırlıklar matrisini, w_o ise sapma değerini simgelemektedir. Çıktıyı hesaplayabilmek için ilk katmandan çıkan sonuçlar bir etkinleştirme fonksiyonundan, f, geçirilir. Bu adım (2.2) eşitliğinde gösterilmiştir.

$$y = f\left(\sum_{j=1}^{d} w_j x_j + w_o\right) \tag{2.2}$$

Problemin yapısına göre çeşitli etkinleştirme fonksiyonlarından yararlanılabilir. Etkinleştirme fonksiyonu nöronun çıkışını, çoğunlukla [0,1] veya [-1,1] olmak üzere, belli bir aralığa eşler. Genelde Sigmoid veya Hiperbolik Tanjant gibi doğrusal olmayan bir etkinleştirme fonksiyonu kullanılır. En bilindik etkinleştirme fonksiyonları Şekil 2.5'te özetlenmiştir.



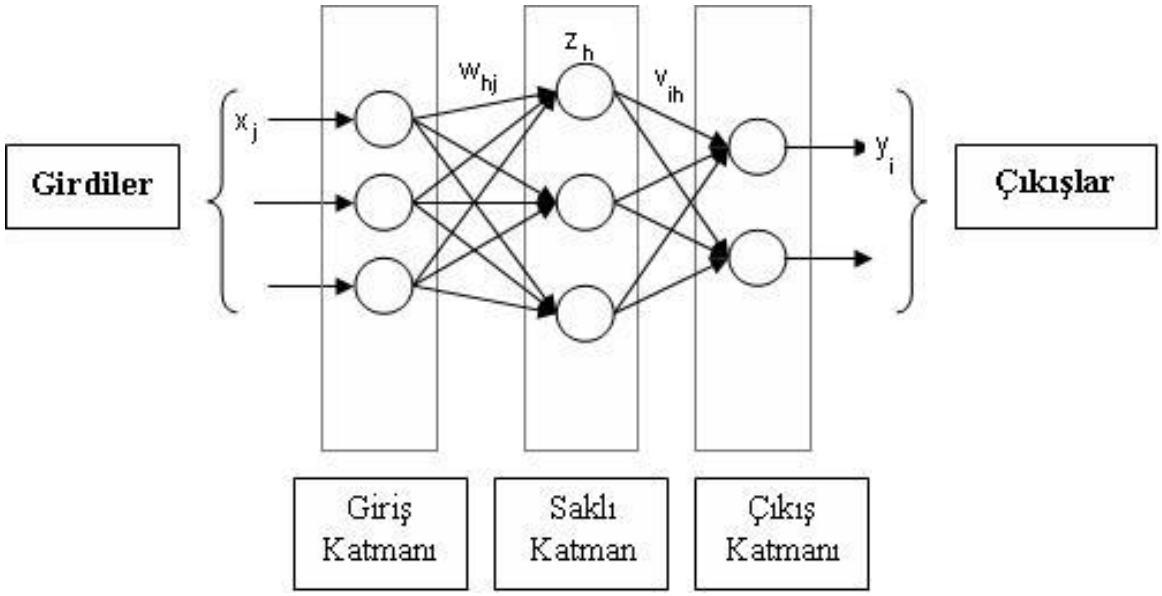
Şekil 2.5 Etkileşim fonksiyonları

Gösterilen etkinleştirme fonksiyonları en genel kullanımlı etkinleştirme fonksiyonları olmakla birlikte bunlardan başka fonksiyonlar da vardır. YSA'da kullanılan etkinleştirme fonksiyonu probleme göre değişkenlik gösterir.

2.2.3 İleri Beslemeli Sinir Ağları

En genel Yapay Sinir Ağı türü İleri Beslemeli Sinir Ağlarıdır. İleri beslemeli sinir ağlarında genellikle girdi, ara ve çıktı olmak üzere birbirlerine ağırlıklarla bağlanan üç katman kullanılır. İleri beslemeli çok katmanlı yapay sinir ağları, çok boyutlu uzayda eğitim kümesine en iyi uyan veri için bir yüzey bulmaya çalışırlar. Girdiler, birçok ara nöronun birlikte aktif hale gelmesi ile kodlanırlar. İleri beslemeli bir sinir ağında Şekil 2.6'da da gösterildiği gibi

sinyaller girdi katmanından çıktı katmanına doğru katmanlaşan bir yapıya sahiptir.



Şekil 2.6 İleri Beslemeli Sinir Ağı

Çok katmanlı sinir ağları birden çok sinir hücresinin birbirine bağlanmasıyla ve adından da anlaşılacağı gibi birden çok katmandan oluşurlar. İlk katman giriş katmanı, son katman çıkış katmanı, aradaki katman(lar) ise saklı katman(lar) olarak adlandırılır. Kullanılacak saklı katman ve her katmandaki nöron sayısı, problemine göre değişkenlik gösterir. Bu sayılara ancak deneme yanılma yoluyla karar verilir (Alpaydın, 2004).

İleri Beslemeli Sinir Ağları, büyük olasılıkla çok miktarda, saklı düğümden (hidden node) faydalandığı takdirde uzamsal olarak sınırlı her tür fonksiyonu yaklaşıklaştırmak için kullanılabilir. Bu, girdi uzayı sabit olduğu sürece sinir ağlarıyla bu fonksiyonları kodlamanın bir yolunu bulmak olası demektir. Eğer yeterince örnek mevcut ise sinir ağının ağırlıklarının bulunabilmesi için Geri Yayılımlı Öğrenme gibi otomatik öğrenme yöntemleri kullanılabilir.

2.2.4 Geri Yayılımlı Öğrenme

Geri Yayılımlı Öğrenme YSA'lar üzerinde kullanımı yaygın olan etkin bir öğrenme yöntemidir. Bu öğrenme yöntemine göre önce yayılma (propagation) işlemi daha sonra da uyum gösterme (adaptation) işlemi gerçekleşir. Geri Yayılımlı Öğrenme denetimli bir öğrenme yöntemidir ve İleri Beslemeli (Feed Forward) çok katmanlı (multilayer) yapılı ağlar üzerinde çalıştırılır. Geri Yayılımlı Öğrenme yönteminde hatalar çıkıştan başlayarak girişe doğru yani geriye doğru azaltılmaya çalışılır. Hatanın bu şekilde geriye doğru yayılabilmesi için öncelikle ağ çıkışındaki hata tespit edilir, daha sonra bu hata yayılarak her katmandaki ağırlıkların değişmesine neden olur. Bu nedenle her bir katmandaki ağırlıklar yeniden

hesaplanır (Mitchell, 1997). Geri Yayılım Algoritması Şekil 2.7'de verilmiştir.

Bütün v_{ik} ve w_{kj} ağırlıklarına (-0.01, 0.01) aralığındaki bir değeri rasgele ata
Yakınsayana kadar tekrarla
  Her rasgele $(x^t, r^t) \in X$ için
    h = 1.....H 'ye kadar
      $z_h \leftarrow f(w_h^T x^t)$
    i = 1.....K ' ya kadar
      $y_i = v_i^T z$
    i = 1.....K ' ya kadar
      $\Delta v_i = \eta (r_i^t - y_i^t) z$
    h = 1.....H 'ye kadar
      $\Delta w_h = \eta (\sum_i (r_i^t - y_i^t) v_{ih}) z_h (1 - z_h) x^t$
    i = 1.....K ' ya kadar
      $v_i \leftarrow v_i + \Delta v_i$
    h = 1.....H 'ye kadar
      $w_h \leftarrow w_h + \Delta w_h$

Şekil 2.7 Geri Yayılım Algoritması (Alpaydın, 2004)

Bu algoritmada x_j ağın girdilerini, z_h saklı katmanın girdilerini, y_i ağın çıktılarını, r_i ağdan elde edilmesi istenen sonucu, w_{hj} birinci katmanın ağırlıklarını, v_{ih} ikinci katmanın ağırlıklarını, H saklı katmanın boyutunu, K çıktı sayısını, f aktivasyon fonksiyonunu ve h öğrenme katsayısını temsil etmektedir.

Bir sinir ağının eğitilmesi aşamasında iki diziden yararlanılır. Bunlardan ilki girdi dizisi diğeri ise ona denk gelen çıktı dizisidir. Sinir ağı önce boyutu önceden belirlenen bir eğitim kümesi ile eğitilip, bu eğitimden elde edilen sabit ağırlıklarla yeni durumu öğrenmeye çalışırsa uyarlanabilirliği olmayan bir öğrenme politikası izlemiş demektir. Buna karşılık, sinir ağı kullanıldığı sürece eğitiliyorsa uyarlanabilir bir öğrenme politikası izliyordur. Bu durumda eğitimin kümesinin sonsuz uzunluğa sahip olduğu söylenebilir (Nerrand vd., 1994).

Eğitim uyarlanabilir olmadığında eğitim fonksiyonu eğitim kümesindeki tüm verinin fonksiyonu olarak tanımlanır. Bu fonksiyon ceza fonksiyonu (cost function) diye de adlandırılır. Ceza fonksiyonu minimum değerine ulaştığında sistem gösterebileceği en iyi performansı sergiliyor demektir. Eğitim gradyan temelli yöntemler kullanan bir eniyileme yordamıdır. Uyarlanabilir eğitimde ise, en aza indirgenmesi, performans ölçütüne göre en iyilenmiş sayılabilecek bir sistem oluşturabilen, zamandan bağımsız bir ceza fonksiyonu

tanımlamak çoğu problemde mümkün değildir. Bundan dolayı eğitim fonksiyonu zamana bağlıdır (Nerrand vd., 1994).

2.2.5 Yinelenen Yapay Sinir Ağları

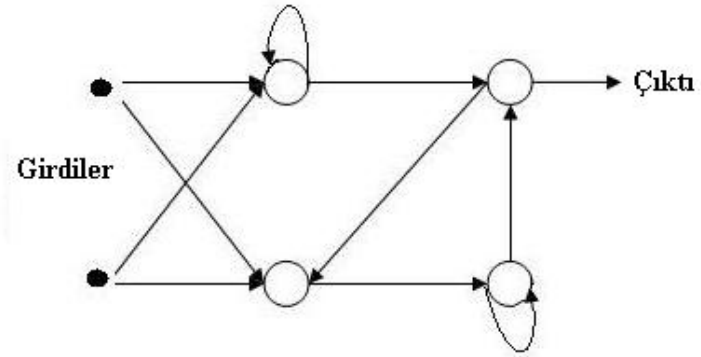
Sinir ağlarının, kontrol mühendisliği alanındaki uygulamalarda, özellikle de doğrusal olmayan sistemleri modellemede faydalı olabileceği düşüncesi kabul edilmiştir. Sinir ağlarının önemli özelliklerinden bazıları, örnekler üzerinden öğrenme kabiliyetleri, ön bilgiye ihtiyaç duymamaları ve doğrusal olmayan sürekli fonksiyonları iyi yaklaşıklayabilmeleridir. Literatür incelendiğinde dinamik öğeler ve içsel geribildirim bağlantıları bulunduran Yinelenen Yapay Sinir Ağlarının bahsedilen bu özellikleri karşılamaya İleri Beslemeli (feedforward) Sinir Ağlarından daha uygun olduğu görülmüştür (Linkens ve Nyongesa, 1996).

Yapılan çeşitli çalışmalarda da YYSA'larının doğrusal olmayan dinamik sistemleri modellemede oldukça etkili olduğu gösterilmiştir (Draye vd., 1996). YYSA'ları üzerinde düşünülmesi gereken konu ağ mimarisinin seçimi, yani nöronların sayısı ve tipi, geribildirim döngülerinin yerleşimi ve uygun bir eğitim algoritmasının oluşturulmasıdır.

Gerçek dünyada modellemek istediğimiz birçok sistem genelde doğrusal olmayan dinamik sistemlerdir. Bu duruma daha çok uçak, motor, robot gibi sitemlerin ileri ve ters dinamiklerini modellemek istediğimiz kontrol alanında rastlanılır. Bu sistemlerin birçoğu aynı zamanda durum bağımlı davranış sergilemeye meyleder.

Sinir ağları doğrusal olmayan dinamik sistemlere veya durum bağımlı sistemlere uygulandığı bazı problemlerde, geribildirim bağlantılarına sahip olan sinir ağları diğerlerine göre önemli ölçüde avantaj sağlayabilirler. Geribildirim bağlantıları özyineli hesaplama sağlar ve durum bilgisini tutabilmeyi olanaklı hale getirir. Bazı durumlarda, geribildirim bağlantıları olan bir sistem çok daha geniş çapta ve sınırsız olabilecek bir ileri beslemeli siteme denk gelebilir. Geribildirim bağlantılarına sahip yapıdaki sinir ağlarına Yinelenen Yapay Sinir Ağları denir.

YSA'nın 1980'lerdeki yeniden doğuşundan beri YYSA'nın yeteneklerini ve sınırlarını araştıran yadsınamaz miktarda çalışma yapılmıştır. Buna rağmen, YYSA İleri Beslemeli Sinir Ağlarına yaygınlık açısından yaklaşamamıştır. Bunun ana nedenlerinden biri Yinelenen Sinir Ağları için genele uygulanabilen öğrenme algoritmaları geliştirmenin zorluğudur. İleri Beslemeli Sinir Ağlarının bu anlamdaki en önemli avantajı Geri Yayılım gibi bayır inişi (gradient decent) en iyileştirme algoritmaları kullanılarak eğitilebilmeleridir (Rumelhart ve McClelland, 1986).



Şekil 2.8 Bir yinelenen yapay sinir ağının yapısı

Yinelenen Yapay Sinir Ağları'nda her nöronun başka bir nörona bağlantısı bulunabilir, hatta nöronlar kendilerine de bağlanabilirler. Bu yapı Şekil 2.8'de gösterilmiştir. Her nöronun kendine ve diğer nöronlara bağlanabiliyor olması demek YYSA'nda ileri beslemelilerde olduğu gibi doğal bir çıktı olmadığı anlamına gelir çünkü YYSA'nda İleri Beslemeli Sinir Ağlarındaki çıktı katmanına denk gelen bir katman bulunmamaktadır. Bu nedenden dolayı, hangi nöronların çıktı olarak kullanılacağına kullanıcı karar vermek durumundadır. İleri Beslemeli Sinir Ağları önceki girdileri göz ardı ederek verilen bir girdiye karşılık aynı çıktıyı üretirken YYSA kısa vadeli bellek de sağlar. Bu demektir ki YYSA'nın tepkileri sadece bağlantı ağırlıklarına değil aynı zamanda önceki girdi sinyallerine de bağlıdır. Önceki girdi sinyallerine bağlılığın sağlanması için sinir ağının geribildirim bağlantılarının bulunması gereklidir.

Yinelenen Yapay Sinir Ağlarında kullanılan iki temel bayır inişi algoritması bulunmaktadır. Bunlardan biri Zaman İçinde Geri Yayılım (Backpropagation Through Time), bir diğeri ise Gerçek Zamanlı Yinelenen Öğrenme'dir (Real Time Recurrent Learning). Bu öğrenmenin diğer adları Dinamik Yayılma (Dynamic Propagation) ve Yinelenen Geri Yayılmadır (Recursive Backpropagation) (Sayed, 1995).

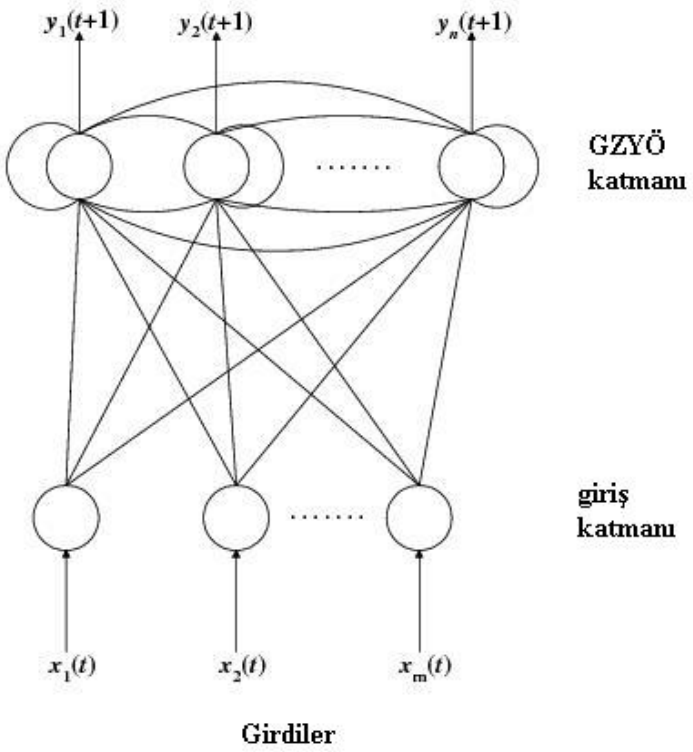
Zaman İçinde Geri Yayılım Algoritması sinir ağının katmanlarını zaman içinde açarak ağı geribildirimli bir sistemden ileri beslemeli bir sisteme dönüştürmeyi hedefler. Dolayısıyla sinir ağında bir sinyal zamanda bir adım ilerlediğinde sinir ağının kopyaları yaratılacak ve geribildirim bağlantıları bir sinir ağı ile bir sonraki sinir ağı arasında kurulacak ileri beslemeli bağlantılara dönüşecektir. Dönüşümden sonra artık sinir ağı büyük bir ileri beslemeli sinir ağıymışçasına eğitilebilir. Bu algoritmanın ayrık zamandaki uygulaması Rumelhart ve McCElland' ın (1986) kitaplarında bu yöntemi ve kullanımını açıklamaları ile birlikte popülerlik kazanmıştır. Zaman İçinde Geri Yayılım Algoritması'nın sürekli sürümü ise daha

sonra PearlMutter (1989) tarafından geliştirilmiştir. Zaman İçinde Geri Yayılım Algoritması'nı sınırlayan büyük etmenlerden biri sinir ağının kaç kopyasının yaratılması gerektiği bilgisine sinir ağı katmanlarını açmaya başlamadan önce erişilmesi gerekliliğidir.

Gerçek Zamanlı Yinelenen Öğrenme Algoritması çıktının türevlerini ve hatayı özyineli bir biçimde güncellemeye dayanır. Bu güncellemeler her döngüde hesaplanır. Ağırlıklar ya her döngünün sonunda ya da dönemin (epoch) en son döngüsünden sonra güncellenir. Bu algoritma ayrık zamanlı Yinelenen Sinir Ağları için farklı araştırmacılar tarafından sunulmuştur. Gerçek Zamanlı Yinelenen Öğrenme Algoritması'nın sürekli sürümü ise Pineda (1996) tarafından gerçeklenmiştir. Bu algoritmanın dezavantajı her bir adımda yüksek miktarda hesaplama yapmasının gerekmesidir. Tezde Gerçek Zamanlı Yinelenen Öğrenme'den yararlanılmıştır.

2.2.6 Gerçek Zamanlı Yinelenen Öğrenme

Gerçek Zamanlı Yinelenen Öğrenme Algoritması yinelenen sinir ağlarını eğitmek için William ve Zipser (1989) tarafından geliştirilmiş bir algoritmadır. Gerçek zamanlı yinelenen öğrenme ağı Şekil 2.9'da gösterildiği gibi iki katmandan oluşur. Bu katmanlardan ilki *m* girdi birimi içeren giriş katmanı, diğeri de *n* birimden oluşan GZYÖ katmanıdır.



Şekil 2.9 GZYÖ ağı (Chang ve Mak, 1999)

Gerçek zamanlı öğrenme yöntemini gerçekleştirmek için girdi kümemizi I, diğer birimlerimizi ise U olarak adlandıralım. U, saklı birimlerden veya çıktılardan oluşabilir. Bu tezde U kümesi sadece çıktılardan oluşacaktır. Şu durumda, bu kümeleri ifade etmek istersek, I kümesinin açılımını $I = \{x_k(t),\ 0<k<m\}$, U kümesinin açılımını ise $U = \{y_k(t),\ 0<k<n\}$ şeklinde yapabiliriz. Bu tanımlarda kullanılan *m* harfi girdi, *n* harfi de çıktı sayısını simgelemektedir. Bu bir yinelenen öğrenme algoritması olduğundan, girdi çıktı ayırmaksızın, ağdaki herhangi bir birime erişilmek istenildiği durumlar olacaktır. Bu nedenle, olası birimi formüle etmekte yarar vardır (2.3).

$$z_k(t) = \begin{Bmatrix} \mathrm{x}_k(t) & eger\ k \in I \\ \mathrm{y}_k(t) & eger\ k \in U \end{Bmatrix} \tag{2.3}$$

n satırlı ve *n+m* sütunlu ağırlık matrisini **W** olarak adlandıralım. Birimler girdilerinin ağırlıklı ortalamasının üzerinden etkinleşme fonksiyonlarını çağırırlar (2.4).

$$net_k(t) = \sum_{l \in U \cup I} w_{kl} z_l(t) \tag{2.4}$$

Etkinleşme fonksiyonu (2.5) doğrusal olmayan bir fonksiyondur.

$$y_k(t+1) = f_k(net_k(t)) \tag{2.5}$$

t anındaki harici girdi *t+1* anına kadar hiçbir birimin çıktısını etkilemediğinden kullanılan ağ, ayrık bir dinamik sistemdir.

U kümesindeki birimler daha önceden kendileri için birer hedef belirlenmiş olan çıktılardır. Her girdiye karşılık için ise bir hedef tanımlanmış olması beklenmez. Hedeflerin belirlenmesi, sistemin öğrenmesini denetlemek için gereklidir. Denetimin sağlanabilmesi için öğrenme sırasında yapılan hataları tespit etmek gerekir ki bu hatalar en aza indirilebilinsin. Bu nedenle bir hata fonksiyonu tanımlanmalıdır. Bu hata fonksiyonu, $e_k(t)$, ancak hedef değeri, $d_k(t)$, belli olan çıktılar üzerinde hesaplanabilir. Hedef değeri belli olan çıktıların indekslerinin oluşturduğu kümeye *T(t)* dersek çıktılardaki hatayı (2.6) eşitliği ile tanımlayabiliriz.

$$e_k(t) = \begin{Bmatrix} d_k(t) - y_k(t) & eger\ k \in T(t) \\ 0 & yoksa \end{Bmatrix} \tag{2.6}$$

Hata bu şekilde tanımlandığında tek bir adımdaki fonksiyonu (2.7) eşitliğindeki gibi gösterebiliriz.

$$E(t) = \frac{1}{2} \sum_{k \in U} [e_k(t)]^2 \tag{2.7}$$

En aza indirgenmek istenen hata fonksiyonu ise bulunduğumuz adımdaki hataya kadar olan toplam hata üzerinden çağrılmalıdır (2.8).

$$E_{toplam}(t_0, t_1) = \sum_{t=t_0+1}^{t_1} E(t) \tag{2.8}$$

Toplam hata önceki tüm hataların şimdiki adımdaki hatayla toplamı olduğundan toplam hatanın gradyanı da, aynı şekilde, önceki hataların gradyanı ile şimdiki hatanın gradyanının toplamıdır (2.9).

$$\nabla_w E_{total}(t_0, t+1) = \nabla_w E_{total}(t_0, t) + \nabla_w E_{total}(t+1) \tag{2.9}$$

Ağa yeni girdiler girdikçe, gradyanın değerleri toplanır (2.10). Bu, ağırlıktaki değişimlerin gözlemlenmesi anlamına gelir.

$$\Delta w_{ij}(t) = -m \frac{\partial E(t)}{\partial w_{ij}} \tag{2.10}$$

Ağa bütün girdiler verildikten sonra her bir ağırlık w_{ij}, $\sum_{t=t_0+1}^{t_1} \Delta w_{ij}(t)$ kadar değiştirilir. Bu nedenle, (2.11) eşitliğindeki hesabı gerçekleştiren bir algoritmaya ihtiyaç vardır.

$$-\frac{\partial E(t)}{\partial w_{ij}} = -\sum_{k \in U} \frac{\partial E(t)}{\partial y_k(t)} \quad \frac{\partial y_k(t)}{\partial w_{ij}} = \sum e_k(t) \frac{\partial y_k(t)}{\partial w_{ij}} \tag{2.11}$$

Bu denklemde tek bilinmeyen, yani daha önceden hesaplanmamış çarpan $\frac{\partial y_k(t)}{\partial w_{ij}}$ çarpanıdır. Bu nedenle bu çarpanı algoritmaya katmak için bir yöntem uygulanmalıdır.

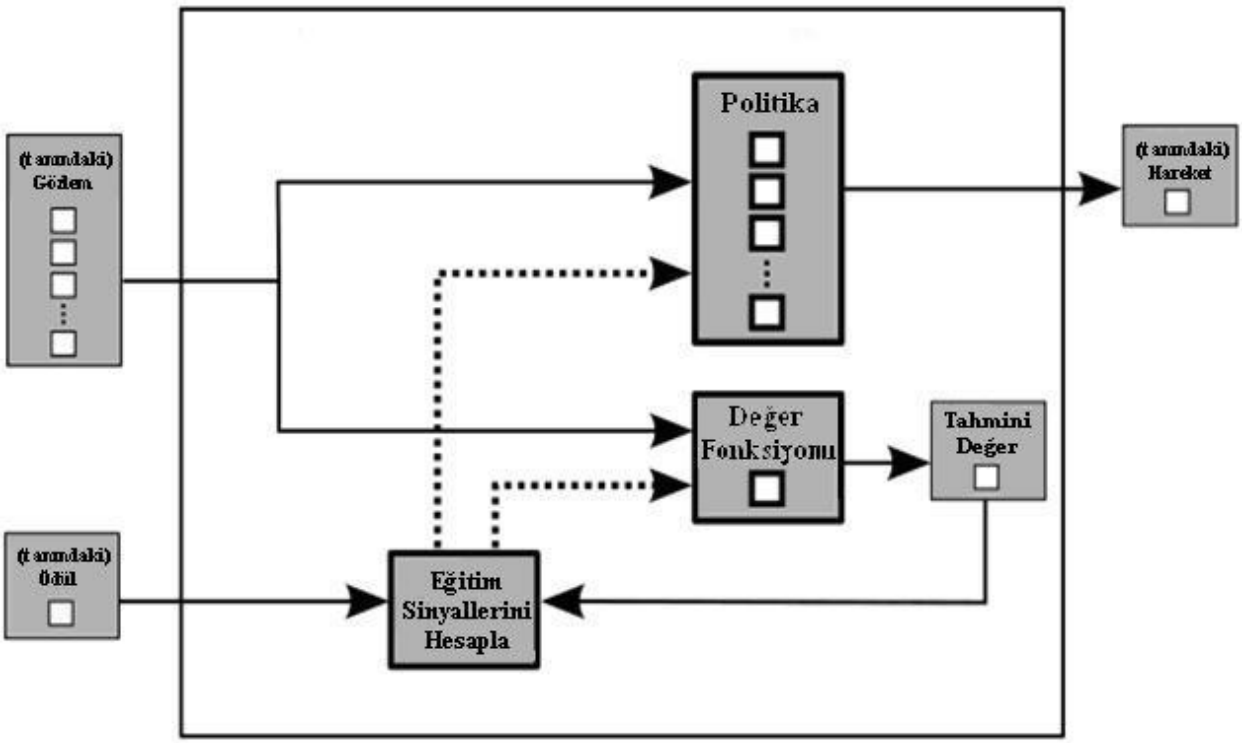
2.3 Destekli Öğrenme

Destekli öğrenme bir öğretici gerektiren fakat çıktının ne olması gerektiğinin ifade edilmediği bir yöntemdir. Bunun yerine, üretilen çıktının doğru veya yanlış olduğu söylenir ve sistem, doğru çıktılar üzerinden ilerler. Bu öğrenme yönteminde bir hedefe ulaşmak için en uygun hamlelerin nasıl seçileceği konusuna odaklanılır. Hedef bir ödül olarak belirlenir. Ödülün büyüklüğü, aynı zamanda hedefe ulaşma isteğinin ölçüsünü de gösterir (Uğurlu ve Biricik, 2006).

Destekli öğrenmede çıktının ne olduğu bilinmemektedir. Sadece doğru ve yanlış sonuç bilgisi mevcuttur. Öyleyse, hedef bir ödül olarak belirlendiğinde amaç, olası en büyük artımlı ödülü

sağlamak olmalıdır. Bunun için belli bir durumdan başlayıp amacı gerçekleştirebilecek hamlelerde bulunmak gerekir. Her hamle sonucunda içinde bulunulan durum değişir. O ana kadar elde edilmiş toplam ödül miktarı, hamle dizisinin başarısını gösterir (Uğurlu ve Biricik, 2006). Ödül, davranışı geliştirmek için nasıl düzeltmelerin gerçekleştirileceğini belirtmez; sadece bir performans ölçüm aracıdır. Bir hareket yapıldıktan sonra alınan destekler, o hareketin tekrarlanma olasılığını arttırırlar.

Çoğu destekli öğrenme etmeninin ana bileşeni, değer fonksiyonu ve hareket politikasıdır. Bir değer bileşeni durum, hareket çiftlerini tahmini değerlere eşler. Hareket politikası ise durumlara hareket atar.



Şekil 2.10 Destekli öğrenme etmeni (Streeter, 2005)

Bir destekli öğrenme etmeninin içsel mekanizması Şekil 2.10'da genel hatlarıyla gösterilmiştir. Buna göre, etmen etrafını gözler ve hareket politikası yardımıyla duruma uygun bir hareket belirler. Değer fonksiyonu da bu durum, hareket çifti için tahmini bir değer hesaplar. Bunun gerçekleştirilebilmesi için ödülden faydalanılması gerekir. Böylelikle, değer fonksiyonu, ödüllendirme prensibi ile çalışmış olur. Kısaca, etmen kendi durumunu, algılayıcıları sayesinde, duyumsar; hareket politikasını kullanarak yapacağı bir sonraki hareketi belirler ve ödül değerini kullanılan bir hesaplama yöntemi yardımı ile yapılan hareketleri destekler.

Destekli öğrenme beraberinde üzerinde düşünülmesi gereken başka konular getirir. Bunlardan biri geçici kredi devri (temporal credit assignment) problemidir. Bu problem, kısaca, önceki hareket sıralarından hangisinin ödüllendirileceği ile ilgilidir.

2.3.1 Geçici Kredi Devri

Etmen etrafıyla etkileşimde olduğu süreç boyunca genel olarak ödüllere sürekli erişemez. Bunun yerine, pozitif ödül alacağı hedef durumuna ulaşabilene dek, ödüllendirilmeyeceği yüzlerce konum alabilir. Ödülleri işleyebilecek, sonraki durum-hareket çiftlerine taşıyabilecek iyi düşünülmüş bir yöntem olmadığı takdirde etmen sadece ödüle erişmeden hemen bir önceki hareketi destekleyebilir.

Bu durum labirente konulmuş bir fare ile örneklendirilebilir. Fare, labirentin sonundaki peynire ulaşamadığı sürece bu ödülden faydalanamayacaktır. O zaman, ödül süreksizdir. İçinde bulunduğu durumu çözmek isteyen farenin, peyniri bulmadan bir önceki hareketi dışındaki hareketleri de desteklemesi gerekecektir. Uzun bir dizi hareket sonunda, bu hareketlerin hangisinin ödüle erişmede faydalı olduğu problemine geçici kredi devri problemi denir. Bu probleme göre, ödüle erişmede faydalı olan hareketler desteklenir (Streeter, 2005).

Geçici kredi devri problemine göre, bazı durumlarda ödüllendirilen bir etmen hangi durum-hareket çiftinin diğerlerinden daha değerli olduğunu bulabilmelidir. Bunun çözümü her durum-hareket çiftinin değerini tecrübeyle öğrenmek ile mümkündür. Uzak olmaktan ise ödüle yakın olmak daha değerli bir durum sayılır. Fakat bu bilgi etmenin çevresinde hazır bulunmaz; onun yerine öğrenilmesi gerekir. Dolayısıyla, destekli öğrenmede önemli adımlardan biri durum-hareket çiftlerine değer biçmek, yani bir değer fonksiyonu belirlemektir. Değer fonksiyonu, belli durumda yapılan bir hareketin bir değere eşleştirilmesine yarar. Değer fonksiyonu yardımı ile süreksiz olan asıl ödül, sürekli bir içsel ödüle dönüşür.

Durum-hareket çifti değeri denildiğinde anlatılmak istenen genelde gelecek ödüllerin beklenilen toplamıdır. Dolayısıyla, bir durum-hareket çiftinin değeri o durum-hareket çiftinden elde edilen değer ile sonradan alınması beklenen ödüllerin toplamıdır. Bunun gerçekleştirilebilmesi için, örneğin, bütün olası durum-hareket çiftleri ziyaret edilip her birinin değerleri hesaplanabilir. Bunun için, bir başlangıç durumu seçilmeli ve bulunulan durumdan itibaren her olası hareket üzerinden hareket edilmelidir. Bu tür bir strateji izlenildiği sürece sonraki durum-hareket çiftlerinin değerleri de sürekli olarak

güncellenecektir.

Bir başka yaklaşım ise etmenlere ödülleri süreksiz bir şekilde vermek yerine sürekli bir ödül fonksiyonu kullanmaktır. Örneğin, bu çalışmada da bir benzerinin olduğu gibi, etmen her adımda belli bir hedefe göre olan uzaklığa ters orantılı bir şekilde ödüllendirilebilir. Böylesine bir değer fonksiyonu güncellemesi seçildiğinde, öğrenme sürecinin başlarından itibaren bilgiye erişim sağlanır. Etmen, bir nevi duruma ait geribildirim alabildiğinden dolayı, bu ödüllendirme biçimi bazı problemlerde oldukça yararlı olabilir. Destekli öğrenmede bu tür bir yaklaşımı kullanmakla etmen, daha en baştan başlayarak her adımda bir ödül gradyanına sahip olur. Buna karşın, bu tür bir yaklaşımın zararı da gözlemlenebilir. Bunun nedeni ise, basit olmayan problemlerle karşılaşan bir etmenin istenmeyen yanlılık (bias) göstererek ancak yerel en iyi çözümlere ulaşabilmesinin kuvvetli bir olasılık olmasıdır. Bu nedenle, daha az ödül verip, etmenin hareket-durum çifti değerlerini kendi kendine öğrenmesini sağlamak daha güvenilir bir çözümdür (Streeter, 2005).

Geçici kredi devri problemini daha iyi anlayabilmek için temel eşitliği biraz açmak faydalı olacaktır. Bunu gerçekleştirebilmek için başlangıçta (2.12) eşitliğinde değinilen varsayımda bulunalım:

$$Q(d,h) = r_t + r_{t+1} + r_{t+2} + \ldots \tag{2.12}$$

Bu eşitlikte *d*, şuandaki durumu; *h* yapılmış hareketi; *Q(d,h)* ise içinde bulunulan durum-hareket çiftinin değerini ifade eder. Buna göre, o andaki durum-hareket çiftinin değerinin yine o anda elde edilen ödül ile o *t* anından sonra elde edilmiş ödüllerin toplamı olduğu varsayılır. Gelecekteki ödüller toplamının sonsuza gitme olasılığını engellemek için ise sonraki ödüller üstsel bir şekilde indirim (discount) faktörü ile azaltılır. Bu işlem, o anda ulaşılan ödülün değerini daha sonra ulaşılacak olan ödüllerin şimdiki değerinden daha değerli kılar.

$$Q(d,h) = r_t + g\, r_{t+1} + g^{\,2} r_{t+2} + \ldots \tag{2.13}$$

Bu eşitlik sadeleştirildiğinde (2.14)'e ulaşılır.

$$Q(d,h) = r_t + g\ Q(d',h') \tag{2.14}$$

Burada, *d'*, etmenin *d* durumundayken *h* hareketini yaptığında ulaştığı bir sonraki durumu simgeler. (2.14) eşitliği, (2.15)'teki gibi de yazılabilir:

$$0 = r_t + g\ Q(d',h') - Q(d,h) \tag{2.15}$$

Bu eşitlik, Q değer fonksiyonunun tamamen doğru olduğu varsayımına dayanmaktadır. Böylesine neredeyse kusursuz bir durumun her zaman, hatta çoğu zaman, söz konusu

olamayacağını düşündüğümüzde, eşitliği, hata payını da ekleyerek, (2.16) eşitliğindeki şekline dönüştürmekte fayda vardır:

$$d_t = r_t + g\ Q(d', h') - Q(d, h) \tag{2.16}$$

Burada, d_t hatayı göstermektedir. Ödül beklenenden yüksek olduğu takdirde hata pozitif değer alır; benzer şekilde, ödül beklenenden düşük olduğunda hata negatif değer alır. Çoğu zaman hata, ödülün aslından daha çok geribildirim vermektedir. Hata değerinin bu özelliğinden yararlanıldığı vakit daha iyi sonuçlar elde edilmesi beklenmektedir. Bu durum göz önüne alındığında hata değerini, değer fonksiyonunu güncellerken kullanma fikri cazip gelmektedir. Böylelikle gelecekte daha kesin sonuçlar alınması olasıdır. (2.17) eşitliği bu son durumu ifade eder:

$$Q(d, h) \leftarrow Q(d, h) + a d_t \tag{2.17}$$

a öğrenme oranını temsil eder. (2.17) eşitliği şimdiki durum-hareket çiftinin değerini tahmin hatasına göre günceller. Hata sıfır olduğunda değer tahmininin kusursuz olduğu düşünülür ve hiçbir değişiklik yapılmaz.

2.3.2 Q-Öğrenme

Q-Öğrenme geçici kredi devri problemini sondaki birkaç durum-hareket çifti değerini bellekte tutmasına gerek kalmayacak şekilde çözer. Bu öğrenme yöntemi, daha önce de üzerinde durulmuş olduğu gibi, belli durumlarda yapılan hareketlerin tahmini değerlerinin artımlı olarak güncellenmesiyle çalışır. Bu çalışma prensibi (2.18) eşitliğinde gösterilmiştir (Alpaydın, 2004).

$$Q(d, h) \leftarrow Q(d, h) + a\,(r_t + g\ \max Q(d', h') - Q(d, h)) \tag{2.18}$$

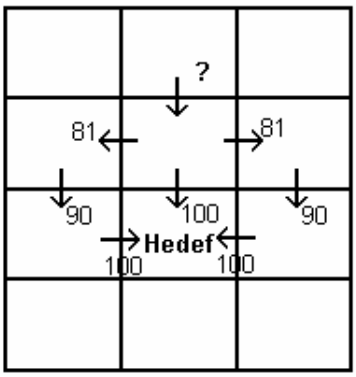
(2.18) eşitliğinden de anlaşılacağı gibi, değer fonksiyonu hesaplanırken bir sonraki durumda hangi hareket yapıldığında en yüksek değer elde edilebiliyor ise o durum-hareket çiftinden faydalanılır. Yararlanılmış olan bu durum-hareket çifti de kendinden bir sonraki olası durum-hareket çiftinden faydalanmış olacağı için sonraki hareket dizileri şuandaki değeri belirlemede etken rol oynamış olur. Tabii, sonraki durum-hareket çiftlerinin değerleri, daha önce de belirtilmiş olduğu gibi belli bir indirim oranı (γ) ile çarpılır. İndirim oranı kredi devri yapısını belirler. γ sıfıra eşit olduğu zaman sadece anlık ödül önem kazanır. Bu, daha sonraki durum-hareket çiftlerinden kazanılacak ödülün şuandaki ödülün hesaplanmasında etkili olmayacağı anlamına gelir. Eğer γ sıfıra eşitlendiğinde Q-öğrenme iyi çalışabiliyorsa, yani öğrenme gerçekleşebiliyorsa, genelde daha kolay ve hızlı çözülebilir bir problemle karşı karşıya

olunduğu düşünülebilir. γ'nın yüksek (bire yakın) olması demek gelecek durum-hareket çifti değerlerinin oranının şimdiki durum-hareket çiftinin hesaplanmasında önemli bir rol oynadığı anlamına gelir. Böylesine bir görüşten yola çıkılarak, γ yüksek olduğu takdirde etmenin uzak görüşlü olduğu söylenir. Q-öğrenme algoritması şekil 2.11'de verilmiştir. Yöntemde yapılacak olan hareketin belirlenmesinde, açgözlü hareket belirleme stratejisi kullanılmıştır.

Her $<d,h>$ değer çifti için $Q(d,h)$ değerini sıfırla
İçinde bulunulan d durumunu gözle
Belirli bir döngü sayısı kadar:
  Bir h hareketi belirle ve o hareketi yap.
  Durum için ödül belirlendiyse, al.
  Yeni d' durumunu gözle.
  $Q(d,h)$ değerini güncelle:
    $Q(d,h) \leftarrow Q(d,h) + \alpha(r_t + \gamma \max Q(d',h') - Q(d,h))$
  d' yeni durumunu, d durumu olarak ata.

Şekil 2.11 Q-Öğrenme algoritması (Alpaydın, 2004)

Q değerlerinin güncellenmesi için örnek bir durum, Şekil 2.12'de verilmiştir. Bu örnekte hedefe ulaşan hareketlerin ödülü 100 olarak belirlenmiştir. Diğer hareketlerin ödülleri yoktur. Sadece Q değerleri hesaplanacaktır. Gitmek istediğimiz koordinattan hedefe varan üç tane yolun önceden bulunduğu varsayılmıştır. Bu yolların Q değerleri {81, 100, 81}'dir. (2.19) eşitliğinde, aranan Q değerinin, bulunmuş olan Q değerlerinin en büyüğü olan 100 üzerinde hesap yapılarak bulunduğu gösterilmiştir.



Şekil 2.12 Q-Öğrenme yöntemi örneği

$$
\begin{aligned}
Q(s,a) &\leftarrow r + g\ \max(a')Q(s',a') \\
&\leftarrow 0 + 0.9 \max\{81,100,81\} \\
&\leftarrow 90
\end{aligned}
\tag{2.19}
$$

2.3.3 Keşif ve Sömürme

Bir etmen şuandaki en iyi çözümünden daha iyi sonuçlara erişebilmek amacı ile etrafını arar veya yeterince iyi sonuçlar elde etmiş olduğu varsayımına dayanarak sömürme aşamasına geçer ve bundan böyle yeni hareketler denemez. Belirli bir başlangıç noktasından başlayarak belirli bir hedefi bulmaya çalışan etmen, her adımda farklı yollar deneyerek, en uygun hareketi bulmaya çalışacaktır. Her denemesinde bildiği tek şey, daha önce izlediği hareket dizisi ile aldığı ödüldür. Ödülleri arttırarak en uygun hareket dizisini bulması için ise, her seferinde farklı yollar izlemesi gereklidir. Bu durumda robot bulunduğu ortamda en iyi hareket dizisini bulabilmek için, yeterli sayıda deneme yapmak zorundadır. Bu aşamaya keşif (exploration) adı verilir (Alpaydın, 2004). Bir kere ortamı öğrenip, her durum için en yüksek ödülü veren hareketleri keşfettiğinde, robot artık hep bu hareketleri yaparak topladığı ödülü en fazla yapacak, bir diğer deyişle, olabildiğince çok ilerleyebileceği hareket dizisini izleyecektir. Bu aşamaya sömürme (exploitation) denir (Alpaydın, 2004). Keşif ile sömürme aşamaları arasında bir seçim yapmak gerekir. Zamanlamayı doğru yapmak bu anlamda önemlidir çünkü öğrenmenin gerçekleşebilmesinde her iki aşama da eşit derecede önem taşır. Genel tutuma göre öğrenme işleminin başlarında etmen hangi hareketlerin hangi durumlarda başarılı olduğunu anlamak, durum-hareket çiftlerinin değerlerini belirlemek için keşif işleminden yararlanır. Sırf arayış içinde olan bir etmenin hiçbir zaman gelişmesi mümkün değildir. Öğrenilenden yarar sağlanmadığı sürece keşif tek başına bir şey ifade etmez. Keşifi sömürme takip eder. Genelde, sömürme işleminde en iyi durum-hareket çiftleri izlenir. Bu aşamada bilgi kullanılır ve bir anlamda problemin öğrenildiği gösterilmeye çalışılır.

Sömürme işlemini gerçekleştirmek için şimdiye kadar birkaç çözüm geliştirilmiştir. Bunlardan en bilineni açgözlü hareket belirleme (e-greedy) stratejisidir. Bu stratejiye göre hareket belirleme işi “Bir a hareketi belirle ve o hareketi yap” mantığına dayanır (Alpaydın, 2004). Açgözlü hareket belirleme stratejisi, herhangi bir durumda, belirli bir ε olasılığı ile rasgele bir hareketi seçer. (1- ε) olasılığıyla da (2.20) eşitliğinde verilen hareketi seçer.

$$
\begin{aligned}
a_t^* &= \arg\max_a Q_t(s,a) \\
a_t &= \begin{cases} 1-e \Rightarrow a_t^* \\ e \Rightarrow rasgele\_hareket \end{cases}
\end{aligned}
\tag{2.20}
$$

Sömürme işleminde kullanılan bir başka yöntem ise SOFTMAX hareket seçim yöntemidir. Buna göre, keşif aşaması sırasında tüm hareketleri eşit olasılıklarla seçmektense, belli bir durumdaki her hareketin seçilebilmesi için hareketlere farklı olasılıklar atanır. İyi hareketlerin olasılıkları daha kötü olanlara göre daha yüksektir. Olasılık atama işlemi sonrasında hareketlere atanan olasılıklara göre sömürme gerçekleştirilir (Streeter, 2005).

2.3.4 Sürekli Ortamlarda Destekli Öğrenme

Q-öğrenme gerçeklenirken Q değerleri genelde bir tabloda tutulur. Her durum-hareket çifti için bir hücre ayrılır. Bu tür bir yaklaşımın dezavantajı sürekli zamanlı problemlere pek uygun olamayabilişidir (Gaskett vd., 1999). Oysa bu tez de dahil olmak üzere birçok çalışmada, Q-öğrenme yöntemini motor komutlarını çağırmada kullanmak için bir yol bulunması gerekmektedir. Bu amaçla, yöntem sürekli hale getirilmeden de geniş durum-hareket alanlarını daraltmak problemi çözmede bir yöntemdir. Bir robotun her ekleminin 120 derecelik açı yapabildiğini ve her bacağında iki eklem olmak üzere dört bacağında toplam sekiz eklemi olduğunu varsayalım. O zaman, eklemlerin birer derecelik açı farkları yapabileceği düşünüldüğünde etmenin toplam 120^4 konum alabilecek olduğu hesaplanır. Bu konumların değerlerini teker teker hesaplamak neredeyse imkansızdır. Onun yerine her durumu özel olarak takip etmek yerine daha az miktarda parametre seçimine gitmek gerekir. Bunun kötü yanı her durumun hiçbir zaman kesin olarak hesaplanamayacağıdır. Durum-hareket çiftleri ancak yaklaşık değerler alabileceklerdir. Ama, yaklaşık değer atamak, aynı zamanda görünmeyen verileri bir genelleştirme içine katmayı mümkün kılar (Streeter, 2005). Gerçekte, etmenin sürekli bir dünyada zaten aynı durumu iki veya daha fazla kere yaşama olasılığı çok düşük olduğu için bu özellik oldukça avantajlıdır. O zaman belli durumlarda durum-hareket uzayını genelleştirip küçültmekte yarar vardır.

Hertfordshire Üniversitesi'nin Uyarlanır Sistemler Araştırma Grubunun (Adaptive Systems Research Group) cisimlendirilmiş sistemler üzerine yaptıkları çalışmalardan birinde durum-hareket uzayı küçültülmüştür (Jacob vd., 2005). Bu çalışma serbestçe gezen bir etmen değil, bir robot kolu üzerinde yapılmıştır. Buna göre her adımda robot kolunun eklemlerinden biri ya -5 ya da +5 derecelik bir hareket yapabilmektedir. Bu sayede robot kolunun yapabileceği hareket sürekli olmaktan çıkarılıp kısıtlanmıştır. Ayrıca benzeri bir hareket-durum uzayı küçültmesi de Japonya'da insansı robotlar (humanoid robots) üzerinde yapılmıştır (Schuitema vd., 2005). Bu çalışmada adından da anlaşılabileceği gibi iki bacaklı bir etmen yaratılmıştır. Amaç, etmene doğru adım atmayı öğretebilmektir. Bu etmen iki bacağı bulunduğundan ve

fazla eklemi olmadığından basit denilebilecek bir yapıya sahiptir. Bu sefer tork değerleri ayrık hale getirilmiştir.

2.3.5 Politika Gradyanlı Destekli Öğrenme

Destekli öğrenme yöntemleri motor kontrolü problemlerine bir çözüm getirse de genelde problem uzayı sorunundan da anlaşılacağı gibi yüksek boyutlu hareket sistemleri genelde ölçeklenemez. Dolayısıyla bu yöntemin boyutu az olan ayrık problem alanlarında kullanılması tercih edilir. Politika Gradyanlı Destekli Öğrenme (Policy Gradient Reinforcement Learning) yöntemi bu anlamda motor kontrolü probleminde umut vadeden sonuçlar vermiştir.

Destekli öğrenmenin kullanıldığı birçok uygulamada fonksiyon tahminine gidilmiştir. Bunlardan en çok kullanılanı daha önce de bahsedildiği gibi değer fonksiyonundan yararlanmaktır. Bu yöntemde tüm çabalar genelde açgözlü olmak üzere bir hareket politikası kullanarak değer fonksiyonunu tahmin etmeye yöneliktir. Bu yaklaşım birçok uygulamada iyi çalışıyor olmasına karşın her zaman iyi sonuçlar vermeyebilir. Bunun bir nedeni stokastik yerine belirlenimci (deterministic) hareket politikaları kullanmaya elverişli olmasıdır; oysa çoğu zaman değişik hareketlerin belirli olasılıklar ile sağlanması daha iyi sonuçlar vermeye belirlenimci bir hareket politikası izlenmesinden daha uygundur. Bir diğer neden ise bir durum hareket çiftinin tahmini değerindeki ufak bir değişim bulunulan durumda o hareketin seçilip seçilmemesini etkileyebilir (Sutton vd., 2000). O zaman her ufak değişimin uygulamanın gidişatını arzu edilenden çok değiştirebilecek etkiye sahip olması olasıdır.

Politika gradyanlı destekli öğrenme klasik destekli öğrenme yöntemlerine bir alternatif oluşturmuştur. Bu destekli öğrenme yönteminde ana amaç belirlenimci bir hareket politikası gerçekleştirmek için bir değer fonksiyonu kullanmaktansa kendi parametreleri olan bağımsız bir fonksiyon yaklaşıklaştırıcısı kullanarak stokastik bir hareket politikasından yararlanmaktır. Bu sayede yerel en iyi politikaya yakınsama mümkün olmaktadır (Sutton vd., 2000).

Politika gradyanlı destekli öğrenme etmenin karar verme mekanizmasını oluşturacak olan bir başlangıç parametre vektörü $p = \{q_1,...,q_N\}$ ile başlar ve bu vektörün her parametresine göre vektörün hedef fonksiyonunun kısmi türevini tahmin eder. Kısmı türev tahminin gerçekleştirilebilmesi için öncelikle rasgele üretilmiş hareket politikaları $\{R_1,...,R_t\}$ değerlendirilir. Bu değerlendirme (2.21) eşitliğinde ifade edilmiştir (Kohl ve Stone, 2004).

$$R_i = \{q_1+\Delta_1,...,q_N+\Delta_N\} \tag{2.21}$$

Bu eşitlikte her Δ_j yapılacak değişiklik miktarıdır ve q_j 'den küçük olan sabit bir değerler arasından rasgele bir şekilde seçilir. Her politika değerlendirmesi sonucunda yapılacak hareket belirlenir. Burada düşünülen parametrelerin elemanlarına göre türev alıp politikanın gradyanını parametre uzayında almaktır. Fakat politikanın gerçek denklemi bilinmemektedir. Bu nedenle türev deneyerek elde edilen sonuçlardan örnekleme ile alınır. Bu yöntem yapay yaşam formuna uygulandığı şekli ile tezin üçüncü bölümünde destekli öğrenme başlığı altında daha ayrıntılı bir şekilde anlatılmıştır.

3. YAŞAM FORMU VE YÖNTEMLERİN UYGULANMASI

3.1 Simülasyon

Simülasyon ortamının doğru bir şekilde modellenmesi üç boyutlu yapay yaşam formlarının gerek yapılandırılması aşamasında gerekse hareket ettirilmesinde önemli bir rol oynar (Taylor ve Massey, 2001). Kinematik bir yaklaşımdansa dinamik bir yaklaşım yapay yaşam formlarının çevrelerine adapte olmalarında daha yararlı olmaya elverişlidir. Bunun nedeni gerçekçiliği sağlayabilmek adına yapay yaşam formunda ve yapay yaşam alanında güç, tork ve sürtünme gibi özelliklerin etkili olması gerekliliğidir. Bu fikirden yola çıkılarak yaşam formuna fiziksel açıdan gerçekçi hareketler yaptırabilmek için Klein (2002) tarafından tasarlanan Breve fizik motoru simülasyon ortamı olarak seçilmiştir.

Breve yapay yaşamın ve çok etmenli sistemlerin simülasyonlarını 3 boyutlu bir dünyada gerçekçi bir şekilde gerçekleştirme amacı ile geliştirilmiş açık kaynaklı bir simülasyon yazılımı paketidir. Bu yazılım kullanıcıların etmenlerden istedikleri davranışları tanımlamasını ve daha sonra etkileşimini gözlemleyebilmesini mümkün kılar. Yaratılan etmenlerin simüle edilebilesi için Breve, fiziksel simülasyon ve çarpışma tespitinin yapılabilmesini sağlar. Breve, benzetimi yapılmış dünyayı görselleştirebilmek amacı ile bir OPENGL görüntüleme motoruna sahiptir ve eklemli gövdelerin simülasyonunu gerçekleştirebilmek için yine OPENGL tabanlı olan ODE (Open Dynamics Engine) fizik motorundan destek alır. Breve fizik motorunun ana amacı gerçekçi 3 boyutlu dünyalar simüle etmek de olsa bir görselleştirme etmen olarak da başarısı kanıtlanmıştır (Klein, 2002). Simülasyonlar etmenlerin davranış ve etkileşimlerini nesneye yönelik bir dil ile tanımlayarak yazılır.

Breve, gerçek zamanlı bir şekilde geri cevap verebilen etkileşimli fiziksel simülasyonlar geliştirmeyi temin eder. Gerçekçi 3 boyutlu dünyaların da simülasyonlarının yapılabilmesi ile bahsedilen özelliğin birlikte bulunuşu, Breve'in karmaşık uyarlanır sistemlerin ve yapay yaşam formlarının yaratılmasına uygun bir ortam olduğunu gösterir. Bu nedenle tezde Breve simülasyon ortamı kullanılmıştır.

Fiziksel anlamda gerçekçi bir ortam sağlandığı andan itibaren yapay yaşam formu ile çevresi arasında birçok farklı dinamik etkileşim oluşturmak ve gözlemek olanaklıdır. Bu sayede çevresi ile dinamik etkileşim haline giren yaşam formu algılayıcıları ile bulunduğu durumu algılayabilecek ve eyleyicilerini kontrol ederek farklı hareketler gerçekleştirebilecektir.

Yaratılan dünyada hareket eden eklemli etmenler eklemler ile birbirlerine bağlanmış sert gövdelerden oluşmuşlardır. Fizik motoru direk olarak etmenlerin gövdeleri üzerinde çalışmaktadır, bu nedenle etmenlerin kendileri de çevrelerini etkileyen fizik kurallarına tabidirler. Bunun dışında her gövde bağlantıda olduğu eklemler tarafından da kısıtlanabilmektedir. Bir eklem iki veya daha fazla gövdeyi birbirine bağlayabilir. Bu eklemler hareketli veya sabit olabilir. Simülasyonun arka planında da çarpışmaları denetleyebilmek amacıyla geçici eklemlerden yararlanılmaktadır.

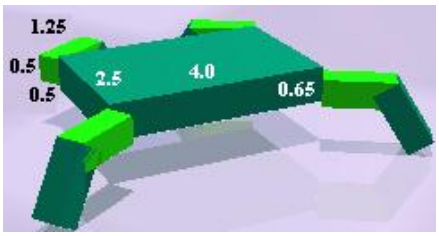
Yaşam formu sonu olmayan düz bir zeminde hareket ettirilmektedir. Gerçekçi bir yapay çevre yaratabilmek için gerçek dünyadaki deniz seviyesindeki yerçekimine yakın olması açısından yer çekimi 9.81 m/s^2 olarak ayarlanmıştır. Sürtünme de zeminle bir çarpışma olduğunda yaratılan geçici temas eklemleri sayesinde modellenmektedir. Hesaplama karmaşıklığını çok arttırmamak için dünyada olan diğer dirençler, değerleri de genelde küçük olduğundan, göz ardı edilmiştir.

3.2 Yapay Yaşam Formlarının Morfolojileri

Bu varlıklar eklemler ile birbirlerine bağlanmış gövdelerden oluşurlar. Bu tezde gövdeler Sims'in çalışmasından örnek alınarak farklı boyutlardaki kutucuklar olarak seçilmişlerdir. Bahsedilen yaşam formlarının algılayıcıları ve eyleyicileri vardır. Algılayıcılar çevreden yaşam formuna veri akışını mümkün kılar. Eyleyiciler ise motor gibi hareket ederek yaşam formlarının çevreyle fiziksel bir etkileşim içinde olmasını sağlar. Her yaşam formunun birbirine eklemler ile bağlı üç boyutlu, kutu görünümlü, parçalardan oluşan bir yapısı vardır. Eklemler birbirlerine kas görevi gören eyleyiciler ile bağlanmışlardır. Sözü geçen eyleyiciler eklemlerin serbestlik derecelerine tork uygularlar, böylelikle hareket oluşumunu veya hareketin devamını sağlarlar. Bu yapı sayesinde eklemler arasında çekme ve itme güçleri mümkün olan tüm serbestlik derecelerinde gerçekleştirilebilinir. Böylelikle kasların esneme ve uzanma hareketleri modellenmiş olur.

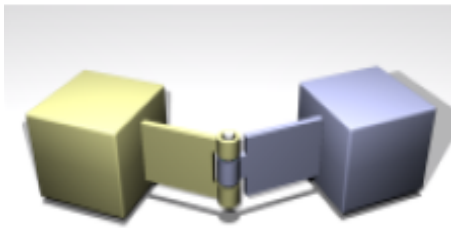
Simülasyon ortamındaki nesneler arttıkça hesaplama karmaşıklığı da artmaktadır. Bu nedenle bu tez çalışmasındaki yöntemler, gerçekçilikten çok ödün vermeyecek şekilde, nispeten basitçe bir yaşam formu üzerinde gerçekleştirilmişlerdir. Tez için yaratılan yaşam formunun morfolojisi Şekil 3.1'de gösterilmiştir. Bu yaşam formu dokuz parçadan oluşmaktadır. Bunlardan biri diğerlerinden büyük olduğundan gövde görevi görmesi beklenmektedir. Diğer parçaların ise bacak rolü oynaması gerçeğe daha uygun olacaktır. Bunun nedeni doğada rastladığımız hayvanlara baktığımızda bu tür bir manzarayla karşılaşmamızdır. Bu nedenle tez

boyunca büyük olan parça ana gövde, ana gövdeye eklemlerle bağlı olan dört parça üst bacaklar, diğer dört parça da alt bacaklar olarak adlandırmıştır. Ana gövdenin boyutu 4,0 x 2,5 x 0,65, bacakların boyutları ise 0,5 x 1,25 x 0,5'tir.



Şekil 3.1 Yaşam formunun morfolojisi

Yaşam formunun sekiz eklemi bulunmaktadır. Bu eklemler menteşeli eklemlerdir (hinge). Bu eklemler kendi eksenleri üzerinde dönerler. Bu eklemin iki algılayıcısı ve bir de eyleyicisi bulunur. Algılayıcılar eklemin o esnadaki dönme açısını ve açısal hızını algılar. Şekil 3.2'de menteşeli bir eklem gösterilmiştir. Menteşeli eklemler tek serbestlik derecesine sahiptirler, dolayısıyla yaratılan yaşam formunun toplam serbestlik derecesi sekizdir. Her bir bacak da iki serbestlik derecesine sahiptir. Bu bacak formu, bacağın yukarı kaldırılıp öne savrulabilmesine imkan tanır ve etmene yeterli miktarda hareketlilik sağlar.

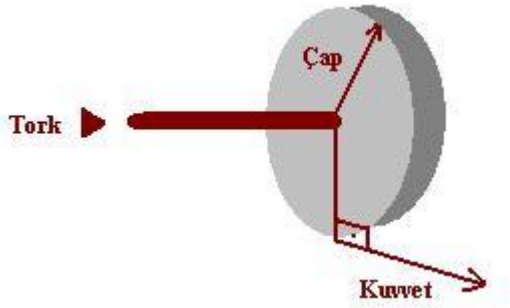


Şekil 3.2 Menteşeli eklem (Russell, 2001)

Her bir eklem, o eklemin etrafındaki parçaların dönmesini sağlayan bir tork üreten bir motor ile gerçeklenirler. Eklemler Newton'un ikinci rotasyon kuralı kullanılarak modellenmiştir. Buna göre eklemlerin işleyişini modellemek için tork, kuvvet, açı ve çap bilgilerinin kullanıldığı basit bir denklem (3.1) ile hesaplanmıştır.

$$t = F \cdot r \cdot \sin(q) \tag{3.1}$$

(3.1) eşitliğinde τ tork, r çapı, F kuvveti, q de r ile F arasındaki çizginin yaptığı açıyı temsil eder (Şekil 3.3). Bu denklem eyleyicinin eklem üzerinde ne kadar tork uygulayacağını hesaplamak için kullanılır.



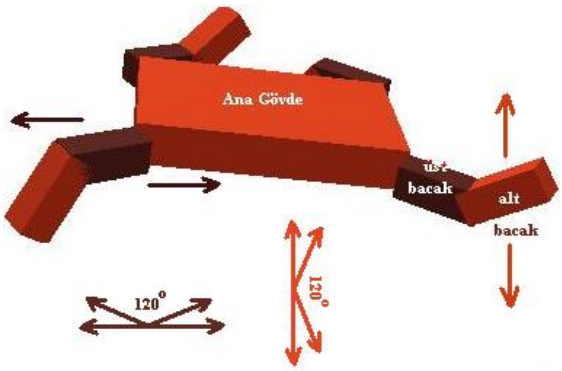
Şekil 3.3 Menteşeli eklemlerin çalışma mekanizması

Simülasyonlar genelde her eklemin zamana bağlı bir sinüs dalgası hareketi yaptığı kabulüne dayanarak gerçekleştirilmiştir. Buna göre ekleme uygulanan tork (3.2) eşitliğinde verilmiştir.

$$t = F \cdot r \cdot \sin(X \cdot (t+Y) - Z) \tag{3.2}$$

Bu denklemde t zamanı temsil eder; X zaman değerini ölçekleyerek dalganın frekansını kontrol eder; Y zaman değerini dengeleyerek dalganın faz kaymasını kontrol eder; Z ise sonuca sayıl bir değer ekleyerek dalganın dikey kaymasını kontrol eder. Kuvvet (F) ve çap (r) çarpımı da dalganın genişliğini belirler.

Her eklem -60 ile 60 aralığında herhangi bir açı değeri alabilir. Ana gövde ile üst bacakları bağlayan eklemlerin normali (0,0,1), üst bacaklar ve alt bacakları bağlayan eklemleri normali ise (1,0,0) olarak belirlenmiştir. Gövdeler ve eklemler arasındaki ilişki Şekil 3.4'te özetlenmiştir.



Şekil 3.4 Yaşam formunun hareket mekanizması

Yaşam formunun birbirine çapraz eklemleri simetrik olarak çalışmaktadır. Böylelikle sekiz eklem yerine dört eklemin kontrolü yeterli olacaktır. Her eklemin biri o anda bulunduğu açı, diğeri açısal hızı olmak üzere iki algılayıcısı olduğuna göre yaşam formunun toplam sekiz algılayıcısı vardır. Algılayıcılar radyan cinsinden sürekli değerler üretirler. Toplam eyleyici sayısı ise dörttür. Eyleyiciler her bir ekleme uygulanan torklardır ve eklemlerin motorlarını

temsil etmektedir. Eyleyiciler tarafından üretilebilecek tork gücü 300 gm-cm ile sınırlıdır, eklemlere belirlenen maksimum tork miktarından fazla tork uyguladıklarında efektif sonuç alınamamaktadır.

3.3 Öğrenme

Öğrenmenin, eklemli bir sistemin fiziksel yapısına denk gelecek şekilde bölümlenmesi yapılarak gerçekleştirilmesi mümkündür. Gerçek dünyadaki canlılar gözlemlendiğinde modüler yapılara sıkça rastlanmaktadır. Örneğin, birçok hayvanda birden fazla bacak mevcuttur. Bu bacaklardan her biri mantıksal olarak parçalara ayrılmışlardır. Bu parçaların en belirginlerinin eklemler olduğu söylenebilir. Bu tezde üzerinde çalışılan etmen de, benzer şekilde, algılayıcıları ve eyleyicileri sayesinde algılayan, hareket eden ve yerel hedefler koyan parçalardan oluşmuştur.

Eklemler sayısız konum alabilirler. Bu nedenden tek bir eklemin bile kontrolü çokça karmaşıktır. Bu durum ters kinematik kullanılarak analitik kontrolün sağlanmasını zorlaştırır. Bundan dolayı hedefi gerçekleştirebilmek için hangi hareketlerin yapılması gerektiğini eklemlerin kendilerinin öğrenmesi daha mantıklıdır.

3.3.1 Genetik Algoritma

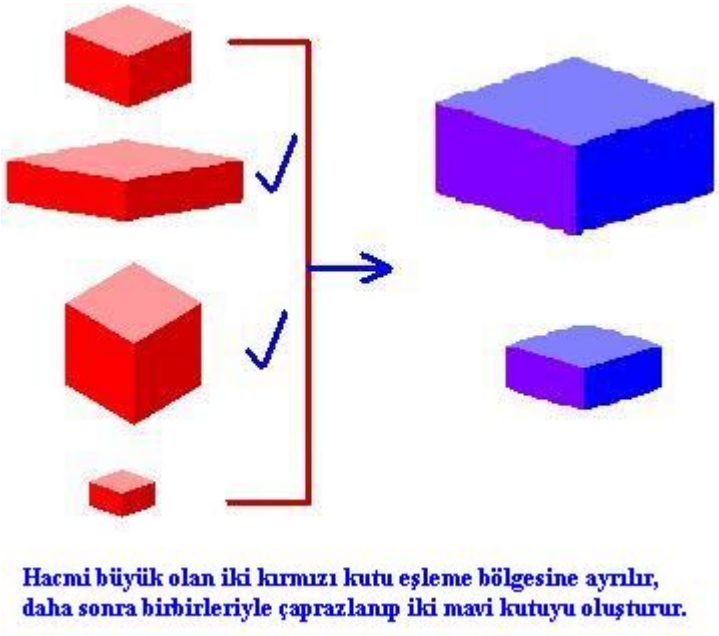
Yaşam formunun hareketini evrimleştirebilmek amacıyla gerçekleştirilen ilk adım nüfus sayısının kararlaştırılması ve o sayı kadar yeni bireyin rasgele bir şekilde oluşturulmasıdır. Küçük nüfuslarla çalışan Genetik Algoritmalar genelde bir çözüme daha çabuk yakınsadığından nüfus boyutu 20 olarak seçilmiştir. Nüfusun bireylerinin her birinin kromozomundaki gen sayısı 9'dur. Bu genlerden biri torku hesaplamada kullanılan X değeri olarak kullanılır; eklemden ekleme farklılık göstermez. Dördü Y, diğer dördü de Z değeri olarak kullanılırlar. Her bireydeki her bir eklemin kendine özgü Y ve Z değerleri vardır. Bu bireyler üzerine belirli olasılıklarla üç işlemden biri uygulanabilir; bunlar üreme, çoğalma ve değişime uğramadır. Üreme bireylerin çaprazlanması ile gerçekleşir; çoğalma işleminde bireyler kopyalanır; değişime uğramada ise bireyler mutasyon geçirir. Bireyler 0,5 olasılık ile ürerler, 0,2 olasılık ile çoğalırlar, 0,3 olasılık ile de değişime uğrarlar.

Üreme ve değişime uğramaya ek olarak bireylerin çoğalabilmesi en iyi bireylerin tamamen yok olması olasılığını azaltmak için uygulanan bir işlemdir. Bunun nedeni çaprazlama yaparken de güzel özellikleri kaybedebilme ihtimalidir. Fakat iyi sonuçları elde tutabilmek

amacı ile çoğunlukla çoğalma işlemi gerçekleştirilirse bu sefer de yeni bireylerin yaratılması zorlaşır ve nüfus fazla değişiklik göstermez. Bu durum evrimleşmeyi engelleyici bir tablo ortaya çıkartır. Bu nedenle çeşitliliği sağlayabilmek için çaprazlama ve mutasyon işleçlerinden faydalanılması şarttır.

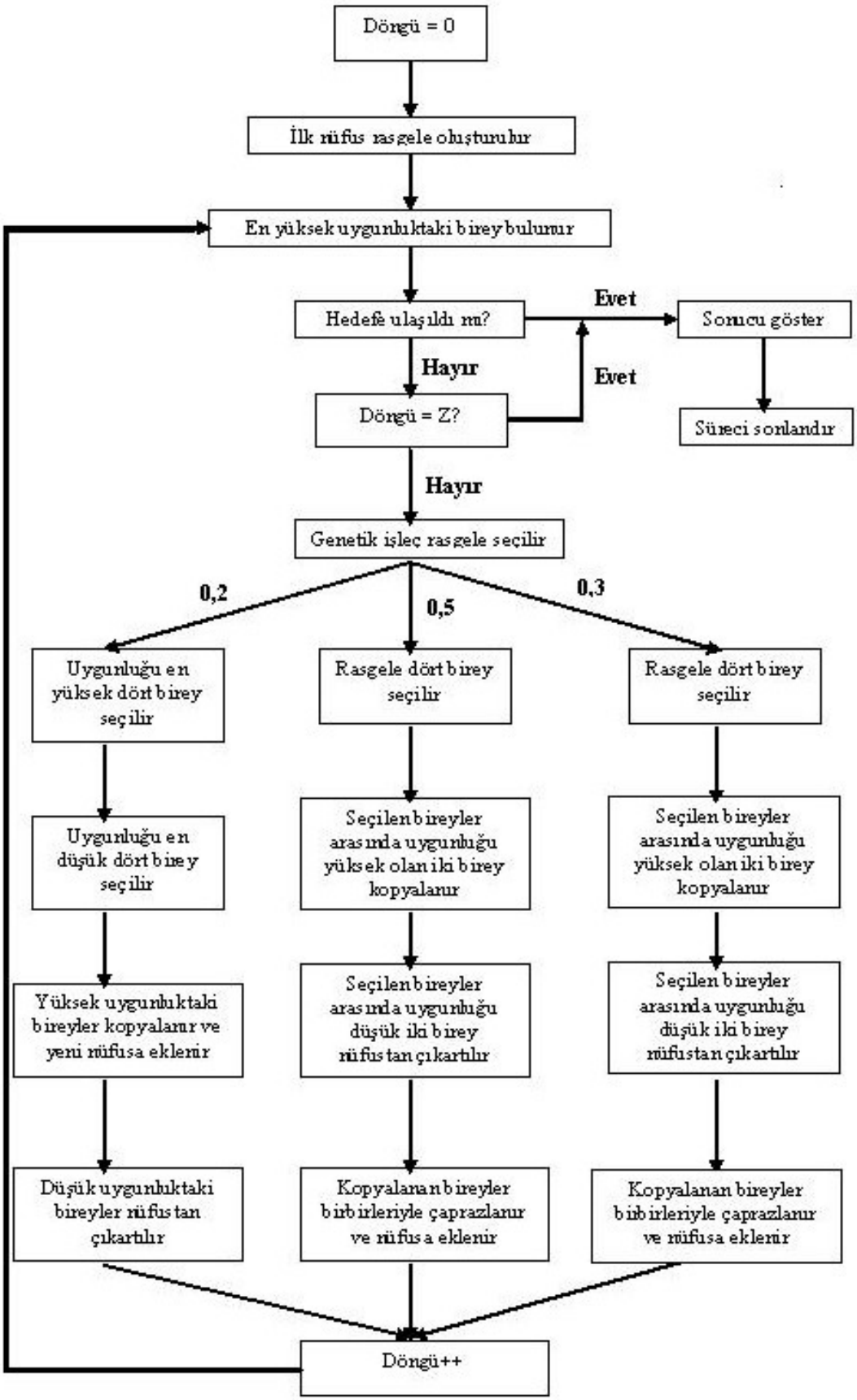
Mutasyon olasılığının oldukça büyük seçilmesi bireylerin küçük bir nüfus içinden seçilmesinden kaynaklanmaktadır. Mutasyon oranı küçüldüğünde çeşitlilikte yine bir azalma olmaktadır. Bunun nedeni bireylerde yeterli miktarda farklılık yaratılmayınca Genetik Algoritma'nın fazlasıyla çabuk yakınsamasıdır. Bu oran çok büyük seçildiği takdirde ise yaşam formu devamlı değişik hareketler yapmakta, hiç yakınsamamaktadır. Bu durum öğrenmenin gerçekleşemiyor olduğuna delalettir. Dolayısıyla belli bir noktadan sonra "evrimleşme gerçekleşti" diyebilmek için tabii ki yakınsamaya ihtiyaç vardır. Üreme olasılığının yüksek tutulmasının nedeni çaprazlama işlecinin hem yakınsamada hem de çeşitlilik yaratmada oldukça başarılı olmasından kaynaklanır.

Yaşam formunun hareketini evrimleştirebilmek amacıyla gerçekleştirilen bir sonraki adım seçilen işleme göre ya turnuva seçimini ya seçkinci stratejiyi ya da mutasyonu gerçekleştirmektir. Üreme işlemini gerçekleştirmek için turnuva seçiminden yararlanılmıştır (Şekil 3.5). Buna göre, öncelikle nüfustan rasgele dört birey seçilir. Daha sonra bireyler uygunluklarına göre sıralanır ve sıranın en önündeki iki birey (turnuvada başarılı olanlar) eşleştirerek nüfusa yeni iki birey dahil edilir. Sıranın sonundaki bireyler ise nüfustan çıkarılır. Mutasyon da benzeri bir şekilde yapılmaktadır. Bu sefer yine turnuva seçimi yapılıyormuşçasına dört birey rasgele bir şekilde seçilir ve bu dört bireyden diğerlerinden daha iyi sonuçlar veren ikisi kopyalanır, kopyalanan bireyler mutasyona uğratılır ve daha kötü olan diğer iki birey nüfustan çıkarılır. Böylelikle iki iyi bireyin mutasyona uğrayan kromozomları diğer iki bireyin yerine geçmiş olur. Çoğalma işlemini gerçekleştirmek için de seçkinci stratejiden faydalanılmıştır. Buna göre nüfusun en iyi dört bireyi kopyalanır ve en kötü dört bireyinin yerini alır.



Şekil 3.5 Her turnuvada dört bireyin hacimlerine göre yarıştığı bir turnuvada birbirleriyle yarıştırılan kutulardan oluşturulan alt küme

Bireylerin arasında seçimlerin yapılabilmesi için bu seçimin neye göre yapılması gerektiğini belirlemeye yani bir ölçüt tanımlamaya ihtiyaç vardır. Bu ölçüt yaşam formunun belli bir zaman süreci boyunca aldığı mesafedir. Üreme ve çoğalma işlemlerinde hangi birey daha ileriye gidebilirse o avantajlıdır, kötü olanlar ise o adımda yok olmaya mahkumdur. Genetik algoritma yakınsamış gibi bir izlenim veriyorsa veya yaşam formu belli bir süre boyunca istenilen yol veya daha uzun bir mesafe kadar ilerleyebilmişse “öğrenme” tamamlanmış sayılır. Yapay yaşam formunun hareketlerini evrimleştirmek amacıyla gerçekleştirilen bu algoritma genel hatlarıyla Şekil 3.6’da verilmiştir.



Şekil 3.6 Genetik Algoritma

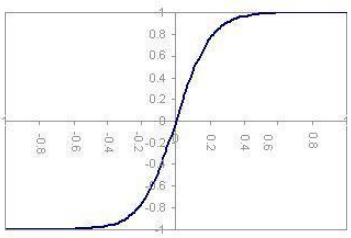
3.3.2 Yapay Sinir Ağları

Dört bacaklı yaşam formu ilk hareketini rasgele yapmaktadır. Müteakip hareketleri gerçekleştirmesi için Yapay Sinir Ağlarının çıktıları, bacaklardaki eklemlere uygulanır. Algılayıcılar bacaklardaki eklemlerin açı ve açısal hızlarını algılarlar ve Yapay Sinir Ağı'nı bu bilgi ile beslerler. Buna göre YSA'nın 4x2 = 8 girdisi vardır. Bacakların hareketi için kullanılan yapay sinir ağlarının çıktıları, bacaklardaki eklemleri hareket ettirecek tork değerine dönüştürülecek değerlerdir. Yaşam formunu hareket ettirebilmek için dokuz çıktı gerekmektedir. Bu çıktılardan biri torku hesaplamada kullanılan X değeri olarak kullanılır; eklemden ekleme farklılık göstermez. Dördü Y, diğer dördü de Z değeri olarak kullanılırlar. Her bireydeki her bir eklemin kendine özgü Y ve Z değerleri vardır. Eyleyiciler bu çıktıları kullanarak torku hesaplayıp bacaklardaki eklemlere uygulandığında, bacakların hedefe ulaşmak için öğrenilen şekilde hareketi sağlanmış olur.

Etkinleştirme fonksiyonu negatif değerler de alabilmek durumundadır çünkü ters tarafa da açı yapılabilmelidir. Eğer hep aynı yönde tork uygulanırsa sonunda hareket eklemin sınırlarında tıkanıp kalacaktır. Bu nedenle etkinleştirme fonksiyonu olarak hiperbolik tanjant kullanılmıştır (Şekil 3.7). Hiperbolik tanjant fonksiyonu (3.3) eşitliğinde verilmiştir.

$$f(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x}) \tag{3.3}$$

Etkinleştirme fonksiyonu -1'den küçük +1'den de büyük değer almadığından çıktıları bu aralığa yansıtmak gerekir. Aynı şekilde girdilerin de çıktılara uygun olacak biçimde küçültülmesi gereklidir yoksa öğrenmek için yapılan ağırlık değişiklikleri girdileri fazla etkilemez. Bu nedenle tüm girdi ve çıktılar [-1,1] aralığında olmak üzere ayarlanmıştır.

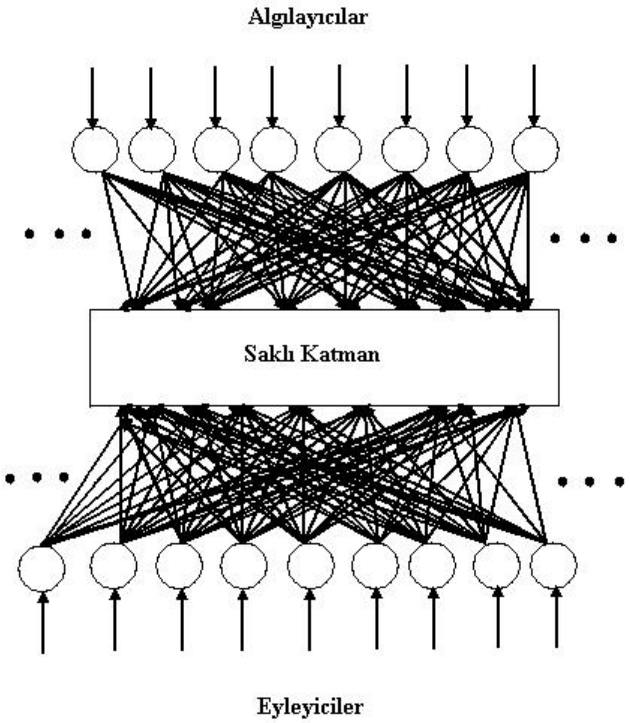


Şekil 3.7 Hiperbolik tanjant fonksiyonu

Yaşam formu, bir hareket yaparken bulunduğu konumu bilmektedir. Bir hareketi yaptıktan sonra geldiği konumu da bilir. Bu iki konum arasındaki fark hesaplandığında, o hareketin ileriye gitme (hedefe ulaşma) konusunda ne kadar başarılı olduğu bulunabilir. Bir başka deyişle, bir hareket yapıldığında elde edilen yer değiştirme miktarı, bacakların hareketini sağlayan sinir ağlarını eğitmek için hata fonksiyonu olarak kullanılabilir. Aynı hata

fonksiyonundan bacakların hareketini seçen ağ yapısının eğitiminde de yararlanılabilir.

Yaşam formunun yol alabilmeyi öğrenmesi için geliştirilen İleri Beslemeli Geri Yayılımlı Yapay Sinir Ağı algılayıcılardan toplanan verileri girdi olarak alır. Ağın çıktıları ise eyleyicilere girdi olarak verilir. Bu öğrenme yöntemi Şekil 3.8'de ana hatlarıyla gösterilmiştir.



Şekil 3.8 Geri yayılımlı hareket öğrenme

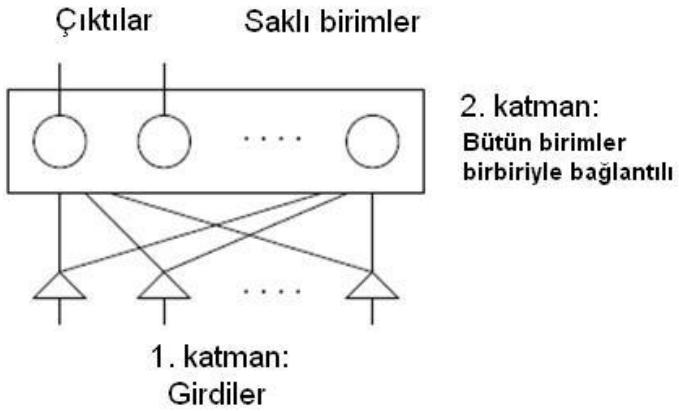
Yaşam formunun, bir adımda, belirli bir mesafeyi kat etmesi hedeflenmektedir. Bu hedefi sağlayana dek Yapay Sinir Ağlarının öğrenme süreçleri devam ettirilmiştir. Bir adımda hedeflenen mesafeyi kat etmeyi başaran yaşam formunun, Yapay Sinir Ağlarının eğitimi durdurulur. Bundan sonra sadece ileri besleme ile hareket ettirici ağlar çalıştırılıp bacaklardaki eklemler hareket ettirilir. İstenilen mesafe kat edilmişse yaşam formunun öğrenmesinin ilk aşaması tamamlanmış demektir. Bu aşamadan sonra daha da iyi sonuçlar elde edebilmek adına öğrenilenler birbirleriyle sırayla yarıştırılacaktır. Burada önemli olan biraz daha uzun mesafelerde etmenin daha iyi gidebilmesini sağlayabilmektir. Yine belli bir mesafe hedefi gerçekleştirildiğinde öğrenme tamamlanır.

İleri Beslemeli Sinir Ağları sadece girdi yöneylerini statik bir biçimde eşler. Beyinlerimizin ise durumsuz bir girdi-çıktı sistemi olmadığı, bunun yerine çok boyutlu ve doğrusal olmayan bir istem olduğu açıktır. Bu nedenle beyinin dinamik fonksiyonlarını modelleyebilmek veya beyinsel özelliklere sahip bir makine tasarlayabilmek için, içsel durumları saklayabilecek ve karmaşık dinamikleri uygulamaya geçirebilecek bir sistem kullanmak önemlidir. Yinelenen

Yapay Sinir Ağları için tasarlanan öğrenme algoritmalarına olan ilgi bundan kaynaklanmaktadır. Bu öğrenme algoritmalarında kısa dönem belleğin yaratılmasını sağlayacak geribildirim bağlantıları oluşturulur ve zaman gecikmelerinden faydalanılır. Kısaca, Yinelenen Yapay Sinir Ağlarının İleri Beslemeli Sinir Ağlarından farkı için sadece girdi alanında değil aynı zamanda iç alanda çalışmaları denilebilir. İç alandan kasıt sinir ağı tarafından daha önce işlenmiş olan bilgilerdir. Bu nedenle tezde, Yapay Sinir Ağlarının kullanımından bir sonraki aşama Yinelenen Yapay Sinir Ağlarını eğitmek olarak belirlenmiştir.

Tezdeki YYSA'ndaki odak noktası denetimli öğrenmedir. Denetimli öğrenmenin yinelenen ağlardaki ana mantığı İleri Beslemeli Sinir Ağlarındakine benzerlik göstermektedir, yani sinir ağına girdi için beklenen çıktı verilmektedir. Yine bir hata fonksiyonu tanımlanır ve gradyanı ağırlıklara göre türetilir. Fakat asıl fark girdi ve çıktıların statik olmamalarında yatar; onlar artık zaman dizileri şeklinde ele alınırlar.

Bu tezde Yinelenen Yapay Sinir Ağı Gerçek Zamanlı Yinelenen Öğrenme ile eğitilmiştir. Bu ağ, daha önce de bahsedildiği gibi, iki katmanlı bir yapıdadır. Sinir ağının birinci katmanı girdilerin değerlerini bir anlamda sıkıştırır ve ikinci katmandaki tüm düğümlere iletir. İkinci katmanın yapısı Hopfield sinir ağına büyük ölçüde benzerlik göstermektedir. Bu katmanda tüm düğümler birbiriyle bağlantılıdır ve her birinin bir içsel durumu vardır. İkinci katmandaki bazı düğümler çıktı düğümü olarak görev görürken diğerleri saklı düğümlerdir. Kullanılan Yinelenen Yapay Sinir Ağı mimarisi Şekil 3.9'da gösterilmiştir. Yine girdi sayısı 8, çıktı sayısı 9'dur. Gizli birim sayısı ise 10'dur.



Şekil 3.9 Kullanılan Yinelenen Yapay Sinir Ağı mimarisi

3.3.3 Destekli Öğrenme

Destekli öğrenme, genelde önceden hiçbir kabul yapılmadan gerçekleştirilen bir öğrenme

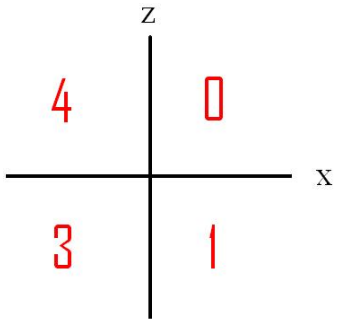
yöntemidir. Fakat eklemli etmenlere, fiziksel dünyanın yapısından kaynaklanan, birtakım kısıtlamalar getirilmek durumundadır. Bu da destekli öğrenmeye de kimi kısıtlamaların getirilmesi gerekliliğini yaratır. Böylelikle, bir anlamda, etmen öğrenme işlemine başlamadan evvel bazı önkoşulların konulduğu düşünülebilir. Bu bağlamda kısıtlar yararlı olmaktadır: öğrenmenin her adımında göz önünde bulundurulması gereken olasılıkların miktarı azalmaktadır. Tezde Q-öğrenme ve Politika Gradyanlı Destekli Öğrenme olmak üzere iki destekli öğrenme yöntemi uygulanmıştır.

Daha önce bahsedilen modüler yapı düşünüldüğünde yerel benzerlik gösteren durumların yerel benzerlik gösteren hareketler gerektireceği kabul edilebilir. Bu yerel algı ve hareket kabulü hedef temelli bir destekli öğrenme yöntemi ile birleştirildiğinde eklemli etmenlerin yaptıkları hareketlerin sonuçlarını algılayıp, verilen hedef doğrultusunda öğrenme gerçekleştirmesi, dolayısıyla da hareketlerini kontrol edebilmesi, sağlanabilir.

Q-öğrenmenin başarılı bir şekilde gerçekleştirilebilmesi için, daha önce de üzerinde durulmuş olduğu üzere, problemin geniş durum-hareket uzayını daraltmak gereklidir. Bu amaçla tezde her eklemin alabileceği (60 – 30), (30 – 0), (0 – -30) ve (-30 – -60) derece olmak üzere dört açı aralığı vardır. Dolayısıyla her eklemin durum uzayı dört boyuttan oluşmaktadır. Sekiz eklemli etmenin eklemlerinin yarısı, aynı şeyi iki kere öğrenmemek için, birbirleriyle simetrik çalıştığından etmenin eklemlerinden dört tanesinin hesaplamaya katılması yeterlidir. Buna göre eklemlerin bulunabileceği toplam durum sayısı $4^4 = 256$'dır. Eklemlerin açı aralıklarını biraz daha küçültmek daha yakın sonuçlar verebilecekse de bu sefer de problem alanının keşfedilmesinin çok daha uzun zaman almasına neden olacağından, tezde böyle bir seçime gidilmemiştir. Her bir ekleme ilerleme, gerileme ve durma hareketlerini gerçekleştiren, üç farklı torktan biri uygulanabilir. Dolayısıyla, etmenin toplam yapabileceği hareket sayısı 4x3 = 12'dir.

Amaç etmenin hareket ettiği sürece olabildiğince yol kat etmesi olduğunda ödül her adımda alınan mesafe üzerinden verilmiştir. Etmen kafasının doğrultusunda ilerlediğinde, ilerlediği mesafe ile orantılı olarak pozitif bir ödül değeri alır, ters tarafa doğru ilerlediğinde ise ilerlediği mesafe ile orantılı olarak negatif bir ödül değeri alır. Aynı doğrultuda ilerleyemeyen etmenlere verilen ödül giderek azalmaktadır. Şekil 3.10'da verilen doğrultulara göre eğer etmenin yeni doğrultusu ile eski doğrultusu arasındaki fark ikiden küçükse bu iki doğrultuya denk gelen açılar birbirinden çıkarılır, ortaya çıkan değerin mutlak değeri alınır ve kat edilen mesafe bu değere bölünür. Doğrultular arasındaki fark üç ise bu iki doğrultuya denk gelen açılar yine birbirinden çıkarılır, ortaya çıkan değerin mutlak değeri alınır, daha sonra negatif

bir hale getirilir ve kat edilen mesafe bu değere bölünür. Böylelikle tam ters yönde gitmeye başladığında etmen deha çok eksi değer almış olur. Diğer durumlarda ise mutlak değeri alınan toplanıp 180 dereceden çıkarılır ve kat edilen mesafe bu değere bölünür.



Şekil 3.10 Etmenin bulunabileceği doğrultu aralıkları

Öğrenmeye keşif aşamasıyla başlanır. Bu aşamada etmen 12 hareketten birini rasgele olarak gerçekleştirir. Daha sonra eklemlerin bulunduğu açısal konumlar değerlendirilerek etmenin 256 durumdan hangisinde bulunduğu hesaplanır. Bir sonraki adım o anda kazanılan ödülü hesaplamaktır. Ödül hesaplandıktan sonra durum-hareket çiftinin Q değeri bulunur. Alınan durum şimdiki durum olarak belirlenir. Bu işlemler sömürme aşamasına kadar devam eder. Sömürme aşamasının bütün durum-hareket çiftleri keşfedildikten sonra başlaması istenmektedir. Her durum-hareket çifti üzerinden düzgün bir şekilde ilerlenseydi bütün durumların üzerinden bir kere geçmek için 256 x 12 = 3072 adım yeterli olurdu fakat hareketler rasgele olarak yapıldığından keşif aşamasının en az 10000 adımdan oluşturulması gerektiği kararına varılmıştır.

Sömürme aşamasında artık bulunulan durumda daha iyi sonuç veren hareketler üzerinden etmenin ilerlenmesi sağlanmıştır. Yeterince iyi değerlendirilememiş belirli bir harekete takılıp kalmamak için de SOFTMAX yönteminden yararlanılmıştır. Buna göre her hareketin seçilme olasılığı o zamana kadar aldığı artımlı ödülü yani o durum-hareket çifti değeri ile belirlenir. Daha yüksek durum-hareket çifti değeri sağlayan hareketlerin seçilme olasılığı daha yüksektir. Bunun yanında keşif de bir anlamda hala devam etmektedir çünkü seçilen hareketlerin üzerinden etmenlerin aldıkları ödüller ve Q değerleri hesaplanmaktadır ve o durum-hareket çifti değeri yeni hesaplanan değere göre güncellenmektedir. Bundan yola çıkılarak öğrenmenin sürekli hale getirildiği söylenebilir.

Dört bacaklı bir yapay yaşam formunun hareket ettirilebilmesi probleminde Q-öğrenme gibi destekli öğrenme yöntemleri tam olarak bir çözüm üretememektedir. Bunun nedeni gerçekte ortamın ayrık olmayışı ve problemin büyük bir alanda sürekli olarak gerçekleşmesi

gerekmesidir. Yani ortam bir anlamda sınırsızdır. Bu durumda Q-öğrenmenin kullandığı türden bir ödül mekanizmasının yaratılması da çok zor olmaktadır. Üstelik eklem sayısı arttıkça yani yapay yaşam formunun morfolojisi daha da karmaşık bir hal aldığında problem uzayı katlanarak büyüyeceğinden Q-öğrenme yönteminin hareket problemi ile başa çıkması daha da zorlaşacaktır. Bu nedenle etmenin üzerinde denenen bir başka destekli öğrenme yöntemi de Politika Gradyanlı Destekli Öğrenme'dir. Tezde gerçekleştirilecek olan Politika Gradyanlı Destekli Öğrenme yöntemi global bir öğrenme yöntemi olan Q-öğrenmenin aksine yerel bir öğrenme yöntemidir. Bu yöntem uygulanırken Kohl ve Stone'un 2004'te gerçekleştirdikleri çalışmadan yararlanılmıştır.

Dört bacaklı yapay yaşam formunun çapraz çiftleri birbirine simetrik olan sekiz eklemi bulunduğundan yapay yaşam formunu hareket ettiren dokuz parametre vardır. Bu parametreler eklemlere uygulanacak olan torkları belirler. Her parametre setinin yapay yaşam formu tarafından yürütülebildiği kabul edildiğinde parametre setleri açık döngü politikası (open loop policy) olarak tanımlanabilir. Bu problem politika gradyanlı destekli öğrenme ile çözülebilir. Buna göre daha önce de bahsedildiği gibi bir parametre seti p oluşturulur. Bu parametre seti kullanılarak bir hareket yapılır, sonra sıradaki parametre seti ile harekete devam edilir. Her fark (Δ_j) değeri $+e_j$, 0, $-e_j$ değerlerinden biri olmak üzere rasgele bir şekilde seçilmiştir. e_j değerlerinden her biri q_j 'den küçük olan sabit bir değerdir.

Daha sonra her bir parametre için bir güncelleme miktarı bulunur. Dokuz tork parametresinden her biri için ayrı 3'lü grup oluşturulur. Bir sonraki adım da gruplardaki vektörlerin skorlarının ortalamalarına bakarak hangi grubun daha başarılı olduğunu bulmaktır. Örneğin yapay yaşam formunun sol üst bacağını hareket ettiren tork için $+e_j$ grubu başarılı olursa sol üst bacağın torku o grubun temsil değeri kadar güncellenir. Kullanılan yöntem Şekil 3.11'de verilen algoritmada özetlenmiştir.

$\pi \leftarrow$ Başlangıç Politikası
Başarı sağlanana kadar:
$\{R_1,...,R_t\} = t$ rasgele üretilmiş hareket politikaları
$\{R_1,...,R_t\}$ 'yi değerlendir
1'den N'ye kadar:
$Ortalama_{+\varepsilon,n} \leftarrow$ n boyutundaki fark değeri pozitif olan R_i'nin ortalama başarısı
$Ortalama_{0,n} \leftarrow$ n boyutundaki fark değeri sıfır olan R_i'nin ortalama başarısı
$Ortalama_{-\varepsilon,n} \leftarrow$ n boyutundaki fark değeri negatif olan R_i'nin ortalama başarısı
Eğer $Ortalama_{0,n}$ $Ortalama_{+\varepsilon,n}$'dan ve $Ortalama_{-\varepsilon,n}$'dan büyükse:
$A_n \leftarrow 0$
Değilse:
$A_n \leftarrow Ortalama_{+\varepsilon,n} - Ortalama_{-\varepsilon,n}$
$A \leftarrow \frac{A}{|A|} * \eta$
$\pi \leftarrow \pi + A$

Şekil 3.11 Politika Gradyanlı Destekli Öğrenme algoritması

Bu algoritmada A düzeltme vektörünü temsil eder. Yapılan iş ana döngünün her tekrarında p 'nin etrafındaki gradyanı hesaplamak amacı ile p 'nin yakınında olan t kadar politikanın örneklenmesi ve en tercih edilir doğrultuda p 'nin η kadar hareket ettirilmesidir.

Örneğin N boyutunun 3 olduğunu varsayalım ve R_1'in aldığı mesafenin 5, R_2'nin 1, R_3'ün ise 3 birim olduğunu ve sadece birinci gözler için hesaplandığında R_1'in 10+3 değeri alarak $+e$ grubuna, R_2'nin 10+0 değeri alarak 0 grubuna, R_3'ün ise 10-3 değeri alarak $-e$ grubuna girdiğinin gözlemlendiğini varsayalım. Gerçekte her grupta birden fazla R vektörü olduğu unutulmamalıdır. Bir sonraki adım R'lerin skorlarının ortalamasını almaktır. Buna göre $Ortalama_{+e,1}$ 5 birim, $Ortalama_{0,1}$ 1 birim, $Ortalama_{-e,1}$ ise 3 birim olur. Bu örnekte $+e$ düzeltmesi en çok yolu aldırmıştır. Öyleyse algoritmada belirtilen formül uygulandığında hareket vektörünün birinci torku için 5-3=2 birimlik bir düzeltme yapılır. Örnekte bu basit bir halde anlatılmıştır, program uygulanırken düzeltme miktarı normalize edilip η ile çarpılır. Bu işlem diğer 3 tork bileşeni için de tekrarlanır. Bir eğitim döngüsü sonunda hareket vektörü bir miktar güncellenir. Bu işlem tekrarlandıkça yeterince iyi bir hareket vektörü elde edilir.

Tezde fark değeri 15 dereceye denk gelecek şekilde belirlenmiştir, hareket politikası sayısı 20, eğitim döngüsü 200 ve η 2 olarak uygulanmıştır.

3.4 Uygulama Sonuçları

Bu bölümde üç farklı öğrenme yöntemi ile eğitilen yaşam formunun yaptığı hareketlerin sonuçlarına ve gerçekleştirdiği kimi hareketleri gösteren örneklere yer verilmiştir. Bunu gerçekleştirmek için her bir öğrenme yöntemi için yapılan 10 deneyin sonuçlarına yer verilmiş ve bu deneylerden iyi, kötü ve ortalama olmak üzere üçü görsel olarak sergilenmek için seçilmiştir. Böylelikle üzerinde çalışılan öğrenme yöntemleri kullanıldıktan sonra yapay yaşam formunun yaptığı hareket dizisi yorumlanmış ve ardışık bir şekilde gösterilmiştir.

3.4.1 Genetik Algoritma

Yapay yaşam formu hareket etme şeklini Genetik Algoritma ile öğrendiğinde sergilediği hareket keskin hatlara sahiptir yani yapay yaşam formunun nasıl hareket ettiği belirgindir ve yaptığı hareket dizisi kolaylıkla anlaşılabilmektedir. Genetik Algoritma ile eğitimi sağlanan yapay yaşam formu yavaş ve emin adımlarla ilerliyormuş izlenimi vermektedir.

Yapılan hareketler başlangıçta rasgele seçilen gen dizisine göre değişkenlik göstermektedir. Bu nedenle öğrenme sonucunda gözlemlenen hareket dizileri birbirlerine çok benzememektedirler. Hatta aynı eğitim süreci içinde yakınsama tamamlanmış gibi göründükten sonra bile kritik bir yerde şans eseri birkaç kere mutasyona uğrayan kromozomlar sonradan farklı hareketlere neden olabilmektedirler. Böylesine bir duruma rastlanması düşük bir olasılık da olsa göz ardı edilmemelidir. Yine de yapılmaya çalışılan hareket genel hatlarıyla benzerlik göstermektedir. Bu benzerlik hareketlerin sıralamasındaki benzerlikten kaynaklanmaktadır. Yapay yaşam formu genelde bir adımda sol ön (simetrik bir yapı kullanıldığından aynı zamanda sağ arka) bacağını, bir sonraki adımda ise diğer bacakları hareket ettirerek dört bacaklı organizmaların adım sıralamasına benzer bir hareket dizisi sergilemektedir. Fakat bacaklara uygulanan kuvvetler ve hareketlerin tam sıralaması diğer öğrenmelere göre farklı olduğundan yapılan hareketlerin yürümeye benzerliği değişkenlik göstermektedir. Bunun dışında, daha az da rastlanıyor olsa, öğrenilen bir diğer hareket biçimi zıplamaya benzerlik gösterir. Bu davranış biçiminde ana kuvvet arka ayaklara uygulanıyor görünümündedir.

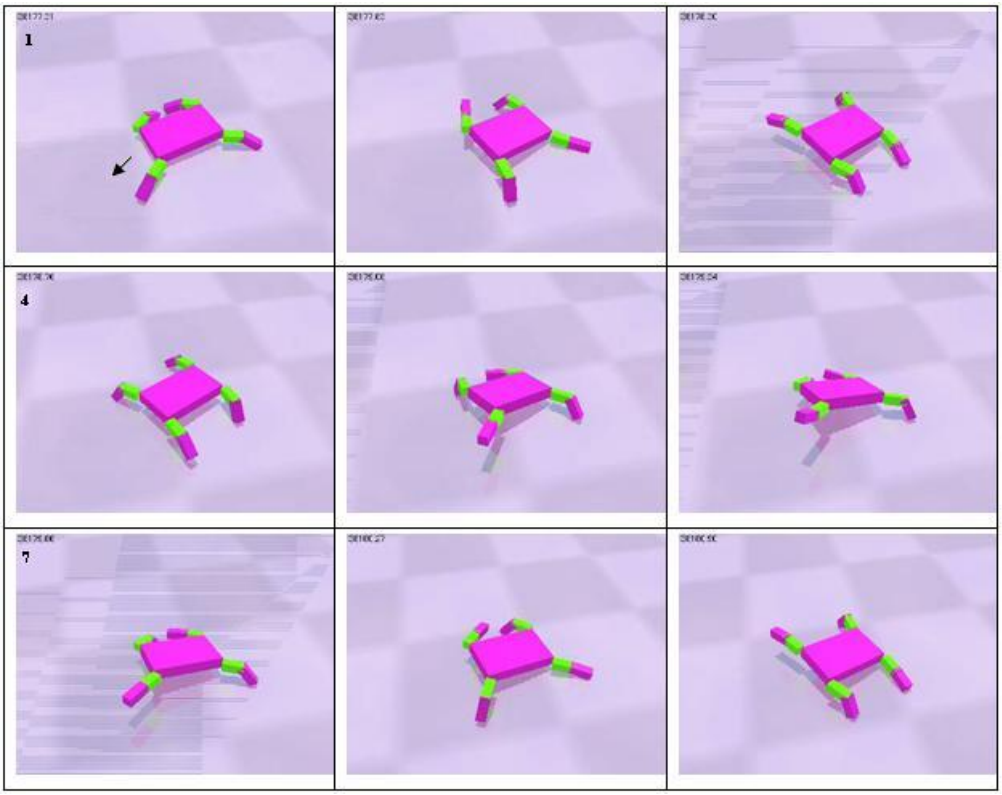
Yapılan denemelerin sonuçlarının hiçbirinde ilerlenen doğrultuda önemli bir sapma fark edilmemiştir fakat böylesine bir gözlem çoğunlukla istenen uzaklık başarı ölçütü sağlanamadığından görsel ölçüt eşiği geçildiğinde öğrenmenin sonlandırılmış olmasından kaynaklanıyor olabilir. Yine de eğitim süreci boyunca iyi sonuçlar vermekte olan yani yapay yaşam formunun daha uzun mesafeler ilerlemesini sağlayan hareket dizilerinin genelde

doğrultu değiştirmeye sebep olmadığı gözlemlenmiştir. Genetik Algoritma ile öğrenmenin en başarısız olduğu nokta öğrenme hızı olmuştur. Eğitim yavaş gerçekleşmektedir. Birbirinden bağımsız 10 deney sonucunda elde edilen veriler Çizelge 3.1'de verilmiştir.

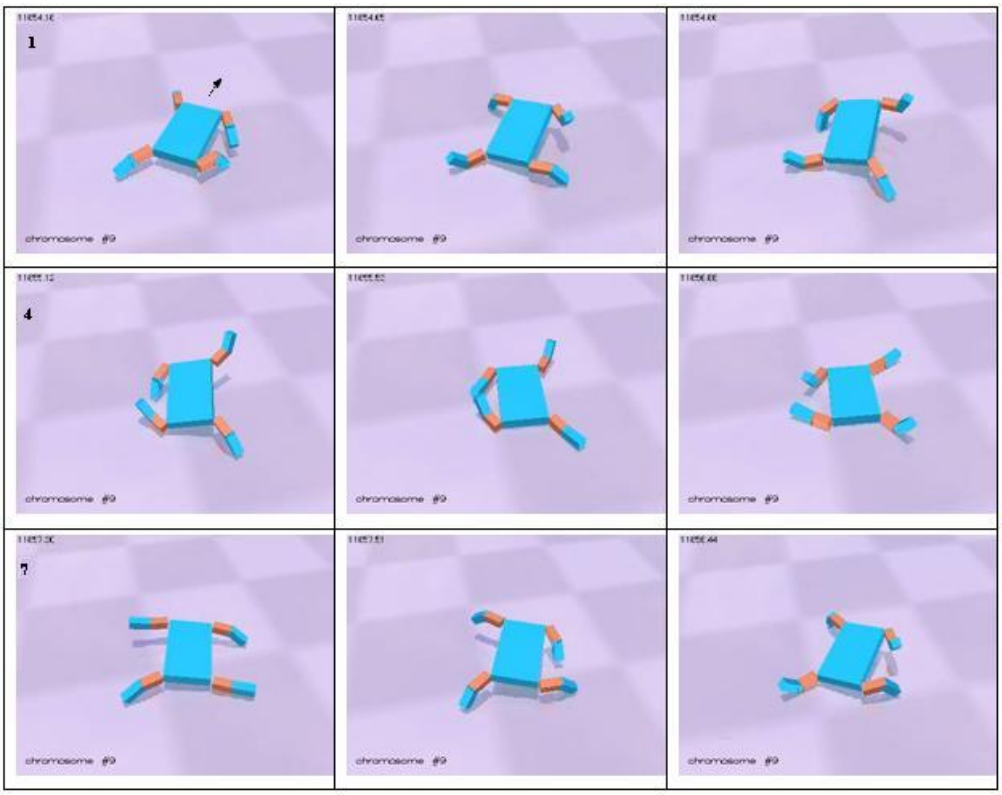
Çizelge 3.1 GA sonuçları

Denemeler	Öğrenim gerçekleşene kadar geçen zaman (sn.)	Kat edilen mesafe (50 saniyede)	Gerçeğe yakınlık (10 üzerinden)
1	46250	109,37	6
2	9990	63,54	9
3	38170	82,76	8
4	11850	75,65	6
5	13770	73,55	5
6	28510	77,46	7
7	25980	74,91	7
8	28615	71,37	6
9	25815	79,39	5
10	7365	53,35	6
Ortalama:	23630	76,14	6,5

Bu deneylerden iyi sonuç veren hareket dizilerinden biri Şekil 3.12'de gösterilmiştir. Bu şekildeki yaşam formu gösterilen doğrultuda oldukça düzgün, yürümeye benzer bir hareket dizisi sergilemektedir. Yapay yaşam formu önce ön sol bacağını (dolayısıyla sağ arka bacağını) ileriye doğru uzatır, sonra bu bacakları bükerek geriye çekerken diğer bacakları öne doğru uzatır. Bu hareket yapılırken yaşam formu hep ayaktadır. Doğrultuda bir sapma oluşmaz.



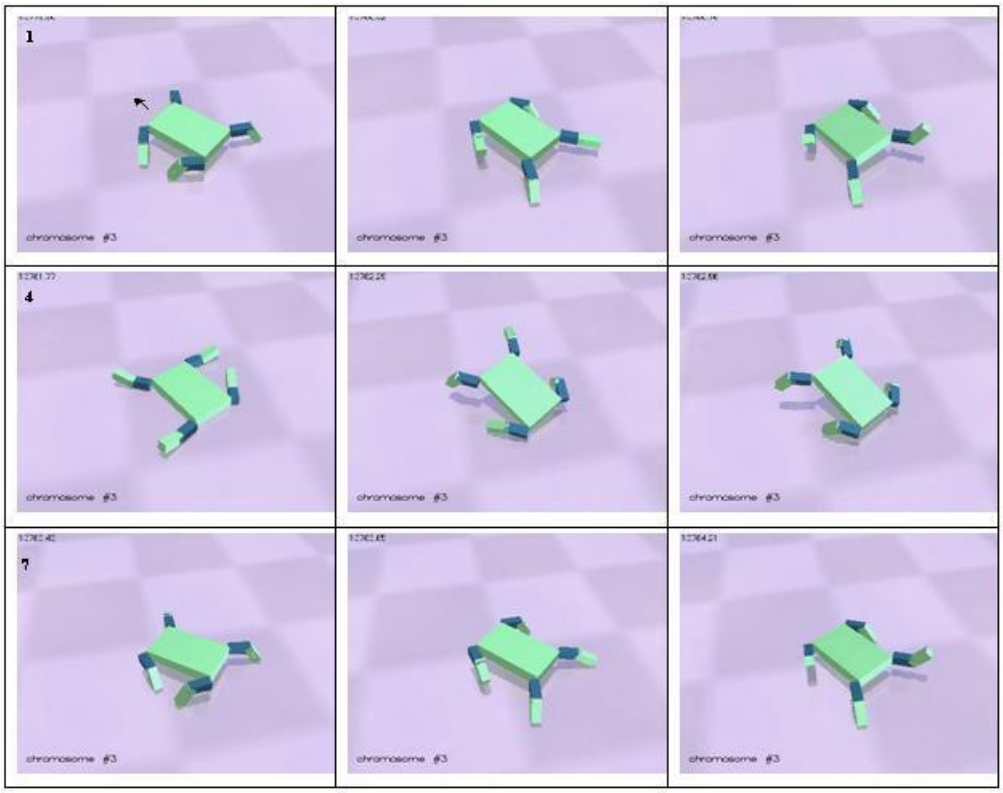
Şekil 3.12 GA iyi hareket dizisi



Şekil 3.13 GA ortalama hareket dizisi

Şekil 3.13'te de genetik algoritma ile eğitilen yapay yaşam formunun gerçekleştirmiş olduğu

bir diğer hareket dizisi gösterilmiştir. Yapay yaşam formu yine bir bacağını öne atarken diğerini geri çekmektedir. Fakat bu sefer her iki adım sonrasında vücudu tamamıyla yere değmektedir. Doğrultuda sapma gözlemlenmemiştir. Şekil 3.14'teki örnek en kötü sonuç vermiş olan genetik algoritma deneyine aittir. Yapay yaşam formu bütün bacaklarını senkronları bozulmuş halde aynı yönde hareket ettirmektedir. İlerleme vücudun ön kısmı ve ön bacaklar biraz yukarı kaldırıp arka bacaklar ile ittirilerek sağlanmaktadır. Doğrultuda sapma gözlemlenmemiştir.



Şekil 3.14 GA kötü hareket dizisi

3.4.2 Yapay Sinir Ağları

Tezde hem İleri Beslemeli Yapay Sinir Ağları hem de Yinelenen Yapay Sinir Ağları yapılarından faydalanmıştır. Daha önce de bahsedildiği gibi İleri Beslemeli Yapay Sinir Ağları içsel durumları saklamaz. Oysa bu tezde üzerinde çalışılan yapay yaşam formu durumsuz bir girdi-çıktı sistemi olarak düşünülmemiştir. Bu nedenle Yinelenen Yapay Sinir Ağları mimarilerinden yararlanılması daha iyi sonuçlar elde edilmesini sağlamıştır. Dolayısıyla bu bölümde yapay sinir ağlarının kullanımından elde edilen sonuçlar arasından Gerçek Zamanlı Yinelenen Öğrenme ile eğitilen yapay yaşam formunun yaptığı hareket dizilerine yer verilmiştir.

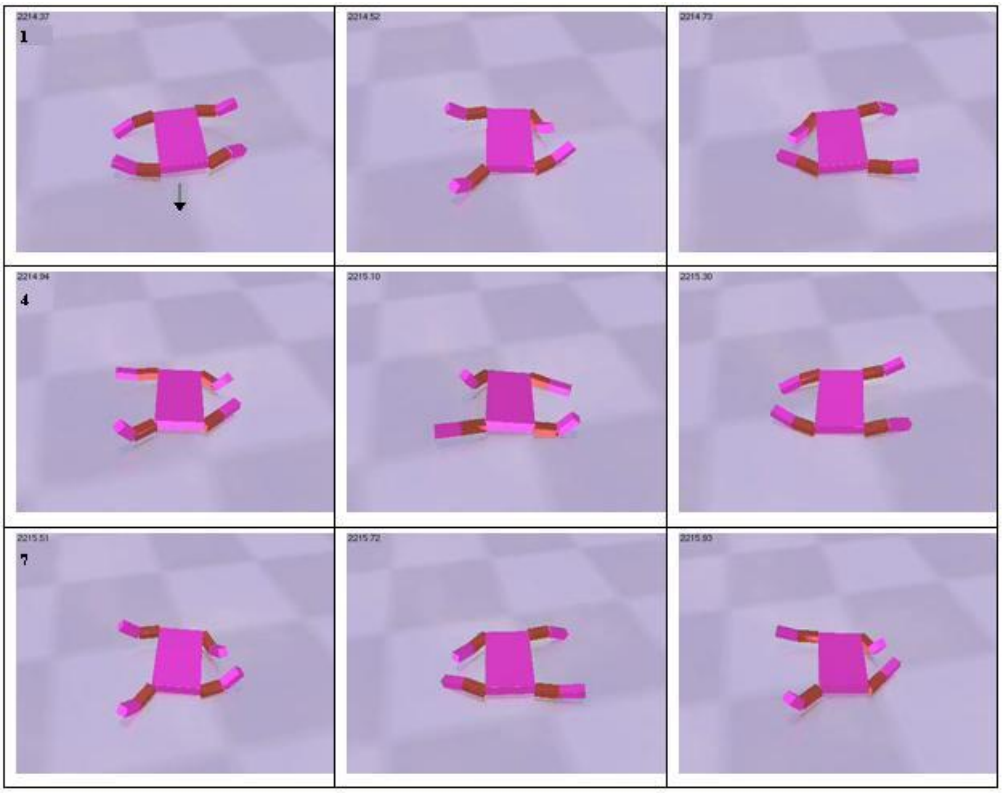
Gerçek Zamanlı Yinelenen Öğrenme ile eğitilen yapay yaşam formunun hareketleri hızlı ve akıcıdır. Geliştirilen hareketler diğer yöntemlerin sonuçlarına göre daha dairesel ve yumuşak bir görüntüye sahiptir. Hareketin bu yapısı ve yapay yaşam formunun hızlı ilerlemesi sonucunda Genetik Algoritma ile öğrenmedeki gibi belirgin hareketler gözlemlenmemektedir. Yine de yapılan hareketlerin genel özellikleri bellidir. Öğrenme sonucu ya çapraz bacaklar hareket ettirilmektedir ya da, daha az rastlanmakla beraber, kuvvetin daha çok arka bacaklara uygulandığı izlenimini yaratan zıplamaya benzer hareketler gerçekleşmektedir. Yapay yaşam formunun hareketleri daha çok karada değil de yerçekiminin daha az olduğu bir ortamda yapılıyormuş görüntüsüne sahiptir. Öğrenilen farklı hareketlerin dairesel görünümü ve hızları birbirlerine benzerlik gösterse de benzer hareketlerin öğrenildiği söylenemez.

Yapay yaşam formu Gerçek Zamanlı Yinelenen Öğrenme yöntemi ile eğitildiği takdirde çok hızlı ilerlemektedir fakat aynı zamanda hareketin doğrultusunda da çoğu zaman sapmalara rastlanmaktadır. Bu nedenle belli bir süre sonra yapay yaşam formunun problem uzayındaki ilk noktasına yakın bir noktaya dönme olasılığı mevcuttur. Dolayısıyla son nokta ile ilk nokta arası mesafeye bakıldığında beklenenden az olduğu görülebilir. Bu durum Gerçek Zamanlı Yinelenen Öğrenme ile yapay yaşam formunun eğitimi sırasında tek başarı ölçütünün ilerlenen mesafe olmaması, aynı zamanda doğrultuya da yer verilmesi gerekliliğini göstermektedir. En iyi sonucu veren uygulamalarda birinde ise hep aynı doğrultuda ilerlemekte olan yapay yaşam formu bir süre sonra doğrultuda bir atlama yapıp yan dönmektedir. Gerçek Zamanlı Yinelenen Öğrenme ile eğitilen yapay yaşam formu hızlı ilerleme sağlayıp güzel görüntü verebiliyor da olsa doğrultusunun önemli miktarlarda değişebiliyor olmasından dolayı dengeli bir yapıya sahip değildir. Birbirinden bağımsız 10 deney sonucunda elde edilen veriler Çizelge 3.2'de verilmiştir.

Çizelge 3.2 GZYÖ sonuçları

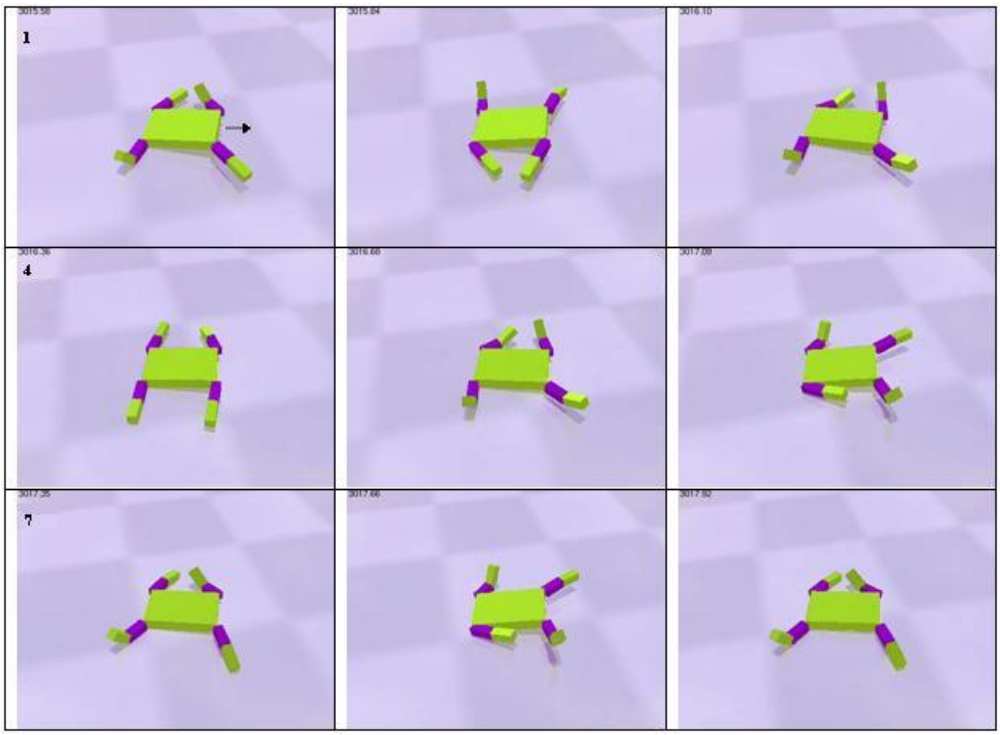
Denemeler	Öğrenim gerçekleşene kadar geçen zaman (sn.)	Kat edilen mesafe (50 saniyede)	Gerçeğe yakınlık (10 üzerinden)
1	3880	188,69	8
2	2100	252,69	9
3	3700	220,56	8
4	2200	77,53	5
5	2900	86,88	6
6	1380	40,93	7
7	520	25,54	8
8	190	79,49	6
9	120	143,26	5
10	1600	157,24	7
Ortalama:	1859	127,28	6,9

Yapay yaşam formu gerçek zamanlı öğrenme yöntemi ile eğitildiğinde gözlemlenmiş iyi sonuçlardan biri Şekil 3.15'te gösterilmiştir. Bu deney sonucunda bacaklar çapraz olarak hareket ettirilmektedir. Bir bacak öne doğru savrulurken diğeri geri çekilmektedir. Bu sefer gerek üst gerek alt bacaklar küçük açılarla hareket ettirilmiştir. Bu da dairesel bir görüntü yaratmıştır. Açıların küçük olması aynı zamanda yaşam formunun yere çok yakın bir pozisyonda ilerlemesine neden olmaktadır. Uzun bir süre aynı doğrultuda ilerlenmiş olup sonradan doğrultuda değişiklik yaşanmıştır.

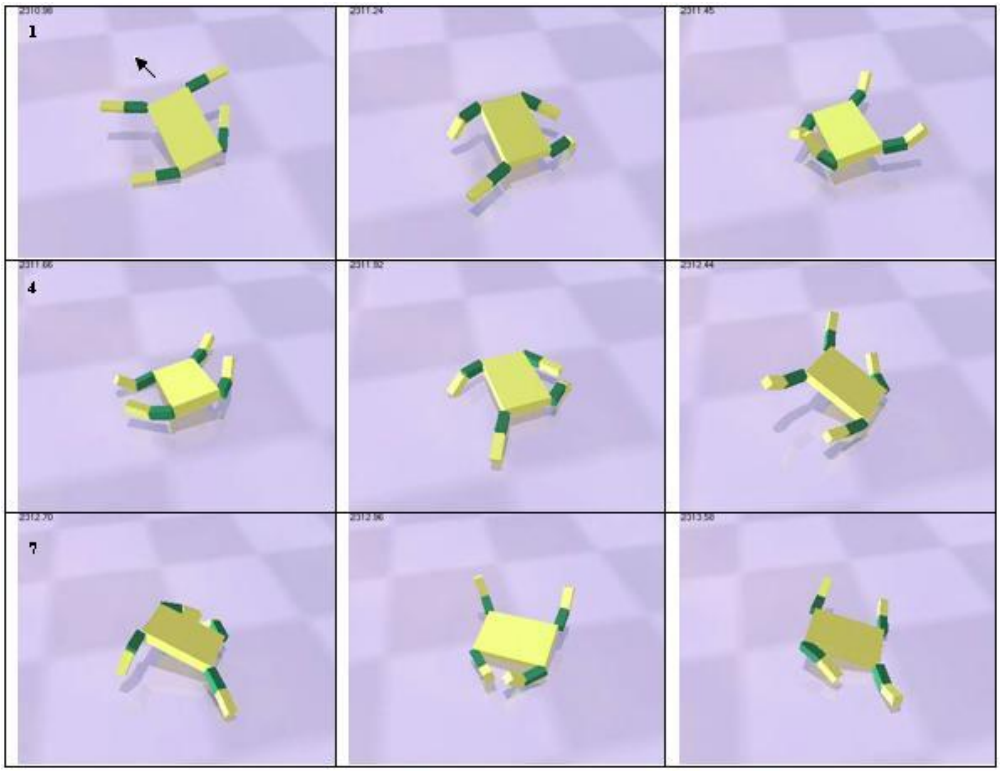


Şekil 3.15 GZYÖ iyi hareket dizisi

Şekil 3.16'da gösterilen deney sonucu ortalama bir sonuçtur. Yine çapraz hareket yapılsa da sağ ön bacak (dolayısıyla da simetriği olan sol arka bacak) hareket hep önde, diğerleri ise daha arkada gerçekleştirmektedir. Açı aralığı biraz daha genişlemişse de diğer yöntemlerde elde edilenlere göre hala küçüktür. Yapay yaşam formunun vücudunun arka tarafı yerle sürekli temas halindedir. Zaman ilerledikçe doğrultuda sapma fark edilebilmektedir.



Şekil 3.16 GZYÖ ortalama hareket dizisi



Şekil 3.17 GZYÖ kötü hareket dizisi

Şekil 3.17'de ise kötü bir hareket dizisi örneklenmiştir. Eklemler yine dairesel hareketler yapmaktadırlar. Yapay yaşam formunun vücudunun ön bölgesi daha çok olmak üzere vücudu yere değip havalanmaktadır. Hareket dizisi tam olarak ne yürüyüşü ne de zıplamayı andırmaktadır. Onun yerine her iki hareket stilinin özellikleri birbirine karışmış izlenimi

yaratmaktadır. Doğrultu sıklıkla değişmektedir.

3.4.3 Destekli Öğrenme

Yapay yaşam formunun hareket uzayı sonsuz denebilecek kadar büyük olduğundan Q-Öğrenme ile eğitilmek istendiğinde bu uzayın ayrık hale getirilmesi gerekmiştir. Bu da büyük bir genelleştirmeye neden olmuştur. Daha önceden iki bacağı ve daha az serbestlik derecesi olan yapılarda kullanılmış ve sonuçlar vermiş olan bu yöntem tezde yararlanılan dört bacaklı yapay yaşam formuna uygulandığında başarısızlığa uğramıştır. Bu problemin altında yatan ana neden problem uzayının genişlemesidir. Bu nedenle bu bölümde diğer bir destekli öğrenme yöntemi olan ve daha başarılı sonuçlar veren Politika Gradyanlı Destekli Öğrenme'nin sonuçlarına yer verilmiştir.

Yapay yaşam formunun hareketini sağlayabilmek için Politika Gradyanlı Destekli Öğrenme uygulandığında yapay yaşam formunun her bir adımda olabildiğince uzağa gitmeye çalıştığı gözlemlenmiştir. Bu amaçla yapay yaşam formu ya mümkün olduğu kadar büyük adımlar atmaktadır ya da zıplama benzeri bir hareketle bulunduğu noktadan uzaklaşmaya çalışmaktadır. Yapay yaşam formu böylelikle yakın görüşlü bir strateji izliyormuş görünümünü vermektedir ki yerel en iyi sonuca ulaşılmaya çalışıldığı göz önüne alındığında bu şaşırtıcı bir gözlem olmaktan çıkmaktadır. Hareketler Genetik Algoritma ile öğrenmenin sonuçlarına benzer şeklide belirgindirler ve hızlı değillerdir.

Genelde öğrenilen adım atış şekli doğada gözlemlenen sağ ve sol bacakların dönüşümlü kullanılmasını andırmaktadır; sadece adımlar beklenenden daha geniştirler. Bu nedenle yapay yaşam formu karşıt çapraz hareketler yaparak aynı doğrultuda kalmayı başarır. Aynı hareketin daha küçük ve sık bir şekilde yapılması sonucun başarısını arttırabilecek gibi görünmektedir.

Ancak yerel en iyiye ulaşılabiliyor olma problemi uygulamada azaltıldığından farklı başlangıç vektörleri ile başlansa da birbirlerine çokça benzer sonuçlara ulaşılmıştır. Bu durum bir anlamda sürprizlerle karşılaştırma olasılığının fazla olmadığını gösterdiği için daha sonra da bu öğrenme yönteminin kullanımının tercih edilme sebebini oluşturabilir.

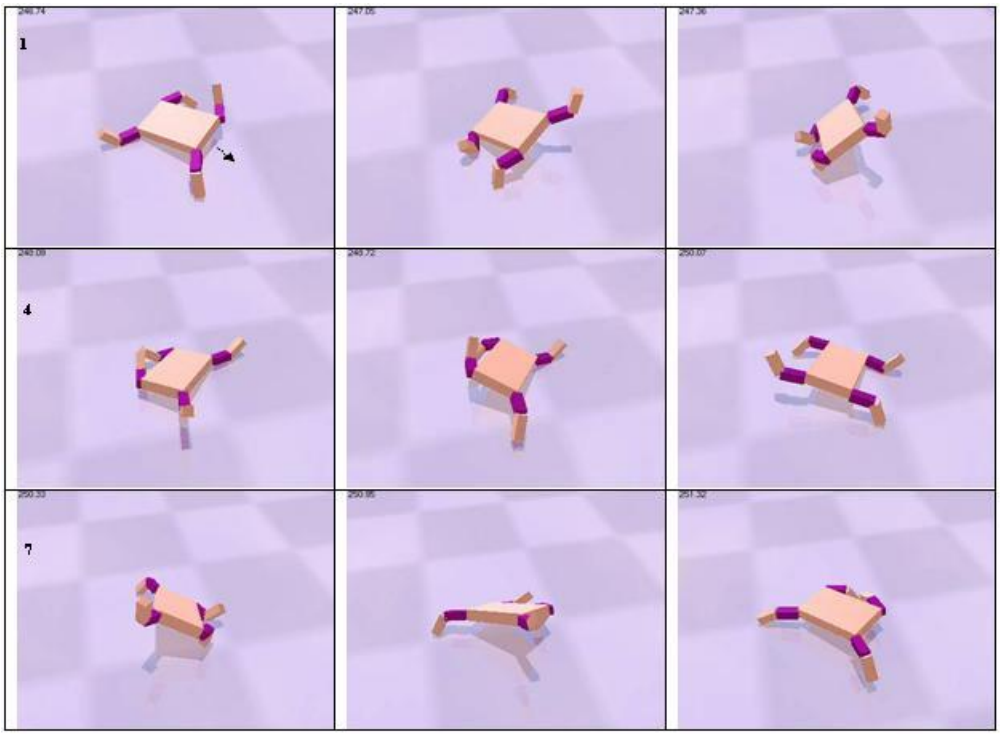
Uzun mesafe ilerlen denemelerde yörüngede sapma gözlemlenmemektedir. Diğerlerinde de sapma büyük oranlarda değildir. Bu nedenle yapay yaşam formu Politika Gradyanlı Destekli Öğrenme ile hareket etmeyi öğrendiği hiçbir deney sonucunda etrafında dönermişçesine bir hareket sergilememektedir. Politika Gradyanlı Destekli Öğrenme eğitimi kısa sürmektedir yani öğrenim erken tamamlanmaktadır. Birbirinden bağımsız 10 deney sonucunda elde edilen

veriler Çizelge 3.3'te verilmiştir.

Çizelge 3.3 PGDÖ sonuçları

Denemeler	Öğrenim gerçekleşene kadar geçen zaman (sn.)	Kat edilen mesafe (50 saniyede)	Gerçeğe yakınlık (10 üzerinden)
1	2290	73,72	5
2	915	64,02	6
3	230	79,45	5
4	3660	44,97	6
5	2976	65,09	4
6	2520	56,54	6
7	230	53,06	6
8	685	35,12	3
9	230	53,20	5
10	460	52,57	4
Ortalama:	1420	57,77	5,0

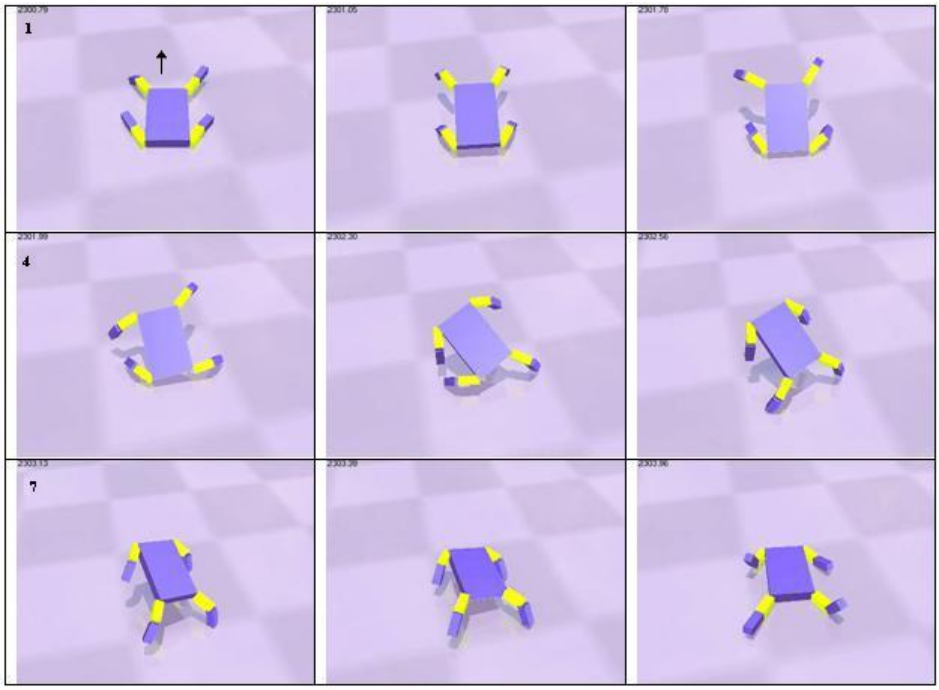
Bu deneylerden iyi sonuç veren hareket dizilerinden biri Şekil 3.18'de gösterilmiştir. Bu deneyde yapay yaşam formu bir ayağını ileri uzatırken diğerini geri çekmektedir. Adımlar büyük ve çaprazdır. Tek bir adımdan sonra yön değişse de ikinci adımda yön karşı tarafa doğru sapma yaptığından iki adım sonrasında doğrultu sabit kalmaktadır. Yaşam formunun vücudunun arka kısmı yerle devamlı temas halindedir. Adımların büyüklüğü fazla enerji harcıyormuş izlenimi yaratmaktadır.



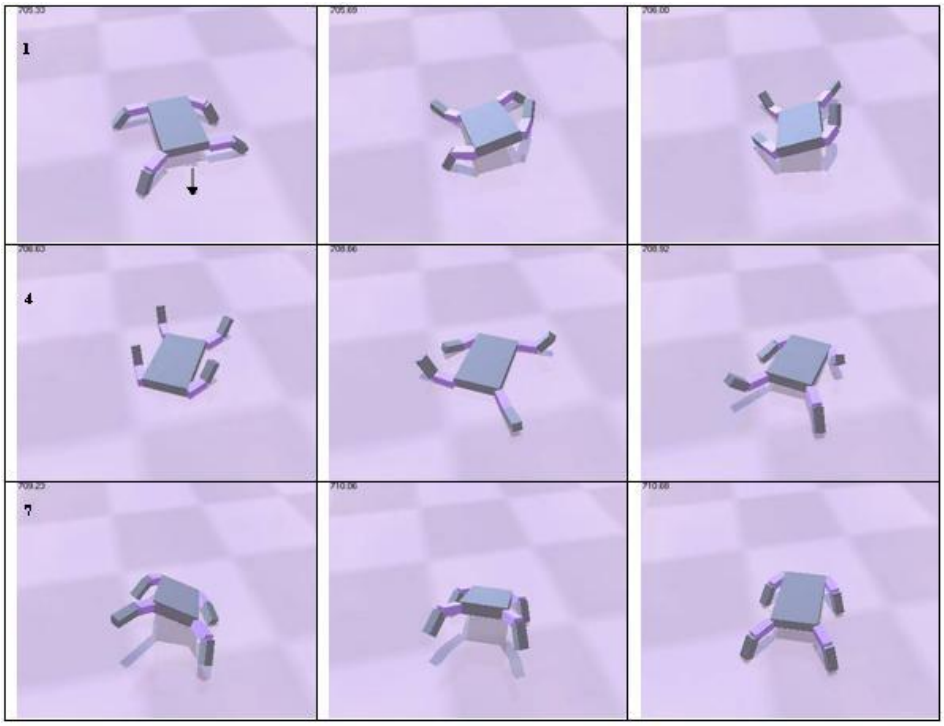
Şekil 3.18 PGDÖ iyi hareket dizisi

Ortalamada bir sonuç sağlayan deneyde (Şekil 3.19) yapay yaşam formunun bütün bacakları

küçük senkronizasyon farklılıkları ile aynı hareketleri yapmaktadır. Arka bacaklar vücudu ittirip havalandırdığından zıplamaya benzer bir hareket yapılmış olur. Her bir zıplama sonucu yapay yaşam formunun tüm vücudu yerle temas etmektedir. Doğrultu adım süresince sapma gösteriyormuş gibi görünse de adım tamamlandığında doğrultuda değişiklik olmadığı görülür.



Şekil 3.19 PGDÖ ortalama hareket dizisi



Şekil 3.20 PGDÖ kötü hareket dizisi

Politika gradyanlı destekli öğrenme sonucu gerçekleşen kötü bir hareket dizisi örneği Şekil

3.20’de gösterilmiştir. Bu örnekte yapay yaşam formunun ön kısmı biraz havalandırıp sonra tamamını yere temas ederek ilerleme sağlamaktadır. Hareket oldukça yavaştır ve doğrultuda sapma gözlemlenmektedir.

4. SONUÇLAR VE ÖNERİLER

Dinamik hareket etme problemi öğrenme yöntemleri üzerinde çalışmak için çok elverişli bir problemdir. Bu durum hareket probleminin zor bir öğrenme probleminin bütün parçalarını içermesinden kaynaklanmaktadır. Zorluğun öncelikli nedeni hareket eden yapay yaşam formunun bünyesinde birçok bağımsızlık derecesi bulundurmasıdır; bu yapı, yapay yaşam formunun her olası durumuna göre performans en iyileştirmesi yapmak istemesini beraberinde getirir. Bir diğer neden ise bir hareket problemini öğrenmeye çalışan yöntemlerin aynı zamanda yeryüzü ile yaşanan çarpışmalardan kaynaklanan süreksiz durumlarla baş edebilmesi gerekliliğidir. Önemli nedenlerden bir başkası da öğrenme yöntemlerinin aynı zamanda ertelenen ödül sorunuyla karşılaşacak olmasıdır; bu sorun belli bir anda yapay yaşam formuna uygulanan torkların daha ileriki adımlarda performansa etki edebilecek olmasıdır. Bu nedenle bu tezde dört bacaklı bir yaşam formunun hareket etmeyi öğrenmesi üzerine çalışılmıştır.

Üzerinde çalışılan yapay yaşam formu eklemli bir yapıya sahip olacak şekilde tasarlanmıştır. Öğrenme de bu yapı üzerinden gerçekleştirilmiştir. Öğrenmenin gerçekleştirilebilmesi için başarı ölçütü etmenin kat ettiği mesafe olarak tanımlanmıştır. Belirlenen başarı ölçüyle öğrenme gerçekleştirildikten sonra elde edilmek istenen sonuç üzerinde çalışılan yapay yaşam formu için fiziksel olarak gerçekçi görünen hareketler elde etmenin yanında yapay yaşam formunun bu hareketleri yaptığında bulunduğu konumdan olabildiğince uzaklaşmasıdır.

Doğada çeşitli hareketler gözlemlemek mümkündür. Canlı varlıklarda bunlardan en çok rastlanılanları sürünme, kayma, zıplama ve yürümedir. Bu çalışmada hareketin türü hakkında bir kısıtlama yoktur. Yaratılan yaratığın morfolojisi bu hareketlerin hangisini yapmaya daha yatkınsa ona benzer hareketlerin gözlemlenmesi beklenmektedir. Üzerinde çalışılan yapay yaşam formu her bacağı iki serbestlik derecesine sahip dört bacaklı bir yapıya sahip olduğundan ve simülasyonu yapılan çevrenin yerçekimi deniz kenarındaki bir kara ortamındakine benzetildiğinden gerçekleştireceği hareketin daha çok yürüme veya zıplamaya benzemesi beklenmektedir.

Bu tezde doğanın benzetilmesi esas alınmıştır. Bu amaçla, canlıların işleyişini ele alan, doğayı modellemeye çalışan üç önemli yöntem ele alınmıştır. Bunlar evrimsel öğrenme, denetimli öğrenme ve destekli öğrenmedir.

Tezde üzerinde durulan öğrenme alanlarından ilki evrimsel öğrenmedir. Evrimsel öğrenme, özellikle bilinmeyen ortamlarda çalışıldığında, önemli bir araçtır. Genel yaklaşımı gerçek

evrimsel teoriye benzer bir yapıya sahiptir; üzerinde çalışılan programın problemin amacına ne kadar uygun olduğuna karar vermek amacıyla bir uygunluk fonksiyonu vardır ve farklı çözümler sağlayabilmek amacı ile gen dizilerine benzer diziler üzerinde çaprazlama ve mutasyon işlemleri gerçekleştirilir. Tezde yapay yaşam formuna yaygın bir evrimleşme yöntemi olan Genetik Algoritma uygulanmıştır.

Yapay yaşam formuna uygulanan bir diğer öğrenme yöntemi denetimli öğrenmedir. Denetimli öğrenmede alınmak istenen sonuçlar tanımlıdır; öğrenmenin sonunda bu sonuca en yakın veriler elde edilmeye yani hata olabildiğince azaltılmaya çalışılır. Algılayıcılar ve eyleyiciler arasında direk bir eşleme sağlama imkanından dolayı hareketin kontrol edilmesinde yapay sinir ağları kullanmanın faydalı olacağı düşüncesinden yola çıkılarak bu tezde denetimli öğrenme yöntemlerinden Geri Yayılımlı ve Yinelenen Sinir ağlarından yararlanılmıştır.

İleri Beslemeli Sinir Ağlarının aksine Yinelenen Yapay Sinir Ağları eski girdilere duyarlı olabilir ve onlara uygun davranabilir. İleri Beslemeli Sinir Ağları sadece girdi yöneylerini statik bir biçimde eşler. Üzerinde çalışılan yapay yaşam formunda ise durumsuz bir girdi-çıktı sistemi olmadığı, bunun yerine çok boyutlu ve doğrusal olmayan bir sistem bulunmaktadır. Bu nedenle yapay yaşam formunun dinamik hareketini modelleyebilmek için içsel durumları saklayabilecek ve karmaşık dinamikleri uygulamaya geçirebilecek bir sistem kullanmak önemlidir. Bu amacı gerçekleştirmek üzere Yinelenen Yapay Sinir Ağları kullanılmıştır. Bu öğrenme algoritmalarında kısa dönem belleğin yaratılmasını sağlayacak geribildirim bağlantıları oluşturulur ve zaman gecikmelerinden faydalanılır. Kısaca, Yinelenen Yapay Sinir Ağlarının İleri Beslemeli Sinir Ağlarından farkı sadece girdi alanında değil aynı zamanda iç alanda çalışmalarıdır. Bu nedenle Yinelenen Gerçek Zamanlı Öğrenme ile eğitilen yapay yaşam formları Geri Yayılımlı Öğrenme ile eğitilenlere göre her anlamda daha iyi sonuçlar vermiştir. Dolayısıyla da diğer yöntemlerle karşılaştırılırken bu sonuçlardan faydalanılmıştır.

Tezde kullanılan bir diğer öğrenme yöntemi ise destekli öğrenmedir. Destekli öğrenmenin temelindeki fikir istenen davranışı elde etmeye yönelik tepkileri ödüllendirerek davranışı koşullandırmaktır. Destekli öğrenme bir öğretici gerektiren fakat çıktının ne olması gerektiğinin ifade edilmediği bir yöntemdir. Bunun yerine, üretilen çıktının doğru veya yanlış olduğu söylenir ve sistem, doğru çıktılar üzerinden ilerler. Tezde Q-öğrenme ve Politika Gradyanlı Destekli Öğrenme olmak üzere iki destekli öğrenme yöntemi uygulanmıştır.

Q-öğrenme yöntemleri motor kontrolü problemlerine bir çözüm getirse de genelde problem

uzayı sorunundan da anlaşılacağı gibi yüksek boyutlu hareket sistemleri genelde ölçeklenemez. Bu nedenle sunulan probleme tam olarak bir çözüm üretememektedir. Bunun nedeni gerçekte ortamın ayrık olmayışı ve problemin büyük bir alanda sürekli olarak gerçekleşmesi gerekmesidir. Yani ortam bir anlamda sınırsızdır. Bu durumda Q-öğrenmenin kullandığı türden bir ödül mekanizmasının yaratılması da çok zor olmaktadır. Politika Gradyanlı Destekli Öğrenme yöntemi bu anlamda daha iyi sonuçlar verdiğinden diğer yöntemlerle karşılaştırılırken bu sonuçlardan faydalanılmıştır.

Bu yöntemler uygulandığında kendi içlerindeki en iyi sonuç veren yöntemler ve parametreler seçildiğinde ortaya Çizelge 4.1'de gösterilen tablo çıkar. Burada gösterilen sonuçlar her bir yöntem için yapılan 10 deneyin ortalamasından oluşmuştur.

Çizelge 4.1 Yöntemlerin ortalama sonuçları

Yöntem	Öğrenim gerçekleşene kadar geçen zaman (sn.)	Kat edilen mesafe (50 saniyede)	Gerçeğe yakınlık (10 üzerinden)
GA	23630	76,14	6,5
GZYÖ	1855	127,28	6,9
PGDÖ	1420	57,77	5,0

Bu çizelgeden de anlaşılacağı gibi öğrenme süresi en yavaş yöntem Genetik Algoritma olmuştur. Hatta eğitim hızı diğer iki yönteme göre çok yavaştır. Bu sonuç Genetik Algoritma için büyük bir dezavantajdır. Gerçek Zamanlı Yinelenen Öğrenme ile Politika Gradyanlı Destekli Öğrenme arasında zaman açısından bir karşılaştırma yapılabilecek kadar bir zaman farkı olmamıştır. Yapay yaşam formu en çok gerçek zamanlı yinelenen öğrenme ile ilerleme sağlamıştır. Hatta gösterilen ilerleme miktarı bile bu yöntem ile eğitilen yapay yaşam formunun hızını ifade etmekte eksik kalmaktadır çünkü yapay yaşam formu çok hızlı ilerliyor da olsa aralarda sapma yaptığı için aynı doğrultuda ilerleyememiştir. Genetik Algoritma ile eğitilen yapay yaşam formları Politika Gradyanlı Destekli Öğrenme ile eğitilenlere göre daha çok mesafe kat etmiş görünmektedir fakat bu çok belirgin bir fark olmadığından yapılan denemelere göre de değişkenlik göstermiş olabileceği göz önüne alınmalıdır. Yöntemler uygulandığında görsel açıdan farklı sonuçlar sergilemiş olsalar da gerçeğe yakınlıklarının birbirlerine yakın olduğu görülmüştür.

Genetik Algoritma ve Politika Gradyanlı Destekli Öğrenme ile eğitilen yapay yaşam formlarının daha belirgin ve yavaş ilerledikleri fakat doğrultuda pek sapma yapmadıkları, Gerçek Zamanlı Yinelenen Öğrenme ile eğitilen yapay yaşam formlarının ise yumuşak ve dairesel görünümlü hareketlerle çok daha hızlı ilerledikleri fakat doğrultularını zaman zaman

değiştirdikleri gözlemlenmiştir. Hareketlerin belirginlik ve belirsizliği eklemlerin yaptığı açı miktarlarına göre değişmektedir. Eklemler daha büyük açılar yaptığında hareketler belirginleşmekte, daha küçük açılar yaptığında ise daha yumuşak bir görüntü vermektedir.

Sonuçlar değerlendirildiğinde Gerçek Zamanlı Yinelenen Öğrenme daha başarılıymış gibi görünse de bu yöntemin doğrultu sapması engellenecek şekilde tasarlanması gerekliliği fark edilmektedir. Ayrıca denetimli öğrenme problemlerinin hepsinde olduğu gibi bu yöntemde de eğitimin sağlanması için belirgin bir eğitim sinyaline ihtiyaç vardır. Bu eğitim sinyali üzerinde daha detaylı çalışma yapılması gerekmektedir.

Genetik Algoritma ile Politika Gradyanlı Destekli Öğrenme daha belirgin sonuçlar verdiklerinden gerçek dünyada kullanılmaları, karadaki robotlara uygulanmaları daha uygun görünmektedir. Bu iki öğrenme yöntemi birbirleri ile etkileşime girince verecekleri sonuçlar daha sonra incelenebilecek konular arasındadır. Bunun gerçekleştirilmesi amacı ile eğitim başında önceden belirlenmiş boyuttaki nüfus yaratılabilir ve üreme aşamasında yine turnuva seçimi kullanılmak şartıyla fark değerleri sabit sayılar olmak yerine genler arasındaki fark olarak alınabilir. Böylelikle Politika Gradyanlı Destekli Öğrenmenin başlangıçtaki tek bir harekete bağlılığı da azaltılmış olur.

KAYNAKLAR

Alpaydın, E., (2004), Introduction to Machine Learning, The MIT Press, Cambridge, Massachusetts.

Baeck, T., Fogel, D. B., ve Michalewicz, T., (2000), Evolutionary Computation 1, Basic Algorithms and Operators, Institute of Physics Publishing, Bristol, UK.

Blickle, T., (1995), "A Comparison of Selection Schemes used in Genetic Algorithms", 11.

Bremermann H. J., (1962), "Optimization Through Evolution and Recombination", Proceedings of the Conference on Self-Organizing Systems, Chicago, Illinois.

Chang, W.F. ve Mak, M.W., (1999), "A Conjugate Gradient Learning Algorithm for Recurrent Neural Networks", Neurocomputing, 24 (1-3):173-189.

Clark, A., (1997), Being There - Putting Brain, Body and World Together Again, The MIT Press, Cambridge, Massachusetts.

Deb, K., (1998), "An Introduction to Genetic Algorithms", International Workshop on Soft Computing and Intelligent Systems, 12 Ocak 1998, Calcutta, Hindistan.

Draye, J., Pavisic, D. ve Libert, G., (1996), "Dynamic Recurrent Neural Networks: A Dynamical Analysis", IEEE Trans. on Systems Man and Cybernetics, Part B, 26(5):692-706.

Farmer, J. D. ve Belin, A. D., (1991), Artificial Life: The Coming Evolution, In Artificial Life II, SFI Studies in the Sciences of Complexity, C.G. Langdon, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), Addison-Wesley.

Fogel, J., Owens J. ve Walsh, J., (1966), Arificial Intelligence through Simulated Evolution, John Wiley, New York

Frankham, R., (1996), "Relationship of Genetic Variation to Population Size in Wildlife", Conservation Biology, 10(6):1500-1508.

Friedberg R. M., (1958), "A Learning Machine Part 1", IBM J., 2:2-13.

Gahot, L.B., (2005), "Using Genetic Algorithms to Evolve Locomotion in Artificial Creatures", Imprimé à l'Ecole Nationale Supérieure des Télécommunications, Temmuz 2005, Paris, Fransa.

Gaskett, C., Wettergreen, D. ve Zelinsky, A., (1999), "Q-learning in Continuous State and Action Spaces", 12th Australian Joint Conference on Artificial Intelligence, Aralık 1999, Sydney, Avustralya.

Goldberg, D., (1989), Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Inc, Massachusetts.

Haykin, S., (1994), Neural networks, A Comprehensive foundation, Prentice-Hall, Inc. New Jersey.

Holland, J., (1962), "Outline for a Logical Theory of Adaptive Systems", J. ACM, 9:297-313.

Jacob, D., Polani, D. ve Nehaniv, C. L., (2005), "Faster Learning in Embodied Systems through Characteristic Attitudes", IEEE Computational Intelligence in Robotics & Automation (IEEE CIRA'05) special session on Ontogenetic Robotics, 27-30 Haziran 2005.

Klein, J., (2002), "BREVE: a 3d Environment for the Simulation of Decentralized Systems and Artificial Life", Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems. The MIT Press.

Kohl, N. ve Stone, P., (2004), "Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion", IEEE International Conference on Robotics and Automation (ICRA 2004), 3: 2619-2624.

Linkens, D. ve Nyongesa, Y., (1996), "Learning Systems in Intelligent Control: An Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications", IEE Proc. Control Theory App., 134 (4):367-385.

Meeden, L. A. (1996), "An Incremental Approach to Developing Intelligent Neural Network Controllers for Robots", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(3):474-485.

Mitchell, M. ve Forrest, S., (1994), "Genetic Algorithms and Artificial Life", Artificial Life, 1(3):267-289.

Mitchell, M., (1996), An Introduction to Genetic Algorithms. The MIT Press, Cambridge, Massachusetts.

Mitchell, T., (1997), Machine Learning, McGraw Hill, Boston.

Negnevitsky, M., (2002), Artificial Intelligence: A Guide to Intelligent Systems, Addison Wesley, İngiltere.

Nerrand, O., Roussel-Ragot, P., Urbani, D., Personnaz, L. ve Dreyfus, G., (1994), "Training Recurrent Neural Networks: Why and How ? An Illustration in Dynamical Process Modeling", IEEE Neural Networks, 5(2):178-183.

Nolfi, S. ve Floreano, D., (2000), Evolutionary robotics, The MIT Press, Cambridge, Massachusetts.

Pavlov, I. P., (1927), Conditioned Reflexes, Oxford Univ. Press, Oxford.

Pearlmutter, B., (1989), "Learning State Space Trajectories in Recurrent Neural Networks", Neural Computation, 1:263-269.

Peschl, M., (1996), "The Representational Relation Between Environmental Structures and Neural Systems: Autonomy and Environmental Dependency in Neural Knowledge Representation", Nonlinear Dynamics, Psychology and Life Sciences, 1(2): 99-121.

Pfeifer, R. ve Scheier, C., (1999), Understanding Intelligence, The MIT Press, Cambridge, Massachusetts.

Pineda, F., (1988), "Dynamics and architecture for neural computation", Journal of Complexity, 4:216-245.

Ray, T. S., (2001), Artificial Life, In Frontiers of Life, Volume One the Origins of Life, R. Dulbecco, D. Baltimore, F. Jacob and L. Montalcini (Eds.), Academic Press.

Reynolds, C. W., (1987), "Flocks, Herds, and Schools: A Distributed Behavioral Model", Computer Graphics, 21(4):25-34.

Rumelhart, D. ve McClelland, J., (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, Volume 1. The MIT Press, Cambridge,

Massachusetts.

Russell, S., (2001), Open Dynamics Engine v0.03 User Guide. http://www.ode.org/ode-latest-userguide.html

Schuitema, E., Hobbelen, D.G.E., Jonker, P.P. , Wisse, M. ve Karssen, J.G.D., (2005), "Using a Controller Based on Reinforcement Learning for a Passive Dynamic Walking Robot", IEEE International conference on Humanoid Robots, 5-7 Aralık 2005, Tsukuba, Japonya.

Sharkey, N. E., (1997), "Neural Networks for Coordination and Control: The Portability of Experiential Representations", Robotics and Autonomous Systems, 22: 345-359.

Sharkey, N. E., (1997), "The New Wave in Robot Learning", Robotics and Autonomous Systems, 22:179-186.

Siegwart, R. ve Nourbakhsh I. R., (2004), Introduction to Autonomous Mobile Robots, The MIT Press, Cambridge, Massachusetts.

Sims, K., (1994), "Evolving Virtual Creatures", Computer Graphics, Annual Conference Series, 28(4):15-22.

Skinner, B. F., (1953), Science and Human Behaviour, Macmillan, New York.

Streeter, T. E., (2005), Design and Implementation of General Purpose Reinforcement Learning Agents, Yüksek Lisans Tezi, Iowa State University.

Sutton, R., McAllester, D., Singh, S. ve Mansour, Y., (2000), "Policy Gradient Methods for Reinforcement Learning with Function Approximation", Advances in Neural Information Processing Systems", 12: 1057-1063.

Syed, O., (1995), Applying Genetic Algorithms to Recurrent Neural Networks for Learning Network Parameters and Architecture, Yüksek Lisans Tezi, Case Western Reserve University.

Taylor, T. ve Massey, C., (2001), "Recent Developments in the Evolution of Morphologies and Controllers for Physically Simulated Creatures", Artificial Life, 7(1):77–87.

Tedrake, R., Zhang, T. W. ve Seung, H. S., (2005), "Learning to Walk in 20 Minutes", Fourteenth Yale Workshop on Adaptive and Learning Systems, Yale University, New Haven.

Thorndike, E. L., (1898), "Animal Intelligence: An Experimental Study of the Associative Processes in Animals", Psychological Review (Monograph Supplement), 2(4): 1-109.

Uğurlu, E. S. ve Biricik, G., (2006), "Olasılık Temelli Hareket Stratejisi Kullanan Q-Öğrenme", Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU'06), 17-19 Nisan 2006, Antalya.

Wahde, M., (2003), An Introduction to Adaptive Algorithms and Intelligent Machines, Gothenburg.

Walter, G. W., (1950), "An Imitation of Life", Scientific American, 182(5):42-45.

Walter, G. W., (1953), The Living Brain, Norton, New York.

Williams, R. J. ve Zipser, D., (1989), "A Learning Algorithm for Continually Running Fully

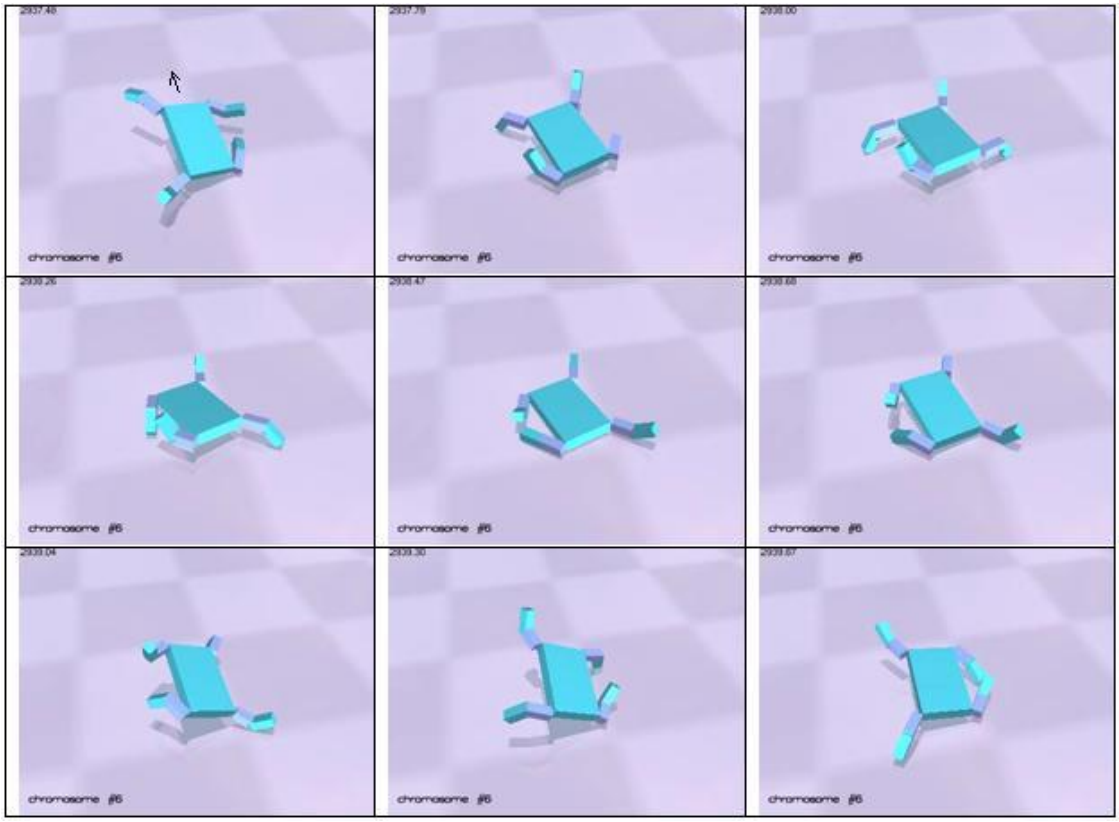
Recurrent Neural Networks", Neural Computation, 1:270-280.

Wolfram, S., (2002). A New Kind of Science. Wolfram Media, Inc, Champaign.

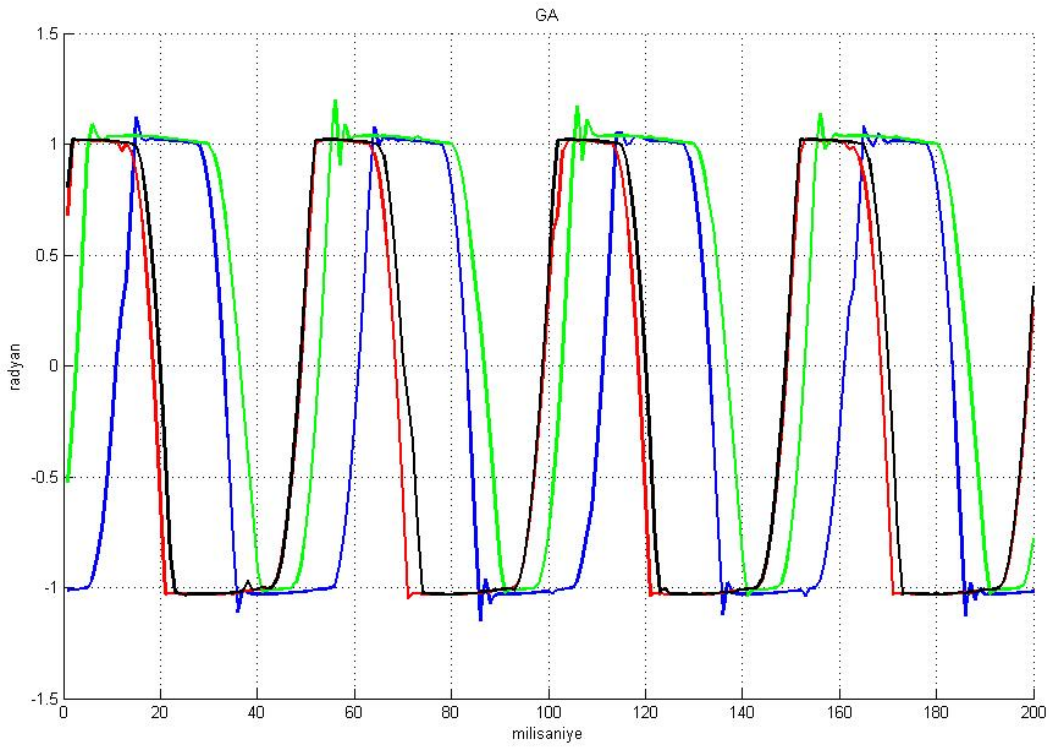
Ziemke, T., (1999), "Remembering How to Behave: Recurrent Neural Networks for Adaptive Robot Behavior," in Medsker and Jain Eds., Recurrent Neural Networks: Design and Applications, Boca Raton, CRC Press, 1999.

EKLER

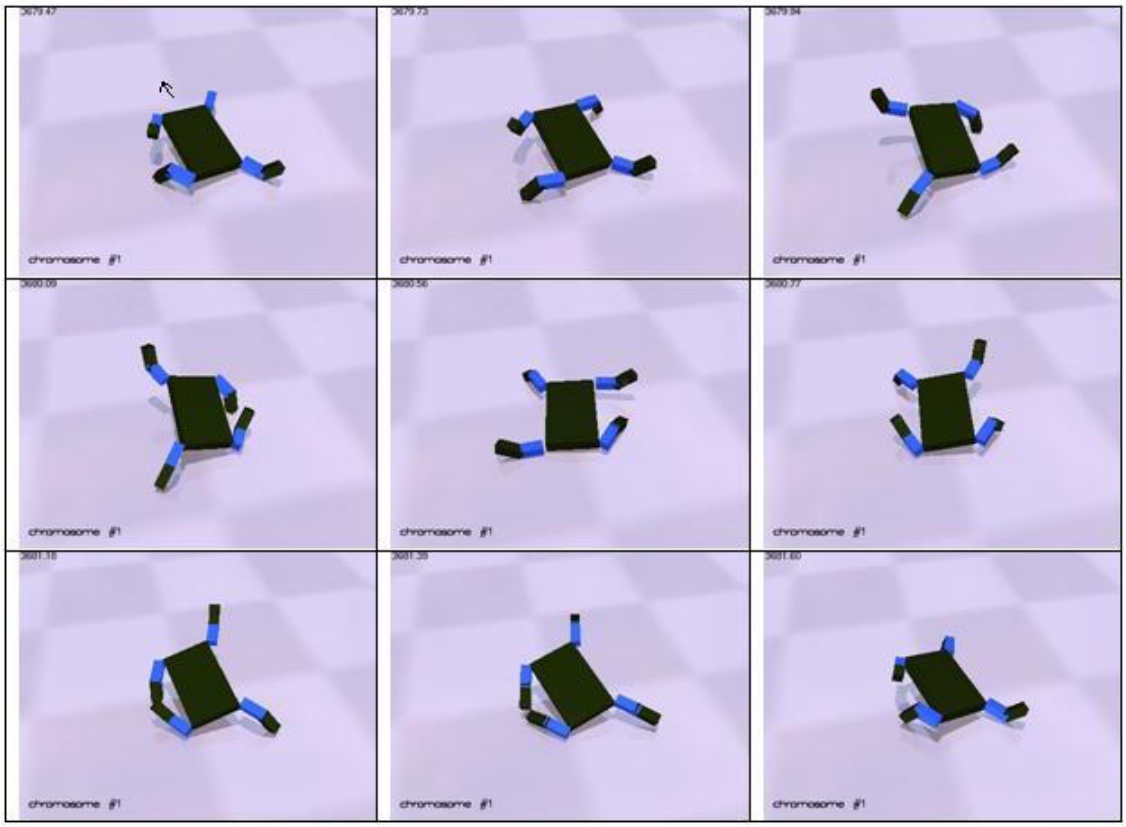
Ek 1 Genetik Algoritma sonuç örnekleri ve açı aralıkları



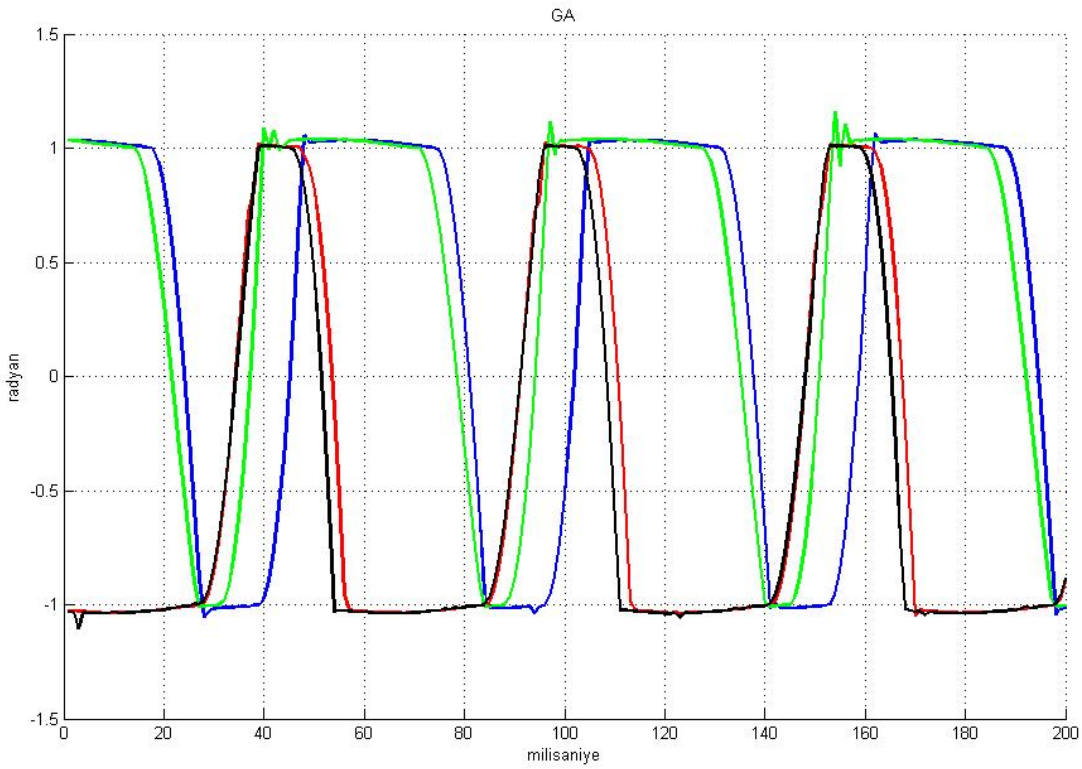
Şekil Ek 1.1 GA örnek 1, mesafe: 115,87



Şekil Ek 1.2 GA örnek 1 açı aralığı

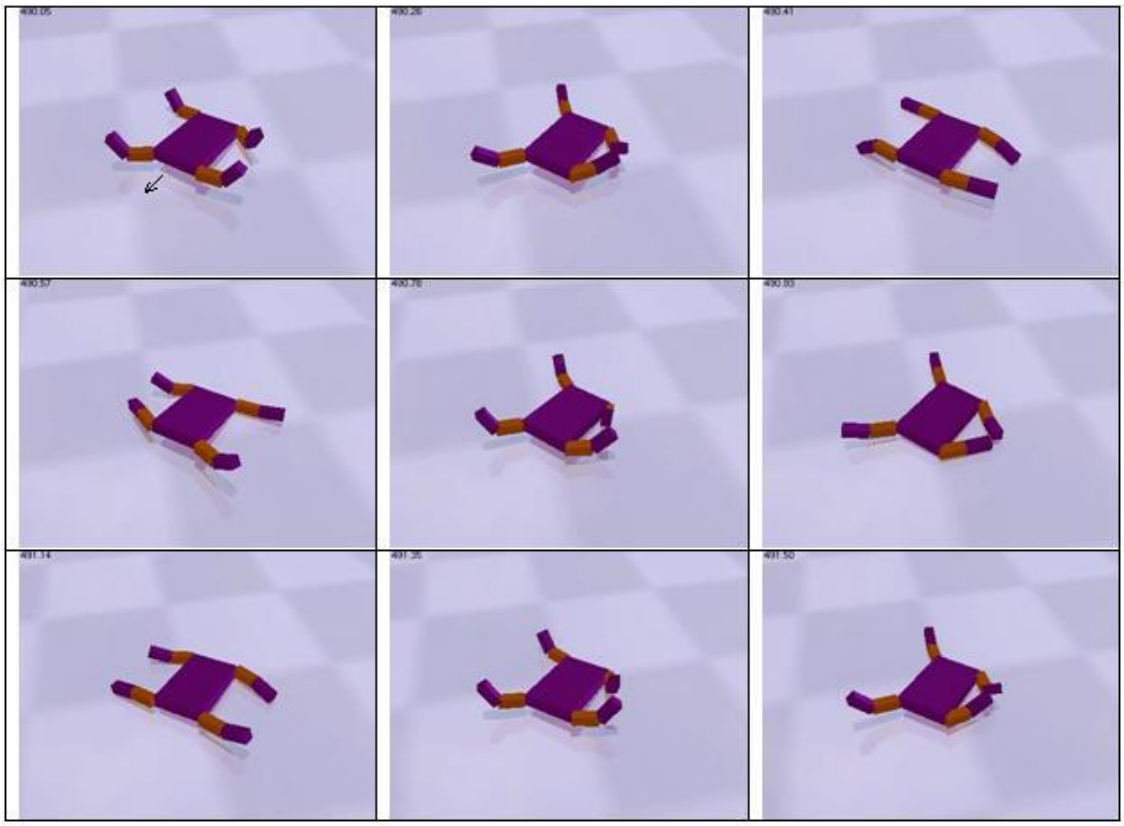


Şekil Ek 1.3 GA örnek 2, mesafe: 93,97

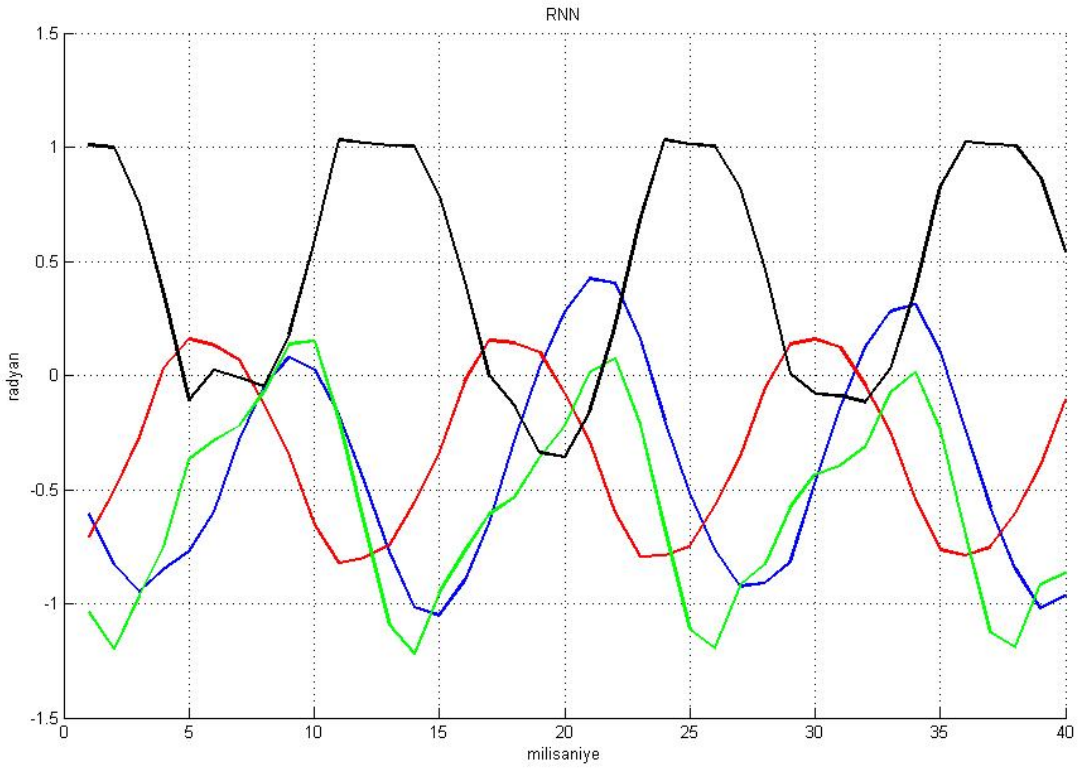


Şekil Ek 1.4 GA örnek 2 açı aralığı

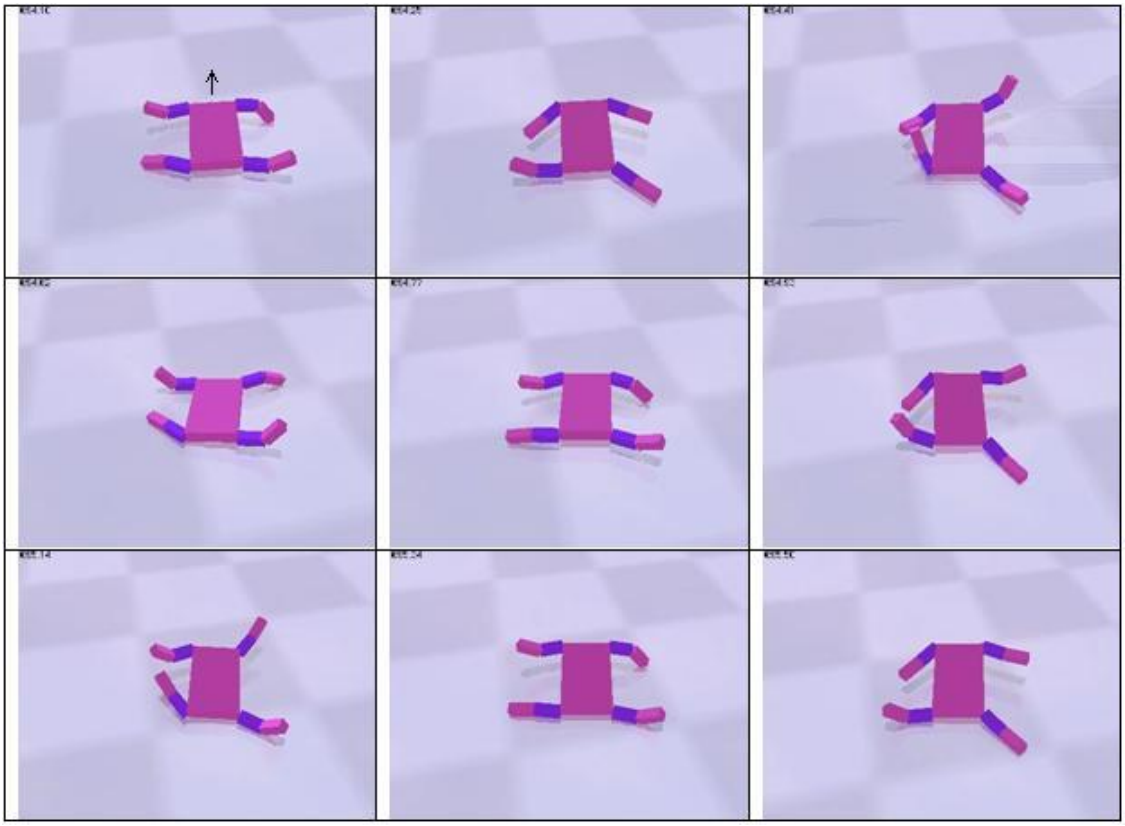
Ek 2 Gerçek Zamanlı Yinelenen Öğrenme sonuç örnekleri ve açı aralıkları



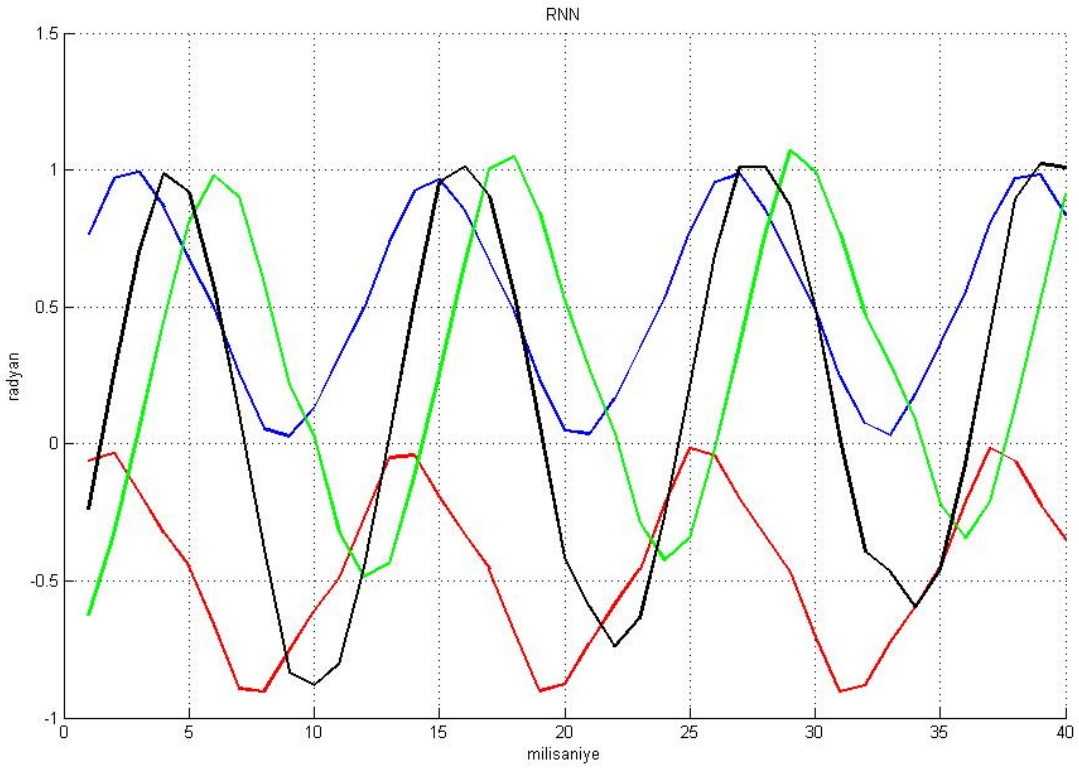
Şekil Ek 2.1 GZYÖ örnek 1, mesafe: 227,15



Şekil Ek 2.2 GZYÖ örnek 1 açı aralığı

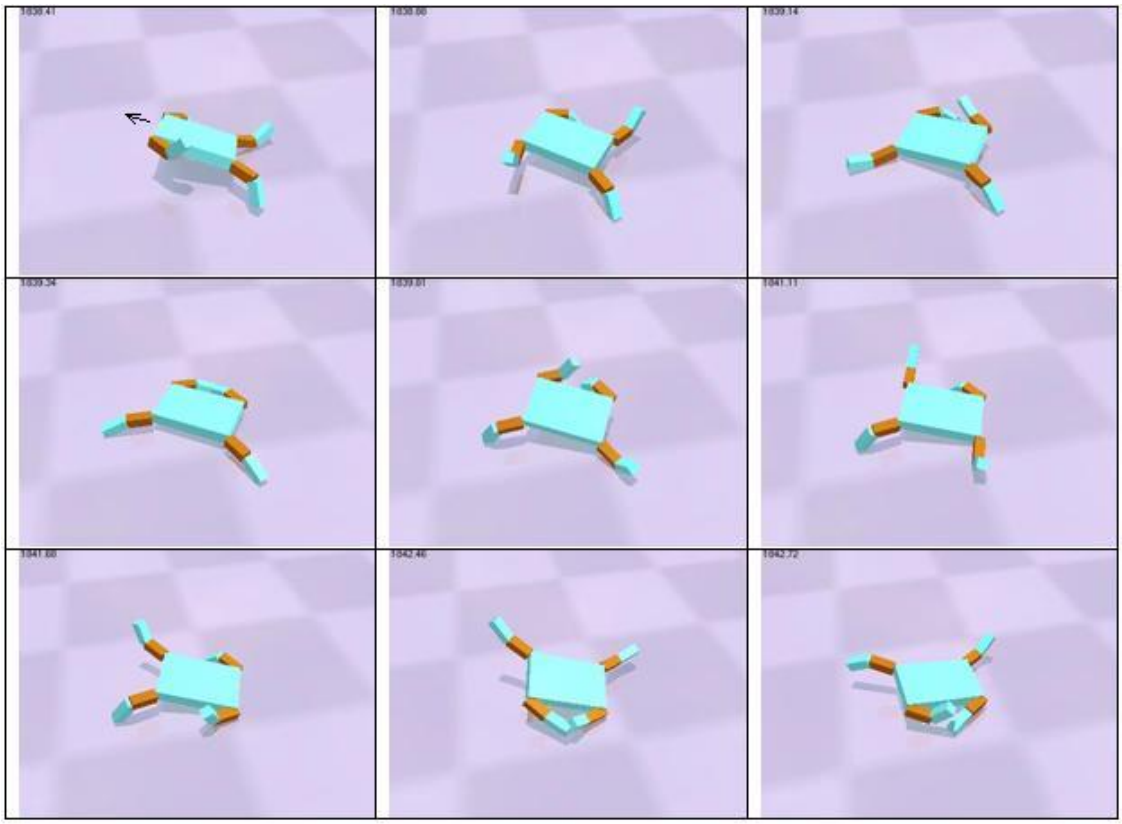


Şekil Ek 2.3 GZYÖ örnek 2, mesafe: 119,98

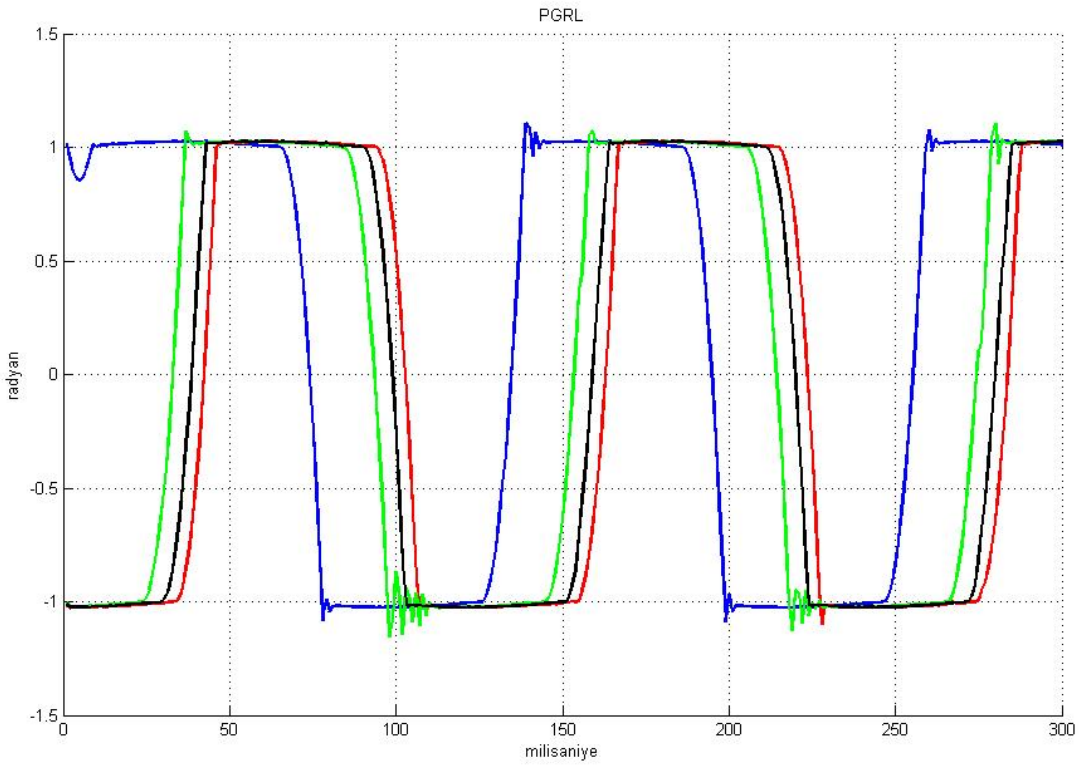


Şekil Ek 2.4 GZYÖ örnek 2 açı aralığı

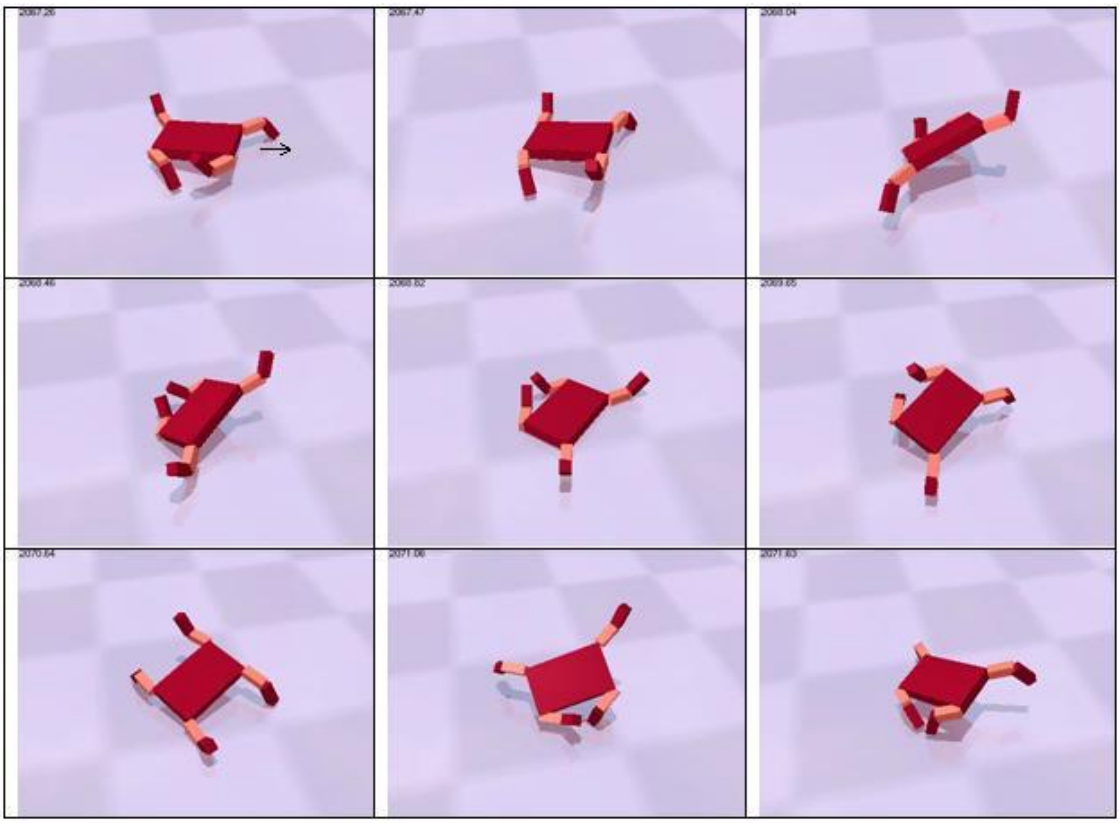
Ek 3 Politika Gradyanlı Destekli Öğrenme sonuç örnekleri ve açı aralıkları



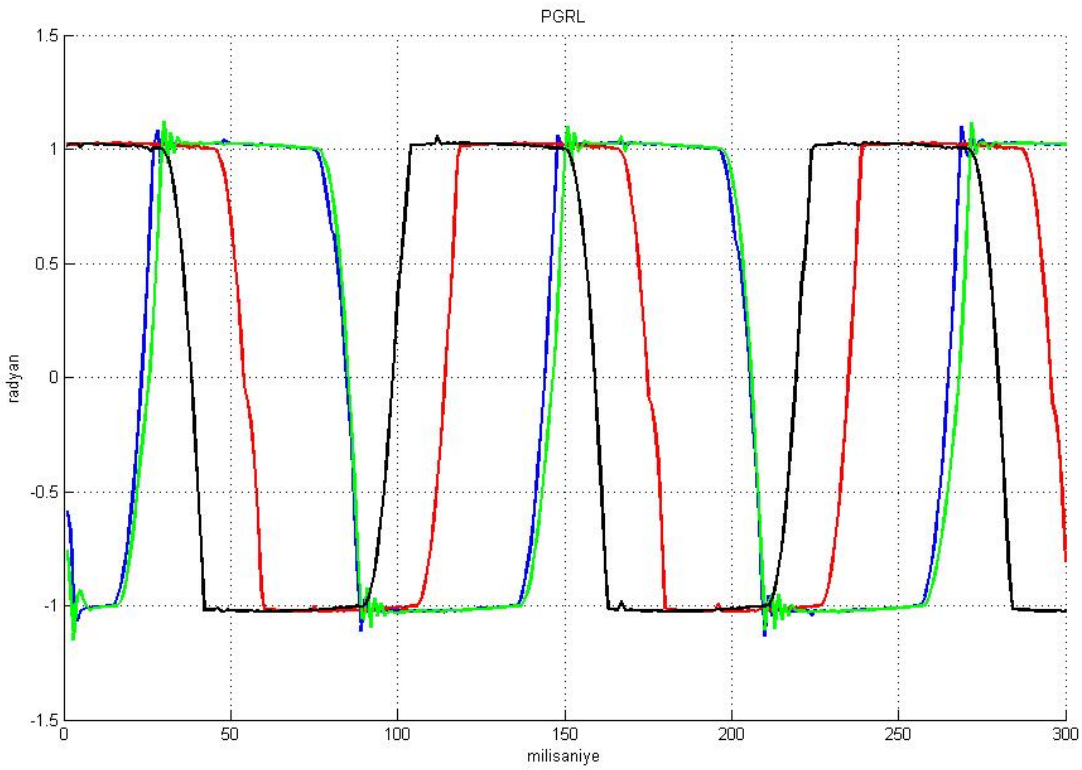
Şekil Ek 3.1 PGDÖ örnek 1, mesafe: 58,13



Şekil Ek 3.2 PGDÖ örnek 1 açı aralığı



Şekil Ek 3.3 PGDÖ örnek 2, mesafe: 58,42



Şekil Ek 3.4 PGDÖ örnek 2 açı aralığı

Ek 4 Türkçe – İngilizce sözlük

açgözlü hareket belirleme stratejisi	e-greedy strategy
algılayıcı	sensor
atış	pitch
bacaklı hareket	legged locomotion
basit refleksli etmen	simple reflex agent
Bayes Ağları	Bayesian Networks
bayır inişi	gradient decent
belirlenimci	deterministic
ceza fonksiyonu	cost function
çaprazlama	crossover
çok karakterli kodlama	many–character encoding
denetimli öğrenme	supervised learning
denetleyici	controller
Derecelendirme Stratejisi	Ranking Selection
destekli öğrenme	reinforcement learning
Dinamik Yayılma	Dynamic Propagation
doğal seçim	natural selection
durum bilgili refleksli etmen	reflex agent with state
eklemli	articulated
eşler arası	peer-to-peer
eşleşme havuzu	mating pool
etkinleştirme fonksiyonu	activation function
etmen	agent
evrim stratejileri	evolutional strategies
evrimsel öğrenme	evolutionary learning
evrimsel programlama	evolutioanry programming
eyleyici	actuator
fenotip	phenotype
geçici kredi devri	temporal credit assignment
Genetik Algoritma	Genetic Algorithm
genom	genome
genotip	genotype
Gerçek Zamanlı Yinelenen Öğrenme	Real Time Recurrent Learning
geri beslemeli kontrol	feedback control
Geri Yayılımlı Öğrenme	Backpropagation Learning
gösterim	representation
hareket	locomotion
hareket mekanizması	locomotion mechanism
hedef temelli etmen	goal-based agent
iç durum	internal state
İleri Beslemeli Sinir Ağları	Feed Forward Networks
indirim	discount
insansı robot	humanoid robot
Karar Ağacı Öğrenme	Decision Tree Learning
keşif	exploration
klasik koşullandırma	classical conditioning
kodlama	encoding
manevra yapma kabiliyeti	maneuverability
mantılılık	rationality

menteşeli eklem	hinge joint
öğrenme etmeni	learning agent
öz denetim	self-regulation
özerklik	autonomy
performans ölçüsü	performance measure
Politika Gradyanlı Destekli Öğrenme	Policy Gradient Reinforcement Learning
Rulet Tekerleği Seçimi	Roulette Wheel Selection
saklı düğüm	hidden node
sapma	bias
sapma	yaw
seçici baskı	selective pressure
Seçkinci Strateji	Elitist Strategy
seçme	selection
serbestlik derecesi	degree of freedom
sinaptik	synaptic
sömürü	exploitation
süperbilgisayar	supercomputer
tek noktalı çaprazlama	1-point crossover
tekerlekli hareket	wheeled locomotion
toplumsallık	sociability
Turnuva Seçimi	Tournament Selection
uyarlanabilirlik	adaptability
uygun bireyin kurtuluşu	survival of the fittest
uygunluk fonksiyonu	fitness function
üreme	reproduction
Veri Madenciliği	Data Mining
Yapay Sinir Ağları	Artificial Neural Networks
Yapay Sinir Hücresi Yakınsaması	Perceptron Convergence
Yapay Yaşam	Artificial Life
Yapay Zeka	Artificial Intelligence
yapı kararlı	structure determined
yapılanış	configuration
yarar fonksiyonu	utility function
yarar temelli etmen	utility based agent
yayılma	propagation
yerleşmişlik	situatedness
Yinelenen Geri Yayılma	Recursive Backpropagation
yönelme	orientation
yuvarlanma	roll
Zaman İçinde Geri Yayılım	Backpropagation Through Time
zihinsel bilim	cognitive science

ÖZGEÇMİŞ

Doğum tarihi	14.02.1982	
Doğum yeri	İstanbul	
Lise	1993 – 2000	Üsküdar Amerikan Lisesi
Lisans	2000 – 2004	Koç Üniversitesi Mühendislik Fak. Bilgisayar Mühendisliği Bölümü
Yüksek Lisans Enstitüsü	2004 –	Yıldız Teknik Üniversitesi Fen Bilimleri Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programı

Çalıştığı kurum(lar)

	2005 – Devam ediyor	YTÜ Elektrik Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü Bilgisayar Yazılımı Anabilim Dalında Araştırma Görevlisi