

YAPAY SİNİR AĞLARI VE ROBOT DENETİM UYGULAMALARI

Robot Denetimi

Robotlar ve uygulamaları makina mühendisliğinden elektrik mühendisliğine, bilgisayar mühendisliğinden iş idaresine kadar geniş teknik alanları kapsar. Bu makalede elektrik mühendisliğine en yakın konulardan biri olan ve YSA uygulamalarının yoğun olduğu robot denetimine eğileceğiz. Robot denetiminin ana hatları aşağıda verilmiştir. Daha ayrıntılı kuramsal bilgi için Craig'in kitabını [1], uygulamalı bilgi için Critchlow'un kitabını [2] öneririz.

ÖZET

Robot denetimi üç önemli aşamada gerçekleştirilir: görev planlaması (task planning), yol planlaması (path planning) ve eylem denetimi (motion control). Her aşamada robot Yapay Sinir Ağları (YSA) çeşitli amaçlarla kullanılmakta, yada kullanım yolu araştırılmaktadır. Bu makalede robot uygulamalarına açık YSA yöntemleri kısaca açıklanıp bilinen araştırma sonuçları özetlenmekte, geleceğe yönelik yorumlar getirilmektedir.

Robot denetimi çok sayıda motorun bir arada ve uyumlu olarak çalışmasını gerektirdiğinden dolayı oldukça karmaşık bir sorundur. Bu nedenle, robot denetim süreci genellikle üç önemli alt-soruna indirgenir: görev planlaması (task planning), yol planlaması (path planning) ve eylem denetimi (motion control). Bunların ilki, üst düzeyde planlama yöntemleri kullanarak robotun yapacağı işle ilgili bilgileri ve verileri basit eylemler zinciri haline getirir. İkincisi, plan sonuçlarını ve robotun çalıştığı ortamdaki engelleri gözönüne alarak robotun geçmesi gerekli noktaları, ve bu noktalardaki hız ve ivmeleri (acceleration) hesaplar. Eylem denetimi ise, robot motorlarının uygulayacağı doğrusal ve açısal kuvvetleri saptar. Bu işlemleri gerçekleştirecek denetim sistemi görev planlamasını her görev için bir kez yapar. Yol planlaması, eylem zincirinin her aşamasında tekrarlanır. Eylem denetimi robotun yaptığı işe bağlı olarak saniyede en az 200 kez motorlara komut göndermek zorundadır.

Robot denetiminin her aşamasında, o aşamaya uygun robot modeli ve ters modeli kullanılır. Robotun motorlarının konumundan (yada eklem konumundan) robot elinin konumunu saptayan denklemler sistemine kinematik

Can IŞIK

Syracuse Üniversitesi Elektrik ve Bilgisayar Mühendisliği Bölümü

model, ek konumundan eklem konumunu saptayan denklemlere ise ters kinematik model denir. Örneğin, eğer robottan istenen iş 3 boyutlu uzayda bir eğriyi belirli bir hızla izlemeyi gerektirir ise, bu eğrinin koordinatlarını robot eklemlerinin koordinatlarına, dolayısıyla motorların koordinatlarına çevirmek gerekir. Robot eline belirli bir hızın yada ivmenin verilmesi için gerekli motor hız ve ivmelerinin hesaplanması da kinematik modellerle gerçekleştirilir. Endüstriyel robotlarda bilgisayara düşen yükü azaltmak amacı ile, kinematik ve ters kinematik hesaplar her denetim dönemi (control period) yerine çok daha seyrek, örneğin 100 dönemde bir yapılmakta, bu hesaplar arasındaki değerler içkestirim (interpolation) yöntemleri ile yaklaşık olarak bulunmaktadır.

Robot eylemiyle ilgili kuvvetleri içeren denklemler sistem dinamiğini göz önüne alır. Motorların uyguladığı kuvvetlerin neden olduğu robot hız ve ivmeleri bulan denklemler sisteme dinamik model, istenen hız ve ivmeleri sağlayacak kuvvetleri bulan denklemlere ise ters dinamik model denir. Kuramsal modellerin iki tür sakıncası vardır. Birincisi, bu modeller basitleştirici varsayımlar içerir. Örneğin, robotların yük altında esnemediği, sürtünme kuvvetlerinin değişmediği (hatta sürtünme olmadığı), robot yükünün değişmediği olağan varsayımlar arasındadır. İkincisi, her ne denli basitleştirilse bile bu tür modelleri içeren denetim yöntemleri doğrusal olmadığından, etkinliklerini, kararlılıklarını saptaması güç ve karmaşıklıkları bir çok bilgisayarlı denetim sisteminin yeteneğinin üstündedir. Modele dayalı ileri besleme (feedforward) yöntemleri her ne kadar bu sorunların bir kısmına çözüm getirsedeyse, kullanılmayan modelin uyarlamalı (adaptive) olmaması bir sakınca olarak görülmektedir.

Yapay sinir ağlarının robot denetimine getirdiği katkılar şöyle özetlenebilir. YSA modelleri, sistemin fiziksel yapısından bağımsız olarak, genel amaçlı çok parametrelili, uyarlamalı bir şekilde geliştirilir. Sistemin fiziksel parametrelerinin önceden bilinmesi gerekmediği gibi, bu parametrelerin sabit kalması da varsayılmaz. Değişen yükler, sürtünme kuvvetleri, dişli boşlukları, esneyen mekanik parçalar kısa sürede YSA uyumlaması ile göz önüne alınır. Çok karmaşık fiziksel modeller yerine basit işlemlerin çok kez tekrarlanmasını gerektiren YSA modelleri bilgisayar gereksinimlerini azaltır. Çok yüksek boyutlu problemler bile basit paralel bilgisayar elemanları yada özel amaçlı (ama ucuz) YSA donanımları ile yeterince hızlı olarak çözülebilir.

Yapay Sinir Ağları

Her ne kadar bu sayıdaki diğer makaleler YSA'nın temel ilkelerini kapsamış olsada, makalenin bütünlüğü açısından kısa bir özet

sunuyoruz. YSA modelleri ile daha ayrıntılı bilgi ve algoritma isteyen okurlara Lippman'ın makalesini [3], YSA'ni dinamik sistemler açısından kuramsal olarak incelemek isteyenlere ise Kosko'nun kitabını [4] öneririz.

YSA modellerinin ana amacı basit işlem birimlerini yoğun bir şekilde birbirine bağlayarak yüksek verim elde etmektir. İşlem birimleri genellikle doğrusal olmayan ve oldukça ilkel işlemleri donanım yada yazılımda

gerçekleştirilebilir. Kaynaklarda en çok rastlanan birim modeli, önce girdilerinin ağırlıklı toplamını bulur, sonra toplamı S şeklinde bir işlevden geçirir.

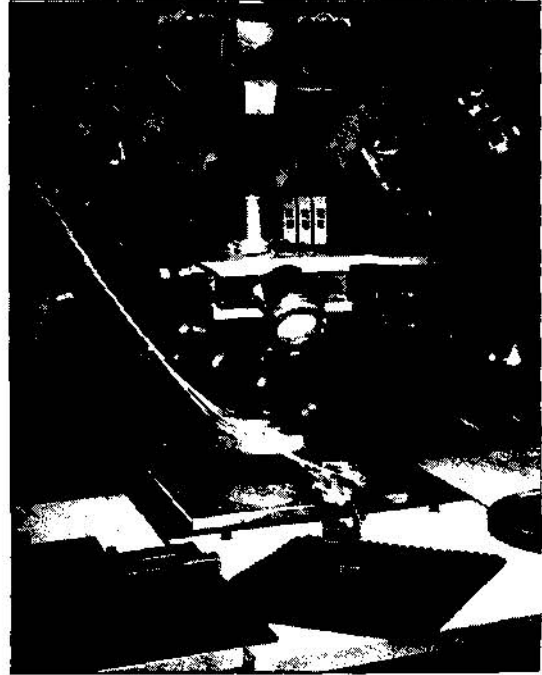
$$y = f \left(\sum_{i=1}^n a_i x_i \right)$$

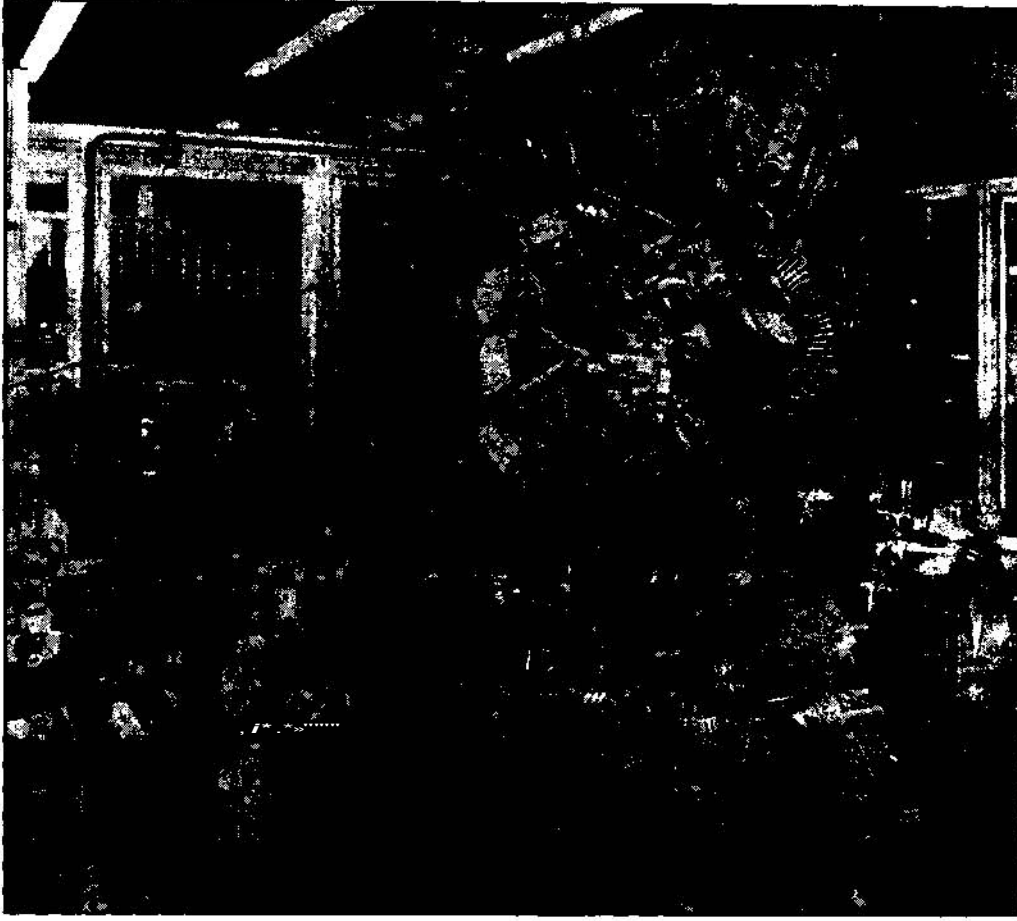
Burada, y biriminin çıktısı, x_i ve a_i $i=1, \dots, n$ biriminin girdileri ve ağırlıkları, $f(s)$ ise uyarı işlevidir. Uyarı işlevlerinin ortak özelliği s'nin çok büyük ve çok küçük değerlerinde doyuma uğramasıdır. Çoğunlukla, aşağıda denklemleri verilen, sigmoid yada hiperbolik tanjant işlevleri kullanılmaktadır.

$$\cdot \sum_{i=1}^n g_{\text{moid}(b)} = \frac{1}{1 + e^{-s}} \quad / \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$$

Her iki işlev de sıfır yakınındaki girdilere doğrusala yakın davranır. Girdilerin ağırlıklı toplamı (s) büyüdükçe çıktı üst sınırdan doyuma ulaşır. ($\lim_{s \rightarrow \infty} f(s) = 1$), küçüldükçe alt sınırdan doyuma ulaşır

$$(\lim_{s \rightarrow -\infty} \text{sig}(s) = 0, \lim_{s \rightarrow -\infty} \text{tanh}(s) = -1).$$





YSA'nın işleyişi birimler arasındaki bağlantıların yapısına dayanır. Robot denetiminde en yaygın olan yapıda birimler katlara bölünür ve bir kattaki birimlerin girdileri bir önceki kattan gelir ve çıktıları bir sonraki kata iletilir. Aynı kattaki birimler arasında bağlantı yoktur. Bu tür YSA'lara ileri beslemeli ağ (feed forward net) denir, ilk kata girdi katı, son kata ise çıktı katı, aradakilere de gizli kat denir. Bağlantı ağırlıkları başlangıçta rastgele olarak seçilir ve YSA'nın her işleyişinde net çıktısını beklenen çıktıya yaklaştıracak şekilde değiştirilir. Böylece, YSA bağlı olduğu sistemin girdi/çıktı bağlantısını ardışık (iterative) bir şekilde öğrenir. Bu öğrenme algoritmasına geri yayılma (back propagation) adı verilir.

Robot denetiminde kullanılan diğer bir YSA türü yinelgen (re-

cursive) ağ adını alır. İleri beslemeli ağlarda olduğu gibi, yinelgen ağlarda da birimler katlara ayrılmış ve aynı kattaki birimler birbirine bağlanmamıştır, ilkinden farklı olarak, bilgi girdi katından çıktı katına doğru akmasının yanısıra, ters yönde de akar. İleri beslemede sinyal belleği olmaz, ama yinelgen ağların sinyal belleği vardır ve dinamik sistem olarak incelenmeleri gerekir. Öğrenme algoritması olarak geri yayılmanın birkaç türevi kullanılmaktaysa da, bu alan henüz temel araştırma konusudur.

Kinematik Uygulamalar

YSA'nın ilk robot uygulamalarından biri Albus'un kullandığı 3 katlı ağıdır [5]. Bu yaklaşımın temel ilkesi, denetim komutlarını analitik olarak hesaplamak yerine çizelgelerden okumaktır. Çizelgelerin kapsamlarının etkin olduğu böl-

geler üstüste çakıştırılmış ve denetim komutu bu çakıştan bellek kapsamlarının ağırlıklı toplamı olarak bulunmuştur.

Geri yayılma yöntemini kullanan uygulamalar çoğunluktadır. Bu yöntemde bilinen analitik kinematik denklemler YSA modelini geliştirmek için örnek çıkarmakta kullanılır. Rastgele seçilen robot eklemlerinin konumları için robot elinin konumları hesaplanır. YSA girdileri olarak robot elinin konumunu veren değişkenlerin değerleri kullanılır. Örne-

ğin, 2 boyutlu uzayda (düzlemde) çalışan bir robotun el konumu, bu elin koordinatları (x ve y değişkenleri) ve yönü (x eksenine yaptığı açı) ile saptanır. YSA çıktısı ise robot elini o konuma getirecek eklem açılarının yaklaşımıdır. Bu yaklaşım, analitik olarak bulunan eklem açıları ile karşılaştırılır, YSA katsayıları aradaki farkı azaltacak şekilde değiştirilir. Başlangıçta, YSA bağlantı katsayıları rastgele seçildiğinden, büyük yanılğı (error) beklenir. Yavaş yavaş bu yanılğı azaltılır ve YSA modeli ters kinematik modele yakınsanır (converges). Psaltis [6], Guez ve Ahmad [7] 2 ve 3 eklemlerli robotların ters kinematik modellerini ileri beslemeli YSA ile elde etmişler ve modelin yakınsanlık alanının (region of convergence) oldukça geniş olduğunu göstermişlerdir. Bu yaklaşımın iki tür sakıncası vardır.



İlki, temel bir sınırlamadır. Bu yöntem YSA katsayılarını devre dışı (off-line) hesaplandığından, robotun çalışması sırasında değişen koşullara uyum sağlayamaz. İkincisi, genel amaçlı bir ters kinematik YSA'nın karmaşıklığını arttıracak bir sınırlamadır. Bu yaklaşımın uygulandığı robotun eklem sayısı için üzerine çıktığında, kinematik sistemin karmaşıklığı YSA modelinin yakınsanlık alanını küçültmektedir. Bu nedenle, örneğin 6 eklemli bir robotun tüm çalışma alanını kapsamak için çok sayıda (en az 8) YSA modeli gerekebilir. Bu sakıncayı ortadan kaldıracak bir YSA çifti Yeung'un makalesinde [8] incelenmiştir. Ortalama duyarlı ağ (context sensitive net) denen bu yapıda bir YSA robotun içinde bulunduğu bölgeye göre ortam parametreleri üretir, ikinci bir YSA ise, bağlantı katsayılarını ortam parametrelerine bağlı olarak değiştirir. Olağan bir YSA ile karşılaştırıldığında, Yeung'un yönteminin hem daha hızlı dengeye geldiği, hemde analitik modele daha yakın olduğu gözlenmiştir.

Kinematik sistemle ilgili başka bir araştırma, robot hareketini TV kamerası ile yapılan ölçümlere bağlamıştır. Kuperstein [9] başlangıçta robot motorlarını rastgele bir şekilde hareket ettirir. Ölçümler robot elinin konumunu istenen konumla karşılaştırıp, aradaki ya-

nılığının (error) büyüklüğüne göre bağlantı katsayılarını değiştirir. Yalnızca 2 boyutlu uzayda çalışan robotlara uygulanmasına karşın, bu yöntemin önemli üstünlükleri vardır. Öğrenme örneği olarak analitik model yerine kamera ölçümleri kullanıldığından uyarlamalı model verir; YSA modeli yalnızca robotun değil, kamera sisteminde analitik denklemlerinin yerini alır.

Bu alandaki son örnek biraz değişik: Salatian ve Zheng'in makalesinde [10] açıklanan araştırma iki ayaklı bir robotun yürüdüğü arazinin açısına göre gövde açısını ve adım konumlarını saptayan bir YSA üzerine. Robottaki duyaçların (sensors) yapısına uygun olarak birbirine bağlanan işlem birimleri, bağlantı katsayılarını duyaçlardan gelen uyarıların olumlu yada olumsuz oluşuna göre değiştirirler. Simulasyon örneklerinde 40 adım içinde robotun dengeli olarak yürüdüğü gözlenmiştir. Bazı duyaçların kuvvet ölçmesine karşın, modelin çıktısı konum olduğundan bu örnek kinematik sistemler bölümünde verilmiştir. Çok ayaklı robotların adım sıralarının YSA ile denetimi ile ilgili ilginç bir yöntem Lewis ve diğerleri [11] tarafından önerilmiştir. Genetik programlama adı verilen bu yöntem, denetim sinyallerinin rastgele bir şekilde (mutasyon gibi) değiştirilmesine ve değişikliklerden iyi sonuç vereninin seçilmesine (doğal seleksiyon gibi) dayalı. Bu yöntem

robot denetimi gibi çok boyutlu problemler için aşırı sayıda kombinasyonlara yol açabildiğinden kolay uygulanır değil. Makale yalnızca başarı kriterlerini karşılaştıran, kuramsal bir yazı, ancak konuda araştırma yapmak isteyenlere yararlı olabilir.

Dinamik Uygulamalar

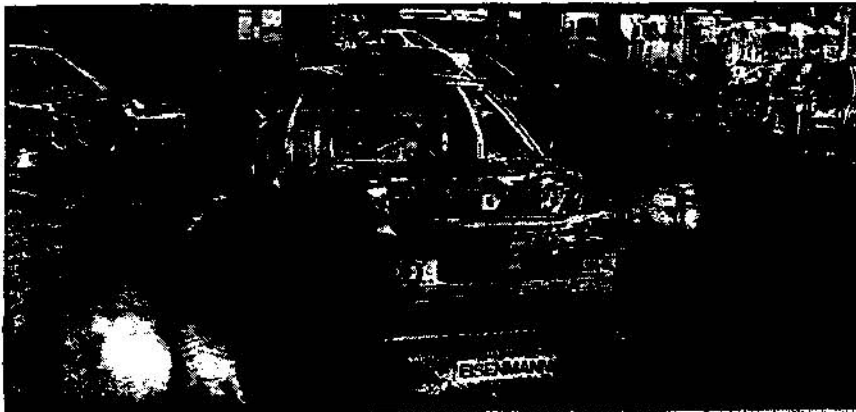
Dinamik robot denetiminin amacı, robotun belli konumu, hızı ve iğmesi için gerekli kuvveti ve dolaşısıyla motorlara uygulanacak denetim sinyalini hesaplamak, geri besleme ile yalnızca modele katılmayan etkenlerin yol açacağı yanılığları düzeltmektir. Geri beslemeye olan yük azalınca denetim sisteminin tepki hızı artar, daha kararlı olur ve yanılığ enerjisi azalır. Dinamik hesaplamaları eksiksiz olarak analitik bir modele dayanarak yapmak olanak dışıdır. YSA bu hesapları uyarlamalı olarak yapmak için kullanılmış ve başarılı sonuçlar elde edilmiştir. Robot dinamik denklemleri vektörel olarak aşağıda verilmiştir

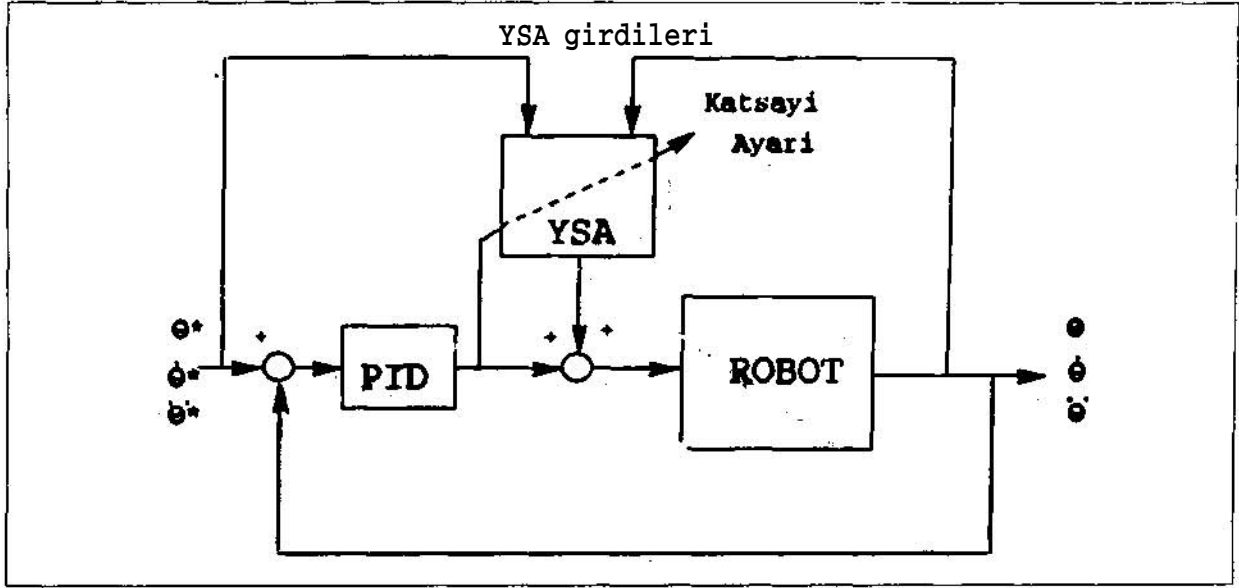
$M-$

$$\tau = M(\theta) \ddot{\theta} + \tau_h(\theta, \dot{\theta}) + \tau_v(\theta) + \tau_g(\theta, \dot{\theta})$$

Bu denklemde T genel kuvvet (doğrusal ve açısal) vektörü, M kütle matrisi, 9 robot eklemlerinin konum vektörü (doğrusal ve açısal), x_n hızla ilgili merkezkaç ve Koriolis kuvvetleri vektörü, t_y yer-

çekim kuvveti vektörü, T» sürütünme kuvveti vektörüdür. Kuramsal olarak robotun konumu, hızı ve ivmesi verilirse denklemdeki her terim hesaplanabilir. YSA na girdi olarak robotun konumu, hızı ve ivmesi verilir, YSA çıktısı gerekli kuvvetin kestirimi olarak kabul edilirse istenen uyarlamalı model elde edilmiş olur. Bulunan sonuca geri besleme denetleyicisinden gelen sinyal eklenirse, ters model adı veri-





Şekil 1: YSA ile dinamik robot denetimi

len bir yaklaşım elde edilmiş olur. Böylece geliştirilen bir robot denetleyicisinin çizimi Şekil 1 de verilmiştir [12].

Bu yaklaşımın ilginç tarafı, YSA'nın bağlantı katsayılarını ayarlamak için geri beslemedeki yanlış sinyalinin kullanılmasıdır. (Kuvvet kestirimi yanlışlığı geri besleme yanlışlığına tekdüze artan bir şekilde bağlıdır). Bu denetim yapısı K. Cılız'ca simüle edilmiş ve konum yanlışlığı yalnızca geri beslemenin kullanıldığı denetleyicidekinin (PID) yüzde biri dolayında bulunmuştur [13]. Bu tür denetleyicilerin kararlılıkları [14] de incelenmiştir.



Psaltis, [6] ters model'i geliştirmek için robotu

önce basit bir denetim ile çalıştırırken denetimde henüz yer almayan bir YSA na girdi olarak robotun konum ve türevlerini uygulayıp YSA çıktısı ile motorlara uygulanan sinyallerin farkı ile katsayıları ayarlamış, YSA modeli dengeye eriştikten sonra YSA çıktısını denetimde uyuramasız olarak kullanmıştır. Kawato, [15] ters modelin ayrıntılı denklemin-

deki her katsayıyı YSA daki "bir bağlantı katsayısına denk getirmiş ve sorunu uyarlamalı denetime (adaptive control) indirgemıştır. Bu yaklaşımın sakıncası, analitik modelde yer almayan etkilerin YSA tarafından modellenememesidir.

Robot dinamik denklemindeki tüm terimleri bir tek YSA ile modellemek yerine her terime ayrı bir YSA ayırmayı deneyenler de olmuştur, işlevsel ayırım denen bu yöntemle Bakey'in [16] yaklaşımı örnek olarak verilebilir. Böyle bir modelde ağların bir kısmı uyarlı, bir kısmı değişmez olarak tutularak bilgisayar yükü azaltılabilir. Aynı kaynağı bu konuda daha ayrıntılı bilgi arayanlara da öneririz.

YSA ile Denetimin Sınırlamaları

YSA'nın yararları yanısıra bazı uygulama güçlükleri ve sınırlamaları vardır. YSA ile her tür Borel Ölçülü (measurable) işlevin yakınsanabileceğini kanıtlayan teoremler kanıtlanmış olsada, belli bir uygulama için kaç katlı ağ gerektiği, her katta kaç birim olacağı, birimlerin çeşitli parametrelerinin değerleri ile ilgili kuram

henüz geliştirilmemiştir. Bağlantı katsayıları dışındaki parametreler sınıma yanılmayla ve deneyimle bulunmakta ve genel tasarım kuralları yavaş yavaş geliştirilmektedir. Ayrıca, YSA yi içeren sistemlerin kararlılığını somut bir şekilde analiz eden yöntemlerin hızla geliştirilmesi gereklidir. Bu güne dek, Lyapunov işlevleri kullanılarak soyut kararlılık çalışmaları yapılmıştır [14].

Sonuçlar

YSA'nın şu ana dek yavaş gelişmesinin en büyük nedeni, özellikle örüntü tanıma (pattern recognition) gibi yüksek eboyutlu sorunlarda, öğrenmenin çok yavaş olması ve bir hesaplama döngüsünün uzun zaman almasıdır. Ancak, robot yada daha genel amaçlı denetim uygulamalarında ağa girdi sayısı binlerce değil 10-20 arasındadır. Girdi çıktı eşleri birbirine fiziksel olarak bağlı olduğundan öğrenilecek işlev az pürüzlü bir yüzeydir. Bu nedenlerle, az sayıda işlem birimleriyle oluşturulan ve hızlı çalışan YSA denetleyicileri yavaş yavaş laboratuarlardan fabrikalara yayılmaktadır. Denetleme yapıları ile birlikte YSA donanımlarında gelişme göstermektedir. Özel ve genel amaçlı YSA yon-

galan ile yazılımdaki hızın 200 katına çıkmış (saniyede 2 milyar bağlantı hesabı) Intel 64 elemanlı, 8K bağlantılı bir yongayı piyasaya çıkarmıştır. Bu tür yongaların yardımı ile 6 eklemli genel amaçlı robotların YSA ile denetimi olanak içine gelmiştir.

Robot denetimi, YSA'nın karmaşık, doğrusal olmayan ve değişken katsayılı sistemlere uygulanabilirliğini gösteren bir örnek olarak görülmeli, sistem modellemesinin güç ama yararlı olacağı alanlara genelleştirilmelidir. Yazarın kanısınca YSA robot denetiminde ve genel olarak endüstride hızla yayılacaktır. Bunun ana nedenleri, çalışmakta olan sistemleri ortadan kaldırmadan, ek bir ileri besleme altsistemi olarak kullanılabilmesi, kuramsal model bilinmese bile sistem ters modelini çıkarması ve standart bir yapının hemen hemen her denetim sorununa uygulanabilmesidir.

Kaynaklar

- [1] J.J. Craig, *Introduction to Robotics*, 2nd ed., Addison Wesley, 1990.
- [2] A.J. Critchlow, *Introduction to Robotics*, Macmillan Publishing, 1985.
- [3] R.P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP magazine*, April 1987.
- [4] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, 1992.
- [5] J. Albus, "A new approach to manipulator control: the cerebellar model articulation controller," *Journal of Dynamic Systems, Measurement and Control*, vol. 97, 1975.
- [6] D. Psaltis, A. Sideris, A. Yamamura, "Neural controllers," *Proc. IEEE Int. Conference on Neural Networks*, 1987.

- [7] A. Guez, Z. Ahmad, "Solution to the inverse problem in robotics by neural networks," *Proc. Int. Conference on Neural Networks*, 1988.
- [8] D.T. Yeung, G.A. Bekey, "Using a context sensitive learning network for robot arm control," *Proc. IEEE Int. Conference on Robotics and Automation*, 1989.
- [9] M. Kuperstein, "Adaptive visual-motor coordination in multijoint robots using parallel architecture," *Proc. IEEE Int. Conference on Robotics and Automation*, 1987.
- [10] A.W. Salatan, Y.F. Zheng, "Gait synthesis for a biped robot climbing sloped surfaces using neural networks," parts I & II, *Proc. IEEE Int. Conference on Robotics and Automation*, 1992.
- [11] A.M. Lewis, A.H. Fagg, A. Solidum, "Genetic programming approach to the construction of a neural network for control of a walking robot," *Proc. IEEE Int. Conference on Robotics and Automation*, 1992.
- [12] K. Ciliz, C. Işık, "Trajectory learning control of robotic manipulators using neural networks," *Proc. IEEE Int. Symposium on Intelligent Control*, 1990.
- [13] K. Ciliz, *Artificial Neural Network Based Control of Nonlinear Systems with Application to Robotic Manipulators*, Ph.D. Dissertation, Syracuse University, September 1990.
- [14] K. Ciliz, C. Işık, "Stability and convergence of neurologic model-based robotic controllers," *Proc. IEEE Int. Conference on Robotics and Automation*, 1992.
- [15] M. Kawato, Y. Uno, M. Isobe, R.A. Suzuki, "Hierarchical model for voluntary movement and its application to robotics," *Proc. IEEE Conference on Neural Networks*, 1987.
- [16] G.A. Bekey, "Robotics and neural networks," in *Neural Networks for Signal Processing*, B. Kosko, Prentice Hall, 1992.



Meral
ÜNVER



İlkay
CEBİ
(20586)



M. Ali
ERKUT
(2008)

AİLELERİNE, YAKINLARINA, DOSTLARINA
VE TÜM ODAMIZ CAMİASINA
BAŞ SAĞLIĞI DİLERİZ