

Mobil robotlarda diferansiyel evrim ile optimal yörünge planlama

Serkan AYDIN*, **Hakan TEMELTAŞ**

İTÜ Elektrik-Elektronik Fakültesi, Elektrik Mühendisliği Bölümü, 34469, Ayazağa, İstanbul

Özet

Bu yayında önerilen yöntemle, mobil robotlarda, çevreden gelen kısıtlamalar (engeller) ve robotun hız ve ivme gibi fiziksel kısıtlamaları dikkate alınarak, önceden görünebilirlik grafik metoduyla (VGM) planlanmış yol üzerinde zaman-optimal yörünge bulunmuştur. Yörüngeler, düz kısımlardan (dönme ivmesi $a_s=0$ ve dönme hızı $v_s=0$) ve eğri kısımlardan (öteleme ivmesi $a_t=0$ ve öteleme hızı $v_t=$ sabit) oluşturulmuşlardır.

Eğri kısımları oluşturmak için sadece θ (dönüş açısı) ve v_t (eğri kısım öteleme hızı) parametrelerinin bilinmesi yeterli olmaktadır. Olası tüm eğri kısımlar kümesi ($\theta \in (0, \pi]^\circ$ ve $v_t [0, 40]$ inç) içerisinde evrimsel bir global optimizasyon metodu olan diferansiyel evrim (DE) ile, engellerle çakışmayan zaman-optimal amaç ölçütünü sağlayan yörünge seçilmesi gerçekleştirilmiştir.

Anahtar Kelimeler: Mobil robot, optimizasyon, yörünge planlama, diferansiyel evrim.

Differential evolution method for mobile robot optimal trajectory planning

Abstract

A method is proposed to find the time optimal trajectory on predefined path of mobile robots, which is found by using visibility graph method. Constraints from environment (obstacles) and physical constraints (i.e. steering and translational accelerations/velocities) are taken into consideration in the method. The planned trajectories are composed of line segments (steering acceleration/deceleration $a_s=0$ and velocity $v_s=0$) and curve segments (translational acceleration/deceleration $a_t=0$ and velocity $v_t= v_t()$). The structures of the curves are determined by only two parameters: θ , turn angles, these are produced by visibility graph method and v_t , translational velocities on the curve segments, these are the elements of the parameter vector. A curves set is formed by all possible curves in parameters range $\theta \in (0, \pi]^\circ$ and $v_t [0, 40]$ inch. Then differential evolution (DE), that is an evolutionary optimization method is used to find time optimal trajectory from this set. The curve segments are formed in two ways: a) serial expansion of the robot's equations (McLaurin series) and b) artificial neural networks (YSA). YSA model is used to form the curves instead of serial expansion of the robot's equations. Thus, the optimization time is decreased to approximately 1/7. There is no need to a controller for tracking the planned trajectory.

Keywords: Mobile robot, optimisation, trajectory planning, differential evolution.

*Yazışmaların yapılacağı yazar: Serkan AYDIN. aydinsk@elk.itu.edu.tr; Tel: (212) 285 35 76.

Bu makale, birinci yazar tarafından İTÜ Elektrik-Elektronik Fakültesi'nde tamamlanmış "Mobil robotlarda evrimsel metotlar ile optimal hareket planlama" adlı doktora tezinden hazırlanmıştır. Makale metni 25.06.2003 tarihinde dergiye ulaştırılmış, 11.07.2003 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 30.06.2004 tarihine kadar dergiye gönderilmelidir.

Giriş

Mobil robotlarda yörünge planlama üzerine son 20 yıldır oldukça yaygın bir şekilde çalışılmaktadır. Literatürdeki bu çalışmaları: 1) Kapalı planlamalar (gerçek zamanda, lokal) ve 2) Açık planlamalar (off-line, global) şeklinde sınıflandırılabilir (Zefran, 1996). Planlama, mobil robot hareket halindeyken yapıyorsa buna kapalı planlama denmektedir. Bu tür planlamalarda amaç ölçütümüz sadece hedefe varmaktan ibarettir. Diğer bir deyişle, kapalı planlamalarda, planlanan yörünge istenilen özellikleri sağlaması (minimum mesafe, minimum zaman vb.) ve mobil robotun hareket performansının en üst seviyede olması her zaman mümkün değildir. Bu şartlar ancak açık planlamalarla yerine getirilebilir. Genellikle açık-planlamalarda yörünge: *yol planlama*, planlanan yolu *düzenleştirme* ve düzenleştirilmiş yolu *izleme* olmak üzere 3 aşamalı bir hiyerarşiye sahiptirler. Açık planlamaları da, açık-ayrık ve açık-sürekli olmak üzere iki kısma ayırmak mümkündür. Açık-ayrık planlamalarda bu 3 aşama sırasıyla gerçekleşmektedir. Önce ayrıklaştırmış uzayda (hücrelere ayırma, görünebilirlik grafik metodu vb.) yol planlanır, daha sonra çeşitli şekillerde planlanan yol düzenlenir. 2. aşamada literatürde, B-spline eğrileri (Zhang ve Knoll, 1995), kübik spiraller (Weber, 2000), polinomal eğriler (Reuter, 1998) ve klothoidler (Scheur ve Xie, 1999) gibi değişik eğri aileleri kullanılmıştır. Son aşamada da bu düzenlenmiş yolu izleyecek bir kontrolöre ihtiyaç duyulmaktadır. Genellikle bu çalışmalarda amaç ölçütü minimum mesafe olarak alınmıştır. Diğer bir deyişle global bir amaç ölçütü olmamaktadır. Açık sürekli planlamalarda ise yörünge planlama 2 aşamada tamamlanabilmektedir. İlk aşamada çalışma uzayı ayrıklaştırılmadan planlama yapılır ve sonraki aşamada da bu planlanan yörünge bir kontrolör ile izlenir. Bu kısımdaki çalışmalarda da genellikle varyasyonel yöntemler kullanılmıştır. Lagrange çarpanları metodu ile oluşturulan denklem sistemi nümerik yöntemlerle çözülmüştür. Çevreden gelen kısıtlamalar ve mobil robotun tüm kısıtlamaları bu şekildeki bir yaklaşımla hesaba katılarak bulunan çözümün global olduğu ortadadır. Ama nümerik çözüm yöntemlerinin zayıflıkları ve planlanan yörünge izlenmesi için bir

kontrolöre ihtiyaç duyması gibi iki dezavantajlarının olduğu söylenebilir.

Bu tezdeki yaklaşımda, yörünge düz kısımlar (dönme hızı $v_s=0$) ve eğri kısımlardan (öteleme hızı $v_t=sabit$) oluşmaktadır. Düz kısımlarda öteleme hızının artıp azalması maksimum ivme (a_t) ile gerçekleştirilmektedir. Eğri parçacıklarında dönme hızının artış ve azalışları da maksimum ivme (a_s) ile gerçekleştirilirken öteleme hızı sabit alınmaktadır. Planlanan hareketin izlenebilir olması ve robotun devrilmesini önlemek amacıyla öteleme hızları olabilecek en yüksek sabit hızla dönülmektedir. Bulunan çözüm bu kabul altında global optimaldir. Bu şartlar altında herhangi bir öteleme hızı ve dönüş açısı için dönüş eğrisinin modellenmesi:

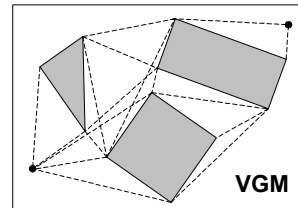
- Robotun denklemlerinin seri açılımıyla (Mclaurin)
- Yapay sinir ağı (YSA) öğrenme yoluyla

olmak üzere iki şekilde gerçekleştirilmiştir. En son olarak *diferansiyel evrim* (DE) ile, serbest konfigürasyon uzayında ($C_{serbest}$), düz kısım ve eğrilerden oluşan, global zaman-optimum amaç ölçütünü sağlayan yörüngeler elde edilmiştir.

Yörünge planlama

Görünebilirlik grafik metodu (VGM)

VGM, hareketli robotların yol planlamaları ile ilgili yapılan ilk çalışmalardan (Latombe, 1991). Bu yöntemde, öncelikle engellerin tüm köşelerini diğer engellerin görünen (engellerle kesişimi boş küme olan) köşelerine birleştiren doğrusal yol parçacıkları oluşturulur. Daha sonra hedef ve robot konfigürasyonları tarafından da görülen engel köşelerine de doğrusal yol parçacıkları çizilir. Bu doğrusal yol parçacıkları topluluğuna Görünebilirlik Grafikleri denir. Bu çalışmada oluşturulan yazılım ile yapılan ve basit bir ortam için gerçekleştirilmiş görünebilirlik grafiği Şekil 1'de verilmiştir.



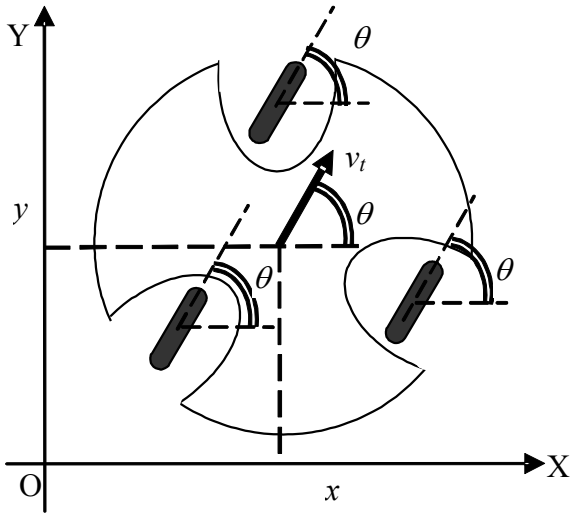
Şekil 1. Bir VGM grafiği

Eğri ve düz kısım denklemleri

Tekerlek yapısının Şekil 2’de verildiği, mobil robot (MR) Nomad 200’ün denklemlerinde düz ve eğri kısımların elde edilişi burada incelenmiştir. Robotun hareket esnasında kaymaması :

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0 \quad (1)$$

nonholonomik kısıtlamasının sağlanmasını gerektirir. Yani verilen bir geometrik konfigürasyon (x, y, θ) altında anlık öteleme hız vektörünün yönü her zaman tekerlek yönüne paraleldir.



Şekil 2. Nomad 200'ün tekerlek yapısı

Scheuer ve Xie (1999) geometrik konfigürasyona öteleme ve dönme hızlarını ve bunların türevlerini katarak (x, y, θ, v_t, v_s) dinamik konfigürasyon olarak isimlendirdiği bir kavramı kullanmıştır. Burada v_t öteleme hızı, v_s ise dönme (steering velocity) hızıdır. Denklem 1 bu uzayda:

$$\begin{aligned} \dot{x} &= v_t \cos(\theta) = v_x \\ \dot{y} &= v_t \sin(\theta) = v_y \\ \dot{\theta} &= v_s \\ \dot{v}_t &= a_t \\ \dot{v}_s &= a_s \end{aligned} \quad (2)$$

şeklinde gösterilebilir (parametrelerin üstündeki nokta zamana göre türev anlamındadır). Burada

a_t ve a_s sırasıyla öteleme ve dönme ivmeleridir. Robotun dinamik özelliklerinden dolayı hız ve ivmelerin üzerine:

$$\begin{aligned} |v_t| &\leq v_{t\max} \\ |v_s| &\leq v_{s\max} \\ |a_t| &\leq a_{t\max} \\ |a_s| &\leq a_{s\max} \end{aligned} \quad (3)$$

kısıtlamaları gelmektedir. Uygulamalarda kontrol vektörü $u=[v_t \ v_s]^T$ olarak alınmıştır. Denklem 2'nin her iki tarafının integralini alırsak:

$$\begin{aligned} x &= x_0 + \int_{t_0}^t v_t \cos(\theta) dt \\ y &= y_0 + \int_{t_0}^t v_t \sin(\theta) dt \\ \theta &= \theta_0 + \int_{t_0}^t v_s dt \\ v_t &= v_{t0} + \int_{t_0}^t a_t dt \\ v_s &= v_{s0} + \int_{t_0}^t a_s dt \end{aligned} \quad (4)$$

elde edilir. Giriş kısmındaki problem tanımımızdan dolayı Denklem 4'ün son iki eşitliğindeki $a_t=a_{t\max}$ ve $a_s=a_{s\max}$ olarak sabit alındığında :

$$\begin{aligned} v_t &= \left\{ v_{t0} + a_t \Big|_{t_0}^t = v_{t0} + a_t(t-t_0) \right\}_{a_t=a_{t\max}} \\ v_s &= \left\{ v_{s0} + a_s \Big|_{t_0}^t = v_{s0} + a_s(t-t_0) \right\}_{a_s=a_{s\max}} \end{aligned} \quad (5)$$

şeklini alır. Denklem 5'i Denklem 4'ün üçüncü eşitliğinde kullanırsak:

$$\begin{aligned} \theta &= \theta_0 + \int_{t_0}^t \left\{ (v_{s0} + a_s(t-t_0)) \right\}_{a_s=a_{s\max}} dt \\ &= \theta_0 + \left\{ (v_{s0} - a_s t_0)(t-t_0) + \frac{a_s(t-t_0)^2}{2} \right\}_{a_s=a_{s\max}} \end{aligned} \quad (6)$$

Denklem 5-6 Denklem 4'ün ilk iki eşitliğinde yerlerine konursa:

$$x = x_0 + \int_{t_0}^t \left\{ \cos \left(\theta_0 + (v_{s0} - a_s t_0)(t - t_0) + \frac{a_s (t - t_0)^2}{2} \right) \right\}_{a_s = a_{s \max}, a_t = a_{t \max}} dt \quad (7)$$

$$y = y_0 +$$

$$\int_{t_0}^t \left\{ \sin \left(\theta_0 + (v_{s0} - a_s t_0)(t - t_0) + \frac{a_s (t - t_0)^2}{2} \right) \right\}_{a_s = a_{s \max}, a_t = a_{t \max}} dt$$

denklemini elde edilir. Denklem 7'deki integrand terimlerinin içerisindeki trigonometrik ifadeler t^2 li ifadeler içerdiğinden dolayı bu integrallere *fresnel* integralleri denir (Kreyzsig, 1993). Belli bir çalışma noktası etrafında trigonometrik terimler:

$$\cos(\theta) = \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n}}{(2n)!} = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} \dots \quad (8)$$

$$\sin(\theta) = \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n+1}}{(2n+1)!} = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} \dots$$

şeklinde Mclaren serilerine açılarak yaklaşık sonuçlar elde edilebilir. Giriş kısmındaki çözüm yaklaşımımızdan hatırlanacağı gibi, yörüngeler düz ve eğri kısımlardan oluşmaktadır. İşlemlerin daha iyi görülebilmesi için, düz kısımlar ve dönüş eğrilerinde, $t_0=0$ alınarak bu kısımlar için Denklem 7 yeniden yapılandırılacaktır. Düz kısımlarda $a_s=0$ ve $v_{s0}=0$ olmasından dolayı

$$x = x_0 + \int_0^t \left\{ (v_{t0} + a_t t) \cos(\theta_0) \right\}_{a_t = a_{t \max}} dt = \left\{ x_0 + \left(v_{t0} t + a_t \frac{t^2}{2} \right) \cos(\theta_0) \right\}_{a_t = a_{t \max}} \quad (9)$$

$$y = y_0 + \int_0^t \left\{ (v_{t0} + a_t t) \sin(\theta_0) \right\}_{a_t = a_{t \max}} dt = \left\{ y_0 + \left(v_{t0} t + a_t \frac{t^2}{2} \right) \sin(\theta_0) \right\}_{a_t = a_{t \max}}$$

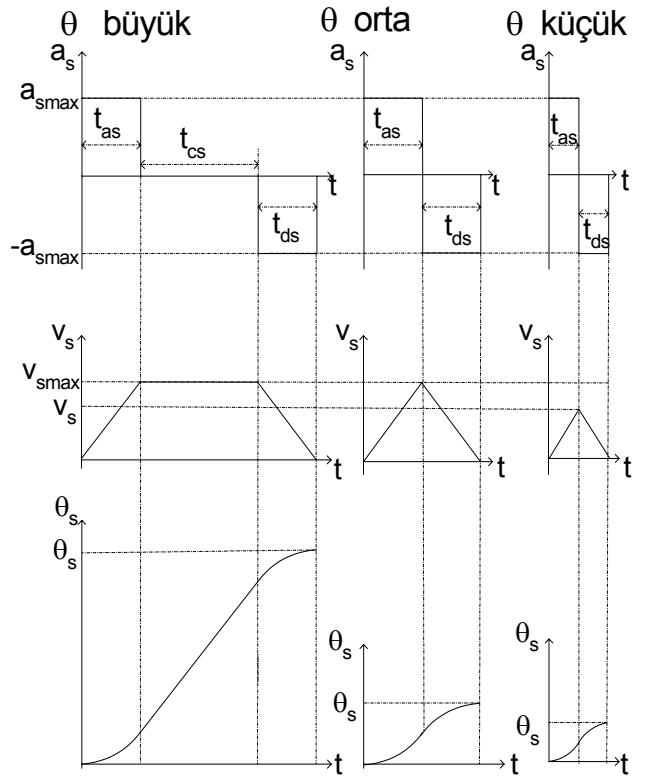
denklemini Denklem 7'den yazılabilir.

Eğri kısımlarda öteleme hızı $v_t=v_{t0}$ olarak sabit olduğundan $a_t=0$ alınınca:

$$x = x_0 + \int_0^t \left\{ \cos \left(\theta_0 + v_{s0} t + \frac{a_s t^2}{2} \right) \right\}_{a_s = a_{s \max}} dt \quad (10)$$

$$y = y_0 + \int_0^t \left\{ \sin \left(\theta_0 + v_{s0} t + \frac{a_s t^2}{2} \right) \right\}_{a_s = a_{s \max}} dt$$

şeklini alır. Denklem 10, Denklem 8'deki gibi seriye açılarak çözülür. Eğri kısımlar için a_s anahtarlama sürelerinin hesaplanması oldukça kolaydır ve Şekil 3'teki gibidir. a_s anahtarlama sürelerine göre Denklem 10'daki θ_0 , v_{s0} ve $a_{s \max}$ yerlerine konularak x ve y değerleri elde edilmektedir (Aydın ve Temeltaş, 2002).



Şekil 3. a_s anahtarlama süreleri

DE optimizasyon

Bir evrimsel (evolutionary) optimizasyon algoritması olan *Diferansiyel Evrim* (DE), kayan-nokta (floating-point) kodlanmış bir tür genetik algoritmadır (Storn, 1999). Rasgele seçilmiş iki parametre vektörünün farkının rasgele seçilmiş 3. bir parametre vektörü ile toplanması esasına dayanmaktadır. Aşağıda, bu genel çalışma esası, amaç ölçütü, parametre vektörü, kısıtlamalar ve populasyon gibi DE'nin tanım denklemleri anlatılarak, yeni populasyon üretimi ve seçim aşaması gibi çalışma teknikleri açıklanacaktır.

D boyutlu V parametre vektörüne bağlı amaç ölçütü $f(V):R^D \rightarrow R$ şeklinde ifade edilebilir. Amaç ölçütü $f(V)$ 'nin $V=\{v_1, \dots, v_D\}$ parametre uzayında minimize veya maksimize edilmesi problemin özünü teşkil etmektedir. Her bir parametre, $v_i^{(L)} \leq v_i \leq v_i^{(U)}$ $i=1, \dots, D$ gibi alt ve üst sınır kısıtlamalarına sahip olabilir. Buna ek olarak amaç ölçütü $f(V)$ üzerinde bazı lineer ve/veya lineer olmayan kısıtlamalar olabilir. Bu kısıtlamalar $c_i(V) \geq 0$ $i=1, \dots, m$ şeklinde ifade edilebilir. NP adet gerçel-değerli V parameter vektörden oluşan G'ninci nesil populasyonu P_G : $P_G = \{V_{1,G}, \dots, V_{NP,G}\}$ ($G=0, \dots, G_{max}$) ve bu üye parametre vektörlerden bir tanesi: $V_{i,G} = \{v_{1,i,G}, \dots, v_{D,i,G}\}^T$ ($i=1, \dots, NP$ $G=0, \dots, G_{max}$) şeklinde gösterilebilir. Burada G_{max} maksimum nesil sayısı ve D ise parametre vektörü boyutu olmaktadır. DE bir amaç ölçütünün optimizasyonunda, P_G 'yi oluşturan G'ninci nesil populasyonunun üyeleri üzerinde işlem yapar. Bu üyelerin sayısı NP optimizasyon süresince sabittir. Genellikle ilk nesil populasyon P_0 'ın parametreleri, Parametre üst ve alt sınır kısıtlamalarına uygun olarak, $v_{j,i,0} = \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)}$ ($i=1, \dots, NP$, $j = 1, \dots, D$) şeklinde üretilir. Burada $\text{rand}_j[0,1]$, $[0.0, 1.0]$ aralığında rasgele bir değerdir. Bu

genel tanım denklemlerinden sonra yeni populasyon üretilme tekniği ve bu esnada gerekli olan seçim aşaması ayrıntılı olarak incelenecektir.

Yeni populasyonun üretilme tekniği

P_{G+1} , G'ninci populasyon P_G 'nin rasgele alınan ve birleştirilen vektörlerinden oluşur. $P_{G+1}^A = (U_{1,G+1}, \dots, U_{NP,G+1})$ ile ifade edilen yeni nesile aday üye $U_{i,G} = (u_{1,i,G}, \dots, u_{D,i,G})$ 10 değişik tipte belirlenebilmektedir (Storn, 1999). Bu çalışmada kullanılan tip Denklem 11'de verilmiştir. Denklem 11'de $i = 1, \dots, NP$, $j = 1, \dots, D$ değişmektedir. $r_1, r_2, r_3, r_4, r_5 \in \{1, \dots, NP\}$, i'den farklı rasgele olarak seçilmektedir. $CR \in [0,1]$, $F \in (0,1+]$ aralıklarındadır. Her i elemanı için yeni rasgele r_1, r_2, r_3, r_4 ve r_5 değerleri üretilir. Gerçel-değerli değişken, F ((0.0,1.0+]), çaprazlama (crossing) değişkeni, CR ([0,1]) ve populasyondaki üye sayısı, NP; DE'nin ampirik olarak belirlenen kontrol parametreleridir.

Seçim aşaması

Kısıtlamaların olmadığı problemlerde, P_{G+1} 'in üyeleri $V_{i,G+1}$, şimdiki populasyon P_G 'nin üyeleri $V_{i,G}$ ve aday populasyon, P_{G+1}^A 'in üyeleri $U_{i,G+1}$ 'den, Lampinen (2001)'den daha hızlı bir şekilde uygulanabilir sonuçların elde edilebildiği:

$$\begin{aligned} & (\forall j \in \{1, \dots, m\}): \\ S1 & \quad c_j(U_{i,G+1}) \geq 0 \wedge c_j(V_{i,G}) \geq 0 \wedge (f(U_{i,G+1}) \leq f(V_{i,G})) \\ & \quad (\exists j \in \{1, \dots, m\}: c_j(V_{i,G+1}) < 0) \wedge (\forall j \in \{1, \dots, m\}): \\ S3 & \quad \min(c_j(U_{i,G+1}), 0) \geq \min(c_j(V_{i,G}), 0) \end{aligned}$$

olmak üzere,

$$V_{i,G+1} = \begin{cases} U_{i,G+1} & \text{Eğer } \{S1 \vee S3 \\ V_{i,G} & \text{diğer durumlarda} \end{cases} \quad (12)$$

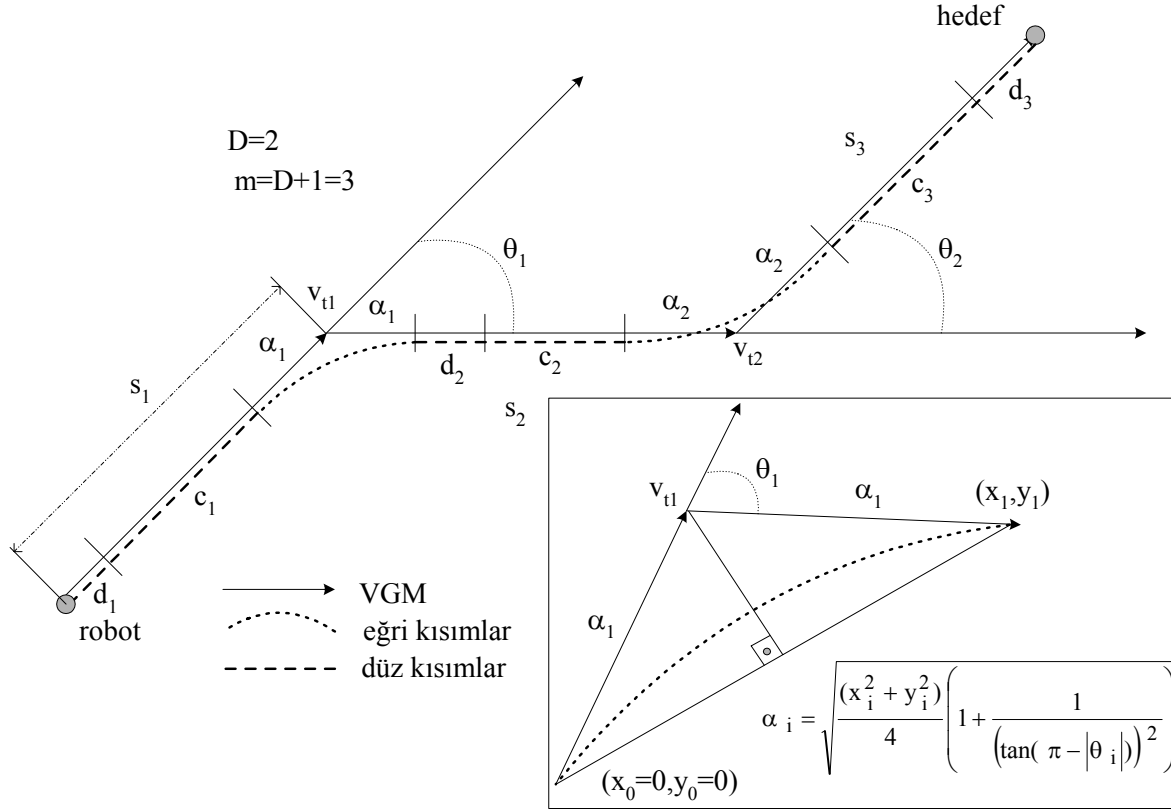
Seçim Kuralına (SK) göre seçilir.

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,r3,G} + F(v_{j,r1,G} - v_{j,r2,G}) & \text{Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} & \text{diğer durumlarda} \end{cases} \\ \text{Eğer } \text{rand}_j[0,1] \leq CR \\ v_{j,i,G} & \text{diğer durumlarda} \end{cases} \quad (11)$$

Yörünge planlamanın optimizasyon problemi olarak oluşturulması

Bu kısımda mobil robotlar için önceden VGM ile planlanmış yörünge üzerinde çoğul lineer/lineer olmayan kısıtlar altında oluşturulacak izlenebilir yörünge planı bir optimizasyon problemi olarak ele alınacaktır. VGM ile üretilen yol, doğrusal yol parçacıklarından (segment) ve dönüşlerden oluşmaktadır. VGM sonucu dönüş sayısı (D), doğrusal yol parçacıklarının uzunlukları (s_i $i=1,..,D+1$) ve açıları (θ_i $i=1,..,D$) elde edilmektedir. Bunlar her çevre veya robot pozisyonu değişiminde değişmektedir. Burada amaç ölçütü $f(V)$ VGM yolunu zamana göre optimize etmek için kullanılmaktadır. Şekil 4'te 3 doğrusal yol parçacığı ve 2 dönüşten ($D=2$) oluşan planlanmış bir VGM yörüngesi görülmektedir. Dönüş noktalarında süreksizlikler içeren bu VGM yolunu Nomad 200 öteleme hızı v_i 'yi sıfıra düşürerek izleyebilir. Sıfırdan farklı bir öteleme hızıyla dönüşler gerçekleştirilirse Şekil 4'teki düzleştirilmiş VGM yolu elde edilir. Böyle-

ce yörünge, bir önceki kısımda ayrıntılı denklemleri verilen *düz kısımlar* ve *eğri kısımlardan* oluşur. Düz kısımlarda dönüş hız ve ivmesinin, eğri kısımlarda ise öteleme ivmesinin sıfır olarak alındığı önceki bölümde anlatılmıştı. Bu yaklaşım ile zaman-optimal problemimiz, dönüş öteleme hızlarını ayarlama problemine dönüşmektedir. Böylece, D dönüşü sahip bir VGM'de, parametre vektörümüz V , eğri kısımlardaki öteleme hızlarından oluşan D boyutlu $V_t=(v_{t1},v_{t2},...,v_{tD})$ vektörü olmaktadır. Şekil 4'te görülen v_{ti} , α_i , d_i ve c_i değişkenlerinin neleri ifade ettikleri ve nasıl hesaplandıkları verildikten sonra, mobil robotlarda hareket planlama problemini çoğul sınırlamalı optimizasyon problemi olarak ifade etmek için kullandığımız denklemler ve eşitsizlikler oluşturulacaktır. v_{ti} , i . eğri kısım üzerindeki öteleme hızıdır. α_i , i . eğri kısmın i . ve $(i+1)$. VGM doğrusal yol parçacıkları üzerindeki mesafesidir. VGM üzerindeki i . dönüş açısı θ_i bilindiğinden dolayı x_0 ve y_0 başlangıç değerleri sıfır alınarak Denklem



Şekil 4. s_1 , s_2 ve s_3 doğrusal yol parçacıklarının boyutları; α_1 α_2 dönüşleri v_{t1} v_{t2} öteleme hızlarıyla yapabilmek için gereken mesafe; d_1 ise 0 'dan v_{t1} 'e, d_2 ise v_{t1} 'den v_{t2} 'ye hız değişimi için gereken mesafe. Kısıtlamalar $c_1=s_1-\alpha_1-d_1$, $c_2=s_2-\alpha_1-\alpha_2-d_2$ olarak ifade edilebilir.

10'dan x_i, y_i değerleri hesaplandıktan sonra α_i mesafesi geometrik olarak hesaplanmaktadır (Şekil 4). d_i , birbirini takip eden iki ardışık dönüş eğrisindeki öteleme hızları aynı değilse birinci öteleme hızını ikinci öteleme hızına artırmak veya azaltmak için d_i kadar bir mesafe gerekmektedir. d_1 mesafesi, öteleme hızının sıfırdan birinci dönüş eğrisinin öteleme hızına ulaşması için gereken mesafe iken, sonuncu doğrusal yol parçasığındaki d_{D+1} mesafesi, öteleme hızının D . dönüş eğrisinin öteleme hızından sıfıra düşmesi için gereken mesafe olmaktadır. c_i , iki ardışık dönüş eğrisinin çakışmamaları için aradaki c_i mesafesinin sıfır veya sıfırdan büyük olması gerekmektedir. Böylece iki ardışık dönüş arasında $c_i \geq 0$ eşitsizliğini sağlayacak bir mesafe kalıyorsa bu ardışık dönüş hızları uygulanabilir demektir. Bu $i=1, \dots, D+1$ 'e kadar m adet eşitsizlik, problemimizdeki lineer olmayan kısıtlamaları oluşturmaktadırlar. Burada parametre vektörü V_t 'nin elemanları olan dönüş öteleme hızlarının da $v_{ti}^{(L)}=0$ alt ve $v_{ti}^{(U)}=v_{t_max}$ (robotun maksimum öteleme hızı) üst kısıtlamalarına sahip olduğunu ifade etmemiz gerekmektedir. i . dönüş eğrisindeki öteleme hızının üst limiti $v_{ti}^{(U)}$, dönüş öncesi ve sonrası doğrusal yol parçacıklarının uzunluklarıyla da sınırlandırılmaktadır. Şekil 4'teki α_i mesafesinin en büyük değeri, $v_{ti-1}=0$ ve $v_{ti+1}=0$ kabul edildiğinde $\min(s_i, s_{i+1})$ olacağı açıktır. Bu anlatılanları:

$$v_{ti}^{(U)} = \begin{cases} v_{tai} & \alpha_i = \min(s_i, s_{i+1}) \quad v_{tai} < v_{t_max} \\ v_{t_max} & \text{diğer durumlarda} \end{cases} \quad (13)$$

denklemlerle ifade edebiliriz. v_{tai} ancak yapay sinir ağları kullanılarak hesaplanabilmiştir. Bu yöntemler bir sonraki kısımda ayrıntılı olarak ele alınacaktır. Parametre sınırlamaları aynı zamanda veri havuzunun boyutlarını da belirlediğinden çözümlerde yakınsamayı hızlandırarak doğru değerlerin bulunmasında oldukça etkin olmaktadır.

Kullanılan değişkenlerin tanımlanmasından sonra çoğul kısıtlamalı optimizasyon probleminin oluşturulmasını ayrıntılı olarak inceleyebiliriz. İlk olarak optimizasyon işleminde kullanılan amaç ölçütü $f(V_t)$ 'yi :

$$f(V_t) = \sum_{i=1}^{D+1} \begin{cases} t_{ci} + t_{di} + \frac{t_{v_{ti}}}{2} & i=1 \\ \frac{t_{v_{ti-1}}}{2} + t_{di} + t_{ci} + \frac{t_{v_{ti}}}{2} & i \in [2, D] \\ \frac{t_{v_{ti-1}}}{2} + t_{ci} + t_{di} & i = D+1 \end{cases} \quad (14)$$

şeklinde yazalım. Burada t_{ci} , i . doğrusal yol parçasığıının c_i mesafesini kat etmek için, $t_{v_{ti}}$ i . eğri kısım için ve t_{di} ise i . doğrusal yol parçasığıındaki d_i mesafesi için gereken sürelerdir. $t_{v_{ti}}$ 'yi aşağıdaki gibi yazılabilir:

$$t_{v_{ti}} = \begin{cases} \frac{\theta_i - a_{s_max} t_{as_max}^2}{2 \sqrt{\frac{v_{s_max}}{\theta_i}}} + 2t_{as_max} & \theta_i > (a_{s_max} t_{as_max}^2) \\ \frac{v_{s_max}}{a_{s_max}} & \text{diğer durumlarda} \end{cases} \quad (15)$$

burada a_{s_max} maksimum dönüş ivmesi, v_{s_max} maksimum dönüş hızı olmak üzere:

$$t_{as_max} = t_{ds_max} = \frac{v_{s_max}}{a_{s_max}}$$

şeklinde yazılabilir. t_{as_max} , v_{s_max} 'a ulaşma ve t_{ds_max} ise v_{s_max} 'tan sıfıra düşme süreleridir. θ_i i . eğri kısımda gerçekleştirilen dönüş miktarıdır (Şekil 4). Denklem 14'teki t_{ci} ise

$$t_{ci} = \begin{cases} t_{cia} + t_{cik} + t_{cid} & v_{tci} = v_{t_max} \quad \text{için} \\ t_{cia} + t_{cid} & v_{tci} < v_{t_max} \quad \text{için} \end{cases} \quad (16)$$

şeklinde ifade edilebilir. Burada, i 'nci doğrusal yol parçasığı üzerinde ulaşılabilen maksimum öteleme hızı olan v_{tci} :

$$K1_i = \sqrt{\frac{v_{ti}^2}{2} + (c_i + d_i)a_{t_max}}$$

$$K2_i = \sqrt{\frac{v_{ti-1}^2}{2} + \frac{v_{ti}^2}{2} + (c_i + d_i)a_{t_max}} \quad \text{ve}$$

$$K3_i = \sqrt{\frac{v_{ti-1}^2}{2} + (c_i + d_i)a_{t_max}}$$

olmak üzere,

$$v_{tci} = \begin{cases} K1_i & K1_i \leq v_{t_max} \wedge i = 1 \text{ için} \\ K2_i & K2_i \leq v_{t_max} \wedge i \in [2, D] \text{ için} \\ K3_i & K3_i \leq v_{t_max} \wedge i = D+1 \text{ için} \\ v_{t_max} & \text{diğer durumlarda} \end{cases} \quad (17)$$

şeklinde elde edilir. Denklem 17'deki :

$$c_i(V) = s_i - \alpha_{i-1} - \alpha_i - d_i \geq 0 \quad i \in [1, D+1] \quad (18)$$

$$d_i = \begin{cases} v_{ti-1}t_{di} + t_{di} \frac{(v_{ti-1} - v_{ti})^2}{2a_{t_max}} & v_{ti-1} < v_{ti} \\ v_{ti-1}t_{di} - t_{di} \frac{(v_{ti-1} - v_{ti})^2}{2a_{t_max}} & v_{ti-1} \geq v_{ti} \end{cases} \quad (19)$$

$$t_{di} = \begin{cases} \frac{v_{ti} - v_{ti-1}}{a_{t_max}} & v_{ti-1} < v_{ti} \text{ ise} \\ \frac{v_{ti-1} - v_{ti}}{a_{t_max}} & v_{ti-1} \geq v_{ti} \text{ ise} \end{cases} \quad (20)$$

olarak bulunur. Denklem 18 ile elde edilen denklemler, yörünge planlama problemimizin lineer olmayan kısıtlamalarını oluşturmaktadır. s_i i. doğrusal yol parçacığının uzunluğu, α_i i. eğri kısmın doğrusal yol parçacığı üzerindeki mesafesini belirtirken, t_{di} ve d_i sırasıyla ardışık iki eğri kısımdaki öteleme hızının uyumlandırma süresi ve bu sürede alınan yolu ifade etmektedir. Denklem 16'daki t_{cia} , v_{ti-1} 'den ((i-1). eğri kısmın öteleme hızı) v_{tci} 'ye ulaşma ve t_{cid} ise v_{tci} 'den v_{ti} 'ye (i. eğri kısmın öteleme hızı) yavaşlama süresi:

$$t_{cid} = t_{cia} = \begin{cases} \frac{v_{tci}}{a_{t_max}} & i = 1 \quad \text{ için} \\ \frac{v_{tci} - v_{ti-1}}{a_{t_max}} & i \in [2, D+1] \quad \text{ için} \end{cases}$$

denklemleriyle bulunmaktadır. $v_{tci} = v_{t_max}$ olma durumunda sabit v_{t_max} hızında gidilen süre t_{cik} :

$$t_{cik} = \begin{cases} \left[c_i - \begin{pmatrix} v_{ti}t_{cia} + \frac{1}{2}a_{t_max}t_{cia}^2 + \\ v_{ti}t_{cid} - \frac{1}{2}a_{t_max}t_{cid}^2 \end{pmatrix} \right] / v_{tci} & i \in [1, D] \\ \left[c_i - \begin{pmatrix} v_{ti-1}t_{cia} + \frac{a_{t_max}t_{cia}^2}{2} + \\ v_{ti-1}t_{cid} - \frac{a_{t_max}t_{cid}^2}{2} \end{pmatrix} \right] / v_{tci} & i = D+1 \end{cases}$$

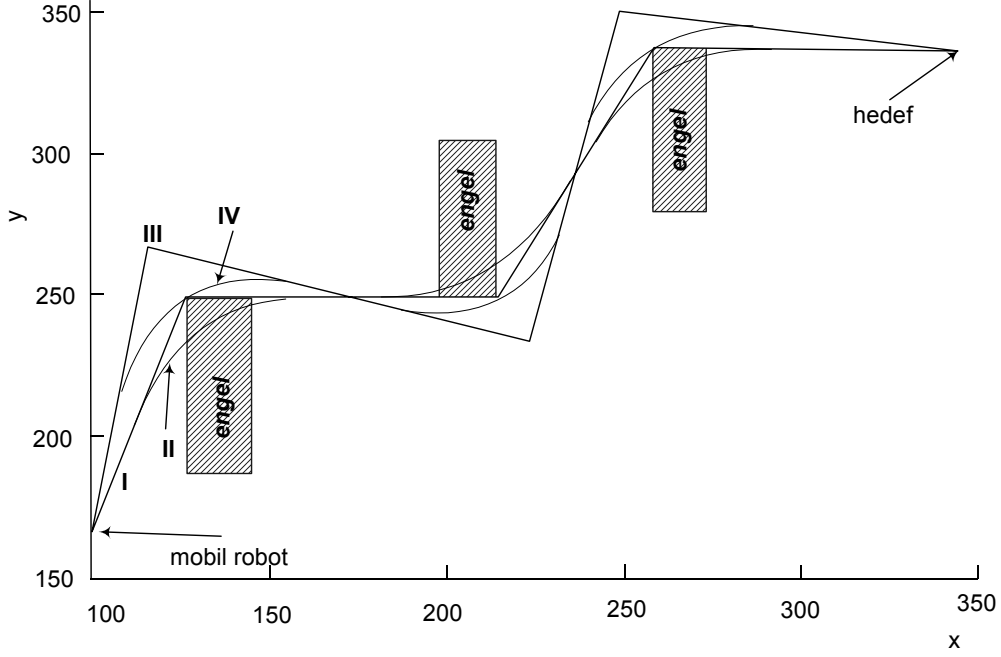
olarak ifade edilebilir. Böylece, Denklem 16'nın hesaplanabilmesi için gereken v_{tci} , t_{cia} , t_{cik} ve t_{cid} değerlerinin hepsi incelenmiş oldu.

Yukarıda verilen 14-20 denklemleri ile amaç ölçütü ve kısıtlamalar hesaplanmaktadır. Yörünge uzunluğu ise:

$$g(V_t) = \sum_{i=1}^{D+1} \begin{cases} d_i + c_i + \frac{t_{vti}v_{ti}}{2} & i = 1 \\ \frac{t_{vti-1}v_{ti-1}}{2} + d_i + c_i + \left(\frac{t_{vti}v_{ti}}{2}\right) & i \in [2, D] \\ \frac{t_{vti-1}v_{ti-1}}{2} + c_i + d_i & i = D+1 \end{cases}$$

denklemleri ile hesaplanmaktadır. Böylece, DE ile minimizasyon için gerekli olan, popülasyondaki her parametre vektörü V_{ti} için amaç ölçütü $f(V_{ti})$ 'nin ve lineer olmayan kısıtlamalar $c_i(V_{ti})$ 'nin aldığı değerlerin nasıl bulunduğu ayrıntılı bir şekilde incelemiş olduk. Bundan sonra, DE ile minimizasyon işlemi gerçekleştirilerek Denklem 14'ü minimize eden V_t parametre vektörü bulunmaktadır. VGM'de ardışık iki doğrusal yol parçacığının birleşimi engellerin köşeleri olmaktadır. Dolayısıyla her $v_{ti} \neq 0$ için oluşan eğri kısım engel üzerinde bulunmaktadır. Bu sorunun üstesinden gelmek için geliştirilen sanal görünebilirlik grafik metodu (VVGGM) önerilmiştir.

Sanal görünebilirlik grafik metodu: VVGGM
Şekil 5'te I ile VGM tarafından üretilen yol gösterilmektedir. VGM yolu üzerinde zaman-optimizasyon işlemi yapılmasıyla Şekil 5'teki II numaralı yörünge elde edilmektedir. Bu yörünge engel üzerinden geçtiği görülmektedir. Engel köşesi ile yörünge arasındaki maksimum

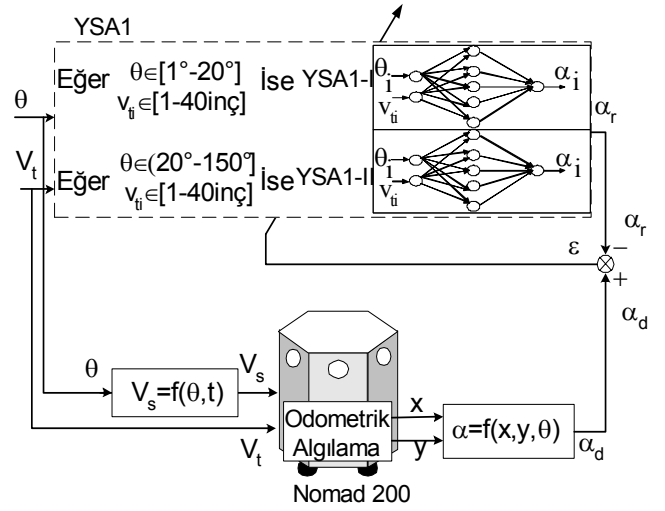


Şekil 5. Sanal görünebilirlik grafik metodu ile elde edilen yol.

mesafe kadar (sapma mesafesi) I numaralı VGM yolu engellerden dışarı doğru taşınarak III numaralı VVGM ismi verilen yol elde edilerek, bu yörünge üzerinde diferansiyel evrim ile zaman-optimal yörünge bulunur (IV). Böylece Şekil 5-IV'te görülen yörünge, engellerle kesişimi bulunmayan zaman-optimal yörünge'dir.

YSA ile eğri kısımların modellenmesi

Bu tez içerisinde işlem süresini azaltmak için yörünge'deki eğri kısımların hesaplanmasında ve dönüş öteleme hızlarının üst sınırlarının bulunmasında, yapay sinir ağları (YSA) kullanılmış ve sonuçları incelenmiştir. Yörüngelerdeki eğri kısımların modellenmesinde önerilen YSA uygulaması Şekil 6'da verilmiş olup bu uygulama YSA1 olarak isimlendirilmiştir. Bu önerilen YSA1 yapısında, 2 giriş, 5 sinir hücreli 1 gizli katman ve 1 çıkıştan (2:5:1) oluşmaktadır. Bu iki giriş: dönüş açısı ($\theta \in [1^\circ - 150^\circ]$) ve dönüşlerdeki öteleme hızı ($v_{ti} \in [1-40\text{inç}]$) şeklinde ifade edilir. Çıkış ise, robot doğrusal yol parçacığı üzerinde dönme hareketinin başladığı noktadan itibaren kalan uzunluk miktarı olan α_i 'dir. Bu çalışmada, çok katmanlı ileri beslemeli ve hatanın geriye yayılımının kullanıldığı bir ağ mimarisi ile öğrenme işlemi önerilmiştir.



Şekil 6. YSA ile eğri kısımların modellenmesi

Öğrenme oranı ve momentum katsayısı sırasıyla 0.9 ve 0.7 alınmıştır. Bu değerlerin uygunluğu deneysel olarak bulunmuştur. Deneysel çalışmalarda 2 girişli, 5 hücreli 2 gizli katmanlı ve 1 çıkışlı bir YSA yapısı (2:5:5:1) istenilen hata toleranslarını en iyi gerçekleyen bir yapı olmuştur. Ancak, (2:5:5:1) eğitimi sonucunda dönüş açılarına düşen bağıl hataların kendi aralarında sınıflandırılabilmesinden dolayı, buradaki fonksiyon değer hesap sürelerini daha

fazla düşürmek için YSA1 yapısında 2 ağ aynı anda eğitilerek aynı hassasiyetteki sonuçlar daha hızlı elde edilmiştir. Böylece YSA1 yapısı iki ayrı ağdan oluşmuştur. Bunlardan biri, YSA1-I ile adlandırılan ve giriş değer aralıkları $\theta \in [1^\circ - 20^\circ]$ ve hız $v_{ti} \in [1-40 \text{ inç}]$ olarak belirlenmiştir. Diğer ağ yapısı YSA1-II olarak adlandırılmış ve giriş değer aralıkları $\theta \in (20^\circ - 150^\circ]$ ve hız $v_{ti} \in [1-40 \text{ inç}]$ olarak verilmiştir. YSA1-I ve YSA1-II için sırasıyla 154 ve 938 giriş çifti kullanıldı. Yine sırasıyla 3239 ve 20698 çiftle testler gerçekleştirildi. Bu YSA yapılarının hata değerleri Tablo 1’de verilmektedir.

Tablo 1. 30000 iterasyon sonrasında gerçekleştirilen YSA1-I, YSA1-II, YSA2-I ve YSA2-II ağ yapılarının test fazı hata değerleri

	Ağ Mimarisi	E_O %	E_M %	E_5 / E_{10}
YSA1-I	2:5:1	0.88	7.19	6 E_5
YSA1-II	2:5:1	0.76	6.22	14 E_5
YSA2-I	2:9:1	2.2	23	17 E_{10}
YSA2-II	2:9:1	4.5	15.7	945 E_{10}

Tablo 1’deki karşılaştırmalarda kullanılan hata değerlerini ifade etmek için :

$$E = \{ \varepsilon_i \mid \varepsilon_i = (\alpha_{di} - \alpha_{ri}) / \alpha_{ri} \quad i \in [1, n] \} \quad (21)$$

bağlı hata kümesini tanımlayalım. Bu hata kümesindeki elemanların ortalaması:

$$E_O = \left(\sum_{i=1}^n 100|\varepsilon_i| \right) / n \quad \% \quad (22)$$

bize E_O ortalama bağlı hata değerini verirken,

$$E_M = \max(|100E|) \quad \% \quad (23)$$

ise hata kümesinin maksimumunu vermektedir. Bu küme içinde bağlı hatası %5 ten yüksek olan eleman sayısını :

$$E_5 = \{ \varepsilon_i \mid |100\varepsilon_i| > 5 \quad i \in [1, n] \} \quad (24)$$

verirken, % 10’dan büyük olan bağlı hataya sahip eleman sayısını ise:

$$E_{10} = \{ \varepsilon_i \mid |100\varepsilon_i| > 10 \quad i \in [1, n] \} \quad (25)$$

vermektedir.

Parametre vektörünün üst sınırı $V_t(U)$ ’nun her elemanı üretilen yoldaki doğrusal yol parçacığı uzunluklarına bağlı olarak (0, $v_{t_max}=40$] inç/s aralığında olabilmektedir. Denklem 13’e göre bu üst limitler belirlenmektedir. Bu amaçla, 2 girişli, 9 sinir hücreli 1 gizli katmanlı ve 1 çıkışlı (2:9:1) YSA yapısı kullanıldı ve bu yapı YSA2 olarak adlandırıldı. Burada da bir önceki kısımda belirtilen gerekçelerden dolayı, iki giriş değer aralığı için iki değişik ağ eğitildi ve bu yapılar YSA2-I ve YSA2-II olarak adlandırıldı. Eğitimde kullanılan giriş çiftleri:

- YSA2-I için : $(\theta_i \in [1^\circ - 20^\circ], \alpha_i)$ 154 adet

- YSA2-II için : $(\theta_i \in (20^\circ - 150^\circ], \alpha_i)$ 938 adet

olarak alınmıştır. 30000 iterasyon ile gerçekleştirilen eğitimden sonra sırasıyla YSA2-I 3239 ve YSA2-II 20698 çiftle yapılan test ile elde edilen bağlı hatalar Tablo 1’de görülmektedir.

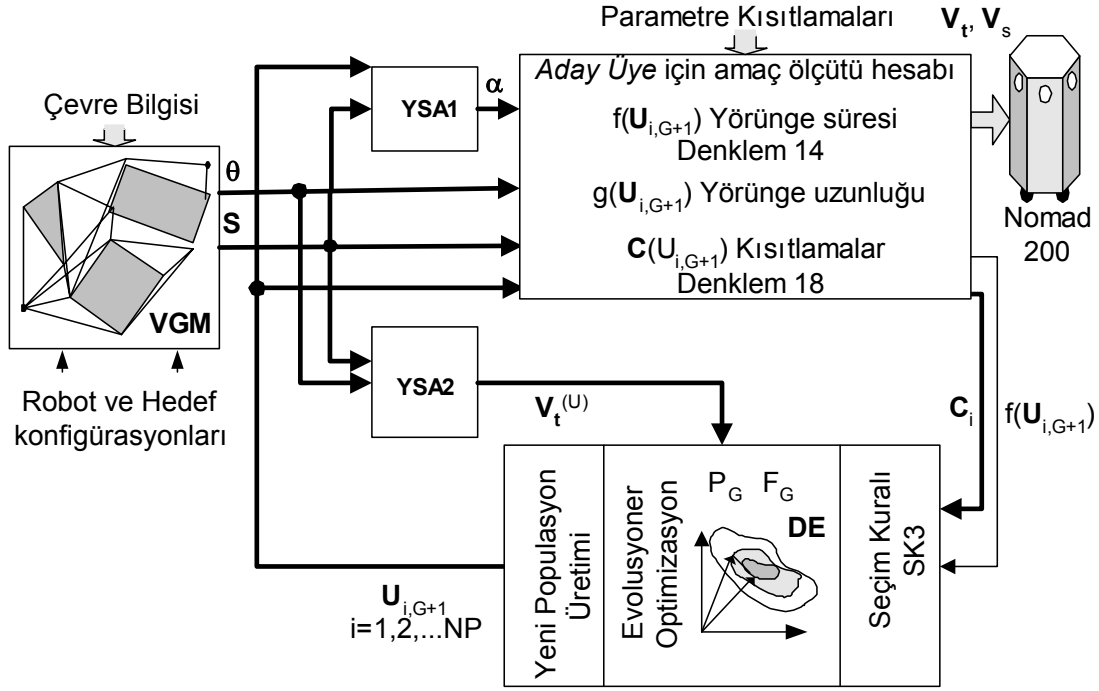
Eğri kısımların modellenmesinde YSA ve Denklem 10’un kullanılması ile ortaya çıkan hesaplama süreleri Tablo 2’de verilmiştir. Bu tabloda, 2:5:5:1 YSA modeli ile işlem süresinin yaklaşık 1/6’ya ve 2:5:1 YSA modeli ile 1/8’e kadar düştüğü görülmektedir.

Tablo 2. Deterministik ve YSA modelleriyle eğri kısım hesaplama süreleri

Model	Süre* (s)	Bir Eğri Hesap Süresi
Denklem 10	40.09	1.68×10^{-3}
2:5:5:1 YSA	6.85	2.88×10^{-4}
2:5:1 YSA	4.91	2.06×10^{-4}

*: 23779 eğri hesabının gerçekleştirilme süresi

Bu kısımda şu ana kadar hazırlanan tüm tekniklerle optimal yörünge planlamanın nasıl oluştuğu ele alınacaktır. Şekil 7’de DE-YSA ile optimal yörünge planlama süreci görülmektedir. Burada öncelikle (S, Θ) vektörleri VGM tarafından üretilmekte, daha sonra YSA1 ağ yapısı (θ_i, v_{ti}) çiftini giriş olarak alıp, α_i çıkışını hesaplamaktadır. Bu oluşan, (S, Θ , α , V_t) parametre



Şekil 7 Soft hesaplmalı DE ile optimal yörünge planlama

kümesi, DE için gerekli olan amaç ölçütü $f(V_t)$ 'nin ve C kısıtlamalarının oluşturulmasında kullanılmıştır.

DE'de kullanılan $v_{ti}^{(U)}$ ($i=1...D$) parametre üst kısıtlamaları da YSA2 ile hesaplanmıştır. Böylece Şekil 7'de açıkça görüldüğü gibi, DE optimizasyon algoritması için gerekli olan (S, Θ , α , V_t , $V_t^{(U)}$, C, $f(V_t)$) gibi bileşenler hesaplanmış olur. Bu bileşenlerin DE optimizasyonunda kullanılarak elde edilen yolun Nomad 200 robotu tarafından gerçekleştirilmesi için $V_t(t)$ ve $V_s(t)$ kontrol değişkenleri elde edilmiş olur.

Şekil 4'te verilen çevre için DE-YSA ile planlanan VGM yolu üzerinde zaman-optimal yörünge sonuçları Tablo 3 ve Tablo 4'te verilmiştir.

Tablo 3. Şekil 5 için optimizasyon sonuçları.

	$f(V_t)$ (s)	v_{t1} (inç/s)	v_{t2} (inç/s)	v_{t3} (inç/s)
II	10.1706	33.2118	33.2118	39.6535
IV	12.1938	29.8211	29.8211	35.5987
I	21.6985	0	0	0
III	25.3119	0	0	0

Tablo 4. Şekil 5 için kısıtlamalar

	$G(V_t)$ (inç)	c_1 (inç)	c_2 (inç)	c_3 (inç)	c_4 (inç)
II	344.51	27.09	0.0	0.0	31.119
IV	377.54	33.77	0.0	0.0	33.881
I	368.68	92.48	88.582	99.03	88.582
III	442.13	107.01	114.06	121.41	99.637

VGM yolunun izlenme süresi 21.6985 s iken, optimizasyon işlemi sonucu oluşan IV numaralı yörünge için gereken süre 12.1938 s'ye düştüğü ve bu yeni yörünge en kısa VGM yolundan yaklaşık % 2 kadar ($\cong 9$ inç) uzun olduğu Tablo 3'te görülmektedir.

Sonuçlar ve tartışma

Bu yayında, mobil robot Nomad 200 için görünebilirlik grafik metodu ile planlanan yol üzerinde zaman-optimal yörünge diferansiyel evrim-yapay sinir ağları kullanılarak bulunmuştur. Yörüngeler, dönüş hız ve ivmesinin 0 olduğu düz ve öteleme hızının sabit dönüş esnasındaki ivmelenmelerin maksimum olduğu eğri kısımlardan oluşmaktadır. Eğri kısımlar robotun denklemlerinin McLoren serisine açılımıyla ve

YSA ile elde edilmiştir. YSA'nın kullanımı hesaplama süresini yaklaşık 1/8'e düşürmüştür.

Mobil robotlarda zaman-optimal hareket problemi çoğul kısıtlamalı optimizasyon problemine dönüştürüldükten sonra, DE ile zaman-optimal yörüngedeki eğri kısımlar bulunmuştur. Oluşturulan yörüngelerin eğri kısımlarının engellerle çakışmasının engellenmesi için VVGM geliştirilmiştir. Kısıtlamalı problemler için Lampinen (2001)'in kullanmış olduğu seçim kuralından daha hızlı yakınsayan farklı bir seçim kuralı önerilmiş ve arama yapılan parametre uzayını daraltmak için parametre vektör elemanlarımız olan dönüş öteleme hızlarının üst sınırları YSA ile bulunarak daha az nesilde optimizasyon işleminin tamamlanması gerçekleştirilmiştir.

Kaynaklar

- Aydın S., Temeltaş H., (2002). A Novel Approach to Smooth Trajectory Planning of a Mobile Robot, *7th International Workshop on Advanced Motion Control*, Maribor, Slovenia.
- Kreyszig E., (1993). *Advanced Engineering Mathematics*. John Wiley & Sons, inc., Singapore.
- Lampinen J., (2001). Solving Problems Subject to Multiple Nonlinear Constraints by The

- Differential Evolution, *Proc. of MENDEL2001 7th Int Conf on Soft Computing*, Brno Czech Republic.
- Latombe J.C., (1991). *Robot Motion Planning*. Kluwer Academic Publishers. Boston.
- Reuter J., (1998). Mobile robots trajectories with continuously differentiable curvature: An optimal control approach, *IEEE/RSJ Proc. Of Int. Conf on Intelligent Robots and System*, Victoria B.C., Canada.
- Scheuer A., Xie M., (1999). Continuous-Curvature Trajectory Planning for Manoeuvrable non-Holonomic Robots, *IEEE/RSJ Proc. Of Int. Conf on Intelligent Robots and System*.
- Storn, R., (1999). System Design by Constraint Adaptation and Differential Evolution, *IEEE Transactions on Evolutionary Computation*, **3**, 1, 22–34.
- Weber H., (2000). A motion planning and execution system for mobile robots driven by stepping motors, *Robotics and Autonomous Systems*, **33**, 207–221.
- Zhang J., Knoll A. (1995). An Enhanced Optimization Approach for Generating Smooth Robot Trajectories in the Presence of Obstacles. *Proc. Of the European Chinese Automation Conference*, London.
- Zefran M., (1996). Continuous Methods for Motion Planning, *PhD Thesis*, University of Pennsylvania, Philadelphia.