

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**4 SERBESTLİK DERECELİ ROBOT KOLU KİNEMATİK
DENKLEMLERİNİN DESTEK VEKTÖR MAKİNASI İLE ÇÖZÜMÜ**

YÜKSEK LİSANS TEZİ

Sanem DOKUZLU

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

MAYIS-2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**4 SERBESTLİK DERECELİ ROBOT KOLU KİNEMATİK
DENKLEMLERİNİN DESTEK VEKTÖR MAKİNASI İLE ÇÖZÜMÜ**

YÜKSEK LİSANS TEZİ

**Sanem DOKUZLU
(518121025)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Yard. Doç. Dr. Gülay ÖKE GÜNEL

MAYIS-2015

İTÜ, Fen Bilimleri Enstitüsü'nün 518121025 numaralı Yüksek Lisans Öğrencisi **Sanem DOKUZLU** ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**4 SERBESTLİK DERECELİ ROBOT KOLU KİNEMATİK DENKLEMLERİNİN DESTEK VEKTÖR MAKİNASI İLE ÇÖZÜMÜ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yard. Doç.Dr. Gülay ÖKE GÜNEL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Yard. Doç. Dr. Hatice KÖSE**
İstanbul Teknik Üniversitesi

Yard. Doç. Dr. Dilek BİLGİN TÜKEL
Doğuş Üniversitesi

Teslim Tarihi : 4 Mayıs 2015
Savunma Tarihi : 29 Mayıs 2015

ÖNSÖZ

Bu tez çalışması sırasında öncelikle, her zaman bilgi ve fikirlerini aktarmaya çalışan değerli tez danışmanım Yrd. Doç. Dr. Gülay Öke Günel'e ve bu sürecin oluşturulmasındaki katkılarından ötürü Araştırma Görevlisi Kemal Uçak'a teşekkürlerimi sunarım. Çalışma sürecinin başından sonuna kadar desteklerini sürekli hissettiren aileme ayrıca sonsuz teşekkür ederim.

Mayıs 2015

Sanem Dokuzlu
(Kontrol Mühendisi)

İÇİNDEKİLER

Sayfa

| | |
|---|-----|
| ÖNSÖZ..... | v |
| İÇİNDEKİLER..... | vii |
| ÇİZELGELİSTESİ..... | vii |
| ŞEKİL LİSTESİ..... | xi |
| ÖZET..... | iii |
| SUMMARY..... | xv |
| 1. GİRİŞ..... | 1 |
| 2. 4 EKSENLİ ROBOT KOLU VE KİNEMATİK DENKLEMLERİ..... | 4 |
| 2.1 4 Eksenli Robot Kolu..... | 4 |
| 2.1.1 Robot kolu özellikleri..... | 4 |
| 2.2 4 Eksenli Robot Kolu Kinematik Denklemleri..... | 5 |
| 2.2.1 İleri kinematik model..... | 5 |
| 2.2.2 Ters kinematik model..... | 8 |
| 2.2.2.1 Geometrik yaklaşım..... | 9 |
| 2.2.2.2 Ters Jakobiye yöntemi..... | 10 |
| 2.2.2.2.1 Jakobiye matrisinin hesaplanması..... | 10 |
| 2.2.2.2.2 Jakobiye matrisinin tersinin hesaplanması..... | 10 |
| 2.3 Çalışma Uzayının Belirlenmesi..... | 11 |
| 3. DESTEK VEKTÖR MAKİNELERİ..... | 12 |
| 3.1 Giriş..... | 12 |
| 3.2 Tarihi Gelişim..... | 14 |
| 3.3 İstatiksel Öğrenme..... | 15 |
| 3.3.1 Vapnik-Chervonenkis teorisi..... | 16 |
| 3.3.2 VC boyutu..... | 16 |
| 3.3.3 Yapısal risk minimizasyonu..... | 16 |
| 3.4 Destek Vektör Sınıflandırması..... | 17 |
| 3.4.1 Lineer olarak ayrılabilen örüntüler..... | 17 |
| 3.4.1.1 Quadratic optimizasyon – Lagrange çarpanları yöntemi..... | 19 |
| 3.4.2 Genel haliyle sınıflandırma..... | 21 |
| 3.4.3 Doğrusal olmayan destek vektör makineleri..... | 23 |
| 3.5 Destek Vektör Makineleri ile Regresyon..... | 25 |
| 3.5.1 ϵ -sensitive kayıp fonksiyonu..... | 26 |
| 3.5.2 Doğrusal regresyon..... | 26 |
| 3.5.3 Doğrusal olmayan regresyon..... | 28 |
| 3.6 En Küçük Kareler Destek Vektör Makinesi..... | 29 |
| 3.6.1 En küçük kareler destek vektör sınıflandırması..... | 29 |
| 3.6.2 En küçük kareler destek vektör regresyonu..... | 30 |

| | |
|---|-----------|
| 4. BENZETİM SONUÇLARI..... | 33 |
| 4.1 4 Eksenli Robot Kolun Destek Vektör Makinası Yardımı ile İleri Kinematik Modellemesi..... | 33 |
| 4.1.1 Grid Arama Yöntemi ile Optimum Parametrelerin Bulunması..... | 37 |
| 4.2 4 Eksenli Robot Kolun Destek Vektör Makinası Yardımı ile Ters Kinematik Modellemesi..... | 42 |
| 4.2.1 Grid Arama Yöntemi ile Optimum Parametrelerin Bulunması..... | 46 |
| 5.SONUÇLAR..... | 52 |
| KAYNAKLAR..... | 53 |
| EKLER..... | 56 |
| ÖZGEÇMİŞ..... | 76 |

ÇİZELGE LİSTESİ

Sayfa

| | |
|---|----|
| Çizelge 2.1 : D-H parametreleri tablosu | 5 |
| Çizelge 4.1 : Robot kolu D-H parametreleri..... | 32 |
| Çizelge 4.2 : Robot kolu mekanik sınır eklem açı değerleri..... | 33 |
| Çizelge 4.3 : Destek vektör makinası regresyon parametreleri | 35 |
| Çizelge 4.4: Destek vektör makinası regresyon eğitim ve test hata değerleri | 37 |
| Çizelge 4.5: Destek vektör algoritması optimum regresyon parametreleri | 40 |
| Çizelge 4.6: Destek vektör makinası regresyon optimum eğitim ve test hataları | 41 |
| Çizelge 4.7:Robot kolu çalışma uzayı sınır değerleri..... | 42 |
| Çizelge 4.8 : Destek vektör makinası regresyon parametreleri | 44 |
| Çizelge 4.9 : Destek vektör makinası regresyon eğitim ve test hata değerleri | 46 |
| Çizelge 4.10: Destek vektör makinası optimum regresyon parametreleri..... | 49 |
| Çizelge 4.11 Destek vektör makinası regresyon optimum eğitim ve test hataları..... | 51 |

ŞEKİL LİSTESİ

Sayfa

| | |
|---|----|
| Şekil 2.1 : Manipülâtörün taslak tasarımı | 4 |
| Şekil 2.2 : D-H yöntemi için koordinat eksenlerinin gösterimi..... | 5 |
| Şekil 2.3 : Koordinat eksenlerinin robot kol üzerinde gösterimi..... | 6 |
| Şekil 2.4 : Robot kolu çalışma uzayı. | 11 |
| Şekil 3.1 : İstatiksel öğrenme..... | 15 |
| Şekil 3.2 : VC boyutları..... | 16 |
| Şekil 3.3 : Optimal hiperdüzlem..... | 17 |
| Şekil 3.4 :Destek vektörleri | 18 |
| Şekil 3.5 : Genelleştirilmiş veri | 20 |
| Şekil 3.6 : Doğrusal olmayan haritalama | 23 |
| Şekil 3.7 : Kayıp fonksiyonları..... | 25 |
| Şekil 3.8 : Soft margin doğrusal destek vektör..... | 26 |
| Şekil 4.1 : Robot kolu MATLAB/Simmechanics modeli..... | 32 |
| Şekil 4.2 : Eğitim verisi giriş değerleri (Robot eklem açıları)..... | 34 |
| Şekil 4.3 : Eğitim verisi çıkış değerleri(Robot uç işlevci konumları) | 34 |
| Şekil 4.4 : Destek vektör makinası performansı (X) | 35 |
| Şekil 4.5 : Destek vektör makinası performansı (Y) | 36 |
| Şekil 4.6 : Destek vektör makinası performansı (Z)..... | 36 |
| Şekil 4.7 : Eğitim hata yüzeyi(X) | 38 |
| Şekil 4.8 : Eğitim hata yüzeyi(Y) | 38 |
| Şekil 4.9 : Eğitim hata yüzeyi(Z)..... | 38 |
| Şekil 4.10 : Test hata yüzeyi (X) | 39 |
| Şekil 4.11 : Test hata yüzeyi (Y). | 39 |
| Şekil 4.12 : Test hata yüzeyi (Z)..... | 39 |
| Şekil 4.13 : Destek vektör makinası performansı (X). | 40 |
| Şekil 4.14 : Destek vektör makinası performansı (Y) | 41 |
| Şekil 4.15 : Destek vektör makinası performansı (Z)..... | 41 |
| Şekil 4.16 : Eğitim verisi giriş değerleri (Uç İşlevci Konumları)..... | 43 |
| Şekil 4.17: Eğitim verisi çıkış değerleri (Eklem Açısı Değerleri) | 43 |
| Şekil 4.18 : Destek vektör makinası performansı (1.Eklem açısı) | 44 |
| Şekil 4.19 : Destek vektör makinası performansı (2.Eklem açısı). | 45 |
| Şekil 4.20 : Destek vektör makinası performansı (3.Eklem açısı). | 45 |
| Şekil 4.21 : Destek vektör makinası performansı (4.Eklem açısı) | 46 |
| Şekil 4.22 : Eğitim hata yüzeyi (1.Eklem açısı) | 47 |
| Şekil 4.23 : Eğitim hata yüzeyi (2.Eklem açısı) | 48 |
| Şekil 4.24 : Eğitim hata yüzeyi (3.Eklem açısı) | 48 |
| Şekil 4.25 : Test hata yüzeyi (1.Eklem açısı). | 48 |
| Şekil 4.26 : Test hata yüzeyi (2.Eklem açısı). | 49 |
| Şekil 4.27 : Test hata yüzeyi (3.Eklem açısı). | 49 |

| | |
|---|----|
| Şekil 4.28 : Destek vektör makinası performansı (1.Eklem açısı)..... | 50 |
| Şekil 4.29 : Destek vektör makinası performansı (2.Eklem açısı)..... | 50 |
| Şekil 4.30 : Destek vektör makinası performansı (3.Eklem açısı)..... | 50 |
| Şekil 4.31: Destek vektör makinası performansı (4.Eklem açısı)..... | 51 |

4 SERBESTLİK DERECELİ ROBOT KOLUNUN KİNEMATİK DENKLEMLERİNİN DESTEK VEKTÖR MAKİNASI İLE ÇÖZÜMÜ

ÖZET

Kinematik problemin çözülmesi robot manipülatörlerinin yörünge takibi açısından büyük önem taşımaktadır. Kinematik modelleme için kullanılan analitik, geometrik ve numerik yöntemlerin her birinin değişik açılardan dezavantajı bulunmaktadır. Analitik ve geometrik yöntemler basit yapılı ve belirli bir geometrik dizilime sahip robot kollarında etkili olabilirken, çok serbestlik dereceli ve karmaşık yapıdaki robot kollarına çözüm sağlayan numerik yöntemlerin hesaplama zorluğu bulunmaktadır. Ayrıca bu yöntemler robot kol çalışma uzayındaki tekil noktalarda cevap verememektedir.

Yapay sinir ağları bu problemlere çözüm olabilecek alternatif bir yöntemdir. Robot koldan alınan giriş-çıkış verileri ile sistem eğitililerek kinematik denklemler modellenenilmektedir. Ancak model kestirimi yapılırken yerel minimum noktalara takılması yapay sinir ağlarının bir dezavantajıdır. Tezde, bu probleme çözüm olarak destek vektör makineleri algoritmasının kullanılması önerilmiştir. Doğrudan global minimum noktasına ulaşmayı öngören bu yöntem, kinematik model kestiriminin en az hatayla yapılmasını sağlamaktadır.

Bu tez çalışmasında, kinematik model çözümlemesine örnek olarak Gelecek Robotik Makine ve Tıp Teknolojileri Ar. Ge. Tic. Ltd. Şti. firmasının T.C. Bilim Sanayi ve Teknoloji Bakanlığı' ı desteğiyle yürütmekte olduğu “Robotik Tekerlekli Sandalye” adlı projesinde kullanılan rehabilitasyon robot kolu ele alınmıştır. 4 serbestlik dereceli olan bu rehabilitasyon robot kolu, paralel 2 parmaklı bir tutucuya sahiptir. Maksimum erişim mesafesi 400-500 mm’dir.

Bu çalışmanın amacı, ileri ve ters kinematik modeli, destek vektör makinası regresyonu ile tahmin ederek analitik, geometrik ve numerik çözümlemenin neden olduğu tekillik sorunu ortadan kaldırmak ve yörünge takibi problemine kinematik tabanlı hızlı ve güvenilir bir çözüm sunmaktır. Analitik yöntemle elde edilen eğitim verileri kullanılarak, 4 eksenli rehabilitasyon robot kolunun ileri ve ters kinematik modelleri destek vektör makinesi yöntemi ile oluşturulmuştur.

SUPPORT VECTOR MACHINE BASED SOLUTION FOR THE KINEMATICS EQUATIONS OF 4-DOF ROBOT ARM

SUMMARY

Solving kinematic problems has a great importance in terms of trajectory tracking of robot manipulators. Every analytic, geometric and numerical methods for solving this problem, caused differently disadvantages. Analytical and geometric methods may be effective in specific geometric arrangements and basic structures, numerical calculations is used for robot arms that have complex structure and many degrees of freedom. In addition, this method can not be answered in the singular point on the robot arm workspace.

Artificial neural networks is an alternative method that may be a solution to kinematic problem. In addition to this, using neural network is becoming very coming due to solve inverse kinematic problem. Neural networks can be used in every configuration of manipulators. Kinematic equations could be modeled with training data that is obtained from system input-output data. On the other hand, a disadvantage of artificial neural networks is that the model is fitted to estimating the local minimum point. In this thesis, an algorithm using support vector machines is proposed as a solution to this problem. This direct method foresees to reach the global minimum point so kinematic model estimation could be done with minimum error.

Applications of support vector machines have many areas from health to agriculture. In control engineering, support vector machine algorithm has an important place. Flight control, robust control, system identification are some of the application topics. Furthermore; in automotive, support vector machine can be used in engine control. In our topic-robotics- forward and inverse kinematics, trajectory planning and positioning problems have various solutions with support vector machine

regression. Especially, support vector machine regression provides an efficient solution to inverse kinematic problem.

In this study, kinematics analysis as an example of "Gelecek Robotik Makine ve Tıp Teknolojileri Ar. Ge. Tic. Ltd. Şti. firm . C. Science, Industry and Technology Ministry 's, which are executed with the support of "Robotic Wheelchair" rehabilitation robot arm used in his project were discussed. The aim of this project is helping disabled people with their daily activities.

The rehabilitation robotic arm has four degree of freedom and a parallel two-fingered holder. Maximum access distance is 400-500 mm. At the end of the project, the final product will be consist a robotic arm which is entegrated on a electric wheelchair, camera, laser scanner , proximity sensors, touchscreen and an embedded PC. Links of the robot arm is placed vertical to each other. Construction of the links can be seperated two types – vertical and horizontal. Apart from this, there are two types of links according to their torque values.

Rehabilitation robot arm project's aim is helping elder or diasbled people who have to use wheelchair for their basic daily activities. The project process is 18 months and the resultin product is a robotic manipulator prototype which can be controllde by a touchscreen. Another feature of this robot manipulator is that link numbers and kinematic structure are adjustable for users' needs. This feature adds an extra advantage to manipulator. With adjustablr kinematic structure, robotic arm can be used for industrial purposes.

Controlling the robot manipulator with inverse kinematic can be possible with learning the forward kinematic model by the help of the support vector machine algorithm. While learning forward kinematic model, optimum parameters of support vector machine regression must be used. In this study, optimum parameter research is accomplished by grid search algorithm. The same learning process is aplied inverse kinematic model. At the end of these processes , forward and inverse kinematic model of the robot arm is obtained.

The kinematic model, that is predicted by support vector machine algorythm, can be used in adaption of a computed torque PD controller's parameters. Computed torque controllers provide a dyanmic based solution to trajectory planning problem. By this

method , controller performans can be reduced and trajectory planning can be done with minimum error.

The aim of this study is that elimination of the singularity problem that is caused by geometric and analytic solutions with the estimation of forward and inverse kinematic models by using support vector machine regression. 4-axis rehabilitation robot arm forward and inverse kinematics model is created with support vector machine method by using the training data obtained from analytical solution.

1. GİRİŞ

Robot manipülatörlerinin kinematik probleminin çözülmesi yörünge takibinin gerçekleştirilmesi açısından büyük önem taşımaktadır. Bu problemin çözümünde kinematik ve dinamik tabanlı olmak üzere iki farklı yaklaşım kullanılmaktadır. Dinamik tabanlı kontrolörler karmaşık yapıda olduğundan, kinematik tabanlı kontrolörler çok daha basit ve güvenilir çözümler sunmaktadır [1]. Robot manipülatörlerinin kinematik problemi ileri ve ters kinematiğin incelenmesini kapsamaktadır.

İleri kinematik problemin kolay bir şekilde çözülebilmesine karşılık ters kinematik problem bir çok zorluk içermektedir. Ters kinematik problemin çözümü için literatürde çeşitli yöntemler sunulmaktadır. Bu yöntemler analitik, geometrik ve numerik çözümler olarak açıklanabilmektedir. Analitik ve geometrik yöntemler düşük serbestlik dereceli ve özel eklem dizilimine sahip robotlara kolayca uygulanabilmektedir. Yüksek serbest dereceli ve herhangi bir geometrik yapıya sahip robot manipülatörlerinde ise karmaşık yapı nedeniyle uygulaması zorlaşmaktadır. Numerik yöntemler ise robot diferansiyel denklemlerinin çözümüne dayanmaktadır. Diferansiyel denklem yapısı, robot geometrisinden bağımsız olduğundan yüksek serbestlik dereceli ve her türlü geometrik yapıya sahip manipülatöre uygulanabilir [2]. Bu yöntemin dezavantajı ise hesaplama yükünün fazla olmasıdır. Ayrıca jakobiyen matrisinin tersinin alınmasına dayandığından tekil noktalarda cevap vermemektedir. Bu noktalarda jakobiyen operatörünün tersi alınamaz. Bu problemin çözümü için de tekil noktalardan kaçınma, gürbüz ters kinematik çözümleme ve genişletilmiş Jakobiyen gibi yaklaşımlar sunulmuştur [3,4 ve 5].

Robot manipülatörlerinin kinematik probleminin çözümünde yapay zeka yöntemlerinin kullanılması yaygınlaşmaktadır [6]. Yapay zeka yöntemleri her türlü robota uygulanabildiği ve tekil noktalara takılmadığı için büyük avantaj sağlamaktadır. Bu yöntemlerden en sık kullanılan yöntem olan Yapay Sinir Ağları, hızlı bir çözüm sunmasına rağmen yerel minimum noktalarına takılabilmektedir. Bu

tezde kullanılan Destek Vektör Makinası yönteminde ise , yerel minimuma takılma problemi ortadan kaldırılarak global minimum noktasına ulaşılmaktadır [7].

Tez kapsamında, Gelecek Robotik Makine ve Tıp Teknolojileri Ar. Ge. Tic. Ltd. Şti. firmasının T.C. Bilim Sanayi ve Teknoloji Bakanlık' ı desteğiyle yürütmekte olduğu “Robotik Tekerlekli Sandalye” adlı projesinde kullanılan rehabilitasyon robot kolunun kinematik problemine çözüm aranmaktadır. 4 serbestlik derecesinden oluşan robot manipülatörü, tekerlik sandalyeye monte edilerek engelli insanlara yardımcı olmayı amaçlamaktadır.

Tezde, giriş bölümünde kullanılan robot manipülatörü, tezin amacı olan problem, çözümü ve kullanılabilirliği tartışılmıştır. İkinci bölümde ise, ele alınan 4 eksenli robot kolun özellikleri anlatılmış , ileri ve ters kinematik denklemleri analitik ve geometrik yöntemler ile çıkarılmıştır. Üçüncü bölümde, kinematik problemin çözümü için kullanılan Destek Vektör Makinası yöntemi sınıflandırma ve regresyon özellikleri ile birlikte anlatılmıştır. Dördüncü bölümde, analitik ve geometrik yöntemler ile bulunan robot kinematik denklem verileri eğitim verisi olarak kullanılarak, Destek Vektör Makinesi regresyonu ile ileri ve ters kinematik model tahmini yapılmıştır. Bunun yanında benzetim sonuçları ve tahmin performansları sunulmuştur. Beşinci bölümde ise, kinematik model tahmin performansları incelenmiş, yöntemin uygulanabilirliği ve devamlılığı tartışılmıştır.

2. 4 EKSENLİ ROBOT KOLU VE KİNEMATİK DENKLEMLERİ

2.1 4 Eksenli Robot Kolu

Bu tez kapsamında incelenen 4 eksenli robot kolun tasarımı, prototip imalatı ve kontrolü, Gelecek Robotik Makine ve Tıp Teknolojileri Ar. Ge. Tic. Ltd. Şti. firmasının T.C. Bilim Sanayi ve Teknoloji Bakanlığı' ı desteğiyle yürütmekte olduğu “Robotik Tekerlekli Sandalye” adlı projesinin ilk fazı kapsamında yapılmıştır.

Proje iki fazdan oluşmaktadır, proje tamamlandığında ortaya konacak olan sistemde elektrikli tekerlekli sandalye üzerine entegre edilmiş bir robot kol, kamera, lazer tarayıcı, uzaklık ve derinlik sensörleri, dokunmatik bir ekran ve gömülü bilgisayar yer alacaktır.

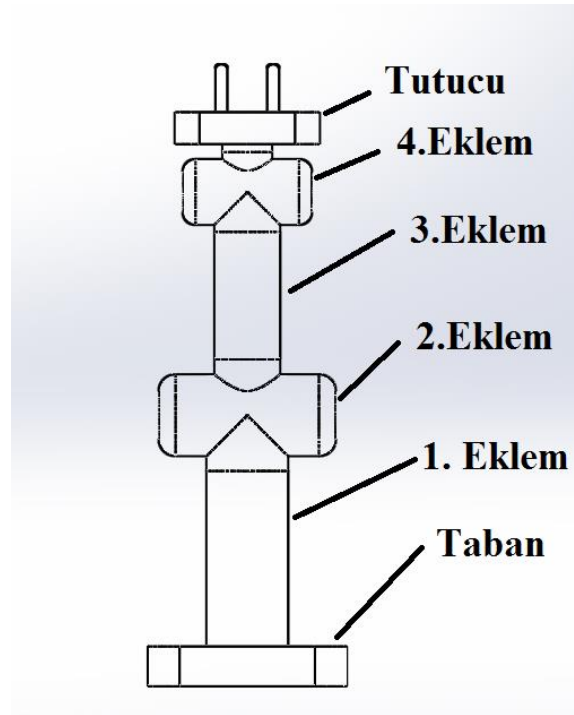
Proje tekerlekli sandalye kullanmak zorunda olan yaşlı veya engelli insanların günlük temel ihtiyaçlarını karşılamalarına yardımcı olması hedefiyle bir sistem geliştirmeyi amaçlamaktadır. Tez kapsamında yapılacak çalışma ile 18 aylık sürecek olan proje birinci fazı sonunda dokunmatik ekran üzerinden kontrol edilebilen bir robot kol prototipi imal etmeyi amaçlamaktadır. Ayrıca robot kolun geliştirilirken robot kolun kullanıcının ihtiyaçlarına göre eklem sayısının ve kinematik yapısının değiştirilebilir olması göz önünde bulundurulacaktır.

Proje sürecinde farklı disiplinlerden üç mühendis yer almıştır. Literatür taraması ile başlanıp, bilgisayar ortamında gerçekleştirilen modelleme ve benzetim, bilgisayar ortamında üç boyutlu tasarım ve prototipin imal edilmesi çalışmalarına paralel olarak kontrol sisteminin geliştirilmesi başlığı altında yazılım geliştirme, PLC programlama ve arayüz geliştirme çalışmaları yapılmıştır. Bu tez kapsamında ise, tasarımı proje ekibi üyeleri tarafından yapılan rehabilitasyon robot kolunun öncelikle ileri ve ters kinematik denklemleri çıkarılmıştır. Daha sonra da destek vektör makinası algoritması ile bu kinematik modele uygun bir çözüm aranmıştır.

2.1.1 Robot kol özellikleri

Literatürdeki çalışmaları ve ürünleri inceledikten sonra geliştirilecek manipülatörün taslak tasarımlarını yaparak çalışmalara devam edilmiştir. Taslak tasarım yapılırken literatür araştırmaları ve proje gereksinimleri göz önünde bulundurulmuştur.

Geliştirilen manipülatör 4 serbestlik dereceli olup paralel 2 parmaklı bir tutucuya sahiptir. Maksimum erişim mesafesi 400-500 mm olup, taşıma kapasitesi nominal hızda 500 gramdır. Proje gereksinimleri gereği manipülatör kinematiği kolayca değiştirilebilsin diye eklemler modüler yapıda oluşturulacaktır. Proje bütçesinin eklemli yapıya sahip parmaklı bir tutucu satın almaya yetersiz olduğundan dolayı projenin birinci fazında paralel 2 parmaklı tutucu kullanılmıştır. Solidworks yazılımı kullanılarak Şekil 2.5' deki niteliksel taslak tasarım geliştirilmiştir.



Şekil 2.1 : Manipülatörün taslak tasarımı.

Geliştirilecek robotik kol üzerindeki birbirini takip eden her bir eklem dönüş yönleri birbirine dik olarak monte edilmiştir. Geliştirilecek eklemleri konstrüktif yapıları açısından yatay ve dikey olarak 2 farklı tipte sınıflandırılabilir; ayrıca eklemleri güç açısından da 2 farklı tipte gruplandırılabilir.

2.2 4 Eksenli Robot Kolu Kinematik Denklemleri

2.2.1 İleri kinematik model

Manipülâtörün ileri kinematik çözümünü bulmak için Denavit-Hartenberg (D-H) yöntemi kullanılmıştır.

D-H yönteminde manipülâtörün kinematiği her bir eklem için 4 adet parametrenin tespit edilmesi ile çözümlenmektedir. Her bir ekleme ait merkez noktası ve koordinat takımı uygun şekilde yerleştirildikten sonra D-H parametreleri belirlenir [6].

Bu parametreler:

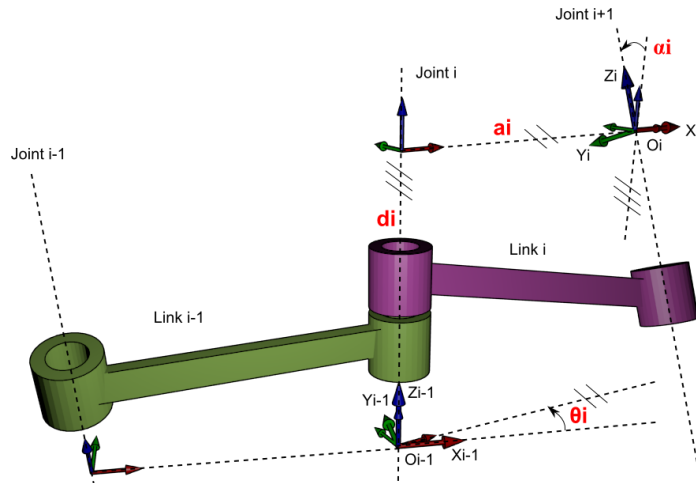
a_i : x_i yönünde z_{i-1} ile x_i kesişiminin O_i ' ye olan uzaklığı.

α_i : x_i yönünde z_{i-1} ile z_i arasındaki açı.

d_i : z_{i-1} yönünde z_{i-1} ile x_i kesişiminin O_{i-1} ' e olan uzaklığı.

θ_i : z_{i-1} yönünde x_{i-1} ile x_i arasındaki açı.

şeklinde tanımlanmaktadır.



Şekil 2.2 : D-H yöntemi için koordinat eksenlerinin gösterimi.

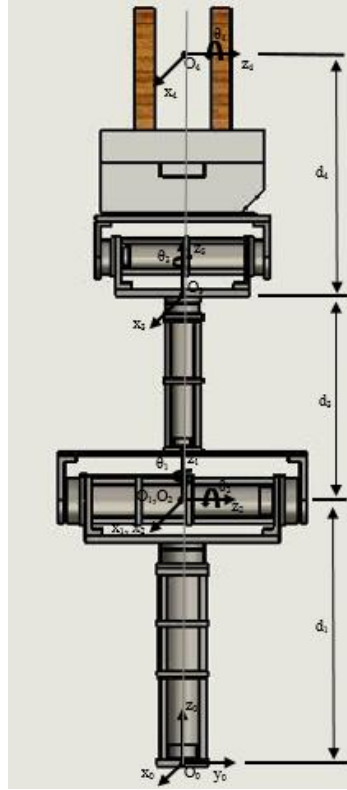
Parametreler tespit edildikten sonra D-H tablosu Çizelge 2.1' de olduğu gibi hazırlanmaktadır.

Çizelge 2.1 : D-H parametreleri tablosu.

| Eklem No | a_i | α_i | d_i | θ_i |
|----------|-------|------------|-------|------------|
| 1 | a_1 | α_1 | d_1 | θ_1 |
| 2 | a_2 | α_2 | d_2 | θ_2 |
| 3 | a_3 | α_3 | d_3 | θ_3 |
| ... | ... | ... | ... | ... |
| n | a_n | α_n | d_n | θ_n |

Geliştirmekte olduğumuz robot kolun eklemlerinin merkez noktaları ve koordinat eksenlerinin yerleştirilmesi ile modelleme çalışmasına başlıyoruz. Her bir ekleme ait

merkez noktaları ve koordinat eksenleri robot kol üzerinde Şekil 2.6' da gösterilmiştir.



Şekil 2.3: Koordinat eksenlerinin robot kol üzerinde gösterimi.

Yerküreyi temsilen $O_0 (x_0, y_0, z_0)$ merkez noktası ve kordinat sistemi yerleştirilmiştir. Birinci ve ikinci eklemin dönüş eksenleri kesişmektedir; bu nedenle birinci eklemi temsilen $O_1 (x_1, y_1, z_1)$ ve ikinci eklemi temsilen $O_2 (x_2, y_2, z_2)$ merkez noktaları ve koordinat eksenleri çakışık olarak yerleştirilmiştir. Üçüncü eklemi temsilen $O_3 (x_3, y_3, z_3)$ merkez noktası ve koordinat sistemi ile dördüncü eklemi temsilen $O_4 (x_4, y_4, z_4)$ merkez noktası ve koordinat sistemi gösterilmiştir.

Robot kol D-H parametreleri tablosu Çizelge 2.2' de yer aldığı şekilde oluşmuştur.

Çizelge 2.2 : Robot kol D-H parametreleri tablosu.

| Eklem No | a_i | α_i | d_i | θ_i |
|----------|-------|-------------|-------|------------|
| 1 | 0 | -90° | d_1 | θ_1 |
| 2 | 0 | 90° | d_2 | θ_2 |
| 3 | 0 | -90° | 0 | θ_3 |
| 4 | 0 | 90° | d_4 | θ_4 |

D-H tablosunun hazırlanması ile birlikte komşu iki eklem arasındaki ilişkiyi gösteren transformasyon matrisleri aşağıdaki şekilde hesaplanmaktadır.

$${}^{n-1}T_n = Trans_{z_{n-1}}(d_n) Rot_{z_{n-1}}(\theta_n) Trans_{x_n}(a_n) Rot_{x_n}(\alpha_n)$$

$$\begin{aligned} Trans_{z_{n-1}}(d_n) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Rot_{z_{n-1}}(\theta_n) &= \begin{bmatrix} c_{\theta_n} & -s_{\theta_n} & 0 & 0 \\ s_{\theta_n} & c_{\theta_n} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Trans_{x_n}(a_n) &= \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Rot_{x_n}(\alpha_n) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_n} & -s_{\alpha_n} & 0 \\ 0 & s_{\alpha_n} & c_{\alpha_n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^{n-1}T_n &= \begin{bmatrix} c_{\theta_n} & -s_{\theta_n}c_{\alpha_n} & s_{\theta_n}s_{\alpha_n} & a_nc_{\theta_n} \\ s_{\theta_n} & c_{\theta_n}c_{\alpha_n} & -c_{\theta_n}s_{\alpha_n} & a_ns_{\theta_n} \\ 0 & s_{\alpha_n} & c_{\alpha_n} & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.1)$$

Uç eyleyicisinin transformasyon matrisini bulabilmek için aşağıdaki hesaplama yapılır.

$${}^0T_n = {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \quad (2.2)$$

Transformasyon matrisi içerisinde rotasyon ve oryantasyon matrislerini barındırır. Rotasyon matrisi transformasyon matrisinin ilk 3 satırının ilk 3 teriminden oluşmaktadır.

$${}^{n-1}R_n = \begin{bmatrix} c_{\theta_n} & -s_{\theta_n}c_{\alpha_n} & s_{\theta_n}s_{\alpha_n} \\ s_{\theta_n} & c_{\theta_n}c_{\alpha_n} & -c_{\theta_n}s_{\alpha_n} \\ 0 & s_{\alpha_n} & c_{\alpha_n} \end{bmatrix} \quad (2.3)$$

Oryantasyon matrisi ise transformasyon matrisinin ilk 3 satırın son elemanlarından oluşmaktadır.

$${}^{n-1}O_n = \begin{bmatrix} a_n c_{\theta_n} \\ a_n s_{\theta_n} \\ d_n \end{bmatrix} \quad (2.4)$$

Manipülâtörün uç eyleyicisinin 3 eksenli uzayda çizgisel pozisyonunu 0O_n oryantasyon matrisinden direk tespit edebilmekteyiz, ancak uç eyleyicinin açısâl pozisyonunu bulabilmek için rotasyon matrisini kullanarak bazı hesaplamalar yapmamız gerekmektedir.

Bu açılar uç eyleyicisinin x, y ve z eksenleri etrafında yaptığı yunuslama (ϕ) , sapma (ψ) ve yuvarlanma (ϑ) açılarıdır.

$$R_{XYZ} = R_{z,\phi} \cdot R_{y,\vartheta} \cdot R_{x,\psi}$$

$$R_{XYZ} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\vartheta & 0 & s_\vartheta \\ 0 & 1 & 0 \\ -s_\vartheta & 0 & c_\vartheta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \quad (2.5)$$

$$R_{XYZ} = \begin{bmatrix} c_\phi c_\vartheta & -s_\phi c_\psi + c_\phi s_\vartheta s_\psi & s_\phi s_\psi + c_\phi s_\vartheta c_\psi \\ s_\phi c_\vartheta & c_\phi c_\psi + s_\phi s_\vartheta s_\psi & -c_\phi s_\psi + s_\phi s_\vartheta c_\psi \\ -s_\vartheta & c_\vartheta s_\psi & c_\vartheta c_\psi \end{bmatrix}$$

Rotasyon matrisleri eşitlenerek

0R_n ile R_{XYZ} matrisleri eşitlenerek yunuslama, yalpa ve yuvarlanma açıları aşağıdaki formüller aracılığı ile hesaplanır. [4]

$${}^0R_n = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{22} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.6)$$

$$\vartheta = \text{atan2}(-r_{31}, \sqrt{r_{11}r_{11} + r_{21}r_{21}})$$

$\vartheta \neq \pm\pi/2$ olmamak koşuluyla;

$$\phi = \text{atan2}\left(\frac{r_{21}}{c_\vartheta}, \frac{r_{11}}{c_\vartheta}\right)$$

$$\psi = \text{atan2}\left(\frac{r_{32}}{c_\vartheta}, \frac{r_{33}}{c_\vartheta}\right) \quad (2.7)$$

2.2.2 Ters kinematik model

Bir robot kolunun ters kinematik modeli iki farklı yaklaşımla hesaplanabilir. İlk yaklaşım , eklemler arasındaki geometrik ilişkiler kullanılarak ulaşılan analitik

çözümdür. Eklem açısı artıkça geometrik ilişkiler karmaşık hale geleceğinden serbestlik derecesi fazla olan robot kollarında uygulanması zordur.

İkinci yaklaşım ise, hız kinematiğinin kullanılarak uygulanan numerik çözümdür. Bu çözümde jakobiyen matrisinin tersinin hesaplanması gereklidir. Hız değeri entegre edilerek konum değerine ulaşılır. Bu çözümün negatif tarafı ise, çalışma uzayındaki tekil noktalarda algoritmanın cevap vermemesidir.

2.2.2.1 Geometrik yaklaşım

4 eksenli robot kolun ters kinematik denklemlerinin geometrik yaklaşımla çıkarılabilmesi için benzer yapıdaki 7 eksenli bir robot kolunun geometrik çözümlemesinden yararlanılmıştır. İlk 4 eksen esas alınmıştır. Humanoid olan kol , rehabilitasyon robot koluyla aynı eklem dizilimine sahiptir [7].Çizgisel konumlar için geometrik çözümü araştırılırsa;

$$\begin{aligned}
 dp &= [px \ py \ pz] \\
 \theta 1 &= atan2(py, px) \\
 \theta 2 &= acos\left(\frac{pz}{d}\right) \\
 Va &= [px \ py \ pz] \\
 A' &= \begin{bmatrix} \cos(\theta 1) \cdot \cos(\theta 2) & -\sin(\theta 1) \\ \sin(\theta 1) \cdot \cos(\theta 2) & \cos(\theta 1) \\ -\sin(\theta 2) & 0.002 \end{bmatrix} \\
 A &= pinv(A') \\
 B &= A \cdot Va \\
 B &= \begin{bmatrix} \cos(\theta 3) \\ \sin(\theta 3) \end{bmatrix} \\
 \theta 3 &= atan2\left(\frac{B(2)}{B(1)}\right) \\
 \alpha 2 &= acos\left(\frac{d^2 + L2^2 - L1^2}{2d \cdot L2}\right)
 \end{aligned} \tag{2.8}$$

$L1 \approx L2$ alındığında;

$$\begin{aligned}
 \alpha 2 &= acos\left(\frac{d}{2L2}\right) \\
 \alpha 1 &= asin\left(L2 \cdot \frac{\sin(\alpha 2)}{L1}\right) \\
 \alpha 1 &= \alpha 2
 \end{aligned} \tag{2.9}$$

d: bilek noktası ile orijin arasındaki uzaklık

$$\begin{aligned}
\theta u &= \pi - \left(\frac{\pi}{2} - \alpha 2 \right) - \angle BAU \\
\theta l &= \pi - \left(\frac{\pi}{2} - \alpha 1 \right) - \angle OAL \\
\angle BAU &= \frac{\pi}{2} \\
\angle OAL &= \frac{\pi}{2} \\
\theta 4 &= \theta u + \theta l \\
\theta 4 &= \alpha 1 + \alpha 2
\end{aligned} \tag{2.10}$$

2.2.2.2 Ters jakobiye yöntemi

2.2.2.2.1 Jakobiye matrisinin hesaplanması

Rotasyonel eklem için;

$$\begin{aligned}
J_{v_i} &= z_{i-1} \times (o_n - o_{i-1}) \\
J_{w_i} &= z_{i-1} \\
J_i &= \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}
\end{aligned} \tag{2.11}$$

Doğrusal eklem için;

$$\begin{aligned}
J_{v_i} &= z_{i-1} \\
J_{w_i} &= 0 \\
J_i &= \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix}
\end{aligned} \tag{2.12}$$

Manipülâtörün jakobiye matrisini aşağıdaki şekilde düzenlenerek elde edilir. [6]

$$J = [J_1 \quad J_2 \quad \dots \quad J_n] \tag{2.13}$$

2.2.2.2.2 Jakobiye matrisinin tersinin hesaplanması

Jakobiye matrisi kare matris olmadığından dolayı tersini hesaplamak için Psuedo-Ters yöntemini kullanılmaktadır. Jakobiye matrisinin Psuedo-Ters'i (J^+) aşağıdaki şekilde hesaplanmaktadır [6].

$$J^+ = J^T (JJ^T)^{-1} \tag{2.14}$$

2.3 Çalışma Uzayının Belirlenmesi

Rehabilitasyon robot kolunun eklem ve hareket alanı sınırlarının bilinmesi gereklidir.

Bunun içinde robot kolunun çalışma uzayı belirlenmelidir.

Çalışma uzayı belirlerken eklemlerin alabileceği maksimum ve minimum açı değerleri kullanılmaktadır.

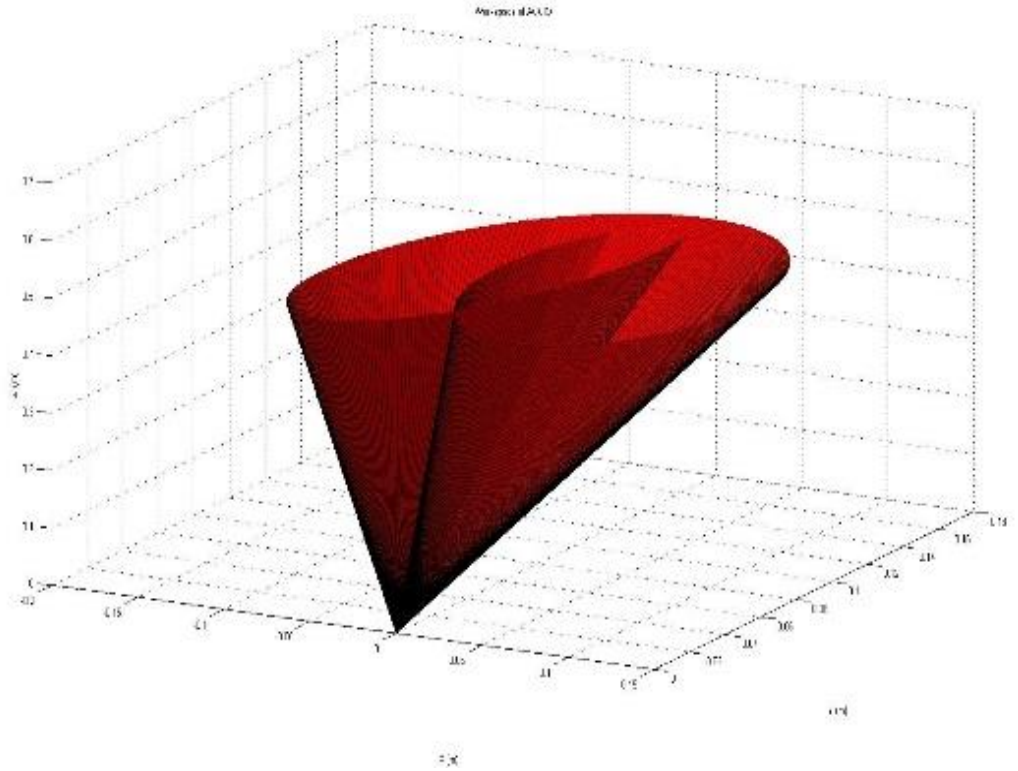
Eklem açıları için sınırlar;

$$\begin{aligned} -120^\circ < \theta_1 < 120^\circ \\ -60^\circ < \theta_2 < 60^\circ \\ -120^\circ < \theta_3 < 120^\circ \\ -60^\circ < \theta_4 < 60^\circ \end{aligned} \quad (2.15)$$

şeklindedir.

Link sınırları yatay ve dikey eklemlerin mekanik hareket sınırlarına göre belirlenmiştir.

Sınırlar arasında 0.50 radyan aralıklarla eklem açıları değiştirilerek ileri kinematik sonuçları alınmıştır. Elde edilen uç işlevci pozisyon değerleriyle çalışma uzayı Şekil 2.3' de görüldüğü üzere çizdirilmiştir.



Şekil 2.4 : Robot kolu çalışma uzayı.

3. DESTEK VEKTÖR MAKİNELERİ

3.1 Giriş

Destek Vektör Makineleri, değişkenlerin arasındaki örüntünün bilinmediği data setlerinin sınıflandırılması problemi için önerilen bir makine öğrenmesi yöntemidir. Destek Vektör Makinelerinin temelini oluşturan çekirdek kavramının temelleri 1909 yıllarına kadar dayandırılmaktadır. İlk olarak 1995 yılında Rus Matemeatikçi Vladimir Vapnik tarafından, "COLT" isimli konferansta önerilmiştir. DVM'lerin yapay sinir ağlarına göre üstünlükleri, optimizasyon probleminin bir karesel programlama problemine dönüştürülüp, global çözümün garanti edilmesidir. Yapay sinir ağlarında karşılaşılan yerel minimum noktalara takılma problemi, optimizasyon probleminin konveks bir yapıya bürünmesiyle aşılmıştır. DVM algoritmaları bu özelliği sayesinde optimizasyon problemlerinin çözümünde dikkate değer bir iyileşme sağlamaktadır.

Destek vektör makineleri yöntemine dayalı uygulamalar sağlıktan tarıma kadar çeşitli alanlarda kullanılmaktadır. Tıp alanındaki uygulamalara bir örnek olarak göğüs kanseri tanısında kullanılan mikrodalga temelli laboratuvar cihazları verilebilir. Bu cihazlarda görüntü tanılama için DVM tabanlı yöntemler kullanılmakta ve gürültü en aza indirilmektedir [1]. Yine kanser araştırmalarında kanserli dokuların sağlıklı dokulardan ayırt edilmesi ve sınıflandırılması için DVM yöntemine başvurulmaktadır [2]. Karmaşık nonlinear sistemlerin modellenmesinde kullanılan “soft sensor” algoritmaları DVM yöntemi kullanılarak geliştirilebilmektedir. Bu uygulamada DVM yönteminin kara-kutu modelleri tanımadaki başarısı etkili olmaktadır [3]. Bunun dışında ses tanıma için ; GMM(Gaussian Mixture Method) modeli ile DVM sınıflandırıcısı beraber kullanılmaktadır. Bu çalışmada aynı zamanda farklı iki çekirdek fonksiyonu önerilmiştir [4].

Kontrol mühendisliği alanında destek vektör makinelerinin kullanımı oldukça yaygındır. Model öngörücülü kontrol yöntemlerinde , model tanılama için DVM regresyona başvurulmuştur. Uçuş kontrolü [5] ve tank sistemlerinin modellenmesi

[6]bu kullanıma örnek olarak verilebilir. Farklı bir kullanım alanı olarak; Karadeniz'deki söğüt ağaç türlerinin ayrıştırılabilmesi için DVM tabanlı model öngörücülü kontrole başvurulmuştur [7]. Ayrıca kazan sistemlerinin farklı yükler altındaki gürbüz (robust) kontrolünde , model tanılama için en küçük kareler DVM regresyon kullanılmıştır [8]. En küçük kareler DVM, aynı zamanda nonlinear özellikteki kontrol valflerinin modellenmesinde kullanılmıştır [9].Sabit mıknatıslı senkron motorların iç model kontrolör ile kontrolünde DVM'den yararlanılmıştır [10]. Bu alanların dışında; otomotivde aracın elektrik kontrol sisteminin DVM ile modellenmesi yeni bir yöntem olarak ortaya konulmuştur [11].

Robotikteki ileri ve ters kinematik, yörünge planlama ve konumlandırma gibi genel problemlere DVM regresyonu ile çeşitli çözümler sunulmuştur. Hareket oldukça karmaşık olan ve nonlinear özellik taşıyan paralel robotların ileri kinematik modelinin çıkarılması bu çözümlere bir örnektir [12]. Yedi ve daha fazla eklem içeren “redundant” robotların ters kinematik denklemlerinin oluşturulması analitik yöntemlerle yine oldukça karmaşık olmaktadır. Aynı zamanda tekil noktalar problem oluşturmaktadır. Ters kinematik probleminin robotun DVM regresyonu ile modellenmesi yardımıyla çözülmesi kesin ve etkin bir çözüm sunmaktadır [13]. Bu yöntem endüstriyel robotlarda da denenmiş ve diğer yapay sinir ağı yöntemlerine göre avantajlı olduğu görülmüştür [14]. Yörünge planlamada en büyük problem olan çalışma uzayındaki engellerden kaçınma, bulanık DVM tabanlı bir algoritma ile çözüme kavuşturulmuştur [15]. Robotik uygulamalarda önemli bir yeri olan görüntü işleme, bazı uygulamalarda DVM yardımıyla yapılmıştır. Mobil robotların kendi kendine konumlandırılmasında, görüntü tanımlama hatalarının giderilmesi için DVM'ye başvurulmuştur [16]. Ayrıca, tarımda kullanılan hasat robotlarında; robotun meyve ve dalları ayırması için görüntü işleme yöntemleri kullanılmaktadır. Görüntü tanımlama işlemi için, çok sınıflı DVM sınıflandırıcıları ile ayırım yapılmaktadır [17].

Destek Vektör Makinesi yöntemi, sınıflandırma ve regresyon olmak üzere iki ayrı alanda kullanılmaktadır. Bu yöntem istatistiksel öğrenme teorisine ve yapısal risk minimizasyonuna dayanmaktadır.

Destek Vektör Makineleri yönteminde parametreler belli değildir. Girdi ve çıktıların dağılımı hakkında herhangi bir ön bilgi yoktur. Eğitim setlerinde girdi ve çıktı değerleri eşlenir. Yeni data setlerini sınıflandırmak için bu eşleşmeler yardımıyla karar fonksiyonları elde edilir.

3.2 Tarihi Gelişim

Yapay Sinir Ağlarına göre temelleri 20. yüzyılın başlarına dayanan DVM'nin tarihsel gelişim süreci aşağıdaki gibi verilmiştir.

1936 yılında, R. A. Fisher örüntü tanıma için ilk algoritmayı önermiştir. (Fisher 1936). Aronszajn (1950) yılında "Kernel" teorisini tanıtmıştır. 1957 yılında, Frank Rosenblatt lineer sınıflandırma için "perceptron" teorisini ortaya atmıştır. Vapnik and Lerner 1963 yılında "Genelleştirilmiş Portre Algoritması"nı yayınlamıştır. Aizerman, Braverman and Rozonoer 1964 yılında kernel çarpımının geometrik açılımını ortaya koymuştur.

Vapnik and Chervonenkis 1964 yılında "Genelleştirilmiş Portre Algoritması"nı geliştirmişlerdir. Cover 1965 yılında büyük genlikli hiperdüzlemleri araştırmıştır. Mangasarian 1965 yılında örüntü tanımlama için benzer algoritmaları kullanmıştır. Smith 1968 yılında ek değerler yardımıyla gürültü ve doğrusal olarak ayrılamama problemine çözüm getirmiştir. Duda ve Hart 1973 yılında girdi uzayındaki geniş marjlinli hiperdüzlemleri incelemiştir. İstatiksel Öğrenme Teorisi alanındaki çalışmalar 1974 yılında Rusya'da Vapnik ve Chervonenkis tarafından başlatılmıştır.

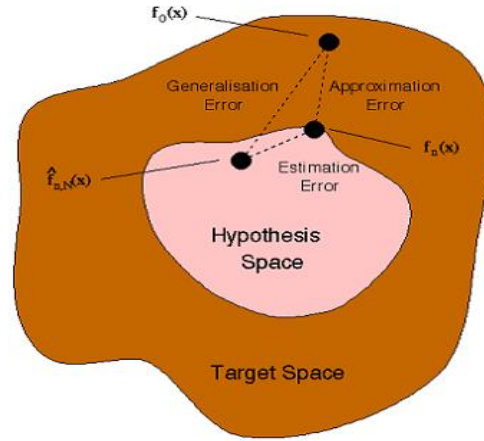
Destek Vektör Makinesi çalışmaları İstatiksel Öğrenme Teorisinin 1979 yılından Vapnik tarafından geliştirilmesiyle başlamıştır. Vapnik ve Chervonenkis 1979 yılında kitaplarının Almanca çevirisini yayınlamışlardır. Vapnik 1982 yılında kitabın İngilizce çevirisini yayınlamıştır. Hassoun 1986 yılında bu konuda bir doktora tezi yapmıştır. Poggio, Girosi ve Wahba 1990 yılında çekirdek fonksiyonun kullanımını incelemiştir. Bennett ve Mangasarian 1992 yılında 1968 yılındaki ek değerlerle ilgili çalışmayı geliştirmiştir. Destek Vektör Makineleri son halini 1992 yılında COLT konferansında Boser, Guyon ve Vapnik tarafından yayınlanan bildiri ile almıştır. 1995 yılında Cortes ve Vapnik soft marj sınıflandırıcıları; yine aynı yılda Vapnik buna bağlı olarak regresyon konusu hakkındaki bildirilerin yayınlamışlardır. Shawe-Taylor ve Cristianini 2000 yılında sınıflandırma ve regresyon için soft marjin algoritmasının istatiksel sınırlarını araştırmıştır.

3.3 İstatiksel Öğrenme

İstatiksel öğrenme teorisi 1960'larda ortaya atılmasına rağmen, fonksiyon kestirimi için kullanılması 1990'ların ortalarını bulmuştur. Bu tarihten itibaren yeni öğrenme algoritmalarının geliştirilmesiyle istatiksel öğrenme, sadece teorik analiz için değil

çokboyutlu fonksiyonların kestirimi için üretilen pratik algoritmaların geliştirilmesi için de kullanılmaya başlanmıştır.

Öğrenme verilerinden oluşan hipotez uzayı ve test verilerinden oluşan hedef uzayı istatistiksel öğrenme teorisinin tanımlanmasına yardımcı olmaktadır. Teorinin amacı, hipotez uzayı içinden seçilecek modelin, hedef uzayındaki verileri en az hata ile kapsamamasını sağlamaktır [18].



Şekil 3.1 : İstatistiksel öğrenme.

- Yaklaşım hatası: Hedef uzayı ile hipotez uzayı arasındaki hatadır. Büyük yaklaşım hataları yanlış modellemekten kaynaklanır.
- Tahmin Hatası: Hipotez uzayı içindeki hatadır. Test datası için uygun olmayan bir öğrenme metodu kullanıldığında oluşur.

Genelleme hatasına aşağıdaki şekilde ulaşılabilir.

$$R[f] = \int_{X \times Y} L(y, f(x)) P(x, y) dx dy \quad (3.1)$$

Empirik Risk Minimizasyonuna göre yaklaşık bir ifade elde edilebilir.

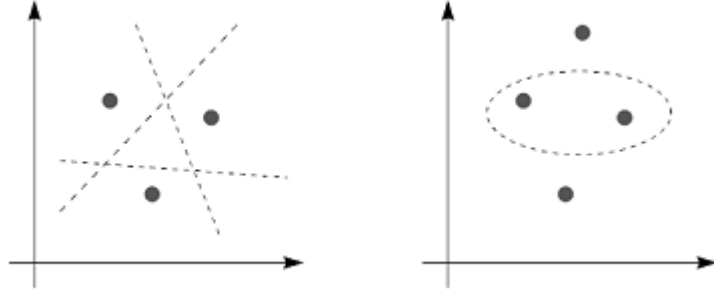
$$R_{emp} [f] = \frac{1}{l} \sum_{i=1}^l L(y^i, f(x^i)) \quad (3.2)$$

3.3.1 Vapnik-Chervonenkis teorisi

Klasik istatistik doğru modeli bildiğini varsayıp parametreleri ararken, istatistiksel öğrenme teorisi modelin formunu bilmediğini kabul edip , doğru olabilecek modeller arasında en uygun modeli seçer.

3.3.2 VC boyutu

Fonksiyonlar kümesinin sahip olduğu bir özelliktir. Kümenin kapasitesi hakkında bilgi verir.



Şekil 3.2: VC Boyutları

2 boyutlu bir uazayda, hatasız bir şekilde mümkün olan tüm yollarla sadece 3 nokta parçalara ayrılabilirdiği için VC boyutu 'tür. N boyutlu bir uzayın VC boyutu N+1'dir.

3.3.3.Yapısal Risk Minimizasyonu :

S_h VC boyutu h olan bir hipotez uzayı olsun.

Yapısal risk minimizasyonu aşağıdaki problemle tanımlanır.

$$\min_{S_h} R_{emp}[f] + \sqrt{\frac{h \ln\left(\frac{2}{h}+1\right) - \ln\left(\frac{\delta}{4}\right)}{l}} \quad (3.3)$$

Yapısal Risk Minimizasyonu, gerçek risk için üst sınırı minimize edecek fonksiyonu araştırmaktadır. Genelleme hatası için bir açıklamadır. Genelleme hatası; deneysel hata, örnek sayısı ve kapasiteden etkilenmektedir [19].

3.4.Destek Vektör Sınıflandırması

3.4.1 Lineer olarak ayrılabilen örüntüler

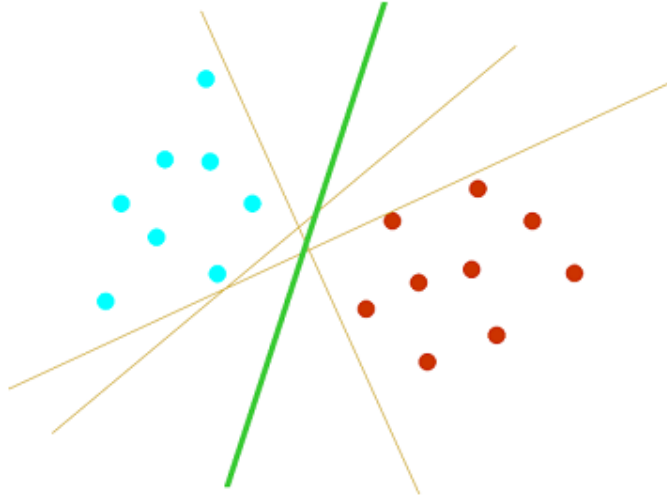
Eğitim verileri D olarak verilmiş olsun.

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\} \quad y \in (-1, 1) \quad l : \text{örnek sayısı}$$

Bu verilerin ayrılmasını sağlayacak sonsuz tane ayırıcı düzlem bulunmaktadır. DVM'de amaç en iyi sınıflandırıcı düzlemi elde etmektir. Bu veri çiftlerinin lineer olarak ayrılabilir optimum hiperdüzlemin denklemi aşağıdaki gibidir.

$$\langle w, x \rangle + b = 0 \quad \langle \rangle : \text{nokta çarpımı} \quad (3.4)$$

w : ağırlık vektörü: bias



Şekil 3.3: Optimal hiperdüzlem.

verilmekte ve bu düzlem EMS (En Büyük Marjınlı Sınıflandırıcı) olarak adlandırılmaktadır.

Optimal hiperdüzlemin en yakın veri çiftine olan uzaklığı maksimum olmalıdır. Aynı zamanda bu hiperdüzlemin aşağıdaki eşitsizlikleri sağlaması gerekir.

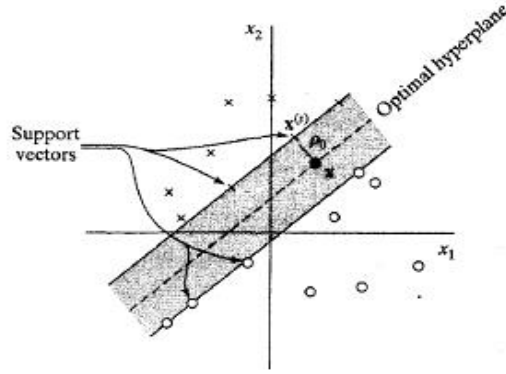
$$\begin{aligned} < w, x^i > + b \geq 1 \quad y^i = 1 \\ < w, x^i > + b \leq -1 \quad y^i = -1 \end{aligned} \quad (3.5)$$

Bu iki eşitsizlik tek bir eşitsizlikte birleştirilebilir. Böylece optimizasyon probleminin kısıtı elde edilir.

$$y^i \cdot [< w, x^i > + b] \geq 1 \quad i = 1, \dots, l \quad (3.6)$$

Problemin amacı, ağırlık vektörü ve bias teriminin optimum değerinin bulmaktır.

Marjin, bir x vektörünün optimal hiperdüzleme olan uzaklığı olarak tanımlanabilir.



Şekil 3.4: Destek vektörleri.

$$d(w, b : x) = \frac{|\langle w, x^i \rangle + b|}{\|w\|}$$

$$p(w, b) = \min_{x^i, y^i = -1} d(w, b : x^i) + \min_{x^i, y^i = 1} d(w, b : x^i) \quad (3.7)$$

$$p(w, b) = \frac{2}{\|w\|}$$

Yeni gelen verilerin sınıflama hatasının en az olması için marjın maksimum olmalıdır. Marjini maksimize etmek için, ağırlık vektörünün normu minimize edilmelidir.

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (3.8)$$

Minimizasyon probleminin amaç fonksiyonunda b bias terimi bulunmamaktadır [20].

3.4.1.1 Quadratic optimizasyon – Lagrange çarpanları Yöntemi

Birincil haldeki minimizasyon problemi, amaç fonksiyonu ve kısıtlar Lagrange çarpanları yöntemiyle birleştirilerek Lagrange fonksiyonu elde edilir.

$$\text{Amaç fonksiyonu: } \Phi(w) = \frac{1}{2} \|w\|^2$$

$$\text{Kısıt : } y^i \cdot [\langle w, x^i \rangle + b] \geq 1 \quad i = 1, \dots, l$$

$$\text{Lagrange Fonksiyonu: } \Phi(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i [y^i \cdot [\langle w, x^i \rangle + b] - 1]$$

Dual problemin çözümü veri setinin eğer noktasının bulunmasını sağlar. Eğer noktasında ağırlık vektörü (w) ve bias terimi (b) minimum, lagrange çarpanları (α) maksimum değerlerini alır.

$$\max_{\alpha} W(\alpha) = \max_{\alpha} (\min_{w,b} \Phi(w, b, \alpha)) \quad (3.9)$$

Lagrange fonksiyonunun w ve b 'ye göre kısmi türevler alınarak Karush-Kuhn-Tucker (KKT) koşulları elde edilir.

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 & \quad \rightarrow \quad \sum_{i=1}^l \alpha_i y_i = 0 \\ \frac{\partial \Phi}{\partial w} = 0 & \quad \rightarrow \quad w = \sum_{i=1}^l \alpha_i y_i x_i \end{aligned} \quad (3.10)$$

Koşullar Lagrange fonksiyonunda yerine konulduğunda dual problemin son haline ulaşılır.

$$\Phi(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i [y^i \cdot [w, x^i]] - b \cdot \sum_{i=1}^l \alpha_i [y^i] + \sum_{i=1}^l \alpha_i \quad (3.11)$$

$$\|w\|^2 = \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j < x_i, x_j > \quad (3.12)$$

Yeni amaç fonksiyonu: $Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j < x_i, x_j >$

Kısıtlar: $\sum_{i=1}^l \alpha_i y_i = 0$

$$\alpha_i \geq 0$$

Optimal hiperdüzlemi belirlemek için verilen model çözülerek optimum α değerleri bulunur. Lagrange çarpanları sıfırdan farklı olan eğitim verileri “destek vektörleri” olarak adlandırılır. Böylece, optimal hiperdüzlem, destek vektörler ve bunlara karşılık düşen Lagrange çarpanlar cinsinden aşağıdaki gibi ifade edilebilir.

$$f(x) = \sum_{i=1}^l \alpha_i y_i < x_i, x > + b \quad (3.13)$$

Bulunan α değerleri yardımıyla optimal hiperdüzlemin ağırlık vektörü ve bias terimi hesaplanır.

$$w^* = \sum_{i=1}^l \alpha_i y_i x_i$$

$$b^* = \frac{-1}{2} < w^*, x_r + x_s > \quad (3.14)$$

$$y_r = -1, y_s = 1$$

arametrelere göre sınıflandırmada kullanılan karar fonksiyonu belirlenir.

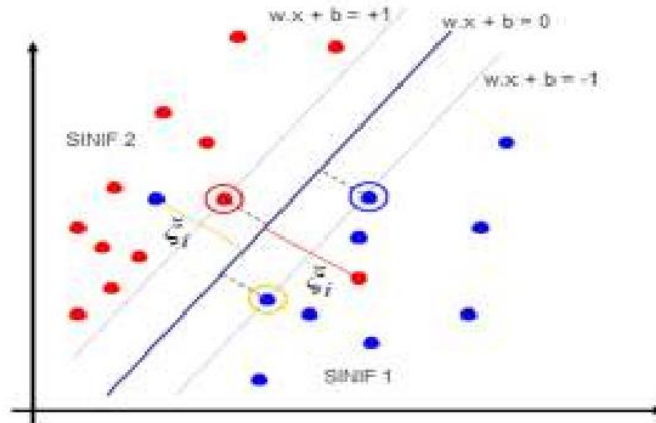
$$f(x) = \text{sgn}(< w^*, x > + b) \quad (3.15)$$

Bu sınıflandırıcıya alternatif olarak, gerçek değerler veren daha kullanışlı bir algoritma sunulmuştur. (Gunn, 1998).

$$f(x) = h . (< w^*, x > + b) \quad h(z) = \begin{cases} -1 & z < -1 \\ z & -1 < z < 1 \\ 1 & z > 1 \end{cases} \quad (3.16)$$

3.4.2 Genel haliyle sınıflandırma

Veri setinin belirli bir hata ile doğrusal olarak ayrıldığı durumlarda, sınıflandırma hatalarını gidermek için amaç fonksiyonuna bir fonksiyon daha eklenir. Bu değişken de minimizasyon probleminde ek bir değişken oluşmasına neden olur [21].



Şekil 3.5: Genelleştirilmiş veri.

$\xi_i \geq 0$: sınıflandırma hatası

$0 \leq \xi \leq 1$ durumu verinin doğru sınıfta olduğunu gösterir.

$\xi > 1$ durumu verinin yanlış sınıfta olduğunu gösterir.

Lineer durumdaki kısıtlar yeni değişkene uyarlanabilir.

$$\begin{aligned} < w, x^i > + b \geq 1 - \xi_i \quad y^i = 1 \\ < w, x^i > + b \leq -1 + \xi_i \quad y^i = -1 \\ y^i \cdot [< w, x^i > + b] \geq 1 - \xi_i \end{aligned} \quad (3.17)$$

Amaç fonksiyonu: $\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i$

Problemin çözümü için amaç fonksiyonu ve kısıtlar lagrange çarpanları yöntemiyle birleştirilip bir Lagrange fonksiyonu elde edilir. Bu Lagrange fonksiyonu yardımıyla dual forma geçiş sağlanır.

$$\begin{aligned} \Phi(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i [y^i \cdot [w, x^i] + b] - \\ 1 + \xi_i] - \sum_{i=1}^l \beta_i \xi_i \end{aligned} \quad (3.18)$$

Dual Problem : $\max_{\alpha} W(\alpha, \beta) = \max_{\alpha, \beta} (\min_{w, b, \xi} \Phi(w, b, \xi, \alpha, \beta))$

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^l \alpha_i y_i &= 0 \\ \frac{\partial \Phi}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^l \alpha_i y_i x_i \\ \frac{\partial \Phi}{\partial \xi} = 0 \quad \rightarrow \quad \alpha_i + \beta_i &= C \end{aligned} \quad (3.19)$$

Koşullar Lagrange fonksiyonunda yerine konulduğunda dual problemin son haline ulaşılır.

$$Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j < x_i, x_j > \quad (3.20)$$

Kısıtlar: $\sum_{i=1}^l \alpha_i y_i = 0$

$$0 \leq \alpha_i \leq C$$

Kısıtlarda görülen C parametresi lagrange çarpanlarının alabileceği değerler için üst sınır limitidir ve kullanıcı tarafından belirlenmesi gereken bir parametredir. C parametresi lagrange çarpanlarına eşit olduğu durumda destek vektörleri optimal hiperdüzlemin üzerinde bulunur. C değerinin yüksek olduğu durumda deneme hatası da artar. Bu değer probleme kapasite belirleyici bir kontrol daha ekler.

$$w^* = \sum_{i=1}^l \alpha_i y_i x_i$$

$$b^* = \frac{-1}{2} \langle w^*, x_r + x_s \rangle \quad (3.21)$$

$$y_r = -1, y_s = 1$$

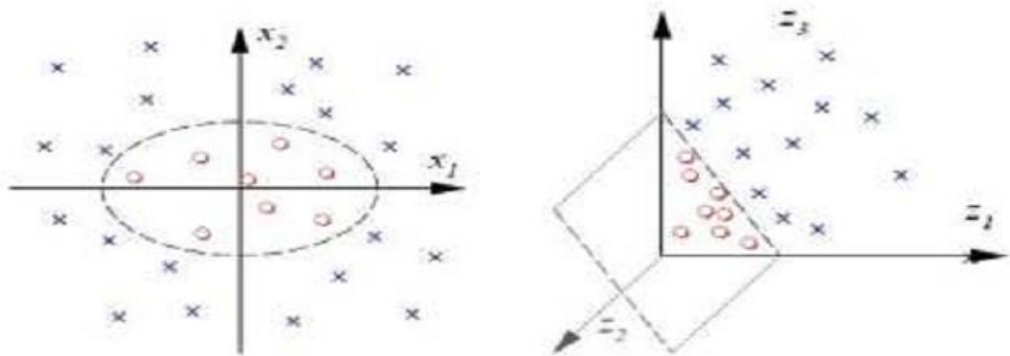
Bulunan parametrelere göre sınıflandırmada kullanılan karar fonksiyonu belirlenir.

$$f(x) = \text{sgn}(\langle w^*, x \rangle + b) \quad (3.22)$$

3.4.3 Doğrusal olmayan destek vektör makineleri

(Çok boyutlu öznitelik uzayı)

Doğrusal olmayan destek vektör makineleri, veri setinin doğrusal bir fonksiyonla tam veya belirli bir hata ile ayıramaması durumunda kullanılan algoritmalardır. Gerçek yaşam problemleri lineer olmayan özellikler içerdikleri için, bir veri setinin lineer bir hiperdüzlem ile ayrıştırılması çoğunlukla mümkün değildir. Dolayısıyla sınıflandırma probleminin çözümü, ayırma eğrisinin tahmin edilmesiyle mümkündür. Ancak uygulamada eğrinin tahmin edilmesi oldukça zordur.



Şekil 3.6: Doğrusal olmayan haritalama.

Bu durumda, veriler doğrusal olmayan haritalama yapılarak (ϕ) giriş uzayından, lineer sınıflandırmaya olanak sağlayan daha yüksek boyuttaki öznitelik uzayına aktarılır. Lineer sınıflandırma formülasyonları yardımıyla veri setini sınıflandırabilecek optimal hiperdüzlem bu yeni boyutta araştırılır.

Veri setinin bir üst uzaya taşımak için $\varphi(x)$ muhtemel geçiş fonksiyonu kullanılır.

$$\text{Optimal hiperdüzlem : } \sum_{i=1}^l w_i \varphi_i(x) + b = 0$$

$$w = \sum_{i=1}^l \alpha_i d_i \varphi_i(x) \quad (3.23)$$

$$\sum_{i=1}^l \alpha_i d_i \varphi_i^T(x) \varphi_i(x) + b = 0$$

Uygulanan haritalama fonksiyonun özelliği açık değildir ve bilindiği durumlarda da çözümü yüksek boyutlu uzayda zor ve karmaşıktır. Bu sorunun çözümü için “çekirdek fonksiyonları” geliştirilmiştir. Çekirdek fonksiyonları; polinom, radial basis ya da sigmoid şeklinden olabilmektedir [22] .

Çekirdek fonksiyonu: $K(x, x_i) = \varphi^T(x) \varphi_i(x)$

Çekirdek fonksiyonu simetrik bir fonksiyondur.

$$K(x, x_i) = K(x_i, x) \quad (3.24)$$

$$\text{Optimal hiperdüzlem : } \sum_{i=1}^l \alpha_i d_i K(x, x_i) = 0$$

Dual problemin kernel fonksiyonu ile ifadesi aşağıdaki gibidir.

$$Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (3.25)$$

Dual problemin kısıtlarında herhangi bir değişim olmaz.

$$\begin{aligned} \sum_{i=1}^l \alpha_i y_i &= 0 \\ 0 &\leq \alpha_i \leq C \end{aligned} \quad (3.26)$$

$$\langle w^*, x \rangle = \sum_{i=1}^l \alpha_i y_i K(x, x_i)$$

Sınıflandırmada kullanılan karar fonksiyonu belirlenir.

$$f(x) = \text{sgn} \left(\sum_{i=1}^l \alpha_i y_i K(x, x_i) + b \right) \quad (3.27)$$

$$b = \frac{-1}{2} \sum_{i=1}^l \alpha_i y_i [K(x, x_r) + K(x, x_s)]$$

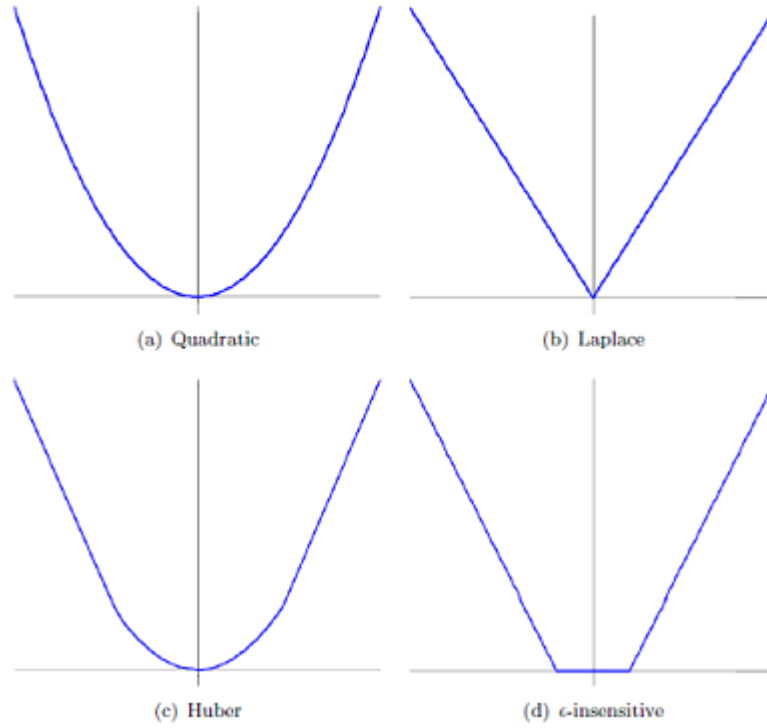
$$y_r = -1, y_s = 1$$

Çekirdek fonksiyonunu bias terimini de içerdği durumlarda karar fonksiyonu değişikliğe uğrar.

$$f(x) = \text{sgn}(\sum_{i=1}^l \alpha_i y_i K(x, x_i)) \quad (3.28)$$

3.5. Destek Vektör Makineleri ile Regresyon

Destek vektör makineleri , farklı kayıp fonksiyonları uygulanarak regresyon problemlerinin çözümünde kullanılabilir.



Şekil 3.7: Kayıp fonksiyonları.

Diğer kayıp fonksiyonları farklı destek vektörleri oluşmasına izin vermezken, Vapnik'in önerdiği ϵ -sensitive kayıp fonksiyonu , regresyon problemine destek vektör makinesinin uygulanmasına olanak sağlamaktadır. Kayıp fonksiyonunda herhangi bir duyarlılık parametresi kullanılmadığında, eğitim verisinin destek vektör olarak tanımlanması söz konusu olur. Bu durumda, bütün veriler hesaplamada kullanılacağından işlem yükü ağırlaşır.

3.5.1 ϵ -sensitive kayıp fonksiyonu

ϵ -sensitive kayıp fonksiyonu, küçük bir ϵ aralığındaki hataların ihmal edilebileceği veri setlerinde kullanılabilir. Mutlak kayıp fonksiyonundan farklı olarak, bahsedilen aralıktaki verilerin regresyon için ideal olduğu ve hata olmadığı kabul edilir. Bu şekilde bir marjın belirlenmesi, ϵ -sensitive kayıp fonksiyonunu DVM regresyonda kullanılması için uygun kılmaktadır.

Kayıp fonksiyonu: $L(d, y) = |d - y|$ d : istenen değer y : gerçek değer

$$\epsilon\text{-sensitive Kayıp Fonksiyonu: } L_\epsilon(d, y) = \begin{cases} |d - y| - \epsilon & |d - y| \geq \epsilon \\ 0 & \text{diğer} \end{cases}$$

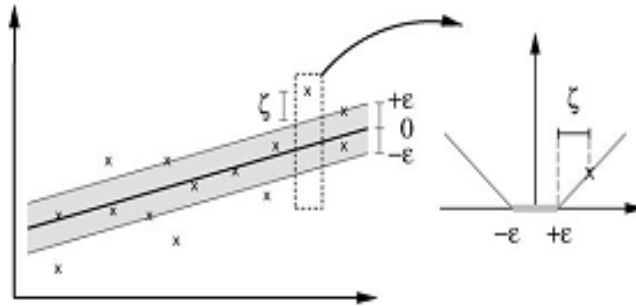
3.5.2 Doğrusal regresyon

Eğitim dataları D olarak verilmiş olsun.

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\} \quad l : \text{örnek sayısı}$$

Bu data çiftleri arasındaki örüntüyü tanımlayabilecek optimum doğrusal fonksiyon:

$$f(x) = \langle w, x \rangle + b = 0 \quad \langle \rangle : \text{nokta çarpımı}$$



Şekil 3.8: Soft margin doğrusal destek vektör.

$$\text{Primer amaç fonksiyonu: } \Phi(w, \epsilon) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\epsilon_i^- + \epsilon_i^+)$$

$$\text{Kısıtlar : } y^i - \langle w, x^i \rangle - b \leq \epsilon + \epsilon_i^+$$

$$y^i - \langle w, x^i \rangle + b \geq \epsilon + \epsilon_i^-$$

$$\epsilon_i^-, \epsilon_i^+ \geq 0$$

Lagrange çarpanları yöntemi ile dual probleme ulaşılır.

$$\begin{aligned} & \Phi(w, \epsilon, \alpha^+, \alpha^-, \eta^+, \eta^-) \\ &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\epsilon_i^- + \epsilon_i^+) - \sum_{i=1}^l (\eta_i^+ \epsilon_i^+, \eta_i^- \epsilon_i^-) - \end{aligned} \quad (3.29)$$

$$\sum_{i=1}^l \alpha_i^+ (\epsilon + \varepsilon_i^+ - y^i + \langle w, x^i \rangle + b)$$

$$- \sum_{i=1}^l \alpha_i^- (\epsilon + \varepsilon_i^- - y^i + \langle w, x^i \rangle + b)$$

Eğer noktası koşulları:

$$\Phi \frac{\partial \Phi}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) = 0$$

$$\frac{\partial \Phi}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) x_i \quad (3.30)$$

$$\frac{\partial \Phi}{\partial \varepsilon} = 0 \quad \rightarrow \quad \alpha_i + \eta_i = C$$

Koşullardan sağlanan eşitlikler yerine konulduğunda dual problem son halini alır.

$$\max_{\alpha_i^+, \alpha_i^-} (\alpha_i^+, \alpha_i^-) = \max_{\alpha_i^+, \alpha_i^-} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-)$$

$$< x_i, x_j > + \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) \cdot y_i$$

$$+ \varepsilon \cdot \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-)$$

$$\sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) = 0 \quad (3.31)$$

$$0 \leq \alpha_i^+, \alpha_i^- \leq C$$

$$w^* = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) x_i$$

$$b^* = \frac{-1}{2} < w^*, x_r + x_s >$$

Regresyon fonksiyonu: $f(x) = \langle w, x \rangle + b$

3.5.3 Doğrusal olmayan regresyon

Sınıflandırma probleminde olduğu gibi , regresyon uygulanacak veri setleri doğrusal bir fonksiyonla ifade edilemeyebilir. Bu durumda doğrusal olmayan haritalama yöntemiyle veri seti bir üst uzaya taşınır ve işlemler üst uzayda gerçekleştirilir. Çekirdek fonksiyonu kullanılarak işlemler basit hale getirilebilir.[22]

$$\begin{aligned} \max_{\alpha_i^+, \alpha_i^-} (\alpha_i^+, \alpha_i^-) = \max_{\alpha_i^+, \alpha_i^-} & -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \\ & \alpha_j^-) K(x_i, x_j) + \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) \cdot y_i + \varepsilon \cdot \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) \end{aligned} \quad (3.32)$$

Kısıtlar doğrusal regresyona göre değişiklik göstermez.

$$\begin{aligned} \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) &= 0 \\ 0 &\leq \alpha_i^+, \alpha_i^- \leq C \\ \langle w^*, x \rangle &= \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) K(x, x_i) \\ b^* &= \frac{-1}{2} \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) (K(x_i, x_r) + K(x_i, x_s)) \end{aligned} \quad (3.33)$$

Regresyon fonksiyonu: $f(x) = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) K(x, x_i) + b$

Çekirdek fonksiyonu bias terimini içeriyorsa regresyon fonksiyonu bias terimini içermez.

$$f(x) = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) K(x, x_i) \quad (3.34)$$

3.6. En Küçük Kareler Destek Vektör Makinesi

Suykens (2001) tarafından önerilen en küçük kareler destek vektör makinesi yöntemi, nonlinear özelliği ve gürültü oranı yüksek veri setlerinde regresyon problemine çözüm sunmaktadır. Bu formülasyon, black-box modelleme problemlerinde kullanılabilir. Aynı zamanda optimal kontrol problemlerinde kullanılmak üzere geliştirilmiştir [23].

3.6.1 En küçük kareler destek vektör sınıflandırması

Bu yöntemde amaç fonksiyonuna en küçük kareler yaklaşımına göre bir terim eklenir.

$$\Phi(w, e) = \frac{1}{2} \|w\|^2 + \gamma \sum_{k=1}^l e_k^2$$

$$\text{Kısıt : } y^i \cdot [< w, \varphi(x_i) > + b] \geq 1 - e_i \quad i = 1, \dots, l$$

Bu durumda optimal hiperdüzlem aşağıdaki eşitlikle tanımlanır.

$$y^i \cdot [< w, \varphi(x_i) > + b] = 1 - e_i \quad (3.35)$$

Lagrange çarpanları yöntemi ile dual problem elde edilir.

$$\Phi(w, b, e, \alpha) = \frac{1}{2} \|w\|^2 + \gamma \sum_{k=1}^l e_k^2 \sum_{i=1}^l \alpha_i [y^i \cdot [< w, \varphi(x_i) > + b] - 1 + e_i] \quad (3.36)$$

Koşullar:

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 & \quad \rightarrow \quad \sum_{i=1}^l \alpha_i y_i = 0 \\ \frac{\partial \Phi}{\partial w} = 0 & \quad \rightarrow \quad w = \sum_{i=1}^l \alpha_i y_i \varphi(x_i) \end{aligned} \quad (3.37)$$

$$\frac{\partial \Phi}{\partial e} = 0 \quad \rightarrow \quad \alpha_i = \gamma e_i$$

Bu problemin sonucu eşitlikte görülmektedir.

$$\begin{bmatrix} 0 & \bar{Y}^T \\ Y & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(x_k)^T \varphi(x_l) \quad k, l = 1, \dots, N \\ &= y_k y_l K(x_k, x_l) \end{aligned} \quad (3.38)$$

3.6.2 En küçük kareler destek vektör regresyonu

Aynı yaklaşım regresyon problemine uygulandığında regresyon fonksiyonu şu şekilde ifade edilebilir.

$$\begin{bmatrix} 0 & \overrightarrow{1^T} \\ \overrightarrow{1} & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}$$

$$\begin{aligned} \Omega_{kl} &= \varphi(x_k)^T \varphi(x_l) \quad k, l = 1, \dots, N \\ &= K(x_k, x_l) \end{aligned} \tag{3.39}$$

$$\begin{aligned} f(x) &= \langle w, \varphi(x) \rangle + b = 0 \\ f(x) &= \sum_{i=1}^l K(x, x_i) + b \end{aligned} \tag{3.40}$$

Bu sonutan daha robust bir regresyon matrisi elde etmek iin ama fonksiyonuna ağırlıklandırma parametreleri eklenebileceğı ngrlmştr.

$$\Phi(w, e) = \frac{1}{2} \|w\|^2 + \frac{1}{2} \gamma \sum_{k=1}^l v_k e_k^{2*} \tag{3.41}$$

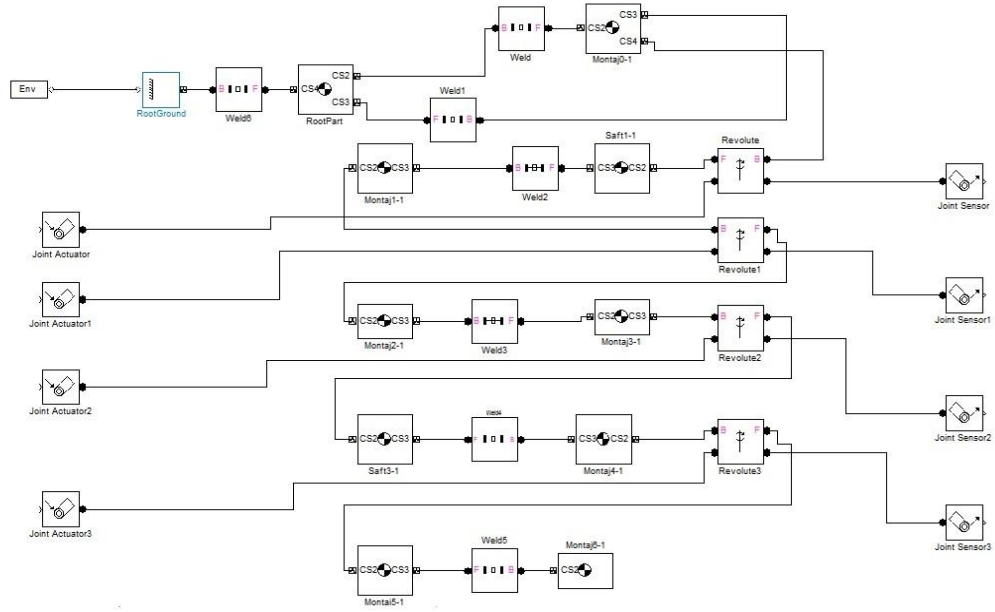
Bu durumdaki optimal hiperdzlem denklemi:

$$y^i = [\langle w^*, \varphi(x_i) \rangle + b^*] + e_i^* \quad i = 1, \dots, l \tag{3.42}$$

Robust bir tahmin yapabilmek iin, ağırlıklandırma katsayısı olan v_k , hata teriminin (e_k) bir fonksiyonu olarak seilmelidir.

4. BENZETİM SONUÇLARI

Bu tez çalışmasında, 2.bölümde anlatılan dört eksenli robot kolun 3.bölümde açıklanan destek vektör makinası algoritması yardımıyla kinematik modellemesi simüle edilmiştir. Rehabilitasyon robot kolunun konstrüksiyon çizimleri SOLIDWORKS programı kullanılarak oluşturulmuştur. Çizim verileri MATLAB/ Simmechanics toolbox yardımıyla MATLAB ortamına aktarılmıştır.



Şekil 4.1 : Robot kolu MATLAB/ Simmechanics modeli.

Denavit-Hartenberg yöntemiyle oluşturulan kinetik model parametreleri MATLAB/Simmechanics Toolbox modelinden aktarılmıştır. Şekil 4.1’de görülen model blokları her bir eklemi temsil etmekte ve eklem koordinat değerlerini içermektedir. İleri ve Ters Kinematik Modelleri için kullanılan parametreler Çizelge 4.1’de görülmektedir.

Çizelge 4.1: Robot Kolu D-H Parametreleri.

| Eklemler | a(m) | d(m) | α (rd) | θ (rd) |
|----------|------|--------|---------------|---------------|
| 1 | 0.00 | 0.2185 | 4.71 | θ_1^* |
| 2 | 0.00 | 0.2345 | 1.57 | θ_2^* |
| 3 | 0.00 | 0.0000 | 4.71 | θ_3^* |
| 4 | 0.00 | 0.1750 | 1.57 | θ_4^* |

Ters kinematik kontrolün gerçekleştirilmesi için öncelikle destek vektör makinası algoritması yardımıyla ileri kinematik modelin öğrenilmesi sağlanmıştır. Optimum parametreler ile ileri kinematik modelin öğrenilmesi sağlandıktan sonra aynı süreç ters kinematik modele uygulanmıştır. Destek vektör makinası algoritması ile oluşturulan model, ters kinematik kontrolörü olarak sisteme eklenmiş ve rehabilitasyon robot kolunun uç işlevci konum kontrolü yapılmıştır.

4.1.Dört Eksenli Robot Kolun Destek Vektör Makinası Yardımı ile İleri Kinematik Modellemesi

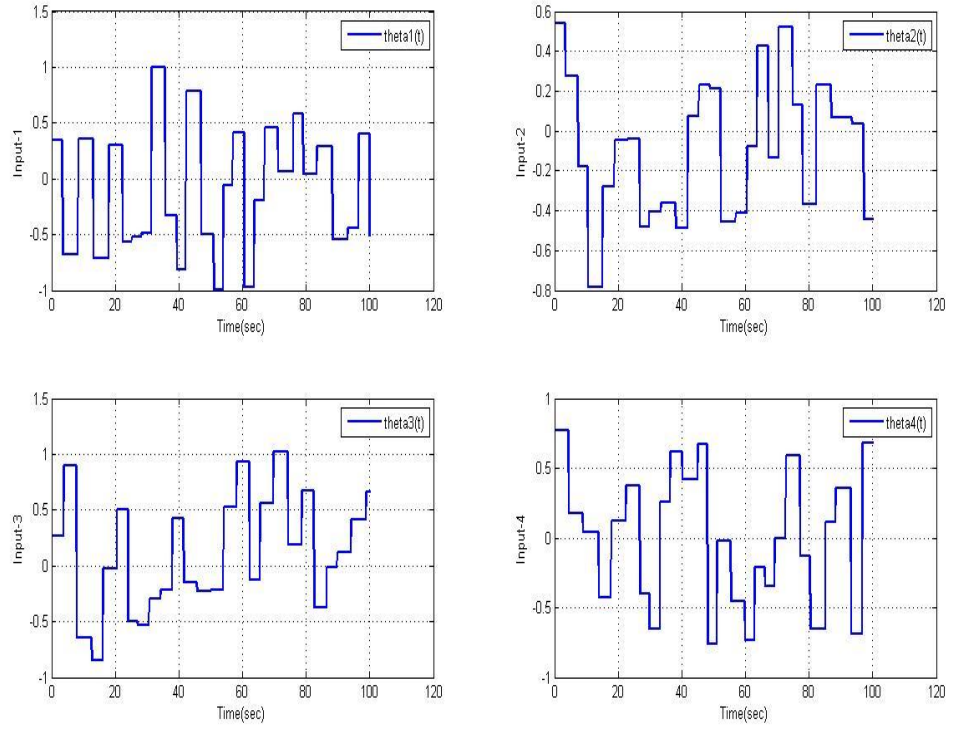
İleri kinematik model, robot eklem açıları ile uç işlevci konumu arasındaki bağıntıyı ifade etmektedir. Simülasyon çalışmasında bu bağıntı, destek vektör makinası algoritması ile tahmin edilmektedir.

Densvit-Hartenberg yöntemi ile bulunan ileri kinematik denklem kullanılarak eğitim seti oluşturulmuştur. Eğitim setini hazırlama fonksiyonu da ; giriş değeri robot kol eklem açıları , çıkış robot kol uç işlevci konumu olacak şekilde oluşturulmuştur. Giriş eklem açıları, ileri kinematik model karakteristiklerini tümüyle yansıtılabilmesi için rastgele fonksiyonu kullanılarak seçilmiştir. Ancak robot kolun hareket sınırlarının göz önünde bulundurulması gerekmektedir. Eklem açıları seçiminde, robot kolun eklemlerinin mekanik sınırlarına dikkat edilmiştir (Çizelge 4.2).

Çizelge 4.2 : Robot Kolu Mekanik Sınır Eklem Açısı Değerleri.

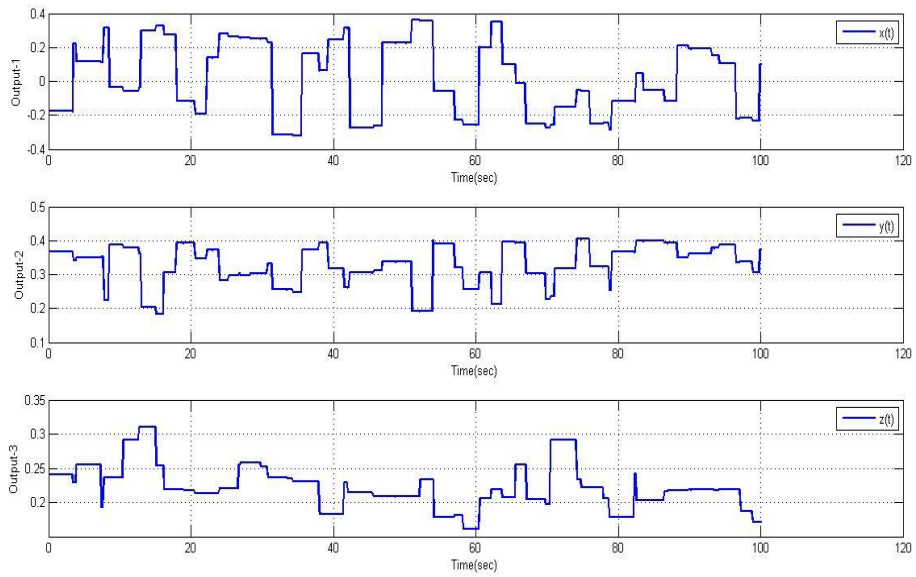
| Eklemler | Sınır Değerleri (rd) | Açı |
|-----------------|-----------------------------|------------|
| 1 | -1.04 - 1.04 | |
| 2 | -0.785 - 0.785 | |
| 3 | -1.04 - 1.04 | |
| 4 | -0.785 - 0.785 | |

Robot kolun mekanik sınırları doğrultusunda oluşturulan rastgele giriş açıları Şekil 4.2’de görülmektedir



Şekil 4.2: Eğitim verisi giriş değerleri (Robot eklem açıları).

Verilen giriş açıları doğrultusunda, Denavit-Hartenberg yöntemi ile oluşturulan kinematik model kullanıldığında, Şekil 4.3’da görülen uç işlevci konumları elde edilmektedir.



Şekil 4.3: Eğitim verisi çıkış değerleri (Robot uç işlevci konumları).

Yukarıda elde edilen giriş robot eklem açıları ve çıkış robot uç işlevci konumları kullanılarak destek vektör makinası algoritması için eğitim seti oluşturulmuştur. Eğitim seti datası ile birlikte destek vektör makinası ile kinematik model tahmini yapılmıştır.

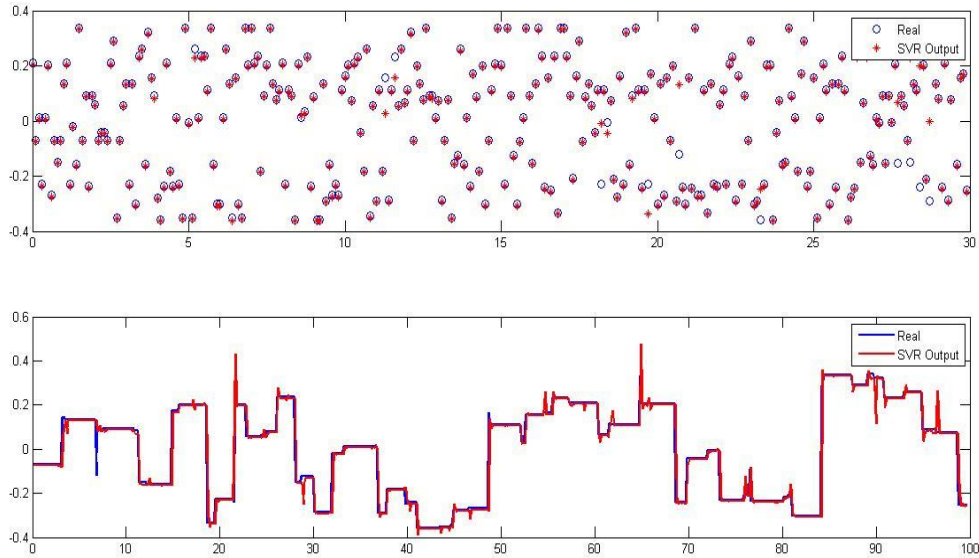
Robot kol ileri kinematik modelin destek vektör makinası algoritması ile kestirimi yapılırken, kestirim ve hata performansına göre belirlenmesi gereken parametreler bulunmaktadır. Bu parametreler gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerleridir.

İleri kinematik model eğitilirken her bir robot kol eklemi için kullanılan parametreler Çizelde 4.3'te görülebilmektedir.

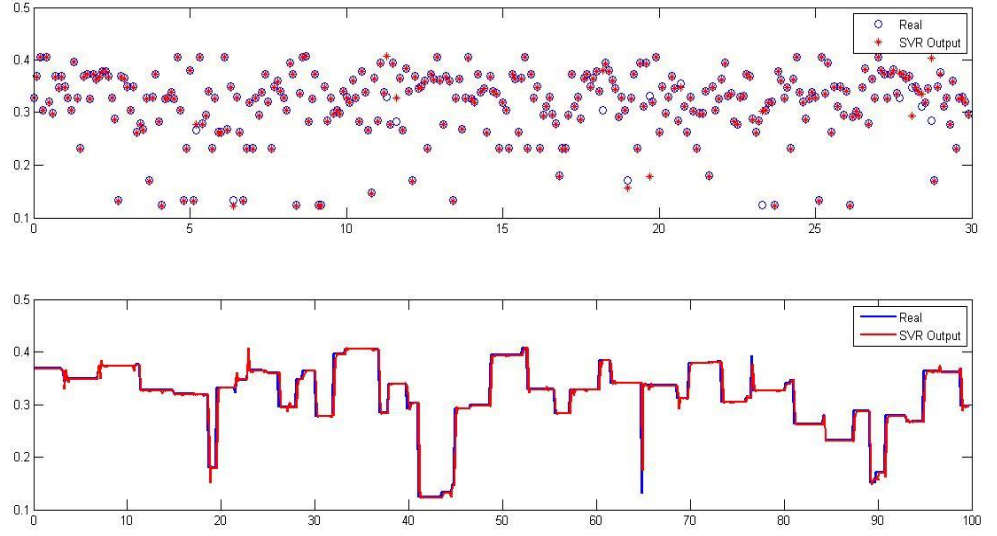
Çizelge 4.3: Destek vektör makinası regresyon parametreleri.

| Konum Değerleri | σ | ε |
|--------------------|----------|---------------|
| X | 5 | 10^{-4} |
| Y | 7 | 10^{-4} |
| Z | 8 | 10^{-4} |

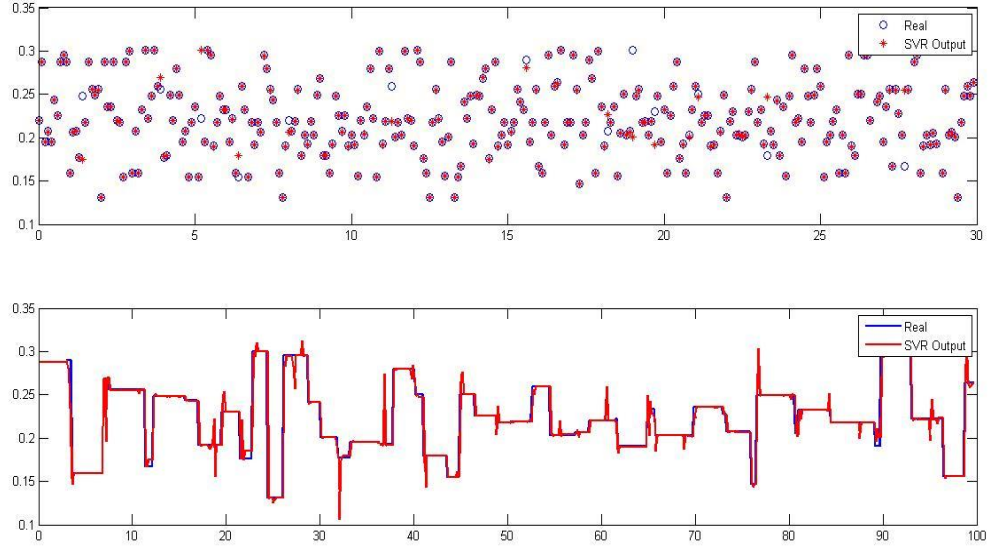
Kullanılan parametre değerlerinin kestirim performansı için Şekil 4.4'te görülen çıkış değerleri incelenebilir. Robot uç işlevci konumlarının ileri kinematik modelden çıkan gerçek değerleri ve destek vektör algoritması ile kestirilen değerleri karşılaştırılmıştır.



Şekil 4.4 : Destek vektör makinası performansı (X).



Şekil 4.5 : Destek vektör makinası performansı (Y).



Şekil 4.6: Destek vektör makinası performansı (Z).

Kestirim performansını belirlemek için çıkış robot uç işlevci konum değerlerinin eğitim ve test hataları hesaplanmıştır. Kullanılan gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerlerinin kestirim performansı Çizelge 4.4'te görülmektedir.

Çizelge 4.4 : Destek vektör makinası regresyon eğitim ve test hata değerleri.

| Konum Değerleri | Eğitim Hatası (mm) | Test Hatası (mm) |
|----------------------------|-----------------------------------|-----------------------------|
| X | 0.0128 | 0.0160 |
| Y | 0.0038 | 0.0052 |
| Z | 0.0020 | 0.0039 |

4.1.1 Grid arama yöntemi ile optimum parametrelerin bulunması

Destek vektör makinası regresyon ile robot kol eklem açıları ve uç işlevci konumu arasındaki fonksiyonu tahmin etme performansı çeşitli parametrelere bağlıdır. Öncelikle maksimum sınıflandırma hatası belirlenmelidir. Bu hata değerinden daha büyük hata değerine bağlı olan noktalar sınıflandırmaya alınmaz.

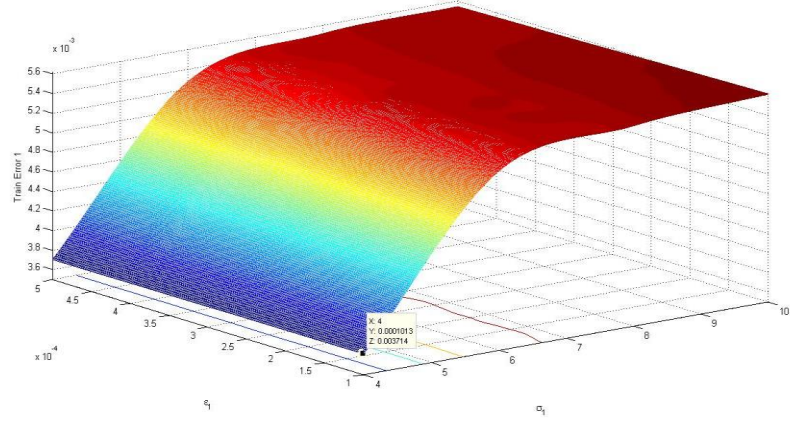
Sınıflandırma problemine göre çekirdek fonksiyonunun seçimi tahmin performansını büyük ölçüde etkilemektedir. Kinematik problemin nonlinear yapısına uygun olarak gauss çekirdek fonksiyonu destek vektör regresyonunda kullanılmıştır.

$$K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \quad (4.1)$$

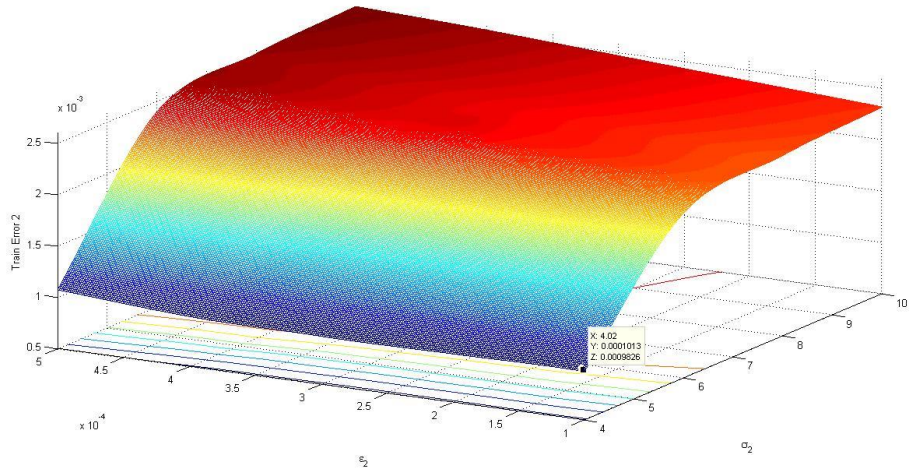
Kullanılan gauss çekirdek fonksiyonunda, gauss eğrisinin çapını belirleyen σ parametresinin seçimi, destek vektör makinası regresyonu ile kinematik modelin tahmin performansını etkileyen başka bir kriterdir.

Sınıflandırma hatası ve gauss çekirdek fonksiyonu çapı grid arama ile belirli değerler arasında aratılmıştır. Bu arama yönteminde yine ayrıca belirlenen basamak aralığı ile iki parametreye de minimum ve maksimum sınırlar arasında değerler atanmıştır. Destek vektör makinası ile eğitim işlemi, her bir değer için tekrarlanmıştır. Çıkan sonuçlar ile eğitim ve test hataları incelenerek hatanın minimum olduğu noktadaki parametre değerleri optimum olarak seçilmiştir.

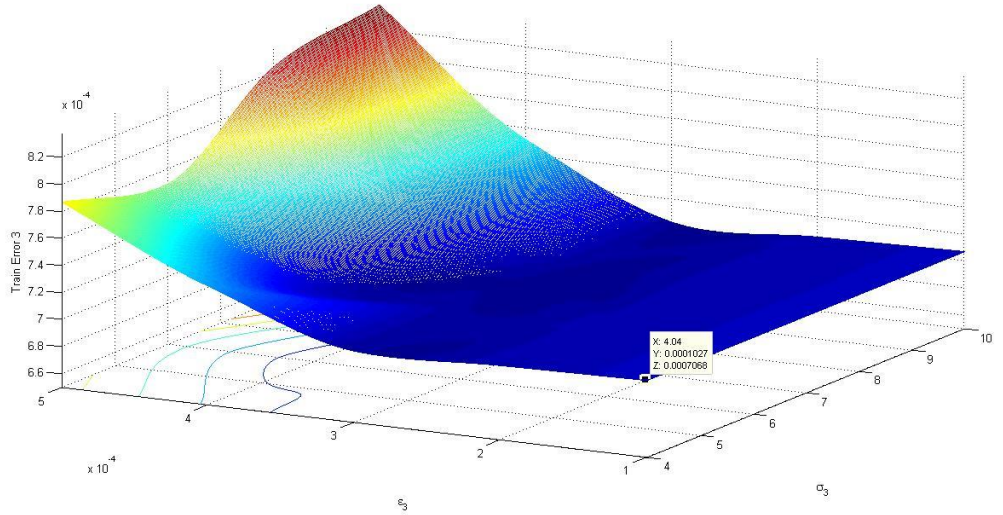
Eğitim ve test hataları, gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε giriş değeri olarak alınmak üzere yüzey olarak incelenmiştir. Eğitim hatası yüzeyleri Şekil 4.7-8-9'da görülebilmektedir.



Şekil 4.7: Eğitim hata yüzeyi (X).

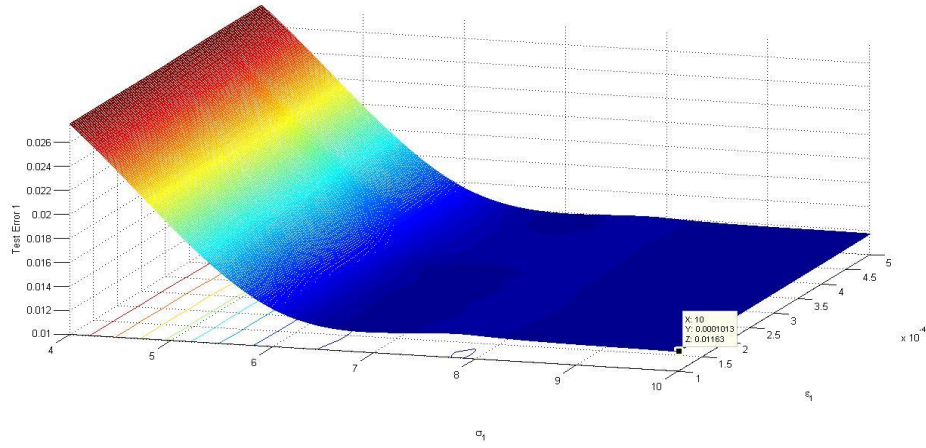


Şekil 4.8 : Eğitim hata yüzeyi (Y).

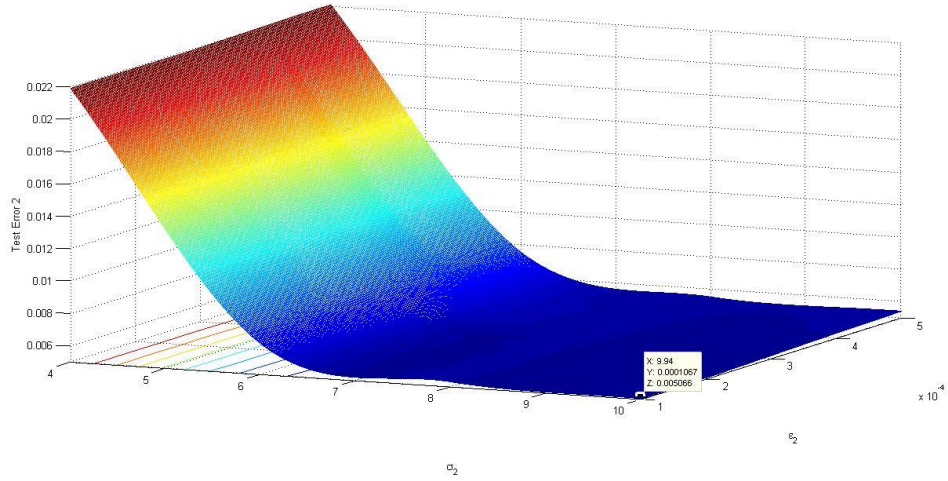


Şekil 4.9 : Eğitim hata yüzeyi (Z).

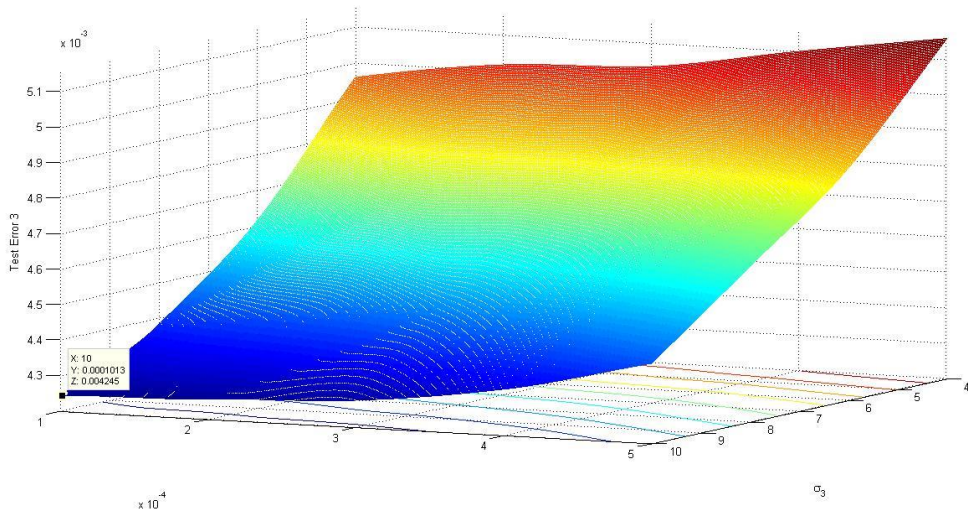
Aynı şekilde test hata yüzeyleri de Şekil 4.10-11-12’de görülmektedir.



Şekil 4.10 : Test hata yüzeyi (X).



Şekil 4.11 : Test hata yüzeyi (Y).



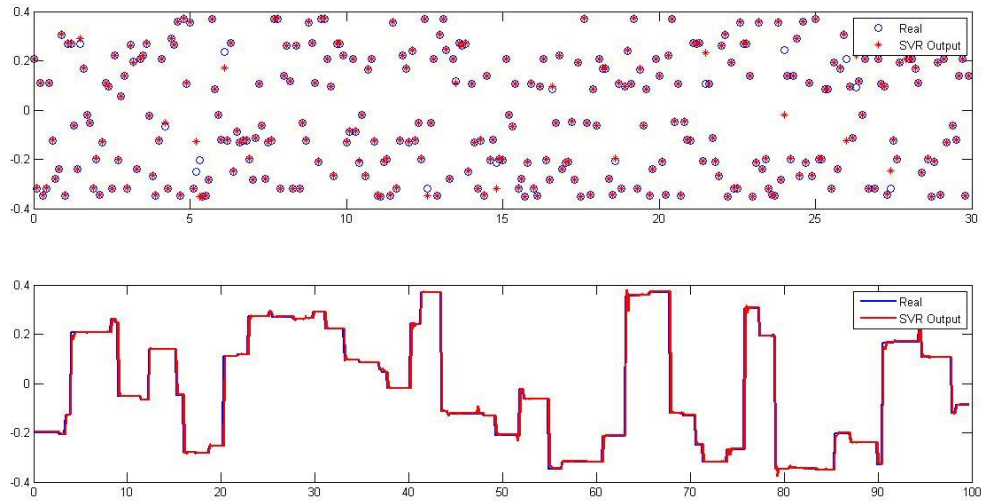
Şekil 4.12: Test hata yüzeyi (Z).

Eğitim ve test hata yüzeylerinde belirtilen minimum hata değerlerine göre optimum gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerleri Çizelge 4.5 'da incelenebilir.

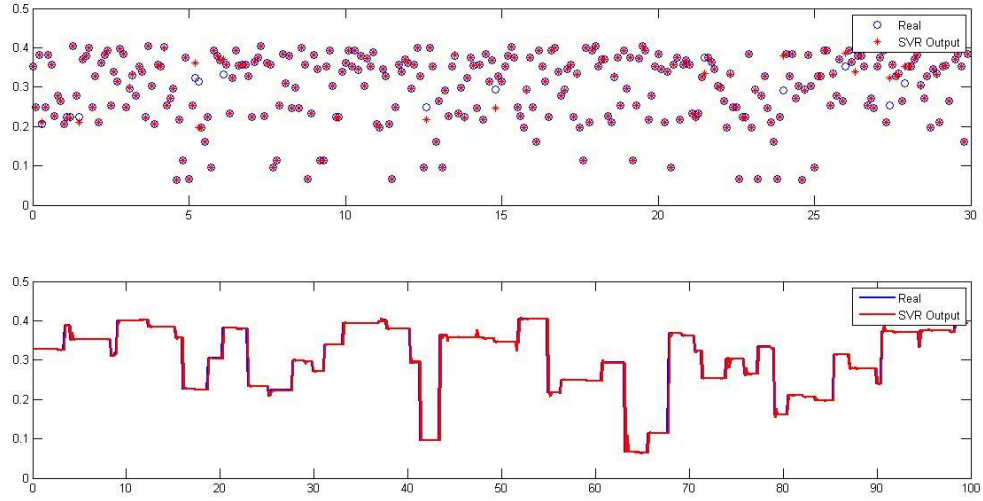
Çizelge 4.5 : Destek vektör algoritması optimum regresyon parametreleri.

| Konum Değerleri | σ | ε |
|--------------------|----------|---------------|
| X | 10 | 10^{-5} |
| Y | 10 | 10^{-5} |
| Z | 10 | 10^{-5} |

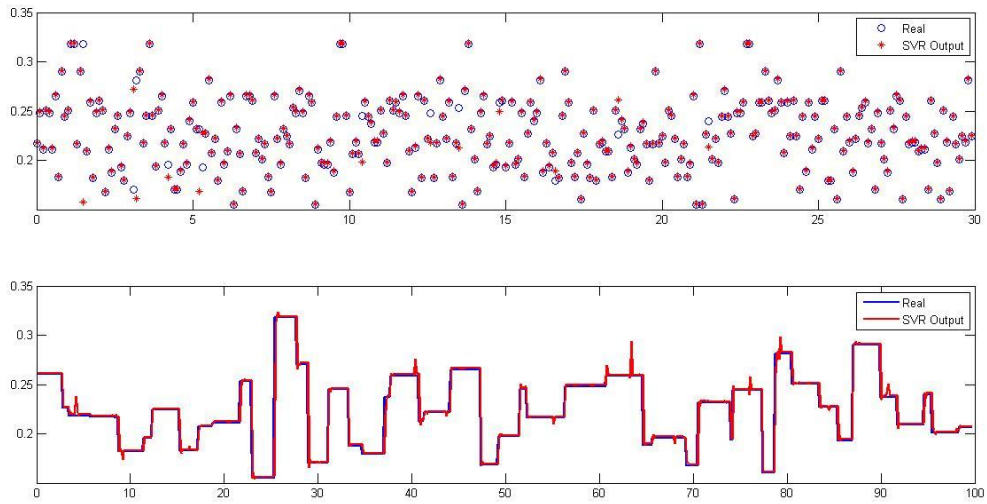
Kullanılan optimum parametre değerlerin kestirim performansı için Şekil 13-14-15'te görülen olan çıkış değerleri incelenebilir. Robot uç işlevci konumlarının ileri kinematik modelden çıkan gerçek değerleri ve destek vektör algoritması ile kestirilen değerleri karşılaştırılmıştır.



Şekil 4.13 : Destek vektör makinası performansı (X).



Şekil 4.14 : Destek vektör makinası performansı (Y).



Şekil 4.15 : Destek vektör makinası performansı (Z).

Kullanılan gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerlerinin kestirim performansı Çizelge 4.6’da görülmektedir.

Çizelge 4.6: Destek vektör makinası regresyon optimum hata değerleri.

| Konum Değerleri | Eğitim Hatası (mm) | Test Hatası (mm) |
|--------------------|--------------------------|---------------------|
| X | 0.0067 | 0.0101 |
| Y | 0.0022 | 0.0040 |
| Z | 0.0034 | 0.0034 |

4.2 Beş Eksenli Robot Kolun Destek Vektör Makinası Yardımı ile Ters Kinematik Modellemesi

Ters kinematik model, uç işlevci konumu ile robot eklem açıları arasındaki bağıntıyı ifade etmektedir. Simülasyon çalışmasında bu bağıntı, destek vektör makinası algoritması ile tahmin edilmektedir.

Geri kinematik modelin bulunmasında öncelikle daha önceki bölümlerde belirtilen ters jacobian methodu kullanılmıştır. Ancak ters jacobian algoritması, robot kol çalışma uzayı içindeki tekil noktalarda istenilen cevabı verememektedir. Destek vektör makinası algoritması için anlamlı bir eğitim setinin hazırlanmasında ters jacobian methodu kullanıldığında, tekil noktalar sağlıklı bir sonuç alınmasını engellemektedir.

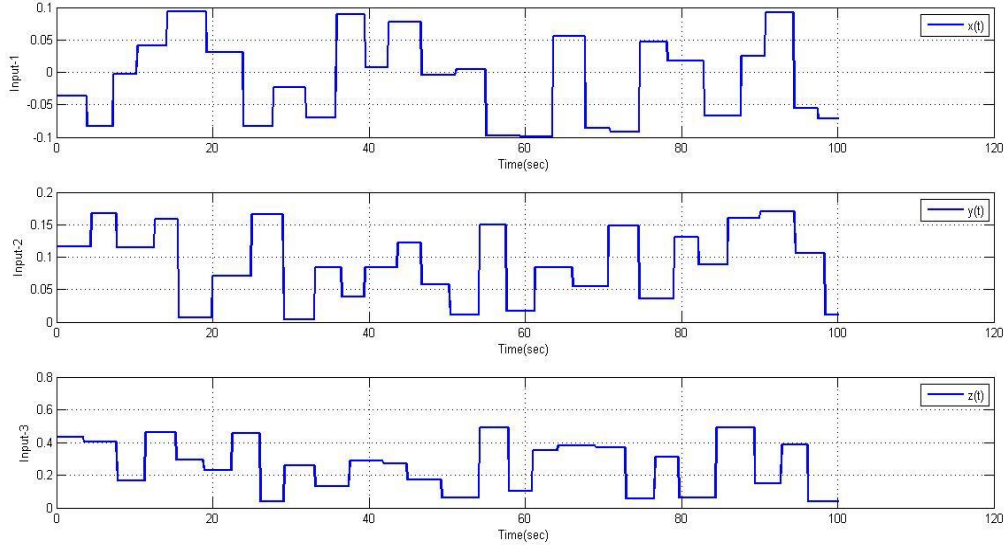
Eğitim seti hazırlanırken ters jacobian methodu yerine, geometrik yaklaşımla robot uç işlevci konumu giriş alınarak, robot eklem açı değerleri belirlenmiştir. Bölüm 2’de belirtilen geometrik yöntem ile eğitim veri seti hazırlanmıştır.

Eğitim setini hazırlama fonksiyonu da ; giriş değeri robot kol uç işlevci konumu, çıkış değeri robot kol eklem açıları olacak şekilde oluşturulmuştur. Giriş uç işlevci konumları , ters kinematik model karakteristiklerini tümüyle yansıtılabilmesi için rastgele fonksiyonu kullanılarak seçilmiştir. Ancak robot kolun çalışma uzayı sınırlarının göz önünde bulundurulması gerekmektedir. Uç işlevci konumu rastgele değerleri belirlenirken, robot kolun çalışma uzayı sınırları arasında kalmasına dikkat edilmiştir. Rehabilitasyon robot kolu çalışma uzayının bulunması Bölüm 2’de anlatılmaktadır. Çalışma uzayı sınırları Çizelge 4.7’de görülebilir.

Çizelge 4.7: Robot kolu çalışma uzayı sınır değerleri.

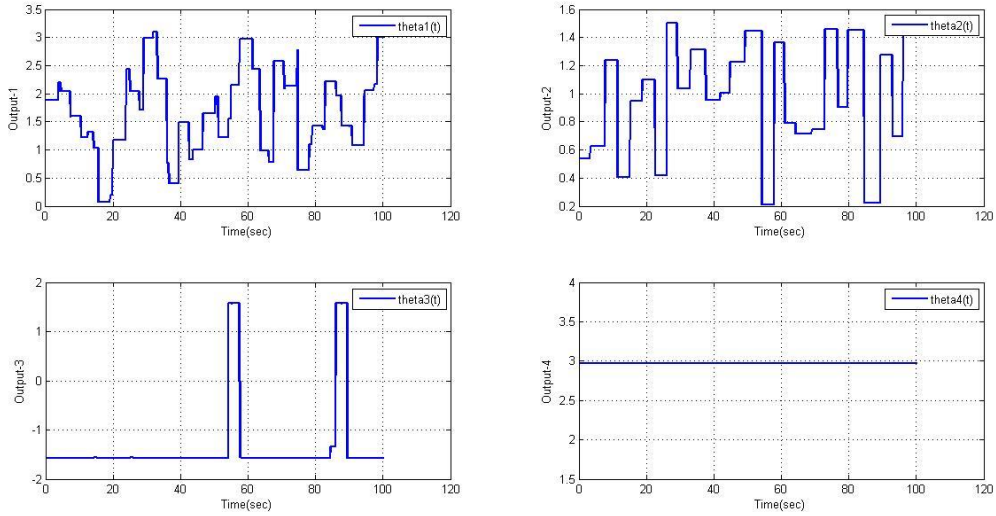
| Eksen | Sınır Değerleri (mm) | Konum |
|-------|----------------------|-------|
| X | -0.1000-0.1000 | |
| Y | 0.0000-0.1750 | |
| Z | 0.0000-0.5000 | |

Robot kolu çalışma uzayı sınırları doğrultusunda oluşturulan rastgele giriş robot uç işlevci konum değerleri Şekil 4.16’de görülmektedir.



Şekil 4.16 : Eğitim verisi giriş değerleri (Uç işlevci konumları).

Verilen uç işlevci konum değerleri doğrultusunda, geometrik yaklaşımla oluşturulan geri kinematik model kullanıldığında, Şekil 4.17’da görülen robot kol eklem açısı değerleri elde edilmektedir.



Şekil 4.17: Eğitim verisi çıkış değerleri (Eklem açısı değerleri).

Yukarıda elde edilen giriş robot uç işlevci konumları ve çıkış robot eklem açıları kullanılarak destek vektör makinası algoritması için eğitim seti oluşturulmuştur. Eğitim seti datası ile birlikte destek vektör makinası ile ters kinematik model tahmini yapılmıştır.

Robot kol ters kinematik modelin destek vektör makinası algoritması ile kestirimi yapılırken, kestirim ve hata performansına göre belirlenmesi gereken parametreler

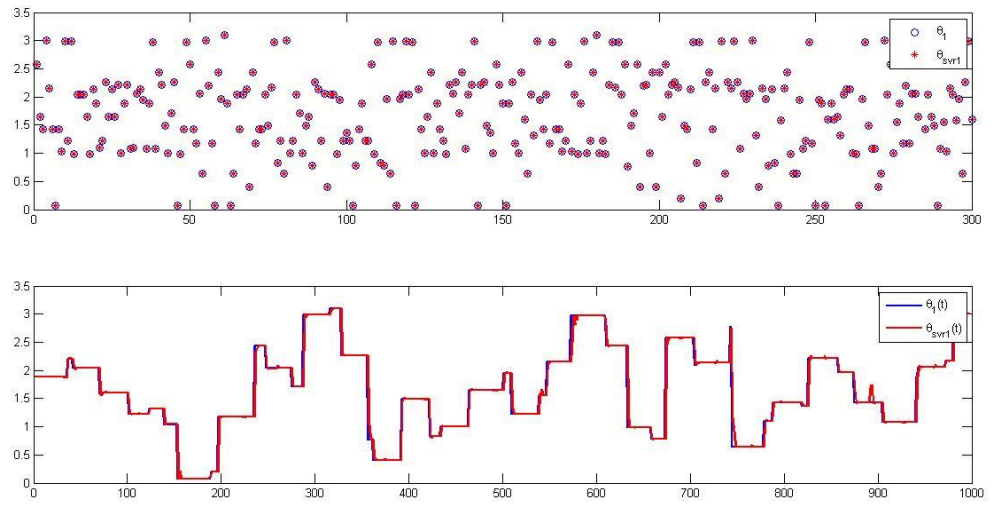
bulunmaktadır. Bu parametreler gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerleridir.

Ters kinematik model eğitilirken kullanılan parametreler Çizelge 4.8’de görülebilmektedir.

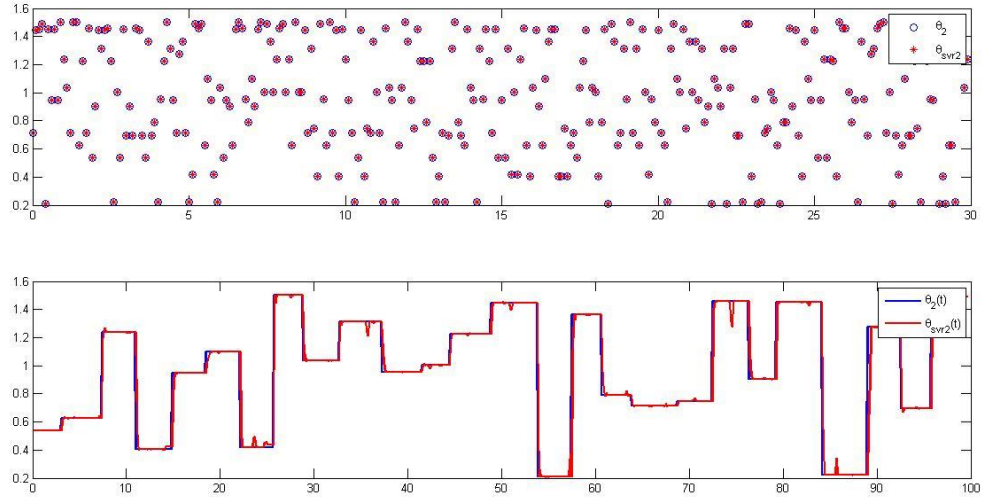
Çizelge 4.8 : Destek vektör makinası regresyon parametreleri.

| Eklem Açısı Değerleri | σ | ε |
|----------------------------------|----------------------------|---------------------------------|
| 1 | 4 | 10^{-4} |
| 2 | 4 | 10^{-4} |
| 3 | 10 | 10^{-4} |
| 4 | 4 | 10^{-4} |

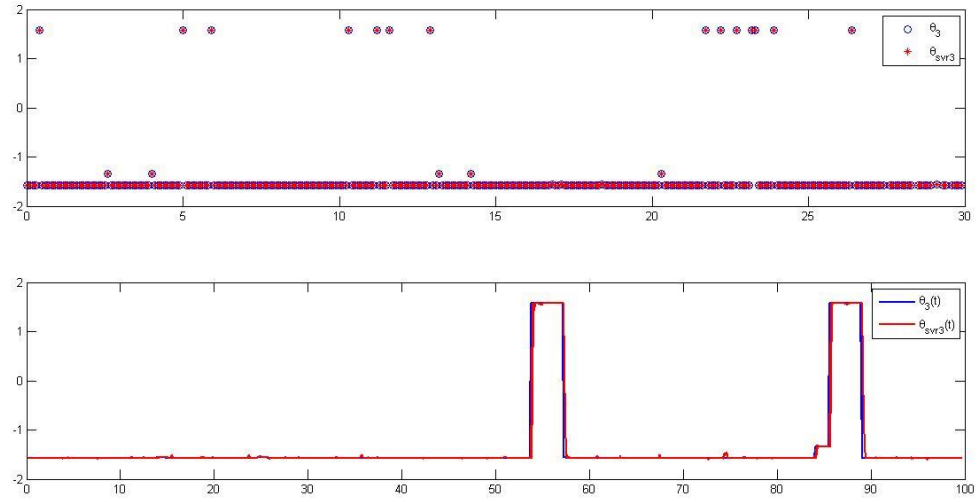
Kullanılan parametre değerlerin kestirim performansı için Şekil 4.18-19-20-21’de görülen olan çıkış değerleri incelenebilir. Robot eklem açısı değerlerinin ters kinematik modelden çıkan gerçek değerleri ve destek vektör algoritması ile kestirilen değerleri karşılaştırılmıştır.



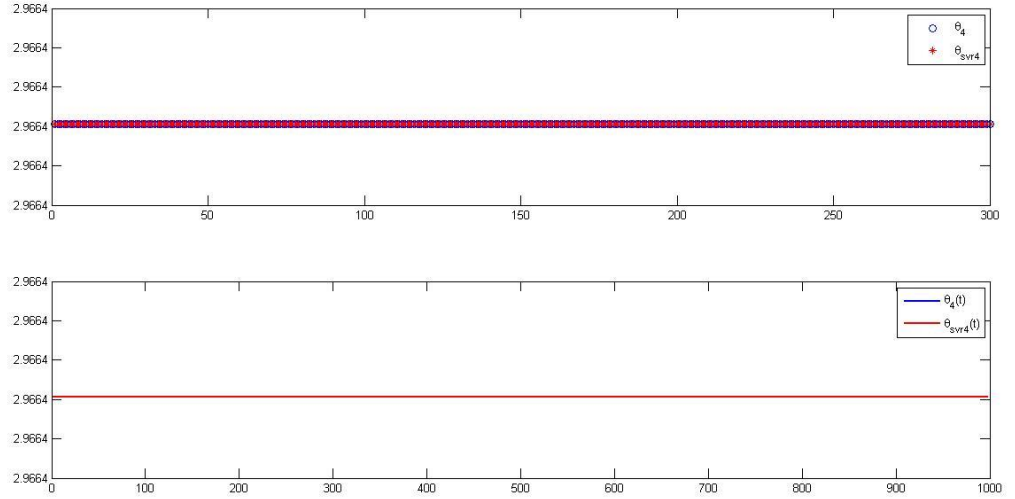
Şekil 4.18: Destek vektör makinası performansı (1.Eklem açısı).



Şekil 4.19 : Destek vektör makinası performansı (2.Eklem açısı).



Şekil 4.20: Destek vektör makinası performansı (3.Eklem açısı).



Şekil 4.21: Destek vektör makinası performansı (4.Eklem açısı).

Kestirim performansını belirlemek için çıkış robot eklem açısı değerlerinin eğitim ve test hataları hesaplanmıştır. Kullanılan gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerlerinin kestirim performansı Çizelge 4.9’da görülmektedir.

Çizelge 4.9: Destek vektör makinası regresyon eğitim ve test hata değerleri.

| Eklem Açısı Değerleri | Eğitim Hatası (rd) | Test Hatası (rd) |
|----------------------------------|-------------------------------|-----------------------------|
| 1 | 9.7809e-005 | 0.0348 |
| 2 | 9.6391e-005 | 0.0219 |
| 3 | 9.9362e-006 | 0.0235 |
| 4 | 9.9362e-006 | 0.0235 |

4.2.1 Grid arama yöntemi ile optimum parametrelerin bulunması

Destek vektör makinası regresyon ile robot kol eklem açıları ve uç işlevci konumu arasındaki fonksiyonu tahmin etme performansı çeşitli parametrelere bağlıdır. Öncelikle maksimum sınıflandırma hatası belirlenmelidir. Bu hata değerinden daha büyük hata değerine bağlı olan noktalar sınıflandırmaya alınmaz.

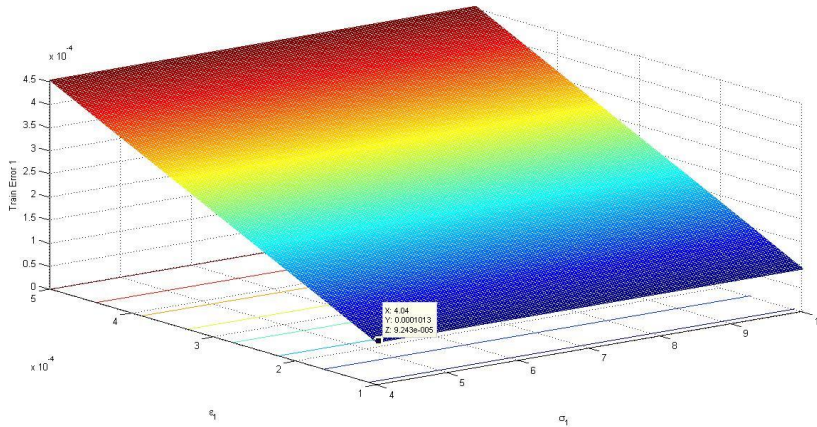
Sınıflandırma problemine göre çekirdek fonksiyonunun seçimi tahmin performansını büyük ölçüde etkilemektedir. Kinematik problemin nonlineer yapısına uygun olarak gauss çekirdek fonksiyonu destek vektör regresyonunda kullanılmıştır.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (4.2)$$

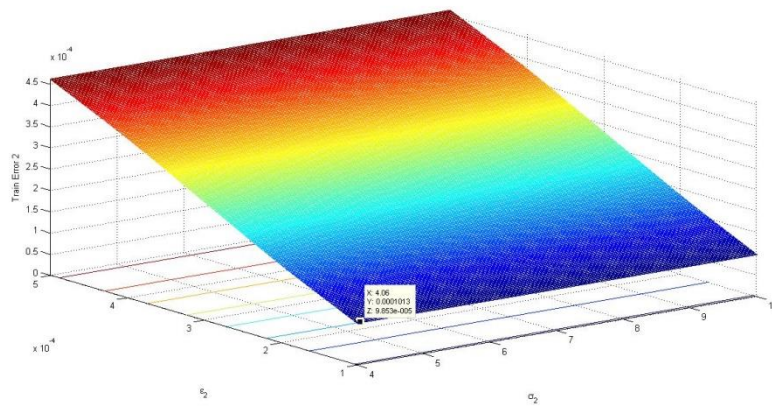
Kullanılan gauss çekirdek fonksiyonunda, gauss eğrisinin çapını belirleyen σ parametresinin seçimi, destek vektör makinası regresyonu ile kinematik modelin tahmin performansını etkileyen başka bir kriterdir.

Sınıflandırma hatası ve gauss çekirdek fonksiyonu çapı grid arama ile belirli değerler arasında aratılmıştır. Bu arama yönteminde yine ayrıca belirlenen basamak aralığı ile iki parametreye de minimum ve maksimum sınırlar arasında değerler atanmıştır. Destek vektör makinası ile eğitim işlemi, her bir değer için tekrarlanmıştır. Çıkan sonuçlar ile eğitim ve test hataları incelenerek hatanın minimum olduğu noktadaki parametre değerleri optimum olarak seçilmiştir.

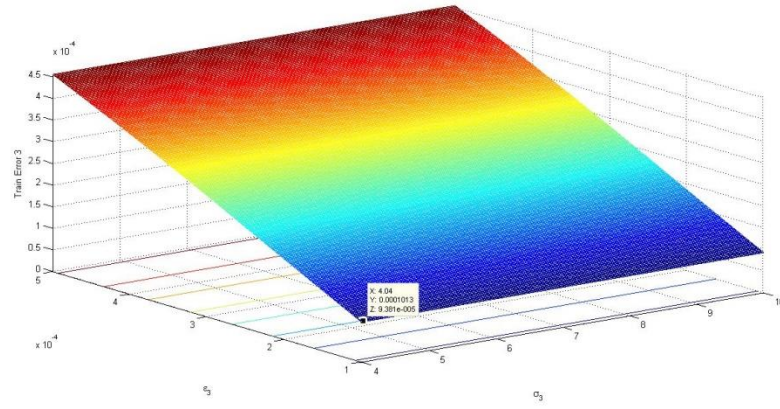
Eğitim ve test hataları, gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε giriş değeri olarak alınmak üzere yüzey olarak incelenmiştir. Robot kolun son eklemi kartezyen koordinat değişimine neden olmadığı için sabit değerde kalmaktadır. Bu sebeple dördüncü eklem için optimum parametre araması yapılmamıştır. Eğitim hatası yüzeyleri Şekil 4.22-23-24'te görülebilmektedir.



Şekil 4.22 : Eğitim hata yüzeyi (1.Eklem açısı).

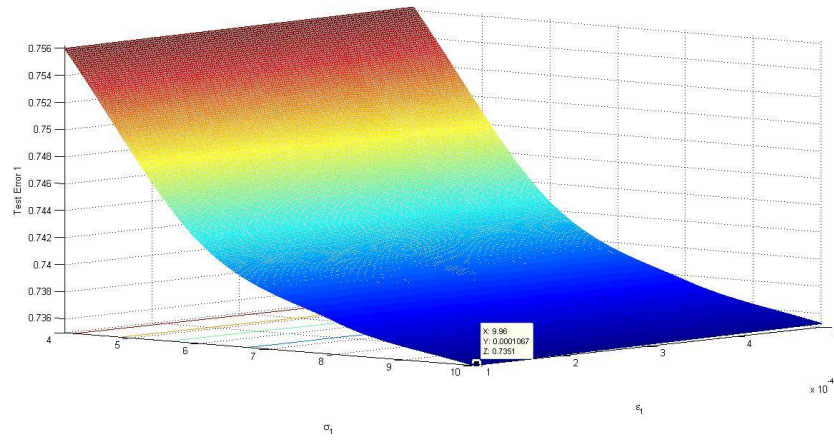


Şekil 4.23 : Eğitim hata yüzeyi (2.Eklem açısı).

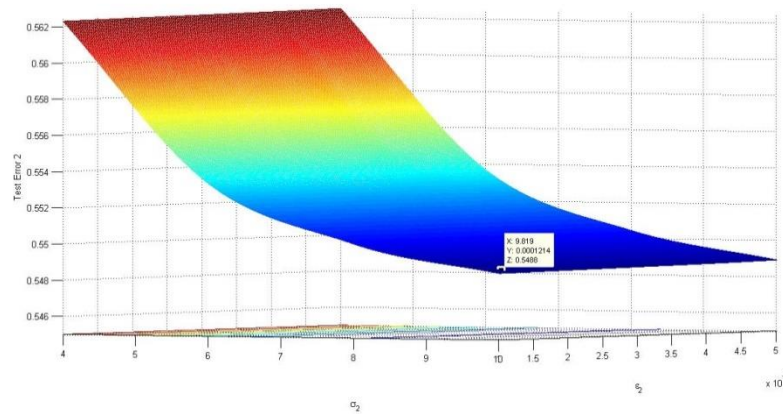


Şekil 4.24 : Eğitim hata yüzeyi (3.Eklem açısı).

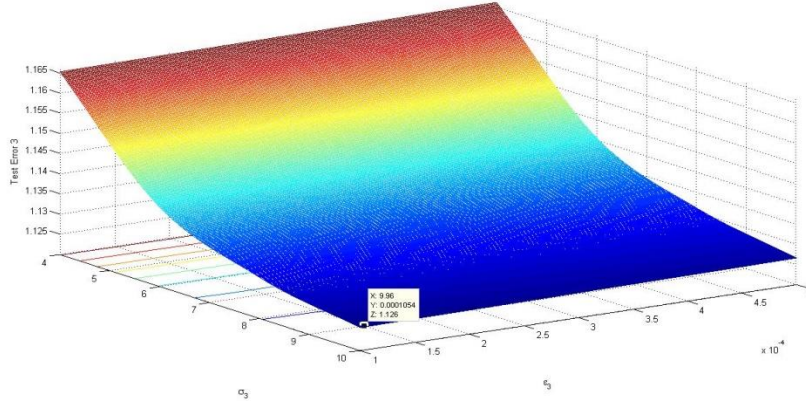
Aynı şekilde test hata yüzeyleri de Grafik 4.25-26-27’de görülmektedir.



Şekil 4.25 : Test hata yüzeyi (1.Eklem açısı).



Şekil 4.26 : Test hata yüzeyi (2.Eklem açısı).



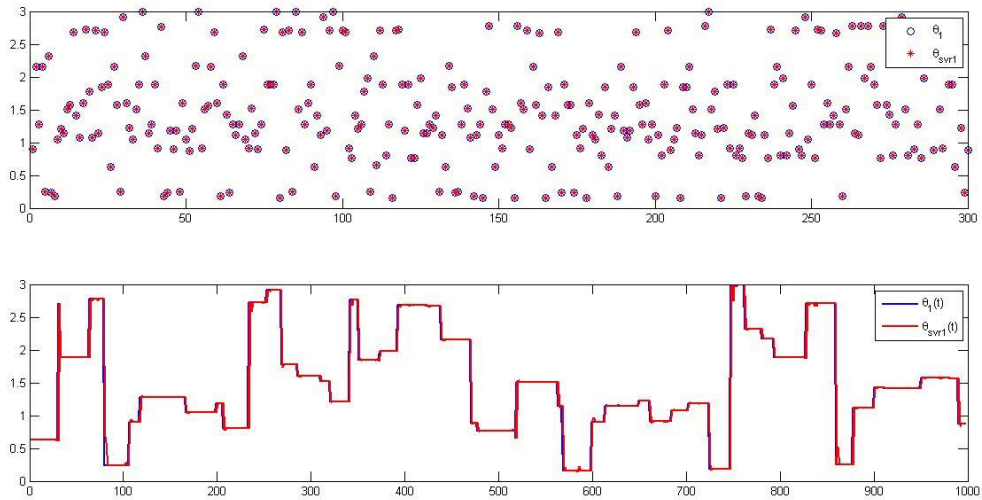
Şekil 4.27 : Test hata yüzeyi (3.Eklem açısı).

Eğitim ve test hata yüzeylerinde belirtilen minimum hata değerlerine göre optimum gauss çekirdek fonksiyonu çapı σ ve maksimum sınıflandırma hatası ε değerleri Çizelge 4.10’da incelenebilir.

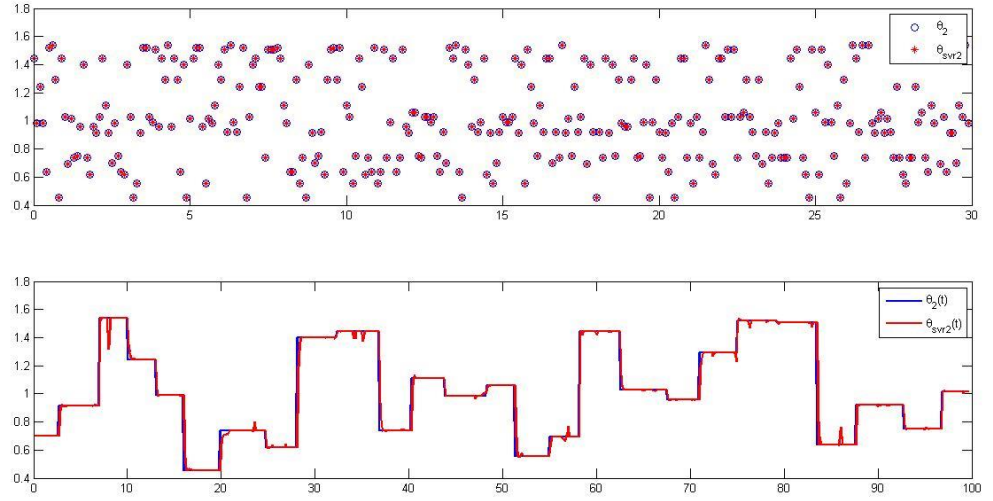
Çizelge 4.10: Destek vektör makinası optimum regresyon parametreleri.

| Eklem Açı Değerleri | σ | ε |
|--------------------------------|----------------------------|---------------------------------|
| 1 | 10 | 10^{-5} |
| 2 | 10 | 10^{-5} |
| 3 | 10 | 10^{-5} |

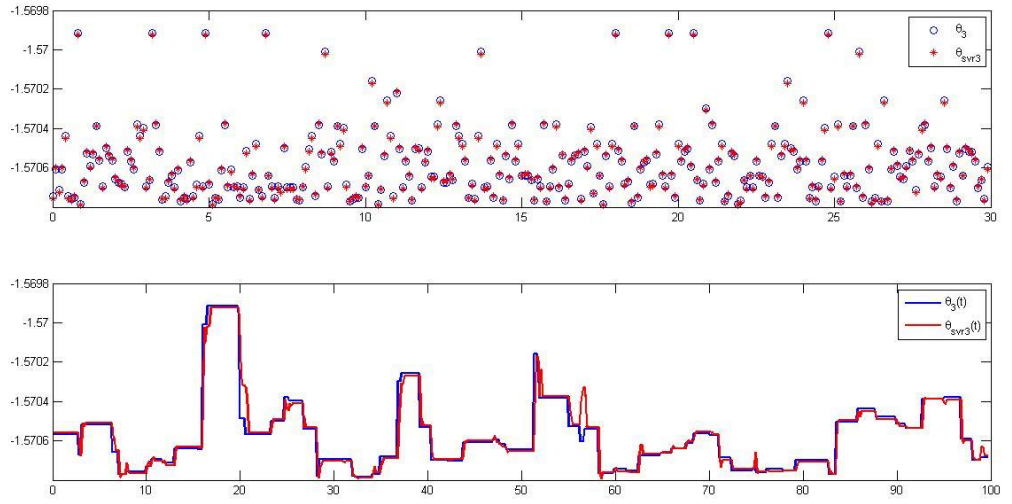
Kullanılan optimum parametre değerlerin kestirim performansı için Şekil 4.28-29-30-31’de görülen olan çıkış değerleri incelenebilir. Robot eklem açılarının ters kinematik modelden çıkan gerçek değerleri ve destek vektör algoritması ile kestirilen değerleri karşılaştırılmıştır.



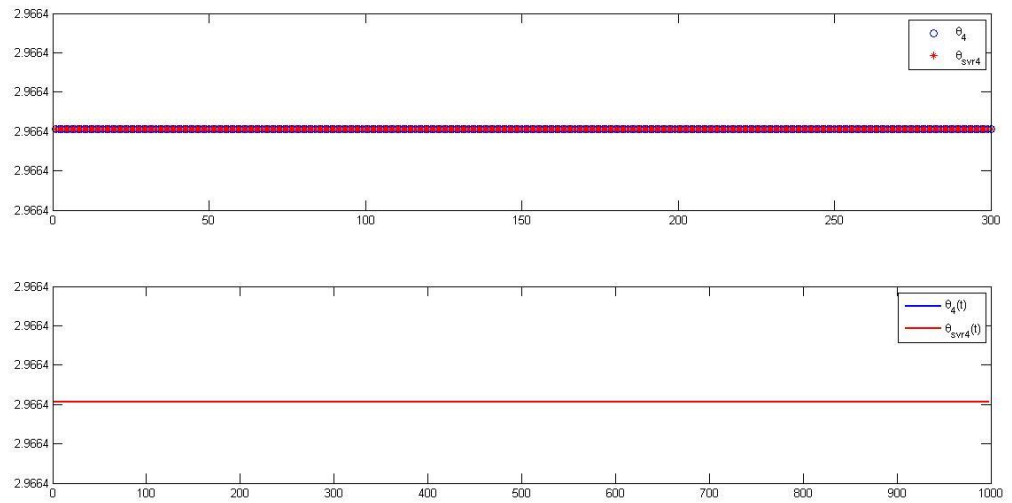
Şekil 4.28 : Destek vektör makinası performansı (1.Eklem açısı).



Şekil 4.29 : Destek vektör makinası performansı (2.Eklem açısı).



Şekil 4.30 : Destek vektör makinası performansı (3.Eklem açısı).



Şekil 4.31: Destek vektör makinası performansı (4.Eklem açısı).

Kullanılan gauss çekirdek fonksiyonu σ ve maksimum sınıflandırma hatası ε değerlerinin kestirim performansı Çizelge 4.11’de görülmektedir.

Çizelge 4.11: Destek vektör makinası regresyon optimum hata değerleri.

| Eklem Aç Değerleri | Eğitim Hatası (rd) | Test Hatası (rd) |
|-------------------------------|-------------------------------|-----------------------------|
| 1 | 9.9985e-006 | 0.0355 |
| 2 | 9.9964e-006 | 0.0137 |
| 3 | 8.6020e-006 | 2.1605e-005 |
| 4 | 8.6020e-006 | 2.1605e-005 |

5. SONUÇLAR

Bu tezin kapsamında, 4 eksenli robot kolun kinematik problemine çözüm aranmıştır. Tasarım parametreleri ve geometrik yapısı T.C. Bilim Sanayi ve Teknoloji Bakanlığı'ı desteğiyle yürütülmekte olan “Robotik Tekerlekli Sandalye” adlı projenin çalışmalarında belirlenen manipülatör, kinematik çözümleme için örnek model olarak kullanılmıştır.

İleri ve ters kinematik model, destek vektör makinası regresyonu ile tahmin edilerek analitik, geometrik ve numerik çözümlemenin neden olduğu tekillik sorunu ortadan kaldırılmış ve nonlinear model basit bir fonksiyona dönüştürülmüştür. Grid arama algoritması kullanılarak, destek vektör makinesi regresyonunda bulunan model tabanlı parametreler minimum eğitim ve test hatasına neden olacak şekilde optimize edilmiştir. Bu şekilde ileri ve ters kinematik model en az hatayla tahmin edilerek, robot kolunun hareket sınırları içindeki rastgele eklem açıları ve çalışma uzayı içindeki rastgele noktalar için kullanılabilir hale gelmiştir. Tahmin edilen kinematik model, yörünge takibi problemine kinematik tabanlı hızlı ve güvenilir bir çözüm oluşturmaktadır.

Destek vektör makinesi algoritması ile kestirilen kinematik model, yörünge takibi problemi için dinamik tabanlı bir çözüm olan hesaplanmış tork PD kontrol parametrelerinin uyarlanmasıyla kullanılabilir. Bu sayede kontrolör performansı iyileştirilerek minimum hata ile yörünge takibi sağlanması öngörülmektedir.

KAYNAKLAR

- [1] **Wolovich, W.A, Elliot H. .**(1984). A Computational Technique for Inverse Kinematics, *23rd Conference on Decision and Control*.
- [2] **Caccavale, F. , Chiacchio, P. , Chiaverini S. , Siciliano, B.** (1996). Experiments of Kinematic Control on a Redundant Robot Manipulator, *Laboratory Robotics and Automation*,8,.25–36.
- [3] **Hu, Z., FU, Z., Fang, H.** (2002). Study of singularity robust inverse of Jacobian matrix for manipulator, *International Conferenece of Machine Learning and Cybernetics*,(s.406-10).
- [4] **Nakamura, Y., Hanafusa, H.** (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control, *Journal of Dynamic Systems, Measurement and Control*,108, 163-71.
- [5] **Wampler, C.V.** (1986). Manipulator Inverse Kinematic Solutions based on vector formulations and damped least-squares methods, *IEEE Transactions on Systems, Man and Cybernetics*,16, .93-101.
- [6] **Hasan, A.T. , Ismail, N., Hamouda, A., Aris I., Marhaban M.H., Al-Assadi H.M.** (2010). Artificial Neural Network-Based Kinematics Jacobian Solution for Serial Manipulator Passing through Singular Configurations , *Advances in Engineering Software*, 41, .359–367.
- [7] **Cristianini N. , Shawe-Taylor J.** (2000). *An Introduction to Support Vector Machines and other Kernel based Learning Methods*, Cambridge University Press.
- [8] **Mike T., and Jane S., Handy 1** (2001). A Robotic System to Assist the Severely Disabled, *Technologies Tomorrow* ,(s.76-77).
- [9] **Ivan V., Oleg I., and Axel G.** (2005) . Rehabilitation Robot FRIEND II The General Concept and Current Implementation, *IEEE 9th International Conference on Rehabilitation Robotics*, Chicago, IL, USA.
- [10] **Url-1** < <http://www.researchgate.net/> > , erişim tarihi 11.04.2015.
- [11] **Url-2** < <http://www.assistive-innovations.eu/index.php/nl/> > , erişim tarihi 11.04.2015.
- [12] **Url-3** < <http://kinovarobotics.com/products/jaco-rehab-edition/> > , erişim tarihi 08.04.2015.
- [13] **Mark W. S., Seth H., and Vidyasagar M.,** (2005). Robot Modeling and Control.

- [14] **Singh, G.K.** (2010). An Analytical Solution for the Inverse Kinematics of a Redundant 7DoF Manipulator with Link Offsets . *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*,(s.2976-2982)
- [15] **Kerhet A. et al.** (2006) . A SVM-Based Approach to Microwave Breast Cancer Detection, *Engineering Applications of Artificial Intelligence* , 19, 807–18.
- [16] **Terrence S. et al.**, (2006) . *Microarray Expression Data*,16, 906–14.
- [17] **Saneej B. Chitralekha and Sirish L. Shah**, (2010), Application of Support Vector Regression for Developing Soft Sensors for Nonlinear Processes,*The Canadian Journal of Chemical Engineering* 88, 139-142.
- [18] **Campbell W. M. et al.** (2006) .Support Vector Machines Using GMM Supervectors for Speaker Verification , *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*,(s.308–11).
- [19] **Jongho S. et al.** (2009) . Model Predictive Flight Control Using Adaptive Support Vector Regression, *Neurocomputing*, 73, 1031–37.
- [20] **Iplikci S.** (2009). Controlling the Experimental Three-Tank System via Support Vector Machines, *IEEE 9th International Conference on Rehabilitation Robotics* , (s.391–400).
- [21] **Turhan K. , Serdar B.** (2013). Support Vector Machines in Wood Identification : The Case of Three Salix Species from Turkey,*Turkish Journal of Agriculture and Forestry*,38, .249-256.
- [22] **Zhang J. , Zhang Y.J., and Zhang Y. H.** (2014). Robust Control Method Based on Machine Learning for Boiler Combustion System,*Applied Mechanics and Materials* ,685,368–72.
- [23] **Aiqin H. and Yong W.** (2014).Pressure Model of Control Valve Based on LS-SVM with the Fruit Fly Algorithm, *Algorithms*, 7, 363–75.
- [24] **Liu G. et al.** (2013) .Synchronous Motor Using Support Vector Machine Generalized Inverse, *Neurocomputing* ,9, .890–98.
- [25] **Cao J. H.** (2014) .Fault Diagnosis for Electrical Control System of Automobile Based on Support Vector Machine, *Applied Mechanics and Materials* 666, 203–7.
- [26] **Morell A., Tarokh M., and Acosta L.** (2013). Solving the Forward Kinematics Problem in Parallel Robots Using Support Vector Regression, *Engineering Applications of Artificial Intelligence* ,26, 1698–1706.

- [27] **Sariyildiz E. et al.** (2012). Support Vector Regression Based Inverse Kinematic Modeling for a 7-DOF Redundant Robot Arm, *2012 International Symposium on Innovations in Intelligent Systems and Applications*, (s.1–5).
- [28] **Sariyildiz E. et al.** (2013) . Intelligent Systems Based Solutions for the Kinematics Problem of the Industrial Robot Arms, *9th Asian Control Conference (ASCC)*, (s.1–6).
- [29] **Gireesh Kumar T.** (2010). Fuzzy Support Vector Machine-Based Multi-Agent ,*Optimal Path Planning Approach to Robotics Environment* ,60, 387–91.
- [30] **Hirayama M. et al.,** (2004). Vision-Based Self-Localization in Non-Stationary Environments by Using Support Vector Machines, *IEEE Conference on Cybernetics and Intelligent Systems*, (s. 123–28).
- [31] **Quing L. et al.** (2014). Identification of Fruit and Branch in Natural Scenes for Citrus Harvesting Robot Using Machine Vision and Support Vector Machine, 7, 115–22.
- [32] **Vapnik V.** (2009) . An Overview of Statistical Learning Theory., *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council* ,10, 988–99.
- [33] **Gunn S.,** (1998) . *Support Vector Machines for Classification and Regression* (pp.235-237). Berlin: Wiley.
- [34] **Farag A. , Mohamed M..** (2004). *Regression Using Support Vector Machines: Basic Foundations* (pp, 256-351). New York: Berkeley.
- [35] **Cortes C., Vapnik V.** (1995) . *Support-Vector Networks* (pp. 78-109). Moscow: Wesley.
- [36] **Haykin S.** (1999) . *Neural Networks: A Comprehensive Foudation*, (2nd Edition,, pp. 36-39). London: Cambridge.
- [37] **Smola A., Schölkopf B.** (2003) . *A tutorial on support vector regression* (pp. 237-256). Berlin: Wiley.

EKLER

EK-A : İleri Kinematik Matlab Kodu

EK-B : Ters Kinematik Matlab Kodu –Ters Jakobiye

EK-C : Ters Kinematik Matlab Kodu –Geometrik

EK-D : Destek Vektör Makinesi Algoritması Matlab Kodu (İleri ve Ters Kinematik)

EK-A İleri Kinematik Matlab Kodu

```
function mposition = forward_kinematic(u)

%   teta=[   1.456;
%           0.846;
%           0.931;
%           0.3925] ;

teta=u;

alfa1=4.71; alfa2=1.57; alfa3=4.71; alfa4=1.57;

d1=0.2185; d2=0.2345; d3=0.0; d4=0.1750;
a1=0.0; a2=0; a3=0.0; a4=0;

A1=[ cos(teta(1))  -sin(teta(1))*cos(alfa1)  sin(teta(1))*sin(alfa1)
      sin(teta(1))*sin(alfa1)  a1*cos(teta(1)) ;
      sin(teta(1))  cos(teta(1))*cos(alfa1)  -
cos(teta(1))*sin(alfa1)  a1*sin(teta(1));
      0  sin(alfa1)  cos(alfa1)
d1  ;
      0  0  0
1  ];

A2=[cos(teta(2))  -sin(teta(2))*cos(alfa2)  sin(teta(2))*sin(alfa2)
a2*cos(teta(2)) ;
      sin(teta(2))  cos(teta(2))*cos(alfa2)  -
cos(teta(2))*sin(alfa2)  a2*sin(teta(2));
      0  sin(alfa2)  cos(alfa2)
d2  ;
      0  0  0
1  ];

A3=[cos(teta(3))  -sin(teta(3))*cos(alfa3)  sin(teta(3))*sin(alfa3)
a3*cos(teta(3)) ;
      sin(teta(3))  cos(teta(3))*cos(alfa3)  -
cos(teta(3))*sin(alfa3)  a3*sin(teta(3));
      0  sin(alfa3)  cos(alfa3)
d3  ;
      0  0  0
1  ];

A4=[cos(teta(4))  -sin(teta(4))*cos(alfa4)  sin(teta(4))*sin(alfa4)
a4*cos(teta(4)) ;
      sin(teta(4))  cos(teta(4))*cos(alfa4)  -
cos(teta(4))*sin(alfa4)  a4*sin(teta(4));
      0  sin(alfa4)  cos(alfa4)
d4  ;
      0  0  0
1  ];

T2=A1*A2;
```

```
T3=T2*A3;  
T4=T3*A4;
```

```
o4=[ T4(13); T4(14); T4(15)];
```

```
    mposition=o4;  
end
```


EK-B Ters Kinematik Matlab Kodu –Ters Jakobiyen

```
function teta = inverse_kinematic(u)

dposition=[ u(1);u(2);u(3)];
% dposition=[0.05 ;0.10 ; 0.15];

% initialize joint angles
teta=[ 0.196555 ; 0.196555; 0.196555; 0.196555 ;0.196555] ;

% joint variables
alfa1=4.71; alfa2=1.57; alfa3=4.71; alfa4=1.57; alfa5=0.0;

d1=0.0; d2=0.2345; d3=0.0; d4=0.2185; d5=0.1750;
a1=0.0; a2=0; a3=0.0; a4=0; a5=0;

% rotation and transformation matrices
A1=[ cos(teta(1)) -sin(teta(1))*cos(alfa1) sin(teta(1))*sin(alfa1)
      sin(teta(1))*sin(alfa1) a1*cos(teta(1)) ;
      sin(teta(1)) cos(teta(1))*cos(alfa1) -
cos(teta(1))*sin(alfa1) a1*sin(teta(1));
      0 sin(alfa1) cos(alfa1)
d1      ;
1      0 0 0
      ];

A2=[cos(teta(2)) -sin(teta(2))*cos(alfa2) sin(teta(2))*sin(alfa2)
a2*cos(teta(2)) ;
      sin(teta(2)) cos(teta(2))*cos(alfa2) -
cos(teta(2))*sin(alfa2) a2*sin(teta(2));
      0 sin(alfa2) cos(alfa2)
d2      ;
1      0 0 0
      ];

A3=[cos(teta(3)) -sin(teta(3))*cos(alfa3) sin(teta(3))*sin(alfa3)
a3*cos(teta(3)) ;
      sin(teta(3)) cos(teta(3))*cos(alfa3) -
cos(teta(3))*sin(alfa3) a3*sin(teta(3));
      0 sin(alfa3) cos(alfa3)
d3      ;
1      0 0 0
      ];

A4=[cos(teta(4)) -sin(teta(4))*cos(alfa4) sin(teta(4))*sin(alfa4)
a4*cos(teta(4)) ;
      sin(teta(4)) cos(teta(4))*cos(alfa4) -
cos(teta(4))*sin(alfa4) a4*sin(teta(4));
      0 sin(alfa4) cos(alfa4)
d4      ;
1      0 0 0
      ];

A5=[cos(teta(5)) -sin(teta(5))*cos(alfa5) sin(teta(5))*sin(alfa5)
a5*cos(teta(5)) ;
      sin(teta(5)) cos(teta(5))*cos(alfa5) -
cos(teta(5))*sin(alfa5) a5*sin(teta(5));
```

```

0          sin(alfa5)          cos(alfa5)
d5          ;
0          0          0
1          ];

T2=A1*A2;
T3=T2*A3;
T4=T3*A4;
T5=T4*A5;

%current position
o5=[ T5(13); T5(14); T5(15)];
mposition=o5 ;

%change of position
dx=dposition-mposition;
j=1;

%updating joint angles until error is met
while abs(dx(1))>0.001 || abs(dx(2))>0.001 || abs(dx(3))>0.001

alfa1=4.71; alfa2=1.57; alfa3=4.71; alfa4=1.57; alfa5=0.0;
d1=0.0; d2=0.2345; d3=0.0; d4=0.2185; d5=0.1750;
a1=0.0; a2=0; a3=0.0; a4=0; a5=0;
%computing jacobian
A1=[ cos(teta(1)) -sin(teta(1))*cos(alfa1) sin(teta(1))*sin(alfa1)
sin(teta(1))*sin(alfa1) a1*cos(teta(1)) ;
sin(teta(1)) cos(teta(1))*cos(alfa1) -
cos(teta(1))*sin(alfa1) a1*sin(teta(1));
0 sin(alfa1) cos(alfa1)
d1          ;
0          0          0
1          ];

A2=[cos(teta(2)) -sin(teta(2))*cos(alfa2) sin(teta(2))*sin(alfa2)
a2*cos(teta(2)) ;
sin(teta(2)) cos(teta(2))*cos(alfa2) -
cos(teta(2))*sin(alfa2) a2*sin(teta(2));
0 sin(alfa2) cos(alfa2)
d2          ;
0          0          0
1          ];

A3=[cos(teta(3)) -sin(teta(3))*cos(alfa3) sin(teta(3))*sin(alfa3)
a3*cos(teta(3)) ;
sin(teta(3)) cos(teta(3))*cos(alfa3) -
cos(teta(3))*sin(alfa3) a3*sin(teta(3));
0 sin(alfa3) cos(alfa3)
d3          ;
0          0          0
1          ];

A4=[cos(teta(4)) -sin(teta(4))*cos(alfa4) sin(teta(4))*sin(alfa4)
a4*cos(teta(4)) ;
sin(teta(4)) cos(teta(4))*cos(alfa4) -
cos(teta(4))*sin(alfa4) a4*sin(teta(4));

```

```

d4      0      sin(alfa4)      cos(alfa4)
      ;
      0      0      0
1      ];

A5=[cos(teta(5))  -sin(teta(5))*cos(alfa5)  sin(teta(5))*sin(alfa5)
a5*cos(teta(5)) ;
      sin(teta(5))  cos(teta(5))*cos(alfa5)  -
cos(teta(5))*sin(alfa5)  a5*sin(teta(5));
      0      sin(alfa5)      cos(alfa5)
d5      ;
      0      0      0
1      ];

T2=A1*A2;
T3=T2*A3;
T4=T3*A4;
T5=T4*A5;

```

```

z0=[0;0;1];
z1=[ A1(9); A1(10); A1(11)];
z2=[ T2(9); T2(10); T2(11)];
z3=[ T3(9); T3(10); T3(11)];
z4=[ T4(9); T4(10); T4(11)];
z5=[ T5(9); T5(10); T5(11)];

```

```

o0=[0;0;0];
o1=[ A1(13); A1(14); A1(15)];
o2=[ T2(13); T2(14); T2(15)];
o3=[ T3(13); T3(14); T3(15)];
o4=[ T4(13); T4(14); T4(15)];
o5=[ T5(13); T5(14); T5(15)];

```

```

a=cross(z0,(o5-o0));
b=cross(z1,(o5-o1));
c=cross(z2,(o5-o2));
d=cross(z3,(o5-o3));
e=cross(z4,(o5-o4));

```

```

%current position
o5=[ T5(13); T5(14); T5(15)];
mposition=o5 ;
dx=dposition-mposition;

```

```

Jv=[ a b c d e];

```

```

%pseudo inverse of jacobian
pJv=pinv(Jv);

%updating joint angles
teta=teta+(pJv*dx);

j=j+1;

end

%joint angle limit
for i=1:1:5

    while(teta(i) > 3.14 )
        teta(i)=teta(i)-6.28;
    end

    while(teta(i) < -3.14 )
        teta(i)=teta(i)+6.28;
    end
end

end

```

EK-C Ters Kinematik Matlab Kodu –Geometrik

```
function teta = inverse_kinematic_geo(u)

%d-h parameter d for link 3
%d3=0.21851111;
d3=0.45;

%desired position
px=u(1);
py=u(2);
pz=u(3);

%x-y projection for teta1
teta1=atan2(py,px);

%
teta2=acos(pz/d3);

Va=[px ;py; pz];

%rotation matrix from link 3 to base frame
As=[ cos(teta1)*cos(teta2)   sin(teta1);
     sin(teta1)*cos(teta2)   cos(teta1);
     -sin(teta2)             0.2;];
A=pinv(As);
%B= [cos(teta3) sin(teta3)
B= A*Va;
teta3=atan2(B(2),B(1));

%????????????
%d=0.21851111;
d=0.45;
alf2=acos((d^2)/(2*d));
alf1=alf2;

bau=(pi/2) - atan(0.45/0.55);
oal=(pi/2) - atan(0.45/0.55);

tetau=1.57+alf2;
tetal=1.57+alf1;
teta4=tetau+tetal;
%teta4=(alf1+alf2);

teta=[ teta1 teta2 teta3 teta4 ];

for i=1:1:4

    if(teta(i) > 3.14 )
        teta(i)=teta(i)-6.28;
    end
```

```
        if(teta(i) < -3.14 )  
            teta(i)=teta(i)+6.28;  
        end  
    end  
end
```

EK-D Destek Vektör Makinesi Algoritması Matlab Kodu (İleri ve Ters Kinematik)

```
clear all
clc
close all

load system2.mat y_system u_control

nu=3;
ny=3;

X_train=[];

Y_train1=[];
Y_train2=[];
Y_train3=[];

y_system1=y_system(1,:);
y_system2=y_system(2,:);
y_system3=y_system(3,:);

u_control1=u_control(1,:);
u_control2=u_control(2,:);
u_control3=u_control(3,:);
u_control4=u_control(4,:);

N_xtr=300;

m=randperm(length(y_system1)-max(nu,ny))+max(nu,ny);
indexoftrain=m(1:N_xtr);

for n=1:length(indexoftrain)
    X_train=[X_train;u_control1(indexoftrain(n)-(0:nu))
    u_control2(indexoftrain(n)-(0:nu)) u_control3(indexoftrain(n)-(0:nu))
    u_control4(indexoftrain(n)-(0:nu))
    y_system1(indexoftrain(n)-(0:ny)) y_system2(indexoftrain(n)-(0:ny))
    y_system3(indexoftrain(n)-(0:ny)) ];

    Y_train1=[Y_train1;y_system1(indexoftrain(n)+1)];
    Y_train2=[Y_train2;y_system2(indexoftrain(n)+1)];
    Y_train3=[Y_train3;y_system3(indexoftrain(n)+1)];
end

%Xtrain ==>  Eđitilecek Giriř Datası
%Ytrain ==>  Eđitilecek Çıkıř Datası
%ss        ==>  Kernel Parametresi
%epsilon==>  Tüp Geniřliđi
```

```

%C      ==> Alfaları (yada betaların) üst sınırı
%Ntr    ==> training datasının uzunluğu

Xtrain=X_train';

%kernel parametres
ss1=10;
ss2=10;
ss3=10;

C=1000;

Ntr=size(Xtrain,2);%Train datasının uzunluğu(boyutu)

for q=1:1:3

%=====Input Space deki Datalar Feture Space e
Taşınıyor=====
%fprintf('hessian matrisi oluşturuluyor.....\n');

switch q

case 1

fprintf('Position x is trained.....\n');

epsilon=1e-5;

Ytrain1=Y_train1';
f=[epsilon*ones(Ntr,1)-Ytrain1' ;epsilon*ones(Ntr,1)+Ytrain1'];
for i=1:Ntr
    for j=1:Ntr
        H1(i,j)=kernelfunction(Xtrain(:,i),Xtrain(:,j),ss1);
    end
end

case 2

fprintf('Position y is trained.....\n');

epsilon=1e-5;

Ytrain2=Y_train2';
f=[epsilon*ones(Ntr,1)-Ytrain2' ;epsilon*ones(Ntr,1)+Ytrain2'];

```



```

for i=1:Ntr
    for j=1:Ntr
        H1(i,j)=kernelfunction(Xtrain(:,i),Xtrain(:,j),ss2);
    end
end

case 3

fprintf('Position z is trained.....\n');

epsilon=1e-5;

Ytrain3=Y_train3';
f=[epsilon*ones(Ntr,1)-Ytrain3' ;epsilon*ones(Ntr,1)+Ytrain3'];

for i=1:Ntr
    for j=1:Ntr
        H1(i,j)=kernelfunction(Xtrain(:,i),Xtrain(:,j),ss3);
    end
end

end

%=====Dual Problemin kısıtlamalarla Minimize
Edilmesi=====

%=====
%               min 0.5*x'Hx + f'x   subject to:  Ax <= b
%               x
%=====

%=====epsilon-svr 2009 =====

H=[H1 -H1;-H1 H1];
H = H+1e-10*eye(size(H));

v1b=[zeros(2*Ntr,1)]; % alfas >= 0
vub=[C*ones(2*Ntr,1)]; % alfas <= C
x0= [zeros(2*Ntr,1)]; % başlangıç noktası [0 0 0 .... 0]

A = [];
b = [];

```

```

        Aeq = [ones(1,Ntr) -ones(1,Ntr)];
        beq = 0;
%===== Quadratic programming Options
=====
        %options = optimset('Display','iter')
        options = optimset;
        options.LargeScale = 'on';
        options.Display = 'on';
        options.MaxIter=100000;
%=====
=====

fprintf('Sistem Egitiliyor.....\n');

%pause
%[alfas] = qp(H, f, A, b, vlb, vub, x0,1,-1); % matlab 5.3

[alfas,fval,eXitflag,output,lambda]=quadprog(H,f,A,b,Aeq,beq,vlb,vub
,x0,options); % matlab 2009 a

fval
eXitflag
output
lambda

disp('Eğitim bitti')
pause(10)

%=====

%=====Betalar Bulunuyor=====
for i=1:Ntr
    beta(1,i)=alfas(i,1)-alfas(i+Ntr,1);
end
%betai =alpha(1:Ntr) - alpha(Ntr+1:2*Ntr);%alternatif olarak bu kodu
kullanabilirsin

%=====

switch q

```

```

case 1

Ntr=size(Xtrain,2);
SVbeta=beta(find(abs(beta)>1e-9)); %destek vektörlerinin betaları
NSVi=length(SVbeta);
XSVi=Xtrain(:,find(abs(beta)>1e-9)); %sadece destek vektörlerini
alıyor

for m=1:Ntr
    x=Xtrain(:,m);
    wx(m)=0;
    for i=1:NSVi
        wx(m)=wx(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss1);
    end
end

bx=Ytrain1-wx;

b=mean(bx);

Xtest=Xtrain;

for m=1:Ntr
    x=Xtest(:,m);
    y_hat1(m)=0;
    for i=1:NSVi
        y_hat1(m)=y_hat1(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss1);
    end
    y_hat1(m)=y_hat1(m)+b;
end

X_test=[];
Y_test=[];

for n=max(nu,ny)+1:length(y_system1)-1

    X_test=[X_test;u_control1(n-(0:nu)) u_control2(n-(0:nu))
u_control3(n-(0:nu)) u_control4(n-(0:nu)) y_system1(n-(0:ny))
y_system2(n-(0:ny)) y_system3(n-(0:ny)) ];
    Y_test=[Y_test;y_system1(n+1)];

end

Xtest=X_test';
Ytest=Y_test';

Ytest1=Ytest;

Ntst=length(Y_test);
for m=1:Ntst
    x=Xtest(:,m);
    y_tst_hat1(m)=0;

```

```

    for i=1:NSVi
y_tst_hat1(m)=y_tst_hat1(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss1
);
    end
    y_tst_hat1(m)=y_tst_hat1(m)+b;
end

case 2

Ntr=size(Xtrain,2);
SVbeta=beta(find(abs(beta)>1e-9)); %destek vektörlerinin betaları
NSVi=length(SVbeta);
XSVi=Xtrain(:,find(abs(beta)>1e-9)); %sadece destek vektörlerini
alıyor

for m=1:Ntr
    x=Xtrain(:,m);
    wx(m)=0;
    for i=1:NSVi
        wx(m)=wx(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss2);
    end
end

bx=Ytrain2-wx;

b=mean(bx);

Xtest=Xtrain;

for m=1:Ntr
    x=Xtest(:,m);
    y_hat2(m)=0;
    for i=1:NSVi
        y_hat2(m)=y_hat2(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss2);
    end
    y_hat2(m)=y_hat2(m)+b;
end

X_test=[];
Y_test=[];

for n=max(nu,ny)+1:length(y_system1)-1

    X_test=[X_test;u_control1(n-(0:nu)) u_control2(n-(0:nu))
u_control3(n-(0:nu)) u_control4(n-(0:nu)) y_system1(n-(0:ny))
y_system2(n-(0:ny)) y_system3(n-(0:ny)) ];
    Y_test=[Y_test;y_system2(n+1)];

end

Xtest=X_test';
Ytest=Y_test';

```

```

Ytest2=Ytest;

Ntst=length(Y_test);
for m=1:Ntst
    x=Xtest(:,m);
    y_tst_hat2(m)=0;
    for i=1:NSVi

y_tst_hat2(m)=y_tst_hat2(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss2
);
        end
    y_tst_hat2(m)=y_tst_hat2(m)+b;
end

case 3

Ntr=size(Xtrain,2);
SVbeta=beta(find(abs(beta)>1e-9)); %destek vektörlerinin betaları
NSVi=length(SVbeta);
XSVi=Xtrain(:,find(abs(beta)>1e-9)); %sadece destek vektörlerini
alıyor

for m=1:Ntr
    x=Xtrain(:,m);
    wx(m)=0;
    for i=1:NSVi
        wx(m)=wx(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss3);
    end
end

bx=Ytrain3-wx;

b=mean(bx);

Xtest=Xtrain;

for m=1:Ntr
    x=Xtest(:,m);
    y_hat3(m)=0;
    for i=1:NSVi
        y_hat3(m)=y_hat3(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss3);
    end
    y_hat3(m)=y_hat3(m)+b;
end

X_test=[];
Y_test=[];

for n=max(nu,ny)+1:length(y_system1)-1

    X_test=[X_test;u_control1(n-(0:nu)) u_control2(n-(0:nu))
u_control3(n-(0:nu)) u_control4(n-(0:nu)) y_system1(n-(0:ny))
y_system2(n-(0:ny)) y_system3(n-(0:ny))    ];

```

```

        Y_test=[Y_test;y_system3(n+1)];

end

Xtest=X_test';
Ytest=Y_test';

Ytest3=Ytest;

Ntst=length(Y_test);
for m=1:Ntst
    x=Xtest(:,m);
    y_tst_hat3(m)=0;
    for i=1:NSVi

y_tst_hat3(m)=y_tst_hat3(m)+SVbeta(i)*kernelfunction(x,XSVi(:,i),ss3
);
        end
        y_tst_hat3(m)=y_tst_hat3(m)+b;
    end

end

end

%=====plot train and test values=====

figure(1)

Ts=0.1;
i1=0:Ts:(length(Ytrain1)-1)*Ts;

subplot(2,1,1)
plot(i1,Ytrain1,'bo');
hold on
plot(i1,y_hat1,'r*');
legend('Real','SVR Output')

i2=0:Ts:(length(Ytest2)-1)*Ts;

subplot(2,1,2)
plot(i2,Ytest1,'b','linewidth',2);
hold on
plot(i2,y_tst_hat1,'r','linewidth',2);
legend('Real','SVR Output')

figure(2)
Ts=0.1;
i1=0:Ts:(length(Ytrain2)-1)*Ts;

```

```

subplot(2,1,1)
plot(i1,Ytrain2,'bo');
hold on
plot(i1,y_hat2,'r*');
legend('Real','SVR Output')

i2=0:Ts:(length(Ytest2)-1)*Ts;

subplot(2,1,2)
plot(i2,Ytest2,'b','linewidth',2);
hold on
plot(i2,y_tst_hat2,'r','linewidth',2);
legend('Real','SVR Output')

figure(3)

Ts=0.1;
i1=0:Ts:(length(Ytrain3)-1)*Ts;

subplot(2,1,1)
plot(i1,Ytrain3,'bo');
hold on
plot(i1,y_hat3,'r*');
legend('Real','SVR Output')

i2=0:Ts:(length(Ytest3)-1)*Ts;

subplot(2,1,2)
plot(i2,Ytest3,'b','linewidth',2);
hold on
plot(i2,y_tst_hat3,'r','linewidth',2);
legend('Real','SVR Output')

pause(1)

%=====Gereksiz değişkenler Siliniyor=====

clear H1 H f vlb vub x0 A i beta;

%=====

```


ÖZGEÇMİŞ



Ad Soyad : Sanem DOKUZLU

Doğum Yeri ve Tarihi : Bursa 28.09.1990

E-Posta : dokuzlus@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : İstanbul Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Kontrol Mühendisliği Bölümü
2012