

## LAZER MARKALAMA ROBOT SİSTEMİ İÇİN İNSAN MAKİNE ARAYÜZÜ TASARIMI

Mehmet Emre KILIÇ<sup>1, a)</sup> Taha Erim ÇOLAK<sup>1, b)</sup> Ferit GÖÇERİ<sup>1, c)</sup> Adnan Erdem TEPE DÜZÜ<sup>1, d)</sup> ve  
Mehmet İsmet Can DEDE<sup>1, e)</sup>

<sup>1</sup>İzmir Yüksek Teknoloji Enstitüsü, Makina Mühendisliği Bölümü 35430, İZMİR

<sup>a)</sup>mehmetkilic@std.iyte.edu.tr, <sup>b)</sup>tahacolak@std.iyte.edu.tr, <sup>c)</sup>feritgoceri@std.iyte.edu.tr,  
<sup>d)</sup>adnantepeduzu@std.iyte.edu.tr, <sup>e)</sup>candede@iyte.edu.tr

İnsan makine arayüzü, bir insanın bir makine, sistem veya cihazla iletişime geçmesini sağlayan bir arabirimdir. Bu çalışmada daha önceki çalışmalarda geliştirilmiş bir lazer markalama robot sistemine tablet üzerinden çizilecek iz verisini almak için insan makine arayüzü eklenmiştir. İnsandan arayüz ile alınan iz verisinin makina tarafından bir yüzeye markalanabilmesi için yörünge planlama algoritmaları kullanılmıştır. Kullanılan yörünge planlama algoritmalarının sonuçları bir görüntü işleme yöntemi ile karşılaştırılmıştır.

**Anahtar Kelimeler:** İnsan-makine arayüzü, yörünge planlama, lazer markalama robotu

### HUMAN MACHINE INTERFACE DESIGN FOR LASER MARKING ROBOT SYSTEM

#### ABSTRACT

A human-machine interface (HMI) is a communication channel that enables humans to interact with machines, systems, or devices. In this study, a human-machine interface (HMI) was added to a previously developed laser marking robot system to capture drawn path data from a tablet. Trajectory planning algorithms were used to enable the machine to mark the path data received from the human via the interface onto a surface. The results of the trajectory planning algorithms were then compared via an image processing procedure.

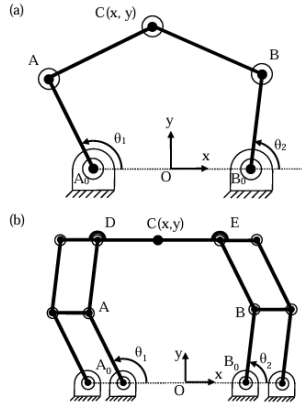
**Keywords:** Human-machine interface, trajectory planning, laser marking robot

#### 1. GİRİŞ

İnsan-makine arayüz etkileşimi kullanıcı ile makinenin iletişimini sağlar. Geliştirilen arayüz algoritmaları ve tasarımlarıyla kullanıcının makineyi kontrol etmesi daha kolay hale gelir [1]. Bu çalışmada 116M272 numaralı TÜBİTAK 1001 projesi kapsamında geliştirilmiş fazla kısıtlı 6R düzlemsel lazer markalama robot sistemine bir tablet arayüzü eklentisi yapılması amaçlanmıştır. Düzlemsel paralel robotun kinematığı Şekil 1(b)'de gösterilmektedir. Fazla kısıtlar sayesinde 6R düzlemsel mekanizmanın hareketli platformu sadece iki eksenle öteleme hareketleri yapabilmektedir. Fazla kısıtlı mekanizmalarda analitik ters kinematik model oluşturmak mümkün değildir. Dolayısıyla Kiper vd. [2] mevcut mekanizmanın kinematik analizini

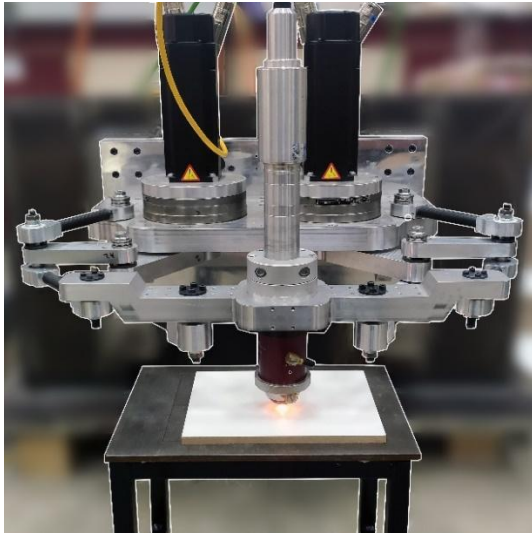
mekanizmanın “gizli robotu” olan 5R mekanizmasına göre bütün uzuv boylarının eşit olduğu varsayımı ile yapmışlardır. Sonrasında kalibrasyon çalışmaları ve mekanizmanın esneklik modelini yörünge planlama algoritmasına ekleyerek lazer markalama işleminin yüksek ivmeli çalışma esnasında iz takip hassasiyetini iyileştirmişlerdir [3]. Şekil 1(a)'da gösterilen  $A_0$  ve  $B_0$  noktalarında mafsallara motorlar bağlanmış olup bu motorlar  $\theta_1$  ve  $\theta_2$  açı değerleri kontrol ederek hareketli platformun iki eksendeki konumunu değiştirmektedir.

Geliştirilen mekanizma 5 g ivme değerlerine çıkabilmektedir. Robotun görseli Şekil 2'de verilmiştir. Robotun hareketli platformuna bir lazer markalama kafası monte edilmiştir.



Şekil 1 (a) Normal kısıtlı 5R mekanizması, (b) Fazla kısıtlı 6R mekanizması [2]

Bu çalışmanın geliştirilen sisteme insan-makine arayüzü eklentisi geliştirerek kullanıcıdan tablet üzerinde iz bilgisi girilmesi hedeflenmiştir. Kayıt altına alınan iz verisi Matlab ortamında oluşturulan yörünge planlama algoritmaları ile zamana bağlı veri haline getirilmiştir. Son aşamada ise ters kinematik denklemleri kullanılarak motorlara beslenecek açı bilgileri elde edilmiştir. Literatürdeki benzer çalışmalar incelendiğinde CNC freze makinelerinde kullanılan CAD programların yerine kullanılmak üzere serbest çizim ve renk ölçütleri kullanarak ‘draw2cut’ yöntemi geliştirildiği tespit edilmiştir. Söz konusu çalışmada, kullanıcı kesim yapılacak malzeme üzerine doğrudan çizim yaparak tasarımını gerçekleştirebilmektedir [4].



Şekil 2. Lazer markalama robotu

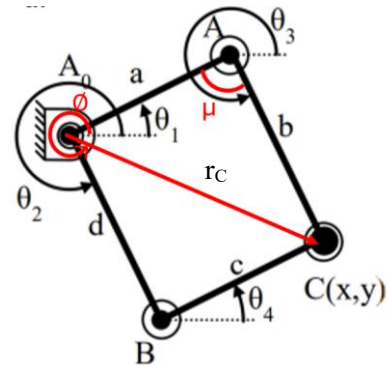
Geleneksel makine arayüzlerinde yüklenmiş belirli fonksiyonlar ile makine kontrol edilirken bizim çalışmamız dahilinde geliştirilen arayüzde kullanıcı serbest çizim

yapabilmekte ve çizilen şekillere uygun şekilde lazer markalama robotu kontrol edilebilmektedir. Söz konusu işleyişi gerçekleştirmek için kullanıcının robot kinematiği veya kontrol algoritmaları konusunda bilgisi olmasına gerek kalmamıştır.

Bir sonraki bölümde lazer markalama robotu ve sistemin işleyişi tanıtılmaktadır. Sonrasında geliştirilen insan-makine arayüzü algoritması açıklanmıştır. Kullanılan yörünge planlama algoritmalarının anlatılmasını takiben deneysel sonuçların değerlendirilmesi için kullanılan görüntü işleme algoritması ve sonuçlar sunulmuştur.

## 2. LAZER MARKALAMA ROBOT SİSTEMİ

Lazer markalama robotu sisteminin gizli robotunun kinematik gösterimi Şekil 3’te gösterilmiştir. C noktasının konumunu bulmak için Şekil 3’teki ifade edilen vektörel yöntemden yararlanılmıştır.



Şekil 3. Lazer mekanizmasının gizli robot gösterimi

Konum seviyesinde düz kinematik denklemleri uzuv boyutlarının aynı olmasından yararlanılarak C noktasının x ve y koordinatları mafsıl değişkenlerin fonksiyonu olarak Denklem 1 ve 2 ile ifade edilmiştir.

$$|\overrightarrow{A_0A}| = |\overrightarrow{A_0B}| = |\overrightarrow{AC}| = |\overrightarrow{BC}| = l$$

Paralel kenar teoremi:  $\theta_1 = \theta_4, \theta_2 = \theta_3$

$$\begin{aligned} \vec{r}_C &= \overrightarrow{A_0A} + \overrightarrow{AC} = \overrightarrow{A_0B} + \overrightarrow{BC} \\ &= l(e^{i\theta_1} + e^{i\theta_3}) = l(e^{i\theta_2} + e^{i\theta_4}) \end{aligned}$$

$$\vec{r}_C = l(e^{i\theta_1} + e^{i\theta_2})$$

$$x = l(\cos \theta_1 + \cos \theta_2) \quad (1)$$

$$y = l(\sin \theta_1 + \sin \theta_2) \quad (2)$$

Pozisyon seviyesinde ters kinematik denklemleri aşağıda belirtilen denklemler ile elde edilmiştir.:

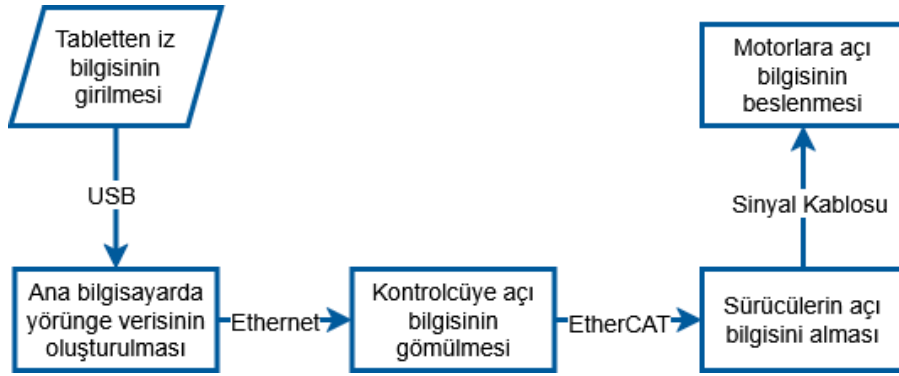
$$\vec{r}_C = x + iy, |\vec{r}_C| = r, \phi = \text{atan2}(y, x)$$

$$\mu = \text{acos}\left(\frac{a^2 + r^2 - b^2}{2ar}\right) = \text{acos}\left(\frac{r}{2l}\right)$$

$$\theta_1 = \phi + \mu \quad (3)$$

$$\theta_2 = \phi - \mu \quad (4)$$

Şekil 4'te insan-makine arayüzü eklenmiş sistemin bilgi akışı gösterilmektedir. USB kablosu ile ana bilgisayara bağlanan tablet



Şekil 4. Lazer sistemi bilgi akış şeması

### 3. İNSAN MAKİNE ARAYÜZÜ

İz oluşturma algoritmaları kullanarak Android Stüdyo üzerinde oluşturulan arayüzün temel çalışması Şekil 5'te gösterilen akış şemasında özetlenmiştir. Kısaca, kullanıcı serbest çizim modunda kalemle tablete değdirdiği andan kaldırdığı ana kadar nokta kaydı alır. Bu noktalar bir darbe aralığı olarak kaydedilir. Nokta kayıt frekansı kullanılan tablete göre değişmektedir ancak bu çalışmada 60 Hz'lik bir tablet kullanılmıştır [5]. Kullanıcı yeni bir dokunma yaptığı zaman yeni bir darbe aralığı başlar.

Serbest çizim modu dışında çizgi, üçgen ve dikdörtgen çizme modları vardır. Bu şekiller yeni darbe aralıkları olarak kaydedilir. Kullanıcı kaydet tuşuna bastığı zaman darbe aralıkları "json" dosya formatı ile tablet içine kaydedilir. Eğer kullanıcı elini kaldırmadan bir çizim yaparsa tek darbeleri bir veri seti oluşur.

Kaydedilen noktaların sayısı ve yoğunluğu, çizimin hızına ve çizimin eğriliğine göre değişiklik gösterir. Keskin köşeler ve dönüm

üzerinden çizim girdisi alınır. Kaydetme seçeneğinin ardından otomatik olarak gönderilen iz verisi ana bilgisayarda koşan Matlab'de oluşturulan yörünge planlama kodları ile konum bilgileri zamana bağlı veri seti haline getirilir. Daha sonra Denklem 3 ve 4'teki ters kinematik çözümler kullanılarak motorların beslenecek açı bilgileri elde edilir. Daha sonra bu bilgiler ethernet ile bağlı olan kontrolcüye gömülür. Kontrolcü açı bilgilerini girdi olarak motor sürücülerine iletir. Motor sürücülerinde gömülü olan kontrol algoritması kullanılarak motorlar sürülür.

noktaları daha sık nokta içerirken düz çizgiler daha az nokta içerir.

Öte yandan arayüzün kullanılabilirliğini arttırmak için serbest çizimin yanı sıra ileri alma, geri alma, silme, kaydetme ve basit geometrik şekil ekleme gibi fonksiyonlar eklenmiştir. Arayüzün ekran görüntüsü Şekil 6'da verilmiştir.

### 4. YÖRÜNGE PLANLAMA

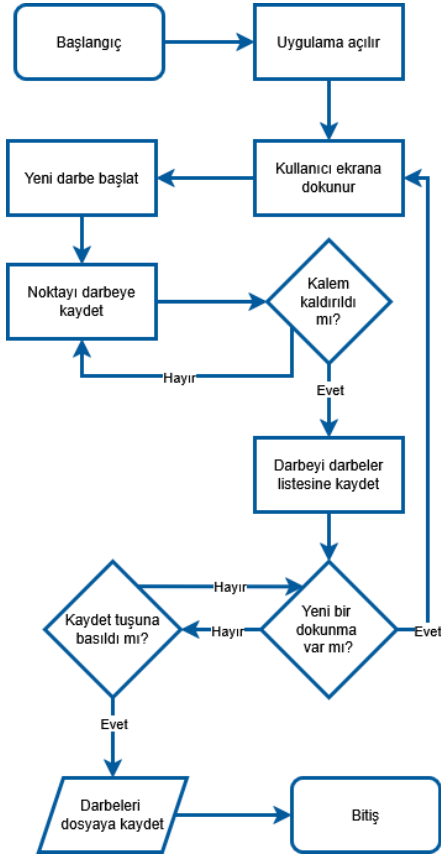
Yörünge planlama için üç farklı algoritma test edilmiş ve bunların iki tanesi son sistemde kullanılmıştır. Bu kısımda 3 yörünge hesaplama algoritmasının mantığı ve en sonunda sistemde nasıl kullanıldıkları anlatılmaktadır.

#### 4.1 S-Eğrisi ile Hız Yörüngesi Planlama

S-Eğrisi yörünge planlama s-şekli verilmek istenen değişkenin ikinci türevine sabit, süreksiz değerler vererek hesaplanan bir yörünge planlama yöntemidir. Standart bir s-eğrisi hız yörüngesi 7 aşamadan oluşur: (1) ivme artışı (pozitif jerk), (2) sabit ivme (0 jerk), (3) ivme azalışı (negatif jerk), (4) sabit hız (0 jerk, 0 ivme), (5) ivme azalışı, (negatif jerk),

(6) sabit ivme (0 jerk), (7) ivme artışı (pozitif jerk).

Bu çalışmada tercih edilen jerk değerleri gidilecek yola kıyasla düşük oldukları için üç aşamalı bir s-eğrisi hız yörünge planlama yöntemi kullanılmıştır: (1) ivme artışı (pozitif jerk), (2) ivme azalışı (negatif jerk), (3) ivme artışı (pozitif jerk). İki ivme artışı aşaması bir birim zaman, ivme azalış aşaması iki birim zaman sürmektedir. Bu tarz bir profil şekil 7’de görülebilir.



Şekil 5. İz kaydetme arayüz uygulamasının çalışmasını anlatan akış şeması

Hesaplama yapılırken iki boyutlu uzay yerine bir boyutlu uzay kullanılmaktadır. Öncelikle iki noktanın arasındaki mesafe ve gidiş yönüne göre açı bilgisi hesaplanır. Sonra bir birim zaman hesaplanır.

$$t_{sabit} = \left(\frac{L}{2j_{max}}\right)^{\frac{1}{3}} \quad (5)$$

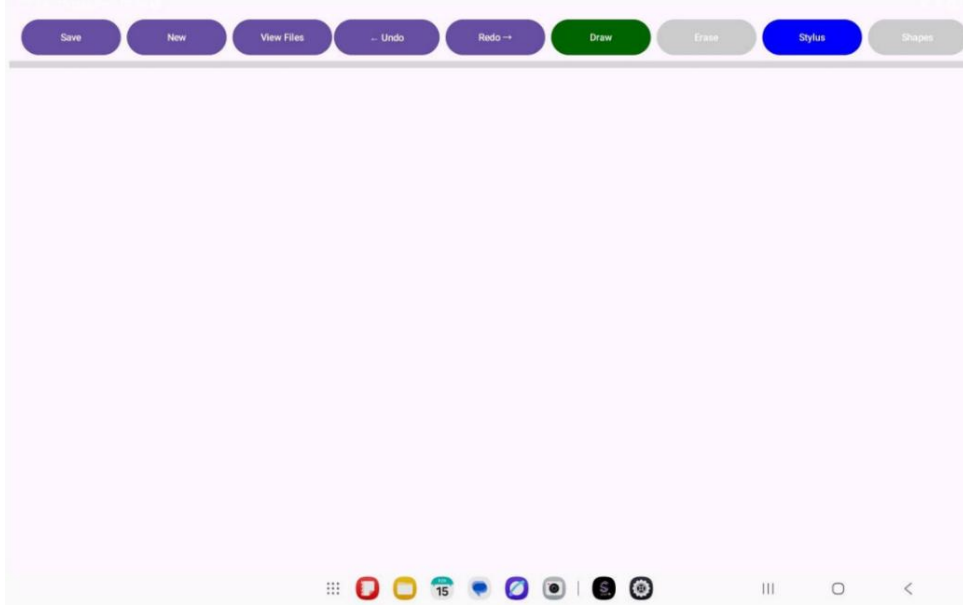
Hesaplanan zaman  $t_{sabit}$  tam olarak örnekleme zamanının katsayısı olmayacağı için yuvarlama yapılarak yeni zaman belirlenir  $t_{ysabit}$  ve yeni maksimum jerk  $j_{ymax}$  belirlenir.

$$j_{ymax} = \frac{L}{2t_{ysabit}^3} \quad (6)$$

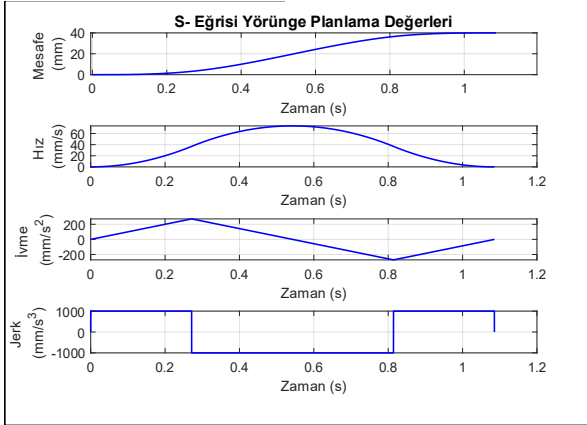
Daha sonra Denklem 5 ve 6’da hesaplanan zaman ve jerk verileri kullanılarak ivme, hız ve mesafe hesaplanır.

En sonunda bir boyutlu olarak hesaplanan mesafe, hız, ivme, jerk, açı ve başlangıç konumu bilgisi kullanılarak iki boyutlu bilgiye çevrilir.

Bu hesaplamalar iki nokta arasında yapılmaktadır. Üçgen ve dikdörtgenler için ardı ardına birden fazla defa kullanılır. Standart bir s-eğrisi grafiği Şekil 7’de verilmiştir.



Şekil 6. Kullanıcı giriş arayüzünün tablet üzerinden alınan görüntüsü



Şekil 7. Standart s-çizgisi pozisyon, hız, ivme ve jerk grafiği

#### 4.2 Dördüncü Derece Polinomla Konum Yörüngesi Planlama

Dördüncü derece polinom planlamada, konum için zamana bağlı dördüncü derece bir polinom kullanılır. Katsayılar göre farklı türleri vardır. Bu çalışmada kullanılan çeşidi aşağıda verilmiştir.

$$q(t) = q_i + v_i(t - t_i) + \left(\frac{1}{2}\right) a_i(t - t_i)^2 + \left(\frac{1}{6}\right) b_i(t - t_i)^3 + \left(\frac{1}{24}\right) c_i(t - t_i)^4 \quad (7)$$

$$t_i \leq t \leq t_{i+1}$$

Denklem 7 bu çalışmada iki boyutlu bir denklem olarak ele alınmıştır. Denklem 7 sadece bir aralıkta (yani segmentte) yani iki nokta arasında kullanılır. Bunun için her aralık için ayrı ayrı hesaplanır. Bu denklemin türevi

alınarak hız ve ivme formülleri türetilir. Denklem, 5 bilinmeyenli olduğu için 5 bilinen değer girilmesi gerekmektedir. Bu çalışmada bu 5 değer başlangıç pozisyonu, bitiş pozisyonu, başlangıç hızı, bitiş hızı ve başlangıç ivmesi olarak seçilmiştir. İlk ve son noktanın hızı sıfır olup, aradaki hızlar noktanın önceki ve sonraki noktaya olan mesafenin arada geçen zamana bölünmesi ile elde edilir yani merkezsiz fark hesaplaması yapılır. İlk noktanın ivmesi sıfır, sonraki noktaların ivmeleri bir önceki aralığın son ivmesi olarak hesaplanır. Bu hesaplanma sonucunda bilinmeyenlerin çözümleri aşağıdaki gibidir.

$$q_i = \text{Başlangıç pozisyonu}$$

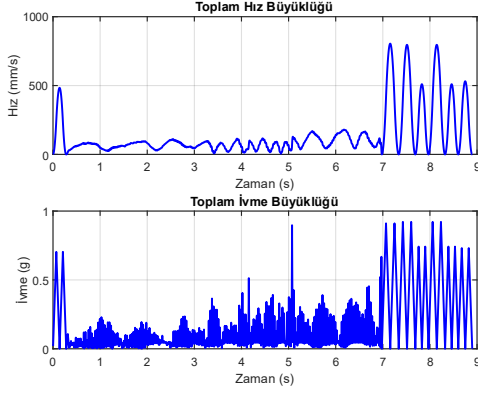
$$v_i = \text{Başlangıç hızı}$$

$$a_i = \text{Başlangıç ivmesi}$$

$$b_i = -\frac{24q_i - 24q_{i+1} + 6\tau v_{i+1} + 18\tau v_i + 6\tau^2 a_i}{\tau^3} \quad (8)$$

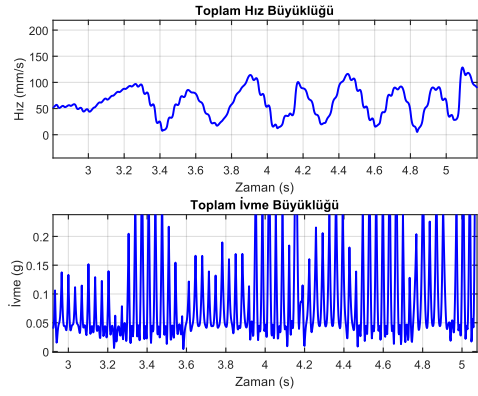
$$c_i = \frac{72q_i - 72q_{i+1} + 24\tau v_{i+1} + 48\tau v_i + 12\tau^2 a_i}{\tau^4} \quad (9)$$

Dördüncü dereceden polinom kullanılarak final ivmesi belirlenemeyeceği için son aralıkta beşinci derece polinom kullanılmıştır.



Şekil 8. Dördüncü derece polinom denklemleri ile elde edilen örnek bir hız ve ivme grafiği

Şekil 8’de hız ve özellikle ivme grafikleri olması gerekenden daha yüksek sıklıkta değişkenlik göstermektedir. Şekil 8’de hareket profilinin bir kısmının yakınlştırılmış hali görülebilir.



Şekil 9. Dördüncü derece polinom denklemleri ile elde edilen toplam hız ve ivme grafiklerinin büyütülmüş hali

Şekil 9’de hızda titreşim ve ivmede tepeleşmeler gözlemlenebilir. Bu davranış biçimi hem yapılan markalamanın doğruluğunu kötüleştirir hem de şeklin daha uzun sürede markalanmasına sebep olur. Bu yüzden dördüncü derece polinom yörünge planlama bu çalışmada tercih edilmemiştir.

### 4.3 Beşinci Derece Polinomla Konum Yörüngesi Planlama

Beşinci derece polinom planlamada konum için zamana bağlı beşinci derece bir polinom kullanılır. Katsayılarına göre farklı versiyonları vardır. Bu projede kullanılan versiyon aşağıda verilmiştir.

$$q(t) = q_i + v_i(t - t_i) + \left(\frac{1}{2}\right) a_i(t - t_i)^2 + \left(\frac{1}{6}\right) b_i(t - t_i)^3 + \left(\frac{1}{24}\right) c_i(t - t_i)^4 + \left(\frac{1}{120}\right) d_i(t - t_i)^5 \quad (10)$$

$$t_i \leq t \leq t_{i+1}$$

Denklem 10 bu çalışmada iki boyutlu bir denklem olarak ele alınmıştır. Denklem 7 sadece bir aralıkta (yani segmentte) yani iki nokta arasında kullanılır. Bunun için her aralık için ayrı ayrı hesaplanır. Bu denklemin türevi alınarak hız ve ivme formülleri türetilir.

Denklem 10, 6 bilinmeyenli olduğu için 6 bilinen değerin girilmesi gerekmektedir. Bu projede bu 6 değer için başlangıç pozisyonu, bitiş pozisyonu, başlangıç hızı, bitiş hızı ve başlangıç ivmesi ve bitiş ivmesi kullanılmıştır.

Denklem 10 bilinen değerler cinsinden çözüldüğü zaman bilinmeyenler için aşağıdaki denklemler elde edilir.

$$q_i = \text{Başlangıç pozisyonu}$$

$$v_i = \text{Başlangıç hızı}$$

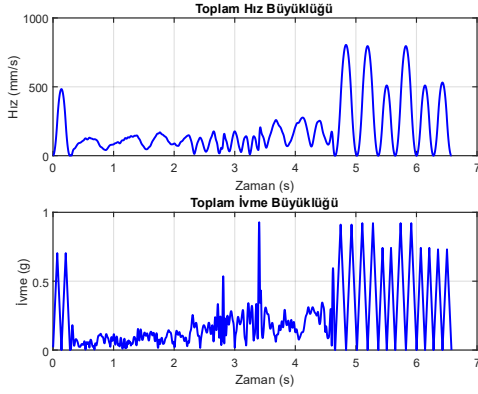
$$a_i = \text{Başlangıç ivmesi}$$

$$b_i = \frac{60q_i - 60q_{i+1} + 36v_i\tau + 24v_{i+1}\tau + 9a_i\tau^2 - 6a_{i+1}\tau^2}{-\tau^3} \quad (11)$$

$$c_i = \frac{12(30q_i - 30q_{i+1} + 16v_i\tau + 14v_{i+1}\tau + 3a_i\tau^2 - 2a_{i+1}\tau^2)}{\tau^4} \quad (12)$$

$$d_i = \frac{60(12q_i - 12q_{i+1} + 6v_i\tau + 6v_{i+1}\tau + a_i\tau^2 - a_{i+1}\tau^2)}{-\tau^5} \quad (13)$$

Şekil 8’nin oluşturulmasında kullanılan aynı iz bilgisi kullanılarak beşinci derece polinomlarla yörünge planlaması gerçekleştirilmiştir. Elde edilen sonuç Şekil 10’da verilmiştir. Şekil 10’daki hız profili incelendiğinde dördüncü derece polinomla elde edilen sonuçlara göre daha yumuşak hız değişimleri ve daha az sıklıkla ivme değişimleri olduğu tespit edilmiştir.



Şekil 10. Beşinci derece polinom denklemleri ile elde edilmiş hız ve ivme grafikleri

Dördüncü ve beşinci derece polinomlar 2 veya daha fazla nokta alır. Yapılan testlerde ortalama bir darbe aralığı yüzlerce noktadan oluşmaktadır.

İki yöntem için de nokta hesabından sonra örnekleme yapılır. Kullanılan lazer makinasının çalışma frekansı 500 Hz olduğu için beşinci derece için Denklem 10, dördüncü derece için Denklem 7 kullanılarak 2ms aralıklarla nokta eklenir. Nokta örneklemeden sonra oluşan noktalar motor sürücülerine iletilmek üzere kayıt altına zaman serisi olarak alınır.

#### 4.4 Genel Sistemin İşleyişinde Kullanılan Yörünge Planlaması

Şekil 11 yörünge planlama için takip edilen adımları bir akış şemasında tarif etmektedir.

Akış diyagramında da görülebileceği gibi öncelikle kaydedilen izden elde edilen noktalar yüklenip ölçekleme/merkezleme işlemi yapılır.

Merkezleme işlemi için şeklin x ve y yönünden merkezleri aşağıdaki formül kullanılarak belirlenir.

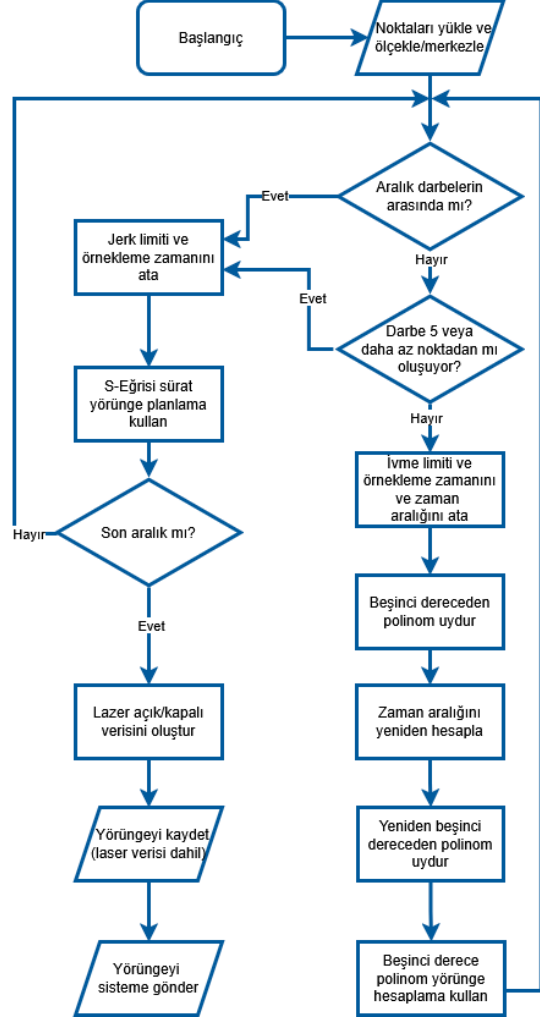
$$q_{orta} = \frac{\min(q) + \max(q)}{2} \quad (14)$$

$$q_{ortalanmış} = q - q_{orta} \quad (15)$$

Ölçekleme için şeklin x ve y koordinatları yönündeki uzunlukları belirlenir, istenen çalışma alanıyla kıyaslanır ve belirlenen ölçek ile çarpılır.

$$\text{ölçek} = \min\left(\frac{x_{\text{çalışma}}}{\max(x) - \min(x)}, \frac{y_{\text{çalışma}}}{\max(y) - \min(y)}\right) \quad (16)$$

$$q_{\text{yeni}} = q_{\text{ortalanmış}}(\text{ölçek}) \quad (17)$$



Şekil 11. Yörünge planlama akış şeması

Sonrasında aralığın darbelerin arasında olup olmadığı belirlenir. Darbe kullanıcının kendi eliyle çizdiği kısımlar anlamına gelmektedir. Darbe arası aralıklar başlangıç noktası ile ilk darbenin ilk noktası, bir darbenin son noktası ile sonraki darbenin ilk noktası, son darbenin son noktası ve başlangıç noktası arasındır.

Eğer ele alınan aralık darbelerin arasındaysa veya aralık beş noktadan daha az noktadan oluşuyorsa S-eğrisi hız yörünge planlama yöntemi kullanılır. Beş noktadan az olup olmadığının kontrol edilmesinin sebebi tableten çizgi, üçgen ve kare yazılabilmesi ve

S-çizimi hız yörünge planlamanın düz çizgiler için uygun olmasındandır.

Eğer aralık darbelerin arasında değilse beşinci derece yörünge hesaplama yöntemi kullanılır. Bunun için öncelikle ize rastgele bir aralık tamamlama süresi  $\tau$  verilir. Polinom uydurması yapılır ve maksimum ivme hesaplanır. Çıkan maksimum ivme istenilen ivmeye (bu çalışmada 1 g ivme seçilmiştir) gelmesi için  $\tau$  değiştirilir ve  $\tau$  2ms'nin katı olacak şekilde yuvarlanır. Yeniden polinom uydurulur ve beşinci derece yörünge hesaplamaya sokulur.

İşlemler yapılırken her bir yörünge hesaplamadan çıkan süre tutulur ve en sonunda lazer açık/kapalı verisini oluşturmak için bu veriler kullanılır. Lazer aç/kapa koşulları da oluşturulduktan sonra oluşturulan yörüngeler ters kinematik ile mafsal uzayı hareketlerine dönüştürülerek kontrol bilgisayarına gönderilir.

## 5. GÖRÜNTÜ İŞLEME

Lazer markalama işleminin doğruluğunu ve sistemin genel başarımını değerlendirebilmek için, kullanıcıdan alınan iz verisi ile elde edilen çıktı arasında görüntü işleme yöntemleriyle karşılaştırma yapılmıştır. Bu karşılaştırma işlemi, sistemin hem iz takibi doğruluğunu hem de uygulanan yörünge planlama algoritmalarının etkinliğini analiz etmek amacıyla gerçekleştirilmiştir.

Markalama işlemi tamamlandıktan sonra, lazerle ahşap yüzeye işlenmiş şeklin yüksek çözünürlüklü görüntüsü bir tarayıcı kullanılarak alınmıştır. Bu görüntü Python programlama dili ve OpenCV kütüphanesi kullanılarak işlenmiştir.

Veri karşılaştırmasının sağlıklı yapılabilmesi için görüntülerin boyut, yönelim ve konum açısından normalize edilmesi sağlanmıştır. Normalize işlemleri kapsamında, çizimlerin x ve y yönlerindeki merkezleri belirlenmiş, orijin etrafında merkezlenmiş ve ardından benzer ölçekleme oranlarıyla aynı koordinat düzlemine taşınmıştır.

Elde edilen görüntüler, kullanıcı çizimi ile birlikte gri tonlamalı (grayscale) biçime dönüştürülerek piksel başına 8-bit çözünürlükte ifade edilmiştir. Bu işlem sonucunda, sistemin markalama doğruluğu görsel ve sayısal olarak

doğrulanmış, iz takibindeki hata seviyeleri 3 ayrı yöntemle göre analiz edilmiştir.

Bu yöntemlerin sonuçları aşağıda özetlenmiştir:

**Hataların Karelerinin Ortalamasının Kökü (İng: RMSE):** İki görüntüde aynı konumda bulunan piksellerin parlaklık değerleri arasındaki kare farklarının ortalamasının karekökü alınarak hesaplanmıştır. Bu metrik, mutlak piksel farklarını temel alır ve hata miktarını 0–255 aralığında ifade eder.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (I_c(x_i, y_i) - I_g(x_i, y_i))^2} \quad (18)$$

$I_c(x_i, y_i)$ : Çıktının piksel parlaklığı  
 $I_g(x_i, y_i)$ : Girdinin piksel parlaklığı  
 $n$ : Toplam piksel sayısı

Bu çalışmada, beşinci derece polinom kullanarak yörünge planlama yapılmış bir çıktı kullanıldığında elde edilen RMSE değeri 23.53 olarak hesaplanmış ve yaklaşık %90.8'lik bir benzerlik oranı elde edilmiştir. Dördüncü derece polinom kullanıldığında ise sonuç % 90.91 benzerlik oranı elde edilmiştir. İki sonuç da birbirine oldukça yakındır.

**Yapısal Benzerlik İndeksi (SSIM):** İnsan görsel algısına daha uygun olan bu metot, parlaklık, kontrast ve yapısal bilgiyi dikkate alarak iki görüntü arasındaki benzerliği değerlendirir. SSIM değeri 0 ile 1 arasında değişmekte olup 1 tam benzerliği ifade eder.

$$SSIM(1,2) = \frac{(2\mu_1\mu_2 + c_1)(2\sigma_{12} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2)}$$

$$SSIM_{toplam} = \frac{1}{M} \sum_{i=1}^M SSIM(1_i, 2_i) \quad (19)$$

$$SSIM_{toplam} = \frac{1}{M} \sum_{i=1}^M SSIM(1_i, 2_i)$$

$\mu_1$ : çıktının piksel örneklem ortalaması

$\mu_2$ : girdinin piksel örneklem ortalaması

$\sigma_1^2$  ve  $\sigma_2^2$ : Çizimlerin örneklem varyansları

$\sigma_{12}$ : Çizimlerin örneklem kovaryansı

$c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$

$k_1 = 0.01$  ve  $k_2 = 0.03$  varsayılan katsayılar

Yörünge planlamada beşinci derece bir polinom kullanıldığında elde edilen SSIM değeri 0.9765 olup, %97.65'lik benzerliği ifade etmektedir. Dördüncü derece polinom kullanılarak yörünge planlaması yapılmış çıktıda ise bu değer % 98.20 olarak gözlenmektedir. İki sonuç da birbirine oldukça yakındır.

**Mekansal Hata (Spatial Mean Error):** Lazer çıktısında oluşan pikseller gruplandırılıp merkez çizgileri belirlenmiş ve bu noktaların tablet çizimine olan en yakın mesafeleri hesaplanarak ortalama hata bulunmuştur. Beşinci derece polinom ile planlanmış yörünge çıktısı kullanılarak hesaplanmış ortalama hata 39.9  $\mu\text{m}$  olarak ölçülmüştür. Bu değer, dördüncü derece polinom kullanılmış çıktı için 110.5  $\mu\text{m}$  olarak hesaplanmıştır. RMSE ve SSIM kullanılmasının aksine; mekansal hatada, çizgi kalınlığından bağımsız olarak geometrik doğruluk ölçtüğü için, beşinci derece polinomun dördüncü dereceye kıyasla daha başarılı bir sonuç verdiği gözlemlenmiştir.

Tablo 1. Farklı yörünge işleme algoritmalarına göre elde edilen görüntü işleme değerleri, (a) 5. Derece polinom kullanımı, (b) 4. Derece polinom kullanımı

Ölçütler	Değerler (a)	Değerler (b)
RMSE (0-255)	23.53	23.19
RMSE (%)	90.8	90.91
SSIM (%)	97.65	98.20
Ortalama Uzaklık ( $\mu\text{m}$ )	39.9	110.5

## 6. SONUÇLAR

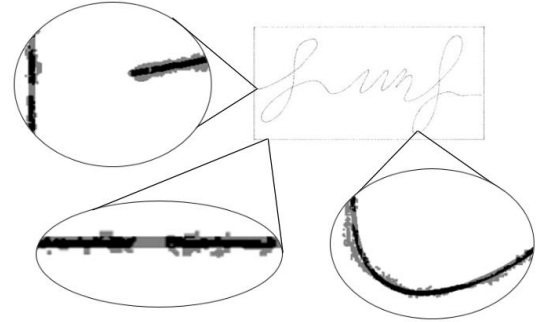
Bu çalışmada bir lazer markalama robot sistemine tablet tabanlı bir insan-makine arayüzü eklenmesi ve bu sistem için yörünge planlama algoritmalarının geliştirilmesi üzerine yapılan çalışmalar anlatılmıştır. Geliştirilen arayüz sayesinde kullanıcı, tablet üzerinden serbest çizimler yapabilmiş ve bu iz verisi robot tarafından başarılı bir şekilde markalanabilmiştir. İzi markalayabilmek için yörünge planlama algoritmaları oluşturulmuş ve uygulanmıştır.

Yörünge planlama aşamasında üç farklı algoritma test edilmiş olup, S-eğrisi ile hız yörünge planlama, dördüncü derece polinom ile konum yörüngesi planlama ve beşinci derece polinom yörünge planlama yöntemleri sistemde kullanılmıştır. Dördüncü derece polinom

yörünge planlama yöntemi, elde edilen hız ve ivme grafiklerindeki titreşim ve tepeleşmeler nedeniyle tercih edilmemiştir. S-eğrisi yöntemi, özellikle düz çizgiler için diğerlerine göre daha doğru sonuçlar verirken, beşinci derece polinom yöntemi daha karmaşık yörüngelerin oluşturulmasında kullanılmıştır.

Sistemin doğruluğunu ve genel performansını değerlendirmek amacıyla görüntü işleme yöntemleri kullanılmıştır. Şekil 12'de robot çıktısı ve insan girdisinin üst üste koyulmuş halini gösterilmektedir.

Lazerle markalanan şeklin yüksek çözünürlüklü görüntüsü ile kullanıcıdan alınan orijinal iz verisi karşılaştırılmıştır. Hataların Karelerinin Ortalamasının Kökü (RMSE), Yapısal Benzerlik İndeksi (SSIM) ve Mekansal Hata analizleri yapılmıştır. Beşinci derece polinom ile yörünge planlama sonucunda RMSE değeri 23.53 (%90.8 benzerlik), SSIM değeri 0.9765 (%97.65 benzerlik) ve ortalama konumlama hatası 39.9  $\mu\text{m}$  olarak hesaplanmıştır. Dördüncü derece yörünge planlama ile benzer sonuçlar elde edilmiştir ancak ortalama konum hatası kıyasında beşinci derece polinom ile yörünge planlamanın daha başarılı olduğu tespit edilmiştir.



Şekil 12. Görüntü işleme kullanılarak oluşturulan karşılaştırma görseli

Bu sonuçlar, geliştirilen insan-makine arayüzü ve yörünge planlama algoritmalarının lazer markalama robot sistemi üzerindeki etkinliğini ve doğruluğunu başarılı bir şekilde ortaya koymaktadır.

## 7. SEMBOLLER VE KISALTMALAR

<u>Sembol</u>	<u>Açıklama</u>
$t_{sabit}$	S-eğrisi yörünge hesaplamak için standart zaman, bir yörünge 4 $t_{sabit}$ sürer.
$L$	İki nokta arasındaki mesafe
$j_{max}$	Maksimum jerk
$q(t)$	Zamana bağlı mesafe, x ya da y yönünde olabilir.
$\tau$	Polinom yörüngede iki nokta arasında hesaplanan standart zaman.
$x_{çalışma}$	Çalışma alanının x yönünde uzunluğu
$y_{çalışma}$	Çalışma alanının y yönünde uzunluğu

## 8. KAYNAKLAR

[1] HMI: *Human-Machine Interface*. (2025, July 21). Inductive Automation. <https://inductiveautomation.com/resources/article/what-is-hmi>

[2] Uzunoğlu E., Özkahya M., vd (2019) Yüksek İvmeli Paralel Robotların Konumlama Hassasiyetinin Artırılmasına Yönelik Yöntemlerin Araştırılması, Uluslararası Katılımlı 17. Makina Teorisi Sempozyumu, İskenderun Teknik Üniversitesi, Hatay.

[3] Paksoy, E. (2021) *Enhancement of Trajectory Following Accuracy of High Acceleration Robots by Using Their Stiffness Properties*, Y. Lisans, İzmir İleri teknoloji Enstitüsü.

[4] Draw2Cut: Direct On-Material annotations for CNC milling. (n.d.-b). <https://arxiv.org/html/2501.18951v1>

[5] Samsung Galaxy Tab S6 Lite 2022 Price, Specifications, Comparison. (2025, Aug 10). GadgetByte Nepal. <https://www.gadgetbytenepal.com/product/samsung-galaxy-tab-s6-lite-2022/?>