

5 Serbestlik Dercele Bir Kaynak Robotunun Xbox 360 Pc Wireless Joystick İle Kumandası İçin Arayüz Programının Hazırlanması

Nülifer GÜNDOĞAN^{1*} (Orcid ID: 0000-0003-0519-2139), Cengiz DOĞAN² (Orcid ID: 0000-0002-1468-8462)

¹Adıyaman Üniversitesi, Besni Ali Erdemoğlu Meslek Yüksek Okulu, Adıyaman

²Harran Üniversitesi, Mühendislik Fakültesi, Şanlıurfa

*Sorumlu yazar (Corresponding author): ngundogan@adiyaman.edu.tr

Geliş Tarihi (Received): 26.06.2022

Kabul Tarihi (Accepted): 28.07.2022

Özet

Bu çalışmada, beş eksenli bir kaynak robotunun kontrolü için Microsoft Visual Studio C# 2010 editöründe bir arayüz yazılımı hazırlanmış olup, bu yazılım içerisinde PC Uyumlu Microsoft Xbox 360 Pc Wireless Joystick'i kullanabilmek için program bloğu hazırlanmıştır. Bu yazılım ile kaynak uygulaması uzaktan kumanda edilebilmekte olup, kullanılan Joystick kaynak robotu el terminaline dönüştürülmüştür. Bunun için ilk olarak Xbox 360 Kablosuz Joystick ve alıcısı için gerekli olan sürücüler bilgisayara yüklenmiş olup, Microsoft Visual Studio C# programında gerekli olan kütüphaneler de hazırlanan arayüze eklenmiştir. Daha sonra hazırlanan bu arayüzde kaynak uygulaması ve kaynak robotu için Joystick'te kullanılacak butonlara ait kod blokları C# editöründe hazırlanmıştır. Bu kod blokları ile Joystick'te kullanılan butonların bir kısmı ile robotun X, Y, Z, A ve B esenlerine uzaktan kumanda ile hareket verilebilmektedir. Bunun yanı sıra bu butonların bir kısmı ile de kaynak makinesini açma kapama, hareket koordinatlarını kaydetme, robotu durdurup çalıştırma ve acil stop gibi eylemler yapılabilmektedir. Robotlu kaynak uygulamalarında Joystick kullanımı ile hem kaynakçının sağlığı korunmakta hem de kullanılan diğer el terminaline göre maliyetler oldukça azalmaktadır. Ayrıca yazılımın milli olması ile ülkemiz teknolojisine yenilik getirmesi ve katkıda bulunması amaçlanmıştır.

Anahtar Kelimeler: Joystick, Microsoft Visual Studio C#, arayüz, robotlu kaynak

Preparing The Interface Program For Controlling 5 Degree of Freedom Welding Robot With The Xbox 360 Pc Wireless Joystick

Abstract

In this study, an interface software was prepared in Microsoft Visual Studio C# 2010 editor for the control of a five-axis welding robot, and a program block was prepared to use the PC Compatible Microsoft Xbox 360 Pc Wireless Joystick in this software. With this software, the welding application can be controlled remotely, and the Joystick welding robot used has been transformed into a hand terminal. For this, the drivers required for the Xbox 360 Wireless Joystick and its receiver were first installed on the computer, and the libraries required in the Microsoft Visual Studio C# program were also added to the prepared interface. In this interface, which was prepared later, the code blocks for the buttons to be used on the joystick for the welding application and welding robot were prepared in the C # editor. With these code blocks, some of the buttons used on the joystick and the robot's X, Y, Z, A and B elements can be moved with a remote control. In addition, some of these buttons can perform actions such as turning the welding machine on and off, recording the movement coordinates, stopping and starting the robot, and emergency stop. With the use of joysticks in robotic welding applications, the health of the welder is protected and the costs are considerably reduced compared to other hand terminals used. In addition, it is aimed to bring innovation and contribute to the technology of our country with the nationality of the software.

Keywords: Joystick, Microsoft Visual Studio C#, interface, robotic weldin

GİRİŞ

Günümüzde üretimde verimliliği arttırmak, maliyeti azaltma, seri üretimi arttırmanın yanı sıra kaliteyi yükseltmek, daha temiz, güvenilir ve ağır çalışma koşulları hafifletilmiş iş imkânları sağlamak üzere birçok üretim alanında hızla kullanılmaya başlanan robotlar, kaynak teknolojisinde de geniş bir uygulama sahası bulmuştur. Kaynak robotları İlk olarak 1970 yılında elektrik direnç nokta kaynağında kullanılmıştır. Daha önceden robotun hareketlerinde kullanılan hidrolik tahrikli sistemler kaynaktan beklenen yüksek hassasiyet gibi konularda yetersiz ve kısıtlayıcı olmuştur. Bunun üzerine bir 10 yıl gecikme ile elektrik tahrikli kaynak robotunun üretimi ile ark kaynak yöntemleri de kaynak robotları ile yapılabilir olmuştur (Türker, 2015). Sanayide imalat ve montaj vb. gibi işlemlerde birçok kaynak uygulamaları yapılmaktadır. Bundan dolayı yapılan kaynağın kalitesi, verimliliği, esnekliği, kaynağın yapılma kolaylığı ve süresi gibi faktörler üreticiler arasında rekabeti arttırmaktadır. Tüm bu faktörler ve bunlar gibi birçok etkenlerden dolayı kaynak işlemlerinde de robota dayalı kaynak uygulamalarına olan rağbeti arttırmaktadır. Robotlu kaynak sistemlerinde yapılan kaynak ile kaynak telinin bir buçuk katında tekrar edilebilir hassasiyetteki kaynak kalitesine sahip birleştirmeler elde edilir. Robotlu kaynak uygulamalarında, elle yapılan kaynağa göre dört kata kadar daha hızlı kaynak uygulaması yapılabilir (Eğilmez, 2005). Robotlu kaynak uygulamalarında en çok kullanılan yöntemlerden biri punta kaynağı olup, MIG/MAG, TIG ve plazma kaynakları da robotlar ile başarıyla yapılabilir (Eğilmez, 2005). Kullanılan ark kaynak yöntemlerinde kaynağın devamlı olarak yapılabilirdiği MIG-MAG ve TIG gazaltı kaynak yöntemleri endüstriyel robotlar

tarafından çok iyi yapılabilir. Özellikle kaynak operatörlerinin bazen çok tehlikeli ve zor çalışma şartlarında kaynak yapıyor olmaları veya kaynak operatörünün erişemeyeceği çalışma ortamlarında kaynak yapılmasının istenmesi, yapılan kaynağın kalitesi, verimliliği, esnekliği, kaynağın yapılma kolaylığı ve süresi gibi faktörler kaynak robotlarının kullanımını önemli ölçüde teşvik etmiştir (Türker, 2015). Robotlu kaynak sistemlerinde yapılan kaynak ile kaynak telinin bir buçuk katında tekrar edilebilir hassasiyetteki kaynak kalitesine sahip birleştirmeler elde edilir. Robotlu kaynak uygulamalarında, elle yapılan kaynağa göre dört kata kadar daha hızlı kaynak uygulaması yapılabilir (Eğilmez, 2005). Bir otomatik kaynak robotu; robot gövdesi, kontrol birimi, bilgisayar, kaynak makinesi ve joystickten oluşmaktadır. Kaynak robotu kontrolünde kaynak torcuna yörunge boyunca hareket verebilmek için kontrol ünitelerine ve programlara ihtiyaç vardır. Kontrol kartları ve yazılımlar sayesinde elektrik sinyalleri ve elektronik parçalardan elde edilen veriler işlenerek çıkış birimlerine gönderilir ve kaynak otomatik olarak istenilen yörungede yapılır. Robotlu kaynak işlemi esnasında takip edilen kaynak yörungesi, öğretim (teaching) programı ile belirlenir. Operatör, robotun kaynak yapılmak istenen yörungedeki koordinatları ve bu koordinatlarda uygun olan kaynak parametrelerini kaynak robotu yazılımı yardımıyla ayarlar. Ayarlanmış olan bu koordinatlarda robotun kontrol ünitesine kaynak yapılacak olan yörunge öğretilmiş olur. Kaynak için gerekli olan torç ilerlemesi, hızı, moment gibi değerler de ayarlandıktan sonra kaynak yapılmaya hazır hale getirilir (Türker, 2015). Kaynak robotu uygulamalarında kontrolör, programlanabilen ve veri giriş/çıkışı yapılabilen farklı uygulama

pencerelerine sahip operatör ara yüzüne sahiptir. Bu kontrolörlerin kullanım açısından en kolayı öğretim / programlama ara yüzü ve ürün / operatör

ara yüzüdür, diğer adıyla el terminalidir. El terminali (ing: teach pendant), manipülatörün uzaktan kontrolünü sağlayan bir araçtır (Şekil 1).



Şekil 1. Teach Pendant, SmartPAD, Öğretme Paneli, El Terminali, Kumanda ve Programlama El Cihazı

YÖNTEM

Bu çalışmada Şekil 2’de gösterilen kaynak robotunun kontrolü

için Microsoft Visual Studio C# 2010 programında hazırlanmış bir arayüz programı ile kaynak yapılmaktadır.



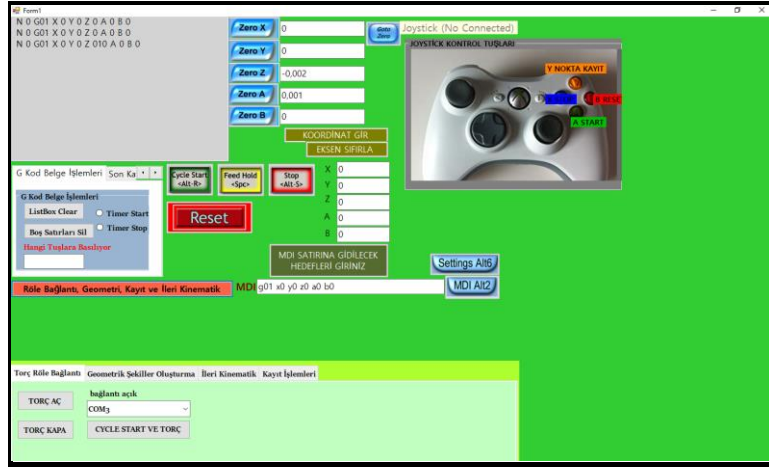
Şekil 2. Kaynak robotu gövdesi

Bu arayüz, robota ait kaynak yörüngesini kinematik hesaplarını yaparak istenilen koordinat veri girişlerine göre oluşturma,

kontrol ünitesine bu verileri göndererek motorlara hareket verme işlemi, kaynak makinesini elektronik ve elektriksel

bağlantılarını yönetip kaynağı hem otomatik hem de manuel aç/kapa işlemleri ve hem robotun hem de kaynağın uzaktan kontrolünü yapabilmeye gibi işlemleri barındıran bir yazılımdır. Microsoft Visual Studio C# editöründe hazırlanan arayüz programı içerisinde Mach3 fonksiyonları kullanılmıştır. Bu arayüz programı kullanılırken Mach3 programı da arka planda çalışmaktadır.

Bu programların hazırlanmasında Mach3 programına ait nesnelere kullanmak için bu nesnelere ait OEM numaralarının değerlerinden yararlanılmıştır. Ayrıca hazırlanan bu OEM numaralarını kod halinde C# editöründe ve makro yazılımlarında script olarak kullanabilmek için bunlara ait Visual Basic scriptlerinden faydalanılmıştır (Şekil 3).



Şekil 3. Microsoft Visual Studio C# editöründe hazırlanan arayüz programı

Hazırlanan bu arayüz ile tüm bu işlemler bilgisayara bağlı bir klavye tarafından yapılabilirken PC Uyumlu Microsoft Xbox 360 Kablosuz Pc Wireless Joystick ve alıcısı (Şekil 4) tarafından uzaktan kontrollü kablosuz bağlantı ile yapılabilmektedir. Bir kaynak el terminalinin yapabileceği işlemlerin birçoğu bu arayüz yazılımı ile PC

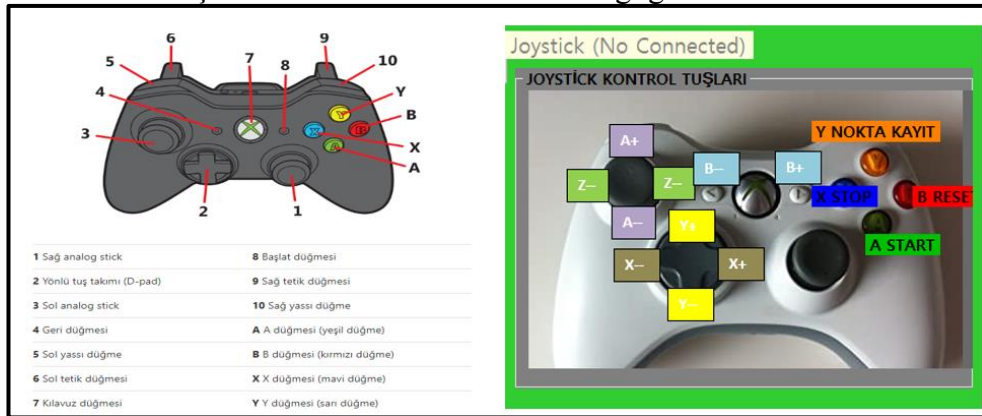
Uyumlu Microsoft Xbox 360 Kablosuz Pc Wireless Joystick tarafından yapılabilmekte olup hazırlanan bu arayüz geliştirilebilir kolaylığı olan bir yazılımdır. Bunu yanı sıra maliyet açısından Xbox 360 Kablosuz Pc Wireless Joystick, el terminaline kıyasla çok daha ucuzdur.



Şekil 4. PC Uyumlu Microsoft Xbox 360 Kablosuz Pc Wireless Joystick ve alıcısı

Microsoft Visual Studio C# programı ile hazırlanan arayüz programı kaynak robotuna ait 3 birimi aynı anda kontrol edebilmektedir. Birincisi robota ait eksen hareketlerini sağlayan Mach3 kontrol kartını ve programını yönetebilmekte, ikincisi kaynak uygulamasını yöneten Arduino kartının yazılımını ve kaynağa ait hız, ivme ve ilerleme gibi parametreleri ayarlayabilme, üçüncüsü ise Joystick kumandasına ait kontrol yazılımıdır. İlk olarak PC Uyumlu Microsoft Xbox 360 Kablosuz Pc Wireless Joystick ve alıcısı için gerekli olan sürücüler (driver) PC’de Aygıt denetimi penceresinden yüklenir. Bu C# editöründe hazırlanmış arayüz programı içerisinde robotun kontrolü için gerekli tüm kod blokları, Joystick kumanda için hazırlanan kod bloklarının içerisinde hazırlanmıştır. Hazırlanan bu

kodlar ile robot ve kaynak kontrolü, hem manuel hem de Joystick ile uzaktan kumanda ile yapılabilir. Şekil 5’de de görüldüğü üzere Microsoft Visual Studio C# programı ile hazırlanan arayüz programında, kullanılacak Xbox Joystick’e ait her tuş bir eylemi yerine getirmek için kodlanmıştır. Eğer kaynak manuel yapılacaksa arayüz programının Xbox resminin üzerinde “*Joystick (NO Connected)*” ifadesi yazar. Eğer kaynak Xbox ile uzaktan kumandalı yapılacaksa bu ifade “*Joystick is Connected*” olarak görüntülenir. Hazırlanan bu arayüzdeki kodların bir kısmı robotun uzaktan kontrolünü yapabilmek için arayüz programı içinde Joystick program bloğu içinde Joystick’e ait butonlara atanmıştır. Çizelge 1’de, Joystick’in hangi butonlarına hangi görevlerin atandığı gösterilmektedir.



(a)

```
private Ojw.CJoystick m_CJoy = new Ojw.CJoystick(Ojw.CJoystick.ID_0); // Joystick beyanı
private Ojw.CTimer m_CTim_Joystick = new Ojw.CTimer(); // Oyun çubuğunun bağlantısını periyodik olarak kontrol etmek için zamanlayıcı

private void FJoystick_Check_Alive()
{
    Joystick Check
}

string DpadEnglish = "";

private void FJoystick_Check_Data()
{
    if (m_CJoy.IsDown(Ojw.CJoystick.PadKey.Button7) != true & m_CJoy.IsDown(Ojw.CJoystick.PadKey.Button8) != true & m_CJoy.IsDown(Ojw.CJoystick.PadKey.POVRight) == true & m_CJoy.IsDown(Ojw.CJoystick.PadKey.POVLeft) != true)
    {
        GetMach3Instance();
        DpadEnglish = "Left ";
        _mInst.DoOEMButton(308); //308 Dem list de X-- butonun değeridir.
        txtDroX.Text = _mInst.GetOEMDRO(800).ToString(); //800 Dem list de X eksenini değeridir.
    }
    else if (m_CJoy.IsDown(Ojw.CJoystick.PadKey.POVLeft) == true & m_CJoy.IsDown(Ojw.CJoystick.PadKey.POVRight) != true)
    {
        GetMach3Instance();
        _mInst.DoOEMButton(307); //308 Dem list de X-- butonun değeridir.
        txtDroX.Text = _mInst.GetOEMDRO(800).ToString(); //800 Dem list de X eksenini değeridir.
    }
}
```

(b)

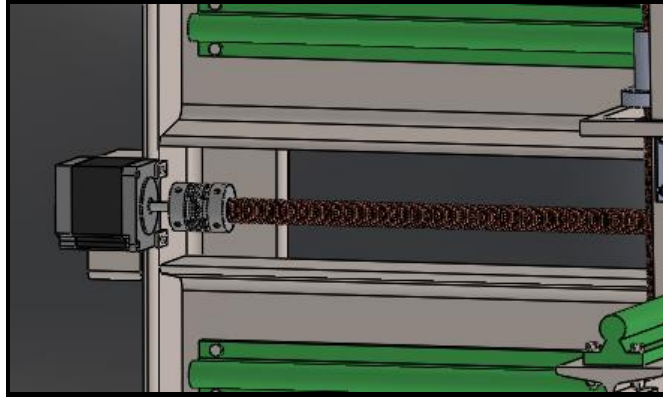
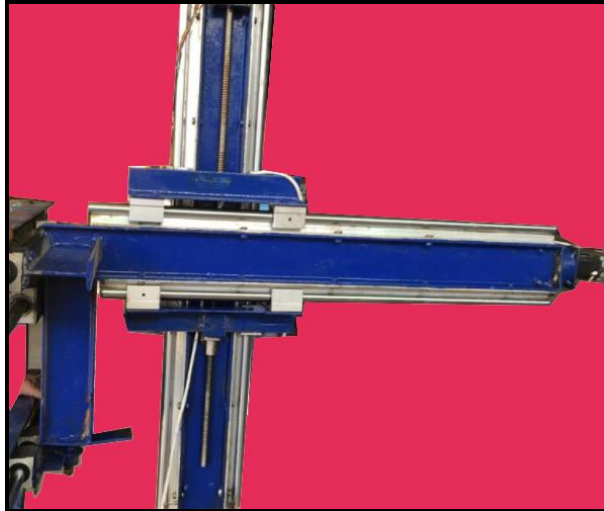
Şekil 5. C# editöründe yazılan arayüz programında, Joystick ile kontrol için (a) Form görüntüsü ve Joystick tuşları (b) hazırlanmış kodların bir kısmı

Çizelge 1. Robota ait sabit eklem uzunlukları

	Xbox üzerindeki tuş	Açıklama
X++	Sağ Yön Ok tuşu	Motor X eksenini sağa hareket yönü
X--	Sol Yön Ok tuşu	Motor X eksenini sola hareket yönü
Y++	Yukarı Yön Ok tuşu	Motor Y eksenini yukarı hareket yönü
Y--	Aşağı Yön Ok tuşu	Motor Y eksenini aşağı hareket yönü
Z++	Sol Analog Stick Sola	Motor Z eksenini sağa hareket yönü
Z--	Sol Analog Stick Sağa	Motor Z eksenini sola hareket yönü
A++	Sol Analog Stick Yukarı	Motor A eksenini sağa hareket yönü
A--	Sol Analog Stick Aşağı	Motor A eksenini sola hareket yönü
B++	Başlat Düğmesi	Motor B eksenini sağa hareket yönü
B—	Geri Düğmesi	Motor B eksenini sola hareket yönü
STOP	X Düğmesi	Robotun hareketini durdurur.
KAYIT	Y Düğmesi	Robotun hareketlerini nokta halinde kaydeder
START	A Düğmesi	Robotun hareketini başlatır
RESET	B Düğmesi	Acil stop
TORC AÇ	Sol Tetik Düğmesi	Kaynak Makinesi Elektrik Akımı Aç
TORC KAPA	Sağ Tetik Düğmesi	Kaynak Makinesi Elektrik Akımı Kapat

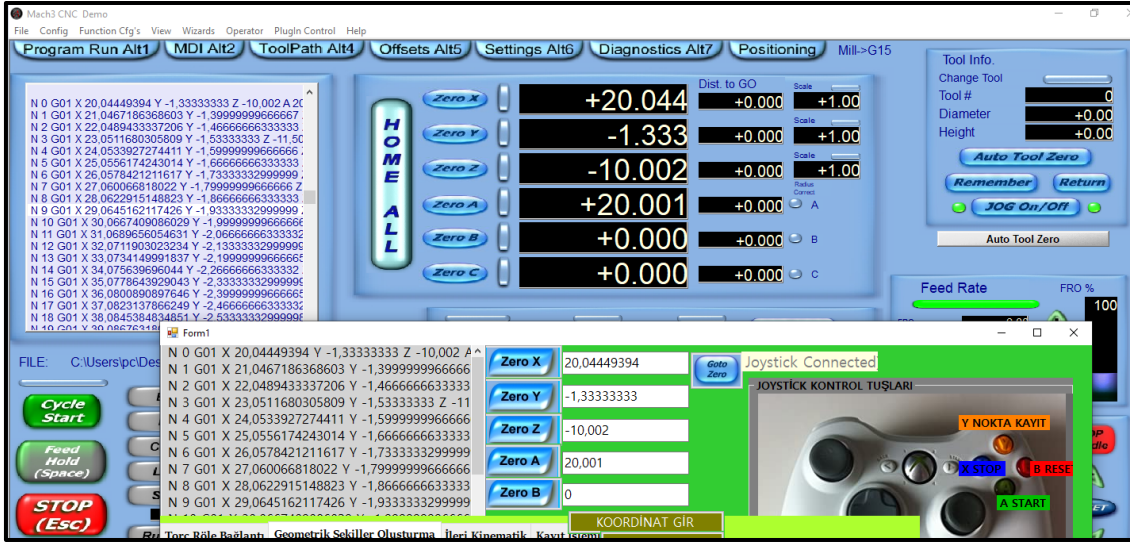
Kaynak mekanizmasında kollara X, Y, Z eksenlerinde lineer hareket vermek için bir step motor-vidalı mil

mekanizması tasarımı ve imalatı yapılmıştır.

**Şekil 6.** Step motor ile bilyalı somun, vidalı mil ve rulman bağlantısı**Şekil 7.** Robota ait kızak, lineer rulman bağlantıları

Bu mekanizma da motora verilen dönme hareketi, motorun kaplinle bağlı olduğu vidalı mil mekanizması ile ileri-geri harekete dönüştürülür (Şekil 6-7). Bu sayede kollara ileri geri hareket verilmiş olur. Dolayısıyla hazırlanan arayüz programda robotun yataydaki kolunun X ekseninde sağa hareketi için Joystick'te Sağ Yön Ok tuşuna, sola hareketi için Sol Yön Ok tuşuna basılır. Robotun dikeydeki kolunun yukarı Y eksenindeki hareketi için Joystick'te Yukarı Yön Ok tuşuna, aşağı hareketi için Aşağı Yön Ok tuşuna basılır. Robotun Z eksenindeki kolunun sağa sola hareketi için Joystick'te Sol Analog Stick Sola ve Sol Analog Stick Sağa tuşlarına basılır. Kaynak robotunun dikeydeki kolunun uç kısmına, torcun hem ileri geri hem de zig-zag kaynak dikişi sağlayabilen motorlar bağlıdır. Bu motorların eksenleri A ve B olarak isimlendirilmiştir. A eksenindeki motora Joystick ile pozitif (+) ve negatif (-) yönde dönme hareketleri verebilmek için Joystick'te Sol Analog Stick Yukarı ve Sol Analog Stick Aşağı tuşlarına basılır. B eksenindeki motora Joystick ile pozitif (+) ve negatif (-) yönde dönme hareketleri verebilmek için ise Joystick'te Başlat Düğmesi ve Geri Düğmesi tuşlarına basılır. Robota tüm bu hareketleri verirken aynı zamanda yapılan hareketlerin koordinatlarını nokta nokta bir text belgesine kayıt etmek için Joystick'te Y düğmesi kullanılmalıdır (arayüzde belge otomatik kayıt için hazırlanan kodların bir kısmı "d:\\\" + *Convert.ToString(i)* + ".txt";

i++"). Bu text belgesi, hazırlanan arayüzde her belgeye kayıt edildiğinde 0'dan başlanarak sayı olarak verilmiş belge isminin sayısını otomatik olarak birer birer arttırarak isimlendirme yapıp D:/ dizinine kayıt edilecek şekilde kodları yazılmıştır. Dolayısıyla belge kayıt için Joystick'te Y tuşuna basıldığında numaralandırılmış şekilde belge ismi ataması yapılarak kaydedilmek istenen her belge otomatik isimle D:/ dizinine kaydedilir. Belgeye ait isimlendirme 0'dan başlayan sayı olarak verilmiş olup (d:/0.txt gibi) bir başka hazırlanan belgenin kayıt isminde ise bu numara otomatik olarak arttırılacak şekilde ayarlanmıştır ((d:/1.txt gibi). Hazırlanan arayüz programında kaynak uygulaması esnasında herhangi bir sebep ile robotun hareketlerini durdurmak veya durduğu yerden tekrar başlatmak için Joystick'te X ve A Düğmesine basılmalıdır. Acil stop için ise Joystick'te B Düğmesine basılmalıdır. Kaynak makinesine manuel olarak müdahalede bulunup açıp kapatmak için ise Joystick'te Sol Tetik Düğmesi ve Sağ Tetik Düğmesine basılmalıdır. Tüm bu işlemlerin dışında Joystick'in diğer tuşlarına herhangi bir görev atamak istersek bu görevin C#'daki kodlarını program bloğu olarak hazırlayabiliriz. Bu arayüz programı ile motorların bulunduğu eksen değerleri eş zamanlı olarak G kodları halinde kaydedilip text dosyası halinde kayıt edilebilmekte ve motorlar hareket ettirilebilmektedir (Şekil 8).



Şekil 8. C# editöründe yazılan arayüz programında, Joystick ile kontrol için hazırlanmış G kodlarının bir kısmı ve Mach3 programındaki görüntüsü

Aşağıdaki kodlar Joystick'teki Y tuşu ile yapılan nokta halinde koordinat kayıtlarının yapıldığı program ve X tuşu ile yapılan Mach3 dosya yükleme program bloğuna ait C# editöründe hazırlanmış kodlardır:

```
if
(m_CJoy.IsDown(Ojw.CJoystick.PadKey.Button4) == true)
// Y Joystick tuşu Mach3 Nokta Nokta Kayıt
{lblY.ForeColor=Color.Red;
c++;
double[] xx31 = new double[100];
double[] yy31 = new double[100];
double[] zz31 = new double[100];
double[] aa31 = new double[100];
double[] bb31 = new double[100];
GetMach3Instance(); if (_mInst != null)
{x31[0]=Convert.ToDouble(txtDroX.Text);
x31[c]=Convert.ToDouble(txtDroX.Text);
y31[0]=Convert.ToDouble(txtDroY.Text);
y31[c]=Convert.ToDouble(txtDroY.Text);
```

```
z31[0]=Convert.ToDouble(txtDroZ.Text);
z31[c]=Convert.ToDouble(txtDroZ.Text);
a31[0]=Convert.ToDouble(txtDroA.Text);
a31[c]=Convert.ToDouble(txtDroA.Text);
b31[0]=Convert.ToDouble(txtDroB.Text);
b31[c]=Convert.ToDouble(txtDroB.Text);
deltax31[c] = (x31[c] - x31[c - 1]) / 10;
deltay31[c] = (y31[c] - y31[c - 1]) / 10;
deltaz31[c] = (z31[c] - z31[c - 1]) / 10;
deltaa31[c] = (a31[c] - a31[c - 1]) / 10;
deltab31[c] = (b31[c] - b31[c - 1]) / 10;
bb31[1] = 0.1;
for (ii = 0; ii <= 10; ii++)
{xx31[ii] = deltax31[c] * ii + x31[c - 1];
yy31[ii] = deltay31[c] * ii + y31[c - 1];
zz31[ii] = deltax31[c] * ii + z31[c - 1];
aa31[ii] = deltaa31[c] * ii + a31[c - 1];
// bb31[ii] = deltax31[c] * ii + b31[c - 1];
if (ii > 0)
{ if (bb31[ii - 1] < 0)
{bb31[ii] = -1 * bb31[ii - 1]; };
```



```

if(bb31[ii-1]>0){ bb31[ii] = 1 * bb31[ii
- 1] - 0.2;};
};
listBox4.Items.Add(Convert.ToString(xx
31[ii]));
listBox1.Items.Add(" N " + "0" + " G01"
+ " X " + Convert.ToString(xx31[ii]) +
" Y " + Convert.ToString(yy31[ii]) + " Z
"+ Convert.ToString(zz31[ii]) + " A " +
Convert.ToString(aa31[ii]) + " B " +
Convert.ToString(bb31[ii]) + " f1000");}
label25.Text = Convert.ToString(c); }
}
else
{lblY.ForeColor = Color.Black;}
if
(m_CJoy.IsDown(Ojw.CJoystick.PadKe
y.Button3) == true)
// X Joystick tuşu Mach3 File Load
{
GetMach3Instance();timer4.Enabled =
true;
if (_mInst != null)
{ _mInst.DoOEMButton(169);
String[] sitesDizi2=new
String[listBox1.Items.Count];
listBox1.Items.CopyTo(sitesDizi2, 0);
System.IO.File.WriteAllLines("C:\\User
s\\pc\\Desktop\\"+ Convert.ToString(n)
+ ".txt", sitesDizi2);
textBox4.Text=
"C:\\Users\\pc\\Desktop\\"+Convert.
ToString(n) + ".txt";
mInst.LoadFile(textBox4.Text);
textBox4.Text=_mInst.FileName();
listBox2.Items.Add(textBox4.Text);
}

```

Tüm bunların dışında bu çalışmada kullanılan Mach3 programı ve kontrol kartı tarafından desteklenen el çarkları da kullanılmaktadır. Buna rağmen bizim çalışmamızda sadece Mach3 programı kullanılmamış olup kaynak kontrolü için Aurdunio gibi başka kartlar ve yazılımları da C# editöründe yazılan arayüz programı içerisinde yönetilmekte olduğundan ve bu el çarkları sadece

Mach3 programında çalışıp C#'da hazırlanan arayüz programında çalışmadığından bu çalışmadaki kaynak robotunda kullanımı yeterli olmamaktadır.

Robotlu kaynak sistemlerinde kablosuz uzaktan erişimli kaynak işlemlerinde, hem kaynak operatörünün güvenliği sağlanmakta ve sağlığı korunmakta, hem de çalışma ortamının da oluşabilecek tehlikelere karşı önlem alınması sağlanmaktadır. Bu çalışmada, kaynak yörüngesinde hareket sağlayan tüm görevler Visual Studio 2010 C# programında hazırlanan arayüz programı ile oluşturularak G Kodlarını metin belgesi halinde çevrilmek amacıyla Mach3 programını destekleyen bir kontrol kartı, her bir eksene ait step motor ve motorlara hareket verecek olan sürücüler kullanılmaktadır. Kaynak robotunun hareketi için hazırlanan arayüz programında, aynı zamanda Mach3 programına ait scriptler de kullanılmıştır. Aslında hazırlanan bu arayüz sadece robot kaynak uygulamasında değil tüm robotlu uygulamalara uyarlanabilecek nitelikte geliştirilebilir bir program olma özelliğine de sahiptir.

SONUÇLAR

Bu çalışmada tasarımı ve imalatı yapılmış kaynak robotunun kollarına, torç taşıyıcı ünitesine hareket verebilmek ve kaynak ile ilgili bir takım işlemleri yapabilmek için Microsoft C# editöründe bir arayüz programı hazırlanmıştır. Bu arayüz programı içerisinde, kaynak yöntemini uzaktan kontrol edebilmek, kaynak operatörünün sağlığını ve güvenliğini sağlamak amacıyla bir Joystick program kod bloğu hazırlanmıştır. Bu sayede operatörün kaynaktan oluşan zararlı gazların etkisinden ve olabilecek herhangi bir tehlikeden korunması amaçlanmıştır. Ayrıca birçok robotlu uygulama için

oldukça az maliyetli Xbox 360 Joystick gibi bir kumanda kullanılarak, bu kumandanın kendimize ait olan yazılımı oluşturulmuştur. Bu yazılım ve Xbox 360 Joystick kumanda, diğer kaynak makinelerinde oldukça maliyetli olan el terminallerinin yapabileceği birçok uygulamayı yapabilecek ve geliştirilebilir niteliğinde olan bir yazılımdır. Bu anlamda da ülkemiz teknolojisine katkı sağlayabilecek nitelikte bir yazılım olma özelliğine sahiptir. Bu çalışmada yazılan arayüz ile verilen yörüngede kaynak robotunun uzaktan kontrollü olarak kaynak yapması sağlanmıştır.

KAYNAKLAR

- Adar, N.G., Ören, H., Kozan, R. 2013. 5 Serbestlik dereceli robot kolunun modellenmesi ve kontrolü. SAÜ, Fen Bil. Der., 17(1): 155-160.
- Akpınar, Ö. 2008. Depolama amaçlı görüntü işleme tabanlı bir kartezyen robot tasarımı. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya, 85s.
- Akyüz, İ.H., Bingül, Z., Kizir, S. 2012. Tek eklemlili esnek robot kolunun modellenmesi, tasarımı ve bulanık mantık ile kontrolü. Otomatik Kontrol Ulusal Konferansı (TOK09), İstanbul, s 713-726.
- Ayyıldız, M., Çetinkaya, K. 2014. 4-Serbestlik dereceli gerçek bir seri robotun ters kinematığının qps0 ile çözümü. I. Uluslararası Endüstriyel Tasarım Mühendisliği Sempozyumu (ISIDE14), 8-9 Mayıs 2014, Karabük, Türkiye.
- Bingül, Z., Küçük, S. 2005. Robot Tekniği. Birsen Yayınevi, İstanbul, 343s.
- Boztay, İ. 2007. Gereğinden çok serbestlik dereceli yılanı bir robotun dizaynı ve imalatı. Pamukkale Üniversitesi Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi, 117s.
- Brucoleri, M., D. , Commare, C, U. L. 2007. Off-line Programming and simulation for automatic robot control software generation. 5th IEEE International Conference on Industrial Informatics, 23-27 June, 2: 191-196.
- Corwin, M., Hohn, R., Tarvin, R. 1975. Method and Apparatus for Programming a Computer Operated Robot Arm. United States/ Ohio Patent No. 3920972.
- Craig, J., J. 2005. Introduction to Robotics. Third Edition, United States of America, Pg408. [Erişim Tarihi: 5.11.2020] http://www.mech.sharif.ir/c/document_library/get_file?uuid=5a4bb247-1430-4e46-942c-d692dead831f&groupId=14040
- Demircan, O., Karakuş, R., Abacı, B. 2008. Bir katı modeli boyayabilen bir mekanizma geliştirilmesi. V. Ulusal Hidrolik Pnömatik Kongresi ve Sergisi Makine Mühendisleri Odası (MMO), 23-26 Ekim, İzmir, s. 423-429.
- Demirtaş, A., O. 2020. Tozaltı Ve Gazaltı Kaynak Yöntemlerinin Kullanımı. 15 Nisan 2020 [Erişim Tarihi: 11.06.2019] <https://muhendisgelisim.com/toz-alti-ve-gazalti-kaynak-yontemlerinin-kullanimi/>
- Dereli, S., Köker, R. 2017. 7-DOF Seri Robotun ters kinematik çözümünde eğitime amaçlı kullanılan çok katmanlı yapay sinir ağının tasarlanması ve sonuçların analizi. Gaziosmanpaşa Bilimsel Araştırma Dergisi (GBAD), (6): 60-71.

- Dişlitaş, S. 2015. Endüstriyel robot programlama (1. Baskı) kitabı. Endüstriyel Robot Programlama Eğitimi İle Mesleki Ve Teknik Eğitimin Güçlendirilmesi Projesi Erpe-Meteg. [Erişim Tarihi: 11.06.2019] <https://docplayer.biz.tr/82248668-Endustriyel-robot-programlama.html>
- Doğan, A. 2019. Scara Robot Tasarımı, İmalatı ve Uygulaması. Harran Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Şanlıurfa, 117s.
- Doğan, S. 2006. Manipülatör Tasarımı Projesi. 29.12.2006 Erişim Tarihi: [5.11.2019] <https://silo.tips/download/profdr-brahm-yksel> adresinden erişildi.
- Duran, M. A., Ankaralı, A. 2010. Üç serbestlik dereceli puma tipi bir manipülatörün PID kontrolü. Selçuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu Teknik-Online Dergisi, 9(1) 1: 79.
- Eğilmez, M., M. 2005. Kartezyen koordinatlı kaynak robotu. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 89s.
- Mühürçü, A. Durmuş, G. 2013. 5 Eklemlili bir robot koluna ait ileri kinematik hesaplama yönteminin ysa ile çözümü.
- Özer, T. 2014. Denso Akademik Robot ve Kuka Robot". [Erişim Tarihi: 24 Ekim 2019] [Online] http://sezaitaskin.cbu.edu.tr/wp-content/uploads/2016/03/Robotik_KUKA_DENSO.pdf
- Reed, C.,H. 1989. Development of a passive robot-controlled reation arm". Phd thesis. Pennsylvania university, 209 p. [Erişim Tarihi: 11.06.2019].
- Sönmez, M. , Nil, M., Yüzgeç,U., Kandilli, İ. 2005. Üç Serbest Dereceli Endüstriyel Bir Robotun Yapay Sinir Ağları ile Denetimi". Machinery Makina Teknik Aylık İmalat ve Teknoloji Kültürü Dergisi. [Erişim Tarihi: 11.06.2019].
- Şenel F. A. , Cetişli, B. 2015. Görüntü İşleme ve Beş Eksenli Robot Kol İle Üretim Bandında Nesne Denetimi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 21(5): 158-161.
- Tapıcı, U.Ç. 2006. Robotla Kaynak Uygulaması İçin Fikstür Tasarımı Ve Kaynak Distorsiyonlarının Analizi. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 87s.
- Tonbul, T.S., Saritas, M. 2002. Beş eksenli bir robot kolunda ters kinematik hesaplamalar ve yörünge planlaması". [Erişim Tarihi: 6.10.2020] https://www.emo.org.tr/ekler/edc6ed006e6f49a_ek.pdf
- Türker, K.S. 2015. Endüstride Kaynak Robotları (Proseslerin İncelenmesi ve Geliştirilmesi)", Gedik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 285s.