

V-REP Robotik Simülâtör ile Robot Kol Simülasyonu

Emin AĞRALI^{1*}, Mehmet ÇAVAS²

¹ Elektrik ve Enerji, Teknik Bilimler MYO, Muş Alparslan Üniversitesi, Muş, TÜRKİYE

² Mekatronik Mühendisliği, Teknoloji Fakültesi, Fırat Üniversitesi, Elazığ, TÜRKİYE

*¹ e.agrali@alparslan.edu.tr, ² mcavas@firat.edu.tr

(Geliş/Received: 11/02/2020;

Kabul/Accepted: 23/07/2020)

Öz: Robotik sistemlerde gerçek zamanlı uygulamaların hatasız bir şekilde çalışabilmesi için tasarımı yapılan robot veya robotik sistemler bilgisayar ortamında simüle edilerek test edilmektedir. Bu durum farklı uygulamalar için olası ortaya çıkabilecek aksaklıkların giderilmesi açısından önemlidir. Bu kapsamda yapılan çalışmada, robotik sistemlerde kullanılan MATLAB ve V-REP simülâtörü senkronize edilerek, V-REP simülâtör bünyesinde bulunan ve referans alınan robot kol eklemleri yüklü ve yüksüz durumda PID kontrolör ile kontrol edilmiş, ileri kinematik ve ters kinematik yöntemlerle yol planlama, yörünge takibi ve konum belirlemesi yapılarak simüle edilmiştir. V-REP simülâtör programı özellikle robotik sistemlerin bilgisayar ortamında gerçek zamanlı simüle edilmesini sağlayan önemli bir simülasyon programıdır. Bu program kullanılarak tasarlanan robot veya robotik sistemler simüle edilerek başka sistemler ile konfigüre edilebilmektedir. Ayrıca bu programın MATLAB ile senkronize çalışması da robotik sistemlerin simülasyonunda önemli avantajlar sağlamakta ve istenilen özelliklerde robot veya robotik sistemler tasarlanarak simüle edilmektedir. Bu çalışmada V-REP simülâtör bünyesinde bulunan ABB IRB140 robot kol konfigüre edilerek yüklü ve yüksüz durumda robot kolun izleyeceği yörünge planlanarak konumlar belirlenmiş ve belirlenen konumlara göre yörünge takibi yapılarak MATLAB ile senkronize edilen robot kolun kontrol simülasyonu gerçekleştirilmiştir.

Anahtar kelimeler: Robotik, V-REP, Kontrol, Simülasyon, PID

Robot Arm Simulation with V-REP Robotic Simulator

Abstract: Robotic or robotic systems designed for real-time applications to work flawlessly in robotic systems are simulated in a short time and cost-effectively in the computer environment. This is important for eliminating potential malfunctions for different applications. In this context, MATLAB and V-REP simulator used in robotic systems were synchronized. Robot arm joints were controlled using PID controller. The robot arm was simulated using planning, trajectory tracking with inverse kinematic methods. V-REP simulator program is an important program that enables real-time simulation of robotic systems in a computer environment. Robot or robotic systems designed using this program can be simulated and configured with other systems. In addition, synchronization of this program with MATLAB provides important advantages in the simulation of robotic systems and robotic or robotic systems with desired features can be designed and simulated. In this study, 7 DOF robot arms and ABB IRB140 robot arms are configured via the V-REP simulator. Later, the path to be followed by the robot arm was planned, the locations were determined, the path follow-up was performed according to the specified locations. Finally, the V-REP was synchronized with MATLAB and the simulation of robotic arm was performed.

Key words: Robotics, V-REP, Control, Simulation, PID

1. Giriş

Modern endüstriler başta olmak üzere malzemelerin taşınması, üretimi, montajı, tıbbi tedaviler, uzay araştırmaları vb. birçok alanda farklı işlemlerin yapılmasında farklı tip ve farklı model robot veya robotik sistemler kullanılmaktadır [1]. Teknolojik gelişmelere bağlı olarak endüstriyel otomasyonun gelişimi ile birlikte robotik sistemler hayatımıza girmiş ve üretimin her aşamasında kullanılır hale gelmiştir [2]. Robotik sistemlerin tasarımı, modellenmesi ve üretimi, bilgisayar, elektronik, mekanik, kontrol sistemleri vb. alanları içeren disiplinler arası çalışmaların ürünü olarak görülmektedir. Bu kapsamda farklı alanlarda farklı uygulamalar için geliştirilen robotik sistemlerin laboratuvar ortamında yapılan deneyleri dikkate alındığında bazı problemleri beraberinde getirdiği de görülmektedir. Bu problemlerin aşılabilmesi için geliştirilen robotik simülâtörler robotik sistemlerin tasarımı, modellenmesi ve üretimi için gerekli testlerin sanal ortamda yapılmasını sağlamakla birlikte aynı zamanda robotik eğitimi için de önemli avantajlar sunmaktadır [3]. Geliştirilen bu simülâtörler, robotik sistemlerin araştırılmasında, yeni stratejilerin ve algoritmaların hızlı ve etkin bir şekilde geliştirilmesinde, test edilmesinde vb. birçok süreçte önemli bir rol oynamaktadırlar [4]. Bu simülâtörlerin sahip olduğu çeşitli kinematik, fizik ve grafik kütüphanelerindeki elementler kullanılarak tasarımın yapıp simüle edilmesinin yanı sıra mimarlık ve kontrol

* Sorumlu yazar: e.agrali@alparslan.edu.tr Yazarların ORCID Numarası: ¹ 0000-0002-0513-402X, ² 0000-0002-0130-1644

metodolojisi açısından da bu elementlerin nasıl etkileşime gireceği ve tasarlanan sistemin performansı ile doğruluğunun ne olacağını belirlemede de önem arz etmektedirler [5]. Simülâtörlerin teorik olarak tasarlanmış bir robot veya robotik sistemin sanal bir ortamda görsel olarak gerekli testlerinin yapılması için uygun bir ortam sağlamalarından dolayı günümüzde Gazebo [6], Webots [7], V-REP vb. çok sayıda robot simülâtörü geliştirilmiştir. Bu platformlar içerisinde V-REP (Sanal Robot Simüle Platformu) simülâtörünün daha kısa bir süre içerisinde 3B simülasyonu oluşturması, her sahne nesnesi ve hepsi aynı anda çalışan dişli veya dişli olmayan bir biçimde eklenmiş gömülü bir komut dosyasına sahip olması, sanal bir dünya oluşturmak ve çalışma zamanında etkileşimde bulunmak için çok sayıda örnek robotik model ile beraber sensörler ve aktüatörlere sahip olması [8] vb. özelliklerinden dolayı bu çalışmada tercih edilmiştir. V-REP simülâtör ve MATLAB (matrix laboratory) gerçek zamanlı uygulamalardaki kullanımlara da imkân sunmaktadır. Robotik Sistemlerde simülasyon yapılan sistemin gerçek zamanlı uygulamaları için geliştirilen robotik platformu olan Robot İşletim Sistemi (ROS) kullanılmaktadır. ROS robot yazılımı yazmak için esnek bir çerçevedir. Çok çeşitli robotik platformlarda karmaşık ve sağlam robot davranışı yaratma görevini basitleştirmeyi amaçlayan araçlar, kütüphaneler ve sözleşmeler koleksiyonudur. MATLAB ve V-REP'in, ROS ile bağlantı sağlayan bir alt yapısı bulunmak ile birlikte literatürde birçok çalışmanın yapıldığı da görülmektedir [9-10].

Günümüzde V-REP simülâtör ve MATLAB tabanlı Robotik sistemlerle ilgili birçok çalışmanın yapıldığı, ancak çoğu çalışmaların sistem entegrasyonu, sistem iyileştirme, sistem geliştirme vb. amaçlar kapsamında olduğu görülmektedir. Robot kol tasarımı ve modellenmesi ile ilgili çok fazla çalışmanın yapılmadığı görülmektedir. Bu nedenle bu çalışmada, V-REP ve MATLAB'ın senkronizasyonu ile gerçekleştirilen çok amaçlı robotik manipülâtör kontrolünde sanal bir platform oluşturulmuş, V-REP ve MATLAB ile önerilen planın geçerliliğini ve uygulanabilirliğini göstermek için V-REP'in ana noktaları, manipülâtör eklemlerinin özellikleri, gömülü komut dosyalarının komut dosyası türleri ve V-REP'in uzak API (Application Programming Interface) dahil olmak üzere bütün tanımlar yapılarak referans alınan robot konfigüre edilerek simüle edilmiştir.

2. Materyal ve Yöntem

Bu çalışmada MATLAB ve V-REP robot simülâtörü senkronize edilerek PID kontrolör ile robot kolun eklemleri kontrol edilmiş, ileri ve ters kinematik yöntemler kullanılarak yol planlama, yörünge takibi ve konum belirleme işlemleri yapılarak V-REP ABB IRB140 robot kol simüle edilmiştir. Çalışmada, kendi sanal platformumuzu oluşturmak için kullanılan V-REP'in genel görünüşü Şekil 1'de görülmektedir.



Şekil 1. V-REP simülâtörün genel görünüşü ve farklı tip robot modelleri.

2.1. İleri ve Ters Kinematik

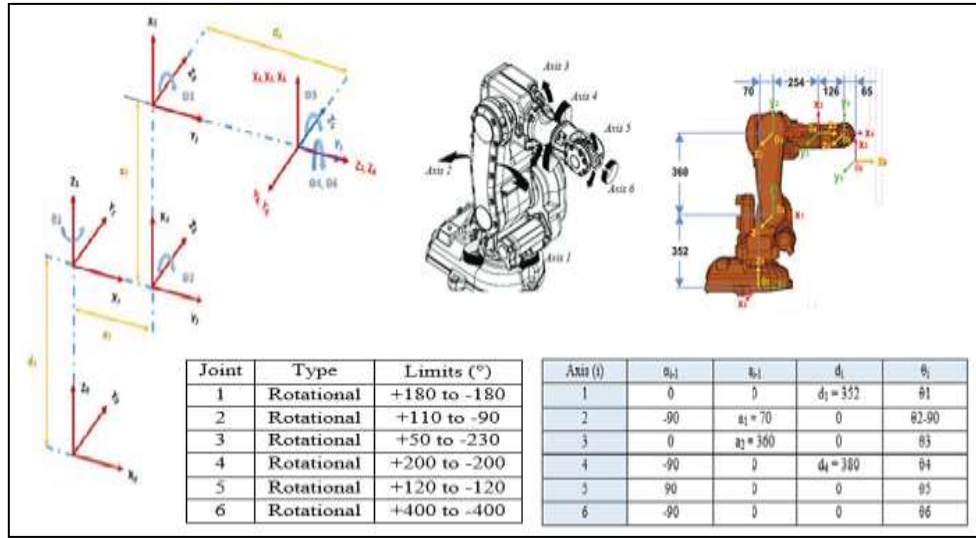
Robot kolun kinematiği, çalışma uzayında matematiksel ifadeler ile konum vektörü ve yönelim matrisinden faydalanılarak nesnelerin konumu ve yönelimi belirlenmektedir. Bunun için üç boyutlu uzayda bir robot kolunun ucuna bağlı bir tutucunun yönelimini ve konumunu tanımlamak için, o robot kolunun merkezine koordinat sistemi yerleştirilmekte ve bu koordinat sistemi sayesinde robot kolu ve bileşenleri arasında bir yönelim ile konum ilişkisi tanımlanarak robot kolun hız, ivme ya da kuvvet analizleri yapılmaktadır [11].

Robot kolun kinematiği merkez nokta ile eklemler ve uç işlevci arasındaki ilişkiyi tanımlarken her bir eklemler kinematik ilişkisi ile kendinden önceki ya da sonraki ekleme göre konumunu ifade etmektedir. Bu durum robot

kolunun konum ve yönelim bilgisi içeren 4x4 dönüşüm matrisleri ile gösterilmektedir. Bu nedenle her bir ekleme bir dönüşüm matrisine sahip olmasından dolayı dönüşüm matrislerinin sayısı robot kolunun serbestlik derecesi ile doğru orantılı olmaktadır. Bu şekilde kullanılan ileri kinematik ile bir robot koluna hedef açı değerleri verilerek robot kol o açı değerlerine ulaştığında aynı zamanda uç noktanın hangi koordinatlarda olacağı bilgisine de ulaşılmış olacaktır ki robot kolun merkezi nokta ile uç nokta arasındaki ilişkiyi elde etmesine ileri kinematik denilmektedir. Ters kinematik ise, robot koluna hedef bir son nokta verildiğinde, robot kolun o noktaya gidebilmesi için o hedef noktanın koordinatlarını kullanarak eklemlerin açılarını hesaplaması olarak ifade edilmektedir [12].

2.2. İleri kinematik ile tork kontrolü

İleri kinematik ile tork kontrolünde D-H (Denavit-Hartenberg) yöntemi kullanılmıştır. D-H parametreleri robotik alanda yaygın olarak kullanılmaktadır. D-H parametreleri kullanılarak uç efektörün dönüşü ve pozisyon vektörleri bulunabilmektedir. Seri kinematik zincirdeki her bir ekleme bir koordinat çerçevesi atanarak D-H değişkenleri bulunmaktadır [13]. Referans alınan ABB IRB140 robotunun kinematik yapısı, çerçeve atamaları, DH tablosu ve eklemlerin hareket açı limitleri Şekil 2’de görülmektedir.



Şekil 2. Referans alınan ABB IRB140 robotunun kinematik yapısı, çerçeve atamaları, DH tablosu ve hareket açı limitleri [13-14].

Oluşturulan D-H parametre tablosundan yararlanılarak Denklem 1 de verilen genel rotasyon matrisi her bir ekleme uygulanmış ve bu denklemler Denklem 2, Denklem 3, Denklem 4 ‘de verilmiştir.

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^0T_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^2T_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_2 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3T_4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

V-REP Robotik Simülâtör ile Robot Kol Simülasyonu

$${}^4_5T = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^5_6T = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Denklem 5’de dönüşüm matrislerinin çarpımı ile işlevcisinin konumu ve yönelimini içeren ve eklem değişkenlerinin birer fonksiyonu olan genel bir dönüşüm matrisi elde edilmiştir.

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Kinematik hesaplamalar MATLAB programında yapıldıktan sonra V-REP programıyla da simülasyon yapılmaktadır. Burada V-REP simülâtör programı simüle edilen sistemin harici bir uygulamadan kontrol edilmesini sağlayan uzak bir API sunmaktadır. V-REP uzak API’si MATLAB programından çağrılabilir ve yaklaşık yüz fonksiyondan oluşmaktadır [15]. Bu şekilde MATLAB ve V-REP simülâtörü senkronize edilerek referans alınan birçok model üzerinden çeşitli simülasyonlar da yapılabilmektedir [16]. Bu çalışmada ilkönce V-REP simülâtörde referans model manipülâtör alınarak gerekli konfigürasyonlar yapılmış ve daha sonra MATLAB ile V-REP simülâtörün iletişimini sağlayan ve manipülâtörün referans konumlarını belirleyen program oluşturularak MATLAB üzerinden V-REP simülâtörü ile ABB IRB140 manipülâtörün ileri kinematik yönteminde belirlenen referans açıları hedef konumlara ulaşması sağlanmıştır. Burada eklemlerin ileri kinematik yöntemi kullanılarak tork kontrolünün gerçekleştirilmesi için tork modunda ayarlanması gerekmektedir. Daha sonra robot kolun her bir linki için dinamik modun pasif edilmesi gerekmektedir. Yukarıda belirtilen işlemler tamamlandıktan sonra MATLAB ile V-REP programı arasındaki iletişimi sağlayacak bir çalışma klasörü oluşturularak bu klasöre MATLAB ile V-REP bağlantısını sağlayan V-REP uzak API dosyaları kopyalanarak iletişim sağlanmıştır. Bu işlemler tamamlandıktan sonra robot kolun ileri kinematik yöntem ile tork kontrolünü sağlayacak program kodları MATLAB’ta yazılmış olup bu kodlar Şekil 3’de görülmektedir.

```
clear all
close all
clc
a1=70;a2=360;d1=352;d4=380;
vrep=remApi('remoteApi'); %prototip dosyasını kullanma (remoteApiProto.m)
vrep.simxFinish(-1); % Her ihtimale karşı, tüm açılmış bağlantıları kapatmak
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);

if (clientID>-1)
    disp('UZAK API sunucusuna bağlandı');
    % eklem açılarını oluşturma
    teta1=input('Birinci eklem açısını giriniz= ');
    teta2=input('İkinci eklem açısını giriniz= ');
    teta3=input('Üçüncü eklem açısını giriniz= ');
    teta4=input('Dördüncü eklem açısını giriniz= ');
    teta5=input('Beşinci eklem açısını giriniz= ');
    teta6=input('Altıncı eklem açısını giriniz= ');
| %MATLAB İTLERİ KİNEMATİK HESAPLAMA %
T01=[cosd(teta1) -sind(teta1) 0 0;...
    sind(teta1) cosd(teta1) 0 0;...
    0 0 1 d1;...
    0 0 0 1];
T12=[cosd(teta2-90) -sind(teta2-90) 0 a1;...
    0 0 1 0;...
    -sind(teta2-90) -cosd(teta2-90) 0 0;...
    0 0 0 1];
T23=[cosd(teta3) -sind(teta3) 0 a2;...
    sind(teta3) cosd(teta3) 0 0;...
    0 0 1 0;...
    0 0 0 1];
T34=[cosd(teta4) -sind(teta4) 0 0;...
    0 0 1 d4;...
    -sind(teta4) -cosd(teta4) 0 0;...
    0 0 0 1];
T45=[cosd(teta5) -sind(teta5) 0 0;...
    0 0 -1 0;...
    sind(teta5) cosd(teta5) 0 0;...
    0 0 0 1];
T56=[cosd(teta6) -sind(teta6) 0 0;...
    0 0 1 0;...
    -sind(teta6) -cosd(teta6) 1 0;...
    0 0 0 1];
T06=T01*T12*T23*T34*T45*T56;
Px=T06(1,4);
Py=T06(2,4);
Pz=T06(3,4);
```

```

%% %% %% %% V-REP İLERİ KİNEMATİK HESAPLAMA %% %% %% %% %% %% %% %% %%
teta1=teta1*(pi / 180);teta2=teta2*(pi / 180);teta3=teta3*(pi / 180);
teta4=teta4*(pi / 180);teta5=teta5*(pi / 180);teta6=teta6*(pi / 180);
eklem_acilar1=[teta1 teta2 teta3 teta4 teta5 teta6];
% eklemier
h=[0 0 0 0 0 0];
[r,h(1)]=vrep.simxGetObjectHandle( clientID,'base_joint1',vrep.simx_opmode_blocking);
[r,h(2)]=vrep.simxGetObjectHandle( clientID,'joint2',vrep.simx_opmode_blocking);
[r,h(3)]=vrep.simxGetObjectHandle( clientID,'joint3',vrep.simx_opmode_blocking);
[r,h(4)]=vrep.simxGetObjectHandle( clientID,'joint4',vrep.simx_opmode_blocking);
[r,h(5)]=vrep.simxGetObjectHandle( clientID,'joint5',vrep.simx_opmode_blocking);
[r,h(6)]=vrep.simxGetObjectHandle( clientID,'joint6',vrep.simx_opmode_blocking);
while true
for i=1:1:6
vrep.simxSetJointTargetPosition( clientID,h(i), eklem_acilar1(i),vrep.simx_opmode_streaming)
end
pause(2);
end
vrep.simxSetJointTargetPosition( clientID,h(1), eklem_acilar1(1),vrep.simx_opmode_streaming)
vrep.simxSetJointTargetPosition( clientID,h(2), eklem_acilar1(2),vrep.simx_opmode_streaming)
vrep.simxSetJointTargetPosition( clientID,h(3), eklem_acilar1(3),vrep.simx_opmode_streaming)
vrep.simxSetJointTargetPosition( clientID,h(4), eklem_acilar1(4),vrep.simx_opmode_streaming)
vrep.simxSetJointTargetPosition( clientID,h(5), eklem_acilar1(5),vrep.simx_opmode_streaming)
vrep.simxSetJointTargetPosition( clientID,h(6), eklem_acilar1(6),vrep.simx_opmode_streaming)
else
disp('uzak API sunucusuna bağlanılamadı');
end
vrep.delete();
disp('Program sona erdi');

```

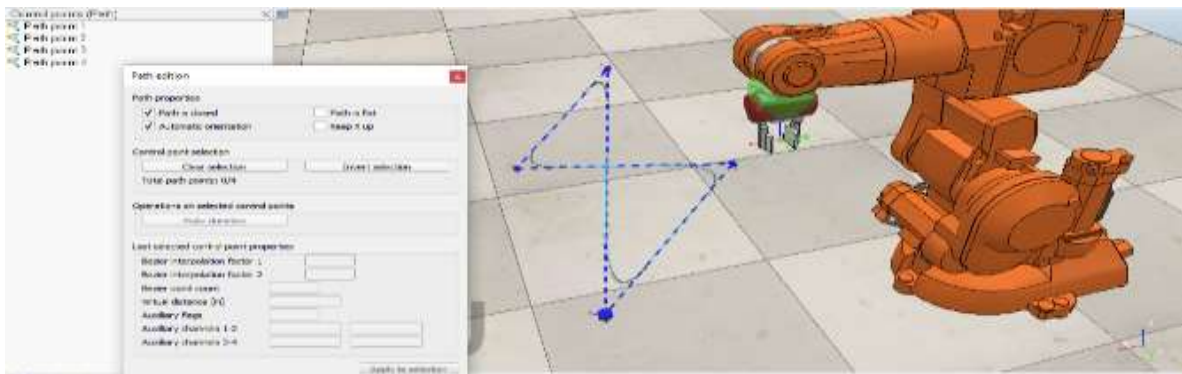
Şekil 3. MATLAB ileri kinematik hesaplama program kodları.

Bu işlemler tamamlandıktan sonra ileri kinematik ile yapılan tork kontrolünde PID kontrolör kullanılarak istenilen referans konumlara daha yakın bir değere ulaşmasını sağlamak için robot eklemlerinin her birinin PID kontrolörü aktif hale getirilmiştir. Burada robot kolun eklemlerinin hareketini sağlamak için motor modu aktif edilmiş ve eklemin maksimum torku ile PID kontrolörün parametreleri birlikte belirlenmiştir.

2.3. Ters kinematik ile yörünge kontrolü

V-REP simülâtörün ters kinematik (IK) hesaplama modülü çok güçlü ve esnek bir yapıya sahip olup hemen hemen her türlü mekanizmanın ters kinematik modunda (IK modu) veya ileri kinematik modunda (FK modu) kullanılmasına imkan sunmaktadır [17]. Bu modül sayesinde bir elemanın belirli bir pozisyona karşılık gelen eklem değerleri bulunmakta ve aynı zamanda görev alanı koordinatlarından ortak uzay koordinatlarına dönüşüm sağlanmaktadır [18]. Bu çalışmada V-REP simülâtör bünyesinde bulunan ABB IRB140 robot kolun ters kinematik yöntem ile belirlenen referans notalarından oluşturulan yörüngeyi takip etmesi için V-REP'in içerisindeki ters kinematik mod aktif edilerek hareketi sağlanmıştır.

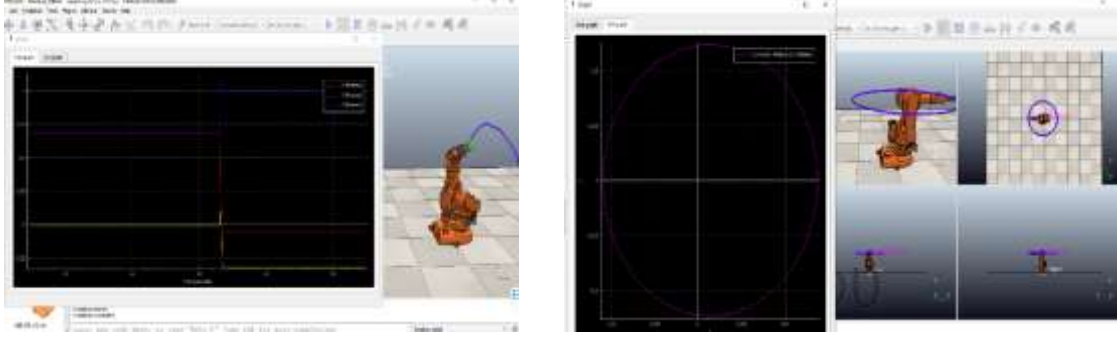
Robot kolun ters kinematik modu aktif edildikten sonra robot kolun uzuvlarına etki eden iç ve dış kuvvetlerin etkisini ortadan kaldırmak için robotu oluşturan her bir uzvun dinamik modu pasif edilmelidir. Robotun dinamik modu pasif, ters kinematik modu ise aktif edildikten sonra robot kolun referans yörünge modeli yörünge noktalarının konumları birbiriyle ilişkilendirilerek oluşturulmaktadır. Yörüngeyi oluşturan noktalar ardışıl bir bağlantı ile açık veya kapalı bir yol şeklinde tamamlanmaktadır. Bu durum Şekil 4'de gösterilmiştir. Bu işlemlerden sonra robot kolun belirlenen referans yörüngeyi ters kinematik yöntem ile takip edebilmesi için uç manipülâtörün merkezine referans hareket noktası ile hedef nokta yerleştirilerek yazılan kod ile birlikte robot kolun hareketi sağlanmaktadır.



Şekil 4. V-REP simülâtörde ters kinematik modun aktif edilmesi ve referans yörünge modeli.

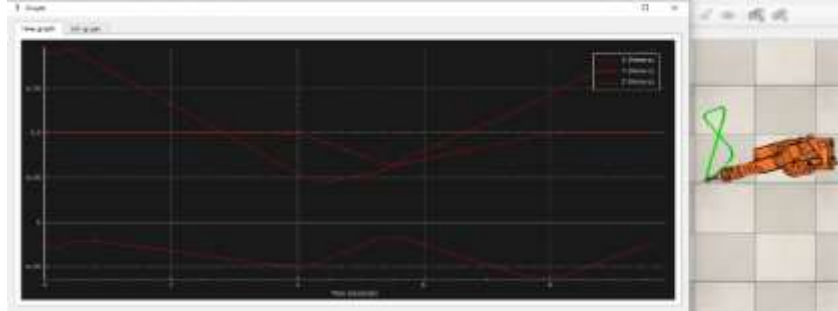
3. Bulgular

Bu çalışmada MATLAB ve V-REP simülâtör arasında iletişim sağlanarak robot kol simülasyonu gerçekleştirilmiş ve MATLAB ortamından girilen açı değerlerine göre robot kolun uç işlevcisinin hedef açı değerlerine göre hangi koordinatlarda olduğu bilgisine ulaşılmıştır. MATLAB programında girilen farklı açı değerlerine göre robotun uç efektörünün farklı konum ve açı değerlerine göre hareket grafikleri Şekil 5'te görülmektedir.



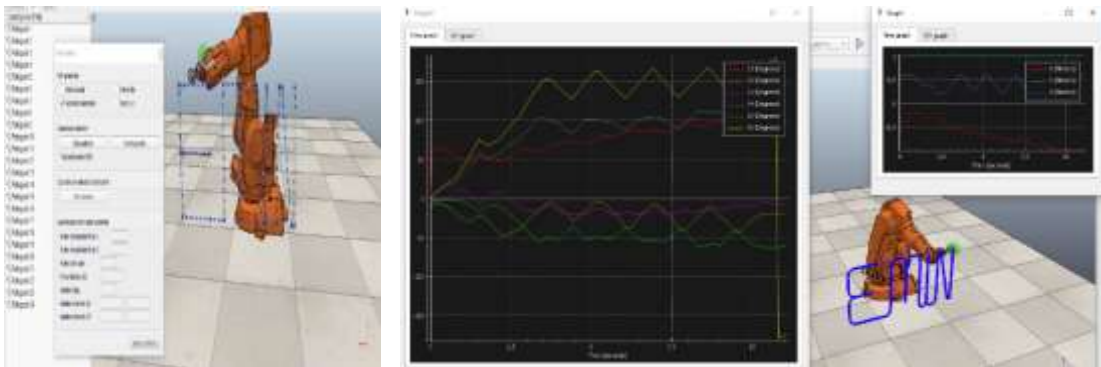
Şekil 5. MATLAB ortamında girilen farklı açı değerlerine göre V-REP simülâtörde robotun uç efektörünün konumu ve eklemlerin açı grafikleri

Ayrıca ABB IRB140 robot kolun ters kinematik yöntemle belirlenen referans yörüngeyi takip ederek hareket etmesi sağlanmış ve bu durum Şekil 6'da gösterilmiştir.



Şekil 6. Robot kolun ters kinematik yöntemle belirlenen referans yörüngeyi takibi.

Öte yandan ters kinematik yöntem kullanılarak alfabetik karakterler ile bir yörünge oluşturulmuş ve robot kolun bu yörüngeyi takip etmesi sağlanarak elde edilen sonuçlar Şekil 7'de gösterilmiştir.



Şekil 7. Robot kolun alfabetik yörüngeyi takibi sonucu eklemlerin hareket açı değerlerinin grafikleri.

ABB IRB140 manipulatörün ileri kinematik yöntemle, yüklü durumdaki tork kontrolü simülasyonu için manipulatöre 2 kg ağırlığındaki dikdörtgen biçiminde bir yük bağlanmış ve parametre düzenlemesi yapılmıştır. Bu durum Şekil 8’de görülmektedir.



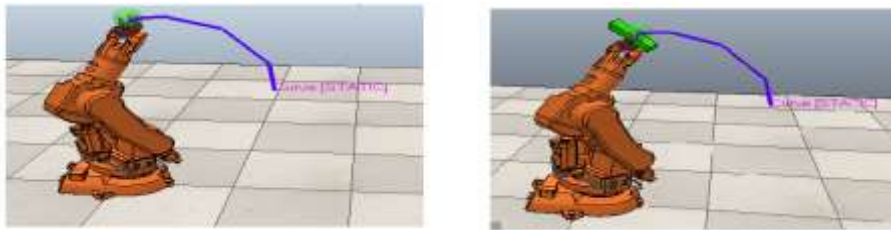
Şekil 8. Manipulatöre bağlanan 2 kg yük ve parametre ayarlaması.

ABB IRB140 manipulatörün ileri kinematik yöntem ile yüklü durumdaki tork kontrolü ve yüksüz durumdaki tork kontrolünde meydana gelen değişimleri tespit etmek için V-REP simülatör ve MATLAB ile yapılan simülasyon sonucu elde edilen koordinatlar Tablo 1’de görülmektedir.

Tablo 1. ABB IRB140 manipulatörün yüklü ve yüksüz durumunda elde edilen koordinatlar.

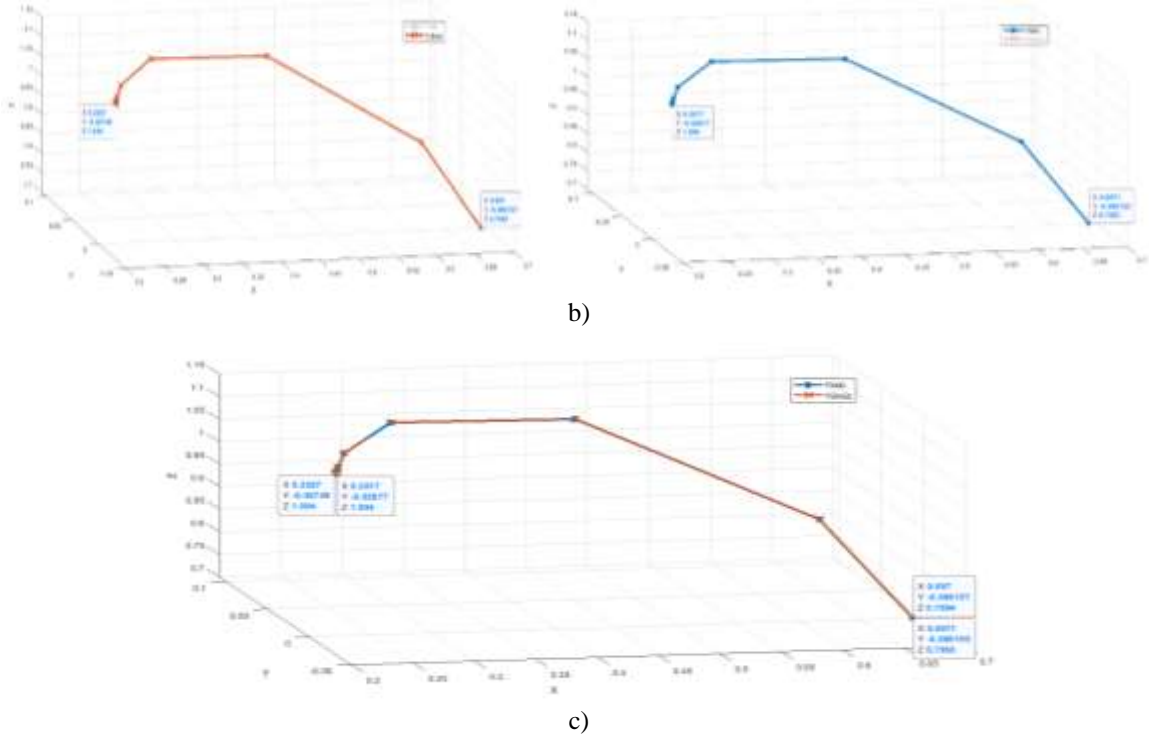
Yüklü			Yüksüz			Fark		
X (Meters)	Y (Meters)	Z (Meters)	X (Meters)	Y (Meters)	Z (Meters)	X (Meters)	Y (Meters)	Z (Meters)
0,686980	-0,006158	0,7094	0,687030	-0,006157	0,709339	0,000050	0,000001	-0,000061
0,686987	-0,006158	0,709386	0,687041	-0,006157	0,709327	0,000054	0,000001	-0,000059
0,687011	-0,006158	0,709373	0,687067	-0,006156	0,709313	0,000056	0,000002	-0,000060
0,687015	-0,006157	0,709359	0,687073	-0,006155	0,7093	0,000058	0,000002	-0,000059
0,687026	-0,006156	0,709345	0,687086	-0,006155	0,709286	0,000060	0,000001	-0,000059
0,687030	-0,006156	0,709332	0,687092	-0,006154	0,709272	0,000062	0,000002	-0,000060
0,687048	-0,006155	0,709322	0,687112	-0,006153	0,709259	0,000064	0,000002	-0,000063
0,687058	-0,006154	0,709312	0,687124	-0,006152	0,709245	0,000066	0,000002	-0,000067
0,687067	-0,006153	0,709302	0,687134	-0,006152	0,709231	0,000067	0,000001	-0,000071
0,687077	-0,006153	0,709292	0,687147	-0,006151	0,709217	0,000070	0,000002	-0,000075
0,201583	-0,027486	1,093983	0,201684	-0,027269	1,093548	-0,000101	-0,002452	-0,000435
0,201604	-0,028489	1,093562	0,201684	-0,027345	1,093548	0,000080	0,001144	-0,000014
0,201701	-0,028771	1,094862	0,202703	-0,027493	1,094788	0,000102	0,001278	-0,000074

Ayrıca V-REP simülatör ile yapılan simülasyon görüntüleri ve MATLAB grafikleri Şekil 9’da görülmektedir.



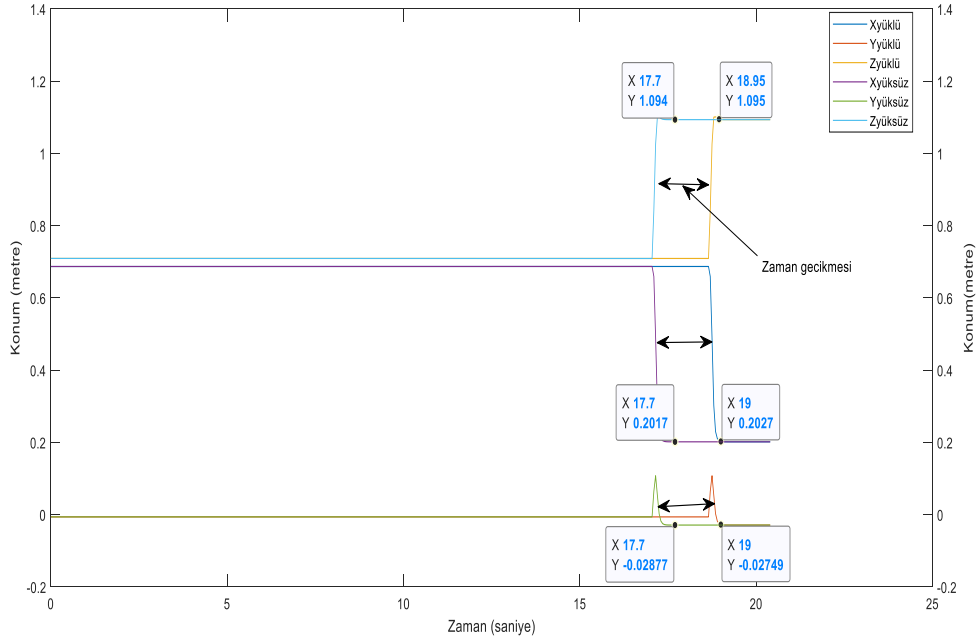
a)

V-REP Robotik Simülör ile Robot Kol Simülasyonu



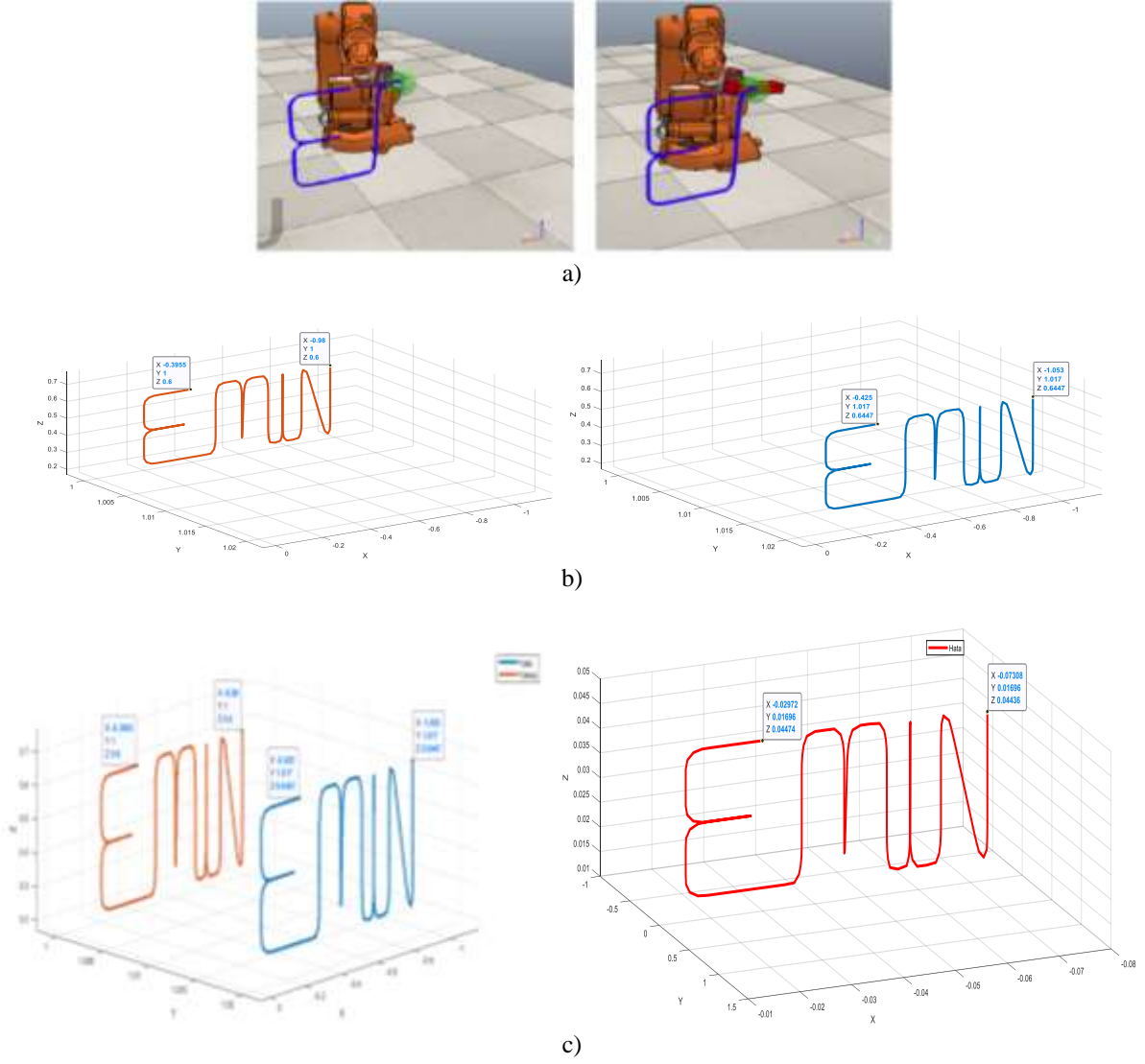
Şekil 9. MATLAB ortamında girilen farklı açı değerlerine göre; a) V-REP simülörde robot kolun yüklü ve yüksüz durumdaki konumu, b-c) MATLAB programında elde edilen konum grafikleri.

Uç manipülörün yüklü durumunun yüksüz duruma göre konumunun koordinatlarını oluşturan MATLAB grafiği Şekil 10'de görülmektedir.



Şekil 10. Manipülörün yüklü durumunun yüksüz duruma göre konumunun MATLAB grafiği.

ABB IRB140 manipülörün ters kinematik yöntemle yörünge kontrolünün yüklü ve yüksüz durumlarda analizini yapmak için V-REP manipülör ile simülasyon yapılmış ve MATLAB ortamında grafikler elde edilmiştir. V-REP simülasyon görüntüleri ve MATLAB grafikleri Şekil 11'de görülmektedir.



Şekil 11. Robot kolun, a) alfabetik yörüngeyi yüklü ve yüksüz durumlardaki takibi simülasyonu b) alfabetik yörünge grafikleri c) yörünge hatası

4. Sonuç

Bu çalışmada MATLAB ve V-REP simülator arasındaki iletişim sağlanarak ABB IRB140 robot kol ileri ve ters kinematik yöntemler kullanılarak PID kontrolör ile tork ve yörünge kontrolü simüle edilmiştir. İleri kinematik yöntemle tork kontrolünde yüksüz durumda MATLAB programından girilen açılış değerlerine göre başarılı bir kontrol sağlandığı Şekil 5’de görülmektedir. Ayrıca manipülatör, şekil 9-a-b ve c’de görüldüğü gibi yüklü ve yüksüz durumları simüle edilmiştir. Bu simülasyon sonuçlarında Tablo 1’de, Şekil 9-c’de ve Şekil 10’da görüldüğü gibi hata oranının çok düşük olduğu görülmektedir. Şekil 10’da yüklü durumun yüksüz duruma göre 1.3 saniyelik bir zaman gecikmesinin ve uç efektör konumunun X-Y-Z eksenlerinde yaklaşık 10 mm’lik bir sapma gerçekleştiği görülmektedir. Çalışmanın diğer bir amacı olan, ABB IRB140 manipülatörün ters kinematik yöntemle yörünge takibi başarılı bir şekilde yapıldığı Şekil 7’de görülmektedir. Ters kinematik yöntemle yörünge kontrolünde yüklü ve yüksüz durumlar için simülasyon başarılı bir şekilde yapıldığı Şekil 11-a’da görülmektedir. Burada yine yüklü durumun yüksüz duruma göre X eksenindeki konumunda ortalama 32 mm, Y eksenindeki konumunda 17 mm, Z eksenindeki konumunda ortalama 44 mm sapma gerçekleştiği şekil 11-b-c’de görülmektedir.

V-REP Robotik Simülâtör ile Robot Kol Simülasyonu

Simülasyon sonucunda elde edilen veriler değerlendirildiğinde MATLAB ve V-REP simülâtör arasındaki iletişimin başarılı bir şekilde yapıldığı, yüklü ve yüksüz durumlara göre robot kolun simülasyonu yapılmış ve robot kolun ileri ve ters kinematik yöntemlerle tork ve yörünge kontrolü PID kontrolör ile düşük bir sapmayla başarılı bir şekilde gerçekleştirilmiştir. Elde edilen veriler ve alınan sonuçlar değerlendirildiğinde V-REP simülâtör ve MATLAB programı kullanılarak farklı robot veya robotik sistemlerin tasarlanabileceği gibi mevcut robotik sistemlerin yeni sistemler için geliştirilebileceğini de göstermektedir.

Kaynaklar

- [1] He XJ, Chen YH. Haptic-aided robot path planning based on virtual tele-operation. *J Robot and Comput Integr Manuf.* 2009; 25, 792–803
- [2] Huang S, Wang M ve Zou Y. V-REP-based virtual platform on multi-pattern neural control of robotic manipulator. *Chinese Control and Decision Conference.* 2019; 1570-1575
- [3] Fabregas E, Farias G, Peralta E, Vargas H ve Dormido S. Teaching control in mobile robotics with V-REP and a Khepera IV library. In *2016 IEEE Conference on Control Applications.* 2016; (pp. 821-826).
- [4] Koenig N & Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. *International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566). 2004;(Vol. 3, pp. 2149-2154).
- [5] Peralta E, Fabregas E, Farias G, Vargas H ve Dormido S. Development of a Khepera IV Library for the V-REP Simulator. *IFAC-PapersOnLine.* 2016;49(6), 81-86.
- [6] Koenig N & Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE Cat. No. 04CH37566). IEEE. 2004; (Vol. 3, pp. 2149-2154).
- [7] Michel O. Cyberbotics Ltd. Webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems.*2004; 1(1), 5.
- [8] Rohmer E, Singh SP & Freese M. V-REP: A versatile and scalable robot simulation framework. *International Conference on Intelligent Robots and Systems.* IEEE. 2013; (pp. 1321-1326).
- [9] Olivares-Mendez MA, Kannan S & Voos H. Setting up a testbed for UAV vision based control using V-REP & ROS: a case study on aerial visual inspection. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS).* IEEE.2014; (pp. 447-458).
- [10] Corke P. Integrating ros and matlab [ros topics]. *IEEE Robotics & Automation Magazine,* 2015;22(2), 18-20.
- [11] Kucuk S, Bingul Z. The inverse kinematics solutions of industrial robot manipulators. In *Proceedings of the IEEE International Conference on Mechatronics, ICM'04.* IEEE. 2004;(pp. 274-279).
- [12] Zhao J, Yang F, Liu W, Liu F, Li F, Wang H & Zhang H. An Approximation Model Based on Kernel Ridge Regression for Robot Kinematics Simulation. *23rd International Conference on Computer Supported Cooperative Work in Design.* IEEE. 2019; (pp. 313-318).
- [13] Arfaq M, Dewanto RS, & Pramadihanto D. Fall Detection in T-FLoW Humanoid Robot: V-REP Simulation. *International Electronics Symposium on Engineering Technology and Applications (IES-ETA).* IEEE. 2018 ;(pp. 224-228).
- [14] Hidayatullah JF, Pramadihanto D, Dewanto RS & Khalilullah AS. Kinematic analysis of 7 DoF head humanoid FLoW: V-REP simulation. *International Electronics Symposium (IES).* IEEE, 2016 (pp. 13-18).
- [15] Azak S, Erdogan E. Performance evaluation of the grid-based FastSLAM in V-REP using MATLAB. *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering.* IEEE. 2018; (pp. 276-281).
- [16] Tongtong L, Tao Y, Zelin Y, Shuxuan L ve Jianming L. Development of Hardware-in-Loop Simulation Platform for Collaborative Robots Based on LinuxCNC and V-rep. *International Conference on Mechatronics and Automation (ICMA)* . IEEE. 2018; (pp. 1323-1328).
- [17] Gudla AG. A methodology to determine the functional workspace of a 6R robot using forward kinematics and geometrical methods.2012
- [18] Almagd M. Forward and inverse kinematic analysis and validation of the ABB IRB 140 industrial robot. *Journal of Mechanical Engineering and Technology (JMET).* 2017; 9(2), 1-20.