

İstanbul Beykent Üniversitesi
Yazılım Mühendisliği Bölümü
Yazılım Mühendisliği Tasarım Projesi

Rapor 2
2024 – GÜZ

**6 Serbest Eksenli Robot Kol Projesi
(6RK)**

Projeyi Hazırlayan

2103013265 – Hakan LOYAN

Danışman

Kemal Gökhan NALBANT

Türü ve Tarihi : Final Raporu – 23 Ocak 2025

ÖZ

Gelişen teknoloji ve insan gücünde israfın önüne geçilmesi adına endüstri devrimlerinin de gerekliliğiyle otonom sistemler ve robotlara olan ihtiyaç kaçınılmaz bir etken.

Son yıllarda yapılan teknoloji festivalleri incelendiğinde henüz lisans seviyesinde olmayan öğrencilerle de gerçekleştirilen projelerin kapsama alanı azımsanmayacak derecede ilerlemiş durumda. Profesyonelleşme ve kapsam genişletme doğrultusunda “know-how” elde etmenin gerektirdiği sürecin karşısında açık kaynak olarak yayımlanan projelerin yeri dikkate değer. Gerek global gerek yerel projelere yöneldiğimizde “Robot Kol” gibi detaylı ve büyük çaplı projelerde kaynak bulmanın zorluğu, herkes tarafından geliştirilebilir projelerin var olmayışı ve endüstride bu robotların ihtiyaç duyduğu yüksek maliyetler kabul ettiğimiz bir gerçek.

Bu çalışmada düşük maliyetle, ulaşılabilir üretim gücü ile 6 serbest eksenli bir robot kolun mekanik, elektronik ve yazılımsal süreçlerinin aslında ulaşılabilir olduğu düşük maliyet ile de istenilen hassasiyetler doğrultusunda bir robot geliştirilebileceği tasarlanmıştır. Proje sonunda tüm kaynaklar, tasarımlar ve proje geliştirme süreçleri açık kaynak olarak paylaşılıp projenin sürekliliği ve robot kol üretiminin kolaylığı hedeflenmektedir.

ABSTRACT

The necessity for autonomous systems and robots has become an inevitable factor, driven by advancements in technology and the need to eliminate inefficiencies in human labor, in line with the requirements of industrial revolutions.

An analysis of recent technology festivals reveals that the scope of projects undertaken, even by pre-undergraduate students, has advanced to a noteworthy degree. In the pursuit of professionalization and broader project scopes, open-source initiatives hold significant value in addressing the lengthy processes required for acquiring “know-how.” When examining both global and local projects, challenges such as the difficulty of sourcing resources for complex and large-scale endeavors like “robotic arms,” the absence of universally accessible development platforms, and the high costs associated with industrial robotics are undeniable realities.

This study proposes the design and development of a six-degree-of-freedom robotic arm, demonstrating that mechanical, electronic, and software processes can indeed be achieved at a low cost with accessible production capabilities. The project aims to show that it is possible to develop a robot meeting the required precision standards at a significantly reduced cost. Upon completion, all resources, designs, and project development processes will be shared as open source, with the goal of ensuring the project’s continuity and facilitating the production of robotic arms.

İçindekiler

III Robotik.....	7
8 Robotiğe Giriş	7
9 Manipülatör Tasarımı	7
9.1 Katı cisimlerin yapılandırılması	8
9.2 Dönüşüm	10
9.3 DH Parametreleri.....	11
9.4 İleri Kinematik Hesaplamaları	12
9.5 Ters Kinematik Hesaplamaları	16
10 “6RK” Robot Kolun Tasarımı ve Üretimi.....	24
11 Elektronik Sistem Tasarımı	26
IV Tasarım Dokümantasyonu	28
12 Tasarım Hedeflerinin Tanımlanması.....	28
13 Önerilen Yazılım Mimarisi	30
13.1 Diyagram Açıklamaları	32
14 Sınıf Diyagramları	32
15 Dinamik Model.....	38
16 Kullanıcı Arayüzü	43
V Test Planları	48
17 Test Edilebilecek ve Edilemeyecek Özellikler	48
14 Test Cases.....	51

Şekiller Listesi

Şekil 7: 8

Şekil 8: 9

Şekil 9: 9

Şekil 10 : 18

Şekil 11: 18

Şekil 12: 19

Şekil 13: 19

Şekil 14: 20

Şekil 15: 22

Şekil 16: 24

Şekil 17: 25

Şekil 18: 26

Şekil 19: 27

Şekil 20: 31

Şekil 21: 33

Şekil 22: 34

Şekil 22: 35

Şekil 23: 36

Şekil 24: 37

Şekil 25: 38

Şekil 26: 39

Şekil 27: 40

Şekil 28: 41

Şekil 29: 42

Şekil 30: 43

Şekil 31: 44

Şekil 32: 45

Şekil 33: 46

Şekil 34: 47

Tablolar Listesi

Tablo 7: 48

Tablo 8: 49

Tablo 9: 50

III Robotik

8 Robotiğe Giriş

Robotik; robotların tasarlanması, inşa edilmesi ve hareket ettirilmesini kapsayan disiplinler arası mühendislik dallarından biridir.

“Bir robot, çeşitli görevlerin gerçekleştirilmesi için değişken programlanmış hareketler yoluyla malzemeleri, parçaları, araçları veya özel cihazları hareket ettirmek üzere tasarlanmış yeniden programlanabilir çok işlevli bir manipülatördür” [Robot Institute of America, RIA].

Robotlar, insanların yerine işleri tamamlayacak makineler olarak farklı mühendislik disiplinlerinin bilgisiyle tasarlanır. Farklı alanlarda ve çeşitli amaçlarla kullanılabilen robotlar, bazı durumlarda insanları taklit edecek şekilde tasarlanır. Bu tür robotlar, yürümek, kaldırmak, konuşmak, bilişsel işlevleri yerine getirmek ve bir insanın yapabileceği diğer eylemleri simüle etmeyi hedefler.

Robot sistemleri, geri besleme sensörleri, kontrol sistemleri, manipülatörler, güç kaynakları ve yazılımların birlikte çalışarak belirli bir görevi yerine getirdiği kompleks yapılardır. Robotların tasarımı, inşası, programlanması ve test edilmesi; makine mühendisliği, mekatronik, elektrik mühendisliği, yazılım mühendisliği, elektronik, yapay zeka, nanoteknoloji ve biyomühendislik gibi birçok disiplinin birleşimini gerektirir. Ayrıca, bazı durumlarda biyoloji, tıp ve kimya gibi alanlar da sürece dahil olabilir.

9 Manipülatör Tasarımı

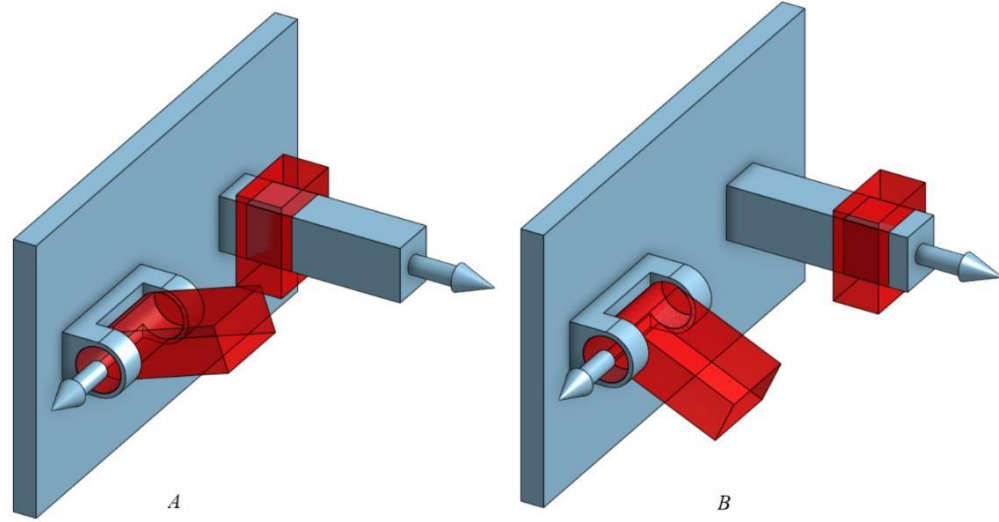
Kontrol algoritmasını doğrudan etkileyen ve robotun iskeleti diyerek nitelendirebileceğimiz robot manipülatörlerin tasarımı ve modellemesi için gerekli olan:

- Katı cisimlerin yapılandırılması,
- Bir noktanın uzay boşluğundaki pozisyonunun ifade edilmesi,
- Hareket kabiliyeti için kinematik denklemler,

maddelerinin tasarım ve modelleme aşamasından önce net bir şekilde tanınması gerekir.

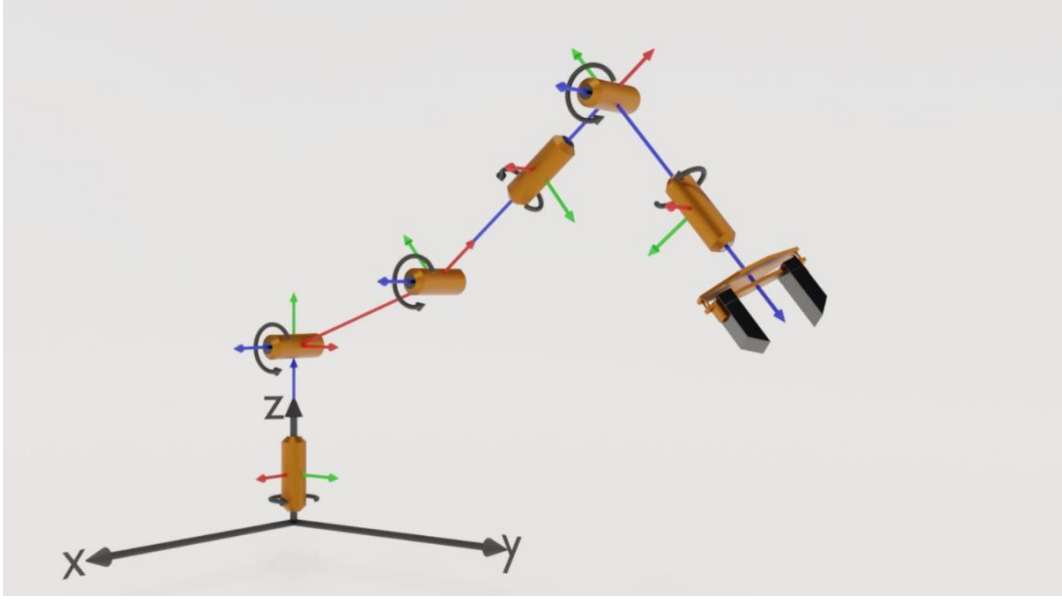
9.1 Katı cisimlerin yapılandırılması

Bir robotu 3 boyutlu olarak tanımlamak, söz konusu robot için yapılandırmayı gerektirecektir. Yapılandırma, tüm konum noktalarının özelliklerini içerir. Robotikte iki tür gövde vardır, deforme edilebilen esnek bir gövde veya robotun tamamen katı olarak algılanabileceği bu projenin odak noktasındaki katı olacak bir gövde. Bağlantılar olarak da adlandırılan katı gövdeler, Şekil 7'de görüldüğü gibi, bir döner veya doğrusal eklemlerle birbirine bağlanır.



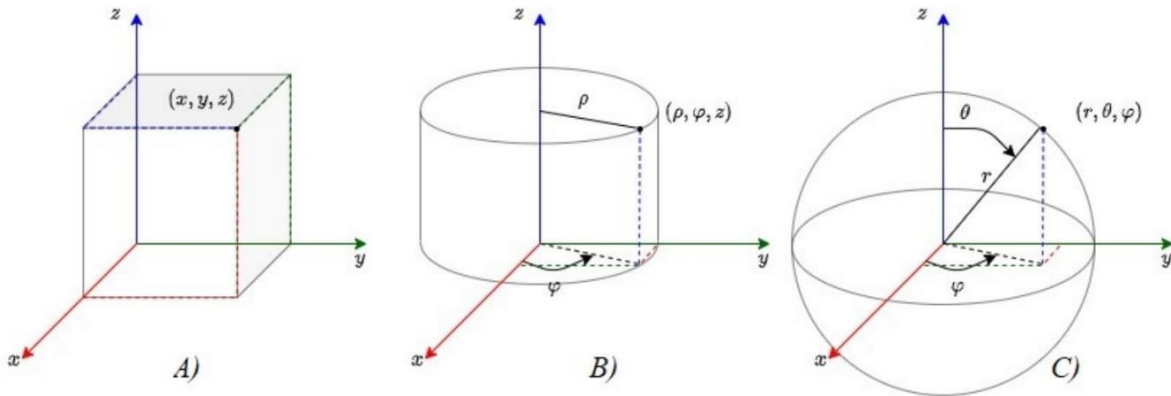
Şekil 7: Kırmızı parçalar döner ve doğrusal eksenleri iki farklı pozisyonda temsil eder. Oklar, dönüşel ve doğrusal eksenleri gösterir.

Robot manipülatörler, genel olarak, iki veya daha fazla bağlantıdan oluşur. Her bağlantının kendi kütle merkezi, eylemsizlik momenti vb. değerleri vardır. Her bağlantı parçası için sabit bir koordinat sisteminde bir konum tanımlamak daha kolaydır, Şekil 8'de siyah renkteki eksenle gösterilen sabit bir koordinat sistemidir. Bu koordinat sistemi, robot hareketleri sonrasında konumunu veya yönelimini değiştirmez. Ancak, (kırmızı, yeşil ve mavi) ile gösterilen ve (x, y ve z) eksenini temsil eden bağlantıların sabit koordinat sistemi (referans) konumu ve yönelimi değiştirebilir. Eklemlerinde yaşanan değişiklikler buna sebep olur. Bundan sonra, herhangi bir bağlantı parçasının kütle merkezinin konumunu tanımlamak için çerçeveler arasında bir dönüşüm kullanılabilir.



Şekil 8: Robot kol, eklemlerin zinciri, eklemler referans çerçeveleriyle birlikte.

Bir noktanın pozisyonunu 3 boyutlu uzayda tanımlamak için birçok yol vardır; kartezyen, silindirik veya küresel koordinat sistemleri Şekil 9’da gösterilmiştir. Bu projede kartezyen koordinat sistemi kullanılacaktır.



Şekil 9: A) Kartezyen, B) Silindirik ve C) Küresel koordinat sistemleri.

9.2 Dönüşüm

Bir noktaya çevirme veya döndürme uygulamak için matrislerin kullanılması, matris çarpımının birleşme özelliği sebebiyle avantajlıdır. Homojen dönüşüm matrisi, (4x4) boyutunda bir matristir ve bir noktayı bir çerçeveden diğerine dönüştüren yani p^l vektörünün p^w vektörüne dönüştüren bilgiyi sağlar. Klasik bir homojen dönüşüm matrisi aşağıdaki gibi görünür:

$$H_l^w = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$p^w = H_l^w p^l$$

İlk (3x3 matrislerden) elemanlar döndürme matrisi R'yi, son sütundaki ilk 3 eleman ise çevirme T'yi içerir. (x, y, z) eksenleri etrafındaki döndürmeler, (ϕ , γ , θ) açılarıyla birlikte aşağıdaki homojen dönüşümlerle verilmiştir:

$$H_{R_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$H_{R_y}(\gamma) = \begin{bmatrix} \cos \gamma & 0 & \sin \gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \gamma & 0 & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$H_{R_z}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bu, eklemlerin çerçevelerinin dönüşüm matrislerini birbiriyle çarparak end-effector'ün pozisyonunun/dönüşümünün hesaplanmasında kolaylık sağlar. Elde edilen dönüşüm matrisi; bağlantı çerçevesi koordinat sisteminden dünya koordinat sistemine olan dönüşümü çift yönlü sağlar.

$$H_e^0 = H_1^0 H_2^1 \dots H_{i-1}^{i-2} H_e^{i-1}$$
$$H_0^e = (H_e^0)^{-1}$$

9.3 DH Parametreleri

DH parametreleri; bir robot manipülatörünün bağlantılarına veya genel olarak bir kinematik zincire referans çerçevesi eklemek için kullanılan dört parametreyi tanımlayan bir konvansiyondur. (Bkz. Şekil 8). Bu konvansiyon, dört dönüşüm işlemine dayanır: önce $z^{\wedge}$ eksenini etrafında bir dönüş, ardından $z^{\wedge}$ eksenini doğrultusunda bir öteleme, daha sonra $x^{\wedge}$ eksenini etrafında bir dönüş ve son olarak $x^{\wedge}$ eksenini doğrultusunda bir öteleme gerçekleştirir. İşlemler sonucu elde edilen homojen dönüşüm matrisi aşağıdaki denklem ile ifade edilmektedir.

$$H_i = H_{R_z}(\theta_i) H_{T_z}(d_i) H_{R_x}(\alpha_i) H_{T_x}(a_i)$$
$$= \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dört nicelik θ_i , d_i , α_i ve a_i olup, referans çerçevesi numarası i ile indekslenmiştir. Bu nicelikler aşağıdaki ifadeleri temsil etmektedir:

- θ_i : x_{i-1} ekseninden x_i eksenine, z_{i-1} eksenini boyunca ölçülen bağlantı dönüş açısı.
- d_i : x_{i-1} ekseninden x_i eksenine, z_{i-1} eksenini boyunca ölçülen mesafe.
- α_i : z_{i-1} ekseninden z_i eksenine, x_i eksenini boyunca ölçülen bağlantı dönüş açısı.
- a_i : z_{i-1} ekseninden z_i eksenine, x_i eksenini boyunca ölçülen mesafe.

Bu niceliklerden ϑ_i veya d_i , bir değişken ya da sabit olabilir. Eğer i eklemi doğrusal ise, d_i bir değişken ve ϑ_i sabittir. Eğer eklem döner ise değişken ϑ_i , sabit ise d_i 'dir. Değişken parametre ϑ_i , eklem açısı veya q olarak adlandırılır. Her ikisi de sabit olursa, bu durumda i bağlantısı sabit bir bağlantı olur.

Bu konvansiyon, çerçeve koordinatlarının nasıl seçileceği ve yönlendirileceğine dair bir dizi kural uygular.

- Her bir eklem için sağ el kuralına uygun çerçeve kullan.
- z_i eksenini, $i + 1$ eklemının dönüş veya öteleme eksenini boyunca hizala.
- O_i orijinini; z_i eksenini ile z_{i-1} eksenini arasındaki ortak dik doğrultunun kesişim noktasında seçin.
- x_i eksenini, z_i ve z_{i-1} eksenleri arasındaki ortak dik doğrultunun yönüyle aynı yönde seçin.

9.4 İleri Kinematik Hesaplamaları

1995 yılında Denavit ve Hartenberg (DH parametrelerinin kaşifleri), robotun eklem zincirlerindeki her bir bağlantıyla ilişik koordinat sistemini oluşturmak için bitişik bağlantılar arasındaki öteleme ve dönme ilişkisini tanımlayan bir matris yöntemi önerdiler. Bu, D-H koordinat sistemi tabanında odaklanmış bir robot kinematik modelidir.

İki ardışık eklem arasındaki dönüşümler, parametre tablosundaki parametrelerin “ A_n ” adı verilen matristeki karşılık gelen yerlerine yerleştirilmesiyle yazılabilir.

$$A_{nf} = \begin{bmatrix} \cos \theta_{nf} & -\sin \theta_{nf} \cos \alpha_{nf} & \sin \theta_{nf} \sin \alpha_{nf} & a_{nf} \cos \theta_{nf} \\ \sin \theta_{nf} & \cos \theta_{nf} \cos \alpha_{nf} & \cos \theta_{nf} \sin \alpha_{nf} & a_{nf} \sin \theta_{nf} \\ 0 & \sin \alpha_{nf} & \cos \alpha_{nf} & d_{nf} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bilinen eklem parametreleri bu matristeki karşılık gelen yerlerine konulduğunda, tabandan (1) başlayarak uca (6) kadar her eklem için aşağıdaki matrisler elde edilir:

$$A_{1f} = \begin{bmatrix} \cos \theta_{1f} & 0 & \sin \theta_{1f} & a_{1f} \cos \theta_{1f} \\ \sin \theta_{1f} & 0 & -\cos \theta_{1f} & a_{1f} \sin \theta_{1f} \\ 0 & 1 & 0 & d_{1f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{2f} = \begin{bmatrix} \cos \theta_{2f} & -\sin \theta_{2f} & 0 & a_{2f} \cos \theta_{2f} \\ \sin \theta_{2f} & \cos \theta_{2f} & 0 & a_{2f} \sin \theta_{2f} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3f} = \begin{bmatrix} \cos \theta_{3f} & 0 & \sin \theta_{3f} & a_{3f} \cos \theta_{3f} \\ \sin \theta_{3f} & 0 & -\cos \theta_{3f} & a_{3f} \sin \theta_{3f} \\ 0 & 1 & 0 & d_{3f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{4f} = \begin{bmatrix} \cos \theta_{4f} & 0 & -\sin \theta_{4f} & 0 \\ \sin \theta_{4f} & 0 & \cos \theta_{4f} & 0 \\ 0 & -1 & 0 & d_{4f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{5f} = \begin{bmatrix} \cos \theta_{5f} & 0 & \sin \theta_{5f} & 0 \\ \sin \theta_{5f} & 0 & -\cos \theta_{5f} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{6f} = \begin{bmatrix} \cos \theta_{6f} & -\sin \theta_{6f} & 0 & 0 \\ \sin \theta_{6f} & \cos \theta_{6f} & 0 & 0 \\ 0 & 0 & 1 & d_{6f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bu raporda basitlik ve kısaltma niyetiyle aşağıdaki gösterim değişiklikleri kullanılmıştır:

$$\begin{aligned} C_n &= \cos \theta_n & S_n &= \sin \theta_n \\ C_{ab} &= C_a C_b - S_a S_b & S_{ab} &= C_a S_b + S_a C_b \end{aligned}$$

Robot tabanından uca kadar toplam dönüşüm matrisi şu şekildedir:

$$T_f = A_{1f} A_{2f} A_{3f} A_{4f} A_{5f} A_{6f} = \begin{bmatrix} n_{xf} & o_{xf} & a_{xf} & p_{xf} \\ n_{yf} & o_{yf} & a_{yf} & p_{yf} \\ n_{zf} & o_{zf} & a_{zf} & p_{zf} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Burada;

$$n_{xf} = C_{6f}(C_{5f}(C_{1f}C_{23f}C_{4f} + S_{1f}S_{4f}) - C_{1f}S_{23f}S_{5f}) + S_{6f}(S_{1f}C_{4f} - C_{1f}C_{23f}S_{4f})$$

$$n_{yf} = C_{6f}(C_{5f}(S_{1f}C_{23f}C_{4f} + C_{1f}S_{4f}) - S_{1f}S_{23f}S_{5f}) - S_{6f}(C_{1f}C_{4f} + S_{1f}C_{23f}S_{4f})$$

$$n_{zf} = C_{6f}(C_{23f}S_{5f} + S_{23f}C_{4f}C_{5f}) - S_{23f}S_{4f}S_{6f}$$

$$o_{xf} = S_{6f}(C_{1f}S_{23f}S_{5f} - C_{5f}(S_{1f}C_{23f}C_{4f} + S_{1f}S_{4f})) + C_{6f}(S_{1f}C_{4f} - C_{1f}C_{23f}S_{4f})$$

$$o_{yf} = S_{6f}(S_{1f}S_{23f}S_{5f} - C_{5f}(S_{1f}C_{23f}C_{4f} + C_{1f}S_{4f})) - C_{6f}(C_{1f}C_{4f} + S_{1f}C_{23f}S_{4f})$$

$$o_{zf} = -S_{6f}(C_{23f}S_{5f} + S_{23f}C_{4f}C_{5f}) - S_{23f}S_{4f}C_{6f}$$

$$a_{xf} = S_{5f}(C_{1f}C_{23f}C_{4f} + S_{1f}S_{4f}) + C_{1f}S_{23f}C_{5f}$$

$$a_{yf} = S_{5f}(S_{1f}C_{23f}C_{4f} - C_{1f}S_{4f}) + S_{1f}S_{23f}C_{5f}$$

$$a_{zf} = S_{23f}C_{4f}S_{5f} - C_{23f}C_{5f}$$

$$p_{xf} = d_{6f}(S_{5f}(C_{1f}C_{23f}C_{4f} + S_{1f}S_{4f}) + C_{1f}S_{23f}C_{5f}) + C_{1f}(a_{1f} + a_{2f}C_{2f} + a_{3f}C_{23f} + d_{4f}S_{23f})$$

$$p_{yf} = d_{6f}(S_{5f}(S_{1f}C_{23f}C_{4f} + C_{1f}S_{4f}) + S_{1f}S_{23f}C_{5f}) + S_{1f}(a_{1f} + a_{2f}C_{2f} + a_{3f}C_{23f} + d_{4f}S_{23f})$$

$$p_{zf} = a_{2f}S_{2f} + d_{1f} + a_{3f}S_{23f} - d_{4f}C_{23f} + d_{6f}(S_{23f}C_{4f}S_{5f} - C_{23f}C_{5f})$$

Elde edilen 4x4 dönüşüm matrisindeki en alt satır etkisiz satır olarak kabul edilir. En üstteki sol 3x3 matris dönme matrisidir ve en sağ sütundaki yukarıdan aşağıya doğru 3x1 matris ise dönüşüm matrisidir.

Buna göre robotun taban koordinat sisteminden robot ucuna doğru dönüş matrisi:

$$T_{Rf} = \begin{bmatrix} n_{xf} & o_{xf} & a_{xf} \\ n_{yf} & o_{yf} & a_{yf} \\ n_{zf} & o_{zf} & a_{zf} \end{bmatrix}$$

Burada:

- $[n_{xf} \ n_{yf} \ n_{zf}]$: birim vektör, robotun uç noktasındaki X ekseninin temel koordinat sistemine göre yönünü gösterir.
- $[o_{xf} \ o_{yf} \ o_{zf}]$: birim vektör, robotun uç noktasındaki Y ekseninin temel koordinat sistemine göre yönünü gösterir.
- $[a_{xf} \ a_{yf} \ a_{zf}]$: birim vektör, robotun uç noktasındaki Z ekseninin temel koordinat sistemine göre yönünü gösterir.

Birim vektör, robotun uç noktasındaki X,Y veya Z eksenlerinin temel koordinat sistemlerine göre yönünü gösterir. Ancak bu yönlerin vektörel form yerine açısal formda ifade edilmesi beklenmektedir. Aşağıda, uç nokta dönüşünün $ZY'Z''$ Euler ve $XY'Z''$ Tait-Bryan açılarında nasıl ifade edileceği gösterilmektedir. Bu açıları hesaplamak için kullanılan matrislere ilgili kaynaktan erişilebilir. [1]

$ZY'Z''$ robot ucu koordinat ekseninin taban koordinat eksenine olan Euler açıları(e simgesi Euler'i temsil eder):

$$Z_{ef} = \arctan2\left(\frac{a_{yf}}{a_{xf}}\right)$$

$$Y'_{ef} = \arctan2\left(\frac{\sqrt{1 - a_{zf}^2}}{a_{zf}}\right)$$

$$Z''_{ef} = \arctan2\left(\frac{o_{zf}}{-n_{zf}}\right)$$

Robotun uç nokta koordinat ekseninin, temel koordinat eksenine göre olan $XY'Z''$ Tait-Bryan açıları (alt simge t , Tait-Bryan'ı temsil etmektedir):

$$X_{tf} = \arctan2\left(\frac{-a_{yf}}{a_{zf}}\right)$$

$$Y'_{tf} = \arctan2\left(\frac{a_{xf}}{\sqrt{1 - a_{xf}^2}}\right)$$

$$Z''_{ef} = \arctan2\left(\frac{-o_{xf}}{n_{xf}}\right)$$

Robotun temel koordinat sisteminden uç noktasına olan öteleme matrisi:

$$T_{Tf} = \begin{bmatrix} p_{xf} \\ p_{yf} \\ p_{zf} \end{bmatrix}$$

Hem ötelemeyi hem de $ZY'Z''$ Euler açılarını içeren dönüşüm matrisi şu şekilde ifade edilebilir:

$$T_{ef} = \begin{bmatrix} p_{xf} \\ p_{yf} \\ p_{zf} \\ Z_{ef} \\ Y'_{ef} \\ Z''_{ef} \end{bmatrix}$$

Bunlar, ileri kinematik hesaplamışını tamamlamaktadır. Devam eden ters kinematik hesaplamalarında türetmiş olduğumuz T_{ef} matrisi girdi alınacak ve θ_n eklem dönüşleri elde edilecektir.

9.5 Ters Kinematik Hesaplamaları

9.4 bölümünde eklem dönme açılarını girerek end-effector'ün kartezyen uzaydaki konumunu bulmuştuk. Gerçek dünyada ise robotlar tam tersi yönde çalışır; robot kontrolcüsü, belirli bir uç efektör konumu için eklem açılarını bilmek zorundadır. Bu işlem, bu bölümde açıklanacak olan ters kinematik hesaplamasıyla gerçekleştirilir. Ters hesaplamanın giriş değişkenleri, önceki bölümde türetilen T_{ef} matrisinin hücre değerleri olacaktır, ancak tüm alt simgeler de f 'den i 'ye değiştirilir ve bu nedenle giriş matrisimiz şu şekilde olacaktır:

$$T_{ei} = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \\ Z_{ei} \\ Y'_{ei} \\ Z''_{ei} \end{bmatrix}$$

Her bir eklemin dönme açıları, T_{ei} matrisindeki değerler kullanılarak hesaplanacaktır. Bunu yapmak için, T_{ei} matrisinden dönüşüm matrisi ters yönde oluşturulacaktır.

Dönüşüm matrisinin öteleme kısmı belirgindir, $[p_{xi} \ p_{yi} \ p_{zi}]$. Dönüşüm matrisinin en sağdaki sütununun üst kısmını dikey olarak oluşturur. Dönüşüm matrisini oluşturmak için, giriş olarak verilen dönme değerlerini ilgili Euler dönüş matrisine yerleştirmek yeterlidir. Kaynak[1]'de ZY'Z'' Euler açılarına karşılık gelen dönüş matrisi hesaplaması zaten belirtilmiştir.

Euler açılarından dönüş matrisi oluşturmak için aşağıdaki yerleştirmeler, gösterimler ve kısaltmalar kullanılmıştır. Ek olarak, ters hesaplamayı göstermek için sadece i indeksi alt simge olarak eklenmiştir:

$$\begin{array}{lll} \alpha_i = Z_{ei} & \beta_i = Y'_{ei} & \gamma_i = Z''_{ei} \\ c_{1i} = \cos(\alpha_i) & c_{2i} = \cos(\beta_i) & c_{3i} = \cos(\gamma_i) \\ s_{1i} = \sin(\alpha_i) & s_{2i} = \sin(\beta_i) & s_{3i} = \sin(\gamma_i) \end{array}$$

Dönüşüm matrisindeki dönüşüm kısmında (sol üstteki 3x3 matris) bu kısaltmalar kullanılarak:

$$T_i = \begin{bmatrix} c_{1i}c_{2i}c_{3i} - s_{1i}s_{3i} & -c_{3i}s_{1i} - c_{1i}c_{2i}s_{3i} & c_{1i}s_{2i} & p_{xi} \\ c_{1i}s_{3i} - c_{2i}c_{3i}s_{1i} & c_{1i}c_{3i} - c_{2i}s_{1i}s_{3i} & s_{1i}s_{2i} & p_{yi} \\ -c_{3i}s_{2i} & s_{2i}s_{3i} & c_{2i} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ayrıca önceki bölümde ileri kinematik hesaplamasında da gösterildiği gibi, bu dönüşüm matrisinin kısaltılmış görünümü şu şekildedir:

$$T_f = \begin{bmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_{1i}A_{2i}A_{3i}A_{4i}A_{5i}A_{6i}$$

Matrisin içinde öteleme değerleri hali hazırda belirlenmiştir. Dönüş değerleri ise şöyledir:

$$\begin{aligned} n_{xi} &= c_{1i}c_{2i}c_{3i} - s_{1i}s_{3i} & o_{xi} &= -c_{3i}s_{1i} - c_{1i}c_{2i}s_{3i} & a_{xi} &= c_{1i}s_{2i} \\ n_{yi} &= c_{1i}s_{3i} - c_{2i}c_{3i}s_{1i} & o_{yi} &= c_{1i}c_{3i} - c_{2i}s_{1i}s_{3i} & a_{yi} &= s_{1i}s_{2i} \\ n_{zi} &= -c_{3i}s_{2i} & o_{zi} &= s_{2i}s_{3i} & a_{zi} &= c_{2i} \end{aligned}$$

Robot manipülâtörünün her bir eklem açısını bulma süreci iki adımda yapılabilir:

- İlk üç eklemin açısını bulmak,
- İlk üç açının kullanılmasıyla sonraki üç küresel eklem açısını bulmak.

Hesaplamalarının okunabilirliğini arttırma amacıyla ilk eklemin bulunmasındaki geometrik açıklama [2] kaynağında sağlanmış olup θ_{1i} olarak kabul edilecektir.

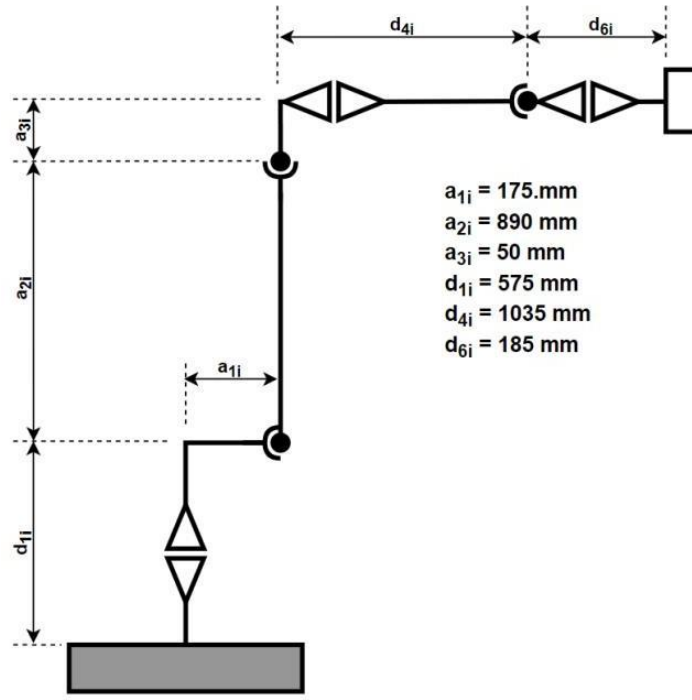
$$P_{04} = P_{06} - P_{46} \text{ where, } P_{06} = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \end{bmatrix}, P_{46} = d_{6i} \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix}$$

$$P_{04} = \begin{bmatrix} x_{05} \\ y_{05} \\ z_{05} \end{bmatrix} = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \end{bmatrix} - d_{6i} \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix} = \begin{bmatrix} p_{xi} - d_{6i}a_{xi} \\ p_{yi} - d_{6i}a_{yi} \\ p_{zi} - d_{6i}a_{zi} \end{bmatrix}$$

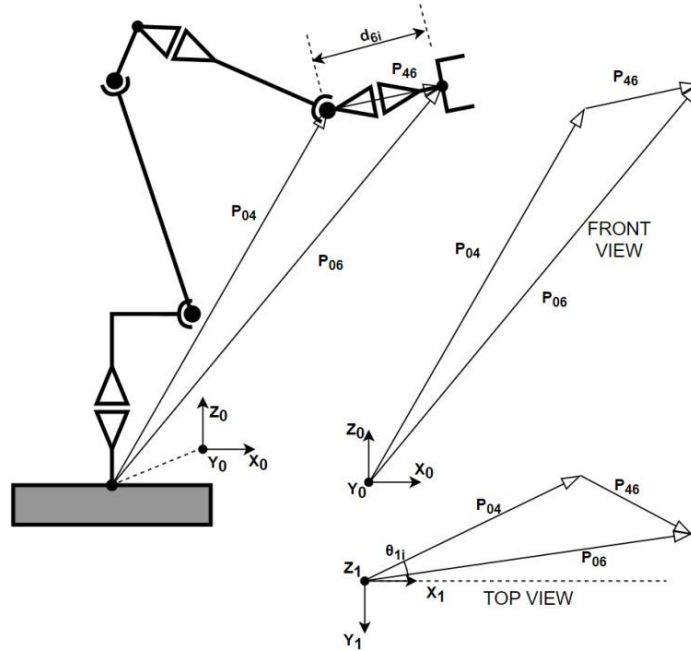
(Projenin bu bölümünde gösterilen robot manipülâtör 6RK'in 2D ve 3D çizimlerindeki verilerin daha karmaşık olması sebebiyle farklı örnek bir robot manipülâtör kullanılarak sağlanmıştır).

Böylece:

$$\theta_{1i} = \theta_{1v} = \arctan2 \left(\frac{p_{yi} - d_{6i}a_{yi}}{p_{xi} - d_{6i}a_{xi}} \right)$$



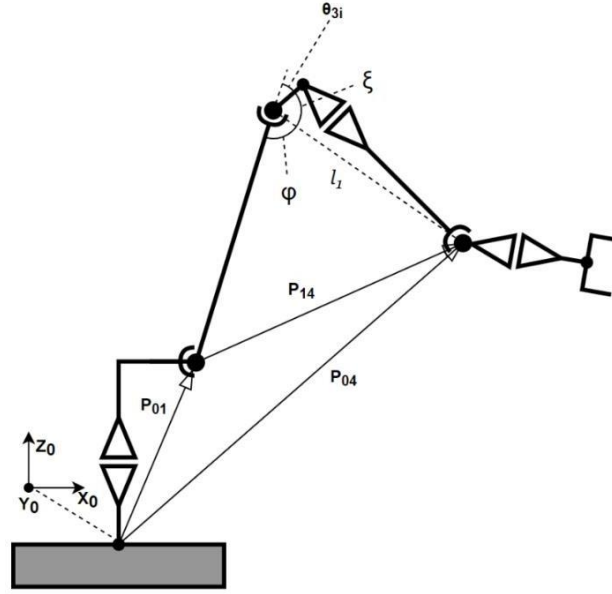
Şekil 10 : Robot manipulatörün anahtar verileriyle birlikte 2D sunumu.



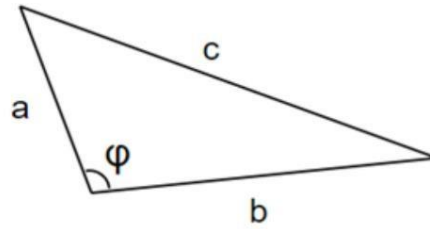
Şekil 11: Grafiksel açıklamasıyla θ_{1i} sağlanması.

Ayrıca Şekil 12’de gösterilen geometrik çözüm ve kosinüs kanunu (Şekil 13) kullanılarak, θ_{3i} şu şekilde türetilbilir:

$$P_{01} = \begin{bmatrix} x_{01} \\ y_{01} \\ z_{01} \end{bmatrix} = \begin{bmatrix} a_{1i} C_{1i} \\ a_{1i} S_{1i} \\ d_{1i} \end{bmatrix}, P_{04} = \begin{bmatrix} x_{04} \\ y_{04} \\ z_{04} \end{bmatrix} = \begin{bmatrix} p_{xi} - d_{6i} a_{xi} \\ p_{yi} - d_{6i} a_{yi} \\ p_{zi} - d_{6i} a_{zi} \end{bmatrix}$$



Şekil 12: θ_{3i} çıkarımının grafiksel çözümlemesi



$$\cos(\varphi) = \frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b}$$

Şekil 13: Kosinüs Kanunu

Böylece:

$$P_{14} = \begin{bmatrix} x_{14} \\ y_{14} \\ z_{14} \end{bmatrix} = P_{04} - P_{01} = \begin{bmatrix} p_{xi} - d_{6i}a_{xi} - a_{1i}C_{1i} \\ p_{yi} - d_{6i}a_{yi} - a_{1i}S_{1i} \\ p_{zi} - d_{6i}a_{zi} - d_{1i} \end{bmatrix}$$

ve,

$$P_{14L} = \sqrt{|P_{14}^T \cdot P_{14}|}, \quad l_1 = \sqrt{a_{3i}^2 + d_{4i}^2}$$

Burada L alt simgesi, ilgili vektörün uzunluğunu temsil etmektedir. Kosinüs kanunu ve iyi bilinen trigonometrik formül uygulanarak;

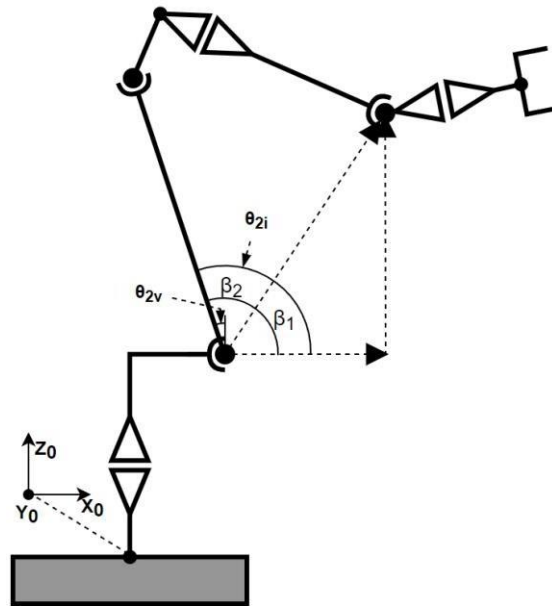
$$\varphi = \arccos\left(\frac{l_1^2 + a_{2i}^2 - P_{14L}}{2l_1a_2}\right), \quad \zeta = \arctan2\left(\frac{d_{4i}}{a_{3i}}\right)$$

Bu nedenle:

$$\theta_{3i} = \theta_{3v} = \varphi - \zeta - \pi$$

Yine Şekil 14'te açıklanan geometrik çözüm, kosinüs kanunu (Şekil 13) ve iyi bilinen trigonometrik formül kullanılarak, θ_{2i} şu şekilde türetilir:

$$P_{14} = \begin{bmatrix} x_{14} \\ y_{14} \\ z_{14} \end{bmatrix} = \begin{bmatrix} p_{xi} - d_{6i}a_{xi} - a_{1i}C_{1i} \\ p_{yi} - d_{6i}a_{yi} - a_{1i}S_{1i} \\ p_{zi} - d_{6i}a_{zi} - d_{1i} \end{bmatrix}$$



Şekil 14: θ_{2i} çıkarımının grafiksel çözümlemesi

Önceki denklemden, β_1 ve β_2 şu şekilde bulunabilir;

$$\beta_1 = \arctan2\left(\frac{z_{14}}{\sqrt{x_{14}^2 + y_{14}^2}}\right)$$

$$\beta_2 = \arccos\left(\frac{a_{2i}^2 - p_{14L}^2 - l_1^2}{2a_2 p_{14L}}\right)$$

Bu geometriden:

$$\theta_{2i} = \theta_{2v} + \frac{\pi}{2} = \beta_1 + \beta_2$$

Ters hesaplamamanın ilk aşaması, ilk üç eklem dönme açısının bulunmasıyla tamamlanır. İkinci ve son aşama, küresel bileği oluşturan diğer üç eklemin geometrik olarak çözülmesiyle gerçekleştirilecektir.

θ_{5i} 'yi hesaplamak için $\theta_{4i} = 0$ olduğunu ya da 4. eklemin olmadığını varsayacağız. Bu durumda, dönme vektörleri (R_z) arasındaki açı bize θ_{5i} 'yi sağlayacaktır.

Bundan önce, sonraki hesaplamalara temel oluşturacak bazı eşitliklerin belirtilmesi gerekmektedir:

$$A_{1i} = \begin{bmatrix} C_{1i} & 0 & S_{1i} & a_{1i}C_{1i} \\ S_{1i} & 0 & -C_{1i} & a_{1i}S_{1i} \\ 0 & 1 & 0 & d_{1i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

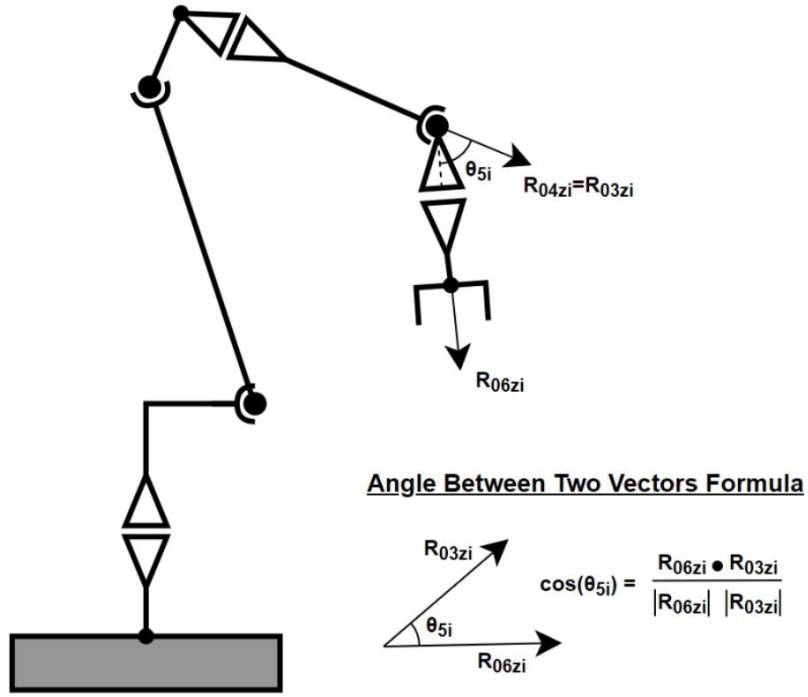
$$A_{2i} = \begin{bmatrix} C_{2i} & -S_{2i} & 0 & a_{2i}C_{2i} \\ S_{2i} & C_{2i} & 0 & a_{2i}S_{2i} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3i} = \begin{bmatrix} C_{3i} & 0 & S_{3i} & a_{3i}C_{3i} \\ S_{3i} & 0 & -C_{3i} & a_{3i}S_{3i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Aşağıdakiler de ayrıca belirtilmelidir:

$$C_{12} = C_1C_2 - S_1S_2 \quad S_{12} = C_1S_2 + S_1C_2$$

5. eklem açısını bulabilmek için, 4. eklemin olmadığı varsayılmaktadır. 1. ile 3. eklemler arasındaki dönüşüm matrisi, aynı zamanda 1. ile 4. eklemler arasındaki dönüşüm matrisi olarak kabul edilecektir. İlk üç eklem açısı bilindiğinden A_{13} ve dolayısıyla A_{14} bulunabilir.



Şekil 15: θ_{5i} çıkarımının grafiksel çözümlemesi

$$A_{14i} = A_{13i} = A_{1i} \cdot A_{2i} \cdot A_{3i} = \begin{bmatrix} C_{1i}C_{23i} & S_{1i} & C_{1i}S_{23i} & C_{1i}(a_{1i} + a_{2i}C_{2i} + a_{3i}C_{23i}) \\ S_{1i}C_{23i} & -C_{1i} & S_{1i}S_{23i} & S_{1i}(a_{1i} + a_{2i}C_{2i} + a_{3i}C_{23i}) \\ S_{23i} & 0 & -C_{23i} & a_{2i}S_{2i} + d_{1i} + a_{3i}S_{23i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Giriş değerlerinden, z eksenı etrafındaki dönme vektörü (R_{6zi}) bilinmektedir. Yukarıdaki “ $R_{4zi} = R_{3zi}$ ” varsayımı sayesinde R_{4zi} de bilinmektedir. Bundan sonra “ $\theta_{5i} = \theta_{5v}$ ” eşitliği ile gösterilen iki vektör arasındaki açı formülü uygulanarak bilinmeyen θ_{5i} bulunacaktır.

$$R_{6zi} = \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix}, \quad R_{3zi} = R_{4zi} = \begin{bmatrix} C_{1i}S_{23i} \\ S_{1i}S_{23i} \\ -C_{23i} \end{bmatrix}$$

R_{6zi} ve R_{4zi} , ikisi de birim vektör olduğu için, matris çarpımı iki vektör arasındaki açının kosinüsünü verecektir.

$$\theta_{5i} = \theta_{5v} = \arccos(\overrightarrow{R_{6zi}} \cdot \overrightarrow{R_{3zi}})$$

Kalan bilinmeyen eklem açıları yalnızca 4. ve 6. eklemler olarak kalmıştır.

A_{13i} A_{16i} bilindiği için A_{46i} aşağıdaki gibi türetilir:

$$A_{46i} = A_{13i}^{-1} \cdot A_{16i} = \begin{bmatrix} I_x & J_x & K_x & L_x \\ I_y & J_y & K_y & L_y \\ I_z & J_z & K_z & L_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Burada tüm I , J , K , ve L değerleri hesaplanmış ve bilinir vaziyettedir.

Başka bir yönden, A_{46i} matrisi, aşağıdaki matrislerin çoğaltılmasıyla sembolik olarak türetilir.

$$A_{46i} = A_{4i} \cdot A_{5i} \cdot A_{6i} = \begin{bmatrix} E_x & F_x & G_x & H_x \\ E_y & F_y & G_y & H_y \\ E_z & F_z & G_z & H_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Burada:

$$\begin{aligned} E_x &= C_{4i}C_{5i}C_{6i} - S_{4i}S_{6i} & F_x &= -(C_{4i}C_{5i}S_{6i} + S_{4i}C_{6i}) & G_x &= C_{4i}S_{5i} & H_x &= C_{4i}S_{5i}d_{6i} \\ E_y &= S_{4i}C_{5i}C_{6i} + C_{4i}S_{6i} & F_y &= -S_{4i}C_{5i}S_{6i} + C_{4i}C_{6i} & G_y &= S_{4i}S_{5i} & H_y &= S_{4i}S_{5i}d_{6i} \\ E_z &= -S_{5i}C_{6i} & F_z &= S_{5i}S_{6i} & G_z &= C_{5i} & H_z &= C_{5i}d_{6i} + d_{4i} \end{aligned}$$

Önceki sembolik matristen, matrisin 2.3 (H_y) elemanının 1.3 (H_y) elemanına bölünmesiyle elde edilen ters tanjantın ($\arctan2$) θ_{4i} 'yi ve 3.2 (F_z) elemanının 3.1 (E_z) elemanına bölünmesiyle elde edilen ters tanjantın θ_{6i} 'yi verdiği kolaylıkla fark edilir.

$$\theta_{4i} = \theta_{4v} = \arctan2\left(\frac{H_y}{H_x}\right) = \arctan2\left(\frac{L_y}{L_x}\right)$$

ve,

$$\theta_{6i} = \theta_{6v} = \arctan2\left(\frac{F_z}{E_z}\right) = \arctan2\left(\frac{I_z}{I_z}\right)$$

Böylelikle geometrik çözümleme yaklaşımı kullanarak 6 eklem tüm bilinmeyen açıları bulunabilmektedir.

10 “6RK” Robot Kolun Tasarımı ve Üretimi

9 numaralı bölümde tanımladığımız robot manipülatörün tasarımı hakkında edindiğimiz bilgileri dahilinde; yaygın CAD (computer-aided design) yazılımlarından biri olan “SolidWorks” programını kullanarak 6RK’in tasarımı ve modellemesi gerçekleştirilmiştir.



Şekil 16: 6RK CAD Görünümü

Robot, kontrol edilebilirliği adımına ulaşmadan önce fiziki olarak sahip olmamız gereken bir niceliktir. Bu noktada tasarımın güncel üretim yöntemleri göz ardı edilmeden yapılması ve montajlanabileceği senaryonun da hazırlanması gerekmektedir. Bu aşamalar tasarım kısmında sağlanmıştır.

Robotun hareketi ve elektronik bağlantılar için gerekli komponentler de bu aşamada tasarıma eklenerek tam bir ön izleme sunulmuştur.

Üretim yöntemi projenin kapsamı ve temel hedefleri doğrultusunda 3 boyutlu baskı yöntemi seçilmiştir. Parça tasarımları “FDM” tipindeki yazıcılar dikkate alınarak yapılmıştır ve montaj ilişkileri de böylelikle sağlanmıştır.

Malzeme tercihi de robotun çalışma koşullarının endüstriyel gereksinimlere sahip olmaması ve kapsam dışına çıkmama kaygısıyla PLA (polilaktik asit) olarak tercih edilmiştir.

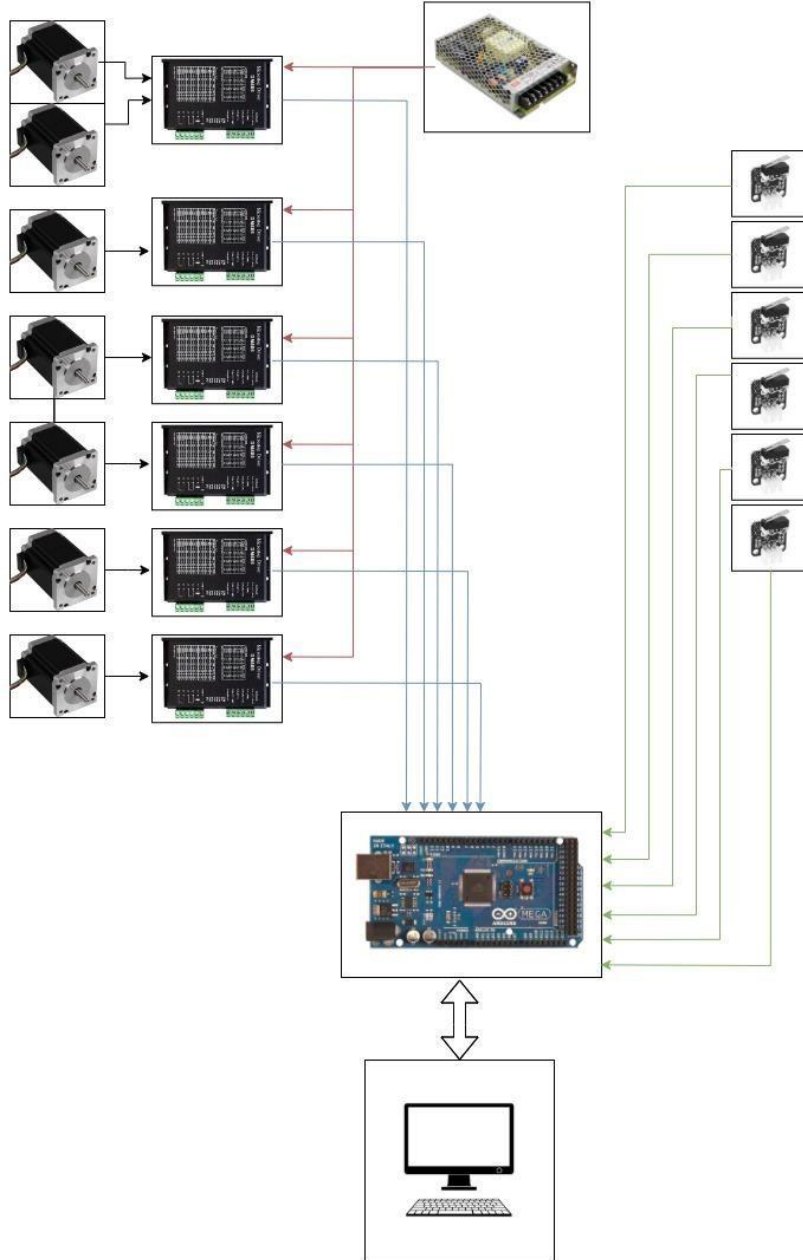
Tasarlanmış 30 adet 3 boyutlu baskı parçası ile toplamda 3369 gram baskı ihtiyacı sağlanmıştır. Montaj için gerekli rulman, kayış, vida vb. malzemeler temin edilip montaj aşamasına geçilmiştir.



Şekil 17: Baskısı alınan parçaların görseli

11 Elektronik Sistem Tasarımı

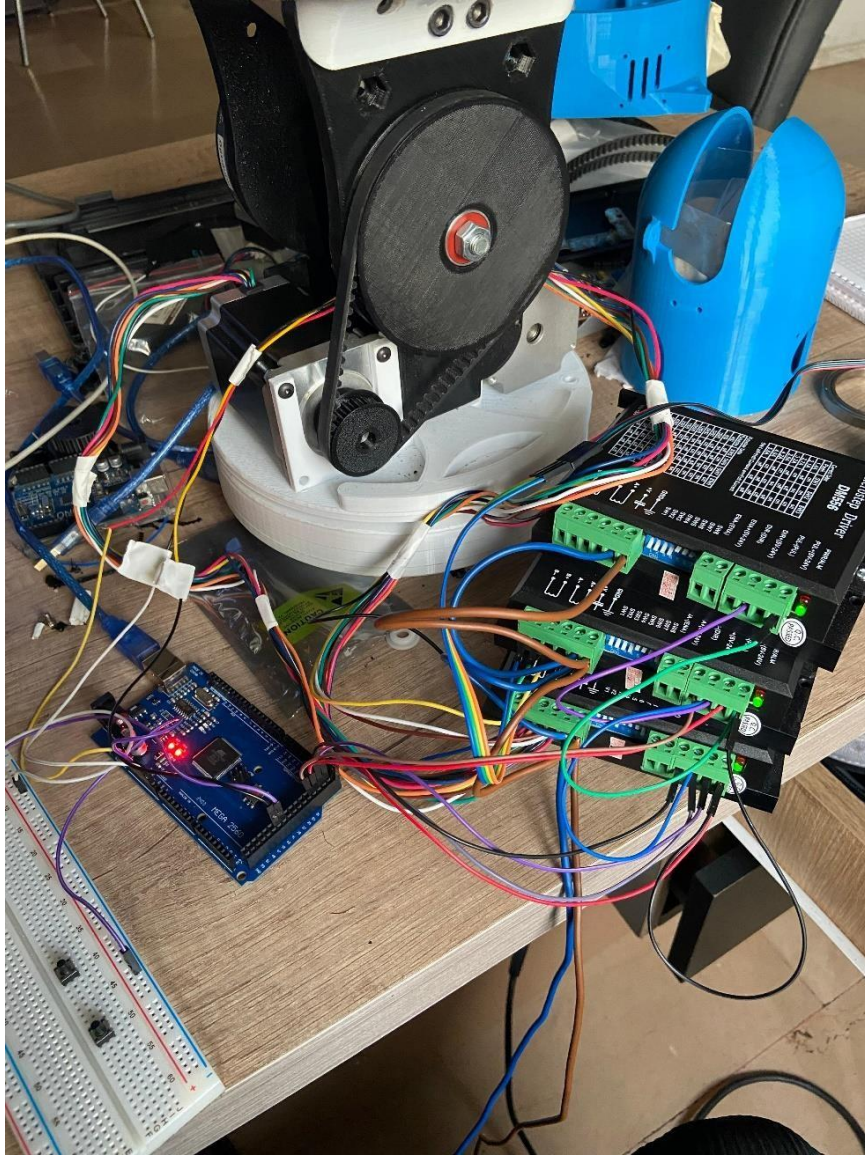
Proje hedeflerinin belirlenmesinde etki eden ve etkilenen başlıklardan biri de elektronik sistemdir. Robotun ihtiyaç duyduğu elektrik, sensör, mikrodenetleyiciler, motorlar ve motor sürücüler gibi komponentlerden bu bölümde kısaca bahsedilmiştir.



Şekil 18: Elektronik Sistem Şeması

Şekil 18’de tanımlanan devre şeması kısaca robotun iletişim ve elektrik bağlantı senaryosunu temsil etmektedir. Step motorlar sürücü aracılığı ile kontrol edilmektedir. Limit switchler Arduino üzerinden “boolean” tipinde veri iletimini sağlamaktadır. Güç kaynağı motorlar ve sürücüler için gerekli beslemeyi sağlamaktadır.

Mikro denetleyici olarak kullandığımız Arduino -sonraki bölümlerde bilmemizde fayda olan- “6RK CONTROL” yazılım programımız ile bilgisayar üzerinden seri haberleşme (UART) kullanılarak kontrol edilmektedir.



Şekil 19: Montaj ve elektronik sistem testine ait bir görsel

6RK CONTROL yazılımının geliştirme sürecindeki minimum gereksinim doğrultusunda fiziksel sistemin oluşumu Şekil 19’da gösterilmiştir.

IV Tasarım Dokümantasyonu

12 Tasarım Hedeflerinin Tanımlanması

Tasarım hedeflerinin tanımlanması, bir yazılım ürününün başarısı için kritik bir adımdır. Bu hedefler, sistemin geliştirme sürecindeki öncelikleri belirleyerek gereksinimlerin doğru bir şekilde karşılanmasını ve projenin etkin yönetimini sağlar. İyi tanımlanmış tasarım hedefleri, geliştirme sürecinde karar alma mekanizmalarını destekler ve nihai ürünün kullanıcı ihtiyaçlarına uygun, verimli ve sürdürülebilir olmasını sağlar.

Reliability (Güvenilirlik):

Robotun doğru bir şekilde çalışmasını sağlamak, kullanıcıların güvenliğini ve sistemin tutarlılığını garanti altına almak için kritik bir tasarım hedefidir.

Efficiency (Verimlilik):

Gerçek zamanlı kontrol ve hızlı tepki süreleri, robot manipülatörünün etkili bir şekilde kullanılabilmesi için gereklidir.

Maintainability (Bakım Yapılabilirlik):

Yazılımın gelecekte güncellenmesi veya iyileştirilmesi kolay olmalıdır. Özellikle farklı robot modellerine uyarlanabilir bir yapı önemlidir.

Modifiability (Değiştirilebilirlik):

Arduino bağlantısı veya web kontrol gibi farklı bileşenlerde değişiklik yapılabilir olmalıdır.

Ease of Use (Kullanım Kolaylığı):

Kullanıcı arayüzünün anlaşılır ve basit olması, kullanıcıların sistemi hızlı bir şekilde öğrenmesini ve etkin bir şekilde kullanmasını sağlayacaktır.

Traceability of Requirements (Gereksinim İzlenebilirliği):

Kullanıcı ihtiyaçlarına uygun özelliklerin yazılımda tam olarak uygulanması ve bu süreçte gereksinimlerin izlenebilir olması önemlidir.

Reusability (Yeniden Kullanılabilirlik):

Yazılımın belirli modülleri, benzer projelerde veya farklı robot manipülatörlerinde yeniden kullanılabilir şekilde tasarlanmalıdır.

Robustness (Güçlülük):

Kullanıcı hataları veya beklenmedik girişler karşısında sistemin stabil kalması ve hata durumlarını yönetebilmesi gereklidir.

Well-defined Interfaces (İyi Tanımlanmış Arayüzler):

Arduino erişimi, robotun ana kontrol sistemi, web bağlantısı gibi karışık işlemlerin kullanıcı tarafından kolayca kullanılabilmesini sağlayan arabirimlerin sistemle entegrasyonunun sorunsuz olarak sağlanması gerekmektedir.

User-friendliness (Kullanıcı Dostu Tasarım):

Farklı cihazlardan erişim sağlayan kullanıcıların kolayca sistemle etkileşim kurması önemlidir.

Güvenilirlik	Verimlilik	Bakım Yapılabilirlik	Değiştirilebilirlik
Kullanım Kolaylığı	Gereksinim İzlenebilirliği	Yeniden Kullanılabilirlik	Güçlülük
İyi Tanımlanmış Arayüzler	Kullanıcı Dostu Tasarım		

13 Önerilen Yazılım Mimarisi

Proje kapsamında oluşturduğumuz deployment diyagramı, 6RK Control yazılımının fiziksel bileşenlerini ve bu bileşenler arasındaki ilişkileri görsel olarak gösterir. Diyagram, sistemin donanım ve yazılım alt bileşenlerinin nasıl yapılandırıldığını, birbirleriyle nasıl etkileşimde bulunduklarını ve verilerin nasıl akış sağladığını anlatmaktadır.

Bu diyagram, özellikle robot manipülatörünün kontrolü için gerekli olan tüm donanım ve yazılım bileşenlerinin bir araya geldiği bağlantıları ve alt sistemlerin işlevlerini ortaya koymaktadır.

Kullanıcılar, web arayüzü üzerinden sisteme bağlanarak, Arduino ile iletişim kurar ve robot manipülatörünü kontrol eder. Ayrıca, yazılımın farklı katmanları (web arayüzü, kontrol katmanı, bağlantı katmanı gibi) ve bunlar arasındaki veri akışını net bir şekilde görselleştirir.

Amaçlar:

- *Sistem Bileşenlerinin Tanımlanması:*

Sistemin her bir bileşeninin ve bu bileşenlerin işlevselliğinin belirlenmesi.

- *İletişim Yapısının Gösterilmesi:*

Yazılım ve donanım arasındaki veri akışını ve bağlantıları açıklamak.

- *Fiziksel Yapıların Göstergesi:*

Robotun kontrol edilmesi için oluşturulan sistemin; kullanılan donanımlar ve yazılımlar arasındaki ilişkilerin görsel bir şekilde sunulması.

- *Geliştirme Sürecinin İyileştirilmesi:*

Diyagram sayesinde geliştirme sürecinde her bileşenin nerede yer aldığı ve nasıl etkileşeceği daha açık bir şekilde anlaşılır.

- *Entegrasyonun Kolaylaştırılması:*

Farklı sistemler arasında entegrasyonun nasıl sağlanacağı ve yazılım ile donanım arasındaki uyumun nasıl sağlandığı konusunda netlik kazandırmak.



Şekil 20: 6RK Control Deployment Diyagramı

13.1 Diyagram Açıklamaları

Web Uygulaması:

Kullanıcının, robotu uzaktan kontrol etmek için kullandığı; HTTPS protokolü ile erişim sağladığı web tabanlı uygulama.

Kontrol Arayüzü:

Kullanıcının robotun bağlı olduğu bilgisayar üzerinden erişim sağladığı ve isterse web kullanıcılarının robota erişimine izin sağladığı; offline ve online seçenekleri ile robotun kontrol edildiği temel masaüstü uygulaması.

Web Sunucusu:

Alan adı “hakanloyan.com” olan web sitesi üzerinde yayınlanacak web uygulamasının “hosting” hizmetini temsil eder.

Veritabanı Sunucusu:

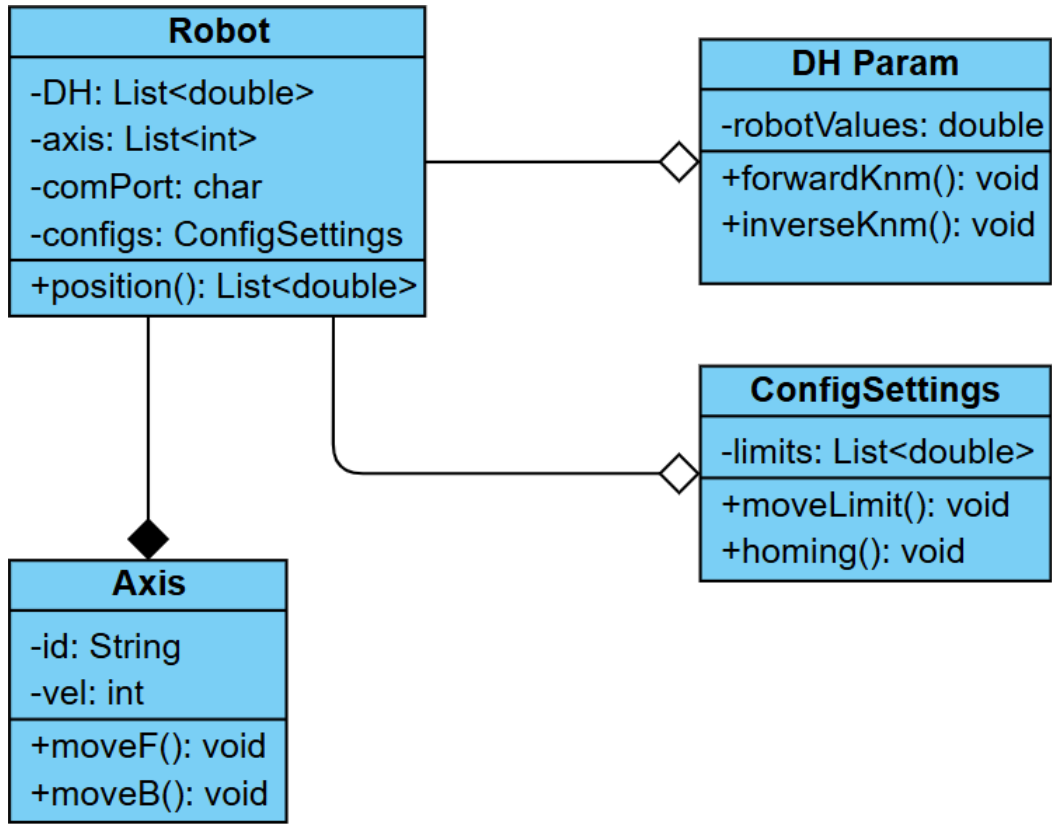
Web uygulamasına farklı cihazlar ve ağlar üzerinden erişimde veritabanının güncel kalması adına “Firebase” üzerinden sağlanan veritabanı sunucusu hizmetini temsil eder.

Arduino:

Masaüstü uygulaması arayüzü üzerinden kullanıcının belirlediği port ile 6RK yazılımıyla seri haberleşme (UART) ile iletişim sağlayan kontrol kartı, robotu kontrol eden donanım.

14 Sınıf Diyagramları

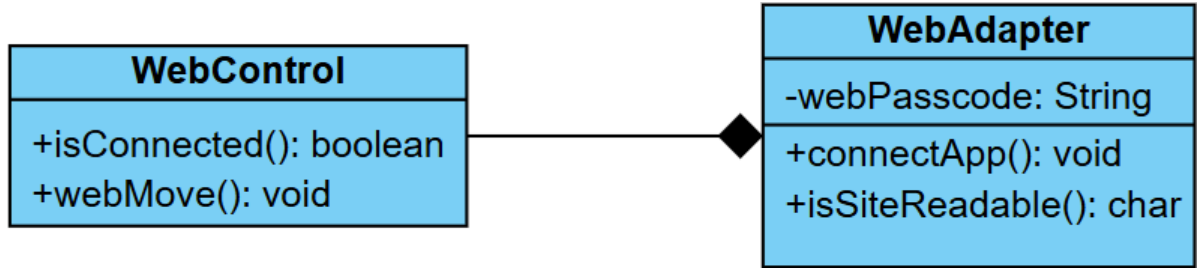
Sınıf diyagramları, bir yazılım sisteminin statik yapısını ve sınıflar arasındaki ilişkileri görsel olarak temsil eden önemli bir araçtır. Bu diyagramlar, sistemdeki nesnelerinin birbirleriyle nasıl etkileşime girdiğini, hangi sınıfların hangi veri ve işlevleri kapsadığını gösterir. Yazılım geliştirme sürecinde, sınıf diyagramları spesifik olarak nesne yönelimli programlama paradigmasını benimseyen projelerde, sistemin mimarisini anlamak ve doğru bir yapı inşa etmek adına büyük bir öneme sahip olmaktadır. Her bir sınıf, bir nesnenin özelliklerini ve işlevselliğini temsil ederken sınıflar arası ilişkiler ise sistemin nasıl çalıştığını daha açık bir şekilde ifade eder.



Şekil 21: Class Diyagramı

Robot sınıfı Axis subclass'ını içermek zorundadır.

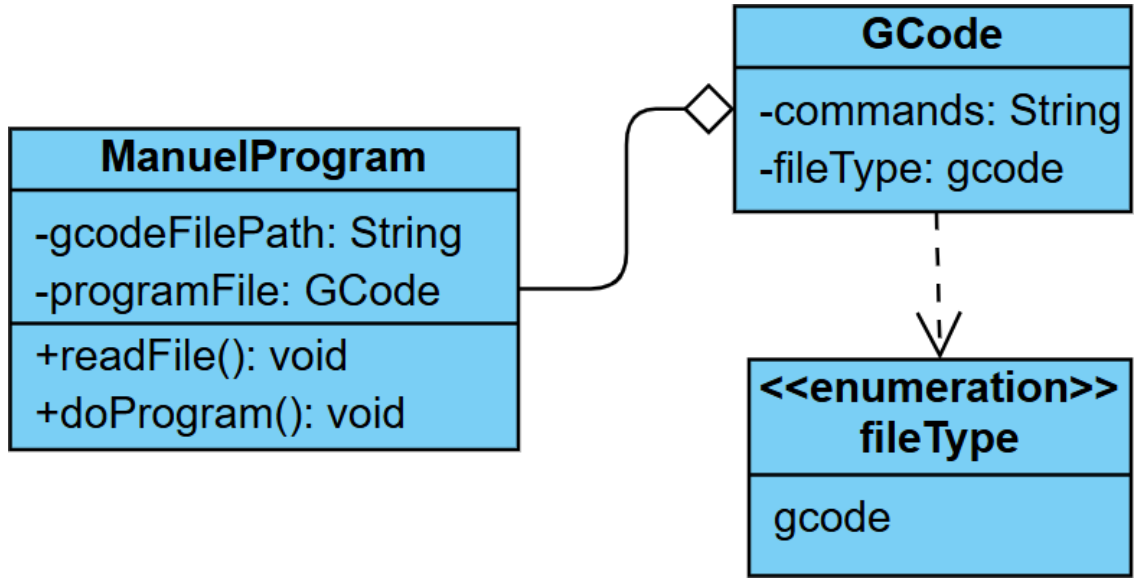
DHParam ve ConfigSettings Robot tanımlamak için bir zorunluluk değildir.



Şekil 22: Class Diyagramı

WebControl nesnesi oluşturmak için WebAdapter subclass'ı bir gerekliliktir.

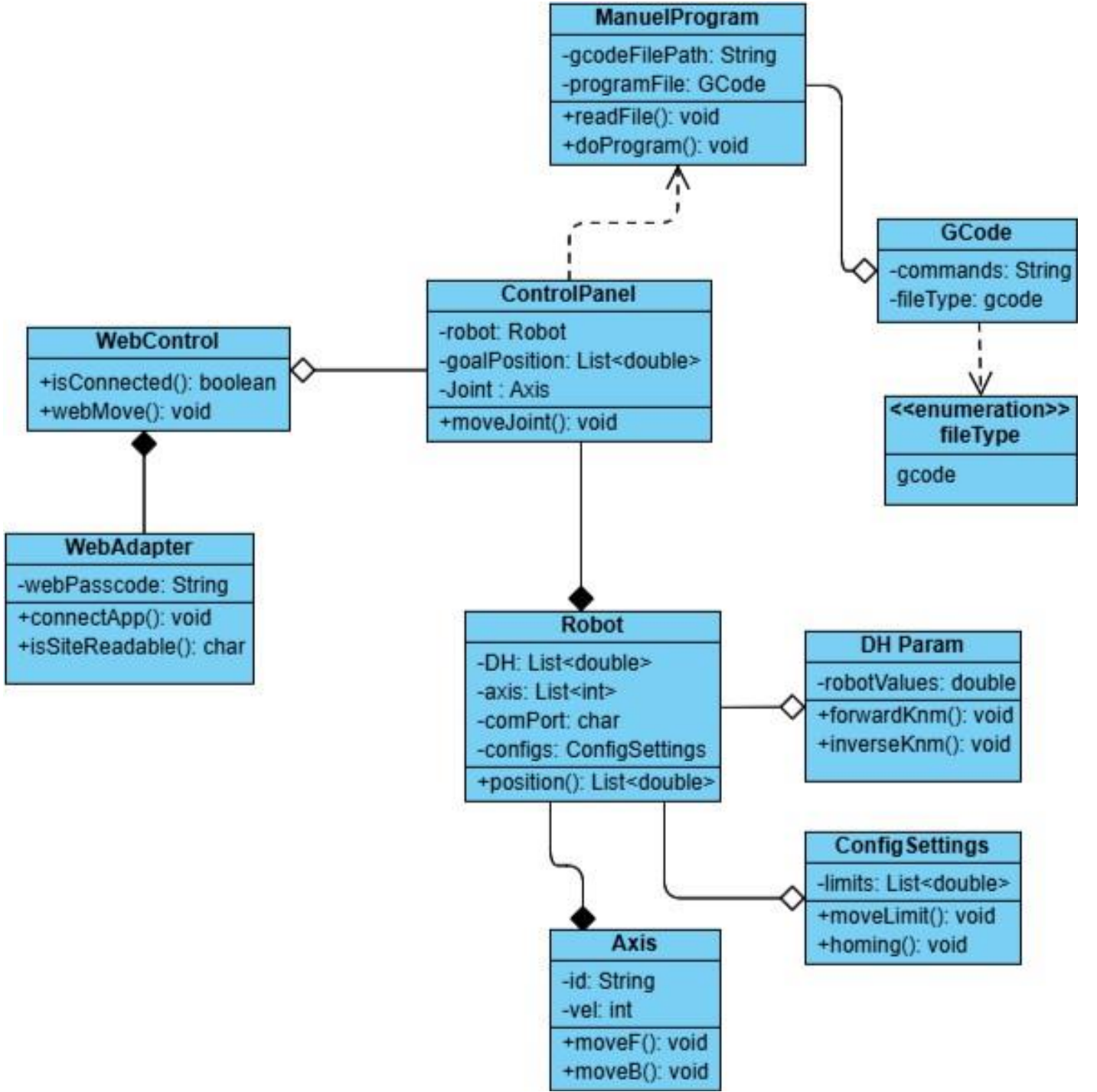
WebControl nesnesini oluşturmak için farklı bir sınıfa association yoktur.



Şekil 22: Class Diyagram

ManuelProgram için GCode sınıfı yani bir program dosyası aggregation olarak konumlanmıştır.

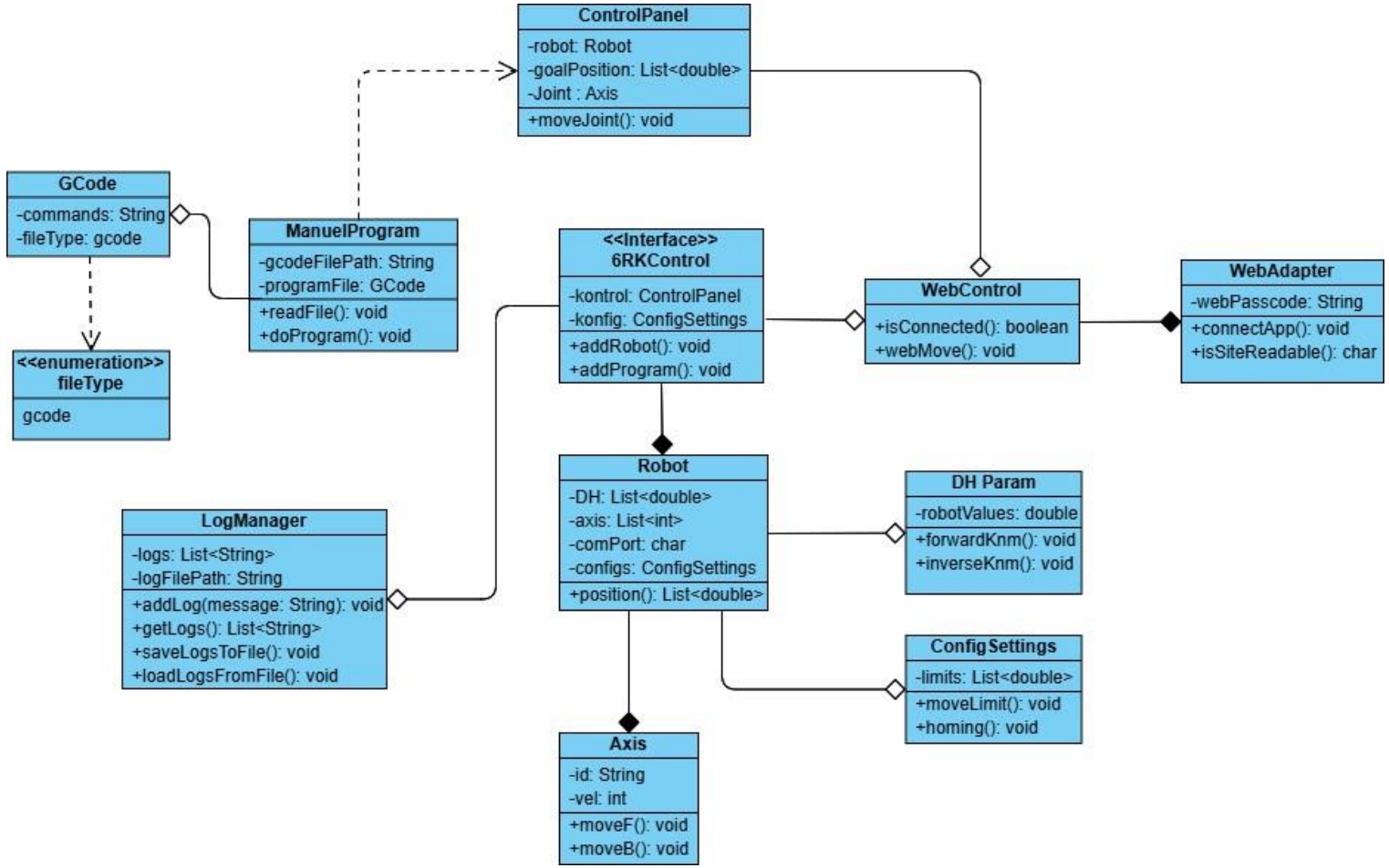
ManuelProgram nesnesi kullanıcı tarafından arayüz üzerinden de oluşturulabileceği için doğrudan bir dosya eklenmesi zorunlu değildir.



Şekil 23: Class Diyagram

ControlPanel sınıfı arayüz üzerinde kullanıcının tüm etkileşimlerini içermektedir.

İlişki bağlantıları Şekil 23'te belirtilmiştir.



Şekil 24: Class Diyagram

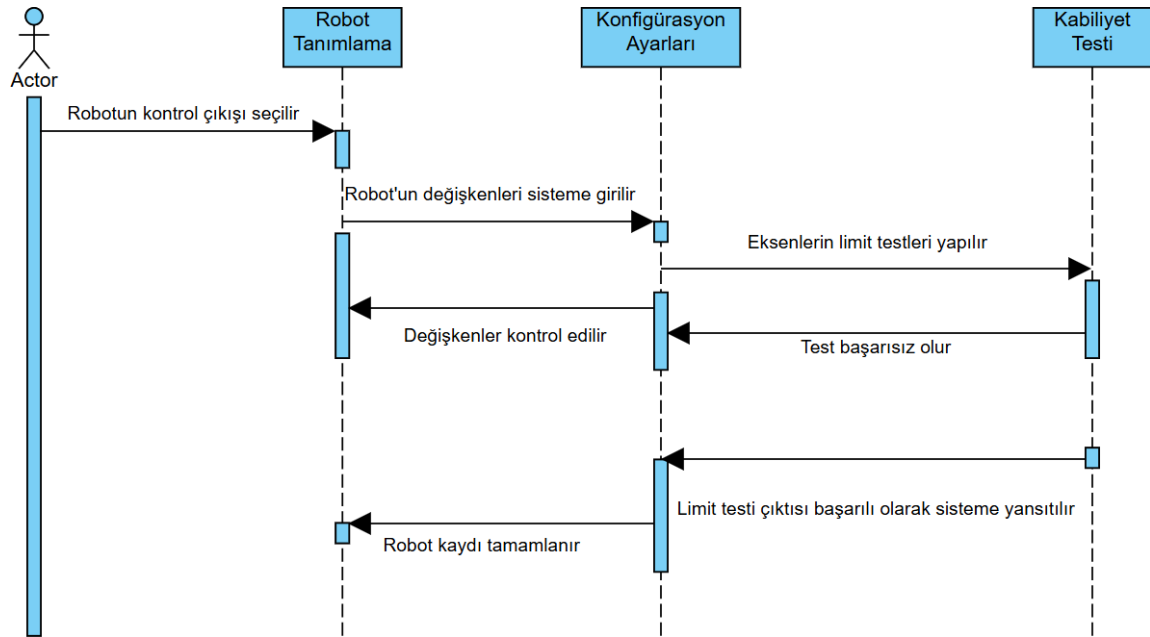
Masaüstü uygulaması kullanıcı arayüzünde bulunan sınıflar Şekil 24’te verilmiştir.

Bu bölümde yer alan önceki sınıf diyagramlarına göre farkı LogManager ile arayüz üzerinden hata kayıtlarının tutulduğu sınıfın da sunulup yazılımın tüm sınıflarının dahil edilmesidir.

15 Dinamik Model

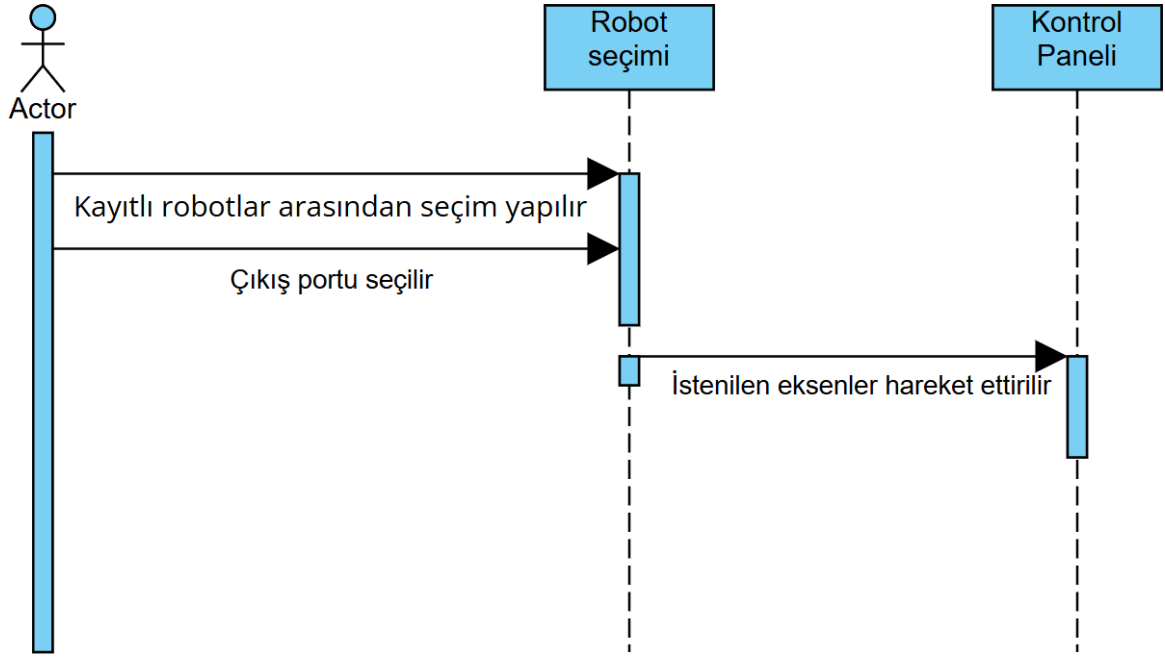
Sequence diyagramları yazılımın iç işleyişini anlamak ve her bir bileşenin birbirleriyle olan etkileşimini zaman sırasına göre izlemek için büyük bir öneme sahiptir. 6RK Control yazılımında robot kolun kontrolü, kullanıcı etkileşimleri ve donanım bileşenleri arasındaki veri akışı ve komut sırası doğru bir şekilde modellemeyi gerektirir. Sequence diyagramları, yazılımda gerçekleştirilen fonksiyon çağrıları, veri iletimi ve geri bildirim süreçlerini ayrıntılı olarak göstererek, her bir modülün rolünü net bir şekilde ortaya koyar. Özellikle konfigürasyon ayarları, giriş-çıkış işlemleri ve web kontrol gibi farklı sekmelerin etkileşimini optimize etmek ve potansiyel hataları tespit etmek için sequence diyagramları kritik bir araçtır.

Bu nedenle, yazılım geliştirme sürecinde sequence diyagramları, yazılımın verimliliğini artırmak, hatasız bir çalışma sağlamak ve sistemin güvenilirliğini test etmek için önemli bir kılavuz sayılabilir.



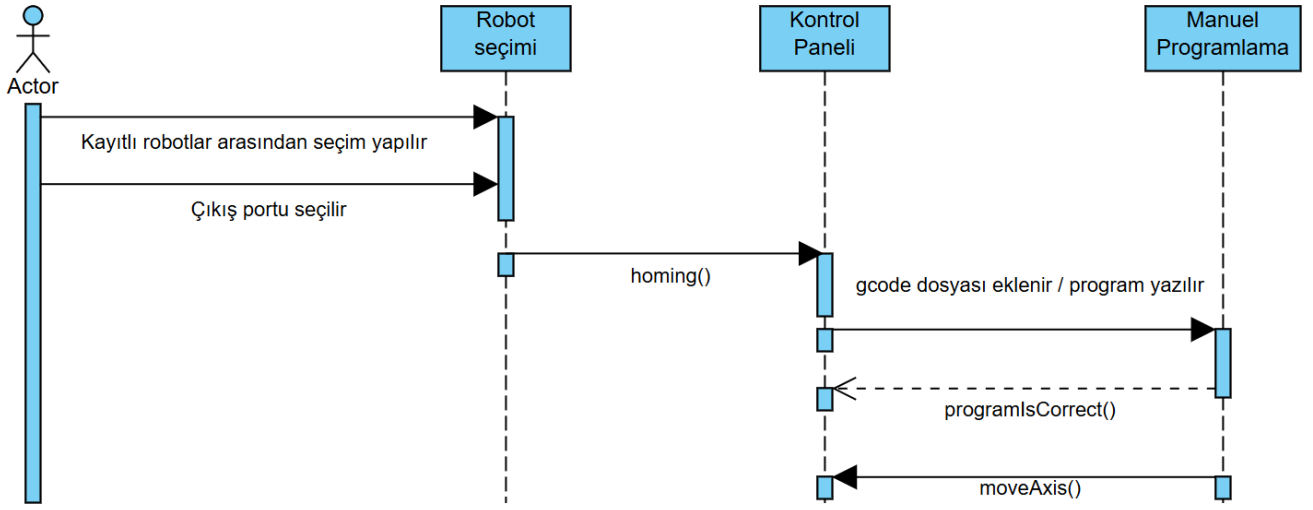
Şekil 25: Sequence Diyagram

Yukarıdaki şekilde kullanıcının arayüz üzerinden programa ilk eriştiğinde yapması gereken “sisteme robot girişi” diyebileceğimiz tanımlama işleminin görsel çözümlemesi yer almaktadır. Çoğu yazılım projesindeki hesap oluşturma gereksinimi gibi somutlaştırılabilir.



Şekil 26: Sequence Diyagram

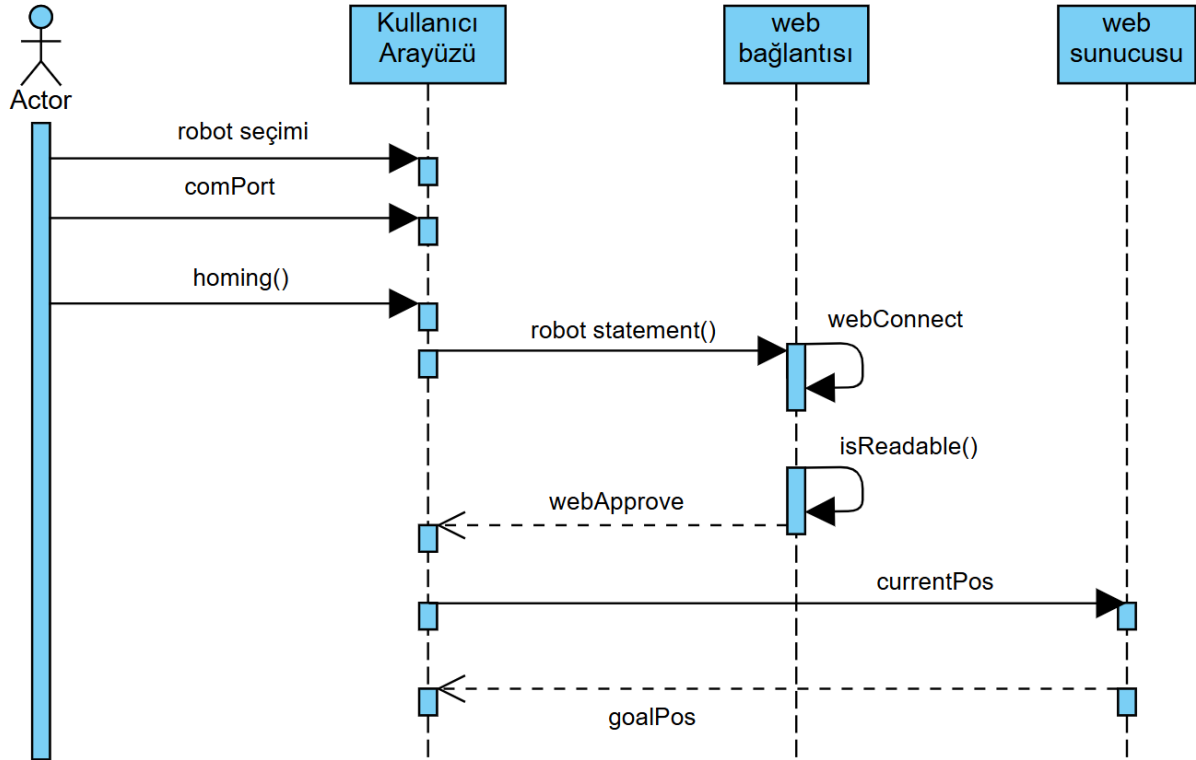
Kullanıcı önceden tanımlanan robotlar arasından isteğine göre tercih ettiği robotu sisteme bağlar. Doğru iletişim portu ile robota erişim sağlar ve tüm bu işlemleri ve robot kontrolünü yine arayüz üzerinden gerçekleştirir.



Şekil 27: Sequence Diyagram

Kullanıcı robotu arayüz üzerinden değil önceden hazırlanmış veya kendi yazdığı program ile kontrol etmek istediği senaryoda neler yapılması gerektiği Şekil 27’de görsel çözümleme ile sunulmuştur.

Arayüz üzerinden kayıtlı robotun doğru iletişim kanalı ile etkinleştirilmesi ile sıfırlama işlemi için homing metodu çalıştırılır. Kontrol paneli üzerinden yapılan bu işlemin ardından dosya yolu veya elle giriş ile robot programlanır ve kontrolü sağlanır.

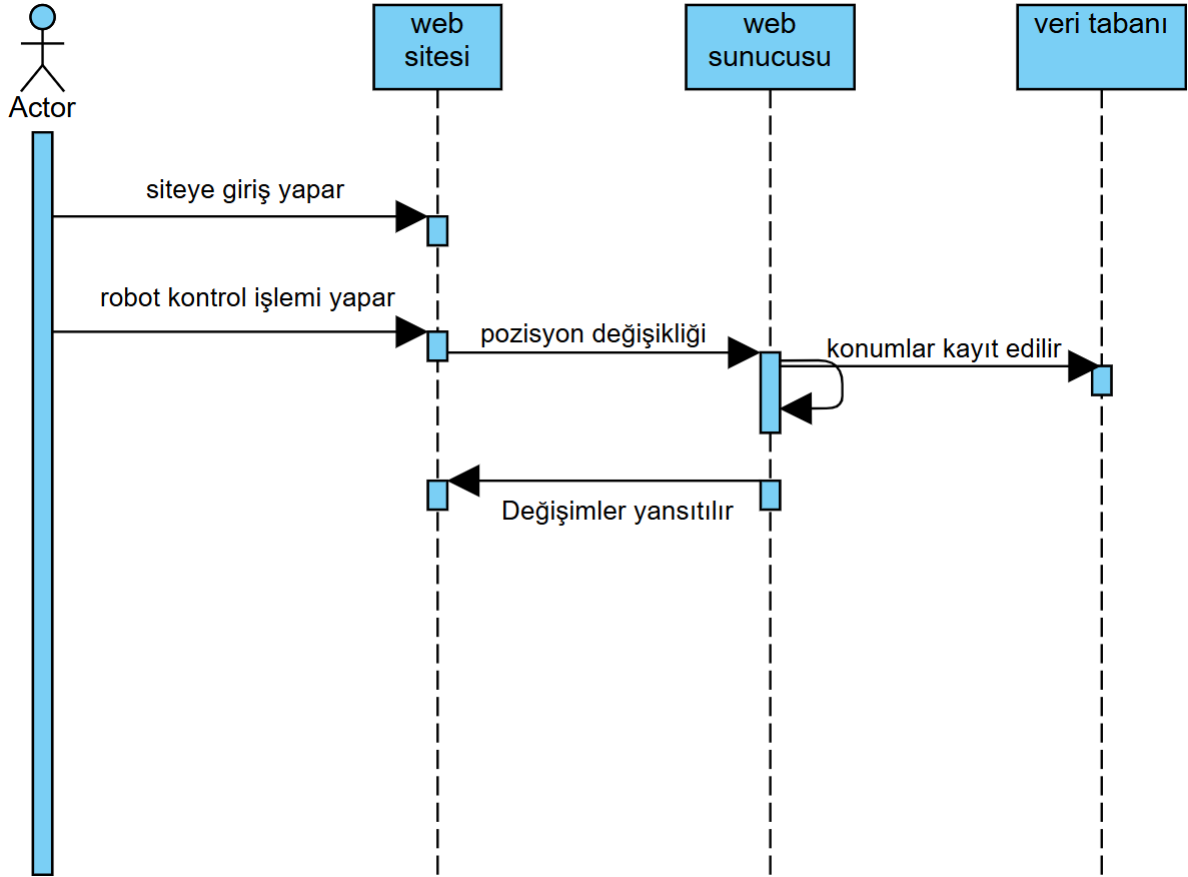


Şekil 28: Sequence Diyagram

Yukarıdaki şekilde kullanıcının sistemi web üzerinden diğer kullanıcıların erişimine uygun hale getirdiği senaryo çözümlenmiştir.

Kullanıcıların farklı cihazlardan web tabanlı uygulama ile robota erişimi için 6RK Control yazılımının kullanıcı arayüzü üzerinden bu erişimin talep edilmesi gerekmektedir.

Kullanıcı arayüz üzerinden robot tanımı, seçimi ve sıfırlama işlemlerini yapmak zorundadır. Ardından arayüzde bulunan web bağlantı sayfasından talebi gerçekleştirdiğinde sistem web sunucusu üzerinden haberleşmeyi sağlar ve ardından web tabanlı uygulama üzerinden kontrol mümkün hale gelir.



Şekil 29: Sequence Diyagram

Şekil 28’de robotun web üzerinden nasıl kontrol komutları aldığı gösterilmişti. Kullanıcı web sitesi üzerinden robot kontrolü sağlayacağı zaman sistem çok daha basittir. Çünkü verilere erişim ve kontroller hala masaüstü uygulaması tarafından takip edilmektedir.

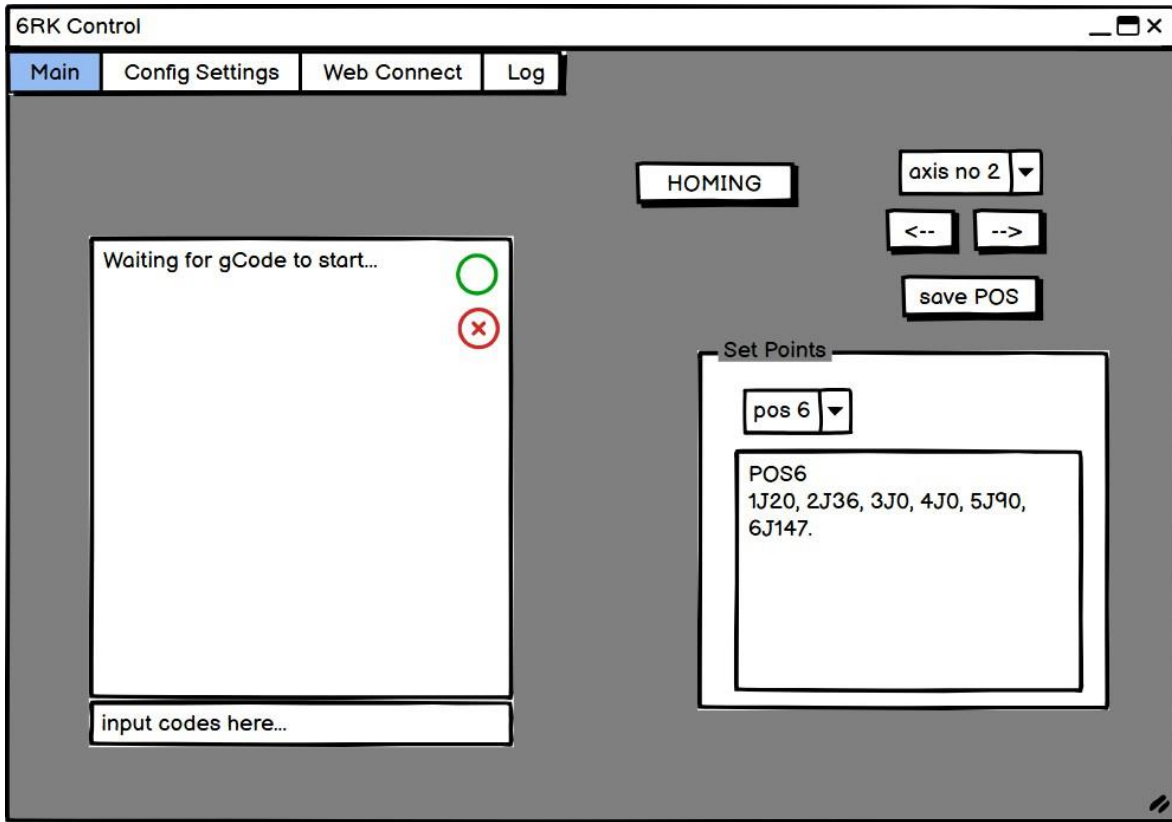
Web sitesine giriş yapan kullanıcı eğer 6RK Control yazılımı online moda geçmiş ise herhangi bir cihaz ve herhangi bir ağ üzerinden bağlandığı site üzerinden robot kontrol butonlarına erişebilmektedir.

Farklı ağlar ve cihazlardan erişimden kaynaklı bir problem yaşanmaması için web sunucusu ve veri tabanı kullanımı gereksinimdir.

16 Kullanıcı Arayüzü

Mockup toollar ile kullanıcı arayüzü (UI) planı yapmak, yazılımın kullanıcı arayüzünü tasarlarken önemli bir aşamadır. Bu araçlar, tasarımcıların ve geliştiricilerin uygulamanın görünümünü ve kullanıcı etkileşimini önceden görselleştirmelerine olanak tanıyarak olası tasarım hatalarını erkenden tespit etmek için alternatif bir yol sağlar.

Bu projede de bu amaçla “Balsamic UI Mockup Tool” kullanılmıştır.



Şekil 30: Ana ekran UI taslağı

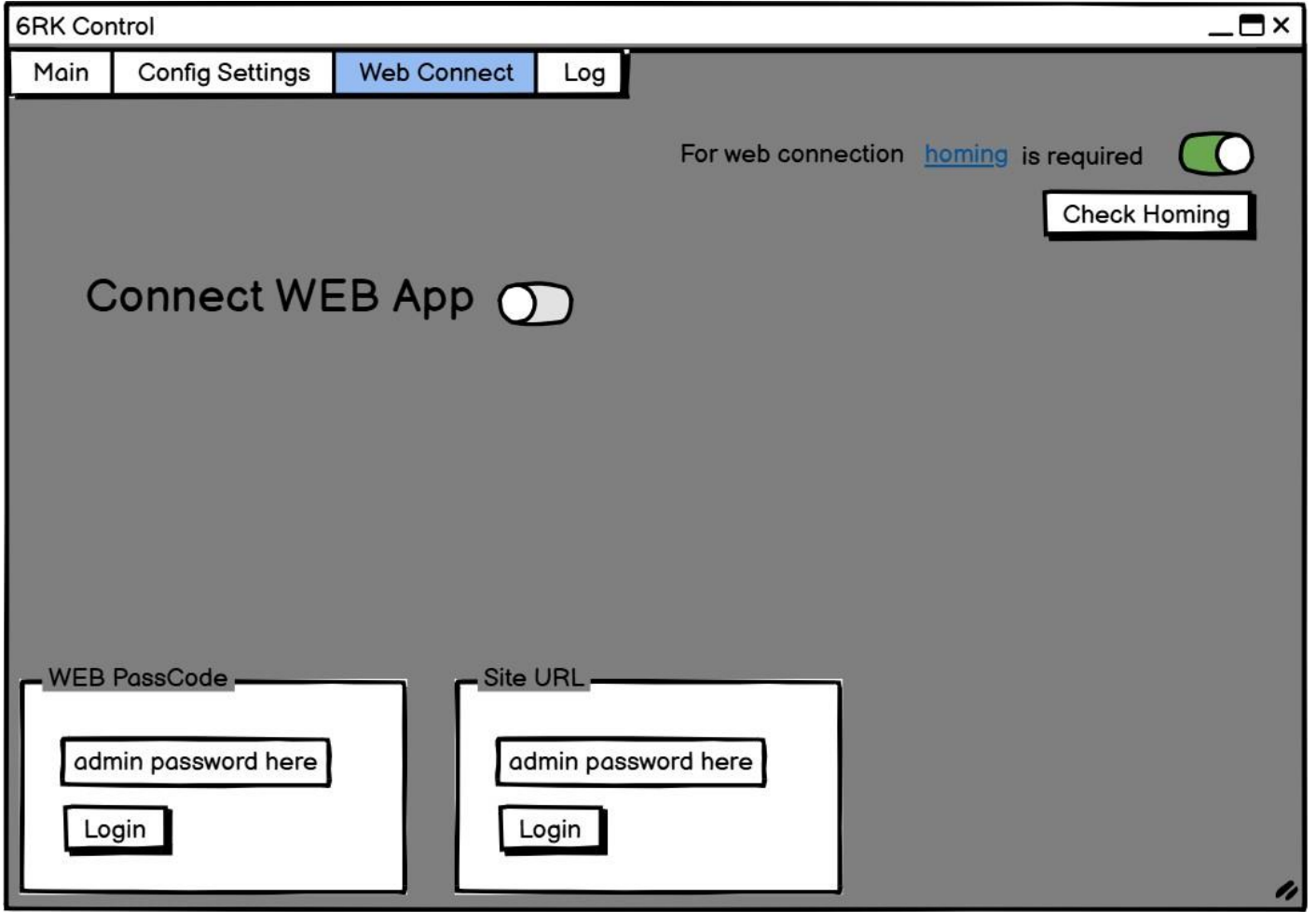
Şekil 30’da bulunan uygulamanın ana sayfasının taslağı üstteki menü çubuğu ile sekmeler arası geçişi mümkün kılmaktadır. Ekranın sol kısmında manuel programlama için dosya ekleme veya programlama alanı ve yeşil, kırmızı butonlar ile programı çalıştırma seçenekleri bulunmaktadır. Ekranın sağ bölümünde öncelikli olarak “homing” butonu bulunmaktadır. Sağ üst kısımdaki seçeneklerden eksenleri tek tek seçebilir, kontrol edebilir ve daha sonrasında planlı bir rota için pozisyonların kaydedilmesi sağlanabilir.

Joint Limits	-	+
J-1		
J-2		
J-3		
J-4		
J-5		
J-6		

Şekil 31: Konfigürasyon ayarları UI Tasarımı

Şekil 31’de Config Settings sekmesinde karşımıza gelen alanlardan robotun bağlı olduğu portu seçebilir yeni robot ekleyebiliriz. Seçilen veya eklenen robotun eksen limitlerini belirledikten ve aşağıdaki “DH Parameters” sekmesine bahsettiğimiz DH parametrelerini girdikten sonra tıklanabilir hale gelen kalibrasyon butonundan robotu kalibre edebiliriz. Sağ alttaki kaydet butonu ile de robotun ayarlaması tamamlanabilir.

Buradaki DH Parametrelerinin detaylıca tanımları, raporun III numaralı bölümünde yapılmıştır.

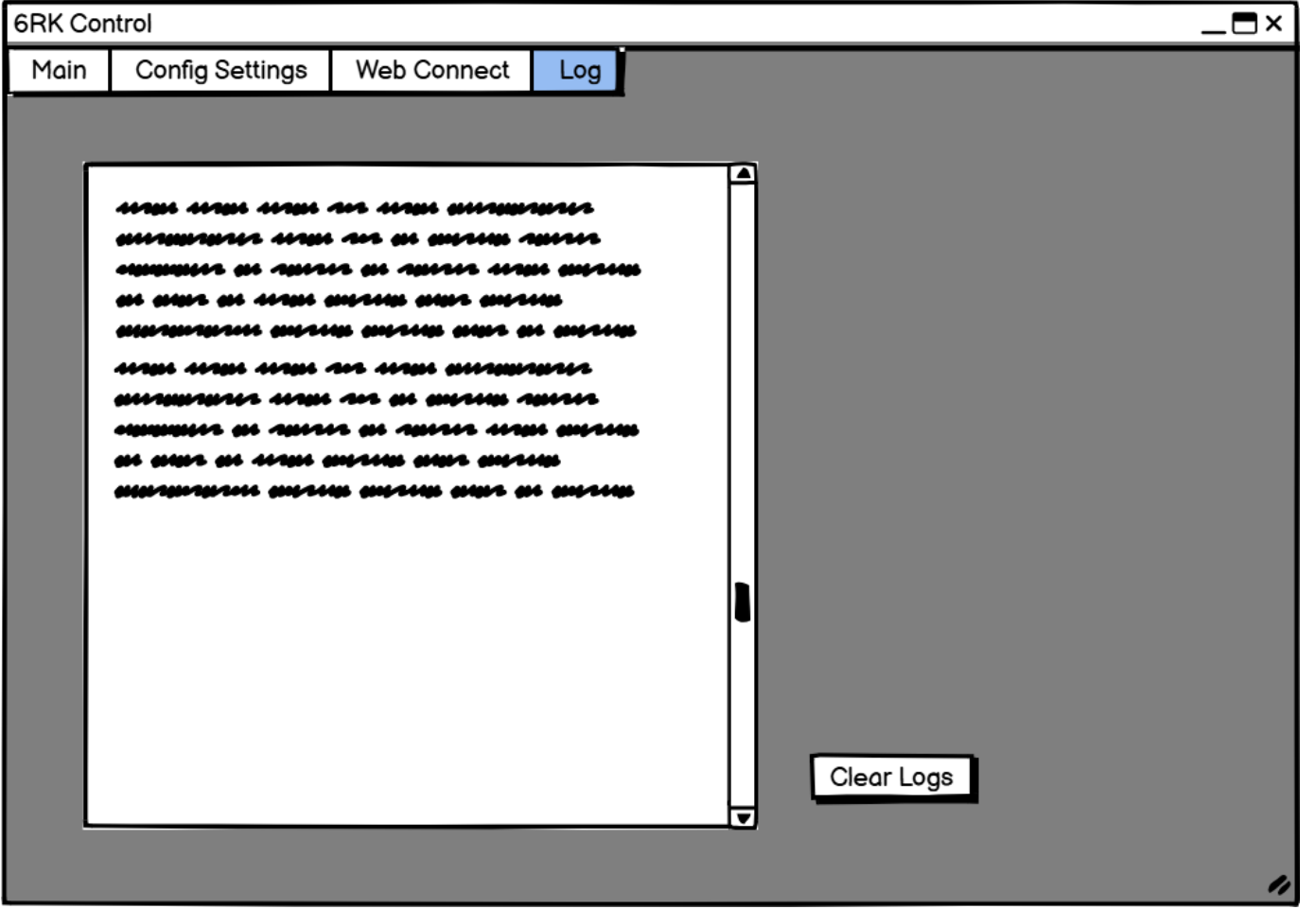


Şekil 32: Web Bağlantı sekmesi UI Tasarımı

Robotun arayüz üzerinden kontrolü dışında kullanıcıların uzaktan kontrol sağlaması için web erişiminin açık olması gerekmektedir.

Arayüz üzerinde Şekil 32’deki Web Connect sayfasından da anlaşılabilceği üzere öncelikli olarak robotun sisteme eklenmesi vb. işlemlerden sonra “homing” sağlanmalıdır. Bağlanılacak site uzantısı ve site ile iletişimin doğrulanması için gerekli “passcode” ayarları yalnızca yönetici tarafından değiştirilebileceği için ekranın alt kısmında iki farklı giriş noktası bulunmaktadır.

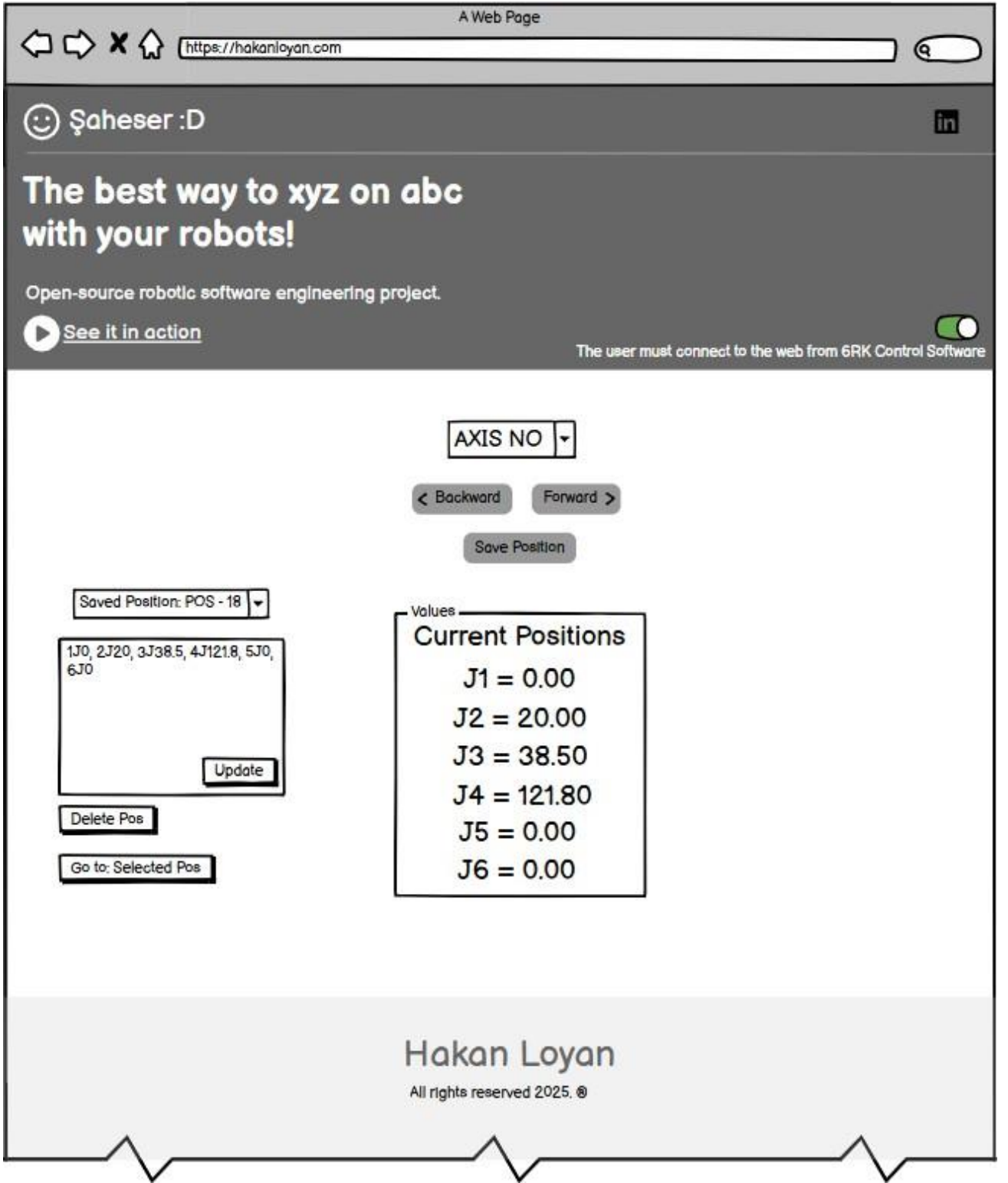
Web erişimini açmak için sağ üstteki check homing butonuna basıldığında üstündeki switch yeşil olmalıdır. Ardından ekranın sol tarafındaki switch kullanıcı tarafından tıklanılarak robot uzaktan erişime açılabilir.



Şekil 33: Log kayıtlarının sahip olduğu UI Tasarımı

Karmaşık yapısı ve hataların incelenebilmesi adına masaüstü uygulamasında Log sekmesinde uygulamada yaşanan, robotun fiziksel olarak karşılaştığı veya web tabanlı uygulama ile olan iletişimde yaşanabilecek tüm sorunlar burada kayıt altına alınmaktadır.

Log kayıtları uygulama dışında önceden belirlenen bir dosyada tutulmaktadır. Ekrandaki Clear Logs butonu ile geçmiş log kayıtları silinebilir.



Şekil 34: Web sitesinin UI Tasarımı

Robotun uzaktan erişimine imkan tanıyan web sitesinin kullanıcı dostu tasarımı. Robot için bağlantı sağlanıp sağlanamadığı, gerekli kontrol butonları, robotun anlık konum verileri ve kayıt altında olan pozisyonlara erişim, güncelleme ve silme gibi işlemler kolaylıkla yapılabilmektedir.

V Test Planları

Test dokümantasyonu, yazılım test bünyesinde büyük önem taşır çünkü testlerin doğru ve etkili bir şekilde gerçekleştirilmesini sağlar. Yazılımın başarısını doğrudan etkileyen test süreçlerinin düzgün bir şekilde yönetilmesi, test ekibinin dikkat etmesi gereken unsurların net bir şekilde tanımlanması ve kaynağın doğru şekilde izlenebilmesi için test dokümantasyonuna ihtiyaç vardır. IEEE 829 Test Plan Dokümantasyonu, yazılım test sürecinin her aşamasını ayrıntılı olarak açıklar ve testin kapsamını, kalite standartlarını, zaman ölçeğini ve alınan riskleri içeren bir yol haritası sunar. Bu yüzden test hazırlıkları, yazılım projelerinin başarısı için en kritik adımlardan biri olarak kabul edilir.

17 Test Edilebilecek ve Edilemeyecek Özellikler

Test Edilen Alt Sistemler	Test Edilen Senaryolar
Sisteme Robot Eklenmesi	-Sisteme yeni bir robotun eklenme testi -Sisteme yeni eklenen robotun ayarlarının değiştirilme testi -Eksen Limitleri Güncellenmesi Testi
Tanımlı Robot Kontrolü	-Sistemden tanımlı robotların seçilme testi - Çıkış portu seçilerek kalibrasyon testi -Arayüz üzerinden eksenlere hareket testi
Manuel Program Kontrolü	-Uygulamadan manuel program alanının tıklama testi -Dosya açma testi -Programı çalıştırma testi
Web Bağlantısı	- Web bağlantısı için istenen gerekliliklerin sağlanması -Uygulama kontrolü ile web bağlantısı testi
Web ile Robot Kontrol	-Web Bağlantısı sağlanması -Farklı ağ ve cihazdan robot hareket testi

Tablo 7: Test Edilen Alt Sistemler

Tablo 7’de 6RK Control uygulamasının test edilebilecek alt sistemleri listelenmiştir. Bu testler doğrultusunda sistemin robot manipölatör kontrolü üzerinde temel olarak tüm fonksiyonlarını gerçekleştirmesi amaçlanmaktadır.

Test Edilemeyen Alt Sistemler	Açıklama
Sistem Geri Bildirimi	-Kullanıcılardan gelen geri bildirimleri toplayacak bir tasarım olmadığı için testi mümkün değil.
Web Tabanlı Hata Giderimi	Proje kapsamında web tabanlı bir log kaydı tutulmadığı için bu test yapılamamaktadır.
Sistem Güvenliği	Sitenin güvenliği, sunucu güvenliği ve veri tabanı altyapısı gibi sistem seviyesindeki testler buraya dahil edilmeyecektir.
Farklı Donanım Entegrasyonu	Kullanıcı sayısı ve sürekli geliştirme ihtiyacından dolayı bu aşamada 6RK robot manipölatöre özelleşmiş olarak kabul edilen yazılımın uyumluluk testleri kapsam dışında kalmaktadır.

Tablo 8: Test Edilemeyen Alt Sistemler

Elle Gerçekleştirilen Test	Açıklama
Sistemin Fiziksel Doğruluk Oranı	-Gerçek dünya üzerinde inşa edilen robot üzerinden referans alınması gerektiği için doğruluk oranı test edilmeyecek.
Log Kayıtları	Sistemdeki yazılımsal hatalar dışında donanımla olan iletişim, gecikme vb. arızalar fiziksel yapının gereklilikleri sebebiyle kapsam dışı kalmaktadır.

Tablo 9: Elle Gerçekleştirilen Testler Tablosu

Tablo 8 ve Tablo 9’da 6RK Control sistemine ait test edilmeyecek alt sistemler ve özellikler listelenmiştir. Kitle gereksinimi ile kullanıcı geri bildirimleri, fiziksel donanımlar sebebi ile uyumluluk özellikleri gibi nitelikler test kapsamı dışında tutulmuştur.

14 Test Cases

Test Case ID: TCH-01

Test Başlığı: Sisteme Robot Eklenmesi

Test Amaçları: Sisteme yeni bir robotun başarılı birr şekilde eklenmesi ve robotun ayarlanabilmesi.

Test Prosedürleri ve Sonuçları:

- Kullanıcı, sistemde "Yeni Robot Ekle" seçeneğine tıklar.
- Yeni robot için gerekli bilgiler girilir.
- Robot veri tabanına kaydedilir ve sistemde görüntülenir.
- Sistemde eklenen robotun doğru şekilde görüldüğü kontrol edilir.

Test Verisi:

- Robot Adı: "6RK Manipülatör"
- Model: "6DOF"
- Özellikler: "Yüksek hassasiyet, 6 eksen hareket kabiliyeti"

Beklenen Sonuç: Kullanıcı, robot bilgilerini doğru şekilde girdikten sonra robot başarılı bir şekilde eklenir ve sisteme kaydedilir.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü: Eksik veri, bağlantı hatası, ver, tabanı hatası

Hata Önemi: Yüksek

Test Case ID: TCH-02

Test Başlığı: Sisteme Yeni Eklenen Robotun Ayarlarının Değiştirilmesi

Test Amaçları: Kayıtlı robotun ayarlarının değiştirilebilmesi.

Test Türü: Fonksiyonel Test

Test Prosedürleri ve Sonuçları:

- Kullanıcı, eklenen robotun ayarlarına gitmek için robotu seçer.
- Robotun ayarları değiştirilir.
- Değişiklikler kaydedilir ve robotun yeni ayarları sisteme işlenir.
- Ayarların başarıyla değiştirildiği kontrol edilir.

Test Verisi:

- Eksen sayısı
- Eksen Limitleri

Beklenen Sonuç:

- Robot ayarları doğru şekilde değiştirildi ve veri tabanına kaydedildi.
- Kullanıcı, değiştirdiği ayarları kontrol ettiğinde doğru sonuçlar alır.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü: Yanlış giriş verisi, sistem hatası (kayıt edilmemesi).

Hata Önemi: Orta

Test Case ID: TCH-03

Test Başlığı: Sisteme Tanımlanmış Robotun Seçilme Testi

Test Amaçları: Sistemdeki robotların doğru şekilde seçilebilmesi.

Test Türü: Fonksiyonel Test

Test Prosedürleri ve Sonuçları:

- Kullanıcı, sistemdeki tanımlı robotlar listesine gider.
- Listedeki bir robot seçilir.
- Seçilen robot, kullanıcı arayüzünde doğru şekilde görüntülenir ve kontrol edilebilir hale gelir.
- Kullanıcı, seçilen robotu kontrol edebilir.

Test Verisi:

- Robot adı

Beklenen Sonuç:

- Robot sistemden doğru şekilde seçilir ve görüntülenir.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü: Yanlış seçim, liste hatası (robotun kayıt edilmemesi/görünmemesi).

Hata Önemi: Yüksek

Test Case ID: TCH-04

Test Başlığı: Arayüzde Eksen Hareket Testi

Test Amaçları: Robotun eksenlerinin arayüz üzerinden kontrol edilmesi.

Test Türü: Birim Test

Test Prosedürleri ve Sonuçları:

- Kullanıcı, bir eksen seçer
- Eksen için hareket komutu gönderir.
- Eksen, doğru yönde hareket eder.
- Tüm eksenler için bu işlemler yapılır.

Test Verisi:

- Eksen hareket ve yön komutu.

Beklenen Sonuç:

- Eksenler istenilen yönde ve sürede hareket eder.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü: Komut hatası.

Hata Önemi: Yüksek

Test Case ID: TCH-05

Test Başlığı: Manuel Program Çalıştırma Testi

Test Amaçları: Program ile robot kontrolünün çalıştığını test etmek.

Test Türü: Fonksiyonel Test

Test Prosedürleri ve Sonuçları:

- Kullanıcı, doğruluğundan emin olunan bir dosyayı sistemde açar.
- Kullanıcı robot ayarlarını yaptıktan sonra “Çalıştır” butonuna basar.
- Robot, program doğrultusunda planlanan rotayı tamamlar.

Test Verisi:

- Gcode komut dosyası.

Beklenen Sonuç:

- Eksenler istenilen yönde ve sürede hareket eder.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü: Bozuk dosya, yanlış hareketler.

Hata Önemi: Yüksek

Test Case ID: TCH-06

Test Başlığı: Web Uygulaması İle Kontrol Testi

Test Amaçları: Uzaktan kontrol ile robot erişimini test etmek.

Test Türü: Entegrasyon Testi

Test Prosedürleri ve Sonuçları:

- Kullanıcı, arayüz üzerinden robot işlemlerini tamamlar ve homing sağlanır.
- Kullanıcı uygulama üzerinden web connect butonuna basar.
- Arayüz üzerinden bağlantının başarılı olduğuna dair sistem mesajı gelir.
- Kullanıcı farklı bir cihaz ve farklı bir ağ kullanarak web sitesine erişir.
- Robotun, uzaktan kontrole açık olduğunu site üzerinden kontrol eder.
- Kullanıcı, site üzerindeki imkanlarla robotu uzaktan kontrol eder.

Test Verisi:

- Web sunucusu
- Veri tabanı
- Kullanıcı istekleri

Beklenen Sonuç:

- Eksenler istenilen yönde ve sürede uzaktan kontrol ile hareket eder.

Gerçek Sonuç: Beklenen sonuç ile aynı.

Hata Türü:

- Web bağlantı hatası
- Ağ problemi
- Sunucu iletişim problemi
- İstikrarsız eksen hareketleri

Hata Önemi: Düşük

SÖZLÜK

6DOF: 6 Degrees of Freedom, 6 serbestlik derecesi.

6RK: 6DOF Robot Mnaipülatör projesinin özel adı.

Joint: Eklem.

Axis: Eksen.

Firestore Veritabanı: Goofle'ın geliştirmiş olduğu bulut tabanlı veri tabanı pladormu.

Uyumluluk Testi: Yazılımın farklı donanımlarda stabil çalıştığını denemek için yapılan test.

Fonksiyonel Testler: Uygulamanın temel işlevselliğini değerlendiren testler.

Class Diagram: Bir yazılım sistemindeki sınıfları, sınıflar arasındaki ilişkileri ve sınıfların özelliklerini (attributes) gösteren bir UML diyagram türüdür.

Sequence Diagram: Bir yazılım sistemindeki nesneler arasındaki zaman sıralı etkileşimleri gösteren bir UML diyagram türüdür.

Aggregation: Bir nesnenin, başka bir nesneye parçalarından biri veya birkaçı üzerinden referans verme durumudur. Bu, "bir parça" veya "birleşik" ilişkisini ifade eder.

Composition: Bir nesnenin, başka bir nesneye bağımlılığını ve içsel bir ilişki kurma durumudur. Bu, "bir parça" ilişkisini ifade eder ve bir nesne diğer nesne olmadan var olamaz.

Association: Bir nesnenin, başka bir nesneye bağımlılığını ve içsel bir ilişki kurma durumudur. Bu, "bir parça" ilişkisini ifade eder ve bir nesne diğer nesne olmadan var olamaz.

Cost-Effectiveness: Bir projenin veya bir ürünün maliyetinin, getirisi ve performansı ile dengeli bir şekilde orantılı olma yeteneğidir.

Robustness: Bir sistem veya yazılımın, hatalara, uygunsuz girdilere veya değişen çalışma koşullarına karşı dirençli olma yeteneğini ifade eder. Sistem, beklenmeyen durumlarla başa çıkabilir.

G-code: Bir sistem veya yazılımın, hatalara, uygunsuz girdilere veya değişen çalışma koşullarına karşı dirençli olma yeteneğini ifade eder. Sistem, beklenmeyen durumlarla başa çıkabilir.

KAYNAK

- [1] Wikipedia The Free Encyclopedia, “Euler Angles” (), Wikipedia The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles [Accessed: Dec. 19, 2021].
- [2] A. Khatamian, “Solving Kinematics Problems of a 6-DOF Robot Manipulator”, Int’l Conf. Scientific Computing, CSC, 2015, pp. 228-233
- [3] Ondočko, Š., Stejskal, T., Svetlík, J., Hrivniak, L., Šašala, M., & Žilinský, A. (2020). Position Forward Kinematics of 6-DOF Robotic Arm. Acta Mechanica Slovaca, 24(2), 30-36. doi: 10.21496/ams.2020.018