

### 3 Serbestlik Dereceli Robot Kolunun Issa Algoritması ile Pid Kontrolü

Ferdi AVCI\*<sup>1</sup>, Mehmet ÇAVAŞ<sup>2</sup>

\*<sup>1</sup> Muş Alparslan Üniversitesi Teknik Bilimler MYO Makine ve Metal Teknolojileri, MUŞ  
<sup>2</sup> Fırat Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği, ELAZIĞ

(Alınış / Received: 22.03.2023, Kabul / Accepted: 20.11.2023, Online Yayınlanma / Published Online: 30.12.2023)

#### Anahtar Kelimeler

Kontrol,  
PID,  
Robotik,  
Simülasyon

**Öz:** Günümüzde, robotik sistem uygulamalarının hatasız çalışabilmesi için yapılan robot ya da robotik sistemlerin tasarımları öncelikle bilgisayar ortamında simüle edilerek test edilmektedir. Bu çalışmada 3 serbestlik derecesine sahip RRR (Dönel-Dönel-Dönel) robot kol tasarımı yapılarak, PID(Oransal-İntegral-Türevsel) kontrolör ile kontrolü sağlanmıştır. PID parametreleri ISSA (Geliştirilmiş Salp Swam Algoritması) ve KKO (Karınca Kolonisi Optimizasyonu) algoritmasıyla optimum düzeyde belirlenmiş, PID kontrolörün performanslarının karşılaştırılması amaçlanmıştır. Bu kapsamda, robotik sistemin mekanik tasarımında SolidWorks programı ve kontrol için Matlab programı kullanılmıştır. Ayrıca bu çalışmada PID parametrelerinin tespiti ve uygulaması birim basamak ve sinüzoidal giriş için incelenmiştir. Robot kolun konum koordinatlarının belirlenmesi için ileri kinematik hesaplamalar yapılmış ve sonuç olarak robot kol başarılı bir şekilde tasarlanarak Matlab ortamında simüle edilmiştir. ISSA ile belirlenen katsayılar KKO algoritması ile belirlenen katsayılara göre, PID kontrolörün referans yörüngesi daha yakından takip ettiği görülmüştür.

### Pid Control of 3 dof Robot Arm with Issa Algorithm

#### Keywords

Control,  
PID,  
Robotik,  
Simulation

**Abstract:** Today, the designs of robots or robotic systems made for the error-free operation of robotic system applications can be tested by simulating in the computer environment. In this study, the RRR (Revolute-Revolute-Revolute) robot arm with 3 degrees of freedom was designed and controlled using a PID (Proportional-Integral-Derivative) controller. PID parameters were determined at the optimum level using ISSA (Improved Salp Swam Algorithm) and KKO (Ant Colony Optimization) algorithms, and it was aimed to compare the performances of the PID controller. In this context, the SolidWorks program was used in the mechanical design of the robotic system and the Matlab program was used for control. In addition, in this study, the determination and application of PID parameters are examined for unit step and sinusoidal input. To determine the position coordinates of the robot arm, advanced kinematic calculations were made and as a result, the robot arm was successfully designed and simulated in the Matlab environment. Coefficients determined by ISSA It has been observed that the PID controller follows the reference trajectory more closely than the coefficients determined by the KKO algorithm.

\*İlgili Yazar, email: f.avci@alparslan.edu.tr

#### 1. Giriş

Modern endüstri tesislerinde malzemelerin üretimi, taşınması, montajı, tıbbi tedaviler, uzay araştırmaları vb. birçok alanda farklı işlemlerin yapılmasında farklı tip ve model robot veya robotik sistemler kullanılmaktadır [1]. Robot kolu, eklemlerle birbirine bağlı uzuvlardan oluşmaktadır. Bu yapılar eğilmeyen, mukavemeti yüksek ancak genellikle hafif malzemelerden imal edilmektedir. Kütleli harekete geçirmek için moment kazandırmak,

durdurmaya çalışırken gereksiz güç harcanmasını engellemek, frenlemede salınımı azaltmak, çabuk ivmelenme gibi parametreler hafif malzemelerde daha iyi sonuçlar vermektedir. Aynı zamanda daha az enerji tüketimi ile kontrol imkânı sağlamaktadır [2].

Literatürde, robot kolların kontrolünün gerçekleştirildiği birçok çalışma bulunmaktadır [3-12]. Matlab programı kullanılarak 4 eksenli robot kolun kinematik hesapları çıkarılarak Matlab&Simulink ortamında ve gerçek modelde test edilen çalışmaların yapıldığı görülmektedir [3]. Üç serbestlik dereceli bir robotun uç işlevcisine PID kontrol ve bulanık mantık kontrolü kullanılarak değişken kuvvet uygulama suretiyle dayanıklılık testi yapılmış çalışmalar da mevcuttur [4]. Diğer bir çalışmada, 3-SD robotun tasarımı SolidWorks'te, simülasyonu Robotic Toolbox kullanarak Matlab&Simulink ortamında gerçekleştirilmiş, robotun kontrolü için gerekli olan kinematik ve dinamik denklemler elde edilerek simülasyon ve gerçek zamanlı çalışmaları yapılmıştır. Gerçek zamanlı uygulamalar için iç döngüde yer çekimi dengelemeli PID kontrolü, dış döngüde empedans kontrol algoritmasından yararlanılmıştır. Aynı zamanda yer çekimi dengelemeli PID kontrolün katsayıları, her bir uzuv için Matlab programının System Identification eklentisi kullanılmıştır [5]. Yüksek dereceli sistemlerin kontrolünde kullanılan Oransal-İntegral-Türev-İvme (Proportional-Integral-Derivative-Acceleration-PIDA) denetçi parametrelerinin Parçacık Sürü Optimizasyon (PSO) algoritması yardımıyla belirlenen ve PIDA denetleyicinin başarımını, literatürdeki Gradyant arama ve Genetik algoritma (GA) optimizasyon yöntemleri ile karşılaştırılması da yapılmıştır [6]. 4 eksenli ve SCARA tipinde bir robot kol düzeneğinin görüntü işleme tabanlı yörünge kontrolüyle çapaklı parçaların temizlenmesi işlemi için yapılan çalışmada, öncelikle robot kola ait kinematik denklemlerin analitik çözümleri elde edilmiş, daha sonra bir kamera ile alınan görüntüler işlenerek, işlenecek parçanın sınırlarını belirlenmiştir. Piksel-koordinat dönüşümleri yapılarak bulunan değerler bilgisayar ortamında kontrol edilen motorlara açılış setleri şeklinde gönderilip, parça geometrisi olarak belirlenen yörünge izlenmesi sağlanmıştır [7].

Robot kol kontrollerinin yapıldığı çalışmalarda bir diğer yöntem ise simülasyon programlarının kullanılmasıdır. Simülasyon programlarında bulunan robot kolların konfigürasyonu tamamlandıktan sonra kontrol için gerekli olan kinematik ve dinamik denklemler elde edilerek simülasyon ve gerçek zamanlı çalışmalar yapılmıştır. Robot kolun yüklü ve yüksüz durumda izleyeceği yörünge planlanarak konumlar belirlenmiş ve belirlenen konumlara göre yörünge takibi yapılmıştır. Robot kolun kontrolünün sağlanması için simülasyon programı Matlab programı ile senkronize edilerek simülasyonu gerçekleştirilmiştir [8]. Ayrıca farklı optimizasyon algoritmaları kullanılarak PID parametreleri belirlenen, robot kolun kontrolü üzerine çalışmalar da yapılmıştır. Bir robot kolun konum kontrolü için PID kontrol parametreleri Genetik Algoritma (GA) ve Arı Algoritması (AA) kullanılarak belirlenen, optimizasyonu ve benzetim çalışmaları Matlab&Simulink ortamında yapılmış robot kolun, dinamik ve kinematik denklemleri elde edilerek matematiksel modelinin oluşturulduğu görülmektedir [9]. Diğer bir çalışmada, iki serbestlik derecesine sahip manipülatör için doğrusal olmayan kontrolörün tasarlandığı, Duruma Bağlı Riccati Denklemi (State Dependent Riccati Equation - SDRE) algoritmasının kullanıldığı çalışmaların da yapıldığı görülmektedir [10]. Bir başka çalışmaya bakıldığında ise, senkron motorlar tarafından hareket ettirilen iki eksenli bir robotun pozisyon kontrolü hem kesir dereceli PID (KDPID) kontrolör hem de iki serbestlik dereceli PID (2DOFPID) kontrolör kullanılarak yapılan çalışma da bulunmaktadır. Bu çalışmanın tasarım ve analizinin yapılması için Matlab/Simulink kullanılmıştır [11]. Son olarak, Farklı birçok alanda kullanılan birçok alanda kullanılan ve doğadaki yaratıklardan esinlenerek yapılan robot robot çalışmaları da bulunmaktadır. Bunlardan biri de farklı mekanizmalara sahip ve farklı işlevleri gerçekleştirebilen yılan robot çalışmalarıdır. Bunlar aktif ve pasif olmak üzere İki kategoriye ayrılır. Yapılan çalışmada ise ikinci kategorideki temel esas alınarak bir yılan robotun tasarımı ve kinematiği gerçekleştirilmiştir [12].

Literatürdeki çalışmalara bakıldığında, robot kol tasarımı için genel olarak SolidWorks programı, kontrol ve simülasyon için ise Matlab programının kullanıldığı görülmektedir. Bazı çalışmalar da robot kolların özellikleri simülasyon programları aracılığı ile yapılmıştır. Kontrol kısmında ise PID kontrolörün parametreleri belirlenirken Genetik Algoritma, Arı Algoritması, Parçacık Sürü Optimizasyon Algoritması, Duruma Bağlı Riccati Denklemi Algoritması vb. farklı optimizasyon algoritmalarının kullanıldığı görülmektedir.

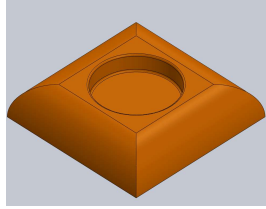
Son yıllarda, robot kollarının SolidWorks ile tasarımı ve Matlab programıyla kontrolünü sağlayan birçok çalışmanın yapıldığı, ancak bu çalışmalar da PID kontrolünde ISSA algoritmasının kullanılmadığı görülmektedir. Robot kolunun belirlenen hareketini daha hassas hale getirilmesini amaçlayan çalışmamızda, SolidWorks ile tasarlanmış ve Matlab programı yardımıyla ISSA algoritması kullanılarak tespit edilen PID katsayılarının kullanıldığı robot kol kontrolünün simülasyonu yapılmıştır. PID katsayılarının belirlenmesinde ISSA algoritması, ITSE performans indeksine göre kontrol edilerek çalıştırılmıştır. Ayrıca bu çalışmada tasarlanan robot kolun PID kontrolü, performans indekslerine göre de incelenmiştir. Farklı iki algoritma kullanarak incelenen sisteme, sinüs ve basamak girişleri uygulanarak elde edilen sonuçlar kıyaslanmıştır. Aynı zamanda daha önce robot kol kontrolü için parametrelerin belirlenmesi amacı ile hiç kullanılmayan ISSA algoritmasının robot kolu kontrolü üzerinde performansı belirlenmiştir.

## 2. Materyal ve Metot

Bu çalışmada 3 serbestlik dereceli robot kolun SolidWorks programıyla tasarımı yapılmış ve tasarlanan robot kol SimMechanics ile MATLAB ortamına alınarak modeli oluşturulmuştur. Modeli oluşturulan robot kolun her bir eklemine PID kontrolör eklenerek kontrol sağlanmıştır. PID kontrolörün optimum kontrol sağlaması için en uygun PID parametreleri ( $K_p$ ,  $K_i$ ,  $K_d$ ) tespitinde ISSA ve KKO algoritmaları kullanılmıştır.

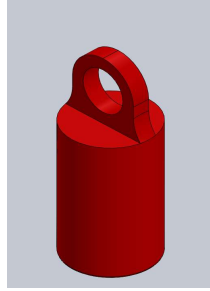
### 2.1. Robot kolunun mekanik tasarımı

3 serbestlik dereceli robot kolunun parçaları SolidWorks programı ile bilgisayar ortamında tasarlanmıştır. Tasarlanan robot kolun zeminini oluşturan platform Şekil 1'de verilmiştir.



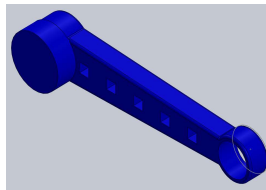
Şekil 1. Robot kolun platform çizimi

Şekil 1'de gösterilen Robot kolun üzerinde yer alacağı platform parçası 100x100x20 mm. boyutlarına sahip olup robot kolun birinci uzvunun yerleşebilmesi için 10 mm'lik silindirik oyuk açılmıştır. Şekil 2'de robot kolun X eksenindeki hareketinin sağlanacağı birinci uzvunun SolidWorks programındaki çizimi görülmektedir.



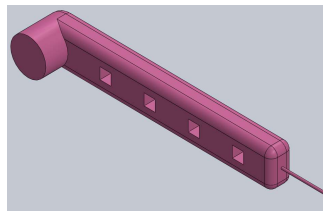
Şekil 2. Robot kolun 1. Uzvu

Şekil 2'de L uzunluğu yani birinci uzvu uzunluğu 100 mm olarak alınmış ve 100 mm'lik ölçü, zemin ile üstte bulunan ve ikinci uzvu merkezi ile çakışacak olan deliğin merkezine kadar olan boyutudur. Robot kolun Y eksenindeki hareketinin sağlanacağı ikinci uzvunun çizimi Şekil 3'de görülmektedir.



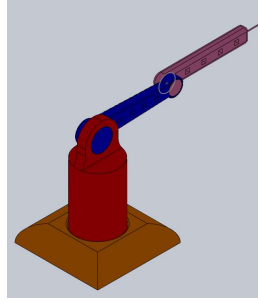
Şekil 3. Robot kolun 2. Uzvu

İkinci uzvunun L uzunluğu 100 mm olarak alınmış ve 100 mm'lik ölçü silindir ile ikinci uzvu merkezi ile çakışacak olan deliğin merkezine kadar olan boyutudur. Şekil 4'de robot kolun Y eksenindeki hareketinin sağlanacağı üçüncü uzvu ve gripper kısmı görülmektedir.



Şekil 4. Robot kolun 3. Uzvu

L boyu, yani üçüncü uzuv uzunluğu 120 mm olarak alınmış ve 120 mm'lik ölçü silindir merkezi ile end efektör olan üç kısım arasındaki boyutudur. Tasarımı yapılacak olan robot kolun her bir parçası çizildikten sonra montaj aşamasına geçilmiş ve gerekli olan montaj ilişkileri verilerek çizilen parçaların montajı yapılmıştır. Şekil 5'de tasarlanan 3 eklemli RRR dizilimli robot kol düzeneğinin SolidWorks ' teki son görünümü görülmektedir.



Şekil 5. Robot kolun SolidWorks görüntüsü

Şekil 5'de görüldüğü gibi tasarlanan robot kolun montajı yapılarak home pozisyonunda konumlandırılmıştır. SolidWorks ortamında tasarımı ve montajı yapılan 3 serbestlik dereceli RRR robot kolun Matlab&Simulink ortamındaki modeli elde edilmiştir.

## 2.2. Kinematik Analiz

Kinematik analizleri eklem uzayından kartezyen uzaya (ileri kinematik) ya da kartezyen uzayından eklem uzayına (geri kinematik) geçişi sağlama için kullanırız.

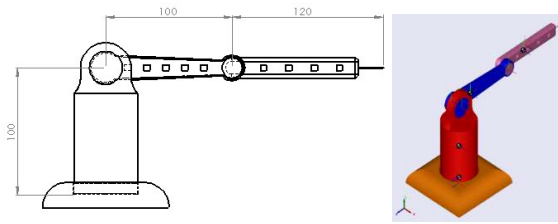
### 2.2.1. İleri kinematik

Tasarlanan 3 serbestlik derecesine sahip RRR robot kolunun ileri kinematik hesaplamalarının yapılabilmesi için DH tablosu oluşturulmuş ve tablo 1'de verilmiştir.

Tablo 1. Robot kolun DH tablosu

i	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	$\theta_1$	90	0	$L_1$
2	$\theta_2$	0	$L_2$	0
3	$\theta_3$	0	$L_3$	0

Şekil 6'da robot kolun 3 boyutlu görünümü ile ileri kinematik için gereken ölçülerin gösterildiği teknik resim çizimi verilmiştir.



Şekil 6. Eksen takımlı robot kol

Tasarlanan robot kolunun ileri kinematik hesabı için DH tablosu dikkate alınarak aşağıdaki her bir eklem için genel dönüşüm matrisi Denklem 1 ve 2'deki gibi tespit edilmiştir.

$${}^0_1T = \begin{bmatrix} \cos(rth1) & -\sin(rth1) & 0 & 0 \\ \sin(rth1) & \cos(rth1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} \cos(rth2) & -\sin(rth2) & 0 & L1 \\ \sin(rth2) & \cos(rth2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^2_3T = \begin{bmatrix} \cos(rth3) & -\sin(rth3) & 0 & L2 \\ \sin(rth3) & \cos(rth3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3_4T = \begin{bmatrix} 1 & 0 & 0 & L3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matrislerde kullanılan L1, L2 ve L3 değerleri robot kolun uzuvlarının uzunlukları olup rth1, rth2 ve rth3 değerleri ise açı değerlerinin radyan birimine çevrilmiş değerleridir. Uç işlevcinin konumu bulunmak için eklem matrisleri çarpımını gösteren ifade Denklem 3’de verilmiştir.

$${}^0T_4 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 \quad (3)$$

İleri kinematik hesabı için kullanılan fonksiyon kodları Şekil 7’de verilmiştir. Kodlar yazılırken denklem 1,2 ve 3’den yararlanılmıştır. İleri kinematik hesabı, referans giriş değeri ile robot modelinin çıkışında yapılmıştır. Bu işlem için Matlab fonksiyonu bloğu kullanılmıştır. Her iki Matlab fonksiyonu bloğunda giriş değerleri derece cinsinden olup matlab fonksiyon bloklarında açılar önce radyan birimine çevrilmiştir. İleri kinematik hesabı yapılırken ilk eklemden başlayarak uç işlevciye kadar olan her bir eklem için rotasyon ve öteleme değerleri bulunan matrisler yazılmaktadır. Tasarlanan robot kolun ileri kinematik hesaplarının sonucunda koordinatlar bulunmuş ve her bir eklem için eklem matrisleri denklem 1 ve 2’ de verilmiştir.

```
function [Px,Py,Pz]= fcn(teta1,teta2,teta3)
l1=100;l2=100;l3=120;%uzunluklar

rth1=teta1*pi/180;rth2=teta2*pi/180;rth3=teta3*pi/180;%radyan donusumu

T_0_1=[cos(rth1) -sin(rth1) 0 0;sin(rth1) cos(rth1) 0 0;0 0 1 0;0 0 0 1];
T_1_2=[cos(rth2) -sin(rth2) 0 l1;sin(rth2) cos(rth2) 0 0;0 0 1 0 :0 0 0 1];
T_2_3=[cos(rth3) -sin(rth3) 0 l2;sin(rth3) cos(rth3) 0 0;0 0 1 0 :0 0 0 1];
T_3_4=[1 0 0 l3;0 1 0 0;0 0 1 0 :0 0 0 1];

T_0_4=T_0_1*T_1_2*T_2_3*T_3_4;
Px=T_0_4(1,4);Py=T_0_4(2,4);Pz=T_0_4(3,4);
```

Şekil 7. İleri kinematik matlab kodları

Şekil 7’de verilen kodda; l ifadeleri uzuv uzunluklarını, rth ifadeleri açarın radyan karşılığını ifade etmektedir. Sonraki satırlarda dönüşüm matrisleri verilmiştir. Verilen matrislerin çarpımı alınarak transformasyon matrisi olan T\_0\_4 matrisi bulunmuştur. Bulunan matrisin dördüncü sütunundaki birinci satır X, ikinci satır Y ve üçüncü satır ise Z koordinatlarıdır.

### 2.2.2. Geri kinematik

Koordinat dönüşümünün sağlandığı bloğun giriş kısmına Eksen-Açıları, Euler Açıkları, Homojen transformasyon matrisi, Kuaterniyon, Döndürme Matrisi, Çeviri Vektörü değerleri girilebilir [13].

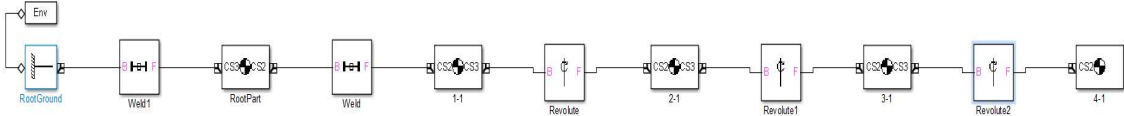
Ters Kinematik bloğu, tanımlanan robot modeline dayalı olarak arzu edilen bir uç efektör pozisyonu için eklem açılarını hesaplamak amacı ile kullanılan Matlab&Simulink bloğudur. Robot modeli, çözücünün uyguladığı tüm eklem kısıtlamalarını tanımlar [14].

Robotun gitmesi istenen koordinatları, robot modele açı olarak girilmesi için kartezyen uzayından eklem uzayına (geri kinematik) geçilmelidir. Bunun için geri kinematik hesabı yapmak amacı ile Matlab&Simulink’ de bulunan inverse kinematics bloğu kullanılmıştır. Ancak inverse kinematics bloğuna girişi, homojen transformasyon matrisi olarak girmek gerektiğinden verilen koordinatlar coordinate transformation conversion bloğuna verilerek homojen transformasyon matrisine dönüştürülmüştür.

Robot modelin end efektörünün gideceği konumunu, homojen transformasyon matrisi olarak, ağırlık değerlerini ve ilk konum tahmin değerleri bloğun girişine verilmiştir. Daha sonra bloğun özelliklerinden diğer parametreler düzenlenmiştir. Bunlara bağlı olarak config çıkışından ekleme verilecek açılar elde edilmiştir

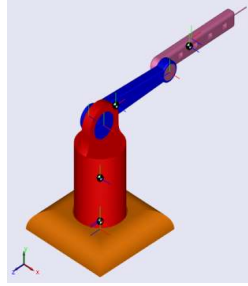
### 2.3. Sistemin Matlab&Simulink modeli

SolidWorks’de kullanılan Simmechanics eklentisi ile elde edilen robot kolun XML dosyasının Matlab&Simulink ortamında çalıştırılması için mech\_import('Montaj.xml') komut satırı kullanılmıştır. Şekil 8’de Matlab&Simulink ortamında oluşturulmuş olan robot kolun Matlab&Simulink modeli verilmiştir.



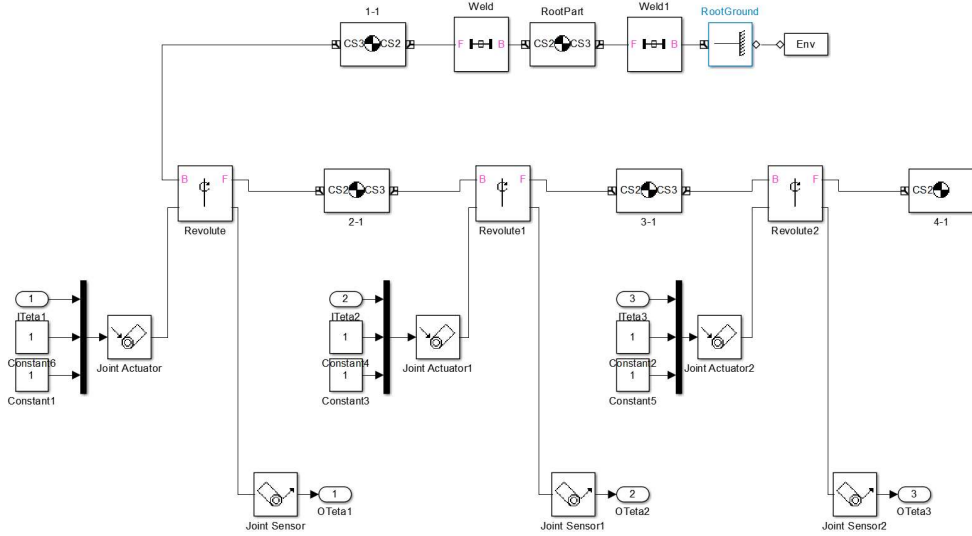
Şekil 8. Matlab&Simulink ortamında oluşturulmuş robot kolun modeli

Şekil 8’de görüldüğü üzere revolute, revolute1 ve revolute3 ismi ile belirtilen bloklar, tasarlanan robot kolun eklemlerini ifade etmektedir. Her eklem bloğunun başında bulunan 1-1, 2-1 ve 3-1 blokları kendilerine ait uzuvların özelliklerini taşımaktadır. Önceki bloklar ise sabit parçayı, yerçekimi ivmesi vb. özellikleri taşımaktadır. Çıkarılan modelin çalıştırılması sonucunda tasarlanan robot kolun Matlab&Simulink programındaki görünümü Şekil 9’da gösterilmiştir.



Şekil 9. Robot kolun Matlab’deki görüntüsü

Robot kolun Matlab&Simulink ortamındaki modelinin her bir eklemine hareketi geçirecek bir aktüatör (joint actuator) ve eklem hareket açısını ölçen bir sensör (joint sensör) eklenmesi ile birlikte oluşturulan model Şekil 10’da görülmektedir.



Şekil 10. Simmechanics eklentisi ile Matlab&Simulink’e aktarılan robot modeli

Şekil 10’da görüldüğü üzere aktüatörü harekete geçirmek için açı, hız ve ivme değerleri girilmelidir. Bu değerlerin girişini mux (kodlayıcı) kullanarak sağlanmıştır. Açı değeri sinüs dalga kaynağından alınmış ve hız için 1, ivme için ise 1 sabit değeri verilmiştir. Sensörden okunan açı değeri ise derece birimindedir. Joint actuatorlerin, actuate with özelliği ile motion yapılmıştır. Revolute bloklarının her birinin dönme yaptığı eksen tanımlanmıştır.

### 2.3.1. ISSA ve KKO algoritmaları

Salps, fiçı biçimli Salpidae familyasından bir deniz organizmasıdır. Genellikle derin denizlerde yiyecek aramak için salp zinciri adı verilen bir sürü oluştururlar. Salp zinciri ön kısımda yer alan bir lider ve onu takip eden bir gruptan oluşmaktadır. Lider besin aramak için gruba kılavuzluk ederken takipçiler önlerindeki salpa göre konumlarını değiştirmektedir. SSA algoritması, Mirjalili ve arkadaşları tarafından bu sürülerin davranışlarının matematiksel olarak modellenmesi ile meydana getirilmiştir [15].

$n$  değişkenli bir optimizasyon problemi ve  $x^i = [x_1^i, x_2^i, \dots, x_n^i]$  öğelerinin bir vektörü ile temsil edilen salp  $i$ 'nin konumu  $x^i$  göz önüne alındığında, salp zincirinin lideri Denklem 1'i kullanarak konumunu güncellemektedir.

$$x_j^1 = \begin{cases} F_j + c_1 \left( (ub_j - lb_j)c_2 + lb_j \right), c_3 > 0.5 \\ F_j - c_1 \left( (ub_j - lb_j)c_2 + lb_j \right), c_3 \leq 0.5 \end{cases} \quad (4)$$

Denklem 4'de  $F_j$ , gıda kaynağı  $F$ 'nin (en iyi bilinen konum)  $j$ . boyutundaki değerdir,  $ub_j$  ve  $lb_j$ ,  $j$ . boyuttaki sırasıyla üst sınır ve alt sınırdır.  $c_2$  ve  $c_3$  parametreleri [0-1] aralıkta rastgele oluşturulmuş sayılardır.  $c_1$  değeri Denklem 5 kullanılarak hesaplanır.

$$c_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \quad (5)$$

Denklem 5'de  $t$ , geçerli iterasyon ve  $T$ , toplam iterasyon sayısıdır. Liderin konumu güncellendiğinde, takipçiler de denklem 6'yı kullanarak konumlarını değiştirmektedir.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-j}) \quad (6)$$

Denklem 6'da  $x_j^i, 2 \leq i \leq n$  ile salp  $i$ 'nin  $j$ . boyutundaki konumdur. Yinelemeli süreç, liderin konumunda daha yüksek rastgele değişikliklere izin veren ve arama uzayının keşfini vurgulayan  $c_1$  için yüksek değerlerle başlamaktadır. Arama aşaması, parametre katlanarak azalırken gerçekleşir ve liderin besin kaynağı çevresinde daha küçük ayarlamalar yapmasına izin vermektedir.

Algoritmanın tanıtılmasından sonra birçok araştırmacı daha iyi performans elde etmek üzere çalışmalar yapmıştır. Bu çalışmalardan biride Tudose ve diğerleri orijinal SSA'da, arama sürecindeki kötü bireyleri elimine eden ve yeni bireyler oluşturan ISSA'yı önermişlerdir. ISSA'nın orijinal salp swarm algoritması için önerdiği iyileştirmeler şöyledir;

- Arama uzayında salpların daha iyi dağılımını sağlamak için başlangıç popülasyonunun oluşturulmasında karşılıklı tabanlı bir öğrenme stratejisini içermektedir.

$$\tilde{x}_j^i = lb_j + ub_j - x_j^i \quad (7)$$

- Önerilen başlatma yöntemi, popülasyon boyutundan ( $N_{tot}$ ) daha büyük bir dizi başlangıç bireyinin ( $N_{init}$ ) seçilmesinden oluşmaktadır. Başlangıç bireylerinin yarısı ( $N_{init}/2$ ) geleneksel rastgele üretim yöntemiyle sağlanırken, popülasyonun geri kalanı denklem 6'ya göre zıt bireyler üretilerek tamamlanmaktadır. Son olarak, amaç fonksiyonu tüm ilk bireyler için değerlendirilmekte ve en iyi  $N_{tot}$  bireyleri ISSA'nın ilk yinelemesinde korunmaktadır.
- Sürünün keşif yeteneklerini artırmak için lider salpa benzer şekilde davranan "keşifçi salps" adı verilen yeni bir salp türü entegre edilmiştir. Denklem 5'e göre her takipçi salp bir öncekinden etkilenmekte ve bu durumda, bir performans sıralamasına göre önceki salps sıralaması ile daha iyi bir yakınsama oranı elde edilebilmektedir. En iyi birey lider salp olarak seçilmekte, takip eden  $N_{exp}$  keşifçi salp olarak ve kalan  $N_{fol}$  ise takipçi salp olarak kabul edilmektedir ( $N_{fol} = N_{tot} - 1 - N_{exp}$ ). Son olarak, son yinelemeler sırasında keşif sürecine daha iyi bir destek sağlamak için yinelemeler ilerledikçe, keşifçi salpların sayısı ( $N_{exp}$ ) doğrusal olarak artmaktadır.
- Üçüncü iyileştirme, Keşif salpları için ek bir konum güncelleme mekanizması olarak klasik bir çaprazlama operatörü eklenmekte, yinelemeli süreç sırasında artan bir olasılık ( $p_{CO}$ ) ile etkinleştirilmektedir. Çaprazlama operatörü, keşif salpı ve besin kaynağı arasında ağırlıklı ortalama olarak tek bir yavru üretmektedir. Ayrıca, aynı  $p_{CO}$  olasılığı ile yavrular üzerindeki gıda kaynağı etkisini artırmak için denklem 8 kullanılarak çaprazlama operatörü için ikinci bir varyasyon sunulmuştur. Denklem 8'de  $r_1$  ve  $r_2$  [0,1] aralığında rastgele oluşturulmuş sayılardır.

$$x_j^i = \begin{cases} F_j \cdot r_2 + x_j^i(1 - r_2), r_1 > 0.5 \\ F_j \cdot \left(1 - \frac{r_2}{2}\right) + x_j^i \cdot \frac{r_2}{2}, r_1 \leq 0.5 \end{cases} \quad (8)$$

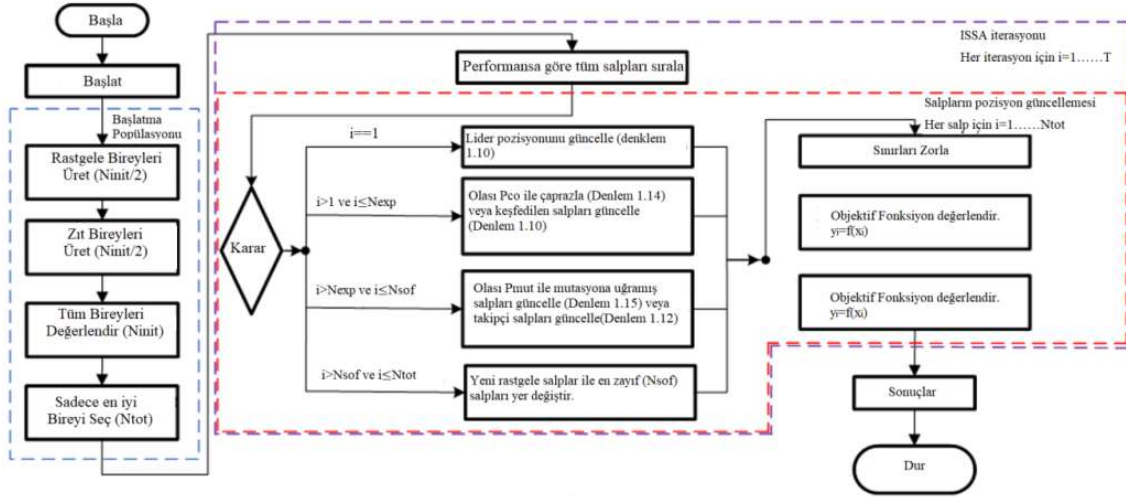
- Yerel optimumda durgunluk riskini azaltmak için sadece mutasyon olasılığı olan takipçi salplara uygulanan bir mutasyon operatörü  $p_{mut}$  içermektedir. Mutasyona uğramış takipçi salp, rastgele seçilen



bir salpin konumunu kopyalamakta ve daha sonra bu konumun etrafında keşifçi bir salpa benzer şekilde davranmaktadır. Arama sürecine müdahale etmekten kaçınmak için, yinelemeler ilerledikçe mutasyon olasılığı doğrusal olarak azalmaktadır.

$$x_j^1 = \begin{cases} x_j + m_1 ((ub_j - lb_j)m_2 + lb_j), & m_3 > 0.5 \\ x_j - m_1 ((ub_j - lb_j)m_2 + lb_j), & m_3 \leq 0.5 \end{cases} \quad (9)$$

Denklem 9'da  $m_1, m_2$  ve  $m_3$  [0-1] aralıkta rastgele oluşturulmuş sayılardır. ISSA, eliminasyon mekanizmasını en düşük performanslı  $N_{SOF}$  salp'lara da entegre etmektedir. Önerilen iyileştirmelerin entegrasyonu ile değiştirilmiş salp sürüsü algoritmasının akış şeması Şekil 11'de verilmiştir [16].



Şekil 11. ISSA akış diyagramı.

1992 yılında doktora tezinde karınca kolonisi algoritmasını (KKO) ilk olarak çalışan kişi Marco Dorigo'dur. Karıncaların belirli bir mantığa göre hareket ettiğini düşünen Dorigo, KKO'yu, gezgin satıcı problemi ve araç rotalama problemi gibi optimizasyon problemlerine uygulamıştır. 200 adet karınca kullanarak deney yapan Dorigo, çeşitli yiyecek ve engeller koyarak her bir karıncanın hareketini incelemiştir. Bu gözlemlerin sonucunda karıncaların en kısa yolu bulduklarını görmüştür [17].

Salgıladıkları feromonlardan dolayı karıncalar her zaman kısa yolu seçmektedirler. Kimyasal bir salgı olan feromoni, bazı hayvanlar kendi cinslerinden başka hayvanları etkilemek için kullanırlar. Karıncalar ilerlerken, belirli bir miktar feromon depo ederler. Olasılığa dayanan bir yöntemle feromonun daha çok olduğu yolu tercih ettikleri görülmüştür. Depoladıkları feromonları, gıdaya giderken seçtikleri yola bırakarak, kendilerinden sonraki karıncalara yol seçiminde yardımcı olurlar. Bu içgüdüsel davranış, onların gıdaya giden en kısa yolu bulmalarını sağlamaktadır [17]. KKO'nun araç rotalama problemi çözümü için uygulama adımları aşağıdaki gibi yapılmaktadır.

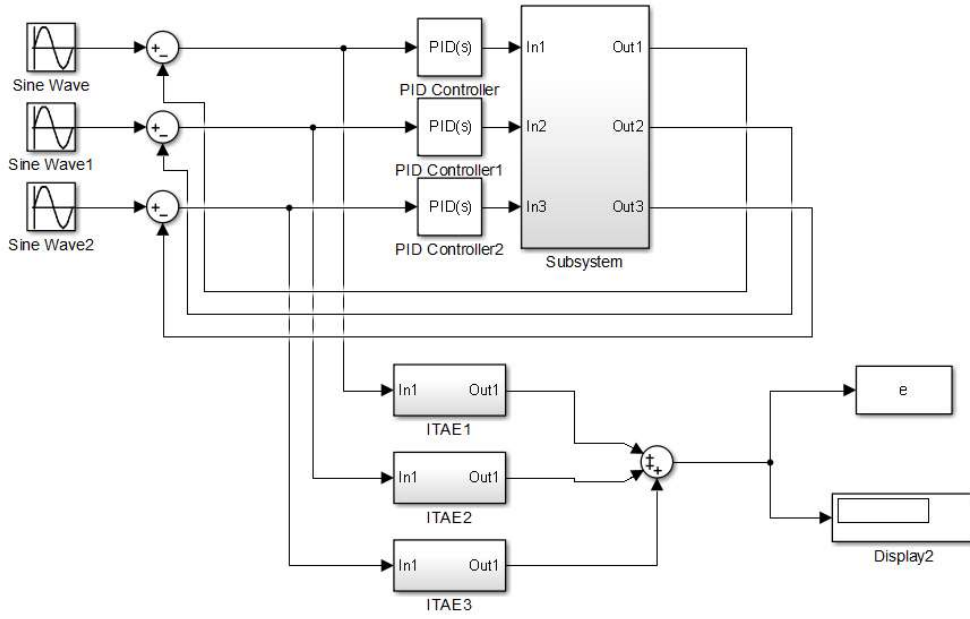
1. Rastgele seçilen şehirlerden "m" karınca serbest bırakılmaktadır.
2. Tanımlanan parametreye göre karınca gezginci veya takipçi olabilmektedir.
3. Her bir karınca gideceği şehri seçerken feromon miktarı ve şehirlerarası mesafeye göre seçmektedir.
4. Her bir kenardaki feromon miktarları güncellenmektedir.
5. Bütün şehirler gezilene kadar 2. ve 3. aşama tekrar edilmektedir.
6. Bütün karıncalar turu tamamladığında en çok feromon içeren kenarlar bu turdaki en iyi sonucu oluşturmaktadır.
7. Karıncaların hafızaları silinmektedir.
8. Durma kriteri sağlanınca durmakta yoksa adımlar tekrarlanmaktadır.

Genel olarak kontrol tasarımında dikkate alınan dört tür performans indeksi bulunmaktadır. Bunlar, mutlak hatanın integrali (IAE), karesel hatanın integrali (ISE), zaman ağırlıklı karesel hatanın integrali (ITSE) ve zamanla çarpılan mutlak hatanın (ITAE) integralidir [18]. Şekil 2'de IAE performans indeksinin Matlab&Simulink modeli verilmiştir. Gösterilen model her bir PID kontrolü için yapıлып, bulunan değerler toplanmıştır. Sistemin IAE performans indeksine göre performansı belirlenmiştir.

Şekil 12'de görüldüğü üzere girişlere referans sinyali olarak sinüs sinyali uygulanmıştır. Robot modelinin olduğu subsystem'in çıkışı PID kontrolör için geri besleme olarak alınmıştır. Bu iki değer arasındaki fark ise PID



kontrolöre verilmiştir. ISSA ve KKO algoritması ile PID katsayılarının tespiti için Matlab&Simulink ortamında oluşturulmuş model Şekil 12’de görülmektedir.



Şekil 12. PID katsayılarının tespiti için hazırlanan Matlab&Simulink modeli

Şekil 12’de sinüs girişi için ISSA ve KKO algoritması tarafından PID parametrelerinin tespiti için ITAE performans indeksine göre hata değeri alınmıştır. Bu hata değeri algoritma kodu tarafından kullanılmış ve birim basamak girişi için, giriş sinyalleri değiştirilmiştir.

ISSA ve KKO algoritması ile PID katsayılarının tespiti için her bir PID kontrol bloğunun katsayılarına Tablo 2’ de verilen değişkenler sırasıyla ISSA için ayrı bir değişken tanımlanıp çalıştırılmıştır. KKO algoritması için ise ayrı değişkenler tanımlanarak algoritmalar çalıştırılmıştır.

Tablo 2. PID katsayılarının belirlenmesi için verilen değişkenler

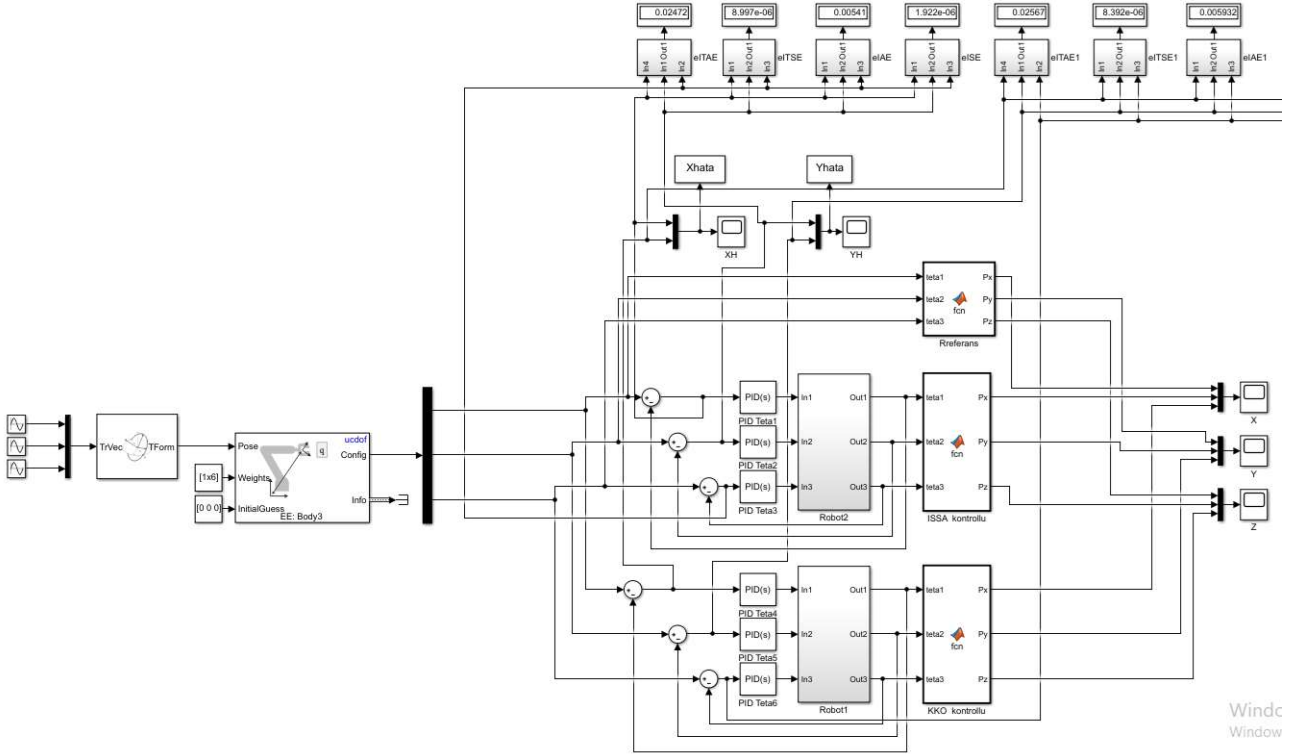
	ISSA			KKO		
	PIDTeta1	PIDTeta2	PIDTeta3	PIDTeta4	PIDTeta5	PIDTeta6
Kp	kp1	kp2	kp3	kkp1	kkp2	kkp3
Ki	ki1	ki2	ki3	kki1	kki2	kki3
Kd	kd1	kd2	kd3	kkd1	kkd2	kkd3

ISSA çalıştırdıktan sonra elde edilen PID parametreleri olan kp, kd, ki değerleri tespit edilmiş ve elde edilen Kp, ki ve kd değerleri Tablo 3’ de verilmiştir.

Tablo 3. Hesaplanan PID parametreler

ISSA ’daki Kontrolör Kazanç Katsayıları	ISSA		KKO ’daki Kontrolör Kazanç Katsayıları	KKO	
	SİNÜS	STEP		SİNÜS	STEP
Kp1	999.9995195	733.8538922	kkp1	959.5959596	363.6363636
Ki1	999.9686933	1000	kki1	585.8585859	878.7878788
Kd1	39.3753549	147.8304805	kkd1	838.3838384	0
Kp2	999.9991774	394.5620992	kkp2	707.0707071	292.9292929
Ki2	999.6168899	999.9996837	kki2	939.3939394	979.7979798
Kd2	7.2576213	41.53488073	kkd2	151.5151515	10.1010101
Kp3	999.9997807	333.979338	kkp3	979.7979798	282.8282828
Ki3	22.42295676	1000	kki3	40.4040404	949.4949495
Kd3	998.9932794	29.89633793	kkd3	838.3838384	30.3030303

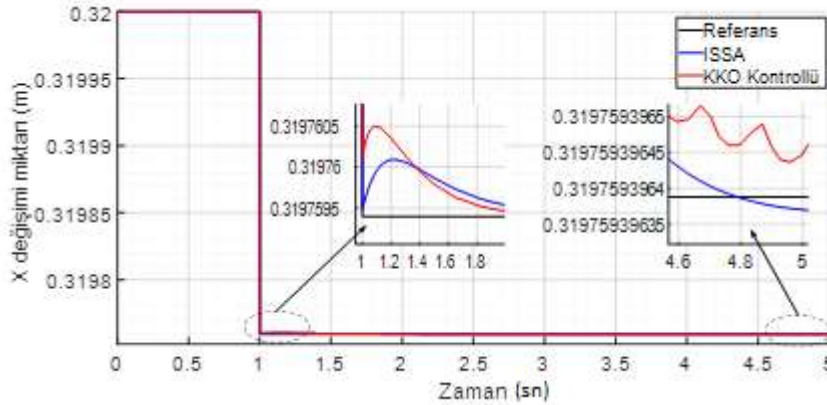
Tablo 3’de verilen PID katsayıları, PID katsayı değişkenlerine atanmış ve ISSA, KKO algoritması ile tespit edilen parametrelere sahip PID kontrolörlü robot kolun çalışma performansının izlenmesi için oluşturulan model Şekil 13’de verilmiştir. Sinüs girişi ya da birim basamak girişi için sistemdeki PID katsayıları optimizasyondaki bulunan katsayılara göre değiştirilerek simülasyon çalıştırılmıştır. Şekil 13’ de görüldüğü üzere performans indekslerinin tespiti için subsystem blokları tasarlanarak modele eklenmiştir. Modele, ileri kinematik fonksiyon bloğu ve PID kontrolörler eklenmiştir.



Şekil 13. KKO ve ISSA algoritmalarıyla PID kontrollü çalışma performansının izlenmesi için oluşturulan model

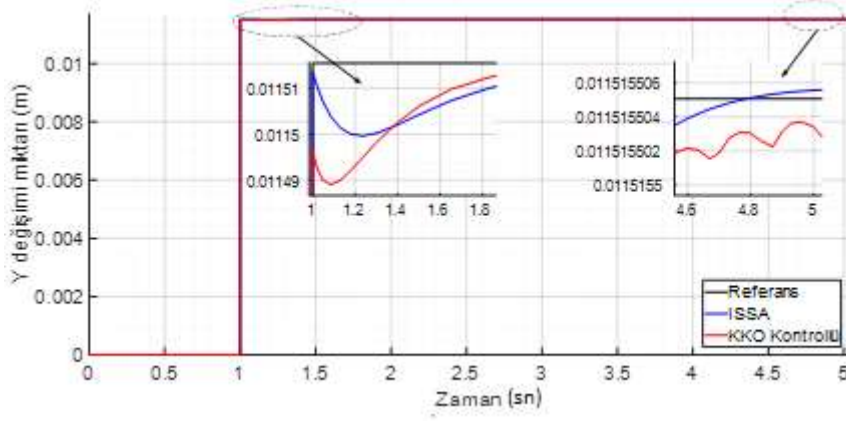
### 3. Bulgular

Yapılan çalışmada tasarlanan 3 serbestlik dereceli robot kolunun, ISSA ve KKO algoritması ile katsayıları tespit edilen PID kontrolör uygulanarak robot kolu uç efektörünün konum analizi yapılmıştır. Bu analizler için birim basamak ve sinüs girişi kullanılarak elde edilen sonuçlar incelenmiştir. Her bir giriş için kontrolörün katsayıları iki algoritma ile optimize edilmiş ve birim basamak girişi için Şekil 14’de X, Şekil 15’de Y ve Şekil 16’da Z koordinatlarının değişim grafikleri verilmiştir. ISSA ve KKO algoritması ile elde edilen katsayıları kullanarak yapılan PID kontrolörün X eksenindeki değişimi ile referans yörünge Şekil 14’de verilmiştir.



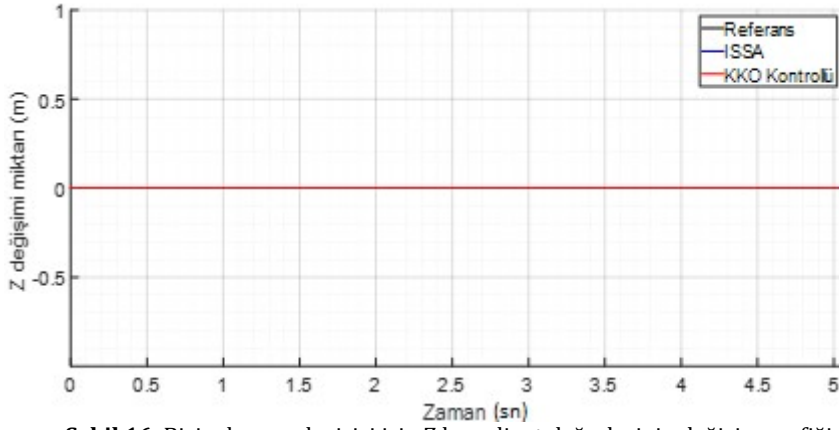
Şekil 14. Birim basamak girişi için X koordinat değerlerinin değişim grafiği

Robot kolu birim basamak girişi uygulandığında, X eksenini için ISSA'nın, referans yörüngeye daha yakın izlediği Şekil 14’de görülmektedir. ISSA ve KKO algoritması ile elde edilen katsayıları kullanarak yapılan PID kontrolörün Y eksenindeki değişimi ile referans yörünge Şekil 15’de verilmiştir.



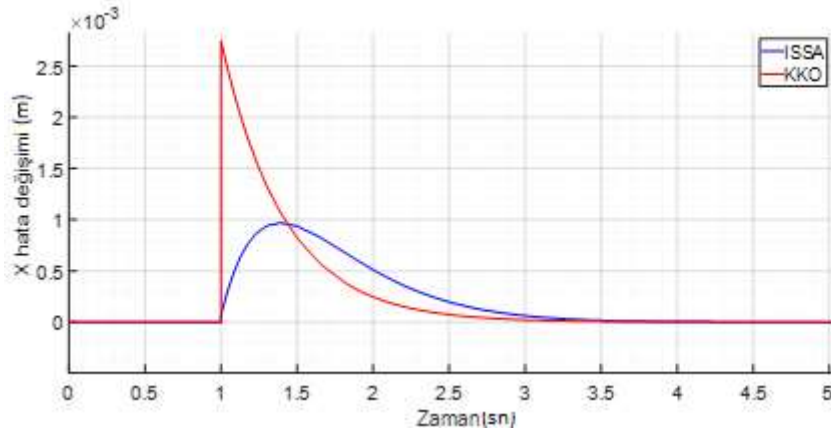
Şekil 15. Birim basamak girişi için Y koordinat değerlerinin değişim grafiği

Robot kola birim basamak girişi uygulandığında, Y eksenini için ISSA 'nın referans yörüngesini daha yakın izlediği Şekilde 15'de görülmektedir.



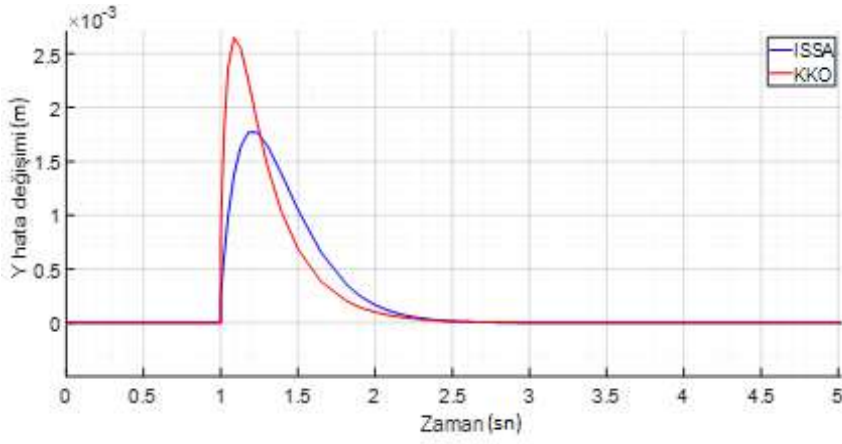
Şekil 16. Birim basamak girişi için Z koordinat değerlerinin değişim grafiği

Şekil 16'da görüldüğü üzere birim basamak olan referans sinyali iki boyutlu olduğundan Z ekseninde herhangi bir değişim olmamış ve bundan dolayı Z hata sinyali incelenmemiştir. Birim basamak girişi için, X eksenindeki kontrolörlerin hata sinyalleri Şekil 17'de verilmiştir.



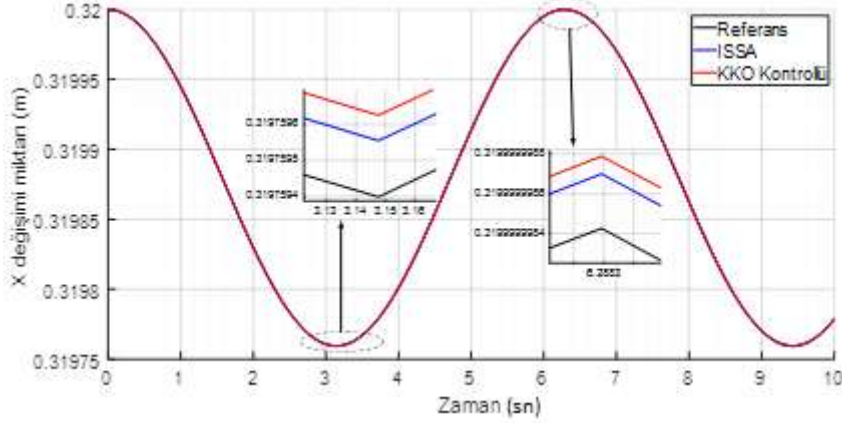
Şekil 17. Birim basamak için X eksenindeki hata grafiği

Robot kola birim basamak girişi uygulandığında, X eksenini için KKO algoritmasının hata sinyali salınımı ISSA'ya göre daha fazla olduğu Şekil 17'de görülmekte ve hata değerinin oturma süresi ise yaklaşık aynı olmaktadır. Birim basamak girişi için, Y eksenindeki kontrolörlerin hata sinyalleri Şekil 18'de verilmiştir.



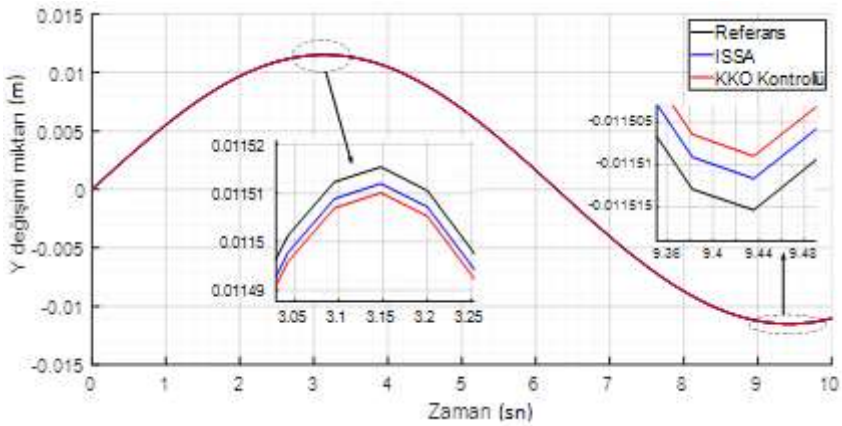
Şekil 18. Birim basamak sinyal girişinin Y eksenindeki hata grafiği

Robot kola birim basamak girişi uygulandığında, Y eksenini için KKO algoritmasının hata sinyalinin salınımı ISSA'na göre daha fazla olduğu Şekil 18'de görülmekte ve hata değerinin oturma süresi ise her iki algortmada elde edilen katsayılar için yaklaşık aynı olmaktadır. Sinüs dalga girişi için Şekil 19' da X, Şekil 20' de Y, Şekil 21' de Z koordinatlarının değişim grafikleri verilmiştir. ISSA ve KKO algoritması ile elde edilen katsayıları kullanarak yapılan PID kontrol sonucu X koordinatındaki değişim ve referans yörünge Şekil 19'da verilmiştir.



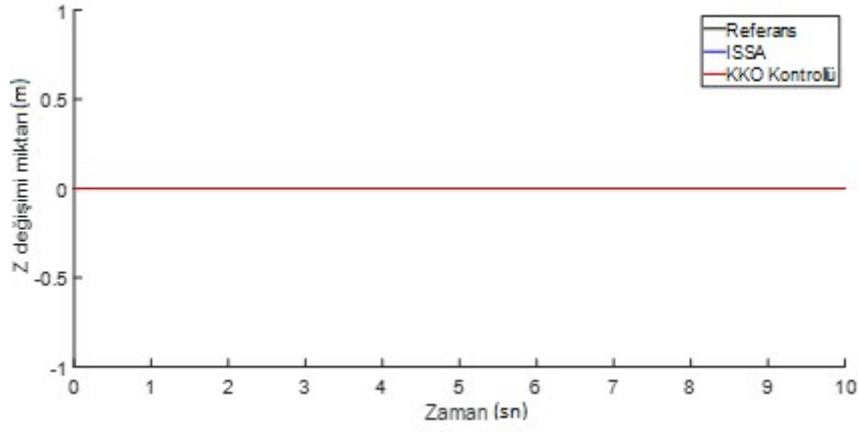
Şekil 19. Sinüs sinyal girişi için X koordinat değerlerinin değişim grafiği

X koordinatı için ISSA ile katsayıları belirlenen kontrolün, referans yörüngesini daha yakın izlediği Şekil 19' da görülmektedir. ISSA ve KKO algoritması ile elde edilen katsayıları kullanarak yapılan PID kontrol sonucu Y koordinatındaki değişim ve referans yörünge Şekil 20'de verilmiştir.



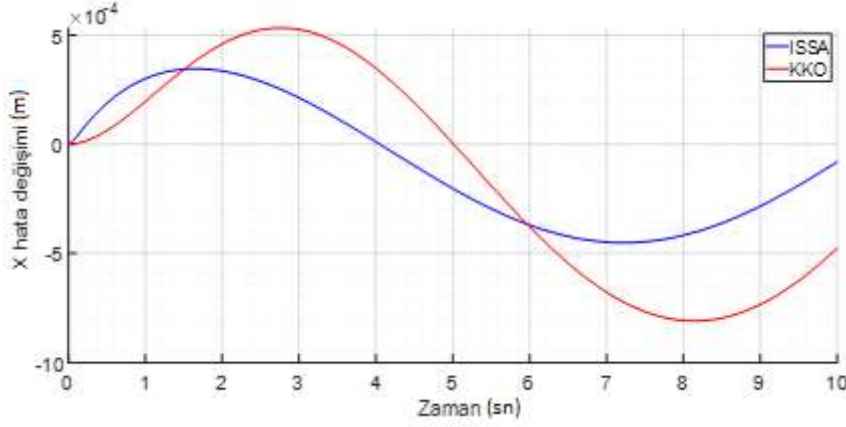
Şekil 20. Sinüs sinyal girişi için Y koordinat değerlerinin değişim grafiği

Y koordinatı için ISSA ile katsayıları belirlenen kontrolün, referans yörüngesini daha yakın izlediği Şekil 20'de görülmektedir.



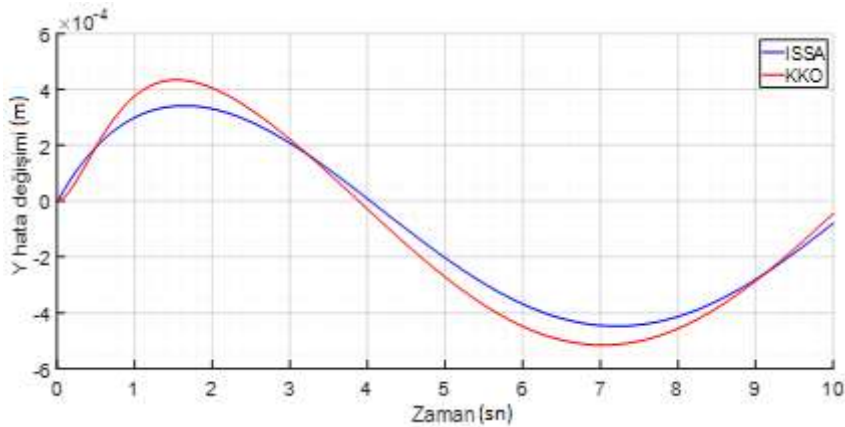
Şekil 21. Sinüs sinyal girişi için Z koordinat değerlerinin değişim grafiği

Şekil 21’de görüldüğü gibi Z koordinatında herhangi bir değişim olmadığından Z hata sinyali incelenmemiş ve şekil 22’de X hata sinyali, şekil 23’de ise Y hata sinyalinin değişim grafikleri verilmiştir. X koordinatının kontrolünde kullanılan kontrolörlerin hata sinyalleri Şekil 22’de verilmiştir.



Şekil 22. Sinüs sinyal girişi için X eksenindeki hata grafiği

X koordinatı için, katsayıları KKO algoritması ile belirlenen kontrolde tepe değerleri arasındaki fark ISSA ile yapılan kontrole göre daha fazla olduğu Şekil 22’de görülmektedir. Y eksenindeki kontrolörlerin hata sinyalleri Şekil 23’de verilmiştir.



Şekil 23. Sinüs sinyal girişi için Y eksenindeki hata grafiği

Y eksenini için KKO algoritması ile katsayıları belirlenen kontrolde tepe değerleri arasındaki fark ISSA ile yapılan kontrole göre daha fazla olduğu Şekil 23’de görülmektedir.

Her bir koordinattaki kazançlar ise ITAE, ITSE, IAE, ISE performans indeksleri açısından değerlendirilmiş ve PID kontrolörlü sistemin performans indeks değerleri Tablo 4’de verilmiştir.

**Tablo 4.** Performans indeksi değerleri

OPTİMİZASYON ALGORİTMASI	PERFORMANS İNDEKSİ	SİNÜS	BİRİM BASAMAK
ISSA	ITAE	0.06744	0.004468
	ITSE	3.73e-05	4.496e-06
	IAE	0.01293	0.003001
	ISE	7.016e-06	3.361e-06
KKO	ITAE	0.084	0.004354
	ITSE	5.941e-05	6.156e-06
	IAE	0.01576	0.003233
	ISE	1.076e-05	5.053e-06

#### 4. Tartışma ve Sonuç

Bu çalışmada SolidWorks kullanılarak 3 serbestlik derecesine sahip robot kol mekanik olarak tasarlanmış ve Matlab&Simulink modeli elde edilmiştir. Matlab&Simulink'te çağrılacak olan İleri kinematik hesapları Matlab'da fonksiyon olarak yazılarak PID kontrolör ile robot kola açılış değerleri verilmiş ve pozisyon kontrolü sağlanmıştır. Simülasyonda ISSA ve KKO algoritması ile elde edilen katsayılar kullanılarak simülasyona sinüs ve birim basamak girişi verilerek çalıştırılmıştır. Farklı iki giriş için çalıştırılan simülasyondan elde edilen sonuçlar karşılaştırılmıştır. Elde edilen veriler değerlendirildiğinde Şekil 13 ve Şekil 14'deki grafiklerde de görüldüğü gibi birim basamak girişi için elde edilen sonuçlarda, ISSA'nın, KKO algoritmasından daha yakın bir şekilde referans sinyalini izlediği ve daha iyi sonuç verdiği görülmektedir. Ayrıca PID kontrolör parametrelerinin ISSA ile optimum düzeyde ayarlanabildiği görülmektedir.

Şekil 18 ve Şekil 19'da verilen grafiklerde görüldüğü gibi sinüs dalga girişi için elde edilen sonuçta da, ISSA'nın, KKO algoritmasından daha yakın bir şekilde referans sinyalini izlediği ve daha iyi sonuç verdiği görülmektedir. ISSA ve KKO algoritması ile PID parametrelerinin belirlendiği simülasyon programının çalıştırılması sonucu elde edilen performans parametreleri Tablo 4'de verilmiştir. Performans parametrelerinin tespiti ise ITEA, ITSE, IAE ve ISE olarak alınmıştır.

Elde edilen veriler ve alınan sonuçlar değerlendirildiğinde, farklı sayıda ekleme sahip robot kolları ve farklı boyutlardaki robotların PID kontrolörlerindeki Kp, Ki ve Kd katsayılarının tespiti için ISSA'nın kullanılabileceği ve tasarlanarak imal edilecek bir robot kola gerçek zamanlı olarak uygulanıp simülasyon değerleri ile gerçek zamanlı deney sonuçlarının karşılaştırılabileceğini göstermiştir. Ayrıca PID katsayılarını belirlemek için farklı ve daha fazla sayıda algoritmalar aynı robot kol için kullanılarak karşılaştırma yapılabileceği belirlenmiştir. Bu çalışma ile farklı sistemlerin PID kontrolörünün katsayıları belirlenirken, ISSA kullanılarak sistemde oluşan hataların minimize edilebileceği ve sistemin daha verimli çalışabileceği ve istenildiği takdirde yapılacak yeni eklemelerle geliştirilebileceği düşünülmektedir.

#### Kaynakça

- [1] He, X., Chen, Y. (2009). Haptic-Aided Robot Path Planning Based On Virtual Tele-Operation, Robot. Comput. Integr. Manuf., vol. 25, no. 4–5, pp. 792–803, doi: 10.1016/j.rcim.2008.12.002.
- [2] Aydos, M., Profile, S., Nalbant, A. D. (2015). PID Tabanlı Robot Kolu Kontrolü: Ziegler-Nichols ve Tyreus-Luyben Metotlarının ODE45 Deneysel Çözümleme ile Karşılaştırılması, Accessed: Dec. 21, 2022. [Online]. Available: <https://www.researchgate.net/publication/321361822>.
- [3] Capkan, Y., Fidan, C. B. (2020). 4 Eksenli Robot Kolun Knematik Hesaplarının Gerçek Model Üzerinde Matlab ile Analz ve Test, HORA 2020 - 2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc., doi: 10.1109/HORA49412.2020.9152851.
- [4] Kayışlı, K., Uğur, M. (2017). 3 Serbestlik Dereceli Bir Robot Kolun Bulanık Mantık ve PID ile Kontrolü, Gazi Üniversitesi Fen Bilim. Derg. Part C Tasarım ve Teknol., vol. 5, no. 4, pp. 223–234, doi: 10.29109/http-gujsc-gazi-edu-tr.339907
- [5] Beyhan, A. (2021). Üç Serbestlik Dereceli Robotik Kolun Kartezyen Uzayda Empedans Kontrolü, BTÜ, Fen Bilimleri Enstitüsü.
- [6] Donuk, K., Özbey, N., Inan, M., Yeroğlu, C., Hanbay, D. (2019). PIDA Denetçi Parametrelerinin PSO Algoritması ile Belirlenmesi, 2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018.
- [7] Hamida, Y., Naser, E. L. (2018). Endüstriyel Amaçlı Robot Kol Tasarımı Ve Üretimi, Accessed: Dec. 21, 2022. [Online]. Available: <https://acikerisim.sakarya.edu.tr/handle/20.500.12619/79304>.



- [8] Ağralı, E., Çavaş, M. (2020). V-REP Robotik Simülasyon ile Robot Kol Simülasyonu, Fırat Univ. J. Eng. Sci., vol. 32, no. 2, pp. 435-444, Accessed: Dec. 21, 2022. [Online]. Available: <https://dergipark.org.tr/en/pub/fumbd/issue/56883/687730>.
- [9] Karakoyun, E., Çakan, A., Kalyoncu, M. (2022). İki Serbestlik Dereceli Bir Robot Kolun Konum Kontrolü için PID kontrol Parametrelerinin Arı Algoritması (AA) Kullanılarak Belirlenmesi, Balıkesir Üniversitesi Fen Bilim. Enstitüsü Derg., vol. 24, no. 1, pp. 111-124, doi: 10.25092/baunfbed.885152.
- [10] Ergocmen, B., Tilki, U. . Two Degree of Freedom Manipulator Control by SDRE Algorithm Trajectory Following and Autolanding with Incremental Nonlinear Dynamic Inversion for NASA GTM Model for Single Engine Operations View Project Fault Tolerant Flight Control with Control Allocation and SDRE for NASA GTM design View project, Accessed: Dec. 21, 2022. [Online]. Available: <https://www.researchgate.net/publication/354366216>.
- [11] Abut, T., Soygüder, S., Alli H. (2012). Bir Yılsarı Robotun Tasarımı ve Kinematik Analizi, Süleyman Demirel Üniversitesi Fen Bilim. Enstitüsü Derg., vol. 18, no. 1, pp. 8-13.
- [12] Güven A.F. (2023). 2DOFPID ve Kesir Dereceli PID Kontrolör ile Ayarlanan Senkron Motorların Kontrolü Yardımıyla İki Eksenli Robot Kolu Pozisyonunun Ayarlanması, Karadeniz Fen Bilimleri Dergisi, vol. 13, no. 2, pp. 625-638.
- [13] Mathworks Help Center. Inverse Kinematics. <https://www.mathworks.com/help/robotics/ref/inversekinematics.html> (Erişim Tarihi: 11.11.2023).
- [14] Mathworks Help Center. Coordinate Transformation Conversion. [https://www.mathworks.com/help/ros/ref/coordinateconversion.html?s\\_tid=srchtitle\\_support\\_results\\_2\\_Coordinate%20Transformation%20Conversion](https://www.mathworks.com/help/ros/ref/coordinateconversion.html?s_tid=srchtitle_support_results_2_Coordinate%20Transformation%20Conversion) (Erişim Tarihi: 11.11.2023).
- [15] Tudose, A. M., Picioroaga, I. I., Sidea, D. O., Bulac, C. (2021). Solving Single-And Multi-Objective Optimal Reactive Power Dispatch Problems Using An Improved Salp Swarm Algorithm, Energies, vol. 14, no. 5, p. 1222, doi: 10.3390/en14051222.
- [16] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., Mirjalili, S. M. (2017). Salp Swarm Algorithm: A Bio-Inspired Optimizer For Engineering Design Problems, Adv. Eng. Softw., vol. 114, pp. 163-191, doi: 10.1016/j.advengsoft.2017.07.002.
- [17] Dikmen, H., Dikmen, H., Elbir, A., EKŞİ, Z., ÇELİK, F. (2014). Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması, Süleyman Demirel Üniversitesi Fen Bilim. Enstitüsü Derg., vol. 18, no. 1, pp. 8-13.
- [18] Barisal, A. K. (2015). Comparative Performance Analysis Of Teaching Learning Based Optimization For Automatic Load Frequency Control Of Multi-Source Power Systems, Int. J. Electr. Power Energy Syst., vol. 66, pp. 67-77, doi: 10.1016/j.ijepes.2014.10.019.