

## Noktasal Kontakta Sahip Zıplayan Robotta Duruş Anında Dengenin Sağlanması

Halit Hülako<sup>1\*</sup>

<sup>1</sup> Makine Mühendisliği, Mühendislik Fakültesi, Hakkari Üniversitesi, Hakkari, Türkiye

\*<sup>1</sup> halithulako@hakkari.edu.tr

(Geliş/Received: 06/01/2024;

Kabul/Accepted: 10/07/2024)

**Öz:** Bu çalışmada, yer ile noktasal bir temas yüzeyine sahip olan zıplayan düzlemsel dinamik bir robotun zıplama sonrasında devrilmeden dengesini koruması incelenmiştir. Robotun dengesini sağlamak için Reaksiyon Tekerleği kullanılmıştır. Zıplama sonrasında yere iniş anı simüle edilmiş ve robotun yere temas ettiği anda dengesini koruyarak istikrarlı bir duruş sergilemesi için Genetik algoritma kullanarak Pekiştirmeli Öğrenme (PÖ) yapısına benzer bir Yapay Zeka Ajanı oluşturulmuştur. Bu yöntemle zıplayan robot, gerçek zamanlı olarak dengelenme misyonu için eğitilmiş ve ardından robotun dengeleme işlemini başarıyla öğrendiği gözlemlenmiştir. Zıplayan robotun dinamik modellenmesi Matlab Simscape MultiBody (MSM) kullanılarak gerçekleştirilmiştir. Nümerik Simülasyona ait grafiksel sonuçlar, dinamik robot modelinin, zıplamadan sonra devrilmeden dengesini 5 saniye boyunca başarıyla sürdürdüğünü ve istenen düzeltmeleri yapabileceğini Yapay Zeka'nın katkısıyla sergilediğini göstermektedir.

**Anahtar kelimeler:** Zıplayan robot, E. pekiştirmeli öğrenme, reaksiyon tekeri.

### Maintaining Stability During Rest for Jumping Robot with Point Contact

**Abstract:** This study investigates a planar modelled dynamic jumping robot with a point ground contact surface. The aim is to maintain the robot's balance after jumping without tipping over. To achieve this, a Reaction Wheel is employed for stabilization. The landing moment after a jump is simulated, and an Artificial Intelligence algorithm similar to Reinforcement Learning (RL), which uses Genetic algorithm, is used to maintain a stable posture by maintaining balance at the moment the robot touches the ground. With this method, the jumping robot was trained for the stabilisation mission in real-time and then it was observed that the robot successfully learnt the stabilisation process. The jumping robot's dynamic modelling was carried out using Matlab Simscape MultiBody (MSM). The graphical results of the numerical simulation show that the dynamic robot model successfully maintains its balance for 5 seconds without tipping over after jumping and exhibits the ability to make the desired corrections with the contribution of Artificial Intelligence.

**Key words:** Jumping robot, E. reinforcement learning, reaction wheel.

### 1. Giriş

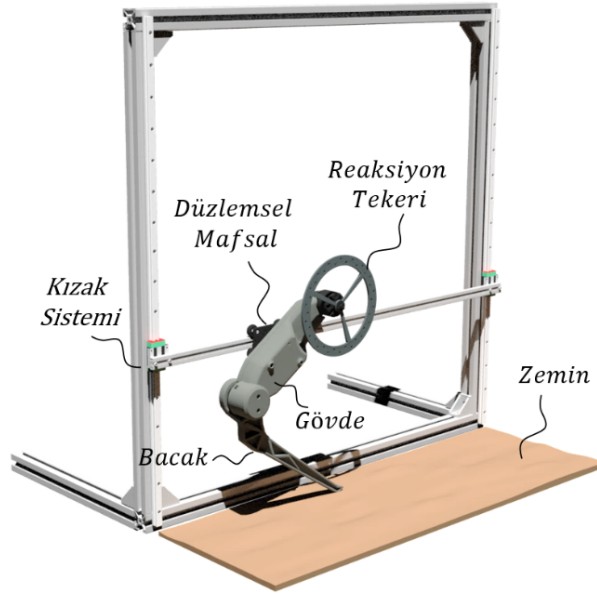
Statik kararlılık kriterlerinde robotun ağırlık merkezinin izdüşümü destek poligonu arasında yer alıyorsa robot devrilmeden dengeli duruş sağlayabilir fakat dinamik sistemlerde destek poligonu bulunmadığından robot dengeli sağlayabilmek için sürekli hareket etmelidir. Dinamik bir sistem olan zıplayan robotta da bacak noktasal bir şekilde yer ile temas ettiğinden dengeli sağlayabilmek için sürekli zıplama hareketi yapmak zorundadır. Robot durup beklediğinde ise devrilmektedir. Beş serbestlik dereceli (SD) düzlemsel bir robot, zıplama sonrasında denge sağlanıp kontrol edilmeye çalışılmıştır. Zıplamadığı esnada devrilmesini önlemek için çözüm olarak Reaksiyon tekeri kullanılmıştır. Robot, bir gövde ve tek eksenli bir bacadan oluşmakta ayrıca gövde üzerine yerleştirilen Reaksiyon Tekerleği vasıtasıyla eksik tahrikli olan ana gövde eksenlerinin hareketleri kontrol edilebilmektedir. Şekil 1'de gösterilen deney düzeneği Matlab Simscape Multibody ile modellenmiştir. Robot, dengeli sağlamak için Reaksiyon Tekerleği ve bacak eklemine bulunan eyleyiciyi kullanarak düşme etkisini kontrol eder.

Günümüzde reaksiyon tekerleri uzay teknolojisi gibi çok önemli alanlarda kullanılmaktadır. Genel olarak eyleyicinin doğrudan bağlanıp harekete geçirmenin mümkün olmadığı eksen takımlarında tercih edilen önemli bir yöntemdir. Çevresiyle sürekli temas halinde olan robotlar yönelimlerini kontrol edebilirken, temas ile hareketlendirmenin yetersiz olduğu yerlerde reaksiyon tekerlekleri tercih edilen bir yöntemdir. Günümüzde reaksiyon tekerlekleri uzayda serbest yüzen uyduların yönelimini kontrol etmek için yaygın olarak kullanılmaktadır [1], [2]. Yang, reaksiyon tekerlerinin uygulaması gereken momentum miktarını (Saturasyon kontrolü) yönetmek için uydularda da eyleyici olarak kullanılan manyetik tork bobinlerini kullanmıştır [3]. Benzer şekilde, bacaklı robotlarla ilgili bazı çalışmalarda, eksik tahrikli ana gövde üzerinde istenen yönelimleri elde etmek ve dengeli yürümeyi desteklemek için reaksiyon tekerli dinamik modeller oluşturulmuştur [4]. Lee vd. bir quadruped robotun stabilize ve denge kabiliyetini artırmak için reaksiyon tekerinden faydalanmışlardır. Model

\* Sorumlu yazar: [halithulako@hakkari.edu.tr](mailto:halithulako@hakkari.edu.tr). Yazarın ORCID Numarası: <sup>1</sup> 0000-0001-8194-5433.

tahmini kontrol tabanlı kontrolcü uygulanarak normalde 4 bacaklı robotların yapamadığı ince çizgi üzerinde yürüme işlemini başarmışlardır [5]. Bokser ve Manchester tek bacaklı reaksiyon tekerli zıplayan robotun indirgenmiş dinamik modeli, model tahmini kontrol ile toplam ağırlık merkezinin sinüzoidal bir referans yörüngesi takip etmesi istemişlerdir. Reaksiyon tekeri, robot referans yörüngesi takip ederken kontroldeki karmaşıklığı minimize ettiği ve robot dinamiğinin daha iyi kontrol edilebildiği gözlemlenmiştir [6]. Çalışmamızda farklı olarak kütle merkezi için üretilen bir referans yörüngesi taktikten ziyade zıplama sonrasında farklı düşme senaryoları ile eğitilip zıplamadığı durumlarda dengeli duruşu korumaya odaklanmaktadır. Fakat zıplayan robot için uygulanan benzer bir sistem bulunmamaktadır. Zhang ve arkadaşları, reaksiyon tekerleği kullanarak robotik bir balığın yuvarlanma açısını kontrol etmeye çalışmışlardır [7]. Gajamohan ve arkadaşları yön değiştirebilen, zıplayabilen ve köşelerinde hassas açılarla durabilen üç eksenli bir küpü (Cubli) tepki tekerlekleri ile kontrol etmeyi başarmıştır [8], [9]. Benzer şekilde, reaksiyon tekerleri literatürde Unicycle olarak adlandırılan tek tekerlekli mobil araçların denge kontrolünü sağlamak için kullanılmıştır [10], [11]. Ayrıca, Trentin ve arkadaşları ters sarkaç sisteminin denge kontrolünü, çubuğun her iki ucunda reaksiyon tekerlekleri kullanarak modellemiş ve gerçekleştirmiştir [12].

Dinamik sistemler için hareket denklemleri genellikle Lagrange veya Newton-Euler yöntemleri kullanılarak türetilir. Ancak, yüksek serbestlik derecesine sahip karmaşık sistemler için hareket denklemleri yüzlerce satır matematiksel denklem ortaya çıkarabilir. Bu nedenle, denklemleri basitleştirmek için genellikle model indirgeme yöntemleri kullanılır. Yüksek serbestlik derecesine sahip karmaşık sistemlerin dinamik modelini elde etmeye yönelik bir yaklaşımda "Multibody Dynamics" simülasyon programlarının kullanılmasıdır [13]. Bir sistemin dinamik modelini elde etmek için, geleneksel matematiksel işlemler yerine Matlab Simscape MultiBody ile blok yapıları kullanılabilir. Bu, matematiksel bir model türetmeye gerek kalmadan sistemin dinamiklerinin hızlı bir şekilde oluşturulmasına olanak tanır. Ek olarak, Matlab Simscape MultiBody fiziksel bir ortam ve sistemin çalışmasına dair görsel animasyonlar sağlar.



**Şekil 1.** Dinamik modeli elde edilecek olan zıplayan robotun bağlı olduğu mekanizma düzeniği görünümü.

Bu makale, robotun modellenmesi ve dengeli duruş görevinin başarılı bir şekilde uygulanması da dahil olmak üzere birkaç bölüme ayrılmıştır. İkinci bölümde, zıplayan robotun kinematik ve dinamik modellenmesi yapılmış ve MSM ile elde edilen Dinamik Sistem Modeli yapısı verilmiştir. Zemin, yay ve sönümlenme elemanları kullanılarak modellenmiştir. Üçüncü bölüm, uygulanan empedans kontrolü de dahil olmak üzere denge görevinin uygulanmasında izlenen prosedürün ayrıntılarını vermektedir. Ayrıca, yere iniş sırasında devrilmeden dengeyi korumak için uygulanan öğrenme algoritmasının özelliklerini sunmaktadır. Dördüncü bölümde sayısal simülasyonlar sunulmakta ve grafiksel sonuçlar tartışılmaktadır. Beşinci bölümde ise sonuç ve araştırmacılar için çeşitli öneriler sunulmaktadır.

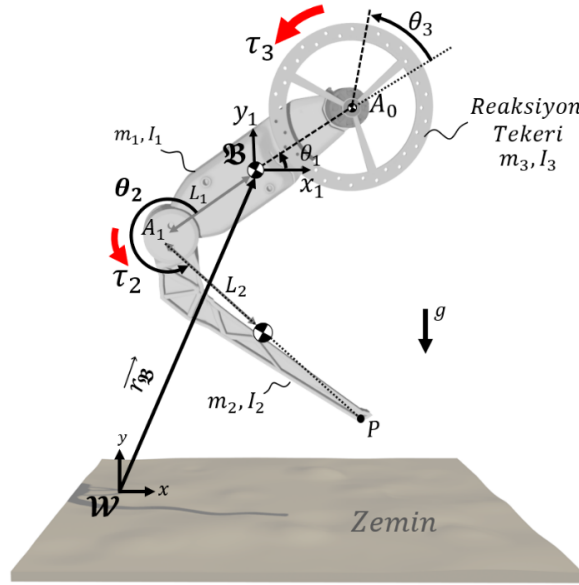
## 2. Kinematik ve Dinamik Modelleme

Şekil 2, reaksiyon tekerli robot modelinin kinematik ve dinamik gösterimlerinin yerleşimlerini göstermektedir. Robot üç rijit gövdeden oluşmaktadır.  $i \in \{1,2,3\}$  uzuv numaraları olmak üzere sırasıyla Ana gövde, bir SD'ye sahip bacak ve Reaksiyon tekerinden oluşmaktadır. Reaksiyon tekerleği ve bacak, döner mafsallar kullanılarak ana gövdenin iki ucuna bağlanmıştır.

Şekil 2'de, modelde  $A_0$  noktası ana gövde ile reaksiyon tekerleğinin bağlantı noktasını,  $A_1$  noktası ise ana gövde ile bacağın bağlantı noktasını temsil ettiğini göstermektedir. Düzlemsel dinamik robot modeli toplam 5 SD'ye sahiptir ve uzayda serbestçe hareket eder. Oluşturulan dinamik düzlemsel robot modeli genelleştirilmiş  $q \in \mathcal{R}^{n+3}$  koordinatlarını kullanmaktadır, burada "n" terimi tahrik edilen eklemleri temsil etmektedir. Genel haliyle, doğrusal olmayan hareket denklemi (Denklem 1) aşağıdaki gibidir:

$$M(q)\ddot{q} + h(q, \dot{q}) = S^T \tau + J^T f_c \quad (1)$$

Eylemsizlik matrisi  $M \in \mathcal{R}^{(n+3) \times (n+3)}$  ile temsil edilir ve  $h \in \mathcal{R}^{n+3}$  ise yerçekimi, Koriolis ve merkezci terimleri içerir. J ifadesi temas noktası için elde edilen Jakobiyen matrisini sağlarken,  $f_c$  robotun bacağı yerle temas halindeyken ortaya çıkan temas kuvvetlerini temsil etmektedir. Eksik tahrikli gövde eksenleri hariç, eklemlerdeki tork değerleri  $\tau \in \mathcal{R}^n$  ile temsil edilir. Seçim matrisi  $S^T$ , n tork değerini seçmek için kullanılan bir boolean matrisidir.

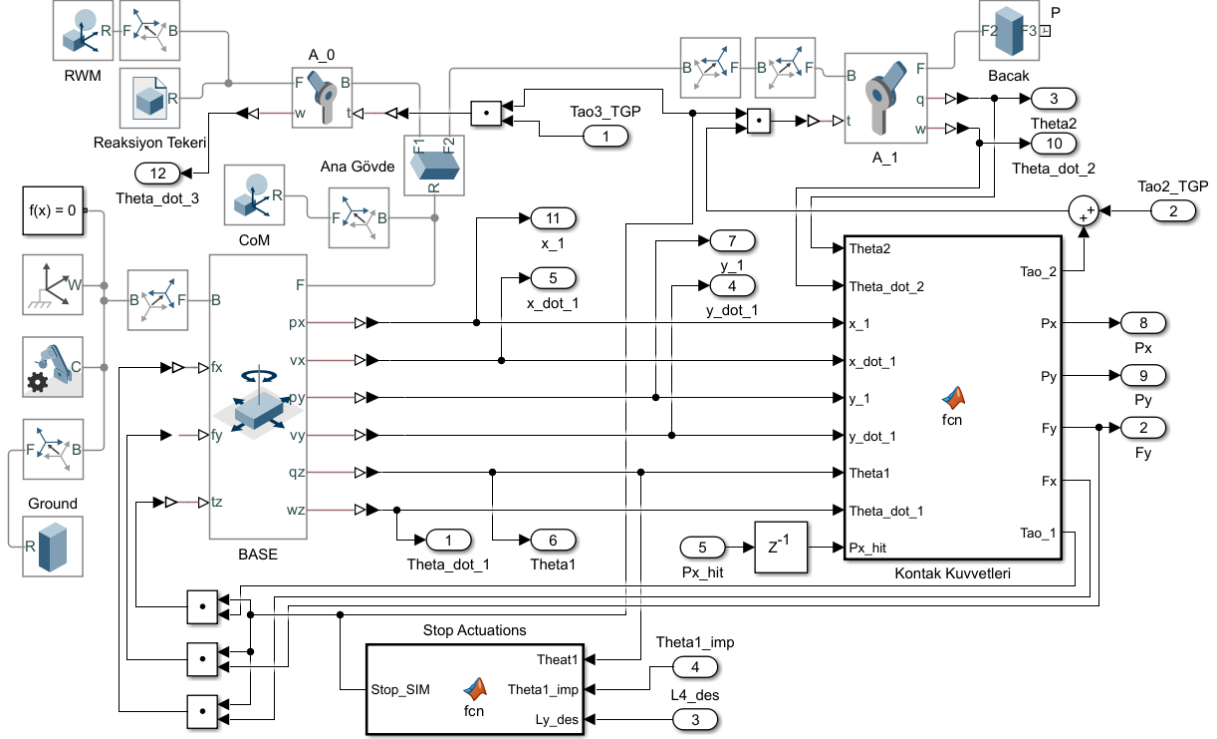


Şekil 2. Robot modeline ait kinematik ve dinamik gösterimler.

$L_1$  ana gövdenin merkezinden  $A_1$  eklemine kadar olan mesafeyi temsil ederken,  $L_2$  bacağın kütle merkezinden bacağın sonuna kadar olan mesafeyi temsil eder.  $P$  noktası bacağın alt kısmında bulunan ve yerle temas eden noktayı temsil etmektedir.  $\mathcal{W}(x, y)$  atalet eksen olarak tanımlanırken, hareketli olan ana eksen  $\mathcal{B}(x_1, y_1)$  robotun ana gövdesinin kütle merkezinde konumlandırılmıştır. Hareketli ana eksen  $\mathcal{B}$ 'nin sabit referans çerçevesine göre konumu  $r_{\mathcal{B}} \in \mathcal{R}^2$  ifadesiyle gösterilir. Ana eksenin açılma yer değiştirmesi  $\theta_1$  ile gösterilir.  $A_0$  ve  $A_1$  eklemlerinin açılma yer değiştirmeleri sırasıyla  $q \in \mathcal{R}^n = (\theta_2, \theta_3)$  ile temsil edilir.  $\theta_1$  yatay x ekseninden ölçülürken,  $\theta_2$  ve  $\theta_3$ ,  $\theta_1$ 'e bağımlı değişkenlerdir. Merkezler için kütle ve eylemsizlik momentleri sırasıyla  $m_i \in \mathcal{R}^3$  ve  $I_i \in \mathcal{R}^3$  ile temsil edilir.  $(x_1, y_1, \theta_1)$  eksik tahrikli gövdenin yön ve konumlarını verir. Eksik tahrikli olan ana eksenleri dolaylı olarak kontrol etmek için  $A_0$  ve  $A_1$  eklemlerine  $\tau \in \mathcal{R}^n$  torkları uygulamak gerekir.

Şekil 2'de gösterilen dinamik robot modeli, hareket denklemleri ile aynı sonuçları üretmek için MSM kullanılarak elde edilmiştir. MSM yapısı,

Şekil 3'te gösterildiği gibi Simulink ve Simscape blok diyagramları kullanılarak modellenmiştir. Robot modellenirken Şekil 1'de verilen robotun bağlı olduğu kızak elemanlarının ağırlıksız olduğu varsayılmıştır. Ayrıca gövde ve bacak, ağırlık merkezleri geometrik merkezlerinde olacak şekilde modellenmiştir.



Şekil 3. MSM ile oluşturulan Robot Dinamiği (RD) alt bloğu.

Katı nesnelere CAD programları gibi katı modelleme yazılımları kullanılarak oluşturulabilir veya doğrudan MSM'nin hazır kütüphanesinden eklenebilir. MSM simülasyon programı temel olarak eklem tiplerini, katı cisim modellerini ve eksenleri istenen noktalara tanımlamak için kullanılan Rigid Transform bloklarını temsil eden birbirine bağlı blok diyagramlarından oluşur. Robotun gövde merkezinde bulunan 3-DOF "B" için Base adında düzlemsel bir mafsallı kullanılmıştır.

Ana gövde üst ucu olan  $A_0$ 'a tek serbestlik dereceli bir döner mafsallı kullanarak reaksiyon tekerine bağlanmıştır, Bacak bloğu tek serbestlik dereceli bir döner mafsallı kullanarak ana gövdenin alt ucu olan  $A_1$ 'e bağlanmıştır. MSM'de,  $A_1$  döner eklemi için yapısal olarak bulunduğu varsayılan  $0,005 \text{ Nm}/(\text{deg}/\text{s})$  değerinde sönümleme miktarı tanımlanmıştır. Bu bileşenler için dinamik parametreler Ana Gövde, Reaksiyon Tekerleri ve Bacak bloklarına girilmiştir. Kontakt Kuvvetleri fonksiyon bloğu, robotun bacağı zeminle temas ettiğinde ortaya çıkan temas kuvvetlerini hesaplar. Ortaya çıkan tork kontrol sinyalleri daha sonra reaksiyon tekerlerine ve bacağına bağlı olan  $A_0$  ve  $A_1$  eklem bloklarına gönderilir. Dinamikleri sağlayan ana yapı tamamlandıktan ve sistem analizi başlatıldıktan sonra, fiziksel ortamda çalışan mekanik modelin animasyonu elde edilmiş olur. MSM'de robotun dinamiğini veren ana yapıya ek olarak kontrolör fonksiyonları eklenerek aynı ortam içerisinde bir geri besleme sistemi oluşturmak mümkündür. Gerçek zamanlı öğrenme denemeleri sırasında kötü ilerleyen dengeleme işlemlerini sonlandırmak için Stop Actuation (SA) bloğu eklenmiştir.

## 2.1 Jakobiyen ve Kontakt Modelleme

Robot bacağına yer ile temas yüzeyinin bir nokta olduğu varsayılmaktadır.  $P = [P_x \ P_y]^T \in \mathbb{R}^2$  noktasının atalet eksenine göre konumu,  $P_x$  ve  $P_y$  değerleri sırasıyla Denklem 2'de verilmiştir.

$$\begin{aligned} P_x &= x_1 - L_1 \cos(\theta_1) + 2L_2 \cos(\theta_1 + \theta_2) \\ P_y &= y_1 - L_1 \sin(\theta_1) + 2L_2 \sin(\theta_1 + \theta_2) \end{aligned} \quad (2)$$

P konum vektörü için elde edilen  $J_P$  Jacobian ifadesi aşağıda verilmiştir (Denklem 3);

$$J_P = \begin{pmatrix} 1 & 0 & L_1 \sin(\theta_1) - 2L_2 \sin(\theta_1 + \theta_2) & -2L_2 \sin(\theta_1 + \theta_2) \\ 0 & 1 & -L_1 \cos(\theta_1) + 2L_2 \cos(\theta_1 + \theta_2) & 2L_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \quad (3)$$

P noktasına etki eden dikey ve yatay kontak kuvvetleri Denklem 4'te verilmiştir.

$$f_c = \begin{cases} [(k_c^x (P_x^{hit} - P_x) - b_c^x \dot{P}_x) \quad (-k_c^y P_y - b_c^y \dot{P}_y)] & \text{if } P_y < 0 \\ [0 \quad 0] & \text{if } P_y \geq 0 \end{cases} \quad (4)$$

P noktasında meydana gelen temas kuvvetleri  $f_c = [f_c^x \quad f_c^y]^T \in \mathcal{R}^2$  ile temsil edilir. Zeminin yatay ve dikey yönlerdeki mekanik sönümlenme ve yay elemanları olarak modellenir. Ayak ucunun zeminle temas anında atalet eksenine göre x eksenindeki konumu  $P_x^{hit}$  ile gösterilir. Zeminin sertliği  $k_c \in \mathcal{R}^2 = (k_c^x \quad k_c^y)$  terimi ile temsil edilirken, zeminin sönümlenme katsayısı  $b_c \in \mathcal{R}^2 = (b_c^x \quad b_c^y)$  ile temsil edilir.  $k_c$  ve  $b_c$  değerleri düşme testleri ile belirlenmiştir.  $\dot{P} \in \mathcal{R}^2$ , ayak ucu noktası P'nin zamana göre birinci türevi Denklem 5'te verilmiştir.

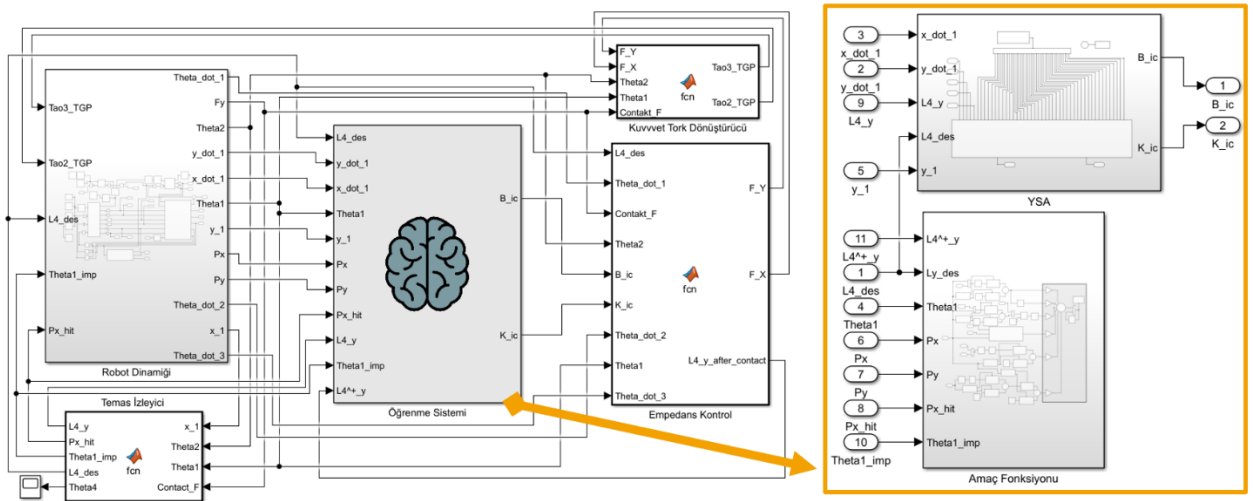
$$\dot{P} = J_P [x_1 \quad y_1 \quad \theta_1 \quad \theta_2]^T \quad (5)$$

P noktası yere temas ediyorsa, kayma olmadığı ve  $\dot{P}(t) = 0$  olduğu varsayılır.

### 3. Dengeleme Prosedürü ve Kontrol Yapısı

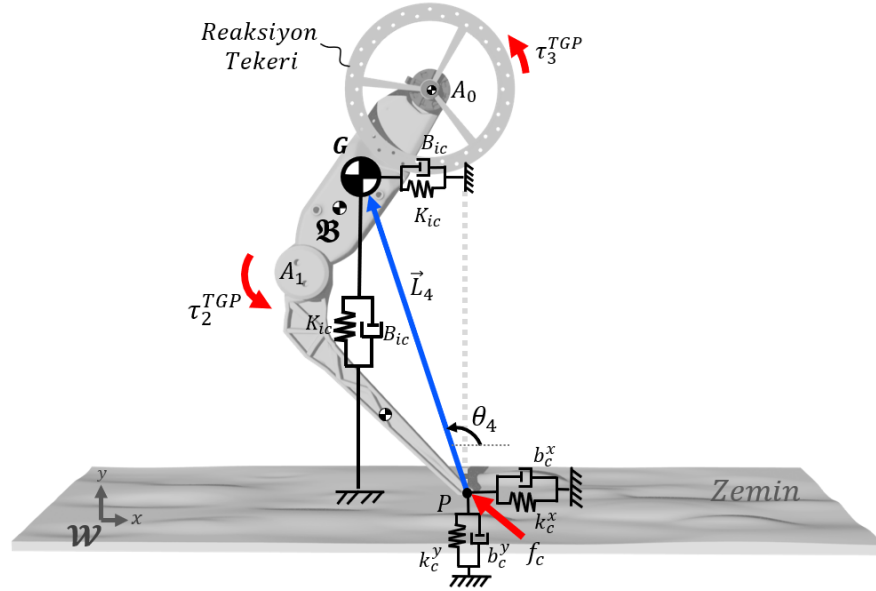
Bu bölümde robotun yere temas ettiği ana odaklanılmaktadır. Robotun çarpma anında devrilmesini önlemek ve dengeli bir duruşu korumak için gerçek zamanlı bir öğrenme yapısı geliştirilmiştir. Yapı, inişin darbe kuvvetlerini etkili bir şekilde emerken aynı zamanda robotun dengesini korumayı öğrenerek, düşmeden dengede kalmasını sağlayacak şekilde tasarlanmıştır. Robot daha sonra nihai konumuna ilerlerken MSM tarafından oluşturulan, robot ve çevresini de içeren öğrenme ve aşamalarının da olduğu tam kontrol yapısı

Şekil 4'te gösterilmektedir. Bu entegre yapı, yerle teması sırasında etkili bir kontrol ve öğrenme sürecini yönetir.



Şekil 4. Robot için MSM ana kontrol mimarisi.

Robot Dinamiği (RD) bloğu, robotu ve çevreyi modelleyen blok mimarisini göstermektedir. Empedans Kontrol (EK) bloğu, uygulanan empedans kontrolüne ait kontrol algoritmasını barındırmaktadır. Öğrenme Sistemi (ÖS) öğrenmeyi gerçekleştirir ve kontrol katsayılarını belirler. ÖS alt blokları sırasıyla YSA ve Amaç Fonksiyonu (AF) olarak adlandırılan Sinir Ağı ve Amaç fonksiyonu için blok yapıları içerir. Temas İzleyici (Tİ) bloğu yere iniş anındaki durum bilgilerini diğer bloklara iletir. RD bloğundan alınan durum gözlemleri diğer blok yapılarını besler. Komple kontrol yapılarından elde edilen aksiyon çıkışları ise RD bloğuna girer.



Şekil 5. Robot yer ile temasıyla başlayan Kartezyen empedans kontrol.

Robotun düşme etkilerini sönümlenmesi, devrilmesini önlemesi ve dengeyi geri kazanması için adaptif empedans kontrolü uygulanmıştır. Bu yaklaşım, robotun dış kuvvetlere tepkisini kontrol etmek, dengeyi sağlamak ve salınımlı özelliklerini gerektiği gibi ayarlamak için çok yönlü bir yöntem sunar. Şekil 5,  $L_4$  vektörünün uzunluğunun ve açısının yatay ve dikey eksenlerde tanımlanan yay ve sönüm elemanları tarafından kontrol edildiğini göstermektedir. Bu kuvvetler robota etki ederek yay sönümlenme hareketini taklit etmesini sağlar [14]. Yayın sertlik değeri ve sönümlenme katsayısı, istenen salınımlı türünü elde etmek için ayarlanabilir. Şekil 5'te gösterilen  $L_4 \in \mathbb{R}^2$  vektörü için kapalı döngü vektör Denklemi 6'daki gibi üstel formda karmaşık sayılarla yazılabilir ve bu da iki 2 adet skaler denklem türetir. Vektörün uzunluğu ve birinci türevi sırasıyla Denklem 7 ve Denklem 8'de,  $\theta_4$  açısı ise Denklem 9'da verilmiştir.

$$\vec{GA}_1 + \vec{A}_1\vec{P} + \vec{PG} = 0 \quad (6)$$

$$\vec{PG} = -L_G e^{i(\theta_1+\pi)} - 2L_2 e^{i(\theta_1+\theta_2)} \quad (7)$$

$$L_4^x(\theta_1, \theta_2) = L_G \cos(\theta_1) - 2L_2 \cos(\theta_1 + \theta_2)$$

$$L_4^y(\theta_1, \theta_2) = L_G \sin(\theta_1) - 2L_2 \sin(\theta_1 + \theta_2)$$

$$V_4 = iL_G \dot{\theta}_1 e^{i(\theta_1)} + i2L_2(\dot{\theta}_1 + \dot{\theta}_2) e^{i(\theta_1+\theta_2)} \quad (8)$$

$$L_4^y = L_G \dot{\theta}_1 \cos(\theta_1) - 2L_2(\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)$$

$$L_4^x = -L_G \dot{\theta}_1 \sin(\theta_1) + 2L_2(\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2)$$

$$\theta_4 = \cos^{-1} \left( \frac{L_G \cos(\theta_1) - 2L_2 \cos(\theta_1 + \theta_2)}{L_4} \right) \quad (9)$$

$L_4^x$  ve  $L_4^y$  sırasıyla yatay ve dikey eksenlerdeki bileşenleri temsil etmektedir. Yer ile temas aşaması sırasında amaç  $\theta_4$  açısını  $\pi/2$  radyanda sabitlemek ve çarpma sırasında  $L_4$  uzunluğunu önceden belirlenmiş bir değerde tutmaktır. Kontrol, bir öğrenme algoritması kullanılarak düşüş denemeleri yoluyla sistemin öğrenilmesiyle

sağlanır. Bu aşamada kararlı bir duruş elde edildiğinde, sistem  $\theta_4$  açısını aynı değerde tutmaya çalışır. Denklem 10 ve 12, empedans kontrolü ile yatay ve dikey eksenlerde uygulanan kuvvet bileşenlerini sağlar.

$$L_4^{xeq} = L_4^{xeq} + \zeta_w \dot{\theta}_3 \quad (10)$$

$$F_{ic}^x = (L_4^{xeq} - L_4^x) K_{ic} - \dot{L}_4^x B_{ic} \quad (11)$$

$$L_4^{des} = \{ L_4^y(t_{imp}) \text{ if, } \text{sign}(f_c) > 0 \quad (11)$$

$$F_{ic}^y = (L_4^{des} - L_4^y) K_{ic} - \dot{L}_4^y B_{ic} \quad (12)$$

Hem yatay hem de dikey yönlerde kullanılan sanal empedans olan  $B_{ic}$  ve  $K_{ic}$  kontrol katsayıları, yapay sinir ağlarından elde edilmiştir. Bu katsayılar, robotun yerden yüksekliği, yatay ve dikey eksenlerdeki hızları gibi rastgele başlangıç koşulları ile YSA tarafından eğitilmiş ve belirlenmiştir. Geçiş aşamasında x ve y yönleri için istenen referans değerler sırasıyla  $L_4^{xeq}$  ve  $L_4^{des}$ 'tir.  $L_4^{xeq}$  ifadesi Denklem 10'daki ilk satırda belirlenmiştir. Reaksiyon tekerinin hız limitlerinin üzerine çıkmasının engellenmesi veya doyum noktasına yaklaşılmasının engellenmesi gerekmektedir [15]. Bu sebeple, yatayda kontrolcünün takip ettiği referans değer için reaksiyon tekerinin açısız hızına bağlı bir yörünge tanımlanmıştır. Burada  $\zeta_w$  ifadesi referansın büyüklüğüne etki eden bir katsayı çarpandır.  $L_4^{des}$  ise Denklem 11'de belirtildiği gibi  $t_{imp}$  çarpma anındaki  $L_4^y$  uzunluğuna eşittir. Sabit çözücü (Solver) aralığı  $t_{int}$  olarak gösterilir.

Ortaya çıkan empedans kuvvetleri, reaksiyon tekerini ve ikinci uzvu tahrik eden tork aksiyon sinyallerine dönüştürülür. Bu sinyaller Denklem 13'te sunulan denklemlerden türetilir.

$$\begin{aligned} \tau_2^{TGP} &= F_{ic}^x L_G \sin(\theta_1) + F_{ic}^y L_G \cos(\theta_1) \\ \tau_3^{TGP} &= F_{ic}^x L_4^y + F_{ic}^y L_4^x \end{aligned} \quad (13)$$

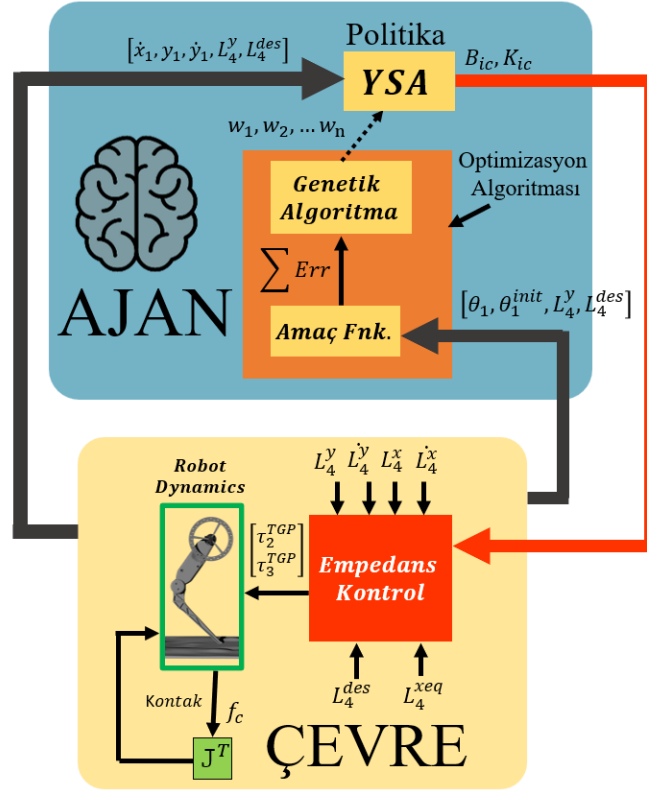
Kontrol sırasında  $\tau_2^{TGP}$  ve  $\tau_3^{TGP}$  sırasıyla A\_1 ve A\_0 bloklarına tork uygular. Empedans kontrol bloğu, YSA bloğundan  $K_{ic}$  ve  $B_{ic}$  çıkış değişkenlerini alır ve bunları Kuvvet-Tork Dönüştürücü (KTD) bloğunda kontrol torklarına dönüştürür.

### 3.1 Öğrenme Algoritması

Bu sistemde, genetik algoritma bir optimizasyon algoritması olarak görev yaparken, yapay sinir ağı politikası bir ajanın davranışını temsil eder. Ajan, çevresiyle etkileşime geçerek görevini öğrenmeye çalışır. Yapay Sinir Ağı veya Politika, ajanın çevresiyle etkileşimde bulunarak elde ettiği gözlem verilerini kullanarak bir eylem üretir. Yapay sinir ağı, genetik algoritma tarafından optimize edilir, böylece ajanın performansını artırmak için en iyi parametre setine evrilir. Bu sistemde genetik algoritma ve yapay sinir ağı birlikte çalışarak, ajanın öğrenmesini ve performansını sürekli olarak iyileştirmesini sağlar. Bu tasarım Evrimsel Pekiştirmeli Öğrenme olarak adlandırılabilir. Bu yöntemde farklı olarak, PÖ'deki Ajan, ödül değerini artırmaya çalışırken, bizim oluşturduğumuz sistemde Ajan, Amaç fonksiyonunu sifra yaklaştırmaya çabalamaktadır. Ayrıca, Ajan' a tüm gözlem verileri girilmesi yerine kritik olan veriler girilmiştir. Ayrıca Politikaya aktör olarak toplam beş girdi, beş ara katman ve iki çıktı hücrelerinden oluşan YSA yapısı kullanılmış ve Kritik için oluşturulan sinir ağı yapısı devre dışı bırakılmıştır. Belirlenen sinir hücresi sayısı toplamda 42 adet ağırlık katsayısı üretmektedir. Buda robotun durum girdilerine karşılık cevap olarak verdiği tepkilerde yeterli miktarda doğrusalsızlık sağlayacağından ağırlık katsayısı sayısı yeterli görülmüştür. Ağırlık katsayıları,  $n \in \{1, \dots, 42\}$  olmak üzere  $w_n$  ile temsil edilir ve Genetik algoritma aracılığıyla her atıştan sonra gerçek zamanlı olarak güncellenir. Hücrelerde kullanılan aktivasyon fonksiyonu "tansig"dir. Eğitim aşamasında, Aktör girişi dinamik modelin  $\dot{x}_1, y_1, \dot{y}_1, L_4^y$  ve  $L_4^{des}$  durum değişkenlerini alır ve çıkışta, düşme etkilerini etkin bir şekilde sönmölemek için  $B_{ic}$  ve  $K_{ic}$  empedans kontrolü için gerekli terimleri sağlar. Bu amaçla kullanılan öğrenme yapısı

Şekil 6'da gösterilmektedir. Simülasyon, YSA ağırlık katsayılarını belirlemek için Matlab Genetik algoritma araç kutusunu kullanır. Sistemimiz için hedeflenen amaç fonksiyonu Denklem 14'te verilmektedir.  $\psi$  terimi, atış denemelerinde kötü sonuçlanan denge kayıpları nedeniyle düşmeyi cezalandırmak için eğitim sırasında eklenir.  $\theta_1^h$  terimi robotun yerle temas ettiği andaki  $\theta_1(t_{imp})$  değerine eşittir ve  $L_4^{des}, L_4^{xeq}$  robotun o andaki dikey ve yatay açıklığını temsil eder. Politika girişine verilen durum değişkenleri değişmiş olsa da çarpışmadan sonra robotun sırasıyla  $L_4^{des}, L_4^{xeq}$  değeri etrafında  $L_4^y$  ve  $L_4^x$ 'i dengelemesi gerekmektedir. Büyük çarpma kuvvetlerine rağmen

yatay yönde maksimum  $\xi$  değerine kadar yer değiştirmeye izin verilmesi istenmiş ve bu sınır aşıldığında Ajanın karar vermesinde etkili olacak büyüklükte cezalandırmalar başlar. İlk terim olan  $\max(0, |L_4^{des} - \xi - L_4^y|^2)$ , maksimum bacak kapanma miktarı olan  $\xi$ 'yi aştığında, yerle temas anındaki çarpma etkisinden dolayı yüksek ceza puanı almaya başlar.  $|L_4^{des} - L_4^y|$  ifadesi robotun denge noktasına dönmeye yardımcı olurken,  $(L_4^{des} - L_4^y)^2$  ifadesi hedef yükseklik ile arasındaki farkı en aza indirir.  $(P_x^{hit} - P_x)^2$  ve  $\lambda_6 P_y^2$  terimleri çarpışmalardan sonra zıplamayı ve yer değiştirmeyi önlemek için eklenmiştir.  $\lambda_j$   $j \in \{1, \dots, 6\}$  ağırlık faktörleridir ve değerleri deneme yanılma yoluyla belirlenmiştir.



Şekil 6. Gerçek zamanlı robot eğitim döngüsü.

$\tau_k^{TGP}$ ,  $k \in \{2, 3\}$  Maksimum kontrol torqu  $\pm u_k^{max}$  ile sınırlıdır. Bu nedenle, Denklem 16'da verilen koşullar uygulanmıştır. Robotun fırlatılması sırasında başlangıç koşullarının kontrol edilebilir sınırlar içinde olmasını sağlamak için, fırlatma simülasyonlarında  $\theta_1$  için başlangıç değerlerinin  $\pi/3 < \theta_1 < 13\pi/18$  aralığında olduğu varsayılmıştır. Buna uygun olarak belirlenen aralıkta bir değer seçilmiştir.  $\dot{y}_1$  için başlangıç değerlerinin;  $\dot{y}_1^{min} < \dot{y}_1 < \dot{y}_1^{max}$  m/s aralığında olduğu varsayılmaktadır.  $y_1$  ve  $\dot{x}_1$  için sırasıyla  $y_1^{min} < y_1 < y_1^{max}$  ve  $\dot{x}_1^{min} < \dot{x}_1 < \dot{x}_1^{max}$  m/s'dir.

$$\text{Err} = \min \sum_{\theta=0}^{\frac{T_{train}-1}{t_{in}}} \left( \lambda_1 \cdot \max(0, |L_4^{des} - \xi - L_4^y|^2) + \lambda_2 |L_4^{des} - L_4^y| + \lambda_3 (L_4^{des} - L_4^y)^2 + \lambda_4 \Psi \right. \quad (14)$$

$$\left. + \lambda_5 (P_x^{hit} - P_x)^2 + \lambda_6 P_y^2 \right) \quad (15)$$

$$\Psi = \begin{cases} (\theta_1 - \theta_1^h)^2, & \text{if } (\theta_1^h < \frac{\pi}{2}) \text{ and } (-\frac{\pi}{5} > \theta_1 > \frac{\pi}{2}) \text{ and } (L_4^{des} > 0) \\ (\theta_1 - \theta_1^h)^2, & \text{if } (\theta_1^h > \frac{\pi}{2}) \text{ and } (-\frac{5\pi}{6} > \theta_1 < \frac{\pi}{2}) \text{ and } (L_4^{des} > 0) \end{cases}$$

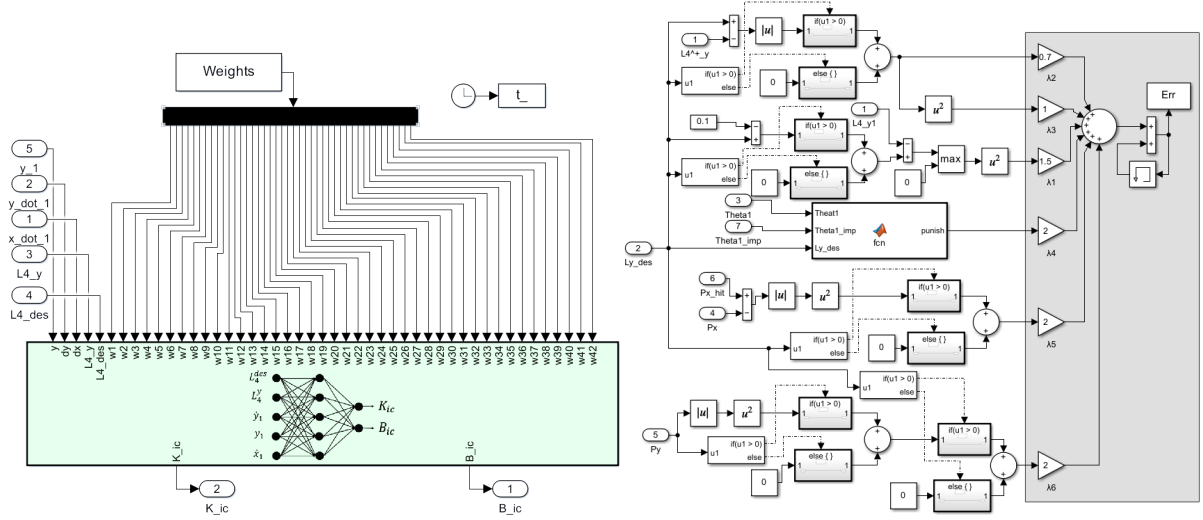


$$s. t. \begin{cases} y_1^{min} \leq y_1 \leq y_1^{max} \\ \dot{y}_1^{min} \leq \dot{y}_1 \leq \dot{y}_1^{max} \\ \dot{x}_1^{min} \leq \dot{x}_1 \leq \dot{x}_1^{max} \\ \tau_k = \begin{cases} u_k^{max} \frac{|\tau_k|}{\tau_k}, & \text{if } |\tau_k| > u_k^{max} \\ \tau_k, & \text{if } |\tau_k| \leq u_k^{max} \end{cases} \end{cases} \quad (16)$$

Simulink blokları ile modellenen AF alt bloğu,

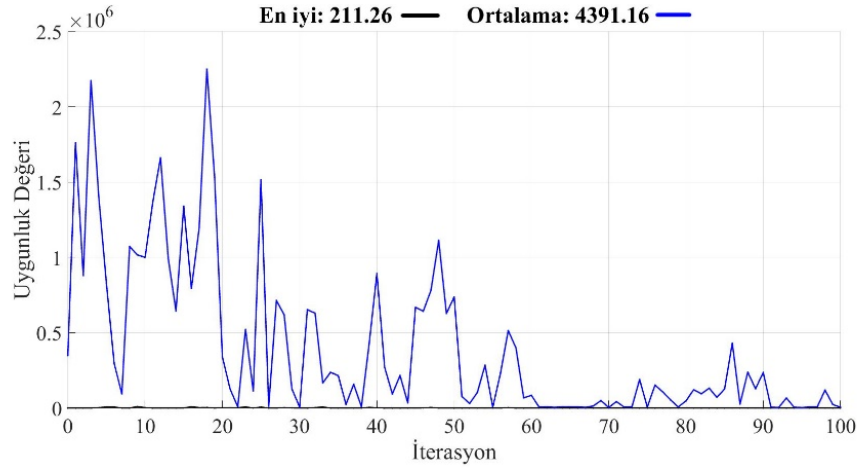
Şekil 7'de (sağda) gösterilmektedir. Her fırlatma denemesinde Genetik Algoritma,

Şekil 7'de (solda) gösterildiği gibi yeni YSA ağırlık katsayılarını test için MS ortamına aktarır. Simülasyon boyunca, AF'den elde edilen toplam hata Err bloğu aracılığıyla Genetik Algoritmaya verilir. Dolayısıyla, her simülasyon sırasında amaç fonksiyonu sıfıra yaklaştıkça, yere konduktan sonraki denge süreci öğrenilir ve bir sonraki simülasyonda daha iyi sonuçlar elde edilmesi için yeni katsayılarla test edilir. Bu gerçek zamanlı öğrenme döngüsü, önceden belirlenen maksimum iterasyon sayısına ulaşılan veya istenen performans aralığına ulaşılan kadar devam eder. Eğitim için seçilen örnekleme zaman aralığı Err bloğuna girilir.



Şekil 7. YSA (Sol), AF (Sağ) alt sistem blok yapıları.

Genetik Algoritma, 100 iterasyon ve 10 popülasyon büyüklüğü ile ayarlanmıştır.  $y_1$ ,  $\dot{y}_1$ ,  $\dot{x}_1$  için simülasyon başlangıç koşulları Denklem 16'da belirtilen aralıkta rastgele seçilmektedir. Eğitim denemeleri boyunca  $\theta_1$  ve  $\theta_2$ 'nin belirlenen bir açıda yere indiği varsayılmıştır. Simülasyon için Çözücü sabit örnekleme süresi  $t_{int} = 1e - 4$  sn olarak ayarlanmış ve Çözücü "ode4 (Runge-Kutta)" olarak seçilmiştir. Eğitim için örnekleme aralığı  $t_{in} = 0,01$  sn olarak seçilmiştir. Her bir eğitim denemesi için toplam simülasyon süresi  $T_{train} = 1,40$  saniyedir. Bu süre sonunda Ajan, toplam hataya göre, Aktör'de bulunan YSA sinir yapısındaki katsayılarla karar vermektedir. Robotun eğitime süreci Matlab 2022 kullanılarak Intel Core i7 işlemci ile donatılmış bir sistem üzerinde yürütülmüştür. Öğrenme sürecini tamamlamak için toplam 1000 adet düşme denemesi gerçekleştirilmiştir. Şekil 8, YSA parametrelerini tahmin ederken performans değerlerini listeleyen Genetik Algoritmayı göstermektedir. Grafikten, yapay zeka ajanının 60. jenerasyondan sonra daha uygun kontrol kararları alabildiği görülebilmektedir.



Şekil 8. İterasyonlar Sırasında Dinamik Sistemin Öğrenme Durumu.

Burada “En İyi” her bir iterasyonda veya nesilde, genetik algoritma tarafından bulunan en iyi çözümün veya en düşük amaç fonksiyon değerini ifade eder. “Ortalama” ifadesi ise her iterasyonda veya nesilde, popülasyondaki tüm çözümlerin bir ortalama değeri, amaç fonksiyonunun ortalaması veya uygunluk değerinin ortalaması şeklinde ifade edilebilir. “Uygunluk değeri”, Her bir bireyin veya çözümün, genellikle bir amaca veya bir uygunluk fonksiyonuna dayanarak ölçülen uygunluk derecesini ifade eder. “İterasyon”, nesil olarak ta ifade edilebilir. Her nesil, bir önceki nesilden türetilen bireylerin evrimleştiği ve yeni bireylerin oluşturulduğu bir aşamadır. En iyi ve ortalama değerler, algoritmanın performansını değerlendirmek için özellikle önemlidir, çünkü bu değerler optimizasyon ilerlemesini ve çözüm kalitesini gösterir.

#### 4. Sayısal Simülasyon

Simulink modeli kullanılarak gerçekleştirilen sayısal simülasyonların sonuçları, dengelemeyi test etmek için geliştirilen kontrol yapısını içermektedir. Robotun kontrolü için gerekli parametreler ve gerçekleştirilen simülasyonlar için başlangıç değerleri Tablo 1 ve 2’de verilmiş olup, bu parametreler Simulink modelindeki ilgili bloklarda tanımlanmıştır.

Tablo 1. Kontrol ve Robot dinamiği parametreleri.

Parametre	Değer	Parametre	Değer	Parametre	Değer
$L_1(m)$	0,2	$\lambda_1$	1,5	$y_1^{min}(m)$	0,2
$L_2(m)$	0,2	$\lambda_2$	0,7	$y_1^{max}(m)$	0,3
$m_1(kg)$	0,63	$\lambda_3$	1	$\dot{y}_1^{min}(m/s)$	0
$m_2(kg)$	$3 \times 10^{-2}$	$\lambda_4$	2	$\dot{y}_1^{max}(m/s)$	0,7
$m_3(kg)$	$14 \times 10^{-2}$	$\lambda_5$	2	$\dot{x}_1^{min}(m/s)$	0
$g(m/s^2)$	9,80665	$\lambda_6$	2	$\dot{x}_1^{max}(m/s)$	0,5
$I_1(kgm^2)$	$2,47917 \times 10^{-3}$	$\xi(m)$	0,1	$u_2^{max}(Nm)$	20
$I_2(kgm^2)$	$1,00563 \times 10^{-4}$	$b_c^y(N/ms^{-1})$	$2 \times 10^2$	$u_3^{max}(Nm)$	15
$I_3(kgm^2)$	$1,5931 \times 10^{-4}$	$t_{ref}(sec.)$	2	$L_4^{xeq}(m)$	0
$k_c^y(N/m)$	$10^5$	$k_c^x(N/m)$	$10^3$	$b_c^x(N/ms^{-1})$	$10^2$

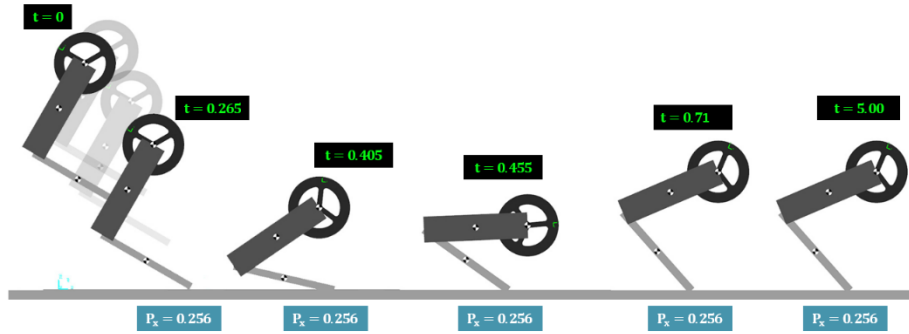
Eğitim sonunda reaksiyon tekerli sisteminin robot üzerindeki performansını değerlendirmek için rastgele başlangıç değerleri seçilip nümerik simülasyon gerçekleştirilmiştir. Fırlatma testi için pozisyon ve hızların başlangıç değerleri Tablo 2’de verilmiştir. Simülasyonun başlangıcında ( $t = 0$ ), reaksiyon tekerleği hareketsizken, robot aşağıda verilen başlangıç açısı ve doğrusal konumlarında ve hızlarında başlatılmıştır.

**Tablo 2.** Simülasyon için başlangıç değerleri.

	$x_1$	$y_1$	$\dot{x}_1$	$\dot{y}_1$	$\theta_1$	$\dot{\theta}_1$	$\theta_2$	$\dot{\theta}_2$	$\theta_3$	$\dot{\theta}_3$
Rastgele Başlangıç Değerleri ( $t = 0$ )	0 m	0,35 m	0,5 m/s	0,7 m/s	1,047 rad	0 rad/s	4,71 rad	0 rad/s	0 rad	0 rad/s

Simülasyonlar sırasında maksimum kontrol torkları  $\tau_3$  ve  $\tau_2$  denklem (16)'da verildiği gibi kısıtlanmıştır. Robotun fırlatılması ve dengelenmesi sırasında alınan simülasyon ekran görüntüsü

Şekil 9'da verilmiştir. Yine fırlatılma testinde robotun MSM modelinin çıktılarının 5 saniyelik test grafikleri Şekil 10-18'de sunulmuştur. Grafiklerde MSMB modelinde ihmal edilebilir farklarla kontrolün sağlandığı ve robotun zıplamadığı süre boyunca istikrarlı bir şekilde dengede durduğu görülebilmektedir.



**Şekil 9.** MSM'in ürettiği animasyondan alınan, robot fırlatılma ve dengelenmesine kadar olan süreye ait ekran görüntüleri.

Robot düşüşe geçtiğinde  $x_1$  ve  $y_1$ 'in doğrusal konum ve hız bileşenleri sırasıyla

Şekil 10 ve

Şekil 11'de verilmiştir.

Şekil 11'de  $y_1$  değerinin 1 saniyelik süre içinde 0,19 metrede sabitlendiği görülmektedir.

Şekil 12 ve 15'te  $\theta_1$  ve  $\theta_2$  değerinin aynı süre içinde dengelendiği ayrıca  $\dot{\theta}_1$  ve  $\dot{\theta}_2$ 'nin 0 rad/s'de sabitlenmiştir.

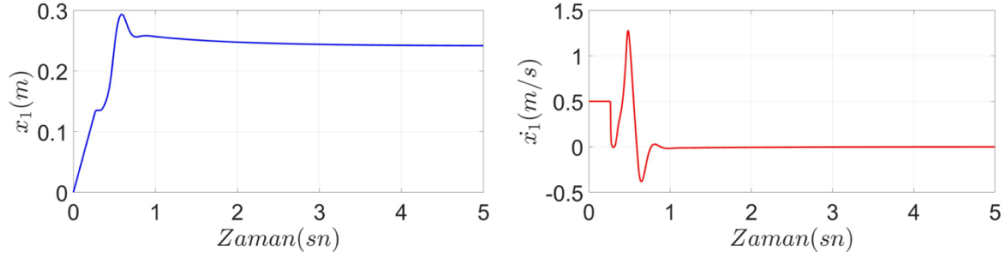
Şekil 14'te (solda), robot topuk ucunun yer ile teması başladığında  $L_4^x$  değerindeki farkın 3 saniye içinde yakınlığı ve sonsuzda kapandığı görülebilmektedir.

Şekil 14'te (sağda), robotun yere düşme etkisiyle aldığı darbeye bağlı olarak  $L_4^y$  değerindeki fark hemen kapanmakta ve  $L_4^{des}$  değerine yaklaşmaktadır. Burada dengelenme işlemine odaklanıldığından dikey bileşende referans etrafında oluşan yaklaşık 2 cm'lik fark uygun görülmüştür.

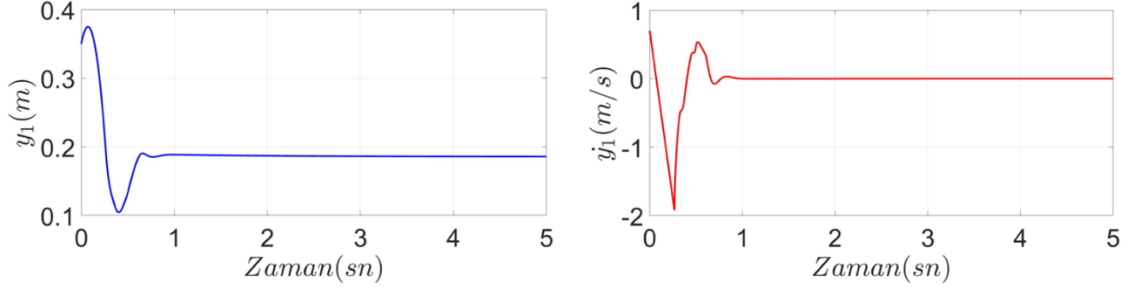
Şekil 15'te (sağda), robot gövdesinin istenen pozisyonda dengesini koruyabilmesi için reaksiyon tekerleğine uygulanan tork eylem sinyali görülmektedir.

Şekil 15'te görüldüğü gibi, darbeye rağmen  $\tau_2$  ve  $\tau_3$  kontrol torkları robotu istenen referansta dengelemeyi başarmıştır. Şekil 16'da  $\theta_4$ , 3 saniye süre içinde referans değere oldukça yaklaşmış ve istenen  $\pi/2$  radyan referans değerine sonsuzda sabitlendiğini göstermektedir. Ayrıca,

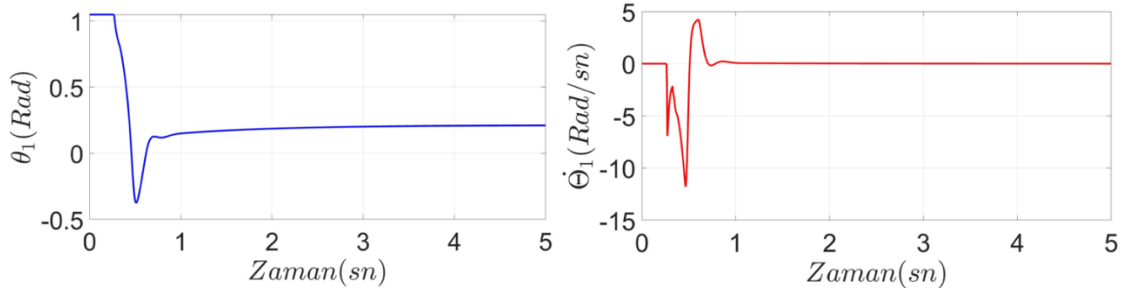
Şekil 17'de denge sağlandıktan sonra reaksiyon tekerinin açısal hızı doyum noktasından uzaklaşmış buna karşılık reaksiyon tekerine verilen  $\tau_3$  kontrol torku sıfırlanmıştır. Aynı eksene hızlarda iken farklı yüksekliklerden fırlatılan robotun dengelenme performansına ilişkin simülasyon sonuçları Şekil 18'de sunulmuştur. Gövde merkezi maksimum yükseklik olan  $y_1 = 0,5$  metrede fırlatıldığında, robot dengesini koruyabilmiştir. Ancak bu yüksekliğin üzerine çıkıldığında, devrilme başlamıştır.



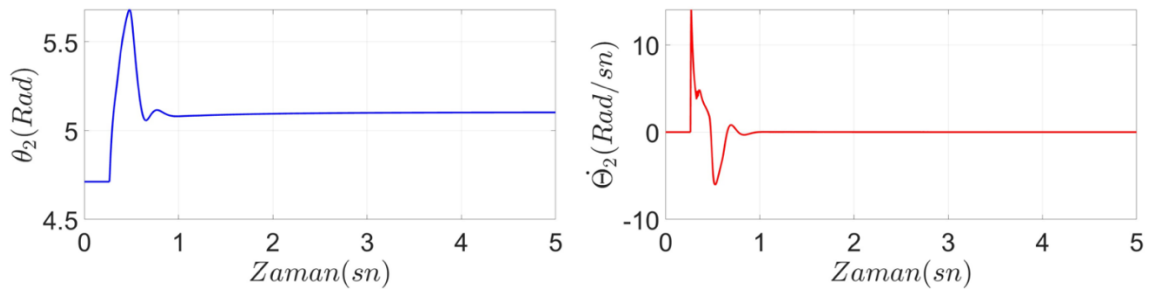
Şekil 10. Fırlatılma esnasında  $x_1$  konumu (solda) ve  $\dot{x}_1$  hızındaki (sağda) 5 saniyelik değişim grafikleri.



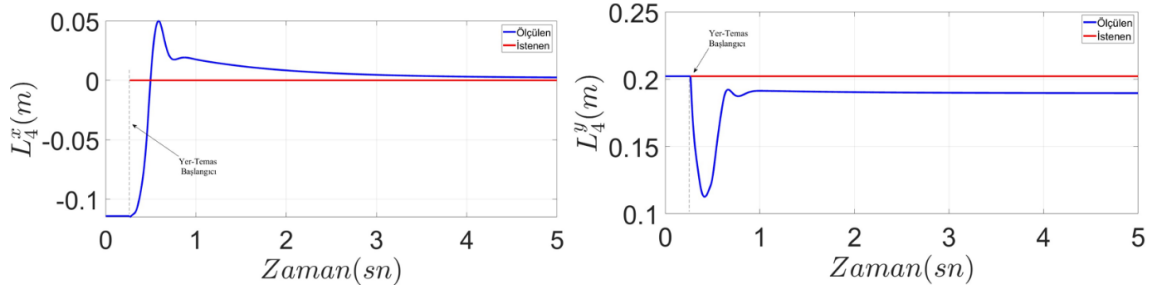
Şekil 11. Fırlatılma esnasında  $y_1$  konumu (solda) ve  $\dot{y}_1$  hızındaki (sağda) 5 saniyelik değişim grafikleri.



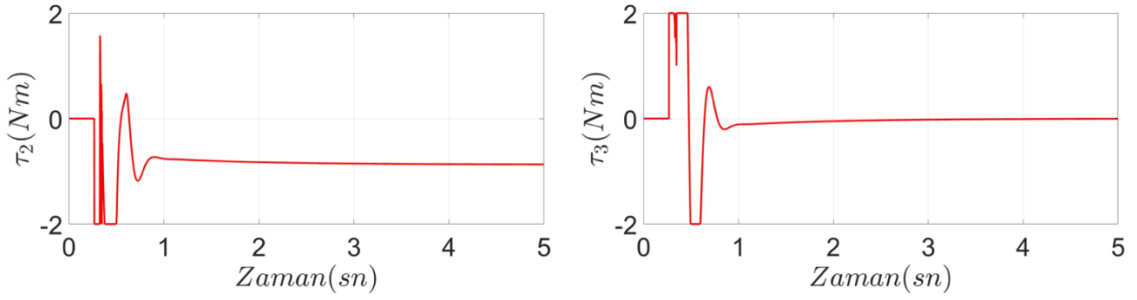
Şekil 12. Fırlatılma esnasında  $\theta_1$  açısai konumu (solda) ve  $\dot{\theta}_1$  açısai hızındaki (sağda) 5 saniyelik değişim grafikleri.



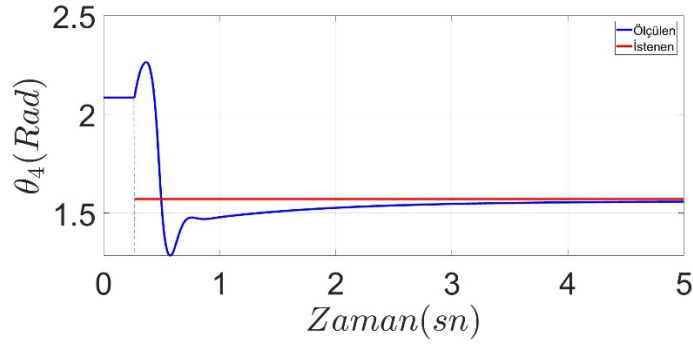
Şekil 13. Fırlatılma esnasında  $\theta_2$ 'in açısai konumu (solda) ve  $\dot{\theta}_2$  açısai hızındaki (sağda) 5 saniyelik değişim grafikleri.



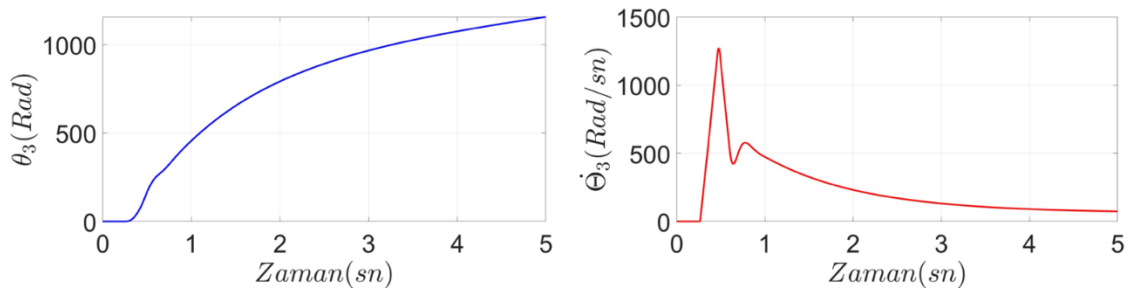
Şekil 14. Simülasyon neticesinde  $L_4^x$  (solda) ve  $L_4^y$ 'deki (sağda) 5 saniyelik değişim grafikleri.



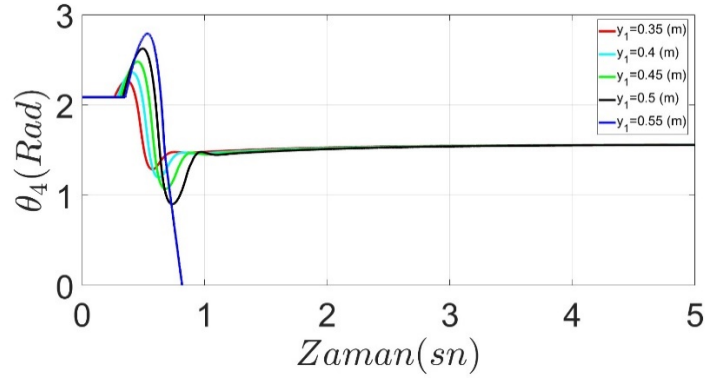
Şekil 15. Simülasyon neticesinde  $\tau_2$  ve  $\tau_3$ 'teki 5 saniyelik değişim grafikleri.



Şekil 16. Simülasyon neticesinde  $\theta_4$ 'teki 5 saniyelik değişim grafikleri.



Şekil 17. Fırlatılma esnasında  $\theta_3$  açısai konumu (solda) ve  $\dot{\theta}_3$  açısai hızındaki (sağda) 5 saniyelik değişim grafikleri.



Şekil 18. Farklı yüksekliklerden fırlatılan robotun denge durumlarını gösteren 5 saniyelik değişim grafiği.

## 5. Sonuç ve Tartışma

Bu çalışmada zıplayan robot, noktasal kontak oluşturacak şekilde modellenmesine rağmen zıpladıktan sonra yerde bulunduğu sürece dengede kalmayı öğrenerek dengesini koruyabilmiştir. Grafikselsel sonuçlar robotun 5 saniye boyunca devrilmediğini ve dengelenmede bir kriter olan kontak noktası ve ağırlık merkezi çizgisine ait açı olan  $\theta_4$ 'ün de  $\pi/2$  radyan değerine 1 saniyelik süre içinde sabitlendiğini göstermektedir. Düzlemsel hareket, simülasyonda etkili bir şekilde kontrol edilmiş olsa da gerçek bir uygulamada dikkate alınması gereken çok sayıda parametre olacaktır. 3D uygulamada gelecek ek ağırlıkların kontrol edilebilirlik açısından maliyetleri olacaktır. Ek kütlelerin neden olduğu sorun, daha güçlü eyleyiciler kullanılarak çözülebilir ve devrilmeden istenen kurulum sağlanabilir. Bir diğer potansiyel çözüm olarak dengelenme görevi için eksik tahrikli gövde eksenlerini aktif olarak hareketlendirmek için fırçasız motorların ve pervanelerin kullanılması kontrolü kolaylaştırabilir. Bu çalışma için, Ana gövde eksenlerinde Reaksiyon tekerleri yerine fırçasız motorlar ve pervanelerin aktüatör olarak kullanılmasıyla iki eksenin sürülmesi, dengede kalma görevinin yerine getirilmesinde oldukça etkili olacaktır. Kısa sürede yüksek itme güçleri elde edilebileceğinden, istenen kontrol daha kolay sağlanabilecektir. Ayrıca, gövde eksenleri boyunca doğrusal hareketlere izin verilmesi, zıplanan yönde konum kontrolüne olanak sağlayacaktır.

## Kaynaklar

- [1] Rui C, Kolmanovsky IV, McClamroch NH. Nonlinear attitude and shape control of spacecraft with articulated appendages and reaction wheels. IEEE Trans Automat Contr 2000; 45(8): 1455-1469.
- [2] Shengmin G, Hao C. A comparative design of satellite attitude control system with reaction wheel. In: Proceedings - First NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2006; pp. 359-362.
- [3] Yang Y. Spacecraft Attitude and Reaction Wheel Desaturation Combined Control Method. IEEE Trans Aerosp Electron Syst 2017; 53(1): 286-295.
- [4] Brown TL, Schmedeler JP. Reaction Wheel Actuation for Improving Planar Biped Walking Efficiency. IEEE Transactions on Robotics 2016; 32(5): 1290-1297.
- [5] Lee CY, Yang S, Bokser B, Manchester Z. Enhanced Balance for Legged Robots Using Reaction Wheels. In: 2023 IEEE International Conference on Robotics and Automation (ICRA); 2023; pp. 9980-9987.
- [6] Graduate BB, Assistant R, Manchester Z. Rex Hopper: Design And Control Of A Monopod Hopper With Reaction Wheels.
- [7] Zhang P, Wu Z, Dong H, Tan M, Yu J. Reaction-Wheel-Based Roll Stabilization for a Robotic Fish Using Neural Network Sliding Mode Control. IEEE/ASME Transactions on Mechatronics 2020; 25(4): 1904-1911.
- [8] Gajamohan M, Merz M, Thommen I, D'Andrea R. The Cubli: A cube that can jump up and balance. In: IEEE International Conference on Intelligent Robots and Systems; 2012; pp. 3722-3727.
- [9] Muehlebach M, Mohanarajah G, D'Andrea R. Nonlinear analysis and control of a reaction wheel-based 3D inverted pendulum. In: Proceedings of the IEEE Conference on Decision and Control, Institute of Electrical and Electronics Engineers Inc.; 2013; pp. 1283-1288.
- [10] Brown TL, Schmedeler JP. Reaction Wheel Actuation for Improving Planar Biped Walking Efficiency. IEEE Transactions on Robotics 2016; 32(5): 1290-1297.
- [11] Han SI, Lee JM. Balancing and velocity control of a unicycle robot based on the dynamic model. IEEE Transactions on Industrial Electronics 2015; 62(1): 405-413.
- [12] Trentin J, da Silva S, Ribeiro J, Schaub H. Inverted Pendulum Nonlinear Controllers Using Two Reaction Wheels: Design and Implementation. IEEE Access 2020; PP: 1.
- [13] Hülako H, Yakut O. Control of Three-Axis Manipulator Placed on Heavy-Duty Pentapod Robot. Simul Model Pract Theory 2021; 108: 102264.

- [14] Zhang S, Zhang H, Fu Y. Leg Locomotion Adaption for Quadruped Robots with Ground Compliance Estimation. *Appl Bionics Biomech* 2020; 2020: 8854411.
- [15] Belascuen G, Aguilar N. Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum. In: 2018 IEEE Biennial Congress of Argentina (ARGENCON); 2018; pp. 1-9.