



**GERÇEK ZAMANLI ENDÜSTRİYEL KONTROL
SİSTEMLERİ İÇİN MAKİNE ÖĞRENMESİ TEMELLİ
YAKLAŞIMLAR**

Süleyman MANTAR



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**GERÇEK ZAMANLI ENDÜSTRİYEL KONTROL SİSTEMLERİ İÇİN
MAKİNE ÖĞRENMESİ TEMELLİ YAKLAŞIMLAR**

Süleyman MANTAR
0000-0002-0552-6066

Doç. Dr. Ersen YILMAZ
(Danışman)

DOKTORA TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2024
Her Hakkı Saklıdır

ÖZET

Doktora Tezi

GERÇEK ZAMANLI ENDÜSTRİYEL KONTROL SİSTEMLERİ İÇİN MAKİNE ÖĞRENMESİ TEMELLİ YAKLAŞIMLAR

Süleyman MANTAR

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ersen YILMAZ

Endüstriyel sistemlerdeki hızlı gelişmeler, üretim süreçlerini daha karmaşık hale getirmiş ve verimlilik beklentilerini artırmıştır. Geleneksel kontrol yöntemleri, bu karmaşıklığı ve belirsizlikleri yeterince ifade edememektedir. Bu nedenle, verimlilik taleplerini karşılayacak ve karmaşık sistemlerin kontrolünü iyileştirecek gelişmiş kontrol sistemleri üzerine çalışmalar sürdürülmektedir. Makine öğrenmesi temelli yöntemler, bu zorlukların üstesinden gelmek için umut verici bir çözüm olarak öne çıkmaktadır. Yapay sinir ağları ile derin öğrenme (DNN) öne çıkan yöntemler arasında yer alsa da, kritik kontrol sistemlerinin standartlara uygunluk sertifikasyonundaki zorluklar nedeniyle alternatif fonksiyon yaklaşımı yöntemlerinin araştırılması önemini korumaktadır.

Bu tez çalışmasında, laboratuvar inkübatör cihazının ekonomik maliyet fonksiyonu ve çoklu kısıtlar dikkate alınarak, model öngörülü kontrol (MPC) tabanlı pekiştirmeli öğrenme (RL) algoritması ile kontrol edilmesi amaçlanmıştır. MPC sisteminin, yanlış bir model kullanılarak tasarlanırsa bile, makine öğrenmesi temelli yöntemler ile gerçek sistemin optimal kontrol politikasını sağlayacak şekilde ayarlanabileceği gösterilmiştir. Pekiştirmeli öğrenme sisteminin değer fonksiyonları, fonksiyon yaklaşımıcısı olarak doğrusal olmayan model öngörülü kontrol (NMPC) sistemi kullanılarak ifade edilmiştir. Aşama maliyeti, terminal maliyeti, kısıtlar ve inkübatör cihazının tahmini sistem modeli, MPC sistemi içerisinde parametrik olarak tanımlanmıştır. MPC sistemi, doğrudan çoklu atış yöntemi ile doğrusal olmayan programlama (NLP) problemi olarak ifade edilmiştir. Elde edilen optimizasyon problemi iç nokta yöntemi ve otomatik türev alma yazılımı ile çözülmüştür. MPC sisteminin parametreleri, en küçük kareler zamansal fark (LSTD) Q-öğrenme yöntemi ile veriye dayalı olarak ayarlanmıştır. MPC sisteminin tasarımında ve benzetim ortamında kullanılmak üzere, inkübatör cihazının matematiksel modeli sistem tanımlama yöntemi kullanılarak elde edilmiştir. Modelleme için Box-Jenkins model yapısı kullanılmıştır ve parametreleri, tahmin hatası yöntemi (PEM) kullanılarak tahmin edilmiştir. Elde edilen sonuçlar, MPC tabanlı RL yönteminin, endüstriyel kontrol sistemlerinde sistem belirsizlikleri ve kısıtlamaları içeren zorlu durumlarda, çevrimiçi öğrenme yeteneği ile sistem performansını optimize ederek referans takip performansını koruyabilen güçlü ve etkili bir çözüm sunduğunu göstermektedir.

Anahtar Kelimeler: Sistem tanımlama, pekiştirmeli öğrenme, model öngörülü kontrol.
2024, xi + 120 sayfa.

ABSTRACT

PhD Thesis

MACHINE LEARNING APPROACHES FOR REAL-TIME INDUSTRIAL CONTROL SYSTEMS

Süleyman MANTAR

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Electronics Engineering

Supervisor: Assoc. Prof. Dr. Ersen YILMAZ

Rapid advancements in industrial systems have made production processes more complex and increased expectations for efficiency. Traditional control methods are insufficient in adequately representing this complexity and uncertainty. Therefore, research on advanced control systems that can meet efficiency demands and improve the control of complex systems is ongoing. Machine learning-based methods have emerged as a promising solution to overcome these challenges. Although artificial neural networks with deep learning (DNN) are among the prominent methods, the challenges in certifying critical control systems for compliance with standards have maintained the relevance of researching alternative function approximation methods.

In this thesis, it is aimed to control a laboratory incubator device using a model predictive control (MPC)-based reinforcement learning (RL) algorithm, considering the economic cost function and multiple constraints. It has been shown that even if the MPC system is designed with an incorrect model, machine learning-based methods can be adjusted to provide the optimal control policy for the real system. Value functions of the reinforcement learning system are expressed using a nonlinear model predictive control (NMPC) system as the function approximation. Stage cost, terminal cost, constraints, and the estimated system model of the incubator device are parameterized with the MPC system. MPC system is expressed as a nonlinear programming (NLP) problem using the direct multiple shooting method. Optimization problem is solved using an interior-point method and automatic differentiation software. Parameters of the MPC system are adjusted using data-driven least-squares temporal difference (LSTD) Q-learning method. Mathematical model of the incubator device, used for designing the MPC system and in the simulation environment, is obtained through a system identification method. Box-Jenkins model structure is used for modeling, and its parameters are estimated using the prediction error method (PEM). Results show that the MPC-based RL method provides a robust and effective solution capable of optimizing system performance with online learning capability, maintaining reference tracking performance in challenging situations involving system uncertainties and constraints in industrial control systems.

Key words: System identification, reinforcement learning, model predictive control.
2024, xi + 120 pages.

TEŞEKKÜR

Doktora çalışmam sırasında bilgisiyle her zaman yanımda olan, göstermiş olduğu yol ve vermiş olduğu cesaret nedeniyle danışman hocam Sayın Doç. Dr. Ersen YILMAZ’a teşekkürlerimi sunarım. Öğrenim sürecim boyunca bilgisini ve tecrübesini esirgemeyen hocam Sayın Prof. Dr. Güneş YILMAZ’a teşekkürlerimi sunarım.

Tez çalışmam sırasında sağladıkları değerli katkılardan dolayı Tez İzleme Komitesinin değerli üyesi Sayın Prof. Dr. Cemal HANILÇI’ye, tez savunma sınavına katılan değerli jüri üyeleri hocam Sayın Prof. Dr. Erdoğan DİLAVEROĞLU ve Sayın Prof. Dr. Hakan GÜRKAN’a teşekkürlerimi sunarım.

Akademik ve çalışma hayatım süresince her zaman destekleriyle yanımda olan EMKO Elektronik A.Ş. Yönetim Kurulu Başkanı Sayın Ayhan ISPALAR’a teşekkürlerimi sunarım. Başta Sayın Erkan EGELİ olmak üzere doktora tezim boyunca bana destek olan EMKO ELEKTRONİK A.Ş. Ar-Ge Merkezi’ndeki çalışma arkadaşlarıma ayrıca teşekkürlerimi sunarım.

Ve, gösterdikleri sonsuz sabır ile beni destekleyen sevgili eşim Duygu, kızlarım Esin ve Derin’e; hayatım boyunca her zaman destekleriyle yanımda olan aileme en içten teşekkürlerimi sunarım.

Bu tez çalışması TÜBİTAK BİDEB 2244 Sanayi-Doktora Programı’nın 118C157 numaralı “Endüstriyel Gerçek Zamanlı Akıllı Kontrol Sistemlerinin Geliştirilmesi” projesi kapsamında tamamlanmıştır. Proje kapsamında sanayi kuruluşu olarak yer alan EMKO ELEKTRONİK A.Ş.’nin Yönetim Kurulu’na teşekkür ederim. Projeye olan desteklerinden dolayı TÜBİTAK’a teşekkürlerimi sunarım.

Süleyman MANTAR

01/11/2024

İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	xi
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	4
3. MATERYAL ve YÖNTEM.....	11
3.1. Pekiştirmeli Öğrenme.....	11
3.1.1. Markov Karar Süreci.....	12
3.1.2. Yaklaşık Değer Fonksiyonu	18
3.1.3. Dinamik Programlama	19
3.1.4. Zamansal Fark Öğrenme	24
3.1.5. En Küçük Kareler Zamansal Fark (LSTD) Algoritması	26
3.1.6. Politika Gradyanı Yöntemleri	28
3.2. MPC Tabanlı Pekiştirmeli Öğrenme.....	29
3.2.1. MPC Fonksiyon Yaklaşımcısı.....	32
3.2.2. İkinci-Dereceden LSTD Q-Öğrenme Algoritması.....	34
3.2.3. LTI İnkübatör Modeli	37
3.2.4. MPC Sistemi	38
3.3. Model Öngörülü Kontrol.....	39
3.3.1. Doğrudan Çoklu Atış Yöntemi	46
3.3.2. Doğrusal Olmayan Optimizasyon	48
3.3.3. Karush-Kuhn-Tucker Koşulları	51
3.3.4. Lagrange Çarpanları Yöntemi.....	53
3.3.5. İç Nokta Yöntemleri.....	57
3.3.6. Sistem Modeli	59
3.4. Sistem Tanımlama.....	60
3.4.1. Sistem Tanımı	62
3.4.2. Öngörü Yöntemi.....	65
3.4.3. Öngörücü Filtre Modeli.....	67
3.4.4. LTI Sistem Yapısı	68
3.4.5. ARX Model Yapısı	69
3.4.6. ARMAX Model Yapısı	70
3.4.7. Çıkış-Hatası (OE) Model Yapısı.....	71
3.4.8. Box-Jenkins Model Yapısı.....	72
3.4.9. Genel Model Yapısı	73
3.4.10. Parametre Tahmin Yöntemi	74
3.4.11. Tahmin Hatası Yöntemi	75
3.4.12. Parametre Tahmin Vektörü.....	77
3.4.13. Kriter Fonksiyonunun Sayısal Minimizasyonu.....	77
3.4.14. Başarımın Değerlendirilmesi	78
3.4.15. Parametre Vektörü Tahmin Sistemi	79
3.4.16. Laboratuvar İnkübatör Cihazı	80
3.4.17. Veri Toplama Sistemi	82

3.4.18. Veri Ön İşleme	83
4. BULGULAR ve TARTIŞMA.....	85
4.1. Sistem Tanımlama Bulguları.....	85
4.1.1. ARX Modeli için Elde Edilen Bulgular	86
4.1.2. ARMAX Modeli için Elde Edilen Bulgular.....	89
4.1.3. Çıkış Hatası Modeli için Elde Edilen Bulgular	92
4.1.4. Box-Jenkins Modeli için Elde Edilen Bulgular	95
4.1.5. Model Yapılarının Başarımları	99
4.2. MPC Tabanlı Pekiştirmeli Öğrenme Kontrol Sisteminin Başarımı.....	102
4.2.1. İnkübatör Modelinin Durum Değerleri	102
4.2.2. MPC Sistemi Tarafından Uygulanan Giriş Sinyali	104
4.2.3. Zamansal Fark Hatası ve Adım Maliyeti	105
4.2.4. b , f ve V_0 Parametreleri	107
4.2.5. A ve B Parametreleri.....	110
5. SONUÇ	113
KAYNAKLAR	115
ÖZGEÇMİŞ	120

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

D_M	Parametre vektörünün bir model yapısında değiştiği değerler kümesi
$e(t)$	t anındaki bozucu etki
$G(q)$	Giriş u ve çıkış y arasındaki transfer fonksiyonu
$G(q, \theta)$	Parametre değer vektör yapısındaki transfer fonksiyonu
$H(q)$	Giriş e ve çıkış y arasındaki transfer fonksiyonu
M	Model yapısı
q	İleri öteleme operatörü
q^{-1}	Geri öteleme operatörü
$V_N(\theta, Z^N)$	Minimize edilecek kriter fonksiyonu
$u(t)$	t anındaki giriş değeri
$y(t)$	t anındaki çıkış değeri
$x(t)$	t anındaki durum vektörü
$\varepsilon(t, \theta)$	Tahmin hatası
θ	Model parametre vektörü
$\varphi(t)$	Regresyon vektörü
$\doteq$	Tanım gereği doğru olan eşitlik ilişkisi
α, β	Adım-boyu parametreleri
γ	İndirgeme katsayısı
s, s'	Mevcut durum, sonraki durum
a	Eylem (davranış, karar)
r	Ödül
$\mathcal{S}$	Sonlu olmayan tüm durumların kümesi
$\mathcal{S}^+$	Son durum dahil tüm durumların kümesi
$\mathcal{A}(s)$	s durumunda mevcut tüm eylemlerin kümesi
$\mathcal{R}$	Tüm olası ödüllerin kümesi, $\mathcal{R} \subset \mathbb{R}$
A_t	t anındaki eylem
S_t	t anındaki durum
R_t	t anındaki ödül
π	Politika (karar verme kuralı)
$\pi(s)$	Deterministik π politikası izlenirken s durumunda uygulanan eylem
$\pi(a s)$	Stokastik π politikası izlenirken s durumunda a eyleminin uygulanma olasılığı
π_θ	θ parametresi ile ilgili politika
$J(\theta)$	θ parametre vektörü için performans ölçütü
$\nabla J(\theta)$	$J(\theta)$ 'nın θ 'ya göre kısmi türevlerinin sütun vektörü
δ_t	t anındaki zamansal fark hatası
V_0	Başlangıç maliyeti
f	Adım (aşama) maliyeti gradyanı parametresi

Kısaltmalar

A3C	Asynchronous Advantage Actor-Critic
ANN	Artificial Neural Network

ARMA	Autoregressive Moving Average
ARMAX	Autoregressive Moving Average with Exogenous Inputs
ARX	Autoregressive with Exogenous Inputs
BJ	Box-Jenkins
DDPG	Deep Deterministic Policy Gradient
DIN	Deutsches Institut für Normung
DNN	Deep Neural Network
DP	Dynamic Programming
DPG	Deterministic Policy Gradient
DQN	Deep Q-Network
EMPC	Economic Model Predictive Control
EN	European Norm
ENMPC	Economic Nonlinear Model Predictive Control
G	Goodness
HVAC	Heating, Ventilation, Air Conditioning
HW	Hammerstein-Wiener
İHA	İnsansız Hava Aracı
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LQR	Linear Quadratic Regulator
LS	Least Squares
LSTD	Least Squares Temporal Difference
LSTDQ	Least Squares Temporal Difference Q-Learning
LTI	Linear Time-Invariant
MAE	Mean Absolute Error
MC	Monte Carlo
MDP	Markov Decision Process
MPC	Model Predictive Control
MSE	Mean Squared Error
NAF	Normalized Advantage Function
NLP	Nonlinear Programming
NLS	Nonlinear Least Squares
NMPC	Nonlinear Model Predictive Control
OCP	Optimal Control Problem
OE	Output Error
PEM	Prediction Error Method
PID	Proportional-Integral-Derivative
PPO	Proximal Policy Optimization
PRBS	Pseudo-Random Binary Sequence
PWM	Pulse Width Modulation
QP	Quadratic Programming
RL	Reinforcement Learning
RTO	Real Time Optimization
SAC	Soft Actor-Critic
SARSA	State-Action-Reward-State-Action
SISO	Single-Input Single-Output
SQP	Sequential Quadratic Programming
TD	Temporal Difference

TD3	Twin Delayed Deep Deterministic Policy Gradient
TF	Transfer Function
TRPO	Trust Region Policy Optimization



ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1.	Pekiştirmeli öğrenmenin tipik blok şeması (Sutton ve Barto, 2017)	12
Şekil 3.2.	MPC tabanlı pekiştirmeli öğrenme sistemi	32
Şekil 3.3.	Model öngörülü kontrol yaklaşımı (Richalet vd., 1978)	40
Şekil 3.4.	Model öngörülü kontrol sistemi	41
Şekil 3.5.	Optimal kontrol problemlerinin çözüm yöntemleri.....	41
Şekil 3.6.	Optimal kontrol problemlerinin çözüm adımları.....	46
Şekil 3.7.	Doğrudan çoklu atış yönteminin gösterimi (Diehl ve Gros, 2011)	48
Şekil 3.8.	Tamamlayıcı gevşeklik koşulunun esnetilmesi (Diehl ve Gros, 2011). ..	58
Şekil 3.9.	Bariyer problemine ait grafiksel gösterim (Diehl ve Gros, 2011).....	59
Şekil 3.10.	Sistem tanımlama yöntemleri (Van Overschee ve De Moor, 1994)	61
Şekil 3.11.	Sistem tanımlama döngüsü (Ljung, 1999).....	62
Şekil 3.12.	Toplamsal bozulmalı doğrusal zamanla değişmeyen (LTI) sistem	64
Şekil 3.13.	ARX model yapısı	70
Şekil 3.14.	ARMAX model yapısı.....	71
Şekil 3.15.	Çıkış-Hatası model yapısı.....	72
Şekil 3.16.	Box-Jenkins model yapısı.....	73
Şekil 3.17.	Tahmin hatası blok şeması	76
Şekil 3.18.	Parametre vektörü tahmin sistemi	80
Şekil 3.19.	İnkübatör cihazının kapağı kapalı durumdaki görünümü.....	81
Şekil 3.20.	İnkübatör cihazının kapağı açık durumdaki görünümü	81
Şekil 3.21.	Veri toplama sistemi.....	82
Şekil 3.22.	Birim basamak ve PRBS cevabı ile PID ayar giriş-çıkış grafikleri.....	83
Şekil 4.1.	İlk senaryoda ARX[2 2] için k-adım tahmin ve benzetim cevapları	87
Şekil 4.2.	İkinci senaryoda ARX[2 2] için k-adım tahmin ve benzetim cevapları ..	87
Şekil 4.3.	Birinci senaryoda ARX[2 2] için kalıntı korelasyonları.....	88
Şekil 4.4.	İkinci senaryoda ARX[2 2] için kalıntı korelasyonları	89
Şekil 4.5.	İlk senaryoda ARMAX[2 2] için k-adım tahmin ve benzetim cevapları ..	90
Şekil 4.6.	İkinci senaryoda ARMAX[2 2] için tahmin ve benzetim cevapları...	91
Şekil 4.7.	Birinci senaryoda ARMAX[2 2] için kalıntı korelasyonları	92
Şekil 4.8.	İkinci senaryoda ARMAX[2 2] için kalıntı korelasyonları.....	92
Şekil 4.9.	İlk senaryoda OE[2 2] için k-adım tahmin ve benzetim cevapları	93
Şekil 4.10.	İkinci senaryoda OE[2 2] için k-adım tahmin ve benzetim cevapları ...	94
Şekil 4.11.	Birinci senaryoda OE[2 2] için kalıntı korelasyonları.....	95
Şekil 4.12.	İkinci senaryoda OE[2 2] için kalıntı korelasyonları	95
Şekil 4.13.	İlk senaryoda BJ[2 2 2] için k-adım tahmin ve benzetim cevapları ...	97
Şekil 4.14.	İkinci senaryoda BJ[2 2 2] için tahmin ve benzetim cevapları	97
Şekil 4.15.	Birinci senaryoda BJ[2 2 2] için kalıntı korelasyonları.....	98
Şekil 4.16.	İkinci senaryoda BJ[2 2 2] için kalıntı korelasyonları.....	99
Şekil 4.17.	İlk senaryo için tahmin aralığına göre uyum dereceleri	100
Şekil 4.18.	İkinci senaryo için tahmin aralığına göre uyum dereceleri	101
Şekil 4.19.	s1 durumunun zamana bağlı değişimi	103
Şekil 4.20.	s2 durumunun zamana bağlı değişimi	104
Şekil 4.21.	a giriş sinyalinin zamana bağlı değişimi	105
Şekil 4.22.	Zamansal fark hatasının zamana bağlı değişimi.....	106
Şekil 4.23.	Adım maliyetinin zamana bağlı değişimi.....	107
Şekil 4.24.	b parametre vektörünün zamana bağlı değişimi.....	108

Şekil 4.25.	f parametre vektörünün zamana bağlı değişimi.....	109
Şekil 4.26.	V0 parametresinin zamana bağlı değişimi	110
Şekil 4.27.	A parametre vektörünün zamana bağlı değişimi	111
Şekil 4.28.	B parametre vektörünün zamana bağlı değişimi	112



ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 4.1. ARX [2 2] modelinin her iki senaryo için başarımları.....	88
Çizelge 4.2. ARMAX [2 2 2] modelinin her iki senaryo için başarımları.....	91
Çizelge 4.3. OE [2 2] modelinin her iki senaryo için başarımları.....	94
Çizelge 4.4. BJ [2 2 2 2] modelinin her iki senaryo için başarımları	98
Çizelge 4.5. Çalışmada kullanılan modellerin karşılaştırılması	99



1. GİRİŞ

Endüstriyel kontrol sistemlerinin önemli bir kısmında halen oransal-integral-türevsel (PID) tip geri beslemeli kontrol cihazlarının kullanılmasına rağmen, yüksek verimlilik ve doğruluk istenen dinamik sistemlerin kontrolü model öngörülü kontrol (MPC) sistemleri ile sağlanmaktadır (Desborough ve Miller, 2002). Çoğu zaman güvenlik endişeleri nedeniyle, MPC sisteminin, sistem çıkışlarına doğrudan müdahale etmesine izin verilmemektedir. Gerçek zamanlı optimizasyon belirli zaman aralıklarında çalıştırılarak MPC için en uygun hedefler belirlenir. Ardından, MPC sisteminin PID kontrol cihazlarının set değerlerini ayarlaması sağlanarak katmanlı bir kontrol yapısı oluşturulur (Nian vd., 2020).

Son zamanlarda yapılan araştırmalarda, gerçek zamanlı optimizasyon, MPC sisteminin amaç fonksiyonun içerisine yerleştirilmektedir. Bu sayede, hem kontrol problemi hem de süreç yönetimi aynı anda ele alınmaktadır. Bu yaklaşım, ekonomik model öngörülü kontrol (EMPC) olarak da bilinmektedir (Ellis vd., 2014). MPC sistemlerinin tasarımında kontrol edilmek istenen sistemin modeline ihtiyaç duyulmaktadır. MPC sisteminin performansı, modellenen sistemin kalitesine bağlıdır (Gros ve Zanon, 2020). Kontrol edilmek istenen sistemin kararlı olarak çalışması yaklaşık bir model kullanılarak sağlanabilirken, optimal kontrol için kesin doğrulukta bir modele ihtiyaç duyulmaktadır. Adaptif kontrol, sistemden alınan ölçüm değerlerini kullanarak MPC parametrelerini veya sistem modelini iyileştirebilir. Ancak bu, tek başına sistemin kararlılığını ve kontrol sisteminin güvenliğini garanti edemez (Aswani vd., 2013).

Son yıllarda yapay zeka alanında yapılan bilimsel yayınlar incelendiğinde, kontrol teorisi, matematiksel optimizasyon ve proses yönetimi gibi çalışma alanlarında birbirine yakın sayıda çalışma yapıldığı görülmektedir. Buna karşın, makine öğrenmesi ile ilgili yapılan çalışmaların sayısı, diğer alanlara göre oldukça fazladır (Perrault ve Clark, 2024). Makine öğrenmesinin üç ana başlığından biri pekiştirmeli öğrenme (RL) algoritmasıdır. Bu algoritma kullanılarak satranç ve Go (Silver vd., 2018) oyunlarında tanınan usta oyuncuların yenilmesi, robotlara verilen yürüme (Wang vd., 2012) ve akrobatik havacılık

(Abbeel vd., 2006) gibi görevlerin herhangi bir ilave kontrol denetimine ihtiyaç duyulmadan yapılabilmesi sağlanmıştır.

Pekiştirmeli öğrenme (RL) yöntemleri, optimal politika veya eylem-durum değer fonksiyonlarının fonksiyon yaklaşımı için genellikle derin sinir ağlarını (DNN) kullanır. İçerisinde DNN veya genel bir fonksiyon yaklaşımı bulunan bir kontrol sisteminin kapalı çevrim davranışını analiz etmek oldukça zor olduğundan, güvenlik açısından kritik öneme sahip kontrol sistemlerinde bu yöntemlerin kullanılmasına onay verilmesi zordur (Gros ve Zanon, 2020).

Pekiştirmeli öğrenme algoritmasının öğrenme ve kontrol kabiliyetinin umut verici olmasına rağmen gerçek kontrol sistemlerinde kullanılabilmesi için kararlılık ve güvenlik gereksinimlerinin de göz önünde bulundurulması gerekmektedir. MPC sistemi, belirsiz ve kısıtlı sistemler için bu gereksinimleri garanti etmektedir fakat başarımı, tahmin edilen modelin doğruluğuna bağlıdır. Her iki yöntemin güçlü ve zayıf yönlerinin birbirini tamamlaması amacıyla yapılan çalışmalar oldukça başarılı sonuçlar vermektedir (Zanon ve Gros, 2020).

Laboratuvar inkübatörleri, mikrobiyal ve hücre kültürlerini korumak ve yetiştirmek için kullanılır. Biyolojik materyallerin yetiştirme ve büyüme süreçleri için belirli sıcaklık gereksinimleri bulunmaktadır (Cappuccino ve Welsh, 2016). Uygun büyüme koşullarını sağlamak, hücre hasarını önlemek ve aynı zamanda hızlı yükselme ve yerleşme süreleri elde etmek için minimum aşım ve minimum yerleşme zamanı performans indekslerinin kontrol sistemi tasarımında dikkate alınması gerekir (Miller vd., 2019). Bu nedenle sistemlerin tasarımı sırasında, kontrol sisteminin performansını, uygulamadan önce değerlendirebilmek için uygun modellerin oluşturulması önemlidir. İnkübatörler izole edilmiş ve kapalı cihazlardır, ancak, malzeme yükleme süreci inkübatör odasında önemli sıcaklık salınımlarına neden olmaktadır. Bu işlem, kapı açma etkisi olarak adlandırılır ve kapı açma sonrası toparlanma, laboratuvar cihazlarının performansında en önemli faktörlerden biridir (Pramuditha vd., 2016).

Bu tez çalışmasında, laboratuvar inkübatör cihazının ekonomik maliyet fonksiyonu ve çoklu kısıtlar dikkate alınarak, MPC tabanlı pekiştirmeli öğrenme algoritması ile kontrol edilmesi amaçlanmıştır. MPC sistemi, yanlış bir model kullanılarak tasarlanmış olsa bile, makine öğrenmesi temelli yöntemler kullanılarak, gerçek sistemin optimal kontrol politikasının sağlanabileceği gösterilmiştir. Optimal politika ve durum-eylem değer fonksiyon yaklaşımı MPC sistemi tarafından sağlanmıştır. Aşama maliyeti, terminal maliyeti, kısıtlar ve inkübatör cihazının tahmini sistem modeli, MPC sistemi içerisinde parametrik olarak tanımlanmıştır. MPC sisteminin parametreleri, en küçük kareler zamansal fark (LSTD) Q-öğrenme yöntemi ile veriye dayalı olarak ayarlanmıştır.

İnkübatör cihazının matematiksel bir modelini oluşturmak için sistem tanımlama yöntemi kullanılmıştır. İnkübatör, doğrusal zamanla değişmeyen tek giriş-tek çıkış (LTI SISO) sistem olarak ele alınmış ve modelleme için dört farklı model yapısı uygulanmıştır: dışsal girdili otoregresif (ARX), dışsal girdili otoregresif hareketli ortalamalar (ARMAX), çıkış hatası (OE) ve Box-Jenkins (BJ). Model yapılarının parametreleri, tahmin hatası yöntemi (PEM) kullanılarak tahmin edilmiştir.

2. KAYNAK ARAŞTIRMASI

Makine öğrenmesi temelli yöntemler, gerçek zamanlı endüstriyel kontrol sistemlerinde robotik kontrol, kestirimci bakım ve üretim süreçlerinin optimizasyonunda kullanılmaktadır. Gerçek zamanlı optimizasyon çalışmalarında, yaklaşım fonksiyonu olarak genellikle derin sinir ağları (DNN) kullanılmaktadır. Ancak, son zamanlarda yapılan güvenli pekiştirmeli kontrol uygulamalarında, yaklaşım fonksiyonu olarak ekonomik MPC sisteminin kullanıldığı görülmektedir. Özellikle güvenlik gereksinimlerinin ön planda olduğu çalışmalarda, klasik sistem tanımlama ve ekonomik MPC yöntemlerinin, güvenli pekiştirmeli öğrenme yöntemleri ile birlikte uygulandığı görülmektedir.

Degrave ve diğerleri (2022), derin pekiştirmeli öğrenme yöntemi kullanılarak, tokamak içerisindeki manyetik alanın kontrolünü sağlamışlardır. Kontrol politikaları, tokamak simülatörü ile etkileşim yoluyla öğrenilmiştir. Plazma kontrolünde makine öğrenmesi yönteminin etkinliği kanıtlanmış ve gelecekteki füzyon araştırmaları için yeni bir kontrol yaklaşımı sunmuşlardır.

Miki ve diğerleri (2022), pekiştirmeli öğrenme algoritması ile dört ayaklı robotların zorlu koşullarda belirlenen rotayı izlemeleri sağlanmıştır. Öğrenme sürecinde model tabanlı algoritmalarla benzetim verilerini kullanarak, robotların gerçek dünya ortamlarına uyum sağlayabilmesi için transfer öğrenmesi yöntemini uygulamışlardır. Sensör verilerinin fazla olması ve karmaşıklığı nedeni ile dikkat tabanlı yinelemeli enkoder yönteminden faydalanmışlardır.

Lazic ve diğerleri (2018), büyük ölçekli bir veri merkezinin soğutma sistemini optimize etmek için pekiştirmeli öğrenme tabanlı MPC sistemi kullanılmıştır. Veri merkezinin termal dinamiklerini modellemek için dışsal girdili otoregresif model (ARX) kullanılmıştır. Bu model üzerinde güvenli sınırlar içinde rastgele keşifler yaparak veri üretilmiştir. Pekiştirmeli öğrenme sistemi bu veriler kullanılarak eğitilmiştir. Eğitilen model MPC sistemi tarafından kullanılarak, soğutma sisteminin kontrol değişkenleri (fan hızları ve su vanası açıklıkları) optimize edilmiş ve enerji verimliliği artırılmıştır.

Gerçek zamanlı optimizasyon ve pekiştirmeli öğrenme alanlarında, genellikle yaklaşım fonksiyonu olarak derin sinir ağları (DNN) kullanılmaktadır. Son zamanlarda yapılan çalışmalarda güvenli pekiştirmeli öğrenme yöntemlerinde DNN yerine MPC yaklaşım fonksiyonu kullanılmakta ve başarılı sonuçlar elde edilmektedir.

Chow ve diğerleri (2018), güvenli pekiştirmeli öğrenme süreçlerinde Lyapunov fonksiyonlarını kullanarak sistemin kararlılığını ve güvenli durumlarda kalmasını garanti eden bir kontrol yaklaşımı önermişlerdir.

Zanon ve Gros (2020), pekiştirmeli öğrenme ile gürbüz MPC sistemini birlikte kullanarak güvenli öğrenme sürecini garanti altına alan, karmaşık ve belirsiz sistemlerde karalı kontrol sağlayan bir yaklaşım sunmuşlardır.

Sistem tanımlama yöntemleri, toplanan giriş-çıkış verilerinden dinamik sistemler için matematiksel modeller oluşturmak için kullanılır (Ljung, 1999). Bu yöntemler, malzeme biliminden biyomedikal mühendisliğine, sinir biliminden doğrusal ve doğrusal olmayan sistemler için otomatik kontrole kadar çok çeşitli araştırma alanlarında kullanılmıştır.

Kaya ve diğerleri (2023), eritilmiş tuz elektroliz sürecinde meydana gelen anot etkisini modellemek amacıyla sistem tanımlama ve makine öğrenmesi temelli yöntemler kullanılmıştır. Bu çalışmada öncelikle deneysel veriler toplanmış, ardından bu veriler dışsal girdili otoregresif (ARX), Hammerstein-Weiner (HW), yapay sinir ağları (ANN) ve transfer fonksiyonu (TF) gibi farklı modelleme yöntemleri ile analiz edilmiştir. Çalışmanın sonuçları, erimiş tuz elektroliz sürecindeki anot etkisinin önceden tahmin edilmesinde, makine öğrenmesi temelli yöntemlerin diğer yöntemlere kıyasla daha yüksek doğruluk sunduğunu göstermiştir.

Shen ve diğerleri (2019), uyku evrelerinin sınıflandırılması için sistem tanımlama temelli bir yöntem önermişlerdir. Yöntem, geleneksel sınıflandırma yöntemlerine kıyasla daha yüksek doğruluk elde etmek amacıyla durum-uzay modeli yapısında bir sistem tanımlama yaklaşımı kullanılmaktadır. Çalışmada, uyku sırasında kaydedilen biyolojik sinyaller analiz edilmektedir. Deneysel sonuçlar, önerilen sistem modelleme yaklaşımı sayesinde

uyku evreleri arasındaki geçişlerin daha iyi takip edilebildiğini ve sınıflandırma doğruluğunun artırıldığını göstermiştir.

Beintema ve diğerleri (2023), doğrusal olmayan sistem tanımlama modelleri kullanarak doğrusal olmayan sistemlerdeki tanımlama problemlerini ayrıntılı olarak incelemiştir. Çeşitli doğrusal olmayan dinamik sistemler üzerinde deneyler yaparak, bu sistemlerin davranışlarını modellemek için ileri düzey algoritmalar geliştirilmiştir. Neuro-fuzzy sistemler ve yapay sinir ağları gibi yöntemleri kullanarak model doğruluğunu artırmayı başarmışlardır. Gedon ve diğerleri (2021) ise, benzer bir yaklaşımla, doğrusal olmayan sistemlerde meydana gelen karmaşık etkileşimleri anlamak için yeni sistem tanımlama teknikleri önermiştir. Bu teknikler, özellikle adaptif kontrol ve tahmin yöntemleri ile birleştirilerek, sistem performansını ve kararlılığını iyileştirmiştir. Her iki çalışma da, doğrusal olmayan sistemlerin daha doğru ve güvenilir bir şekilde modellenmesine olanak tanıyan yenilikçi yöntemlerin geliştirilmesine önemli katkılarda bulunmuştur.

Oymak ve Özay (2019), giriş-çıkış verilerini kullanan doğrusal sistem tanımlamanın örnek karmaşıklığını analiz etmiştir. Bu çalışmada, doğrusal sistemlerin modelleme sürecinde gerekli olan veri miktarı ve bu verilerin kalitesi incelenmiştir. Özellikle, sistem parametrelerinin doğru bir şekilde tahmin edilebilmesi için gerekli olan örnek sayısının, sistemin karmaşıklığı ve gürültü seviyesi gibi faktörlere bağlı olarak nasıl değiştiği araştırılmıştır. Çalışma, teorik analizler ve benzetimler yoluyla, çeşitli doğrusal sistemler için örnek karmaşıklığının belirlenmesine yönelik kapsamlı bir çerçeve sunmuştur. Bu bulgular, doğrusal sistem tanımlama sürecinde daha verimli veri toplama stratejileri geliştirilmesine katkı sağlamıştır.

Bagherian ve diğerleri (2021), doğrusal olmayan sinir devreleri için doğrusal olmayan sistem tanımlamanın başarılı bir şekilde kullanılma potansiyelini göstermiştir. Bu çalışmada, biyolojik sinir devrelerinin karmaşık dinamiklerini modellemek amacıyla çeşitli doğrusal olmayan sistem tanımlama teknikleri uygulanmıştır. Özellikle, yapay sinir ağları ve neuro-fuzzy modeller gibi yöntemlerin doğrusal olmayan sinir devrelerinin davranışlarını tahmin etmede yüksek doğruluk sağladığı gösterilmiştir. Çalışma, bu tekniklerin biyomedikal mühendislikte ve nörolojik araştırmalarda potansiyel

uygulamalarını vurgulayarak, doğrusal olmayan sistem tanımlamanın bu alanlarda nasıl etkili bir araç olabileceğini ortaya koymuştur.

Moradimaryamnegari ve diğerleri (2022), iki model öngörülü kontrol algoritmasını birbirleriyle karşılaştırmış ve sistem tanımlama yöntemini kullanan algoritmanın diğerinden daha iyi performans gösterdiğini ortaya koymuştur. Bu çalışmada, doğrusal olmayan sistemlerin kontrolünde kullanılan iki farklı model öngörülü kontrol (MPC) algoritması incelenmiştir. İlk algoritma, geleneksel MPC yaklaşımına dayanırken, ikinci algoritma sistem tanımlama yöntemleri kullanılarak geliştirilmiştir. Deneysel sonuçlar, sistem tanımlama tabanlı MPC algoritmasının, sistem dinamiklerini daha iyi modelleyerek ve kontrol sinyallerini daha doğru şekilde üreterek daha üstün bir performans sergilediğini göstermiştir. Bu bulgular, sistem tanımlama yöntemlerinin model öngörülü kontrol uygulamalarında potansiyel faydalarını ortaya koymaktadır.

Armenise ve diğerleri (2018), açık kaynaklı bir sistem tanımlama yazılım paketi sunmuştur. Paket, giriş-çıkış aktarım fonksiyonu modelleri ve durum-uzay modelleri için sistem tanımlama yöntemlerini içermektedir. Yazılım, kullanıcıların deneysel verilerden sistem modelleri oluşturmasını kolaylaştırmak amacıyla geliştirilmiştir. Paket, hem doğrusal hem de doğrusal olmayan sistemlerin tanımlanması için farklı algoritmalar ve araçlar sunarak geniş bir uygulama yelpazesine hitap etmektedir. Bu yazılımın, araştırma ve endüstriyel uygulamalarda sistem tanımlama süreçlerini daha erişilebilir ve etkili hale getirmesi hedeflenmiştir.

Tahmin hatası yöntemleri (PEM), en çok kullanılan sistem tanımlama yöntemlerinden biridir. Bu yöntemler, model parametrelerini belirlemek için tahmin edilen sistem çıktıları ile gerçek sistem çıktıları arasındaki farkı (hata) minimize etmeye çalışır. PEM, doğrusal ve doğrusal olmayan sistemlerin tanımlanmasında yaygın olarak kullanılmakta olup, yüksek doğruluk ve güvenilirlik sunar. Bu yöntemler, özellikle kontrol sistemleri, sinyal işleme ve dinamik sistem modelleme gibi alanlarda etkili sonuçlar elde etmek için tercih edilir. PEM'in başarısı, kullanılan veri miktarı, veri kalitesi ve seçilen model yapısına bağlı olarak değişebilir, bu nedenle doğru veri toplama ve model seçimi kritik öneme sahiptir.

Pillonetto ve Ljung (2023), tam Bayes çerçevesine dayalı doğrusal bir sistem tanımlama yöntemi önermiştir. Prosedür, sinir aktivitesi tahmini ve sıcaklık tahmini amaçları için uygulanmıştır. Bu yöntem, sistem parametrelerinin ve model yapısının belirlenmesi için Bayes istatistiğini kullanarak, belirsizlikleri ve ön bilgileri dikkate alır. Önerilen yöntemde, model parametrelerinin olasılık dağılımları güncellenerek, daha doğru ve güvenilir tahminler elde edilir. Bu yaklaşım, özellikle sistem parametrelerinin bilinmediği veya belirsiz olduğu durumlarda avantaj sağlar. Pillonetto ve Ljung'un çalışması, doğrusal sistemlerin tanımlanmasında Bayes yöntemlerinin etkinliğini göstererek, geleneksel yöntemlere kıyasla daha esnek ve güçlü bir modelleme aracı sunmaktadır.

Fotouhi ve diğerleri (2017), iki elektrikli araç aküsü türü için model tanımlama çalışması sunmuştur. Eşdeğer devre akü modeli PEM ile modellenmiştir. Bu çalışmada, elektrikli araç akülerinin dinamik davranışlarını modellemek amacıyla eşdeğer devre akü modelleri kullanılmıştır. Modelleme sürecinde, tahmin hatası yöntemleri (PEM) kullanılarak akülerin giriş-çıkış verileri analiz edilmiş ve bu verilerden yola çıkarak model parametreleri tahmin edilmiştir. Çalışma, eşdeğer devre modellerinin, akülerin performansını ve durumunu doğru bir şekilde tahmin edebildiğini göstermiştir. Fotouhi ve ekibinin çalışması, elektrikli araç akü sistemlerinin tasarımı ve yönetimi için değerli bir referans niteliği taşımaktadır.

Gupta ve diğerleri (2020), ısı değişim sistemini sistem tanımlaması yoluyla modellenmiştir. Uygulanan model yapıları için parametre tahmin görevi PEM kullanılarak yapılmıştır. Bu çalışmada, ısı değişim sisteminin dinamik davranışlarını anlamak ve kontrol etmek amacıyla sistem tanımlama teknikleri kullanılmıştır. Deneysel veriler toplanarak, sistemin giriş-çıkış ilişkileri analiz edilmiş ve uygun model yapıları seçilmiştir. Model parametreleri, tahmin hatası yöntemi (PEM) kullanılarak tahmin edilmiştir. Çalışmanın sonuçları, elde edilen modellerin ısı değişim sisteminin performansını ve dinamik tepkilerini doğru bir şekilde temsil edebildiğini göstermiştir. Bu çalışma, ısı değişim sistemlerinin tasarımı ve optimizasyonunda sistem tanımlama yöntemlerinin etkinliğini vurgulamaktadır.

Fatima ve diğerleri (2022), insansız hava aracı için bir uçuş dinamiği modelinin geliştirilmesini ele almıştır. Sistem tanımlama, gerçek uçuş verilerine uygulanmış ve farklı yapılara sahip çeşitli modeller incelenmiştir. Bu çalışmada, insansız hava aracının (İHA) uçuş dinamiklerini doğru bir şekilde modellemek amacıyla çeşitli sistem tanımlama teknikleri kullanılmıştır. Deneysel veriler, uçuş sırasında toplanmış ve bu veriler kullanılarak İHA'nın dinamik davranışlarını temsil eden matematiksel modeller oluşturulmuştur. Çalışmada, model parametreleri tahmin hatası yöntemleri (PEM) ve diğer optimizasyon teknikleri ile tahmin edilmiştir. Elde edilen modeller, İHA'nın uçuş kontrol sistemlerinin tasarımı ve performans değerlendirmesi için kullanılmıştır. Fatima ve ekibinin çalışması, insansız hava araçlarının güvenilir ve hassas bir şekilde kontrol edilmesine yönelik önemli katkılar sağlamaktadır.

Bnhamdoon ve diğerleri (2020), kuadkopter otopilot sistemini, sistem tanımlama yöntemi ile tanımlamışlardır. Önerilen yöntemin otopilot dinamiklerini yakalama ve çıktıları doğru bir şekilde tahmin etme yeteneğine sahip olduğu gösterilmiştir. Bu çalışmada, kuadkopter otopilot sistemi için dinamik modeller geliştirilmiştir. Sistem tanımlama teknikleri kullanılarak, kuadkopter giriş-çıkış verileri analiz edilmiş ve uçuş dinamikleri için uygun model yapıları oluşturulmuştur. Çalışma kapsamında, deneysel veriler toplanarak, model parametreleri tahmin hatası yöntemi (PEM) ile tahmin edilmiştir. Elde edilen modeller, kuadkopter kararlılığını ve kontrol performansını artırmak amacıyla otopilot sistemlerinde kullanılmıştır. Bnhamdoon ve arkadaşlarının çalışması, kuadkopter gibi karmaşık sistemlerin kontrol edilmesinde sistem tanımlama yöntemlerinin etkinliğini vurgulamaktadır.

Paschke ve diğerleri (2019), kısa vadeli oda sıcaklığı modellerini PEM kullanarak elde etmiştir. k-adım ileri tahminin başarılı performans sonuçları verdiği gösterilmiştir. Bu çalışmada, iç ortam sıcaklığının dinamik değişimlerini tahmin etmek amacıyla kısa vadeli modeller geliştirilmiştir. Deneysel veriler toplanarak, bu veriler PEM kullanılarak analiz edilmiştir ve oda sıcaklığını doğru bir şekilde tahmin eden matematiksel modeller oluşturulmuştur. Elde edilen modeller, enerji verimliliğini artırmak ve iç ortam konforunu sağlamak amacıyla HVAC (Isıtma, Havalandırma ve İklimlendirme) sistemlerinin

optimizasyonunda kullanılabilir. Paschke ve arkadaşlarının çalışması, iç ortam sıcaklık kontrolünde kısa vadeli tahmin modellerinin önemini ve etkinliğini göstermektedir.

Bu tez çalışmasında, literatürdeki çalışmalardan farklı olarak, laboratuvar inkübatör cihazının ekonomik MPC tabanlı pekiştirmeli öğrenme algoritması ile optimal kontrol politikasını sağlayacak şekilde durum referansını izlemesi sağlanmıştır. Laboratuvar inkübatör cihazı Box-Jenkins yapısında PEM yöntemi ile modellenmiştir.



3. MATERYAL ve YÖNTEM

3.1. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme, çevre ile etkileşimler yoluyla kazanılan deneyimlere dayalı olarak hedeflere ulaşmak için ne yapılması gerektiğini öğrenme sürecidir. Öğrenen, etmen olarak adlandırılır ve çevre, etmenin faaliyetlerde bulunduğu fiziksel dünyadır. Her eylemden sonra etmen, çevreden geri bildirim olarak ödül sinyalleri alır. Etmenin hangi eylemi gerçekleştirmesi gerektiğine dair açık talimatlar olmadan, etmen ödül sinyallerini maksimize eden eylemleri keşfeder. Bu yönüyle, denetimli ve denetimsiz öğrenmeden farklıdır. Denetimli öğrenme, bir denetleyiciden etiketli veri seti gerektirirken, denetimsiz öğrenme etiketlenmemiş bir veri setinde altta yatan yapıyı bulmayı amaçlar.

Pekiştirmeli öğrenme aşağıdaki durumlarda geniş kullanım alanları bulmaktadır;

- Ortamın modeli biliniyor fakat analitik bir çözüm mevcut değilse;
- Sadece ortamın benzetimi verilmiş ise (Gosavi, 2009);
- Ortam hakkında bilgi toplamanın tek yolu onunla etkileşim kurmak ise çözüm için pekiştirmeli öğrenme kullanılabilir.

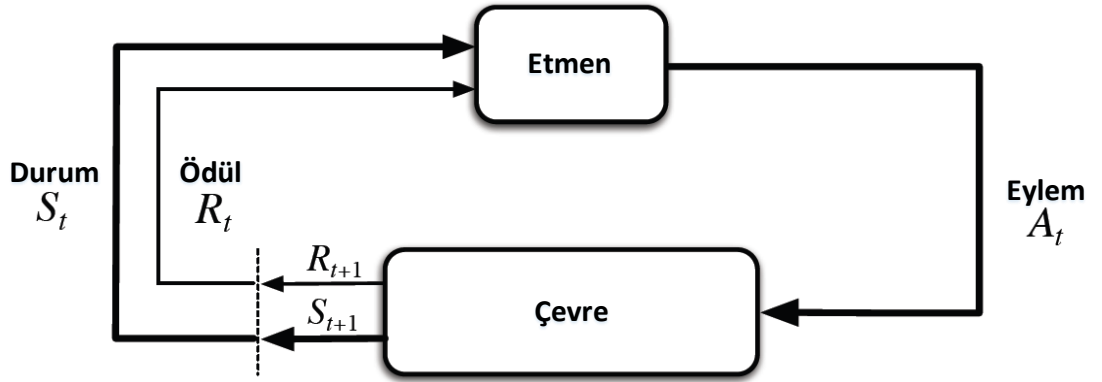
İlk iki durumda, model hakkında belirli bir bilgi mevcut olduğundan, bu problemler planlama problemleri olarak ele alınabilir. Ancak, son durum özgün bir öğrenme problemidir. Buna rağmen pekiştirmeli öğrenme, her iki durumu da makine öğrenmesi problemlerine dönüştürebilir.

Politika ve değer iterasyonunda değerler ve geçiş olasılıklarının bilinmesi gerekmele beraber pekiştirmeli öğrenme bu değerler olmadan tecrübe kazanıp bunu problemin çözümünde kullanabilmektedir (Watkins, 1989).

Pekiştirmeli öğrenmenin denetimli öğrenmeden farkı, girdi ve çıktıların etiketlenmiş olmasına gerek olmamakla beraber; en uygun davranışların kesinlikle doğru olmasına gerek yoktur. Bunun yerine keşfedilmemiş verileri keşfetme ve bilinen verileri kullanma arasında bir denge kurmaya çalışır (Kaelbling vd., 1996).

Çevre, genellikle Markov karar süreci (MDP) olarak ifade edilir çünkü birçok pekiştirmeli öğrenme algoritması optimizasyon amacıyla dinamik programlama yöntemlerini kullanmaktadır (Van Otterlo ve Wiering, 2012). Geleneksel dinamik programlama yöntemleri ile pekiştirmeli öğrenme algoritmaları arasındaki önemli fark, pekiştirmeli öğrenmenin MDP'nin tam bir matematiksel modeline sahip olmaması ve büyük MDP'lerde kesin yöntemler yerine yaklaşık yöntemlerin uygulanabilir olmasıdır.

3.1.1. Markov Karar Süreci



Şekil 3.1. Pekiştirmeli öğrenmenin tipik blok şeması (Sutton ve Barto, 2017)

Basit bir pekiştirmeli öğrenme modeli, sonlu Markov karar süreci (MDP) olarak modellenir:

- $\mathcal{S}$, çevrenin alabileceği tüm durumları içeren sonlu durum kümesi,
- $\mathcal{A}(s)$, etmenin s durumunda uygulayabileceği tüm davranışları içeren sonlu eylem (aksiyon) kümesi,
- $\mathcal{R}$, olası tüm ödülleri içeren sonlu ödül kümesi, $\mathcal{R} = \{r_1, r_2, \dots, r_n\} \subset \mathbb{R}$,
- $p(s', r \mid s, a)$, belirli bir $s \in \mathcal{S}$ durumunda, $a \in \mathcal{A}(s)$ davranışı uygulandıktan sonra, $s' \in \mathcal{S}$ durumuna geçip $r \in \mathcal{R}$ ödülünün elde edilme olasılığıdır:

$$p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \quad (3.1)$$

Beklenen ödülleri hesaplamak için iki farklı fonksiyondan faydalanılır:

- s durumunda a davranışını sergiledikten sonra beklenen ödülleri, $r: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$:

$$r(s, a) \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a) \quad (3.2)$$

- s durumunda a davranışını sergileyip s' durumuna geçildikten sonra beklenen ödüller, $r: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$:

$$r(s, a, s') \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)} \quad (3.3)$$

Pekiştirmeli öğrenme etmeni ortamı ile ayrık zaman dilimlerinde etkileşime girmektedir. Her t anında etmen şu anki durum olan s_t durumunu ve r_t ödülünü alır. Ardından mevcut davranışlar arasından bir a_t davranışı seçer ve daha sonra ortama gönderir. Ortam yeni s_{t+1} durumu ve r_{t+1} ödülüne geçer; böylece geçiş ile ilgili dizi $(s_t, r_t, a_t, r_{t+1}, s_{t+1})$ belirlenmiş olur. Pekiştirmeli öğrenme etmeninin amacı beklenen kümülatif ödülü arttıran $\pi: A \times S \rightarrow [0,1]$, $\pi(a, s) = Pr(a_t = a \mid s_t = s)$ şeklinde bir politika öğrenmektir.

Pekiştirmeli öğrenme yönteminde, etmen, gerçekleştirmesi gereken amaç veya hedefin doğrudan bir tanımına sahip değildir. Çevre, etmenin hedefini ödül sinyalleri aracılığıyla dolaylı olarak iletir. Etmen, bu ödül sinyallerini alır, değerlendirir ve toplam beklenen ödülü olabildiğince arttırmak için politika fonksiyonunu iteratif olarak günceller.

Etmenin eğitilmesindeki asıl amaç, anlık ödülü arttırmaktan çok, getiri olarak adlandırılan toplam ödülü maksimize etmektir. Her t anında alınan anlık ödül $R_t \in \mathbb{R}$ olmak üzere, getiri Denklem (3.4)'teki gibi hesaplanır:

$$G_t = R_{t+1} + R_{t+2} + \dots + R_{t+n} = \sum_{i=t+1}^{t+n} R_i \quad (3.4)$$

Bu yaklaşım, etmen ile çevre arasındaki etkileşimin alt bölümlere ayrıldığı ve sonlanma durumlarının mevcut olduğu problemlerin çözümünde daha anlamlı hale gelmektedir.

Oyunlar, labirent problemlerinin çözümü ve benzeri tekrarlayan görevler örnek olarak gösterilebilir.

Eğer belirlenen görev sürekli ise getirinin değeri sonsuza yakınsar. Bu nedenle, indirim faktörü adı verilen bir parametre kullanılır ve indirgenmiş toplam getiri, Denklem (3.5)'deki gibi hesaplanır:

$$G_t = \gamma^0 R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (3.5)$$

Burada, indirim faktörü, $0 \leq \gamma \leq 1$ arasında değer alan bir ayar parametresidir ve gelecekte alınacak ödüllerin bulunulan zamandaki değerini hesaplamak için kullanılır. Eğer $\gamma = 0$ olarak ayarlanırsa, etmen sadece anlık ödülle ilgilenir ve uzun vadeli ödülleri göz ardı eder. Eğer $\gamma = 1$ olarak ayarlanırsa, etmen gelecekteki ödüllerin arttırılması için çaba gösterir.

Pekiştirmeli öğrenme teorisi ve algoritmaları açısından, ardışık zaman adımlarındaki getirilerin özyinelemeli olarak yazılması büyük önem taşır ve Denklem (3.6) ile ifade edilir:

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \quad (3.6)$$

Etmen, hangi eylemleri gerçekleştireceğine, deterministik veya stokastik olabilen politika doğrultusunda karar verir. Politika temel olarak durumları eylemler ile eşleştirir. Eğer politika deterministik ise, ayrık bir eşleştirme yapar ve Denklem (3.7)'deki gibi tanımlanır:

$$\pi(s) = a \quad (3.7)$$

Eğer politika stokastik ise, her durum için eylemler üzerinde bir olasılık dağılımı tanımlar ve Denklem (3.8) ile ifade edilir:

$$\pi(a | s) = \mathbb{P}[A = a | S = s] \quad (3.8)$$

Bir s durumunun π politikası altındaki durum-değer fonksiyonu $v_\pi(s)$ ile gösterilir. Etmen s durumunda iken π politikasını izlediğinde elde edeceği indirgenmiş getiri değerini ifade eder ve Denklem (3.9) ile tanımlanır:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t | S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right], \forall s \in \mathcal{S} \text{ için}, \quad (3.9)$$

Bir s durumunda iken, π politikası altında, a davranışını sergilemenin değeri $q_\pi(s, a)$ ile gösterilir ve eylem-değer fonksiyonu olarak adlandırılır. Etmen s durumundan başlayıp a davranışını sergiledikten sonra π politikasını takip ettiğinde elde edeceği indirgenmiş getiri değerini ifade eder ve Denklem (3.10) ile tanımlanır.

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right] \quad (3.10)$$

Pekiştirmeli öğrenme ve dinamik programlamada kullanılan değer fonksiyonlarının temel özelliği Denklem (3.6)'da gösterildiği gibi özyinelemeli olarak yazılabilmeleridir. Politika ifadeleri, değer fonksiyonunu anlık ödül ve indirgenmiş gelecekteki değerlerin toplamı olarak ayrıştırarak özyinelemeli olarak yazılabilir. Bu ifade sonucunda ortaya çıkan denklemler Bellman denklemleri olarak adlandırılır. Bir sonraki durum s' olmak üzere, durum-değer fonksiyonu v_π için Bellman denklemi, Denklem (3.11)'de ifade edilmektedir:

$$\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) [r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']] \quad (3.11) \\
&= \sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_\pi(s')], \forall s \in \mathcal{S} \text{ için,}
\end{aligned}$$

Yazılan ifade, Bellman optimallik ilkesi gereğince açıklanacak olursa, bir durumun değeri sadece o durumda alınan ödül ve sonrasında gelecek durumların değerlerine bağlıdır. Aynı şekilde eylem-değer fonksiyonu için Bellman denklemi, Denklem (3.12)'de ifade edilmektedir:

$$q_\pi(s, a) = \sum_{s', r} p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s', a') \right] \quad (3.12)$$

Denklemlerden de görülebileceği gibi, bir kez s' durumunun değeri elde edildiğinde veya yaklaşık olarak hesaplandığında, s durumunun değeri de öz yinelemeli olarak elde edilir veya yaklaşık olarak hesaplanabilir.

Pekiştirmeli öğrenmede etmenin hedefi, değer fonksiyonları ile ölçülen ödülleri olabildiğince arttırmaktır. Bu nedenle, etmen, diğer tüm politikalar kadar veya daha fazla getiri sağlayacak bir politika keşfetmeye çalışır. Politikalar arasındaki karşılaştırma, tüm durumlar için beklenen getiri değerlerinin karşılaştırılması ile yapılır ve Denklem (3.13)'teki gibi ifade edilir:

$$\pi \geq \pi' \text{ ancak ve ancak } v_\pi(s) \geq v_{\pi'}(s) \quad \forall s \in \mathcal{S} \text{ için} \quad (3.13)$$

Her zaman diğer politikalardan daha iyi bir veya birden fazla politika bulunmaktadır. Bu politikaların tamamına optimal politika adı verilir ve π_* ile gösterilir. Tüm optimal politikalar için durum-değer fonksiyonu ortaktır. Optimal durum-değer fonksiyonu olarak adlandırılan bu fonksiyon, Denklem (3.14) ile gösterilmektedir.

$$v_*(s) \doteq \max_{\pi} v_{\pi}(s), \forall s \in \mathcal{S} \text{ için} \quad (3.14)$$

Aynı şekilde, optimal eylem-değer fonksiyonu Denklem (3.15) ile gösterilir:

$$q_*(s, a) \doteq \max_{\pi} q_{\pi}(s, a), \forall s \in \mathcal{S} \text{ ve } \forall a \in \mathcal{A}(s) \text{ için} \quad (3.15)$$

Optimal eylem-değer fonksiyonu, s durumunda iken a davranışında bulunduktan sonra optimal politikanın takip edilmesinden elde edilecek indirgenmiş beklenen getirinin değerini ifade eder. Eylem-değer fonksiyonu, durum-değer fonksiyonu cinsinden Denklem (3.16)'daki gibi yazılabilir:

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]. \quad (3.16)$$

Denklem (3.11) ile verilen değer fonksiyonu, bir politikaya ait olduğundan, Bellman denklemi ile ifade edilen kendi içerisinde tutarlılık koşulunu sağlamalıdır. Fakat optimal değer fonksiyonu v_* herhangi bir politikaya bağlı olmadan da tanımlanabilir. Bu ifade optimal durum-değer fonksiyonu için Bellman optimallik denklemi olarak bilinir ve Denklem (3.17) ile ifade edilir:

$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s', r} p(s', r \mid s, a)[r + \gamma v_*(s')]. \end{aligned} \quad (3.17)$$

Benzer şekilde, optimal eylem-değer fonksiyonu için Bellman optimallik denklemi, Denklem (3.18) ile ifade edilir:

$$\begin{aligned}
q_*(s, a) &= \mathbb{E} \left[R_{t+1} + \gamma m_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a \right] \\
&= \sum_{s', r} p(s', r \mid s, a) \left[r + \gamma m_{a'} q_*(s', a') \right].
\end{aligned} \tag{3.18}$$

3.1.2. Yaklaşık Değer Fonksiyonu

Optimal değer fonksiyonları, her durumda maksimum ödülü sağlayacak aksiyonları seçmek için teorik olarak en ideal çözümü sunar. Ancak, gerçek dünyada bu optimal politikaları bulmak, özellikle karmaşık ve büyük ölçekli problemler söz konusu olduğunda pratik değildir. Çevrenin tüm dinamikleri bilinse bile, optimal politikaları elde etmek için Bellman denklemlerinin çözülmesi, yüksek hesaplama maliyetine ve büyük bellek gereksinimlerine yol açar. Bu nedenle, pekiştirmeli öğrenme (RL) uygulamalarında doğrudan optimal çözümleri bulmak yerine, yaklaşık değer fonksiyonları kullanılır. Bu fonksiyonlar, çok fazla işlem gücü gerektiren optimal değerlerin yerine kabul edilebilir seviyede doğruluk sağlayan pratik çözümler sunar.

Büyük sistemlerde, tüm durumlar için tablo tabanlı fonksiyonlar tutmak çoğu zaman mümkün olmadığından, parametrelili ve genelleştirilmiş fonksiyon yaklaşımları tercih edilir. Bu tür fonksiyonlar, optimal çözüme yaklaşırken karşılaşılma olasılığı yüksek durumlara odaklanarak performansı artırır.

Yaklaşık değer fonksiyonları, RL yöntemlerinin bellek ve işlem gücü sınırları içinde çalışmasını sağlar. Bu fonksiyonlar, her durum için ayrı değerler saklamak yerine, az sayıda parametreyle çok sayıda durum için genelleştirilmiş değer tahminleri yapar. Böylece, RL sistemleri, optimal çözümler yerine, yakınsayan fonksiyonlarla öğrenme sürecini tamamlayarak, gerçek zamanlı kontrol ve optimizasyon süreçlerinde kullanılabilir. Bu yaklaşım, RL algoritmalarının hem bellek verimliliğini artırır hem de zaman kısıtlamaları altında etkili kararlar alınmasını mümkün kılar. Yaklaşık değer fonksiyonları, karmaşık sistemlerin kontrolünde sağladığı performans ve pratiklik açısından vazgeçilmez hale gelmiştir.

Doğrusal olmayan fonksiyon yaklaşımının değer fonksiyonu olarak kullanıldığı başarılı örneklerden biri, TD-Gammon adlı tavla oyunu uygulamasıdır. Bu uygulamada, değer fonksiyonu yapay sinir ağları ile ifade edilmektedir. Uygulama, sık karşılaşılan oyun senaryolarında doğru karar verirken, nadir karşılaşılan durumlarda hata yapabilmektedir (Sutton ve Barto, 2017). Tavla oyununda olduğu gibi kritik olmayan durumlarda yapılan hataların genel performansı etkilemediği kabul edilse de, hayati öneme sahip kontrol sistemlerinin standartlara uygunluğunun sertifikalandırılmasında karşılaşılan zorluklar nedeniyle, yapay sinir ağları yerine alternatif fonksiyon yaklaşımı yöntemlerinin araştırılması güncel bir konudur.

Bu çalışmada kullanılan pekiştirmeli öğrenme (RL) kontrol sisteminin değer fonksiyonu, doğrusal olmayan fonksiyon yaklaşımı ile ifade edilmektedir. Fonksiyon yaklaşımıcısı olarak, parametrelendirilmiş doğrusal olmayan model öngörülü kontrol (NMPC) sistemi kullanılmıştır.

3.1.3. Dinamik Programlama

Dinamik programlama yöntemi, 1950'lerde Richard Bellman tarafından dinamik sistemlerin kontrol edilmesi ve yöneylem araştırmaları ile ilgili optimizasyon probleminin çözümünün sağlanması amacı ile geliştirilmiştir (Bellman, 1952). Dinamik programlama, optimizasyon probleminin daha küçük alt problemlere bölünerek çözülmesi yaklaşımına dayanır. Doğrusal olmayan ve zamanla değişen sistemlerin kontrol problemlerinin çözümünde kullanılabilmektedir.

Bir probleme dinamik programlamanın uygulanabilmesi için alt problemlerin bu problemin içinde özyinelemeli halde bulunuyor olması gerekir. Bu şartı sağlayan problemlerin değerleri arasında bir ilişki vardır (Cormen vd., 2001). Bu ilişki Bellman denklemi ile ifade edilir.

Bu işlemin nasıl yapılacağı ise Bellman optimallik ilkesinde belirtilmiştir; optimal politikanın, başlangıç durumu ve yapılacak ilk davranış ne olursa olsun, kalan

davranışların ilk davranıştan kaynaklanan duruma göre optimal bir politika oluşturma özelliğine sahiptir (Bellman, 1957).

Dinamik programlama ve pekiştirmeli öğrenme yaklaşımının temel fikri, en uygun politikanın bulunması amacıyla değer fonksiyonlarının kullanılmasına dayanır. Denklem (3.11) ile gösterilen v_π Bellman denklemi, iteratif bir güncelleme kuralı olarak Denklem (3.19)'daki gibi ifade edilir:

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_k(s')] \end{aligned} \quad (3.19)$$

Burada, $v_k = v_\pi$, değer fonksiyonunun sabit noktası olarak kabul edilmiştir. Bu koşullar altında, $k \rightarrow \infty$ iken v_k iterasyonu, v_π değerine yakınsar. Bu algoritmaya iteratif politika değerlendirme adı verilir. Buradan da görüleceği gibi iteratif politika değerlendirme sürecinde belirli bir π politikası takip edilmesi durumunda, yaklaşık değer fonksiyonları ($v_0, v_1, v_2, \dots$) zamanla gerçek değer fonksiyonu olan v_π 'ye yaklaşır. İteratif politika değerlendirme algoritması aşağıda gösterilmektedir:

Algoritma 1	İteratif Politika Değerlendirme ($V \approx v_\pi$)
1. Başlangıç	
	Giriş : π , politikası değerlendirilecek
	$\theta > 0$, tahminin doğruluğunu belirleyen bir eşik değeri belirlenir
	$V(s) \in \mathbb{R}, \forall s \in \mathcal{S}$ durumları için keyfi olarak seçilir ($V(\text{terminal}) = 0$ hariç)
2. Değer İterasyonu	
	tekrar et:
	$\Delta \leftarrow 0$
	tekrar et: tüm $s \in \mathcal{S}$ değerleri için
	$v \leftarrow V(s)$
	$V(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a) [r + \gamma V(s')]$
	$\Delta \leftarrow \max(\Delta, v - V(s))$
	kontrol et: $\Delta < \theta$
son	

Bu çalışmada, yaklaşık değer fonksiyonlarının iterasyonu sırasında sabit politika olarak doğrusal olmayan model öngörülü kontrol (NMPC) sistemi kullanılmıştır.

Değer fonksiyonunun belirli bir politika için hesaplanmasının amacı, daha iyi bir politika bulmak ve mevcut politikayı iyileştirmektir. Politika iyileştirme sürecinde, s durumunda iken mevcut politikanın aksine farklı bir a eylemi seçilir ve ardından mevcut π politikası takip edilir. Bu şekilde bir davranış sergiledikten sonra elde edilen eylem-değer fonksiyonunun değeri, Denklem (3.20)'deki gibi ifade edilir:

$$\begin{aligned} q_{\pi}(s, a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r \mid s, a)[r + \gamma v_{\pi}(s')]. \end{aligned} \quad (3.20)$$

Elde edilen değer, $v_{\pi}(s)$ değerinden büyük ise, $a \neq \pi(s)$ şeklinde uygulanan eylemin, seçildiği politikanın mevcut politikadan daha iyi olduğu anlamına gelmektedir. Buradaki özel durum, politika iyileştirme ilkesine dayanmaktadır. Mevcut π politikası ve π' politikaları deterministik politikalar olmak üzere, Denklem (3.21) ifadesi sağlanıyorsa;

$$q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s), \quad \forall s \in \mathcal{S}. \quad (3.21)$$

o halde π' politikasının en az π politikası kadar iyi olduğu sonucuna varılabilir. Bu durumda, Denklem (3.22) ifadesi sağlanmaktadır:

$$v_{\pi'}(s) \geq v_{\pi}(s), \quad \forall s \in \mathcal{S}. \quad (3.22)$$

Denklem (3.22)'nin sağlanması, Denklem (3.23) ile ifade edilebilmektedir:

$$\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2}) \mid S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s] \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \mid S_t = s] \\
&\vdots \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \mid S_t = s] \\
&= v_{\pi'}(s)
\end{aligned} \tag{3.23}$$

Belirli bir politikada yapılan iyileştirmenin tüm durumlar ve tüm olası eylemler için yapılması istenildiğinde, $q_\pi(s, a)$ fonksiyonuna göre en iyi eylemin her durumda seçilmesi gerekmektedir. Buradan, yeni π' politikası, Denklem (3.24) ile elde edilir:

$$\begin{aligned}
\pi'(s) &\doteq \operatorname{argmax}_a q_\pi(s, a) \\
&= \operatorname{argmax}_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \operatorname{argmax}_a \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_\pi(s')]
\end{aligned} \tag{3.24}$$

Mevcut politikanın değer fonksiyonuna göre daha iyi hale getirilmesi süreci, politika iyileştirme olarak ifade edilir. Eğer politika iyileştirme süreci sonunda elde edilen politika, mevcut politika kadar iyi, fakat daha iyi değil ise, $v_\pi = v_{\pi'}$ eşitliği sağlanır. Bu ifade, Denklem (3.24) üzerinde yerine yazılırsa, Denklem (3.25) elde edilir.

$$\begin{aligned}
v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_{\pi'}(s')]
\end{aligned} \tag{3.25}$$

Elde edilen Denklem (3.25), Bellman optimallik denklemi olup, politika iyileştirme süreci iteratif olarak tekrarlandığında elde edilen politikanın optimal politikaya yakınsadığı görülebilmektedir. İterasyonunun her adımında politika bir önceki adımdan daha iyi hale gelir. Markov karar süreci (MDP) sonlu olduğu için, sınırlı sayıda

yinelemede optimal çözüme ulaşılır. Bu yöntem politik iterasyonu adı verilir ve aşağıdaki tabloda görüldüğü gibi ifade edilmektedir:

Algoritma 2	Politika İterasyonu ($\pi \approx \pi_*$)
1. Başlangıç	$V(s) \in \mathbb{R}$ ve $\pi(s) \in \mathcal{A}(s)$ tüm $s \in \mathcal{S}$ durumları için keyfi olarak seçilir
2. Politika Değerlendirme	tekrar et: $\Delta \leftarrow 0$ tekrar et: tüm $s \in \mathcal{S}$ değerleri için $v \leftarrow V(s)$ $V(s) \leftarrow \sum_{s',r} p(s', r s, \pi(s)) [r + \gamma V(s')]$ $\Delta \leftarrow \max(\Delta, v - V(s))$ kontrol et: $\Delta < \theta$ (tahminin doğruluğunu belirleyen küçük pozitif bir sayı)
3. Politika İyileştirme	politika-sabit $\leftarrow$ evet tekrar et: tüm $s \in \mathcal{S}$ değerleri için eski-eylem $\leftarrow \pi(s)$ $\pi(s) \leftarrow \arg \max_a \sum_{s',r} p(s', r s, a) [r + \gamma V(s')]$ eğer eski-eylem $\neq \pi(s)$, o halde politika-sabit $\leftarrow$ hayır eğer: politika-sabit ise bitir ve $V \approx v_*$ ve $\pi \approx \pi_*$ geri dön aksi halde: 2 adımına dön
son	

Politika iterasyonunun her adımında politika değerlendirme işlemi yapılmaktadır. Politika değerlendirme işlemi, durum kümesi üzerinde birden fazla geçiş gerektiren uzun ve iterasyon gerektiren bir hesaplama yapılmasını gerektirebilir. Politika değerlendirme iterasyonu tekrarlandığında, hesaplanan değer fonksiyonu v_π fonksiyonuna ancak sonsuz yineleme sonucunda ulaşabilir. Politika iterasyonunda uygulanan politika değerlendirme adımı, politika iterasyonunun yakınsama özelliğini kaybetmeden çeşitli şekillerde kesintiye uğratılabilir. Politika değerlendirmesinin sadece bir kez uygulandıktan sonra durdurulması özel bir duruma karşılık gelmektedir. Bu işleme değer iterasyonu adı verilir ve Denklem (3.26) ile ifade edilir:

$$\begin{aligned}
v_{k+1}(s) &\doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s',r \mid s,a) [r + \gamma v_k(s')]
\end{aligned} \tag{3.26}$$

Değer iterasyonu işlemi sonucunda v_* değerinin elde edilebilmesi için teorik olarak sonsuz sayıda iterasyon yapılması gerekmektedir. Ancak, değişimlerin belirli bir eşik değerinden daha küçük olduğu durumda algoritma sonlandırılır. Değer iterasyonu algoritması aşağıdaki tabloda gösterilmiştir.

Algoritma 3	Değer İterasyonu ($\pi \approx \pi_*$)
1. Başlangıç	$V(s) \in \mathbb{R}$ ve $\pi(s) \in \mathcal{A}(s)$ tüm $s \in \mathcal{S}$ durumları için keyfi olarak seçilir $\theta > 0$, tahminin doğruluğunu belirleyen bir eşik değeri belirlenir
2. Değer İterasyonu	tekrar et: $\Delta \leftarrow 0$ tekrar et: tüm $s \in \mathcal{S}$ değerleri için $v \leftarrow V(s)$ $V(s) \leftarrow \max_a (\sum_{s',r} p(s',r \mid s,a) [r + \gamma V(s')])$ $\Delta \leftarrow \max(\Delta, v - V(s))$ kontrol et: $\Delta < \theta$
3. Politika Çıkarımı	$\pi \approx \pi_*$ olmak üzere, aşağıdaki şekilde deterministik bir politika elde edilir $\pi(s) = \arg \max_a \sum_{s',r} p(s',r \mid s,a) [r + \gamma V(s')]$
son	

3.1.4. Zamansal Fark Öğrenme

Optimal kontrol problemleri pekiştirmeli öğrenme yaklaşımını kullanan birçok farklı yöntem ile çözülebilmektedir. Bu yöntemler çeşitli benzerlik ve farklılıklarına göre sınıflandırılabilir. Bu yöntemlerin genel sınıflandırılması, MDP modelinin bilgisine sahip olup olmadıklarına göre model tabanlı veya modelden bağımsız yöntemler olarak yapılabilir. Model tabanlı yöntemlerde, dinamik programlama, değer iterasyonu ve politika iterasyonu gibi yöntemler kullanılarak Bellman denklemleri yinelemeli olarak çözülür, değer fonksiyonları yaklaşık olarak elde edilir ve optimal politikalar bulunur.

Modelden bağımsız yöntemler ise çevre hakkında tam bilgiye sahip değildir ve deneme yanılma yaklaşımıyla gelecekteki eylemlerini gerçekleştirmek için bir politika öğrenmeleri gerekir. Modelden bağımsız yöntemler için daha fazla sayıda veri toplama adımı gerektirmektedir. Modelden bağımsız yöntemlere örnek olarak zamansal fark (TD) ve Monte Carlo (MC) yöntemleri verilebilir.

Monte Carlo yöntemlerinde, değer tahmini ve politika belirleme işleminin yapılabilmesi için etmenin tüm görev döngüsünü tamamlaması gerekir. Değerlendirmenin yapılabilmesi için etmenin duruma uğraması ve getiriye alması beklenir. Gözlenen bu G_t değeri, $V(S_t)$ fonksiyonu için hedef olarak kullanılır. Basit bir Monte Carlo yöntemi Denklem (3.27)'deki gibi ifade edilebilir:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)] \quad (3.27)$$

Zamansal fark (TD) tahmin yöntemlerinde ise sadece bir sonraki adımın tamamlanması yeterlidir. TD yönteminin güncelleme kuralı en basit şekilde, Denklem (3.28) ile ifade edilecek olursa;

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \quad (3.28)$$

burada $R_{t+1} + \gamma V(S_{t+1})$, güncelleme adımında hedef olarak kullanılmaktadır. Buradaki güncelleme işlemi bir adım ilerisi ile sınırlı olduğundan TD(0) olarak da adlandırılmaktadır. Zamansal fark yöntemlerinde hedef ile yapılan tahmin arasındaki farka zamansal fark hatası adı verilir ve Denklem (3.29) ile ifade edilir.

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \quad (3.29)$$

Etmenin kontrol problemini çözebilmesi için durum-değer fonksiyonu yerine eylem-değer fonksiyonunu tahmin etmesi gerekmektedir. Politikaya bağlı (politika-içi) öğrenme yöntemlerinde, etmen kendi davranış politikasını takip eder ve bu politikayı geliştirir. Buradaki temel amaç eylem-değer fonksiyonunu öğrenmek ve bu sayede kontrol

problemini çözebilmektedir. Değer fonksiyonunun zamansal fark yöntemi ile tahminine benzer bir şekilde, mevcut durumda iken bir eylemde bulunulur ve bir sonraki zaman adımında elde edilen getiri ve eylem-durum fonksiyon değeri karşılaştırılır. Buradaki sıralama $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}$ şeklinde olduğundan bu yöntem Sarsa olarak adlandırılmaktadır. Politikaya bağlı zamansal fark kontrol (Sarsa) algoritmasına ait güncelleme kuralı, Denklem (3.30)'daki gibi ifade edilir:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]. \quad (3.30)$$

Aynı şekilde zamansal fark hatası ise Denklem (3.31) ile ifade edilir.

$$\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t). \quad (3.31)$$

Politika-dışı zamansal fark kontrol yöntemi ise Q -öğrenme yöntemi olarak bilinir (Watkins, 1989) ve Denklem (3.32) ile tanımlanmaktadır:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \quad (3.32)$$

Bu yöntemde diğer yöntemlerden farklı olarak, öğrenilen eylem-değer fonksiyonu Q , doğrudan q_* , optimal eylem-değer fonksiyonuna yaklaşmaktadır.

3.1.5. En Küçük Kareler Zamansal Fark (LSTD) Algoritması

En küçük kareler (LS) TD doğrusal fonksiyon yaklaşımı kullanılarak, TD(0) algoritmasının ihtiyaç duyduğu veri miktarının azaltılması amaçlanır. TD(0) algoritmasında parametreler her adımda iteratif olarak güncellenirken, LSTD yönteminde çözüm, lineer cebir hesapları ile doğrudan hesaplanabilir.

TD(0) algoritmasıyla doğrusal fonksiyon yaklaşımında, ağırlık vektörü (w) asimptotik olarak Denklem (3.33) ile gösterilen sabit noktaya ulaşmaktadır.

$$w_{TD} = A^{-1}b \quad (3.33)$$

Burada, $A = E[x_t(x_t - \gamma x_{t+1})^T]$ ve $b = E[R_{t+1}x_t]$ olarak tanımlanmaktadır. Bu ifadede, x_t , durumun özellik vektörüdür ve R_{t+1} ise bir sonraki adımda alınan ödüldür. TD(0) algoritması, sabit noktayı iteratif olarak bulur. Ancak, LSTD bu çözümü daha verimli bir şekilde, doğrudan cebirsel olarak hesaplar.

LSTD yönteminde, A ve b matrislerinin tahminleri doğrudan kullanılır. TD sabit noktası, Denklem (3.34) ve Denklem (3.35) ile tanımlanır:

$$\hat{A}_t \doteq \sum_{k=0}^{t-1} x_k(x_k - \gamma x_{k+1})^T + \varepsilon I \quad (3.34)$$

$$\hat{b}_t \doteq \sum_{k=0}^{t-1} R_{k+1}x_k \quad (3.35)$$

Burada, I birim matristir ve εI terimi, $\hat{A}_t$ matrisinin tersi alınabilir olmasını sağlamak için eklenen küçük bir $\varepsilon > 0$ değeridir. Bu iki matris tahmini sayesinde TD sabit noktası, Denklem (3.36)'da gösterildiği gibi hesaplanır:

$$w_t \doteq \hat{A}_t^{-1}\hat{b}_t \quad (3.36)$$

$\hat{A}_t^{-1}$ matrisinin doğrudan hesaplanması, özellik vektörünün boyutu d olmak üzere, $O(d^3)$ zaman karmaşıklığına sahiptir. Ancak, LSTD algoritmasında, matris tersi, Denklem (3.37)'de gösterildiği gibi Sherman-Morrison formülü kullanılarak verimli bir şekilde güncellenir (Sutton ve Barto, 2017).

$$\begin{aligned} \hat{A}_t^{-1} &= (\hat{A}_{t-1} + x_t(x_t - \gamma x_{t+1})^T)^{-1} \\ &= \hat{A}_{t-1}^{-1} - \frac{\hat{A}_{t-1}^{-1}x_t(x_t - \gamma x_{t+1})^T\hat{A}_{t-1}^{-1}}{1 + (x_t - \gamma x_{t+1})^T\hat{A}_{t-1}^{-1}x_t} \end{aligned} \quad (3.37)$$

3.1.6. Politika Gradyanı Yöntemleri

Politika tabanlı yöntemler, eylem-durum değer fonksiyonunu yaklaştırmak yerine doğrudan politikayı optimize ederler. Bu şekilde, eylemler doğrudan tahmin edilebildiğinden ve her olası ayrık eylem için bir Q değeri belirtilmesine gerek olmadığından, politika tabanlı yöntemler sonsuz ve sürekli eylem uzaylarını kullanabilirler. Politika, π_θ olarak sinir ağları ile parametrelendirilir. Beklenen getiriyi maksimize etmek için politika parametreleri, gradyan iniş yöntemiyle optimize edilir ve Denklem (3.38)'de gösterilmektedir:

$$\max_{\theta} J(\pi_{\theta}) = \max_{\theta} E_{\pi_{\theta}}[G_t \mid s_t = s] \quad (3.38)$$

Amaç fonksiyonunun gradyanını elde etmek zor olduğundan dolayı, politika gradyanı teoreminin yardımı ile $J(\pi_{\theta})$ 'nin gradyanı, Denklem (3.39)'da gösterildiği gibi ifade edilir:

$$\begin{aligned} \nabla J(\pi_{\theta}) &= \mathbb{E}[\nabla_{\theta} \log \pi_{\theta}(a \mid s) G_t] \\ &= \mathbb{E}[\nabla_{\theta} \log \pi_{\theta}(a \mid s) Q_{\pi}(s, a)] \end{aligned} \quad (3.39)$$

Ardından, güncelleme kuralı, Denklem (3.40)'da gösterildiği gibi ifade edilir:

$$\theta_{t+1} = \theta_t + \alpha R_{t+1} \nabla_{\theta} \log \pi_{\theta}(a \mid s) \quad (3.40)$$

Politika tabanlı yöntemlerin bu biçimi, temel olarak, politika parametrelerini güncellemek için örnek bölümleri kullanan REINFORCE olarak bilinir. Ancak, stokastik politika farklı bölümlerde farklı eylemler seçebilir ve eylemlerdeki küçük farklılıklar bile sonucu değiştirebilir. Sonuç olarak, bu varyans, modelin farklı ve çelişkili iniş yönlerinde öğrenmesine neden olarak bir sorun teşkil eder. Önerilen bir çözüm, durum değer fonksiyonu $V\pi(s)$ 'yi getiriden çıkarmaktır. Bu, ödülleri ortalama eyleme göre yeniden dengelemeye yardımcı olur. G_t yerine farklı ifadeler konularak yeni yöntemler elde edilir. Ayrıca, temelde güncellenen politikaların önceki politikalardan sapmasını engelleyen politika tabanlı yöntemler de vardır.

Aktör-kritik yöntemler, değer tabanlı ve politika tabanlı algoritmaların bir birleşimidir ve her ikisinin faydalı yönlerini birleştirir. Aktör-kritik yöntemler, politika gradyanı yöntemlerinin bir uzantısı gibidir ve politika öğrenimi için bir ağın yanı sıra değer fonksiyonu öğrenimi için de başka bir ağı sahiptir. Burada, aktör terimi, kritik tarafından sağlanan yön doğrultusunda eylemleri seçen parametrelendirilmiş bir politikayı ifade eder ve kritik terimi, aktör tarafından seçilen eylemleri eleştiren durum değeri veya durum-eylem fonksiyonlarının tahminlerini ifade eder. Zamansal fark hatası, genellikle her iki ağı da güncellemek için kullanılır. Aktör-kritik yöntemlerin REINFORCE'a göre avantajlarından biri, tüm yörüngeyi kullanmak yerine zamansal farka dayanmasıdır.

Bu kategori için en önemli algoritmalar arasında derin deterministik politika gradyanı (DDPG) (Lillicrap vd., 2016; Silver vd., 2014), asenkron avantajlı aktör kritik (A3C) (Mnih vd., 2016), proksimal politika optimizasyonu (PPO) (Schulman vd., 2017), güvenli bölge politika optimizasyonu (TRPO) (Schulman vd., 2015) ve esnek aktör kritik (SAC) (Haarnoja vd., 2018) yer almaktadır.

3.2. MPC Tabanlı Pekiştirmeli Öğrenme

Bu çalışmada, kontrol edilmek istenen gerçek sistem olarak kullanılan inkübatör modeli, ayrık Markov karar süreci (MDP) olarak tanımlanmıştır. MDP durum geçiş dinamikleri Denklem (3.41)'deki gibi gösterilebilir:

$$\mathbb{P}[s_+ | s, a] \quad (3.41)$$

Burada s ve a durum-giriş çifti olarak, s_+ ise bir sonraki durumu ifade eder. Aşama maliyet fonksiyonu $L(s, a)$ ile gösterilir ve bazı durum-giriş çiftleri için sonsuz değerini alabileceği, Denklem (3.42)'de gösterilmektedir:

$$L(s, a) = l(s, a) + J_\infty(h(s, a)) + J_\infty(g(a)) \quad (3.42)$$

Gösterge fonksiyonu, Denklem (3.43) ile tanımlanır:

$$J_{\infty}(x) = \begin{cases} \infty & \text{eğer } x_i > 0 \text{ bazı } i \text{ değerleri için} \\ 0 & \text{değilse} \end{cases} \quad (3.43)$$

l fonksiyonu için tanımlanan kısıtlar, Denklem (3.44)'te gösterildiği gibidir:

$$g(a) \leq 0, \quad h(s, a) \leq 0 \quad (3.44)$$

$l(s, a)$, belirli durum s ve kontrol girişi a çiftlerine verilen maliyeti temsil eder. I_{∞} , gösterge fonksiyonu olup, belirli bir kısıtın ihlal edilip edilmediğini kontrol eder. Eğer kısıt ihlal edilirse maliyeti sonsuz yapar, aksi takdirde maliyet sıfır olur. $g(a) \leq 0$, kontrol girişlerine yönelik saf kısıtları temsil eder. Bu kısıt kontrol girişlerinin belirli sınırlar içinde kalmasını sağlar. $h(s, a) \leq 0$, durum ve kontrol girişi ile ilgili karma kısıtları temsil eder. Bu kısıt, belirli durum ve giriş çiftlerinin izin verilen sınırlar içinde kalmasını sağlar. İndirim faktörü $0 < \gamma \leq 1$ ifadesinin de eklenmesi ile Denklem (3.41) ve Denklem (3.42) ile optimal politika $\pi^*(s)$ elde edilir.

MDP literatürü için Denklem (3.41) gösterimi kullanılmasına rağmen, kontrol literatüründe genellikle $s_+ = f(s, a, \zeta)$ notasyonu kullanılır. Burada, ζ , stokastik bir değişken ve f ise muhtemelen doğrusal olmayan bir fonksiyondur. MDP ile ilişkili durum-eylem değer fonksiyonu Q^* ve değer fonksiyonu V^* , Bellman denklemleri ile Denklem (3.45) ve Denklem (3.46)'da gösterildiği gibi tanımlanır:

$$Q_*(s, a) = L(s, a) + \gamma \mathbb{E}[V_*(s_+) \mid s, a] \quad (3.45)$$

$$V_*(s) = Q_*(s, \pi_*(s)) = \min_a Q_*(s, a) \quad (3.46)$$

Gerçek sistem altında bulunan MDP'nin, ilgili adım maliyeti L ve indirim faktörü γ ile tanımlanan sınırları-belli bir problem olduğu varsayılmıştır. Bellman denklemi üzerinde tanımlanmış değer fonksiyonlarının sınırları bellidir ve bazı kümeler üzerinde sonlu

değerlere sahiptir. Buradan, gerçek sistemin modelinin durum geçiş dinamiklerinin, Denklem (3.47)'de gösterildiği gibi olduğu kabul edilmiştir:

$$\mathbb{P}[\hat{s}_+ | s, a] \quad (3.47)$$

Denklem (3.47), Denklem (3.41)'den farklı olarak deterministik modelleri de içermektedir. Bu durumda, aşama maliyet fonksiyonu, Denklem (3.48)'deki gibi tanımlanmış olsun:

$$\hat{L}(s, a) = \begin{cases} Q_*(s, a) - \gamma \mathcal{V}^+(s, a) & \text{eğer } |\mathcal{V}^+(s, a)| < \infty \\ \infty & \text{değilse} \end{cases} \quad (3.48)$$

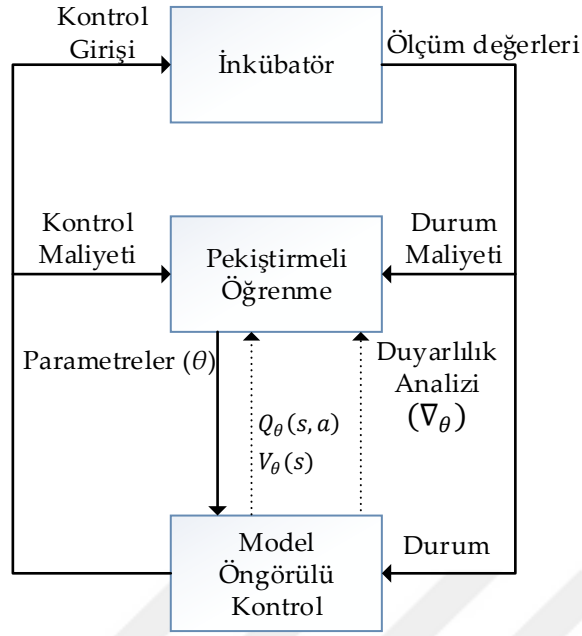
Burada, bir sonraki durum-giriş maliyet fonksiyonu, Denklem (3.49)'da gösterildiği gibi tanımlanır:

$$\mathcal{V}^+(s, a) = \mathbb{E}[V_*(\hat{s}_+) | s, a] \quad (3.49)$$

Denklem (3.48) ile önerilen adım maliyeti şartlı önermesi, hem eylem değeri fonksiyonu hem de değer fonksiyonu sonsuz değerler aldığı anda bile, sınırları belirli bir $\hat{L}$ elde etmek için kullanılan bir teknik yöntemdir.

Bazı koşullar altında, gerçek dinamikler için adım maliyeti L 'yi minimize eden optimal politika π^* , adım maliyeti $\hat{L}$ ile birleştirilmiş model kullanılarak da elde edilebilir. Dolayısıyla, adım maliyetini değiştirerek yanlış bir modele dayanarak optimal politikayı oluşturmak mümkündür. Denklem (3.48) ile ifade edilen şartlı önermenin değerlendirilmesi için gereken zorlukları aşmak ve bunun yerine $\hat{L}$ fonksiyonunu doğrudan verilerden öğrenmek mümkündür (Gros ve Zanon, 2020).

MPC tabanlı pekiştirmeli öğrenme sisteminin blok şeması Şekil 3.2'de gösterilmektedir.



Şekil 3.2. MPC tabanlı pekiştirmeli öğrenme sistemi

3.2.1. MPC Fonksiyon Yaklaşımcısı

MPC sisteminde istenmeyen durumları ve girişleri kısıtlamak için Denklem (3.44) kullanılır. MPC sistemi, hatalı sistem modeli f_θ üzerine kurulu olacağından, gerçek $L(s, a)$ yerine değiştirilmiş adım maliyeti $\hat{L}(s, a)$ fonksiyonunu minimize etmeye çalışacaktır. Sonuç olarak, saf giriş kısıtlamaları $g(a) \leq 0$ sabit olsa bile, MPC şemasında kullanılan karma kısıtlamaların da $\hat{L}(s, a)$ 'nın sonlu olduğu alanı kapsayacak şekilde değiştirilmesi gerekmektedir. $\hat{L}$ önceden bilinmediğinden, sonlu olduğu alan, diğer şartların yanı sıra f_θ ve gerçek sistemin durum geçiş olasılıkları arasındaki farklılıklara bağlıdır ve öğrenilmesi gerekmektedir. Bu nedenle, MPC sistemine, pekiştirmeli öğrenme yöntemi ile ayarlanabilen θ parametreleri kullanılarak, karma kısıtlar $h_\theta(s, a) \leq 0$ öğretilir.

Pekiştirmeli öğrenme ile sonsuz ceza değerleri kullanılamayacağından dolayı, maliyet fonksiyonu L ve karma kısıtlar h_θ 'nin gevşetilmiş halinin kullanılması gerekir.

MPC sistemi θ ile parametrize edilerek, pekiştirmeli öğrenme algoritması için bir fonksiyon yaklaşımıcısı olarak kullanır. Değer fonksiyonu V^* için parametrik gösterim Denklem (3.50) ile Denklem (3.53) arasındaki gibi tanımlanır:

$$V_\theta(s) = \min_{u,x,\sigma} \lambda_\theta(x_0) + \gamma^N (V_\theta^f(x_N) + w_f^\top \sigma_N) + \sum_{k=0}^{N-1} \gamma^k (l_\theta(x_k, u_k) + w^\top \sigma_k) \quad (3.50)$$

şartlarını sağlamak üzere,

$$x_{k+1} = f_\theta(x_k, u_k), x_0 = s, \quad (3.51)$$

$$g(u_k) \leq 0, \quad (3.52)$$

$$h_\theta(x_k, u_k) \leq \sigma_k, h_\theta^f(x_N) \leq \sigma_N. \quad (3.53)$$

Burada, modelin parametrik ifadesi f_θ , kısıtların parametrik ifadesi h_θ , adım maliyetinin parametrik ifadesi l_θ ve terminal maliyeti V_θ^f ile gösterilir. İlave durum maliyeti ise λ_θ ile gösterilmiştir. w_f , öngörü ufkunun terminal durumundaki gevşeklik değişkeni σ_N için uygulanan ağırlıktır. w ise, öngörü ufkunun ara aşamalarındaki kısıtların gevşetilmesi için ENMPC yapısındaki σ_k gevşeme değişkenleri ile ilgili ağırlığı ifade eder.

Pekiştirmeli öğrenme algoritmasında kullanılan politika, Denklem (3.54) ile tanımlanır:

$$\pi_\theta(s) = u_0^* \quad (3.54)$$

Burada, u_0^* , Denklem (3.50)'deki minimizasyon probleminin s durumu için çözümünden bulunan $u_0^*, \dots, u_{N-1}^*$ giriş dizisinin ilk elemanıdır. Burada, N , MPC sisteminin öngörü ufkudur. Q_θ eylem değer fonksiyonu, Denklem (3.56) ve Denklem (3.57) şartlarını sağlamak üzere, Denklem (3.55) ile tanımlanır:

$$Q_\theta(s, a) = \min_{u, x} \quad (3.50) \quad (3.55)$$

$$\text{Denklem (3.51) - (3.53),} \quad (3.56)$$

$$u_0 = a. \quad (3.57)$$

Önerilen parametrik ifadeler, Bellman'ın temel denklemlerini sağlar:

$$\pi_\theta(s) = \arg \min_a Q_\theta(s, a), \quad (3.58)$$

$$V_\theta(s) = \min_a Q_\theta(s, a) \quad (3.59)$$

Bu eşitlikler, parametrik MPC şemasının Bellman denklemlerini sağladığını ve dolayısıyla durum ve eylem değer fonksiyonlarının doğru bir yaklaşıklama sağladığını gösterir (Gros ve Zanon, 2020).

3.2.2. İkinci-Dereceden LSTD Q-Öğrenme Algoritması

En küçük kareler zamansal fark Q-öğrenme (LSTDQ) (Lagoudakis vd., 2002) yöntemi, temel zamansal fark yöntemlerine göre daha hızlı yakınsadığından dolayı, verinin verimli kullanılmasını sağlamaktadır (Esfahani vd., 2021). Q-öğrenme algoritması, yaklaşık eylem-değer fonksiyonu Q_θ için gözlemlenen verilere uyum sağlayacak parametreler arayarak, Q_* fonksiyonunu yaklaşık olarak elde etmeyi ve dolaylı olarak π_* optimal politikasını çıkarmayı amaçlar.

Bellman rezidü hatasının en küçük kareler yöntemi, θ parametre fonksiyonuna göre Denklem (3.60) ile ifade edilirse,

$$\min_{\theta} \mathbb{E}[\|Q_*(s, a) - Q_\theta(s, a)\|^2]. \quad (3.60)$$

Burada, Q_θ , Denklem (3.55) - Denklem (3.57) ile ifade edilen gerçek optimal eylem-değer fonksiyonu Q_* için bir fonksiyon yaklaşımıdır. Bellman optimallik denkleminin, zamansal fark (TD) temelli öğrenme yaklaşımlarında kullanılan yaklaşık ifadesi, Denklem (3.61)'de gösterildiği gibidir:

$$Q_*(s, a) \approx L(s, a) + \gamma Q_\theta(s_+, \pi(s_+)) \quad (3.61)$$

Burada, $L(s, a)$, aşama maliyetidir. Denklem (3.61), en küçük kareler yönteminde yerine yazılırsa, birinci dereceden zamansal fark Q-öğrenme için en küçük kareler problemi, durum ve eylem geçişlerinden elde edilen veriler ile Denklem (3.62) ve Denklem (3.63) kullanılarak çözülebilir:

$$\mathbb{E}[\delta \nabla_\theta Q_\theta(s_i, a_i)] = 0, \quad (3.62)$$

$$\delta = L(s_i, a_i) + \gamma V_\theta(s_{i+1}) - Q_\theta(s_i, a_i), \quad (3.63)$$

Burada, δ zamansal fark hatasıdır. Değer fonksiyonları, $V_\theta(s_{i+1}) = Q_\theta(s_{i+1}, \pi(s_{i+1}))$ ve $Q_\theta(s_i, a_i)$ olmak üzere, MPC fonksiyon yaklaşımından elde edilmektedir. En küçük kareler probleminin çözümü için değiştirilmiş Newton yöntemi, Denklem (3.64) ile ifade edilmektedir (Esfahani vd., 2021).

$$\theta \leftarrow \theta - \alpha A^{-1} b \quad (3.64)$$

Burada, $\alpha > 0$ olmak üzere, adım büyüklüğüdür. A ve b , Denklem (3.65)'te gösterildiği gibi ifade edilmektedir:

$$A = \mathbb{E}[\delta \nabla_\theta^2 Q_\theta + \nabla_\theta Q_\theta \nabla_\theta^\top \delta], \quad b = \mathbb{E}\left[\delta \frac{\partial Q_\theta}{\partial \theta}\right]. \quad (3.65)$$

Newton güncelleme kuralında yer alan gradyan ve Hessian ifadelerinin bulunabilmesi için Q_θ fonksiyonunun duyarlılıklarının hesaplanması gerekmektedir. Denklem (3.55) ile

Denklem (3.57) arasında tanımlanan doğrusal olmayan programlama probleminin duyarlılık analizi yapılarak, Q_θ fonksiyonunun gradyan ve Hessian nicelikleri yaklaşık olarak hesaplanabilmektedir (Büskens ve Maurer, 2001). Doğrusal olmayan model öngörülü kontrol (NMPC) problemine ait Lagrange fonksiyonu, Denklem (3.66) ile ifade edilirse,

$$\mathcal{L}_\theta = \Phi_\theta + \lambda^\top G_\theta + \mu^\top H_\theta \quad (3.66)$$

burada, H_θ eşitsizlik kısıtlarını, Φ_θ ise optimizasyon probleminin maliyetini ifade eder. λ değişkeni, eşitlik kısıtları G_θ ile ilişkili Lagrange çarpanları vektörünü ifade eder. μ değişkeni ise, eşitsizlik kısıtları ile ilgili Lagrange çarpanları vektörünü ifade eder. Denklem (3.55) ile Denklem (3.57) arasında ifade edilen NMPC yapısı için, primal değişkenler $p = \{X, U\}$ ile tanımlanacak olursa;

- $X = \{x_0, x_1, \dots, x_N\}$, sistemin öngörülen durum yörüngesi,
- $U = \{u_0, u_1, \dots, u_{N-1}\}$, sistemin öngörülen giriş yörüngesi,

olmak üzere, burada N , sistemin öngörü ufkunu ifade etmektedir. Primal-dual değişkenler ise, $z = \{p, \lambda, \mu\}$ ile ifade edilmektedir. Buna bağlı olarak, optimizasyon probleminin duyarlılık analizi sonucunda Denklem (3.67) ve Denklem (3.68) ifadeleri elde edilebilmektedir (Büskens ve Maurer, 2001).

$$\frac{\partial Q_\theta}{\partial \theta} = \frac{\partial \mathcal{L}_\theta(s, a, z^*)}{\partial \theta}, \quad (3.67)$$

$$H(Q_\theta) = \frac{D}{D\theta} \left(\frac{\partial \mathcal{L}_\theta(s, a, z^*)}{\partial \theta} \right) \approx \frac{\partial^2 \mathcal{L}_\theta(s, a, z^*)}{\partial \theta^2} \quad (3.68)$$

Burada, z^* , doğrusal olmayan programlama (NLP) probleminin primal-dual çözümü ve D , toplam türevi ifade eder. L_θ adım maliyeti, Denklem (3.69)'daki gibi tanımlanmıştır:

$$L_\theta(s_k, a_k) = l_\theta(x_k, a_k) + w^\top \max(0, h_\theta(x_k, a_k)) \quad (3.69)$$

NMPC yapısı ve adım maliyeti verilmiş, MPC tabanlı RL kontrol sisteminin sözde kodu Algoritma 4 tablosu ile gösterilmiştir.

Algoritma 4	MPC Tabanlı RL
Başlangıç koşullarını al: s_0, a_0, sonlandır=0	
Başlangıç durumu için giriş değerini al: a_k girişini (3.50) ile hesapla sonlandır=0 iken devam et	
1. s_k ve a_k değerlerini oku a. $Q_\theta(s, a) \leftarrow$ Denklem (3.55) 2. a_k girişini gerçek sisteme uygula, a. $L_\theta(s_k, a_k) \leftarrow$ Denklem (3.69) b. sonlandır $\leftarrow$ sonlandırma koşulu sağlanıyor mu? c. $s_k \leftarrow s_{yeni}$ 3. $\pi_\theta(s)$ ve $V_\theta(s) \leftarrow$ Denklem (3.50) a. $a_k = \pi_\theta(s)$ 4. Duyarlılıkları hesapla, parametreleri güncelle a. $\delta \leftarrow$ Denklem (3.63) b. ∇_θ ve $\nabla_\theta^2 \leftarrow$ Denklem (3.67)-(3.68) [$Q_\theta(s, a)$] c. $\theta_{yeni} \leftarrow$ Denklem (3.64) d. $\theta \leftarrow \theta_{yeni}$ 5. $a_{k+1} \leftarrow a_k$ 6. $s_{k+1} \leftarrow s_k$ 7. $k \leftarrow k+1$	
son	

3.2.3. LTI İnkübatör Modeli

Bu tez çalışmasında, gerçek proses modeli olarak kullanılan inkübatör cihazının dinamikleri, Denklem (3.70) ile tanımlanmıştır (Mantar ve Yılmaz, 2024):

$$x_{k+1} = \begin{bmatrix} 0,999678 & 0 \\ 0 & 0,995843 \end{bmatrix} x_k + \begin{bmatrix} 0,022502 \\ -0,022459 \end{bmatrix} u_k + \begin{bmatrix} e_k \\ 0 \end{bmatrix} \quad (3.70)$$

Burada, e_k , $[-10^{-2}, 0]$ aralığında rastgele, bağımsız ve düzgün dağılımlı bir değişkendir.

3.2.4. MPC Sistemi

Benzetim çalışmasında kullanılan MPC sisteminin amaç fonksiyonu, kısıtlara bağlı olmak üzere Denklem (3.71) ile tanımlanmıştır.

$$\begin{aligned} \min_{x,u} V_0 + \frac{\gamma^N}{2} (x_N - x_{ref})^T S_N (x_N - x_{ref}) + \sum_{k=0}^{N-1} f^T \begin{bmatrix} x_k - x_{ref} \\ u_k \end{bmatrix} + \\ \sum_{k=0}^{N-1} \frac{1}{2} \gamma^k \left(\|x_k - x_{ref}\|^2 + \frac{1}{2} \|u_k\|^2 + w^T s_k \right) \end{aligned} \quad (3.71)$$

Kısıtlar, Denklem (3.72)'de gösterildiği gibi tanımlanmıştır:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + b, \\ \begin{bmatrix} -5 \\ 0 \end{bmatrix} + \underline{x} - s_k &\leq x_k \leq \begin{bmatrix} 0 \\ 70 \end{bmatrix} + \bar{x} + s_k \\ 0 &\leq u_k \leq 1, \quad x_0 = s \end{aligned} \quad (3.72)$$

Pekiştirmeli öğrenme yöntemi ile ayarlanan MPC parametreleri, Denklem (3.73)'de gösterildiği gibi tanımlanmıştır:

$$\theta = (V_0, \underline{x}, \bar{x}, b, f, A, B) \quad (3.73)$$

Burada, $w^T = [10 \ 10]$ matrisi, ara aşamalardaki kısıtların gevşetilmesi için kullanılan ENMPC yapısındaki gevşeme değişkenleri σ_k ile ilişkili ağırlığı ifade eder. İndirim faktörü $\gamma=0,7$ olarak tanımlanmıştır. S matrisi, Riccati denkleminin çözümü ile elde edilmiştir. MPC sisteminin referans ufku parametresi $N=7$ olarak seçilmiştir.

Gerçek proses modelinin sistem matrislerinde yapılan değişiklikler ile elde edilen hatalı sistem modeli, Denklem (3.74)'te gösterildiği gibi tanımlanmıştır:

$$A = \begin{bmatrix} 0,799678 & 0 \\ 0 & 0,895843 \end{bmatrix}, \quad B = \begin{bmatrix} 0,012502 \\ -0,012459 \end{bmatrix} \quad (3.74)$$

Başlangıç durumunda, yukarıda tanımlanan parametreler hariç, tüm parametrelere ilk değer olarak sıfır atanmıştır.

V_0 , başlangıç maliyeti parametresi olup, kontrol edilen sistemin başlangıç durumundan itibaren belirli bir maliyetle başladığını ifade eder. Bu terim, sistemin başlangıç performansını veya başlangıç durumunu optimize etmek için kullanılan ek bir maliyet işlevi görür. $\underline{x}$ ve $\bar{x}$, alt ve üst durum vektörleri olup başlangıçta sıfır olarak tanımlanmışlardır. b , model ayar parametresi olarak kullanılır. Bu terim, modelin sistemin gerçek dinamiklerine daha yakın olmasını sağlamak için kullanılır ve modelin doğruluğunu artırmak amacıyla belirli durum geçiş matrisine ilave edilmiştir. f , aşama maliyet gradyanı olarak tanımlanmıştır.

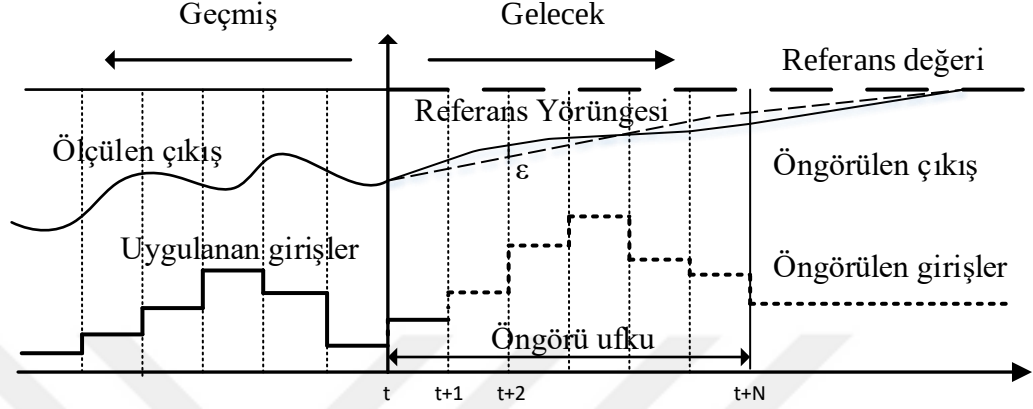
Denklem (3.64), Newton yöntemi ile çözülmüş olup, adım değeri $\alpha = 0,05$ olarak tanımlanmıştır.

3.3. Model Öngörülü Kontrol

Model öngörülü kontrol (MPC) (Richalet, 1976) yaklaşımında, kısıtlı, doğrusal veya doğrusal olmayan sistemler için geri beslemeli optimal kontrol probleminin (OCP) çözülerek sistemin kararlı bir şekilde kontrol edilmesi sağlanır. Şekil 3.3’de gösterildiği gibi, her t anında, sistemin mevcut durumu ve sistem modeli dikkate alınarak öngörü ufku boyunca optimal kontrol problemi çözülür ve $t+1$ anında sisteme uygulanacak olan kontrol girişi bulunur. MPC yaklaşımı, endüstriyel kontrol sistemlerinde verimi ve performansı artırmak amacıyla, özellikle proses endüstrisinde yaygın olarak kullanılmaktadır.

Gerçek zamanlı optimizasyon (RTO) ile optimal geri beslemeli kontrol, proses endüstrisinde bulunan endüstriyel kontrol sistemlerinde uzun yıllardır model öngörülü kontrol (MPC) ismi ile uygulanmaktadır (Diehl ve Gros, 2011). Optimizasyon işlemi genellikle iki adımda gerçekleştirilir (James B Rawlings ve Amrit, 2009). İlk aşamada, ekonomik hedefleri gözeterek kararlı durumları sağlayacak set değerleri bulunur. Bu işleme gerçek zamanlı optimizasyon (RTO) adı verilir. İkinci aşamada, set değerleri

model öngörülü kontrol (MPC) sistemine gönderilir ve dinamik optimizasyon işlemi yapılır. MPC, içerisinde bulunan proses modelini kullanarak sistemin set değerlerine uygun şekilde çalışmasını ve kararlı durumda kalmasını sağlar.

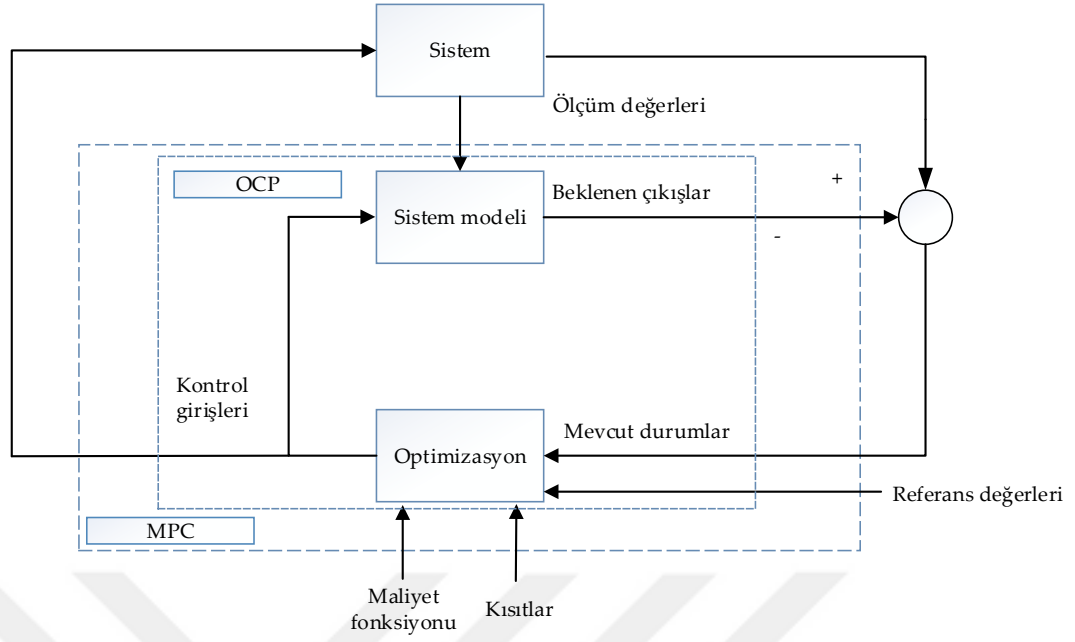


Şekil 3.3. Model öngörülü kontrol yaklaşımı (Richalet vd., 1978)

MPC sistemleri üç temel bileşenden oluşmaktadır:

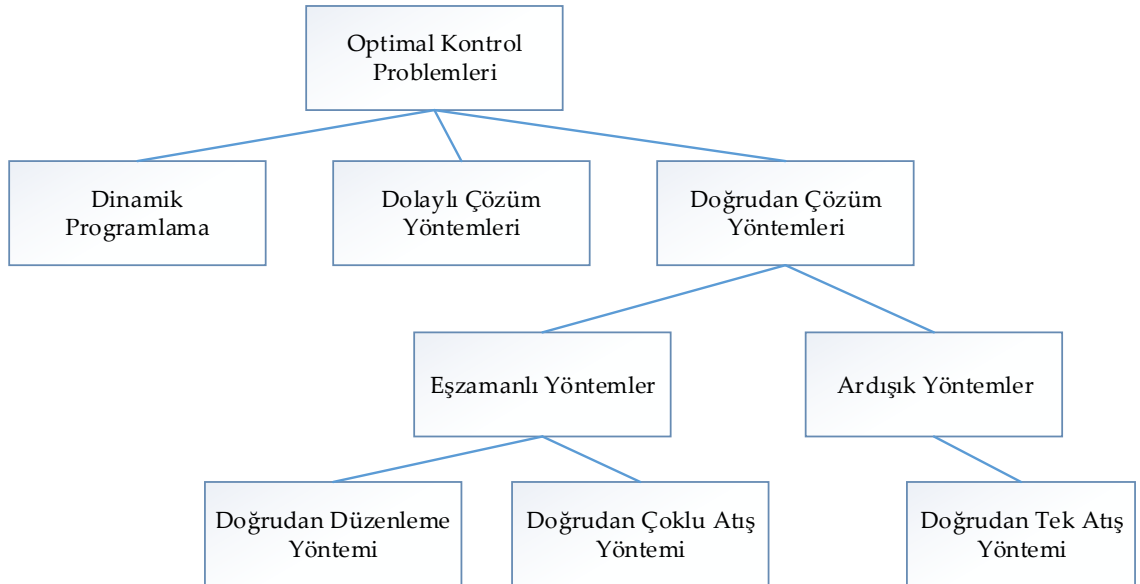
- Sistem modeli, sistemin gelecekteki davranışını tahmin etmek için kullanılan matematiksel modeldir. Bu model, doğrusal veya doğrusal olmayan, sürekli veya ayırık zamanlı olabilir. Modelin doğruluğu, MPC algoritmasının performansı için kritik öneme sahiptir.
- Amaç fonksiyonu, kontrol girişlerinin kalitesini değerlendirmek için kullanılan matematiksel bir fonksiyondur. Genellikle, sistemin istenen referans değerinden sapmasını, kontrol girişlerinin maliyetini ve kısıtlamaların ihlalini en aza indirmeyi hedefler.
- Kontrol kuralı, amaç fonksiyonunu optimize eden (minimize eden veya maksimize eden) kontrol eylemlerinin hesaplanması sürecidir. Bu genellikle bir optimal kontrol problemi (OCP) çözülerek gerçekleştirilir. Kontrol kuralı, sistemin mevcut durumuna ve gelecekteki tahminlerine göre hesaplanır.

Model öngörülü kontrol sisteminin blok şeması, Şekil 3.4'te gösterilmektedir.



Şekil 3.4. Model öngörülü kontrol sistemi

Optimal kontrol problemlerinin yaygın olarak kullanılan çözüm yöntemleri, Şekil 3.5'te gösterilmiş olup; dinamik programlama, doğrudan ve dolaylı yöntemler olmak üzere üç başlıkta incelenir.



Şekil 3.5. Optimal kontrol problemlerinin çözüm yöntemleri (J. Andersson vd., 2012)

Dinamik programlama (DP) yöntemi, ayırık durum ve kontrol uzaylarına sahip sistemlere uygulanır. Sürekli durum uzayına sahip ayırık zamanlı sistemlere ise ayrıklaştırma yoluyla uygulanabilir. Ancak ayrıklaştırma işleminin hesaplama maliyeti, durum uzayının boyutuna bağlı olarak üstel şekilde artar. Bu durum, Bellman'ın boyutsallık laneti olarak bilinir. Dinamik programlama yöntemi, sürekli zamanda ise Hamilton-Jacobi-Bellman denklemi şeklinde ifade edilir.

Dolaylı çözüm yöntemlerinde, öncelikle Pontryagin'in maksimum ilkesi ve matematiksel araçlar kullanılarak optimallik koşullarını analitik olarak belirleyen gerekli koşullar elde edilir. Ardından bu koşulları sağlayan çözümler bulunmaya çalışılır. Problemin çözümü analitik olarak arandığından oldukça hassas sonuçlar verebilmektedir. Bununla beraber, genellikle çözüm sırasında elde edilen diferansiyel denklemlerin kısıtlar altında çözülmesi oldukça zordur.

Doğrudan çözüm yöntemleri, optimal kontrol problemini doğrusal olmayan programlama (NLP) problemine dönüştürerek çözmeye çalışır. Bu yöntemde kısıtlar optimizasyon problemine dahil edilerek çözülebilmektedir. Doğrudan çözüm yöntemleri, yaygın olarak eşzamanlı ve ardışık olmak üzere iki başlıkta incelenir. Yaygın olarak kullanılan yöntemlerden tekli atış yöntemi ardışık; çoklu atış ve eşdizimlilik yöntemleri ise eşzamanlı yöntemlerdir. Doğrusal olmayan programlama (NLP) problemleri, iç nokta ve ardışık kuadratik programlama (SQP) yöntemleri ile çözülebilmektedir.

Model öngörülü kontrol sisteminin tasarlanabilmesi için sistem modeline ihtiyaç duyulmaktadır. Sistem modeli kısmi diferansiyel denklemler ile Denklem (3.75)'deki gibi ifade edilir.

$$\dot{x}(t) = f(t, x(t), u(t)) \quad (3.75)$$

Burada, f durum geçiş denklemlerini, $x \in \mathbb{R}^{n_x}$ durum değişkenlerini, $u \in \mathbb{R}^{n_u}$ kontrol girişlerini göstermektedir. Optimal kontrol girişlerinin performans ölçütüne karşılık gelen maliyet fonksiyonu, genelde giriş ve durum değişkenlerinden oluşur ve Denklem (3.76) ile ifade edilir:

$$J = L(x(t_f)) + \int_{t_0}^{t_f} l(t, x(t), u(t)) dt \quad (3.76)$$

Burada, t_0 başlangıç zamanı ve t_f bitiş zamanı olmak üzere, l , $[t_0, t_f]$ zaman aralığındaki toplam maliyeti, L ise t_f anındaki son durumun maliyetini ifade eder. MPC, her örnekleme periyodunda, Denklem (3.77) ile gösterilen optimizasyon problemini çözerek en uygun kontrol girişini hesaplar:

$$\begin{aligned} \underset{u(t)}{\text{minimize}} \quad & J = L(x(t_f)) + \int_{t_0}^{t_f} l(t, x(t), u(t)) dt \\ \text{olmak üzere, } & \dot{x}(t) = f(t, x(t), u(t)), [t_0, t_f], \\ & h(t, x(t), u(t)) \leq 0, \\ & x(t_0) = x_0 \end{aligned} \quad (3.77)$$

Burada, x_0 , t_0 anındaki sistemin başlangıç durumunu ifade eder. h ile tanımlanan fonksiyon ise giriş ve durumlar ile ilgili kısıtları ifade eder ve Denklem (3.78)'deki gibi gösterilir:

$$\begin{aligned} x_{min} &< x(t) < x_{max} \\ u_{min} &< u(t) < u_{max} \end{aligned} \quad (3.78)$$

Doğrusal model öngörülü kontrol problemleri, ayrık zamanlı doğrusal sistem modelleri, doğrusal kuadratik konveks maliyet fonksiyonları ve doğrusal kısıtlamalar ile ifade edilen konveks problemlerden oluşur.

Doğrusal bir sistemin sürekli zamanlı durum-uzay denklemleri, Denklem (3.79) ve Denklem (3.80)'de gösterildiği gibi tanımlanır:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (3.79)$$

$$y(t) = Cx(t) + Du(t) \quad (3.80)$$

Sistemi oluşturan ifadeler; $x(t) \in \mathbb{R}^n$ durum vektörü, $u(t) \in \mathbb{R}^m$ giriş vektörü ve $y(t) \in \mathbb{R}^p$ ise çıkış vektörüdür.

Burada, $A \in \mathbb{R}^{n \times n}$ durum geçiş matrisi, $B \in \mathbb{R}^{n \times m}$ giriş matrisi, $C \in \mathbb{R}^{p \times n}$ çıkış matrisi ve $D \in \mathbb{R}^{p \times m}$ doğrudan geçiş matrisidir.

$k \in \mathbb{N}$ ayrık zaman indeksi olmak üzere, ayrık zamanlı doğrusal bir sistemin durum-uzayı gösterimi Denklem (3.81) ve Denklem (3.82)'de gösterildiği gibidir:

$$x(k+1) = A(k)x(k) + B(k)u(k) \quad (3.81)$$

$$y(k) = Cx(k) + Du(k) \quad (3.82)$$

Maliyet fonksiyonunun ayrık zamanda gösterimi Denklem (3.83)'teki gibidir:

$$J = x_N^T Q_N x_N + \sum_{k=0}^{N-1} x_k^T Q_k x_k + u_k^T R_k u_k \quad (3.83)$$

Burada, x_k ve u_k durum ve giriş değişkenleridir; N öngörü ufku ifade etmektedir. $Q_k \in \mathbb{R}^{n \times n}$ durum için belirlenen ağırlık katsayısı, $R_k \in \mathbb{R}^{m \times m}$ ise giriş maliyeti ağırlık katsayısıdır.

Maliyet fonksiyonu, sistemin kontrol etmesi istenen amaç fonksiyona göre tanımlanır. Eğer sistemin bir referansı takip etmesi isteniyorsa, maliyet fonksiyonu Denklem (3.84) ile tanımlanabilir.

$$l(x_k, u(k)) = \|x_k - x_{\text{ref}}(k)\|_2 + \|u(k) - u_{\text{ref}}(k)\|_2 \quad (3.84)$$

$$\begin{aligned}
& \min_{\substack{u_0 \dots u_{N-1} \\ x_1 \dots x_N}} \sum_{k=0}^{N-1} l_k(x_k, u_k) + L_N(x_N) \\
& \text{olmak üzere } x_{k+1} = A_k x_k + B_k u_k, k = 0, \dots, N-1 \\
& \underline{x} \leq x_k \leq \bar{x}, k = 0, \dots, N \\
& \underline{u} \leq u_k \leq \bar{u}, k = 0, \dots, N-1 \\
& x_0 = x(0)
\end{aligned} \tag{3.85}$$

Denklem (3.85) ile verilen optimizasyon probleminde, l_k ve L_N fonksiyonlarının tipinin dışbükey olması durumunda, optimizasyon problemi çeşitli arama algoritmalarıyla çözülebilir. Maliyet fonksiyonları ikinci dereceden bir fonksiyon ise, optimizasyon problemi karesel programlama (QP) problemi olarak çözülebilir.

QP problemi, karesel bir maliyet fonksiyonuna ve doğrusal kısıtlamalara sahip bir optimizasyon problemidir. Optimizasyon değişkenlerinin vektörü, aynı zamanda karar değişkenleri olarak da adlandırılır ve $w \in \mathbb{R}^n$ ile gösterilir. En temel formda formüle edilmiş bir QP problemi, Denklem (3.86)'daki gibi tanımlanır:

$$\begin{aligned}
& \min_{w \in \mathbb{R}^n} \frac{1}{2} w^T H w + f^T w \\
& \text{olmak üzere } E_{eq} w = F_{eq} \\
& E w \leq F.
\end{aligned} \tag{3.86}$$

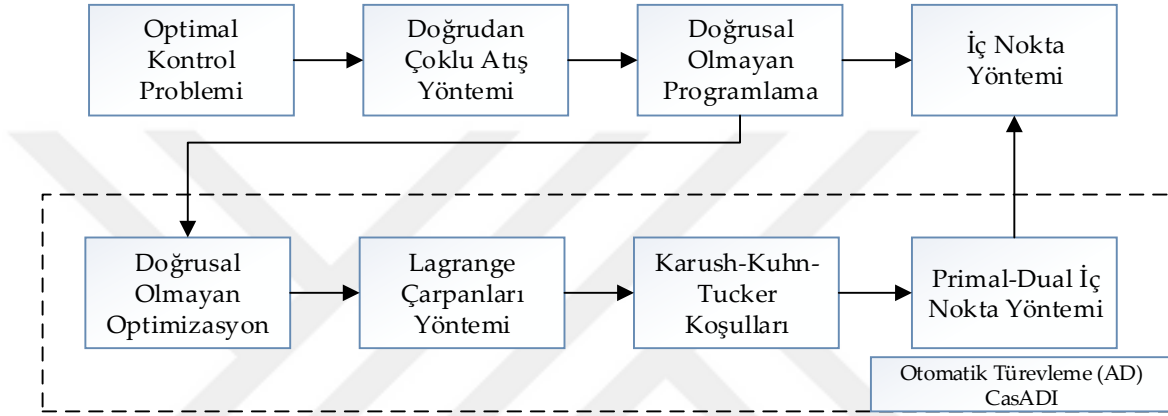
$H \in S_+^{n \times n}$ ve $f \in \mathbb{R}^n$ olmak üzere, amaç fonksiyonu; $E_{eq} \in \mathbb{R}^{p \times n}$ ve $F_{eq} \in \mathbb{R}^p$ olmak üzere, eşitlik kısıtı; $E \in \mathbb{R}^{q \times n}$ ve $F \in \mathbb{R}^q$ olmak üzere, eşitsizlik kısıtı olarak tanımlanır.

Optimizasyon problemi içerisinde doğrusal olmayan sistem dinamikleri, maliyet fonksiyonu veya kısıt bulunması durumunda, model öngörülü kontrol yaklaşımı, doğrusal olmayan model öngörülü kontrol (NMPC) problemi olarak adlandırılır.

MPC sisteminin performansı, geleneksel maliyet fonksiyonundan farklı olarak, doğrudan ekonomik bir maliyet fonksiyonuna göre optimize edilebilir. Bu sistemlere ekonomik MPC (EMPC) sistemi olarak adlandırılır. EMPC sistemi tasarımı; enerji verimliliği, üretim maliyeti veya kar gibi ekonomik hedefler doğrudan kontrol probleminin içerisinde

amaç fonksiyonu olarak tanımlanır. Doğrusal olmayan MPC sistemleri, ekonomik NMPC (ENMPC) olarak adlandırılır.

Tez çalışmasında, ENMPC içerisinde yer alan doğrusal olmayan optimizasyon probleminin (NLP) çözümü için Casadi (J. A. E. Andersson vd., 2019) ve Ipopt (Wächter ve Biegler, 2006) yazılım kütüphaneleri kullanılmıştır. Bu çalışmada yer alan optimal kontrol problemlerinin çözüm adımları Şekil 3.6’da gösterilmektedir.



Şekil 3.6. Optimal kontrol problemlerinin çözüm adımları

3.3.1. Doğrudan Çoklu Atış Yöntemi

Doğrudan çoklu atış yöntemi, sürekli zamanlı optimal kontrol problemlerinin sayısal yöntemler ile çözümünde, kontrol fonksiyonunu belirli zaman aralıklarında parametrik hale getirerek ve zaman ufkunu birden fazla alt aralığa bölerek uygular. Bu yöntem, uzun süreli integrasyonlarda oluşabilecek hata birikimini önleyerek, her alt aralıkta bağımsız integrasyonlar yapılmasına olanak tanır (J. B. Rawlings vd., 2017).

Doğrudan yöntemlerde, kontrol fonksiyonu $u(t)$, sabit bir zaman ızgarası üzerinde parametrik olarak ifade edilir. Bu işlem, kontrol girişinin istenilen zaman boyunca belirli aralıklarla temsil edilmesini, böylece sürekli zamanlı bir problemin sonlu boyutlu bir optimizasyon problemine dönüştürülmesini sağlar. $[0, T]$ zaman aralığı, N adet alt aralığa Denklem (3.87)’de ifade edildiği şekilde bölünecek olursa;

$$0 = t_0 < t_1 < \dots < t_N = T \quad (3.87)$$

bu noktalar, kontrol fonksiyonunun tanımlanacağı zaman adımlarını belirler. Genellikle zaman aralıkları eşit seçilir ve kontrol fonksiyonunu parçalı sabit bir fonksiyon olarak Denklem (3.88) ile temsil edilir:

$$u(t) = q_i, \quad t \in [t_i, t_{i+1}] \quad (3.88)$$

Burada, q_i , her bir zaman aralığında sistemi optimize edecek olan kontrol parametresidir. Her bir $[t_i, t_{i+1}]$ alt zaman aralığı için, yapay başlangıç değeri s_i olmak üzere, Denklem (3.89)'da verilen adi diferansiyel denklem çözülür.

$$\begin{aligned} \dot{x}_i(t, s_i, q_i) &= f(x_i(t, s_i, q_i), q_i), \quad t \in [t_i, t_{i+1}], \\ x_i(t_i, s_i, q_i) &= s_i. \end{aligned} \quad (3.89)$$

Her adımda $x_i(t_i, s_i, q_i)$ yörünge parçaları elde edilir. Aynı şekilde, nümerik integraller Denklem (3.90) ile hesaplanır:

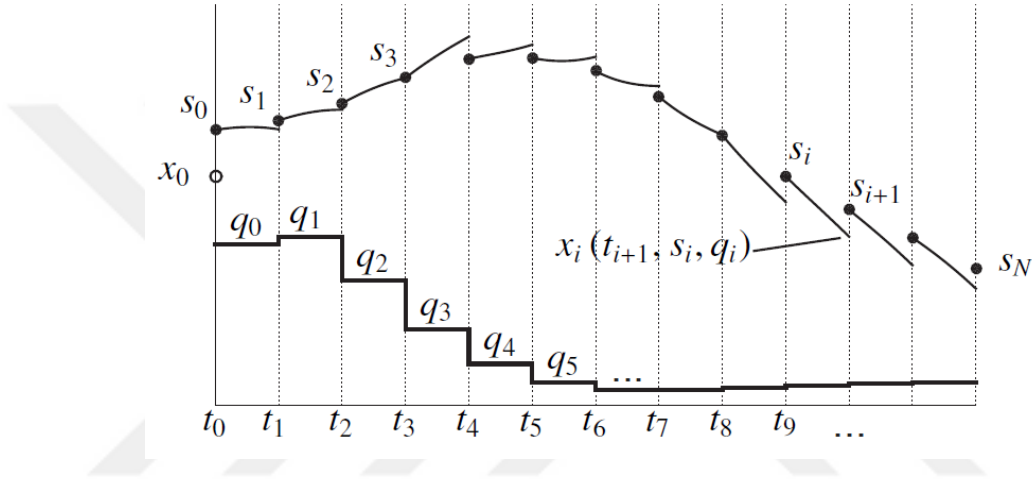
$$l_i(s_i, q_i) := \int_{t_i}^{t_{i+1}} L(x_i(t, s_i, q_i), q_i) dt. \quad (3.90)$$

Yörünge parçalarının birleştirilme işlemi, doğrusal olmayan programlama uygulamasına $s_{i+1} = x_i(t_{i+1}, s_i, q_i)$ süreklilik şartı ilave edilerek sağlanır. Bu süreklilik koşulu, çoklu atış yöntemi ile elde edilen tüm zaman aralıklarındaki alt çözümlerin tutarlı bir şekilde birleşmesini sağlar. Yani, her bir alt aralık için çözüm sonunda elde edilen durum $x_i(t_{i+1}, s_i, q_i)$, bir sonraki alt aralık için başlangıç durumu s_{i+1} 'e eşit olmalıdır.

Çoklu atış yöntemine dayalı bir optimal kontrol probleminin ayrıklaştırılmasından elde edilen doğrusal olmayan programlama problemi, Denklem (3.91)'daki gibi ifade edilir (Diehl ve Gros, 2011):

$$\begin{aligned}
& \min_{s,q} \sum_{i=0}^{N-1} l_i(s_i, q_i) + E(s_N) \\
& \text{olmak üzere} \quad \begin{aligned} & x_0 - s_0 = 0, \\ & x_i(t_{i+1}, s_i, q_i) - s_{i+1} = 0, \quad i = 0, \dots, N-1 \\ & h(s_i, q_i) \leq 0, \quad i = 0, \dots, N \\ & r(s_N) \leq 0 \end{aligned}
\end{aligned} \tag{3.91}$$

Doğrudan çoklu atış yönteminin grafiksel gösterimi Şekil 3.7’de gösterilmektedir.



Şekil 3.7. Doğrudan çoklu atış yönteminin gösterimi (Diehl ve Gros, 2011)

3.3.2. Doğrusal Olmayan Optimizasyon

Optimal kontrol teorisi, dinamik bir sistemin belirli bir süre boyunca hedef fonksiyonunu optimize edecek bir kontrol stratejisi bulmayı amaçlayan bir disiplindir (Ross, 2009). Bu teori, sistemin mevcut durumunu ve dinamiklerini dikkate alarak, istenen hedefe en verimli ve etkili şekilde ulaşmak için gerekli olan kontrol girişlerini belirlemeye odaklanır. Optimal kontrol teorisi, çeşitli mühendislik ve ekonomi uygulamalarında, sistem performansını en üst düzeye çıkarmak veya maliyetleri en aza indirmek için kullanılır.

Eniyileme veya optimizasyon, belirli kısıtlar altında bir hedef fonksiyonun minimum veya maksimum değerini bulma amacıyla gerçekleştirilen sistematik bir süreçtir. Hedef

fonksiyonun parametreleri, gerçek veya tam sayı değeri olacak şekilde tanımlı bir aralıkta seçilir ve en uygun çözüm aranır.

Tarihsel olarak optimizasyon, programlama kavramı ile özdeşleşmiştir. Burada program, lojistikte olduğu gibi, deterministik plan anlamında kullanılmaktadır. Optimizasyon problemlerinde programlama, belirli bir hedefe ulaşmak için kaynakları ve süreçleri önceden planlama anlamına gelir.

Doğrusal olmayan programlar genel olarak Denklem (3.92)'deki gibi ifade edilir (Diehl ve Gros, 2011):

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & f(w) \\ \text{olmak üzere} \quad & g(w) = 0 \\ & h(w) \leq 0 \end{aligned} \quad (3.92)$$

Burada, $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $g: \mathbb{R}^n \rightarrow \mathbb{R}^{n_g}$ ve $h: \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ iki kez türevlenebilir fonksiyonlardır. f amaç fonksiyonu, g eşitlik kısıtları vektörü, h ise eşitsizlik kısıtları vektörüdür.

Öncelikle, tüm kısıtları tek noktada toplayan uygulanabilirlik kümesi Ω , Denklem (3.93)'te gösterildiği gibi tanımlanacak olursa:

$$\Omega := \{w \in \mathbb{R}^n \mid g(w) = 0, h(w) \leq 0\} \quad (3.93)$$

Uygulanabilirlik kümesi, optimizasyon probleminin çözümü için potansiyel aday olan noktaları içerir. Optimal çözüm, bu kümenin içindeki en iyi noktadır.

Optimizasyonda ilgilenilen noktalar, amaç fonksiyonunu minimize eden uygulanabilir noktalardır. Bu noktalar global ve lokal minimum olarak ikiye ayrılır. $w^* \in \mathbb{R}^n$ noktasının global minimizasyonu sağlayabilmesi, ancak ve ancak $w^* \in \Omega$ ve $\forall w \in \Omega: f(w) \geq f(w^*)$ koşulu sağlanabiliyorsa mümkündür. Burada, $f(w^*)$ değeri global minimum olarak adlandırılır.

Global minimum noktasını bulmak genellikle zordur ve çoğu algoritma sadece yerel minimum noktasının bulunmasını ve optimal koşulların yerel olarak doğrulanmasını sağlar. $w^* \in \Omega$ ve w^* 'in bir komşuluğu $\mathcal{N}$ varsa, bu durumda $\forall w \in \Omega \cap \mathcal{N}: f(w) \geq f(w^*)$ koşulu sağlanıyorsa, $w^* \in \mathbb{R}^n$ noktası lokal minimizasyonu sağlar. Burada $f(w^*)$ değeri lokal minimum olarak adlandırılır.

Eğer Denklem (3.92)'de bulunan kısıtlar g ve h afin ise ve amaç fonksiyonu doğrusal kuadratik bir fonksiyon ise, meydana gelen problem kuadratik optimizasyon problemi veya kuadratik program (QP) olarak adlandırılır ve Denklem (3.94) ile ifade edilir:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & c^\top w + \frac{1}{2} w^\top B w \\ \text{olmak üzere} \quad & A w - b = 0, \\ & C w - d \leq 0. \end{aligned} \tag{3.94}$$

Burada, problemi oluşturan bileşenler: $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{n_g \times n}$, $b \in \mathbb{R}^{n_g}$, $C \in \mathbb{R}^{n_h \times n}$, $d \in \mathbb{R}^{n_h}$ ve Hessian matrisi $B \in \mathbb{R}^{n \times n}$ 'dir. Hessian matrisi ismini, $f(w) = c^\top w + \frac{1}{2} w^\top B w$ fonksiyonu için, ikinci türev matrisi olan $\nabla^2 f(w) = B$ 'den alır.

- $c^\top w$, doğrusal terimi ifade eder,
- $\frac{1}{2} w^\top B w$, kuadratik terimi ifade eder,
- B , amaç fonksiyonunun Hessian matrisi olup, amaç fonksiyonunu ikinci türevini ifade eder.

B matrisinin özdeğerleri, kuadratik programın konveks olup olmadığını belirler; yani, problemin polinom zamanda global optimal çözüme ulaşp ulaşamayacağı bu özdeğerlere bağlıdır. Eğer $B \leq 0$ ise, buna konveks QP denir. Eğer $B \succ 0$ (pozitif tanımlı) ise, buna katı konveks QP denir.

Katı konveks QP sınıfının özelliği, her zaman tekil minimuma sahip olmasıdır. Başka bir deyişle, bu tür problemlerde çözüm benzersizdir ve global optimal çözüme ulaşılması

kesindir. Konvekslik, optimizasyon problemlerinin çözülebilirliği ve optimal çözümün global olup olmadığı açısından kritik bir rol oynar.

Bir $\Omega \subset \mathbb{R}^n$ kümesinin konveks olabilmesi için Denklem (3.95) ile ifade edilen şartı sağlaması gerekir:

$$\forall x, y \in \Omega, t \in [0,1]: x + t(y - x) \in \Omega \quad (3.95)$$

Bir fonksiyon, tüm sekantlar grafiğın üzerinde kalıyorsa konveks olarak adlandırılır. Bir $f: \Omega \rightarrow \mathbb{R}$ fonksiyonunun konveks olabilmesi için, Ω 'nın konveks olması ve Denklem (3.96) ile ifade edilen şartı sağlaması gerekir:

$$\forall x, y \in \Omega, t \in [0,1]: f(x + t(y - x)) \leq f(x) + t(f(y) - f(x)) \quad (3.96)$$

Uygulanabilir Ω kümesi ve amaç fonksiyonu $f: \Omega \rightarrow \mathbb{R}$ konveks olan optimizasyon problemine konveks optimizasyon problemi denir. Konveks optimizasyon probleminde, her lokal minimum aynı zamanda global minimumdur.

3.3.3. Karush-Kuhn-Tucker Koşulları

Sürekli optimizasyonda incelenen önemli konulardan biri de, uygulanabilir bir nokta olan $w^* \in \Omega$ 'nin birinci dereceden gerekli optimallik koşullarını sağlayabilmesidir. Eğer bu koşulları sağlayamıyorsa, w^* , yerel minimizasyon sağlayamaz. Ancak bu koşulları sağlıyorsa, w^* yerel minimizasyon için güçlü bir adaydır. Eğer problem konveks ise, bu koşullar global optimal çözümün bulunmasını garanti etmek için yeterlidir. Bu nedenle, doğrusal olmayan optimizasyon algoritmalarının çoğu bu tür noktaları arar. Birinci dereceden optimallik koşulları, çözümün geçerli olduğunu ve doğru şekilde bulunabileceğini gösteren koşulları sağlar. Bu şartlar sağlanmadığında, çözüm sırasında belirsizlikler ortaya çıkar.

Karush-Kuhn-Tucker (KKT) koşulları, kısıtlı optimizasyon problemleri için birinci dereceden gerekli koşullarıdır. Bu koşullar, doğrusal olmayan programlama

problemlerinin çözümlerinin optimal olabilmesi için gerekli koşulları sağlar. Lagrange çarpanları yöntemi ile sadece eşitlik kısıtlarını içeren doğrusal olmayan programlama problemleri çözülebilmektedir. KKT koşulları ile bu yöntemin eşitsizlik kısıtlarını içerecek şekilde genişletilmesi sağlanır.

Lineer bağımsızlık kısıt yeterliliği (LICQ), ancak ve ancak $w^* \in \Omega$ noktasında, $i \in \{1, \dots, n_g\}$ için tüm $\nabla g_i(w^*)$ vektörleri ve $i \in \mathcal{A}(w^*)$ için tüm $\nabla h_i(w^*)$ vektörlerinin doğrusal olarak bağımsız olması durumunda sağlanır. LICQ koşuluna daha derin bir anlam kazandırmak amacıyla, tüm aktif eşitsizlik kısıtları ve eşitlik kısıtlarını birleştirerek, bu fonksiyonları tek bir sütun vektörü içerisinde toplamak mümkündür. Bu bağlamda, $\tilde{g}(w)$ fonksiyonu, Denklem (3.97)'de gösterildiği gibi tanımlanır:

$$\tilde{g}(w) = \begin{bmatrix} g(w) \\ h_i(w) (i \in \mathcal{A}(w^*)) \end{bmatrix} \quad (3.97)$$

LICQ, Jacobian matrisi $\frac{\partial \tilde{g}}{\partial w}(w^*)$ 'nin tam satır derecesine sahip olmasına eşdeğer olarak ifade edilebilir. Bu koşul, Karush-Kuhn-Tucker koşullarının formüle edilmesini sağlar.

Eğer w^* , Denklem (3.92) ile gösterilen NLP probleminin yerel bir minimum noktası ise ve w^* noktasında LICQ şartı sağlanıyorsa, o halde $\lambda \in \mathbb{R}^{n_g}$ ve $\mu \in \mathbb{R}^{n_h}$ olarak adlandırılan çarpan vektörlerinin varlığı söz konusudur. Bu vektörler Denklem (3.98)'de bulunan koşulları sağlar:

$$\begin{aligned} \nabla f(w^*) + \nabla g(w^*)\lambda^* + \nabla h(w^*)\mu^* &= 0 \\ g(w^*) &= 0 \\ h(w^*) &\leq 0 \\ \mu^* &\geq 0 \\ \mu_i^* h_i(w^*) &= 0, i = 1, \dots, n_h \end{aligned} \quad (3.98)$$

Durağanlık koşulu, amaç fonksiyonu ve kısıtlar birleştirilerek elde edilen Lagrange fonksiyonunun w^* 'ya göre türevinin sıfır olmasıdır ve Denklem (3.99) ile gösterilir:

$$\nabla_w L(w^*, \lambda, \mu) = 0 \quad (3.99)$$

Burada $L(w^*, \lambda, \mu)$, Denklem (3.100) ile ifade edilen Lagrange fonksiyonunu ifade eder;

$$L(w, \lambda, \mu) = f(w) + \sum_{i=1}^{n_g} \lambda_i g_i(w) + \sum_{i=1}^{n_h} \mu_i h_i(w) \quad (3.100)$$

Eşitlik koşulları, $g(w^*) = 0$ ile ifade edilir. Tamamlayıcılık koşulları, eşitsizlik kısıtları ve tamamlayıcı gevşeklik koşulları ile ifade edilir. Bu koşullar, aktif ve pasif eşitsizlik kısıtlarını tanımlar ve tamamlayıcı gevşeklik koşulunu ifade eder. Yani, bir kısıt aktif ise ($h_i(w^*) = 0$), ilgili Lagrange çarpanı (μ_i) pozitif olacaktır; pasif ise ($h_i(w^*) < 0$), $\mu_i = 0$ olacaktır.

Bu teorem, optimizasyon problemlerinde Karush-Kuhn-Tucker (KKT) koşulları olarak bilinen gerekli optimallik koşullarını sağlar. LICQ koşulu sağlandığında, bu çarpanlar kullanılarak optimal çözüm bulunabilir ve bu çözüm noktalarının doğruluğu garanti edilir.

3.3.4. Lagrange Çarpanları Yöntemi

Lagrange çarpanları yöntemi, doğrusal olmayan MPC sistemlerinde, parametrik doğrusal olmayan programlama problemlerinin gerçek zamanlı optimizasyonu ve duyarlılık analizinde kullanılmaktadır (Büskens ve Maurer, 2001; Gros ve Zanon, 2020).

Başlangıç koşulu x_0 olan doğrusal olmayan ayrık zamanlı dinamik sistem dinamikleri Denklem (3.101)'de gösterildiği gibi tanımlanır (Lewis vd., 2012):

$$x_{k+1} = f^k(x_k, u_k) \quad (3.101)$$

Burada, f^k , zamanla değişen dinamikleri; durum vektörü x_k , boyutu n olan bir vektör; ve kontrol girişleri u_k , boyutu m olan bir vektördür. Denklem (3.101), k anındaki durumu ve

giriş göz önünde bulundurarak $k+1$ anındaki durumu belirlediği için bir kısıt ifade eder. f , n adet fonksiyondan oluşan bir vektördür.

Mayer-Lagrange performans indeksinin genel ifadesi, Denklem (3.102)'deki gibi tanımlanır,

$$J_i = \phi(N, x_N) + \sum_{k=i}^{N-1} L^k(x_k, u_k) \quad (3.102)$$

Burada, $[i, N]$ sabit bir örnekleme zamanı ile ayrık zamanlı olarak örneklenen sistemi kontrol etmek istediğimiz zaman aralığını belirler. $\phi(N, x_N)$, son adımda bulunan durum ve son adıma bağlı bir fonksiyonu ifade eder, Mayer terimi olarak da adlandırılır. $L^k(x_k, u_k)$ fonksiyonu, $[i, N]$ arasındaki her adımı ifade eden k anına ve girişe bağlı zamanla değişen bir fonksiyondur, Lagrange terimi olarak da adlandırılır.

Optimal kontrol problemi, kontrol edilmek istenen sistemi, Denklem (3.78)'e uygun olarak ve performans indeksi Denklem (3.102)'nin değerini optimize edecek şekilde, x_k^* yörüngesi boyunca yönlendiren $u_k^*, \forall k \in [i, N]$ kontrol girişlerinin bulunmasıdır.

Performans indeksi, minimum enerji probleminin çözümü için, Denklem (3.103)'de belirtildiği gibi yazılabilir:

$$J = \frac{1}{2} x_N^T S x_N + \frac{1}{2} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) \quad (3.103)$$

Burada, Q , R ve S ağırlık matrisleri olup, $\phi = \frac{1}{2} x_N^T S x_N$ ve $L = \frac{1}{2} (x_k^T Q x_k + u_k^T R u_k)$ kuadratik fonksiyonlardır.

Problemin çözümü için Lagrange çarpanları yönteminden faydalanılır. $\lambda_k \in R^n$ olmak üzere, Denklem (3.101) ile ifade edilen kısıt, Denklem (3.102)'ye eklenerek genişletilmiş performans indeksi Denklem (3.104)'te gösterildiği gibi yazılır:

$$J' = \phi(N, x_N) + \sum_{k=i}^{N-1} [L^k(x_k, u_k) + \lambda_{k+1}^T (f^k(x_k, u_k) - x_{k+1})] \quad (3.104)$$

Hamiltonian fonksiyonu, Denklem (3.105)'deki gibi tanımlanarak,

$$H^k(x_k, u_k) = L^k(x_k, u_k) + \lambda_{k+1}^T f^k(x_k, u_k) \quad (3.105)$$

genişletilmiş performans indeksi Denklem (3.106)'daki gibi yazılabilir:

$$J' = \phi(N, x_N) - \lambda_N^T x_N + H^i(x_i, u_i) + \sum_{k=i+1}^{N-1} [H^k(x_k, u_k) - \lambda_k^T x_k] \quad (3.106)$$

Ardından denklemde bulunan λ_k , x_k ve u_k değişkenlerindeki artışa bağlı olarak J' fonksiyonunda meydana gelecek artış incelenir. Lagrange çarpanları teoremine göre, kısıtlanmış bir minimum noktasında dJ' değerinin sıfır olması gerekmektedir ve Denklem (3.107) ile tanımlanır.

$$\begin{aligned} dJ' = & (\phi_{x_N} - \lambda_N)^T dx_N + (H_{x_i}^i)^T dx_i + (H_{u_i}^i)^T du_i \\ & + \sum_{k=i+1}^{N-1} [(H_{x_k}^k - \lambda_k)^T dx_k + (H_{u_k}^k)^T du_k] + \sum_{k=i+1}^N (H_{\lambda_k}^{k-1} - x_k)^T d\lambda_k \end{aligned} \quad (3.107)$$

Denklemin içerisindeki katsayıların, toplamaların, Denklem (3.108) ve Denklem (3.109) ile gösterilen eşitliklerin sıfır olması gerekmektedir.

$$\left(\frac{\partial \phi}{\partial x_N} - \lambda_N \right)^T dx_N = 0 \quad (3.108)$$

$$\left(\frac{\partial H^i}{\partial x_i} \right)^T dx_i = 0 \quad (3.109)$$

Buradan, kısıtlanmış bir minimum için gerekli koşullar, Denklem (3.110) ile Denklem (3.112) arasındaki gibi bulunur ve bu denklemlere durağanlık koşulları adı verilir.

$$x_{k+1} = \frac{\partial H^k}{\partial \lambda_{k+1}}, k = i, \dots, N - 1 \quad (3.110)$$

$$\lambda_k = \frac{\partial H^k}{\partial x_k}, k = i, \dots, N - 1 \quad (3.111)$$

$$0 = \frac{\partial H^k}{\partial u_k}, k = i, \dots, N - 1 \quad (3.112)$$

Doğrusal olmayan sistemler için genel performans indekslerine sahip optimal kontrol probleminin çözümü, Denklem (3.113) ile Denklem (3.119) arasındaki ifadeler ile sağlanır (Lewis vd., 2012).

Sistem modeli:

$$x_{k+1} = f^k(x_k, u_k), k > i \quad (3.113)$$

Performans indeksi:

$$J_i = \phi(N, x_N) + \sum_{k=i}^{N-1} L^k(x_k, u_k) \quad (3.114)$$

Hamiltonian:

$$H^k = L^k + \lambda_{k+1}^T f^k \quad (3.115)$$

Optimal kontrol durum denklemi:

$$x_{k+1} = \frac{\partial H^k}{\partial \lambda_{k+1}} = f^k(x_k, u_k) \quad (3.116)$$

Eşdurum denklemi:

$$\lambda_k = \frac{\partial H_k}{\partial x_k} = \left(\frac{\partial f^k}{\partial x_k} \right)^T \lambda_{k+1} + \frac{\partial L^k}{\partial x_k} \quad (3.117)$$

Durağanlık koşulu:

$$0 = \frac{\partial H^k}{\partial u_k} = \left(\frac{\partial f^k}{\partial u_k} \right)^T \lambda_{k+1} + \frac{\partial L^k}{\partial u_k} \quad (3.118)$$

Sınır koşulları:

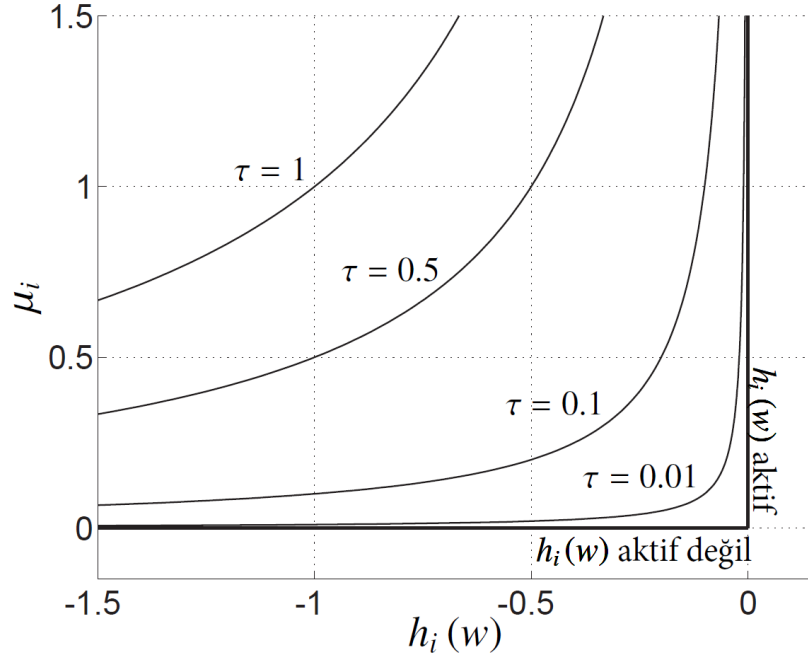
$$\begin{aligned} \left(\frac{\partial L^i}{\partial x_i} + \left(\frac{\partial f^i}{\partial x_i} \right)^T \lambda_{i+1} \right)^T dx_i &= 0 \\ \left(\frac{\partial \phi}{\partial x_N} - \lambda_N \right)^T dx_N &= 0 \end{aligned} \quad (3.119)$$

3.3.5. İç Nokta Yöntemleri

İç nokta yönteminin temel fikri, tamamlayıcılık koşullarından kaynaklanan köşeli L şeklindeki yapıyı, genellikle hiperbol olan köşesiz bir yaklaşımla değiştirmektir. Bu amaçla, düzgünleştirme sabiti $\tau > 0$ tanıtılır ve KKT koşulları yerine köşesiz bir denklem sistemi kullanılır. Bu denklem sistemi Denklem (3.120)'deki gibi ifade edilir:

$$\begin{aligned} \nabla f(w^*) + \nabla g(w^*)\lambda^* + \nabla h(w^*)\mu^* &= 0 \\ g(w^*) &= 0 \\ \mu_i^* h_i(w^*) + \tau &= 0, i = 1, \dots, n_h. \end{aligned} \quad (3.120)$$

Tamamlayıcı gevşeklik koşullarının esnetilmesi, Şekil 3.8’de gösterilmektedir.



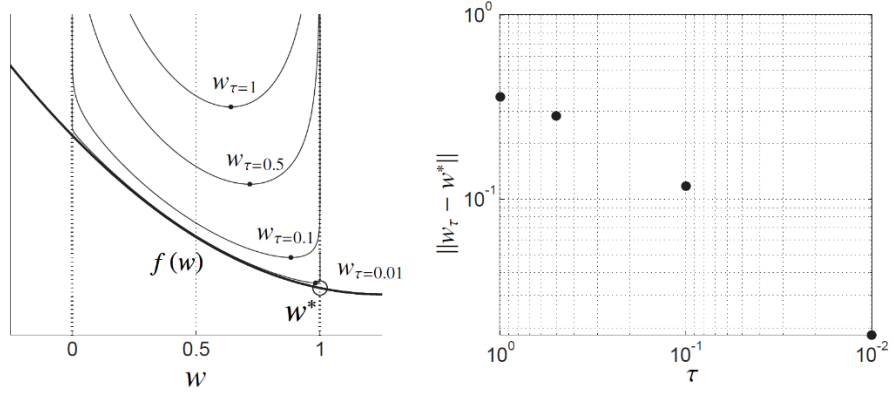
Şekil 3.8. Tamamlayıcı gevşeklik koşulunun esnetilmesi (Diehl ve Gros, 2011)

İç nokta yönteminde, genellikle τ değerine büyük bir başlangıç değeri verilir ve bu değere bağlı olarak elde edilen doğrusal olmayan denklem sistemi, Newton yöntemi ile çözülür. Daha sonra, τ değeri iteratif olarak küçültülür ve her seferinde, bir önceki iterasyonda elde edilen çözüm, bir sonraki iterasyon için başlangıç değeri olarak kullanılır.

Denklem (3.120) ile verilen KKT koşulları yorumlanırken, son koşul kullanarak $\mu_i^* = -\frac{\tau}{h_i(w^*)}$ ifadesi bulunur ve ilk denkleme yerleştirilir. Burada, $\nabla_w(\log(-h_i(w))) = \frac{1}{h_i(w)} \nabla h_i(w)$ olduğu göz önüne alındığında, düzgünleştirilmiş KKT koşullarının aslında bariyer problemine ait optimallik koşulları olduğu görülebilir ve Denklem (3.121)’de belirtildiği gibi ifade edilir.

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & f(w) - \tau \sum_{i=1}^{n_h} \log(-h_i(w)) \\ \text{olmak üzere} \quad & g(w) = 0 \end{aligned} \quad (3.121)$$

Bariyer problemine ait grafiksel gösterim Şekil 3.9’da gösterilmektedir.



Şekil 3.9. Bariyer problemine ait grafiksel gösterim (Diehl ve Gros, 2011)

Doğrusal olmayan iç nokta yönteminin başarılı uygulama örneklerinden, açık kaynaklı IPOPT (Wächter ve Biegler, 2009, 2006) kütüphanesi, doğrusal olmayan optimizasyon problemlerinin çözümünde, endüstriyel ve akademik çalışmalarda yaygın olarak kullanılmaktadır.

3.3.6. Sistem Modeli

Zamana bağlı olmayan doğrusal (LTI) dinamik bir sistem, durum uzayı modeli ile ayrık zamanlı olarak, Denklem (3.122)’deki gibi ifade edilsin (J. B. Rawlings vd., 2017):

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \\ x(0) &= x_0 \end{aligned} \tag{3.122}$$

Burada, $x \in \mathbb{R}^n$ durum vektörü, $u \in \mathbb{R}^m$ giriş vektörü ve $y \in \mathbb{R}^p$ çıkış vektörü olmak üzere; $A \in \mathbb{R}^{n \times n}$ durum geçiş matrisi, $B \in \mathbb{R}^{n \times m}$ giriş matrisi, $C \in \mathbb{R}^{p \times n}$ çıkış matrisi ve $D \in \mathbb{R}^{p \times m}$ doğrudan iletim matrisidir. Çoğu uygulamada $D = 0$ olarak kabul edilir. Sistem durumu ölçülebilir ve $C = I$ olarak kabul edilmiştir. Sistemin başlangıç durumunun bilindiği ve deterministik olduğu kabul edilmiştir.

$k = 0, 1, \dots, N - 1$ olmak üzere, N ufku boyunca sistemin benzetimi Deklem (3.123)’teki gibi gösterilebilir (Diehl ve Gros, 2011):

$$f_{\text{sim}}(x_0; u_0, \dots, u_{N-1}) = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_0 \\ Ax_0 + Bu_0 \\ A^2x_0 + ABu_0 + Bu_1 \\ \vdots \\ A^Nx_0 + \sum_{k=0}^{N-1} A^{N-1-k}Bu_k \end{bmatrix} \quad (3.123)$$

Doğrusal bir sistemin başlangıç durumundan herhangi bir duruma N adımda gidebilmesi için Denklem (3.124) ile gösterilen eşitliğin sağlanması gerekir:

$$x_N = \underbrace{[A^{N-1}B \quad A^{N-2}B \quad \dots \quad B]}_{=\mathcal{C}_N} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \quad (3.124)$$

Kontrol edilebilirlik matrisi $\mathcal{C}_N \in \mathbb{R}^{n_x \times Nn_u}$ ile gösterilir ve derecesi n_x 'e eşit ise sistem kontrol edilebilir olarak kabul edilir.

3.4. Sistem Tanımlama

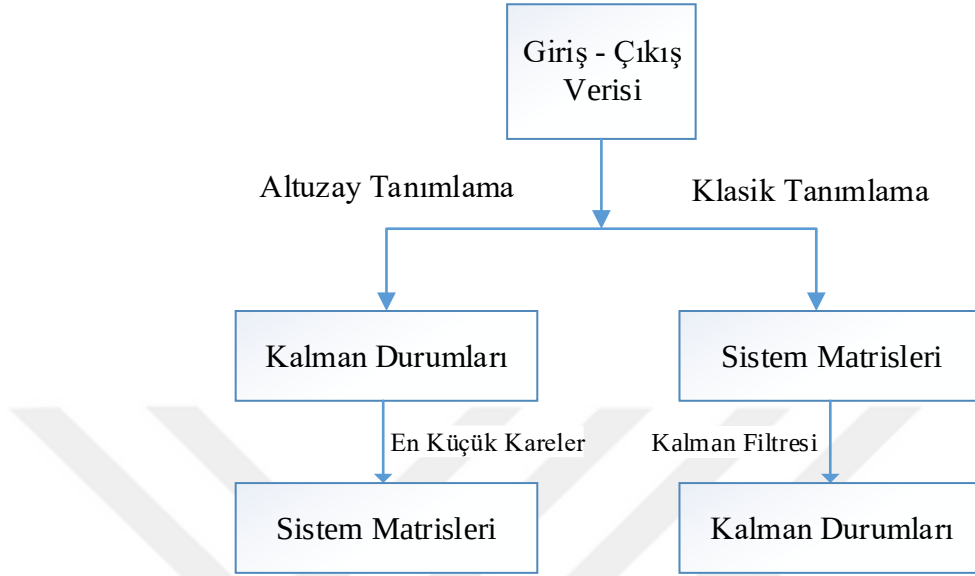
Sistem tanımlama terimi ilk olarak Zadeh (1956) tarafından, dinamik sistemlerin modellerinin tahmin edilmesi amacı ile kontrol çalışma grubunda kullanılmıştır. Sistem tanımlama yöntemlerinin geliştirilmesi sırasında iki farklı yol izlendiği görülmektedir:

- Alt uzay yöntemleri,
- Tahmin hatası yaklaşımı

Alt uzay yöntemleri, birim darbe cevabını kullanarak doğrusal durum uzay modellerinin bulunmasını amaçlar. Ho ve Kalman (1966), Akaike (1976), Larimore (1983), Van Overschee ve DeMoor (1996) bu yönteme ismini veren çalışmalar yapmışlardır.

Tahmin hatası yaklaşımı ise daha çok istatistiksel zaman serileri ve ekonometri çizgisindedir. Bu yaklaşım ve türevleri, Åström ve Bohlin (1965) tarafından yapılan

çalışmaya dayanmaktadır. Aynı zamanda Ljung (1999) temel bakış açısı olarak bu yöntemi kullanmaktadır. Sistem tanımlama yöntemleri Şekil 3.10'da gösterilmektedir.



Şekil 3.10. Sistem tanımlama yöntemleri (Van Overschee ve De Moor, 1994)

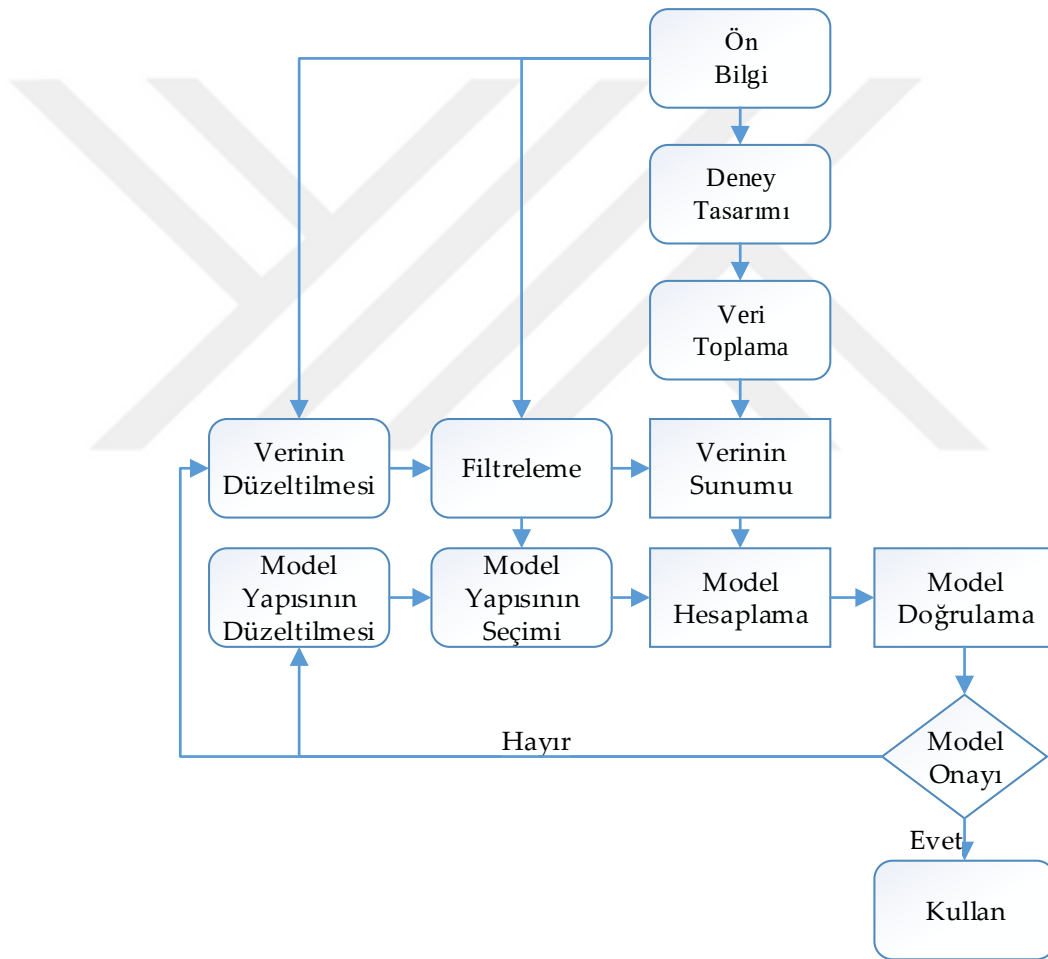
Sistem tanımlama, giriş-çıkış verisi kullanılarak sistem modelinin elde edilmesini amaçlar. Alt uzay tanımlama yaklaşımında, durumlar (Kalman filtresi) giriş ve çıkış verileri kullanılarak elde edildikten sonra sistem matrisleri elde edilmektedir. Klasik sistem tanımlama yönteminde (PEM) ise öncelikle sistem matrisleri bulunur ve ardından Kalman durumları elde edilir.

Sistem tanımlama, dinamik sistemlerin deneysel veriler ve istatistiksel yöntemler kullanılarak matematiksel olarak modellenmesidir (Söderström ve Stoica, 1989). Dinamik sistemlerin temel özelliği, sistemin gelecekteki durumunun geçmişteki verilere bağlı olmasıdır (Ljung, 2008).

Matematiksel model, gerçek bir sistemin belirli bir amaç için uygun olan özelliklerinin bir tanımıdır. Sistem tanımlama, dinamik sistemlerin belirli kullanım alanları için matematiksel modellerini türetmeyi amaçlayan analitik bir süreçtir. Sistem tanımlama yöntemleri, gerçek sistemden toplanan giriş-çıkış verilerini kullanarak dinamik sistemler için matematiksel modeller oluşturmakta kullanılır. Klasik sistem tanımlama döngüsü

Şekil 3.11'de gösterilmektedir. Dikdörtgen kutularla temsil edilen görevler bilgisayar tarafından, oval şekillerle temsil edilenler ise kullanıcı tarafından manuel olarak gerçekleştirilir.

Sistem tanımlama, bir sistem için sonlu sayıda sayısal değer veya katsayı içeren bir model bulmayı amaçlamaktadır. Genellikle, tüm katsayılar doğrudan sistemin davranışının fiziksel mekanizmalarından belirlenemez. Bazı katsayılar, model yapılarına ait bilinmeyen parametreler olarak matematiksel modele dahil edilmelidir.



Şekil 3.11. Sistem tanımlama döngüsü (Ljung, 1999)

3.4.1. Sistem Tanımı

Bu çalışmada, modellenen inkübatör cihazı, doğrusal zamanla değişmeyen tek giriş-tek çıkışlı (SISO) kara-kutu bir sistem olarak ele alınmıştır. Örneklenmiş-veri sistemi,

sistemin dürtü yanıtı ve ayrık zamanlı giriş-çıkış sinyalleri ile Denklem (3.125)'te ifade edildiği şekilde tanımlanabilir.

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) \cdot t = 0,1,2, \dots \quad (3.125)$$

Uygulamada, gerçek sistemler bozucu çevresel faktörlere maruz kalır. Ölçüm ekipmanları, donanım ve yazılım filtrelerine rağmen sinyalleri gürültüsüz okuyamaz. Veri toplama cihazı, örneklenen değerleri sıfır gecikme ile kaydedemez. Sistem, kontrol edilemeyen bozucu etkilere maruz kalır. Toplamsal bozulma sinyali içeren doğrusal bir model Denklem (3.126)'daki gibi temsil edilebilir:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t) \quad (3.126)$$

Burada, $u(t)$, $y(t)$ ve $v(t)$ sırasıyla doğrusal sistemin giriş, çıkış ve bozulma sinyallerini gösterir. Bozulma sinyalini, bozulma modelinin dürtü yanıtı ve sıfır ortalamalı, varyansı σ_e^2 olan beyaz gürültü $e(t)$ fonksiyonu olarak, Denklem (3.127)'deki gibi yazılabilir:

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) \quad (3.127)$$

Tüm sinyaller skaler değerli fonksiyonlardır ve $t \in \mathbb{Z}$ ayrık zaman örnek indeksini numaralandırmak için kullanılır. İleri ve geri kaydırma operatörleri sırasıyla q ve q^{-1} ile gösterilir, Denklem (3.128) ve Denklem (3.129) ile ifade edilir:

$$qu(t) = u(t+1) \quad (3.128)$$

$$q^{-1}u(t) = u(t-1) \quad (3.129)$$

Eğer sistem kararlıysa ve bozulma transfer fonksiyonu birim katsayılı ise, transfer fonksiyonlarının kaydırma operatörleri ile sembolize edilmiş hali, Denklem (3.130) ve Denklem (3.131) ile gösterilebilir:

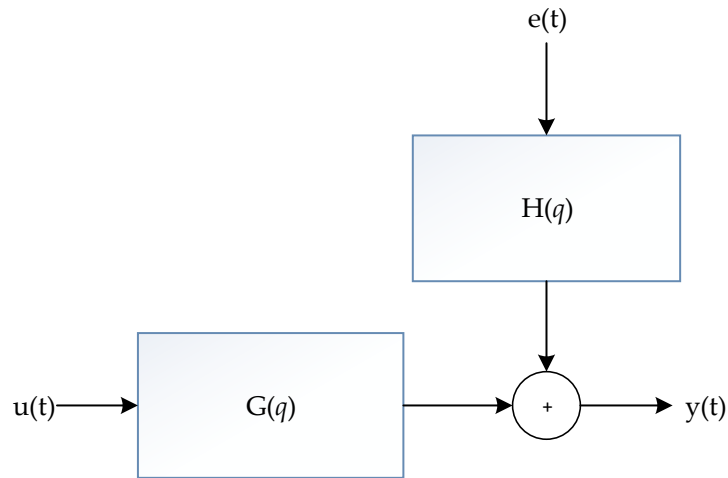
$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k} . \quad \sum_{k=1}^{\infty} |g(k)| < \infty \quad (3.130)$$

$$H(q) = 1 + \sum_{k=1}^{\infty} h(k)q^{-k} , h(0) = 1 \quad (3.131)$$

Burada, G, doğrusal sistem modelinin transfer fonksiyonu ve H, bozulma modelinin transfer fonksiyonu olarak tanımlanır.

Toplamsal bozulmalı doğrusal zamanla değişmeyen bir sistem, Denklem (3.132)'de gösterildiği gibi ifade edilir (Ljung, 1999). LTI sistem yapısı Şekil 3.12'de gösterilmektedir.

$$y(t) = G(q)u(t) + H(q)e(t) \quad (3.132)$$



Şekil 3.12. Toplamsal bozulmalı doğrusal zamanla değişmeyen (LTI) sistem

3.4.2. Öngörü Yöntemi

$v(t)$ 'nin gelecekteki değerleri, Denklem (3.133)'teki gibi tanımlandığında tahmin edilebilir.

$$v(t) = H(q)e(t) = \sum_{k=0}^{\infty} h(k)e(t-k) \quad (3.133)$$

H 'nin kararlı olduğunu varsayarsak, Denklem (3.134) ifadesi yazılabilir.

$$\sum_{k=0}^{\infty} |h(k)| < \infty \quad (3.134)$$

Bozulma modelinin tersi alınabilir ise, $v(s)$ ve $s \leq t-1$ biliniyorsa, o halde $e(t)$ Denklem (3.135)'te gösterildiği gibi hesaplanabilir:

$$e(t) = \tilde{H}(q)v(t) = \sum_{k=0}^{\infty} \tilde{h}(k)v(t-k), \sum_{k=0}^{\infty} |\tilde{h}(k)| < \infty \quad (3.135)$$

Bozulma sinyali $v(s)$ 'nin $s \leq t-1$ için geçmiş değerlerine sahip olduğumuzu ve $v(t)$ değerini tahmin etmek istediğimizi varsayalım:

$$v(t) = H(q)e(t) = \sum_{k=0}^{\infty} h(k)e(t-k) \quad (3.136)$$

Eğer $H(q)$ tersi alınabilir ve birim katsayılı ise,

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) = e(t) + \sum_{k=1}^{\infty} h(k)e(t-k), h(0) = 1 \quad (3.137)$$

Denklem (3.137)'deki ikinci terimidir, bu nedenle $t-1$ anında bilinir. Geçici olarak $m(t-1)$ ile gösterilirse,

$$m(t-1) = \sum_{k=1}^{\infty} h(k)e(t-k) \quad (3.138)$$

ve $t-1$ anındaki $v(t)$ 'nin tahmin değeri $\hat{v}(t | t-1)$ olarak tanımlanırsa,

$$\begin{aligned} \hat{v}(t | t-1) &= \left[\sum_{k=1}^{\infty} h(k)q^{-k} \right] e(t) = [H(q) - 1]e(t) \\ &= \frac{H(q)-1}{H(q)} v(t) = [1 - H^{-1}(q)]v(t) = \sum_{k=1}^{\infty} -\tilde{h}(k)v(t-k) \end{aligned} \quad (3.139)$$

$v(t)$ 'nin $t-1$ anındaki tahmin değeri $\hat{v}(t | t-1)$, Denklem (3.140)'ta belirtildiği gibi gösterilebilir:

$$H(q)\hat{v}(t | t-1) = [H(q) - 1]v(t) = \sum_{k=1}^{\infty} h(k)v(t-k) \quad (3.140)$$

Eğer $y(s)$ ve $u(s)$ değerlerini $s \leq t-1$ için biliyorsak, bozulma değerlerini, Denklem (3.141)'daki gibi tanımlayabiliriz:

$$v(s) = y(s) - G(q)u(s) \quad (3.141)$$

Tahmin etmek istediğimiz çıkış değeri, Denklem (3.142)'deki gibi gösterilir:

$$y(t) = G(q)u(t) + v(t) \quad (3.142)$$

$y(t)$ 'nin koşullu beklenen değeri, Denklem (3.143)'te gösterildiği gibi olacaktır:

$$\begin{aligned} \hat{y}(t | t-1) &= G(q)u(t) + \hat{v}(t | t-1) \\ &= G(q)u(t) + [1 - H^{-1}(q)]v(t) \\ &= G(q)u(t) + [1 - H^{-1}(q)][y(t) - G(q)u(t)] \end{aligned} \quad (3.143)$$

Terimler tekrar düzenlendiğinde, çıkış değerinin bir-adım-ileri tahmini, Denklem (3.144)'de ifade edildiği gibidir:

$$\hat{y}(t | t - 1) = H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t) \quad (3.144)$$

Denklem (3.144) ve Denklem (3.141) kullanılarak tahmin hatası $y(t) - \hat{y}(t | t - 1)$, Denklem (3.145)'teki gibi tanımlanır:

$$y(t) - \hat{y}(t | t - 1) = H^{-1}(q)G(q)u(t) + H^{-1}(q)y(t) = e(t) \quad (3.145)$$

Burada $e(t)$, $y(t)$ 'nin geçmiş verilerinden tahmin edilemeyen kısmını ifade eder. Bu nedenle, Denklem (3.145) aynı zamanda t anındaki inovasyon olarak da adlandırılır.

3.4.3. Öngörücü Filtre Modeli

Öngörücü seçimi, çıkış ölçüm hatalarına karşı hassasiyet ile hatalı başlangıç koşullarının hızla azalması arasındaki bir denge olarak görülebilir. Bu dengeyi sağlamak için tasarım değişkenleri tanıtmak amacıyla, Denklem (3.146)'daki gibi bir filtre seçilmelidir:

$$W(q) = 1 + \sum_{\ell=k}^{\infty} w_{\ell} q^{-\ell} \quad (3.146)$$

Bu filtre bozulma bileşeni bulunmayan sistem modeline uygulanırsa,

$$W(q)y(t) = W(q)G(q)u(t) \quad (3.147)$$

Bu ifade Denklem (3.148)'deki gibi düzenlenebilir:

$$y(t) = [1 - W(q)]y(t) + W(q)G(q)u(t) \quad (3.148)$$

Denklem (3.148)'in sağ tarafı, $s \leq t - 1$ için sadece $y(s)$ ve $u(s)$ değerlerine bağlıdır. Buradan, $y(t)$ değeri, Denklem (3.149)'da gösterildiği gibi tahmin edilir:

$$\hat{y}(t | t - k) = [1 - W(q)]y(t) + W(q)G(q)u(t) \quad (3.149)$$

3.4.4. LTI Sistem Yapısı

Belirli bir amaca uygun olarak, gerçek sistemin özelliklerini tanımlamak için sistem modelleri kullanılır. Doğrusal zamanla değişmeyen, toplamsal bozulmalı model yapısı Denklem (3.150)'deki gibi gösterilmektedir:

$$y(t) = G(q)u(t) + H(q)e(t) \quad (3.150)$$

Sonlu sayıda sayısal değer ve katsayı kullanılarak sistemin temsil edilmesi amaçlansa da, tüm katsayıların sistemin fiziksel mekanizmalarından belirlenmesi mümkün değildir. Bu nedenle, bazı katsayılar, belirlenmesi gereken parametreler olarak tahmin modeline ilave edilmelidir. Parametre vektörü θ ile gösterilirse, model gösterimi Denklem (3.151)'de belirtildiği gibi tanımlanır:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (3.151)$$

Burada, parametre vektörü θ , Denklem (3.150)'de gösterildiği gibi, Öklidyen d-boyutlu uzayın bir alt kümesi olan D_M üzerinde tanımlanır:

$$\theta \in D_M \subset R^d \quad (3.152)$$

Son iki denklem bir model değil, modeller kümesidir. Bir-adım-ileri tahmin denklemi, parametre vektörüne bağlı olarak, Denklem (3.153)'teki gibi tanımlanır:

$$\hat{y}(t | \theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]y(t) \quad (3.153)$$

Denklem (3.152)'de olduğu gibi parametrik modeller kümesini tanımlayan ifadelerle model yapısı denir ve $\mathcal{M}$ harfi ile gösterilir.

3.4.5. ARX Model Yapısı

Bir sistem, fiziksel özellikleri hakkında önceden bilgiye sahip olunmaksızın matematiksel bir model olarak temsil edilebilir. Bu bağlamda, G ve H transfer fonksiyonlarının rasyonel fonksiyonlar şeklinde parametrik olarak tanımlanması, pay ve payda katsayılarının bilinmeyen parametreler olarak ele alınması anlamına gelir. Bu tür parametrik tanımlamaların çeşitli biçimleri ve uygulama yöntemleri mevcuttur. Söz konusu model yapıları, literatürde kara-kutu (black-box) modeller olarak da adlandırılmaktadır.

Bir sistemin giriş-çıkış ilişkisi, lineer fark denklemi ile Denklem (3.154)'teki gibi gösterilebilir:

$$\begin{aligned} y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) \\ = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) \end{aligned} \quad (3.154)$$

Beyaz gürültü terimi $e(t)$, fark denkleminin denklem hatası model yapısını ifade eder. Hesaplanmak istenilen ayarlanabilir parametreler, Denklem (3.155)'teki gibi gösterilir:

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b}]^T \quad (3.155)$$

Eğer parametreler polinom fonksiyonları olarak gösterilirse:

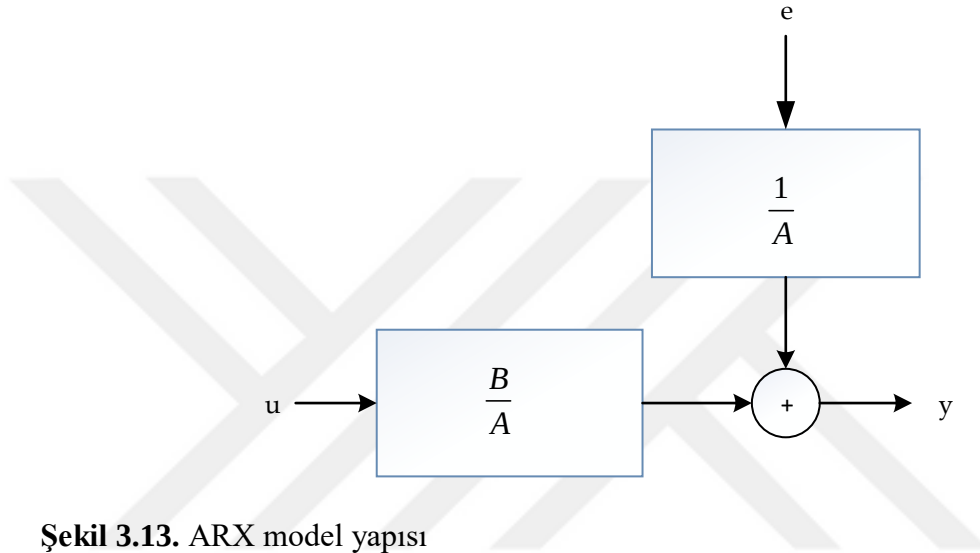
$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \quad (3.156)$$

$$B(q) = b_1 q^{-1} + \dots + b_{n_b} q^{-n_b} \quad (3.157)$$

O halde, transfer fonksiyonu modelleri, Denklem (3.158)'de tanımlanır:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \quad H(q, \theta) = \frac{1}{A(q)} \quad (3.158)$$

Lineer fark denklemi, Denklem (3.154), otoregresif ve dışsal (exogenous) parçalara sahiptir. Bu iki parçadan oluşan yapı, ARX modeli olarak adlandırılır. ARX model yapısı Şekil 3.13'te gösterilmektedir.



Şekil 3.13. ARX model yapısı

3.4.6. ARMAX Model Yapısı

Lineer fark denklemi, Denklem (3.154), bozulma teriminin özelliklerine sahip değildir. Denklem hatası, beyaz gürültünün hareketli ortalaması olarak tanımlanarak daha esnek bir model elde edilebilir:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) \quad (3.159)$$

Yeni parametrelerin polinom fonksiyonu, Denklem (3.160)'da gösterilmektedir:

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \quad (3.160)$$

Bozulma terimi ile beraber giriş-çıkış ilişkisi, Denklem (3.161) ile gösterilir:

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (3.161)$$

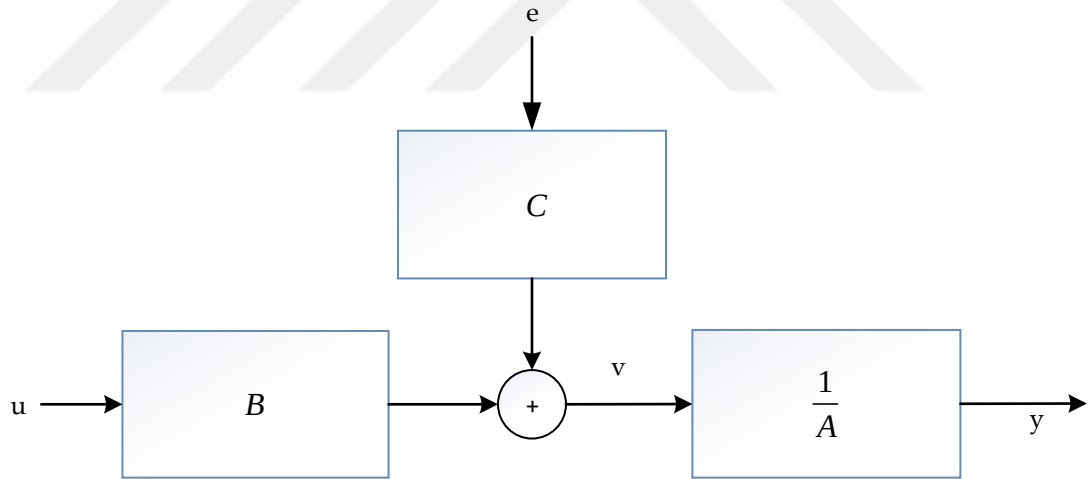
Transfer fonksiyonu gösterimi, Denklem (3.162) ile tanımlanır:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \quad H(q, \theta) = \frac{C(q)}{A(q)} \quad (3.162)$$

Parametre vektörü θ , Denklem (3.163)'teki gibi tanımlanır:

$$\theta = [a_1 \dots a_{n_a} b_1 \dots b_{n_b} c_1 \dots c_{n_c}]^T \quad (3.163)$$

Otoregresif, dışsal ve hareketli ortalama bölümleri içeren model, ARMAX modeli olarak adlandırılır. ARMAX model yapısı Şekil 3.14'te gösterilmektedir.



Şekil 3.14. ARMAX model yapısı

3.4.7. Çıkış-Hatası (OE) Model Yapısı

Bozulmamış çıkış $w(t)$ ve giriş, lineer bir fark denklemi olarak Denklem (3.164)'te gösterildiği gibi yazılabilir:

$$\begin{aligned} w(t) + f_1 w(t-1) + \dots + f_{n_f} w(t-n_f) \\ = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) \end{aligned} \quad (3.164)$$

Çıkış sinyali, Denklem (3.165) ile gösterilirse,

$$y(t) = w(t) + e(t) \quad (3.165)$$

Polinom fonksiyonu, Denklem (3.166)'da gösterildiği gibi olacaktır:

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f} \quad (3.166)$$

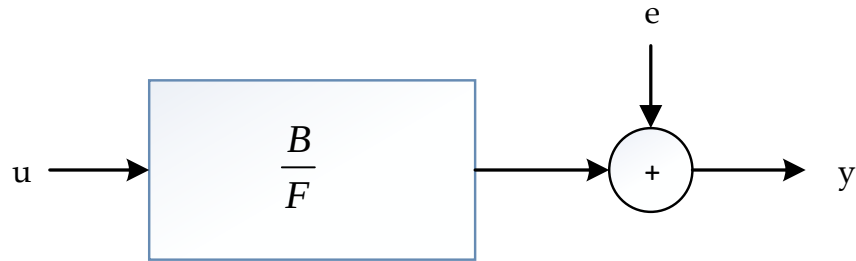
Çıkış hatası modeli, Denklem (3.167)'de gösterildiği gibi yazılırsa,

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \quad (3.167)$$

Parametre vektörü θ Denklem (3.168) ile gösterilir:

$$\theta = [b_1 \ b_2 \dots b_{n_b} f_1 \ f_2 \dots f_{n_f}]^T \quad (3.168)$$

Çıkış hatası model yapısı Şekil 3.15'te gösterilmektedir.



Şekil 3.15. Çıkış-Hatası model yapısı

3.4.8. Box-Jenkins Model Yapısı

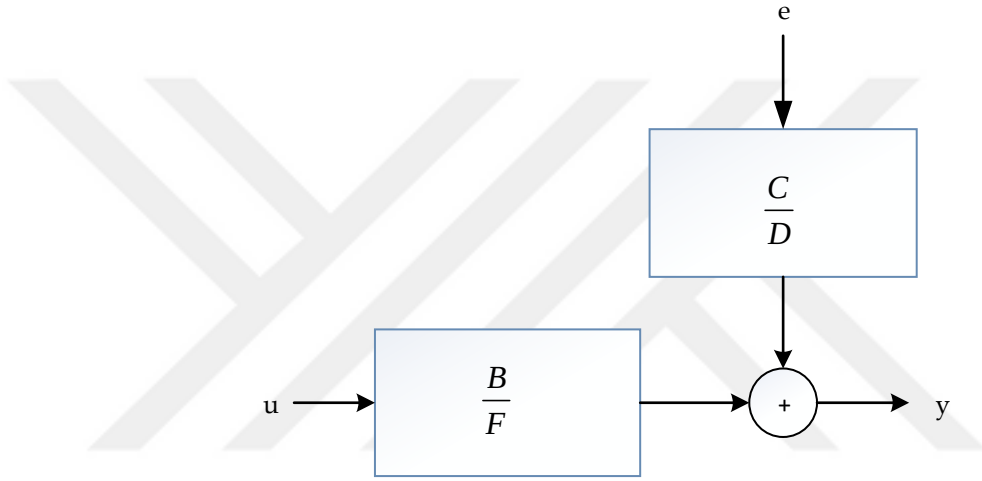
Box-Jenkins model yapısı, çıkış hatasını bir ARMA modeli olarak tanımlar:

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (3.169)$$

Öngörücü modeli, Denklem (3.170) ile tanımlanmaktadır:

$$\hat{y}(t | \theta) = \frac{D(q)B(q)}{C(q)F(q)}u(t) + \frac{C(q) - D(q)}{C(q)}y(t) \quad (3.170)$$

Box-Jenkins model yapısı, Şekil 3.16’da gösterilmektedir.



Şekil 3.16. Box-Jenkins model yapısı

3.4.9. Genel Model Yapısı

A, B, C, D ve F polinomları kullanılarak 32 farklı model seti yazılabilir. Genel bir model yapısı, Denklem (3.171)’da gösterildiği gibi tanımlanabilir:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (3.171)$$

3.4.10. Parametre Tahmin Yöntemi

Belirlenen model yapısı $\mathcal{M}$ ve bu yapıyı parametre vektörü $\theta \in D_{\mathcal{M}} \subset R^d$ ile parametrik halde ifade eden modeller $\mathcal{M}(\theta)$ ile gösterilsin. Böylece tanımlanan model kümesi, Denklem (3.172)'de gösterildiği gibi ifade edilir:

$$\mathcal{M}^* = \{\mathcal{M}(\theta) \mid \theta \in D_{\mathcal{M}}\} \quad (3.172)$$

Denklem (3.172) ile ifade edilen bu küme, parametre vektörü θ 'nın değiştirilmesiyle elde edilebilecek tüm modelleri içerir. Her model, gelecekteki çıkışların tahmin edilmesini sağlayan bir öngörücüye karşılık gelir. Öngörücü tanımı, doğrusal bir filtre olarak ifade edilirse:

$$\mathcal{M}(\theta): \hat{y}(t \mid \theta) = W_y(q, \theta)y(t) + W_u(q, \theta)u(t) \quad (3.173)$$

Denklem (3.173), Denklem (3.174) ile gösterilen temel sistem tanımı için, bir adım ileriye yönelik öngörüye karşılık gelir:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (3.174)$$

Bu durumda, öngörücü filtre tanımları, Denklem (3.175)'te verilmiştir:

$$W_y(q, \theta) = [1 - H^{-1}(q, \theta)] \quad W_u(q, \theta) = H^{-1}(q, \theta)G(q, \theta) \quad (3.175)$$

Öngörücü, doğrusal olmayan bir filtre şeklinde de tasarlanabilir; bu durumda Z^{t-1} geçmiş verilerin genel bir fonksiyonu olarak yazılırsa:

$$\mathcal{M}(\theta): \hat{y}(t \mid \theta) = g(t, Z^{t-1}; \theta) \quad (3.176)$$

$\mathcal{M}(\theta)$ modeli, aynı zamanda ilgili öngörü hatalarının karakteri hakkında da varsayımlar içerebilir. Örneğin, hataların varyansları $\lambda(\theta)$ veya olasılık dağılımları $f_e(x, \theta)$, örnek olarak gösterilebilir.

Sistem üzerinden toplanan verilerin kümesi, Denklem (3.177) ile ifade edilirse:

$$Z^N = [y(1), u(1), y(2), u(2), \dots, y(N), u(N)] \quad (3.177)$$

Burada çözülmesi gereken sorun, $\mathcal{M}^*$ kümesi içerisindeki uygun $\mathcal{M}(\hat{\theta}_N)$ modelinin belirlenmesi, yani $\hat{\theta}_N$ parametre vektörünün uygun değerlerinin belirlenmesi için Z^N içerisindeki bilgiyi nasıl kullanacağımıza karar verebilmektir. Kısaca, veri kümesi Z^N ile D_M kümesinin nasıl eşleştirileceğidir:

$$Z^N \rightarrow \hat{\theta}_N \in D_M \quad (3.178)$$

Bu şekilde bir eşleştirmeye parametre tahmin yöntemi denir.

3.4.11. Tahmin Hatası Yöntemi

Belirli bir $\mathcal{M}(\theta_*)$ modeli tarafından verilen tahmin hatası, Denklem (3.179) ile tanımlanır:

$$\varepsilon(t, \theta_*) = y(t) - \hat{y}(t | \theta_*) \quad (3.179)$$

Denklem (3.179) ile verilen tahmin hatası dizisi, $\mathbb{R}^N$ şeklinde bir vektör olarak görülebilir. Bu öngörü hatası dizisinin, kararlı bir doğrusal filtre $L(q)$ aracılığıyla, Denklem (3.180)'de ifade edildiği şekilde filtrelendiğini varsayalım:

$$\varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta), \quad 1 \leq t \leq N \quad (3.180)$$

Filtreleme işlemi, Denklem (3.181)'deki gibi ifade edilir:

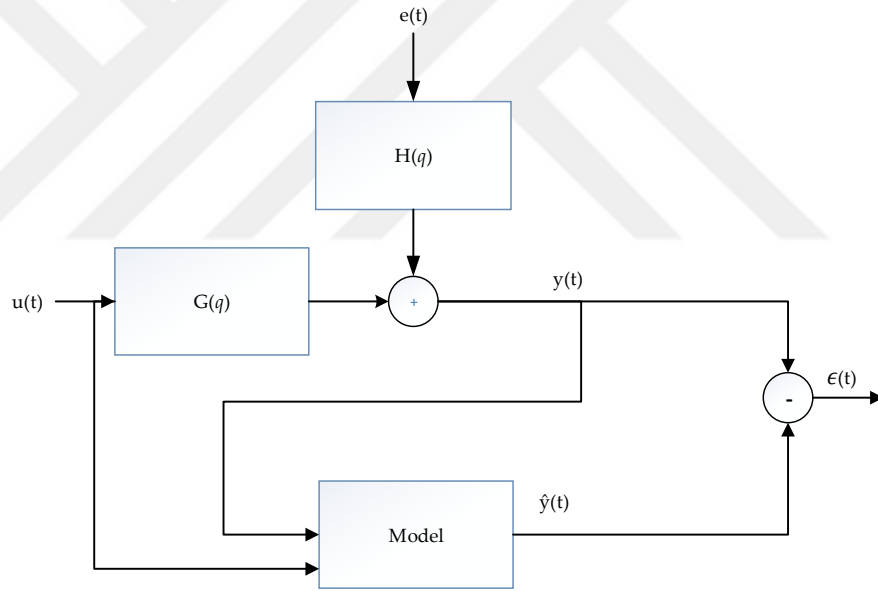
$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \ell(\varepsilon_F(t, \theta)) \quad (3.181)$$

Burada, $\ell(\cdot)$ genellikle pozitif ve skaler olarak tanımlı bir fonksiyondur.

$V_N(\theta, Z^N)$ fonksiyonu, verilen Z^N veri kümesi için, model parametresi θ ile tanımlanmış skaler değerli bir fonksiyondur. Bu fonksiyon, $\mathcal{M}(\theta)$ modelinin geçerliliğini ölçmek için kullanılan bir ölçüttür. Buradan, $\hat{\theta}_N$ parametre vektörünün tahmini, Denklem (3.182) ile gösterilmektedir:

$$\hat{\theta}_N = \hat{\theta}_N(Z^N) = \underset{\theta \in D_M}{\operatorname{argmin}} V_N(\theta, Z^N) \quad (3.182)$$

Bu şekilde yapılan bir eşleştirme, tahmin hatası tanımlama yöntemi (PEM) olarak adlandırılır. Tahmin hatası blok şeması Şekil 3.17’de gösterilmektedir.



Şekil 3.17. Tahmin hatası blok şeması

Sistem modeli, belirli bir amaca uygun olan gerçek sistemin özelliklerinin bir tanımıdır. "Sistem tanımlama, dinamik sistemlerin belirli amaçlara hizmet etmek üzere modellerini oluşturma veya seçme konusudur" (Ljung, 1999).

3.4.12. Parametre Tahmin Vektörü

Parametre tahmini, en iyi modeli verecek şekilde model yapılarına ait parametre vektörü θ 'yı tahmin etmek için yapılan arama sürecidir. Bu çalışmada, parametre vektörü θ 'yı tahmin etmek için tahmin hatası tanımlama yöntemi (PEM) kullanılmıştır. Eğer sistemin giriş-çıkış verisi Denklem (3.183)'deki gibi yazılırsa (G. Pillonetto ve Ljung, 2023):

$$Z^N = \{y(1), u(1), y(2), u(2), \dots, y(N), u(N)\} \quad (3.183)$$

Uyum kriteri, sistemin gerçek çıkışı ile modelin tahmin edilen çıkışı arasındaki tahmin hatası, Denklem (3.184)'teki gibi yazılabilir:

$$V_N(\theta) = \sum_{t=1}^N (y(t) - \hat{y}(t | \theta))^2. \quad (3.184)$$

PEM için θ 'nın tahmini, Denklem (3.184)'ün minimizasyonu ile, Denklem (3.185)'te gösterildiği gibi tanımlanır:

$$\hat{\theta}_N = \hat{\theta}_N(Z^N) = \underset{\theta \in D_M}{\operatorname{argmin}} V_N(\theta, Z^N). \quad (3.185)$$

3.4.13. Kriter Fonksiyonunun Sayısal Minimizasyonu

Denklem (3.185)'in minimizasyonu, bir doğrusal olmayan en küçük kareler (NLS) problemi tanımlar. Bu problemi çözmek için iteratif arama yöntemleri olan gradyan inişi ve Gauss-Newton yöntemleri kullanılabilir (Ljung, 1999; Gianluigi Pillonetto vd., 2014; Söderström ve Stoica, 1989). SISO kara-kutu modelinin genel tahmincisi, Denklem (3.186)'da tanımlanmıştır.

$$\hat{y}(t | \theta) = \frac{D(q)B(q)}{C(q)F(q)} u(t) + \left[1 - \frac{D(q)A(q)}{C(q)}\right] y(t), \quad (3.186)$$

Parametre vektörlerinin gradyanları, tahmin modelinin parametreler ile ilgili türevleri alınarak hesaplanabilir (Ljung, 1999) ve Denklem (3.187) ile Denklem (3.191) arasında gösterilmektedir.

$$\frac{\partial}{\partial a_k} \hat{y}(t | \theta) = -\frac{D(q)}{C(q)} y(t - k) \quad (3.187)$$

$$\frac{\partial}{\partial b_k} \hat{y}(t | \theta) = -\frac{D(q)}{C(q)F(q)} u(t - k) \quad (3.188)$$

$$\frac{\partial}{\partial c_k} \hat{y}(t | \theta) = \frac{1}{C(q)} \varepsilon(t - k, \theta) \quad (3.189)$$

$$\frac{\partial}{\partial d_k} \hat{y}(t | \theta) = -\frac{1}{C(q)} v(t - k, \theta) \quad (3.190)$$

$$\frac{\partial}{\partial f_k} \hat{y}(t | \theta) = \frac{D(q)}{C(q)F(q)} w(t - k, \theta) \quad (3.191)$$

Yardımcı değişkenler, Denklem (3.192) ve (3.193) ile tanımlanır:

$$w(t, \theta) = \frac{B(q)}{F(q)} u(t), \quad (3.192)$$

$$v(t, \theta) = A(q)y(t) - w(t, \theta). \quad (3.193)$$

3.4.14. Başarımın Değerlendirilmesi

Model yapıların performansı, ortalama kare hata (MSE), ortalama mutlak hata (MAE) ve uyum iyiliği (G) olmak üzere üç ölçüt kullanılarak değerlendirilmiştir. Bu ölçütlerin tanımları, Denklem (3.194) ile Denklem (3.196) arasında ifade edilmektedir (Mustafaraj vd., 2010; Paschke vd., 2019; Gianluigi Pillonetto vd., 2014).

Ortalama Karesel Hata (MSE):

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N [y_i - \hat{y}_i]^2 \quad (3.194)$$

Ortalama Mutlak Hata (MAE):

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (3.195)$$

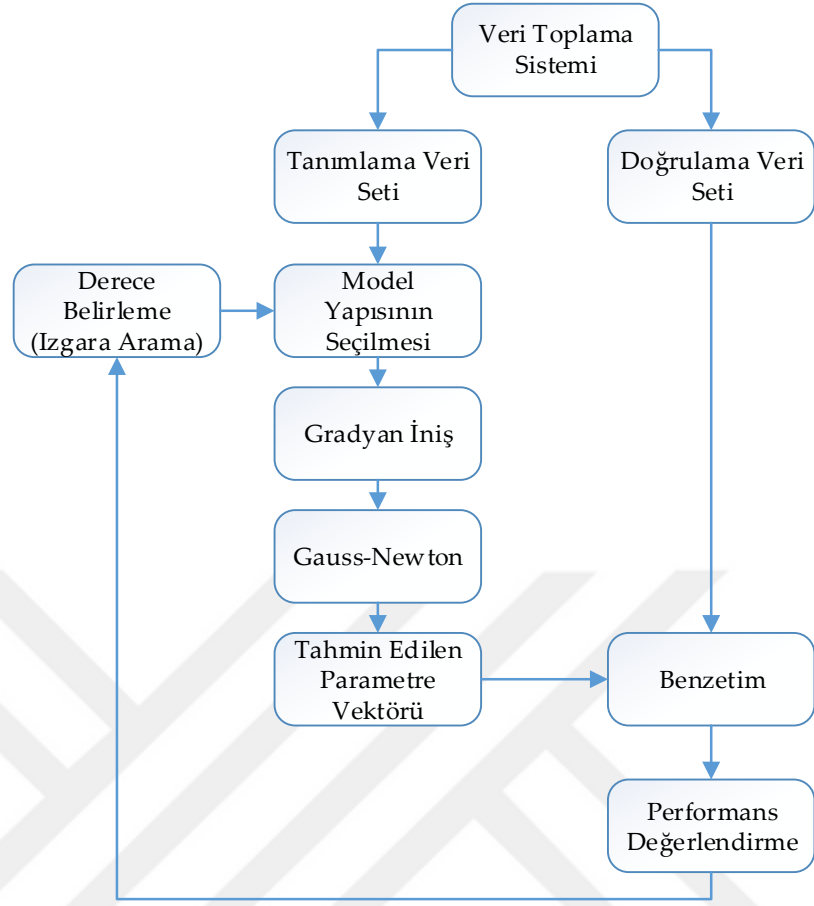
Uygunluk Ölçüsü (G):

$$G = \left(1 - \frac{\sqrt{\sum_{i=1}^N (\hat{y}_i - y_i)^2}}{\sqrt{\sum_{i=1}^N \left(y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)^2}} \right) \times 100, \quad (3.196)$$

Burada, N , veri setinin boyutunu; y ve $\hat{y}$ ise sırasıyla ölçülen ve tahmin edilen çıkış değerlerini temsil eder.

3.4.15. Parametre Vektörü Tahmin Sistemi

Sistemin genel yapısı, Şekil 3.18'de özetlenmiştir. Parametre vektörü θ , dört farklı model yapısı için, ardışık iki aşamalı bir süreçte gradyan inişi ve Gauss-Newton yöntemleri kullanılarak tanımlama veri seti üzerinde tahmin edilmiştir. Model yapılarının performansı, doğrulama veri seti üzerinde MSE, MAE ve G ölçütleri kullanılarak değerlendirilmiştir. Model yapılarının derecesi, en iyi tahmin sonuçlarını elde etmek için ızgara arama algoritması ile belirlenmiştir.



Şekil 3.18. Parametre vektörü tahmin sistemi

3.4.16. Laboratuvar İnkübatör Cihazı

Bu çalışmada, Şekil 3.19 ve Şekil 3.20’de görülen laboratuvar inkübatör cihazı kullanılmıştır.



Şekil 3.19 İnkübatör cihazının kapağı kapalı durumdaki görünümü

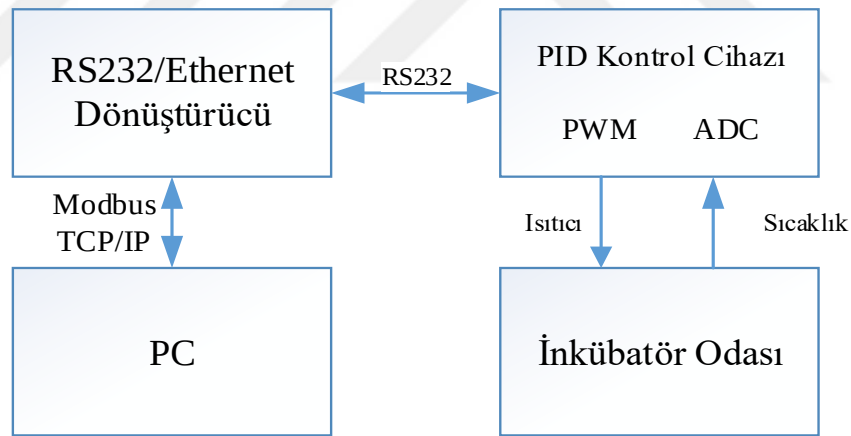


Şekil 3.20 İnkübatör cihazının kapağı açık durumdaki görünümü

Veri toplama amacıyla, inkübatöre harici bir DIN/EN 60751 Sınıf B Pt-100 sıcaklık sensörü ve 1/16 DIN standart tek döngülü PID kontrol cihazı ilave edilmiştir. Sıcaklık sensörünün termal hassasiyeti 0,1°C/37°C olup, PID kontrol cihazı saniyede 10 okumaya sahip bir seri iletişim ara yüzüne sahiptir.

3.4.17. Veri Toplama Sistemi

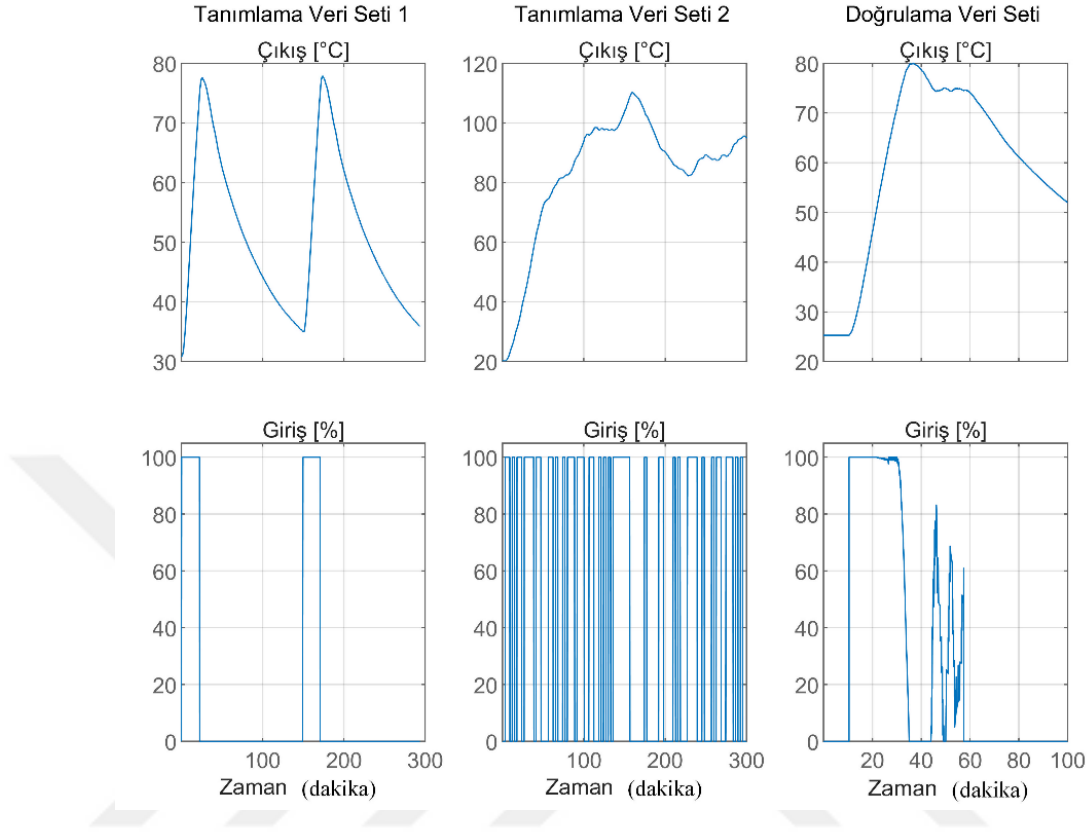
Çalışmada kullanılan veri toplama sistemi, Şekil 3.21’de gösterilmiştir. Veri toplama süreci başlatıldığında, sistemin kararlı durumda olduğu kabul edilmiştir. PID kontrol cihazı ile PC arasındaki iletişim gecikmesini en aza indirmek amacıyla Seri-Ethernet dönüştürücü kullanılmıştır. PID kontrolörünün giriş-çıkış verileri, Modbus protokolü kullanılarak toplanmıştır. PID kontrolörünün kontrol sinyali çıkışı, darbe genişlik modülasyonu (PWM) tekniği kullanılarak inkübatöre gönderilmiştir. PWM sinyalinin periyodu 1 saniye olarak ayarlanmış ve görev döngüsü, periyodun %0,1’i olarak seçilmiştir. Sıcaklık sensörünün okuma hassasiyeti, 0,1 °C olarak yapılandırılmıştır.



Şekil 3.21. Veri toplama sistemi

Üç ayrı veri seti, tanımlama veri seti-1, tanımlama veri seti-2 ve doğrulama veri seti, 0,1 saniye örnekleme süresi ile veri toplama çerçevesi kullanılarak elde edilmiştir. Tanımlama veri seti-1, 75 °C set değeri ve 40 °C histerezis değeri seçildiğinde, 300 dakika boyunca birim basamak yanıtından toplanmıştır. Tanımlama veri seti-2, 300 dakika boyunca sözde rasgele ikili dizi (PRBS) yanıtından toplanmıştır. Doğrulama veri seti, PID ayar süreci sırasında 100 dakika boyunca oluşturulmuştur.

Tanımlama ve doğrulama verileri için giriş-çıkış grafikleri, Şekil 3.22’de göstermektedir.



Şekil 3.22. Birim basamak ve PRBS cevabı ile PID ayar giriş-çıkış grafikleri

3.4.18. Veri Ön İşleme

Veri toplama örnekleme periyodu, 0,1 saniye olarak seçilmiştir. Toplanan ham verilerin boyutu, eğitim veri seti için yaklaşık 180.000 örnek ve doğrulama veri seti için 138.000 örnektir. Grafiksel temsilde yapılan görsel analiz sonrasında, giriş ve çıkış arasındaki gecikmenin 60 saniye olduğu belirlenmiştir. Eğitim aşamasına geçilmeden önce, veri seti gecikme miktarı kadar kaydırılmıştır.

Veri toplama süreci başladığında, sistemin kararlı durumda olduğu kabul edilmektedir. Toplanan veriler, min-max tipi normalizasyon ile Denklem (3.197)’de gösterildiği gibi normalize edilmiş ve (0,1) aralığında ölçeklendirilmiştir. İşlem için gerekli hesaplama süresini azaltmak amacıyla veriler 1 Hz’de yeniden örneklenmiştir. Aynı nedenle, toplanan veri setleri kırılmıştır. Tanımlama veri setinin yalnızca ilk yarısı tanımlama

süreci için, doğrulama veri setinin ise ilk 100 dakikası doğrulama verisi olarak kullanılmıştır.

$$X_{\text{norm}} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (3.197)$$



4. BULGULAR ve TARTIŞMA

4.1. Sistem Tanımlama Bulguları

Deneyler, laboratuvar inkübatörü için C++ dilinde gerçekleştirilmiştir. Deneylerde, inkübatörün modellenmesi için dört farklı model yapısı, ARX, ARMAX, OE ve BJ, kullanılmıştır. Laboratuvar inkübatörü, bir LTI SISO sistemi olarak ele alınmıştır.

Deneylerde iki senaryo yürütülmüştür. Birinci senaryoda, eğitim veri seti sistemin basamak yanıtından toplanırken, ikinci senaryoda sistemin PRBS yanıtından toplanmıştır.

Her iki senaryoda da, tüm model yapıları için model derecesi seçimi, bilinen ızgara arama algoritması kullanılarak yapılmıştır. Model yapılarındaki polinomların dereceleri, [1, 10] aralığında ızgara araması gerçekleştirilerek seçilmiştir. Model yapıları için polinomların en iyi dereceleri, uyum iyiliği (G) ölçüsüne göre belirlenmiştir.

Model yapılarına ait parametre vektörü (θ), her iki senaryo için de PEM kullanılarak tahmin edilmiştir. $V_N(\theta)$ minimizasyonu, ardışık iki aşamalı süreçlerde sayısal olarak gerçekleştirilmiştir. İlk aşamada, parametreler gradyan inişi yöntemi kullanılarak tahmin edilmiştir. İkinci aşamada, parametrelerin değerleri Gauss-Newton yöntemi için başlangıç değerleri olarak kullanılmıştır ve parametrelerin nihai değerleri tahmin edilmiştir.

Tahmin edilen parametre vektörlerine (θ) sahip model yapılarına ait yanıtlar, laboratuvar inkübatörünün PID ayar sürecinden toplanan doğrulama veri seti için hesaplanmıştır. Deneylerde, 2, 10 ve 20 adım öteye tahmin yanıtları hesaplanmıştır. Ayrıca, model yapısının sonsuz ufuk tahmin yanıtı, yani açık döngü benzetim modelinin yanıtı, sağlanmıştır. Model yapıların performansları, MSE, MAE, G ölçüleri ve kalıntı korelasyonları açısından incelenmiştir ve birbirleriyle karşılaştırılmıştır.

4.1.1. ARX Modeli için Elde Edilen Bulgular

ARX modelinin model derecesi, birinci senaryo için $n_a = 2$ ve $n_b = 2$ olarak belirlenmiştir. ARX [2 2] parametre vektörü θ , tanımlama veri seti-1 üzerinde tahmin edilmiştir. Birinci senaryoda ARX modeli için polinomlar Denklem (4.1) ve Denklem (4.2)'deki gibi elde edilmiştir.

$$A(q) = 1 - 0,83094q^{-1} - 0,16883q^{-2}, \quad (4.1)$$

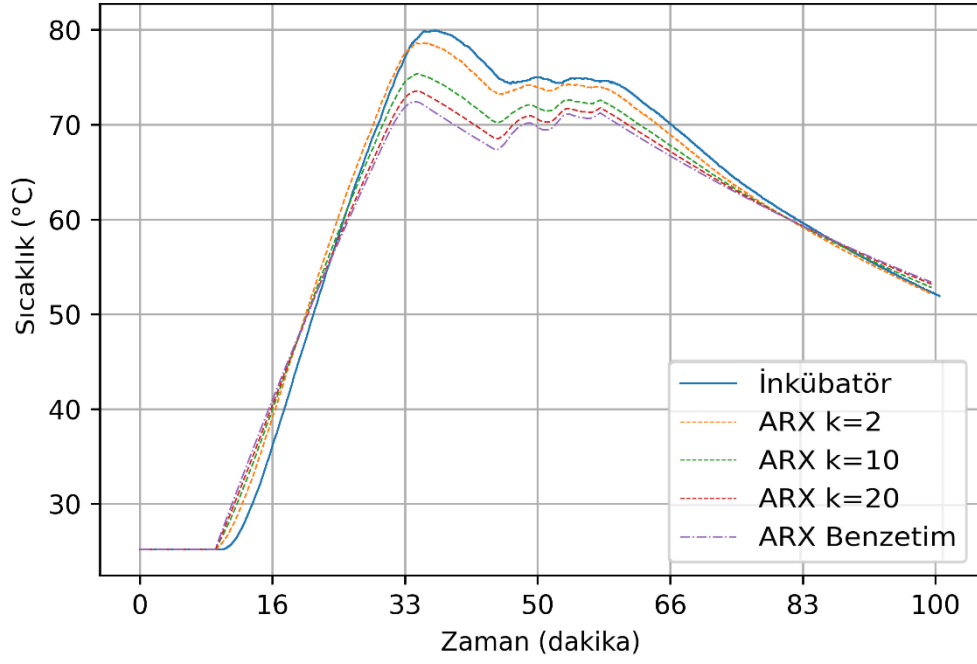
$$B(q) = 4,02159 \times 10^{-5}q^{-1} - 0,16883q^{-2}, \quad (4.2)$$

İkinci senaryo için de ARX modelinin model derecesi $n_a = 2$ ve $n_b = 2$ olarak belirlenmiştir ve ARX [2 2] parametre vektörü θ , tanımlama veri seti-2 üzerinde tahmin edilmiştir. İkinci senaryoda ARX modeli için polinomlar Denklem (4.3) ve Denklem (4.4)'de olduğu gibi elde edilmiştir.

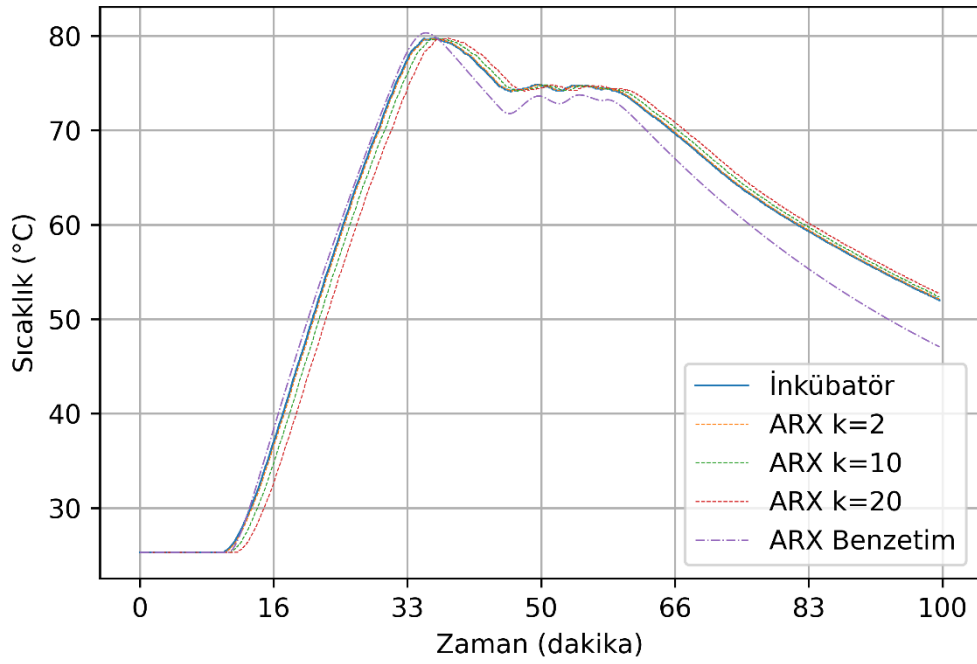
$$A(q) = 1 - 1,9876q^{-1} + 0,9868q^{-2}, \quad (4.3)$$

$$B(q) = 4,474 \times 10^{-5}q^{-1} - 3,785 \times 10^{-5}q^{-2}, \quad (4.4)$$

Modelin performans değerlendirmesini yapmak amacıyla her iki senaryoda da aynı doğrulama veri seti kullanılmıştır. Modelin 2 adım, 10 adım ve 20 adım öteye tahmin ve benzetim yanıtları doğrulama veri seti için hesaplanmıştır. Her iki senaryonun yanıtları, inkübatörün gerçek çıkışı ile karşılaştırılmış ve birinci senaryo için Şekil 4.1'de, ikinci senaryo için ise Şekil 4.2'de görselleştirilmiştir.



Şekil 4.1. İlk senaryoda ARX[2 2] için k-adım tahmin ve benzetim cevapları



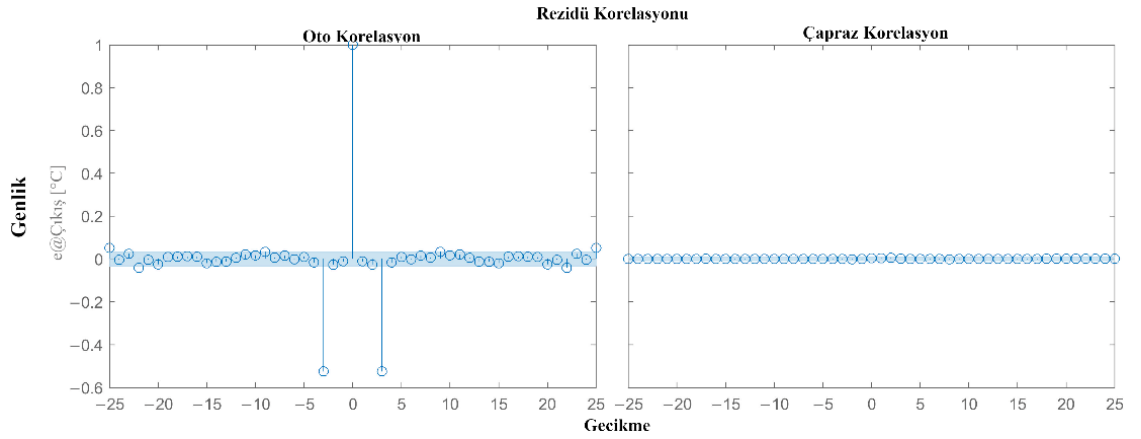
Şekil 4.2. İkinci senaryoda ARX[2 2] için k-adım tahmin ve benzetim cevapları

Sonuçları daha ayrıntılı analiz etmek için, modelin k-adım öteye tahminleri ($k = 2, 10$ ve 20) ve benzetim için MSE, MAE ve G ölçütleri hesaplanmıştır. Modelin her iki senaryo için sonuçları Çizelge 4.1'de sunulmuştur.

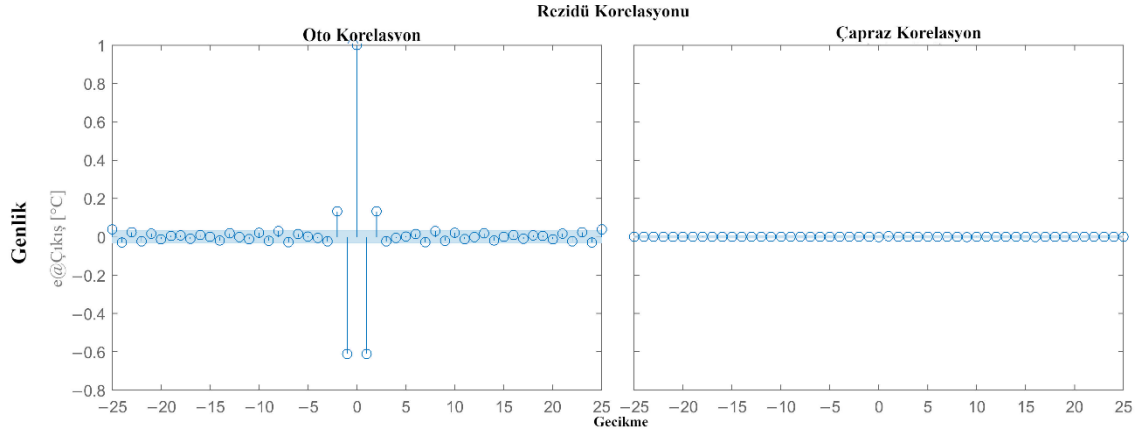
Çizelge 4.1. ARX [2 2] modelinin her iki senaryo için başarımları.

Doğrulama Ölçütü	G		MSE		MAE	
	Adım	PRBS	Adım	PRBS	Adım	PRBS
Tahmin k: 2	95,6526	98,7377	$5,47874 \times 10^{-5}$	$5,0382 \times 10^{-6}$	0,00494216	0,0016
Tahmin k: 10	86,1103	94,1006	0,00055925	0,00011005	0,0175226	0,0074
Tahmin k: 20	80,7099	88,3159	0,00107867	0,00043167	0,0244397	0,0145
Benzetim	77,2656	85,1858	0,00149826	0,00069394	0,0289541	0,0211

Ayrıca, model için kalıntı korelasyonları, otokorelasyon ve çapraz korelasyonlar, birinci senaryo için Şekil 4.3'te ve ikinci senaryo için Şekil 4.4'te gösterilmiştir.



Şekil 4.3. Birinci senaryoda ARX[2 2] için kalıntı korelasyonları



Şekil 4.4. İkinci senaryoda ARX[2 2] için kalıntı korelasyonları

4.1.2. ARMAX Modeli için Elde Edilen Bulgular

ARMAX modelinin model derecesi, birinci senaryo için $n_a = 2$, $n_b = 2$ ve $n_c = 2$ olarak belirlenmiştir. ARMAX [2 2 2] parametre vektörü θ , tanımlama veri seti-1 üzerinde tahmin edilmiştir. Birinci senaryoda ARMAX modeli için polinomlar Denklem (4.5) ve Denklem (4.7) arasındaki gibi elde edilmiştir.

$$A(q) = 1 - 1,99366q^{-1} - 0,993663q^{-2}, \quad (4.5)$$

$$B(q) = -6,32413 \times 10^{-5}q^{-1} - 6,62595 \times 10^{-5}q^{-2}, \quad (4.6)$$

$$C(q) = 1 - 1,32538q^{-1} - 0,328971q^{-2}, \quad (4.7)$$

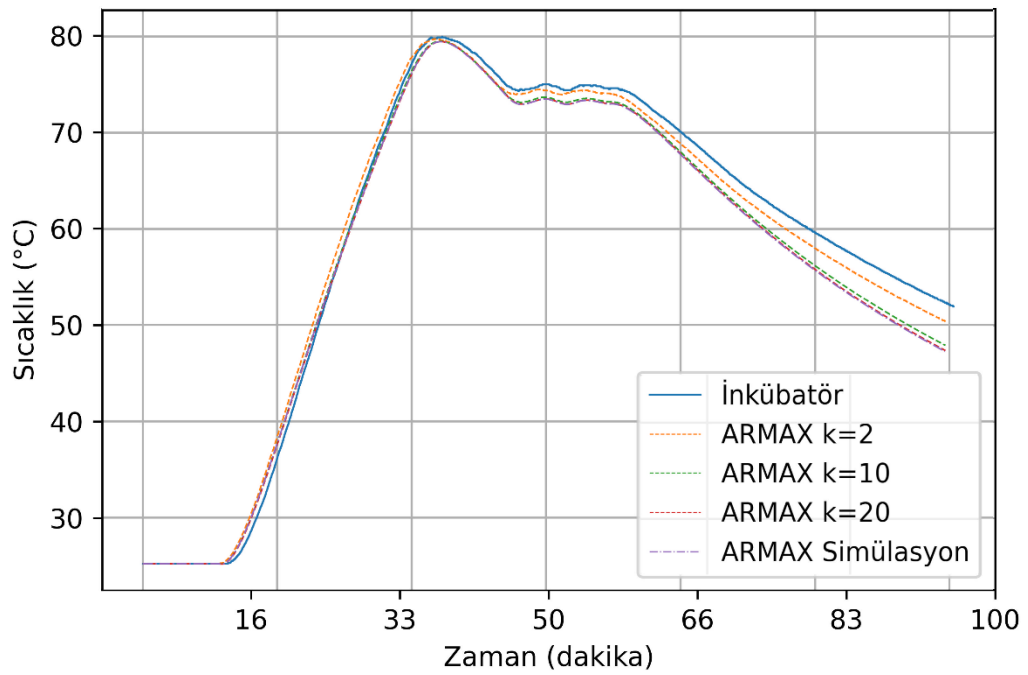
İkinci senaryo için de ARMAX modelinin model derecesi $n_a = 2$, $n_b = 2$ ve $n_c = 2$ olarak belirlenmiştir. ARMAX [2 2 2] parametre vektörü θ , tanımlama veri seti-2 üzerinde tahmin edilmiştir. İkinci senaryoda ARMAX modeli için polinomlar Denklem (4.8) ve Denklem (4.10) arasındaki gibi elde edilmiştir.

$$A(q) = 1 - 1,9947q^{-1} + 0,99474q^{-2}, \quad (4.8)$$

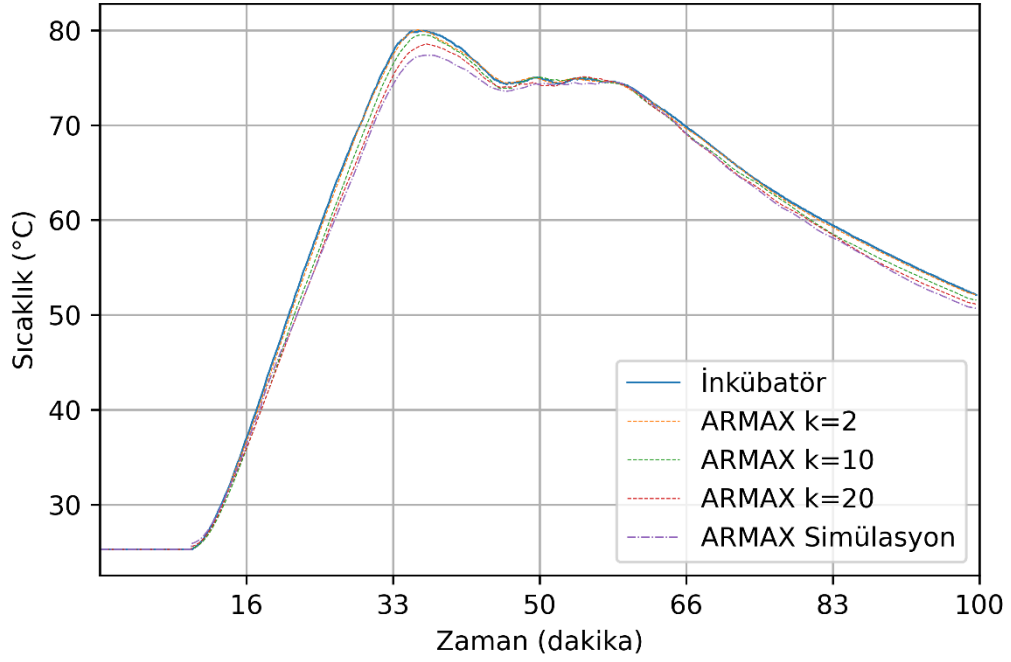
$$B(q) = -3,7203 \times 10^{-5}q^{-1} + 3,9892 \times 10^{-5}q^{-2}, \quad (4.9)$$

$$C(q) = 1 - 1,0346q^{-1} + 0,08449q^{-2}, \quad (4.10)$$

Modelin performans deęerlendirmesini yapmak amacıyla her iki senaryoda da aynı doęrulama veri seti kullanılmıřtır. Modelin 2 adım, 10 adım ve 20 adım ileri tahmin ve benzetim yanıtları doęrulama veri seti için hesaplanmıřtır. Her iki senaryonun yanıtları, inkübatörün gerek ıkıřı ile karřılařtırılmıř ve birinci senaryo için řekil 4.5'te, ikinci senaryo için ise řekil 4.6'da grselleřtirilmiřtir.



řekil 4.5. İlk senaryoda ARMAX[2 2] için k-adım tahmin ve benzetim cevapları



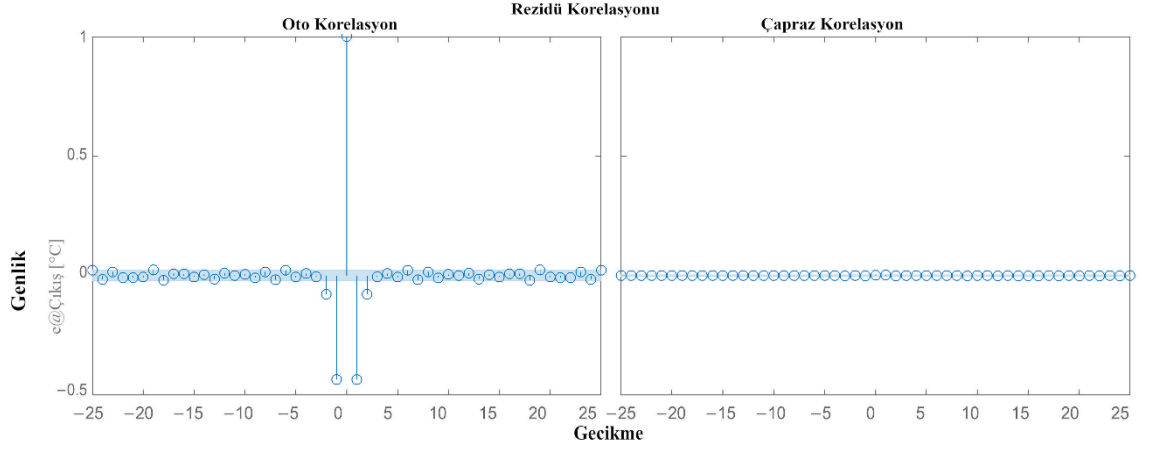
Şekil 4.6. İkinci senaryoda ARMAX[2 2 2] için tahmin ve benzetim cevapları

Sonuçları daha ayrıntılı analiz etmek için, modelin k-adım ileri tahminleri ($k = 2, 10$ ve 20) ve benzetim için MSE, MAE ve G ölçütleri hesaplanmıştır. Modelin her iki senaryo için sonuçları Çizelge 4.2'de sunulmuştur.

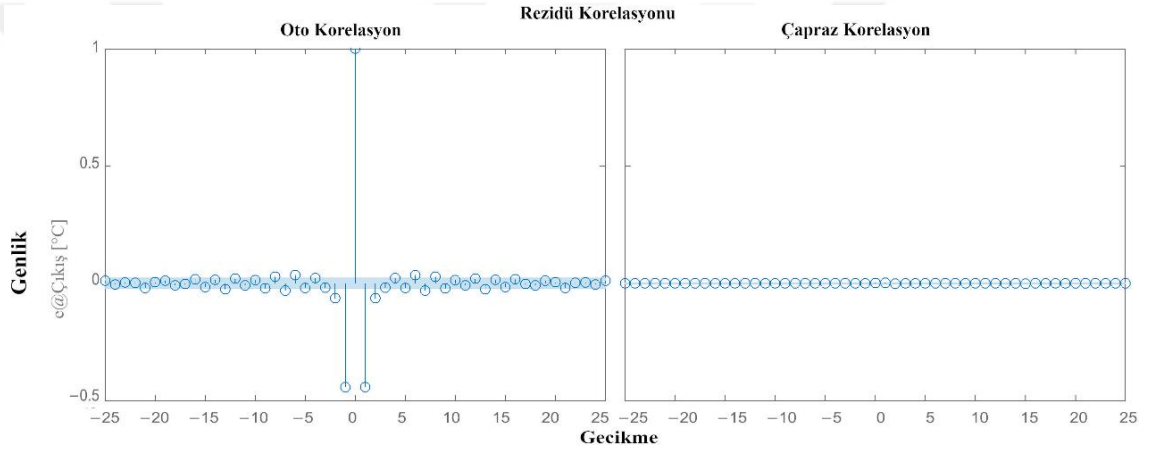
Çizelge 4.2. ARMAX [2 2 2] modelinin her iki senaryo için başarımları

Doğrulama Ölçütü	G		MSE		MAE	
	Adım	PRBS	Adım	PRBS	Adım	PRBS
Tahmin k: 2	94,9755	98,5564	$7,3182 \times 10^{-5}$	$6,6380 \times 10^{-6}$	0,0064504	0,0020
Tahmin k: 10	87,8214	95,2388	0,000429946	$7,2225 \times 10^{-5}$	0,0167575	0,0065
Tahmin k: 20	86,2470	92,2919	0,000548299	0,00018899	0,0189746	0,0105
Benzetim	86,1957	90,5396	0,000552393	0,00028334	0,0191662	0,0128

Ayrıca, model için kalıntı korelasyonları, otokorelasyon ve çapraz korelasyonlar, birinci senaryo için Şekil 4.7'de ve ikinci senaryo için Şekil 4.8'de gösterilmiştir.



Şekil 4.7. Birinci senaryoda ARMAX[2 2 2] için kalıntı korelasyonları



Şekil 4.8. İkinci senaryoda ARMAX[2 2 2] için kalıntı korelasyonları

4.1.3. Çıkış Hatası Modeli için Elde Edilen Bulgular

Çıkış hatası modelinin model derecesi, birinci senaryo için $n_b = 2$ ve $n_f = 2$ olarak belirlenmiştir. OE [2 2] parametre vektörü θ , tanımlama veri seti-1 üzerinde tahmin edilmiştir. Birinci senaryoda OE modeli için polinomlar Denklem (4.11) ve Denklem (4.12)'deki gibi elde edilmiştir.

$$B(q) = -5,54454 \times 10^{-5}q^{-1} + 5,89843 \times 10^{-5}q^{-2}, \quad (4.11)$$

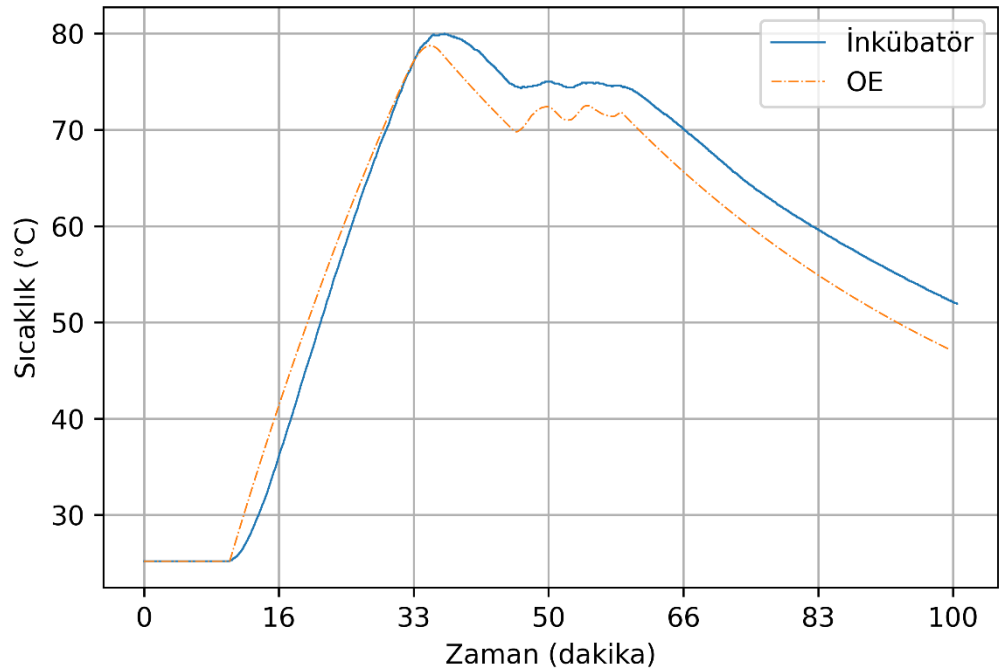
$$F(q) = 1 - 1,99252q^{-1} + 0,99252q^{-2}, \quad (4.12)$$

İkinci senaryo için de OE modelinin model derecesi $n_a = 2$ ve $n_b = 2$ olarak belirlenmiştir ve OE [2 2] parametre vektörü θ , tanımlama veri seti-2 üzerinde tahmin edilmiştir. İkinci senaryoda OE modeli için polinomlar Denklem (4.13) ve Denklem (4.14)'te olduğu gibi elde edilmiştir.

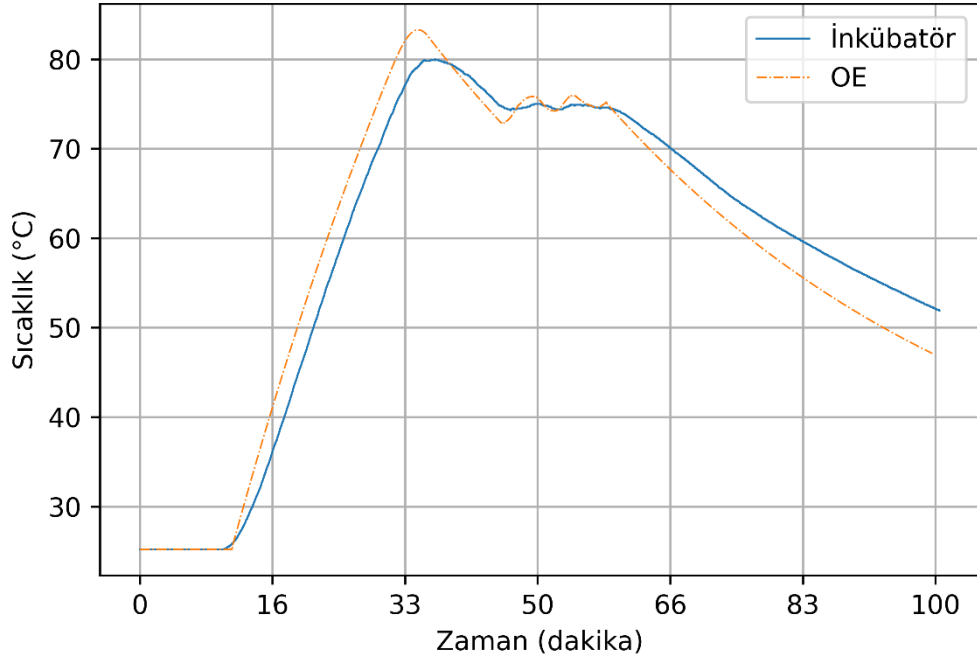
$$B(q) = -3,086 \times 10^{-5}q^{-1} + 3,409 \times 10^{-5}q^{-2}, \quad (4.13)$$

$$F(q) = 1 - 1,993q^{-1} + 0,9929q^{-2}, \quad (4.14)$$

Modelin performans değerlendirmesini yapmak amacıyla her iki senaryoda da aynı doğrulama veri seti kullanılmıştır. Modelin 2 adım, 10 adım ve 20 adım öteye tahmin ve benzetim yanıtları doğrulama veri seti için hesaplanmıştır. Her iki senaryonun yanıtları, inkübatörün gerçek çıkışı ile karşılaştırılmış ve birinci senaryo Şekil 4.9'da, ikinci senaryo için ise Şekil 4.10'da görselleştirilmiştir.



Şekil 4.9. İlk senaryoda OE[2 2] için k-adım tahmin ve benzetim cevapları



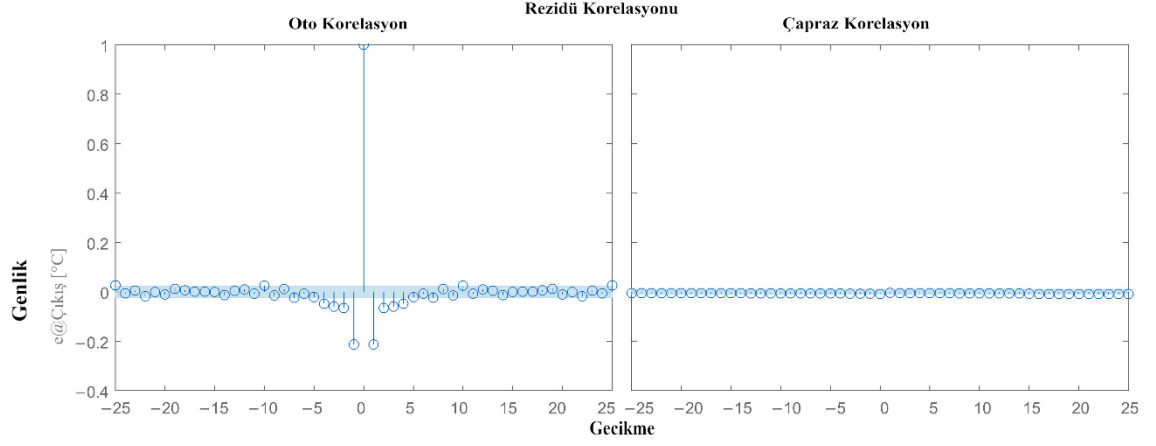
Şekil 4.10. İkinci senaryoda OE[2 2] için k-adım tahmin ve benzetim cevapları

Model için benzetim yanıtına yönelik MSE, MAE ve G ölçümleri hesaplanmıştır ve her iki senaryonun sonuçları Çizelge 4.3'te gösterilmiştir.

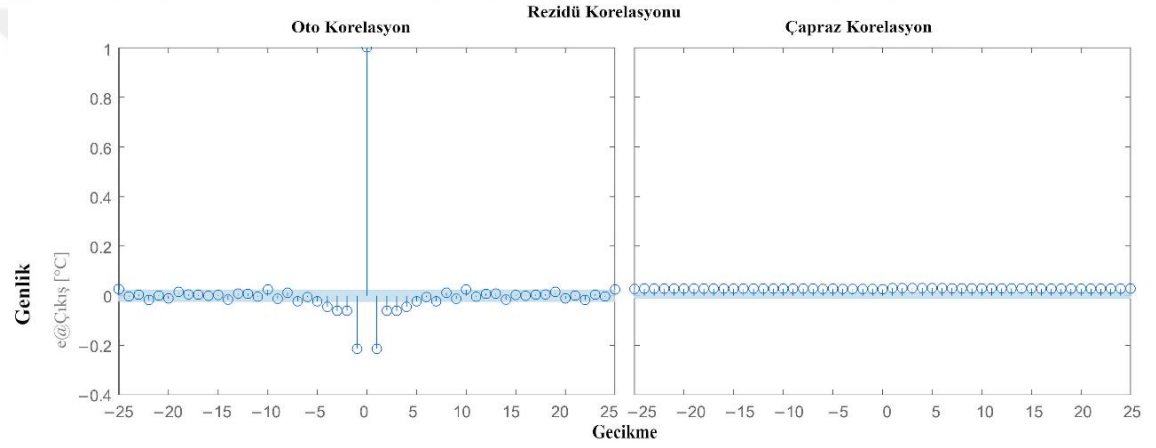
Çizelge 4.3. OE [2 2] modelinin her iki senaryo için başarımları.

Doğrulama Ölçütü	G		MSE		MAE	
	Adım	PRBS	Adım	PRBS	Adım	PRBS
Benzetim	83,0517	84,0369	0,000856901	0,00080505	0,0240122	0,0235

Ayrıca, model için kalıntı korelasyonları, otokorelasyon ve çapraz korelasyonlar, birinci senaryo için Şekil 4.11'de ve ikinci senaryo için Şekil 4.12'de gösterilmiştir.



Şekil 4.11. Birinci senaryoda OE[2 2] için kalıntı korelasyonları



Şekil 4.12. İkinci senaryoda OE[2 2] için kalıntı korelasyonları

4.1.4. Box-Jenkins Modeli için Elde Edilen Bulgular

Box-Jenkins modelinin model derecesi, birinci senaryo için $n_b = 2, n_c = 2, n_d = 2$ ve $n_f = 2$ olarak belirlenmiştir. BJ [2 2 2 2] parametre vektörü θ , tanımlama veri seti-1 üzerinde tahmin edilmiştir. Birinci senaryoda BJ modeli için polinomlar Denklem (4.15) ve Denklem (4.18) arasındaki gibi elde edilmiştir.

$$B(q) = -5,36891 \times 10^{-5}q^{-1} - 5,66662 \times 10^{-5}q^{-2}, \quad (4.15)$$

$$C(q) = 1 - 0,663758q^{-1} - 0,33229q^{-2}, \quad (4.16)$$

$$D(q) = 1 - 0,0039244q^{-1} - 0,995552q^{-2}, \quad (4.17)$$

$$F(q) = 1 - 1,99376q^{-1} + 0,993759q^{-2}, \quad (4.18)$$

İkinci senaryo için de Box-Jenkins modelinin model derecesi $n_b = 2, n_c = 2, n_d = 2$ ve $n_f = 2$ olarak belirlenmiştir ve BJ [2 2 2 2] parametre vektörü θ , tanımlama veri seti-2 üzerinde tahmin edilmiştir. İkinci senaryoda BJ modeli için polinomlar Denklem (4.19) ve Denklem (4.22) arasındaki gibi elde edilmiştir.

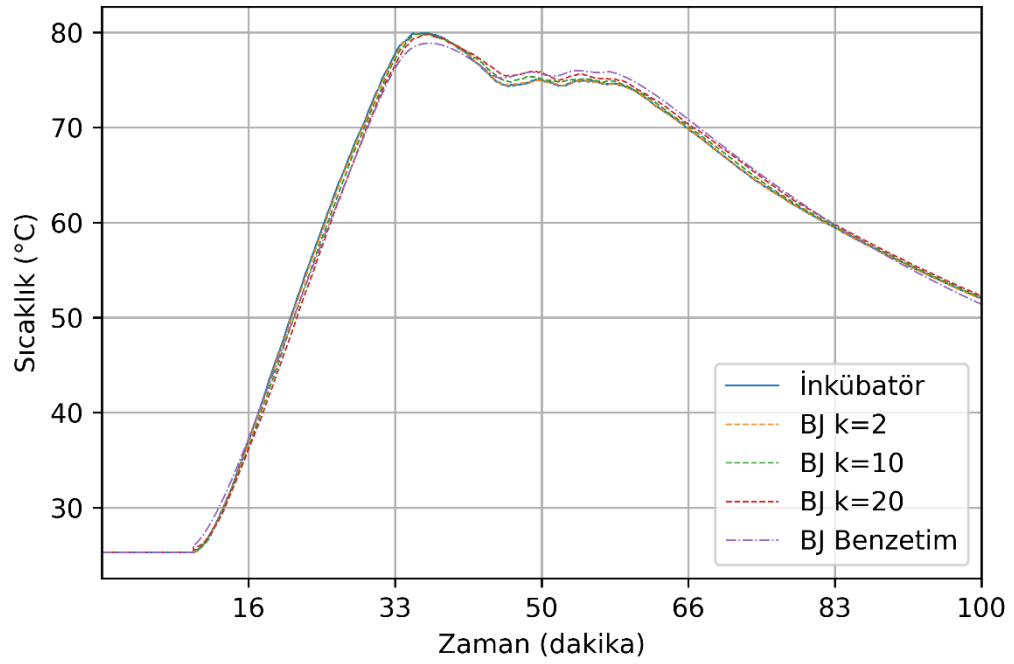
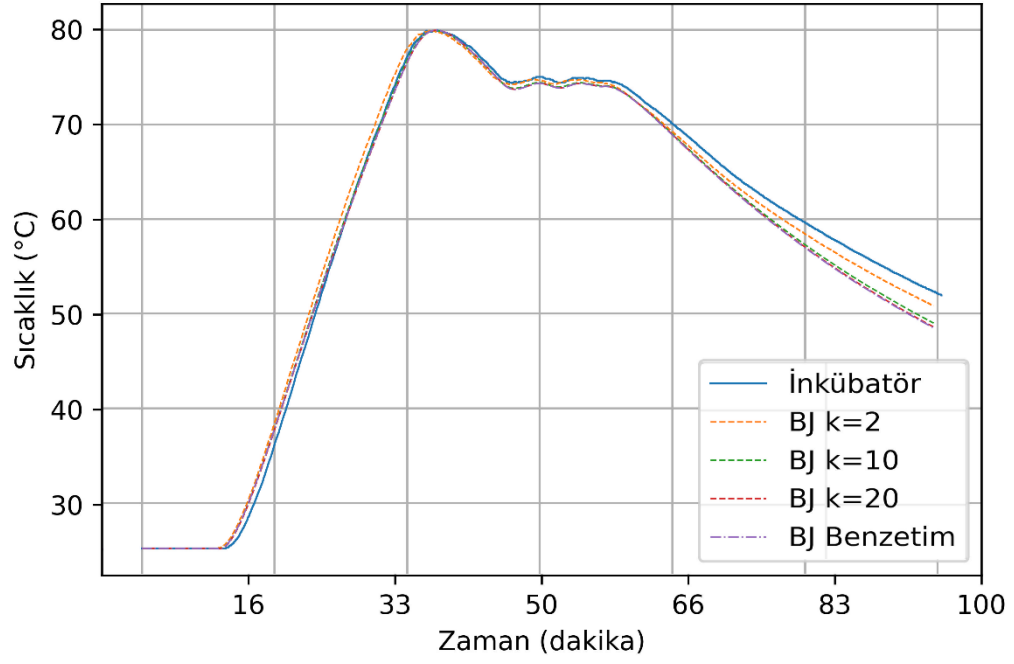
$$B(q) = 0,0001333q^{-1} - 0,0001312q^{-2}, \quad (4.19)$$

$$C(q) = 1 + 0,1308q^{-1} - 0,865q^{-2}, \quad (4.20)$$

$$D(q) = 1 - 1,296q^{-1} + 0,2967q^{-2}, \quad (4.21)$$

$$F(q) = 1 - 1,996q^{-1} + 0,9957q^{-2}, \quad (4.22)$$

Modelin performans değerlendirmesini yapmak amacıyla her iki senaryoda da aynı doğrulama veri seti kullanılmıştır. Modelin 2 adım, 10 adım ve 20 adım öteye tahmin ve benzetim yanıtları doğrulama veri seti için hesaplanmıştır. Her iki senaryonun yanıtları, inkübatörün gerçek çıkışı ile karşılaştırılmış ve birinci senaryo için Şekil 4.13'te, ikinci senaryo için ise Şekil 4.14'te görselleştirilmiştir.

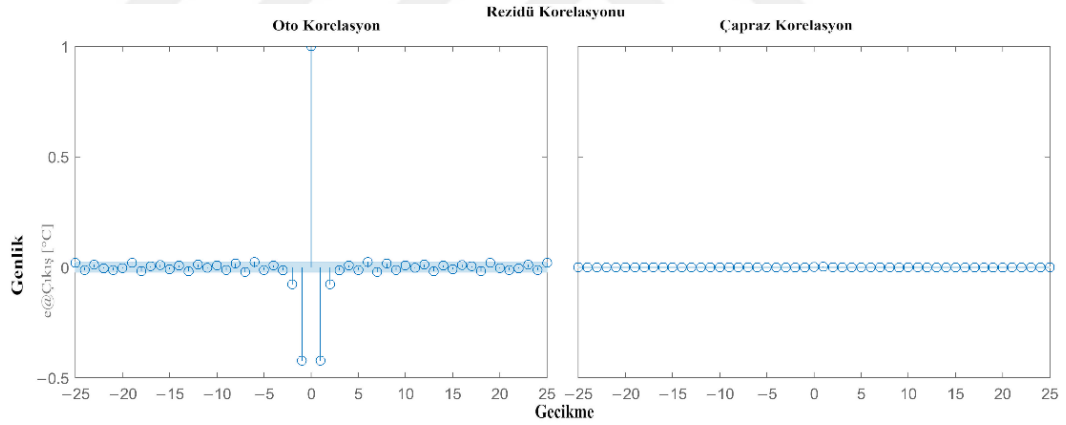


Sonuçları daha ayrıntılı analiz etmek için, modelin k-adım ileri tahminleri ($k = 2, 10$ ve 20) ve benzetim için MSE, MAE ve G ölçütleri hesaplanmıştır. Modelin her iki senaryo için sonuçları Çizelge 4.4'te sunulmuştur.

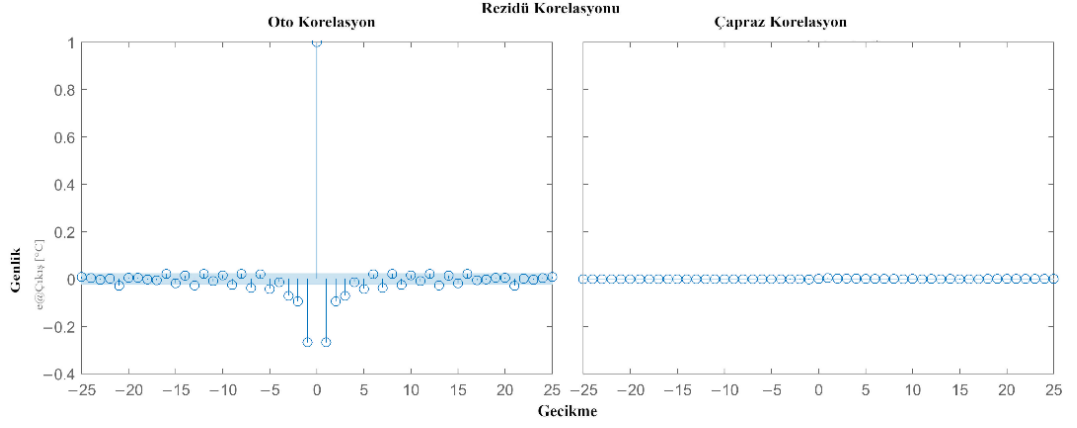
Çizelge 4.4. BJ [2 2 2 2] modelinin her iki senaryo için başarımları

Doğrulama Ölçütü	G		MSE		MAE	
	Adım	PRBS	Adım	PRBS	Adım	PRBS
Tahmin k: 2	95,6526	98,7377	$5,47874 \times 10^{-5}$	$5,0382 \times 10^{-6}$	0,00494216	0,0016
Tahmin k: 10	86,1103	94,1006	0,00055925	0,00011005	0,0175226	0,0074
Tahmin k: 20	80,7099	88,3159	0,00107867	0,00043167	0,0244397	0,0145
Benzetim	77,2656	85,1858	0,00149826	0,00069394	0,0289541	0,0211

Ayrıca, model için kalıntı korelasyonları, otokorelasyon ve çapraz korelasyonlar, birinci senaryo için Şekil 4.15'te ve ikinci senaryo için Şekil 4.16'da gösterilmiştir.



Şekil 4.15. Birinci senaryoda BJ[2 2 2 2] için kalıntı korelasyonları



Şekil 4.16. İkinci senaryoda BJ[2 2 2 2] için kalıntı korelasyonları

4.1.5. Model Yapılarının Başarımları

Çalışmada belirlenen modellerin performanslarının karşılaştırmalı bir analizini yapmak amacıyla, her iki senaryonun sonuçları Çizelge 4.5'te gösterildiği gibi tek bir tabloda sunulmuştur.

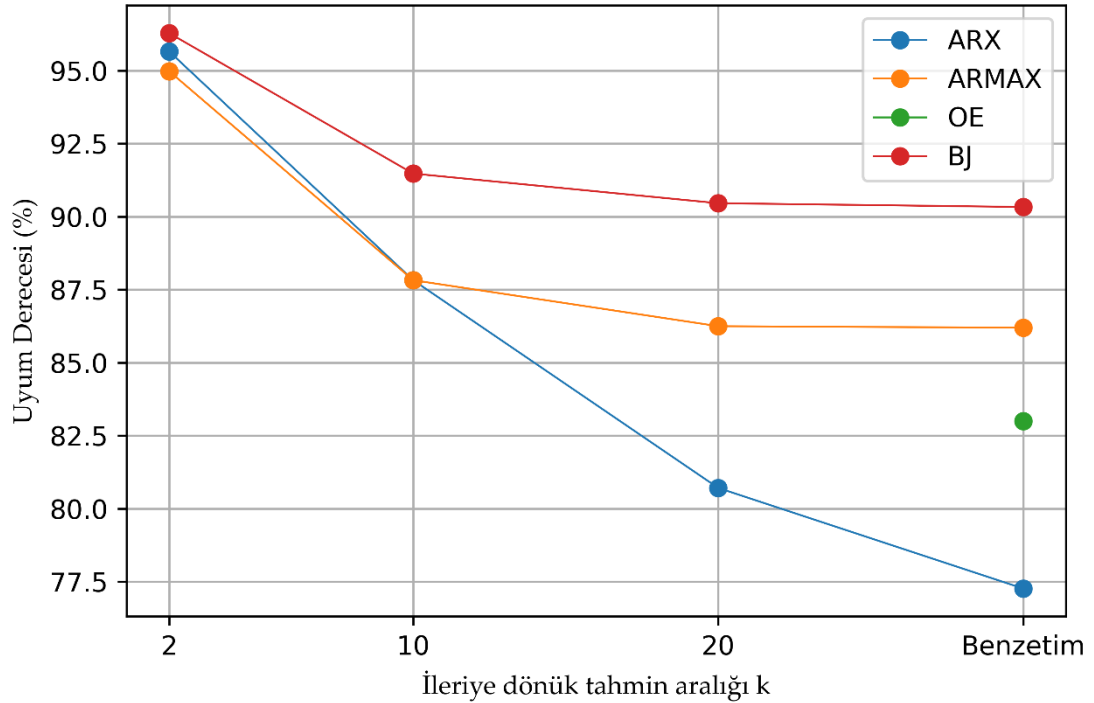
Çizelge 4.5. Çalışmada kullanılan modellerin karşılaştırılması

Doğrulama Kriteri	G		MSE		MAE	
	Adım	PRBS	Adım	PRBS	Adım	PRBS
ARX [2 2]						
Tahmin k: 2	95,6526	98,7377	$5,47874 \times 10^{-5}$	$5,0382 \times 10^{-6}$	0,00494216	0,0016
Tahmin k: 10	86,1103	94,1006	0,00055925	0,00011005	0,0175226	0,0074
Tahmin k: 20	80,7099	88,3159	0,00107867	0,00043167	0,0244397	0,0145
Benzetim	77,2656	85,1858	0,00149826	0,00069394	0,0289541	0,0211
ARMAX [2 2 2]						
Tahmin k: 2	94,9755	98,5564	$7,3182 \times 10^{-5}$	$6,638 \times 10^{-6}$	0,00645041	0,0020
Tahmin k: 10	87,8214	95,2388	0,000429946	$7,2225 \times 10^{-5}$	0,0167575	0,0065
Tahmin k: 20	86,247	92,2919	0,000548299	0,00018899	0,0189746	0,0105
Benzetim	86,1957	90,5396	0,000552393	0,00028334	0,0191662	0,0128
OE [2 2]						
Benzetim	83,0517	84,0369	0,000856901	0,00080505	0,0240122	0,0235
BJ [2 2 2 2]						

Tahmin k: 2	96,2799	99,4335	$4,01182 \times 10^{-5}$	$9,1998 \times 10^{-7}$	0,0041233	0,0007
Tahmin k: 10	91,4721	98,1200	0,000210816	$1,0131 \times 10^{-5}$	0,0108614	0,0021
Tahmin k: 20	90,4577	96,4682	0,000263956	$3,5756 \times 10^{-5}$	0,0124288	0,0040
Benzetim	90,3288	95,0771	0,000271132	$7,0355 \times 10^{-5}$	0,0123917	0,0074

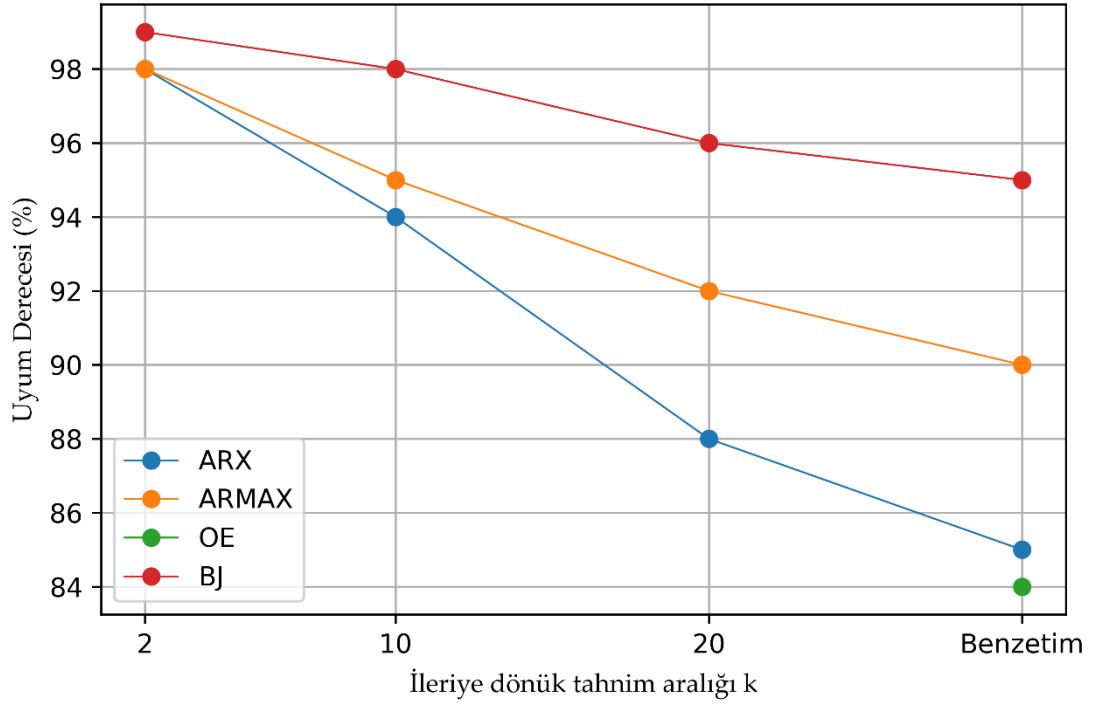
Çizelge 4.5'te görüldüğü gibi, tüm k-adım-ileri tahminler ve benzetim yanıtları için en iyi sonuçlar, her iki senaryo için de BJ modeli tarafından elde edilmiştir.

Ayrıca, görsel yorumlama yapmak amacıyla, her iki senaryo için de Şekil 4.17'de ve Şekil 4.18'de uyum dereceleri ile ileriye dönük tahmin aralılarının karşılaştırıldığı grafikler verilmiştir. Benzetim yanıtının sonsuz ufuk tahminine karşılık gelmesi nedeniyle, bu yanıtlar figürlerin tahmin ufku ekseninde en sağda yer almaktadır.



Şekil 4.17. İlk senaryo için tahmin aralığına göre uyum dereceleri

Şekil 4.17'den, birinci senaryo için şu sonuçlar çıkarılabilir; ARX, k = 2 için ARMAX'tan daha iyi bir G değerine sahipken, ARMAX k = 10 ve 20 adım-ileri tahminler ve benzetim için daha iyi G değerlerine sahiptir.



Şekil 4.18. İkinci senaryo için tahmin aralığına göre uyum dereceleri

OE'nin, benzetim için G değeri ARX'ten yüksek olmasına rağmen, pratik uygulamalarda kullanılan inkübatörler için uygun bir model değildir. Bunun nedeni, OE'nin giriş sinyalindeki değişiklikler nedeniyle benzetim ve gerçek çıkış arasında yüksek sapmalara sahip olmasıdır. BJ, tüm k-adım-ileri tahminler ve benzetim dahil olmak üzere en iyi G değerlerine sahiptir.

Şekil 4.18'ten, ikinci senaryo için şu sonuçlar çıkarılabilir: ARX, ARMAX ve BJ $k = 2$ için benzer G değerlerine sahipken, ARMAX $k = 10$ ve 20 adım-ileri tahminler ve benzetim için ARX'ten daha iyi G değerlerine sahiptir. BJ tüm yanıtlar için en iyi G değerlerine sahiptir ve benzetim için en kötü G değeri OE'ye aittir.

Her iki senaryo için yapılan kalıntı analizi, OE dışındaki tüm modellerin iyi bir güven düzeyine sahip olduğunu göstermektedir. ARX, ARMAX ve BJ modellerinin çapraz korelasyon grafikleri, kalıntılar ve giriş arasındaki çapraz korelasyonların, çalışma için %99 olarak belirlenen güven bölgesi içinde olduğunu göstermektedir.

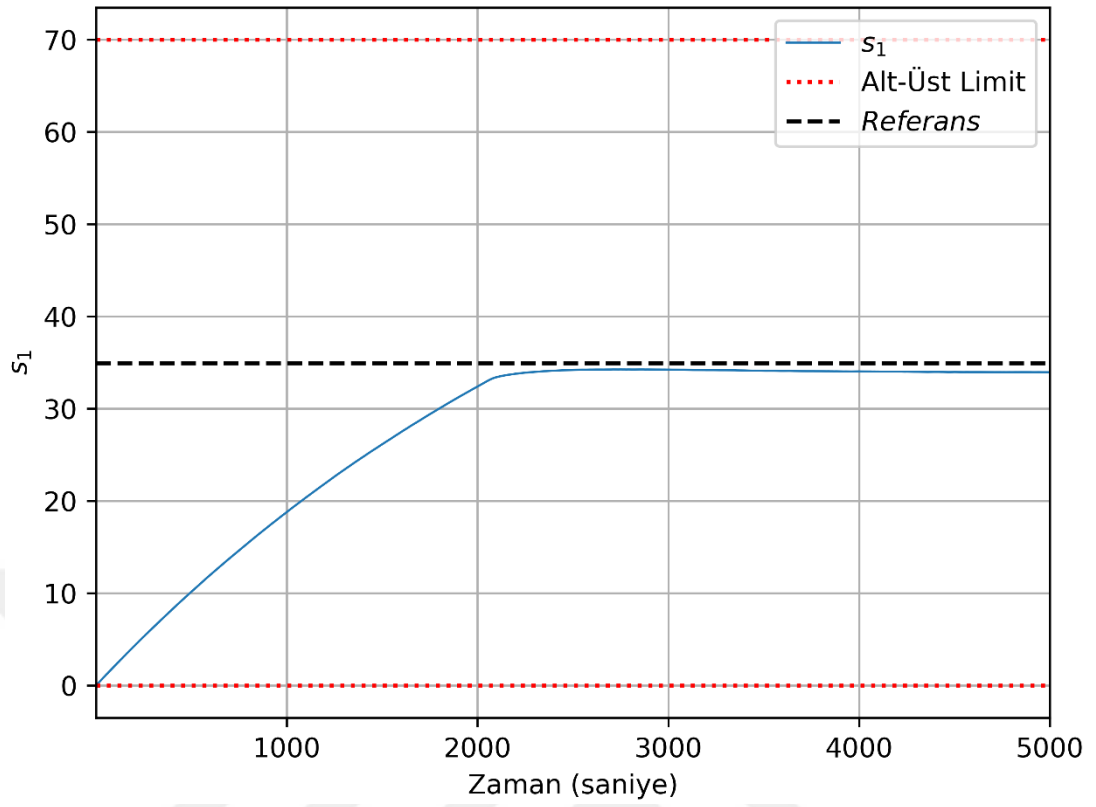
4.2. MPC Tabanlı Pekiştirmeli Öğrenme Kontrol Sisteminin Başarımı

Deney çalışmaları, laboratuvar inkübatör modeli için Python dili ile geliştirilen benzetim ortamında yapılmıştır. Pekiştirmeli öğrenme ortamı olarak Gym (Brockman vd., 2016) kütüphanesi kullanılmıştır. Doğrusal olmayan optimizasyon problemlerinin çözümleri Casadi ve Ipopt kütüphanelerinin yardımı ile çözülmüştür. Inkübatör modeli gerçek proses olarak kabul edilmiş olup, ve hatalı olarak modellenen MPC sistemi için yaklaşık bir model olarak kabul edilmiştir. MPC sisteminin öngörü ufku N , pekiştirmeli öğrenme sisteminin indirim faktörü γ ve Newton yönteminin adım değeri α ızgara arama algoritması ile 0,05 olarak seçilmiştir. Öğrenme çevrimi için maksimum öğrenme adımı 5000 olarak seçilmiş olup, ayırık zamanlı modellerin örnekleme periyodu 1 saniyedir. Öğrenme çevrimi boyunca θ parametre vektörünün aldığı değerler her adım için kaydedilmiştir.

Inkübatör modelinin durum referansı, gerçek sistemin %50 çıkış ile kararlı durumuna karşılık gelen yaklaşık durum değerleri, $s_1=35$ ve $s_2=-2,7$ olarak belirlenmiştir. Inkübatör modelinin durum geçişleri sırasında, durum vektörüne $[-10^{-2}, 0]$ aralığında rasgele, bağımsız ve uniform dağılımlı bir hata ilave edilmektedir.

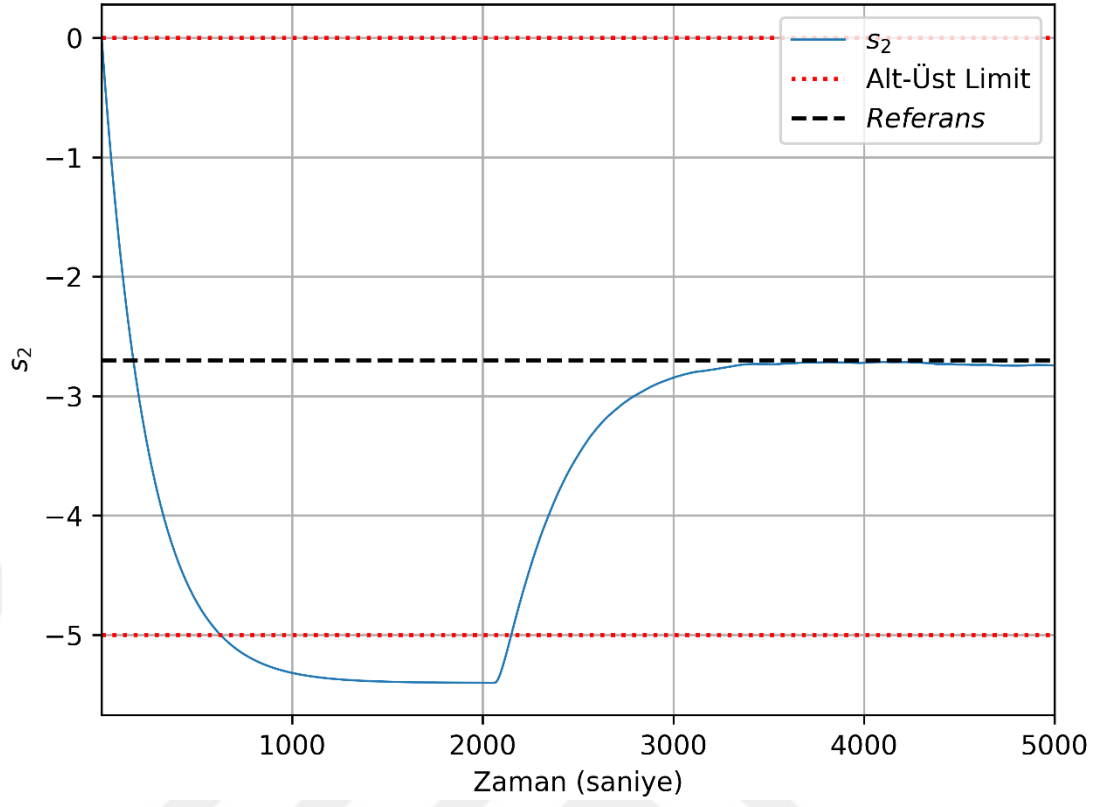
4.2.1. Inkübatör Modelinin Durum Değerleri

Gerçek sistem olarak kabul edilen inkübatör modelinin kontrol benzetimi sırasında, s_1 ve s_2 durumlarının zamana bağlı olarak aldığı değerler Şekil 4.19 ve Şekil 4.20’de gösterilmiştir. s_1 durumunun referans değerine yakınsadığı, alt ve üst limit değerlerini aşmadığı görülmektedir.



Şekil 4.19. s_1 durumunun zamana bağlı değişimi

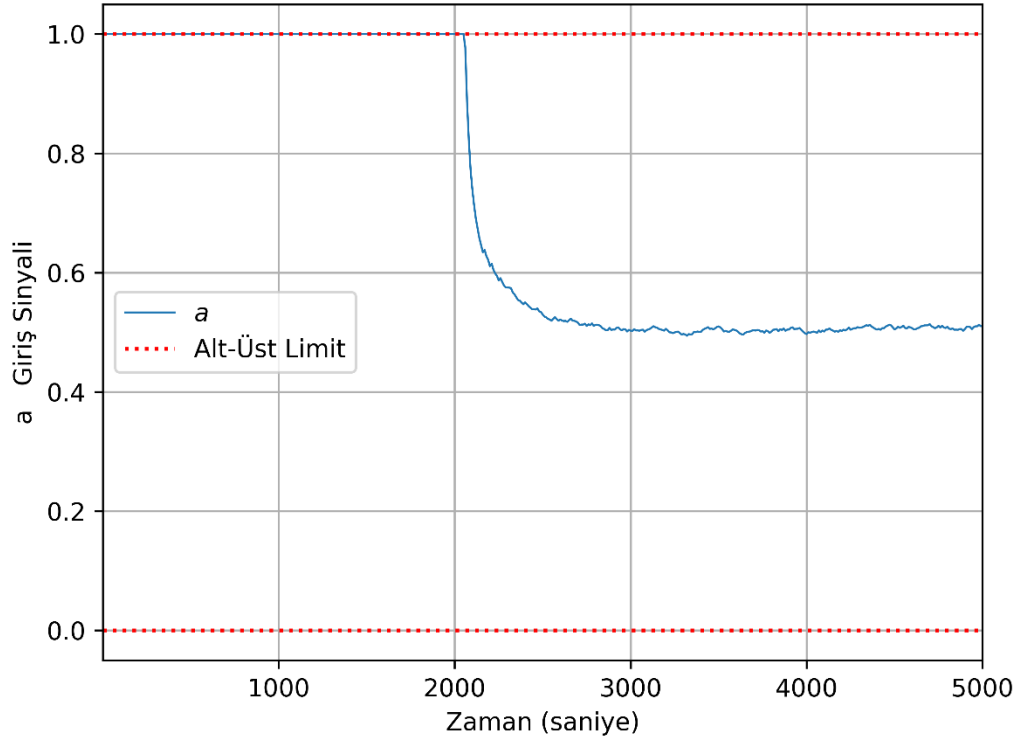
s_2 durumunun, öğrenmenin başlangıcında gevşeklik parametrelerinden dolayı alt limiti aştığı, fakat sonrasında referans durumuna yakınsadığı görülmektedir.



Şekil 4.20. s_2 durumunun zamana bağlı değişimi

4.2.2. MPC Sistemi Tarafından Uygulanan Giriş Sinyali

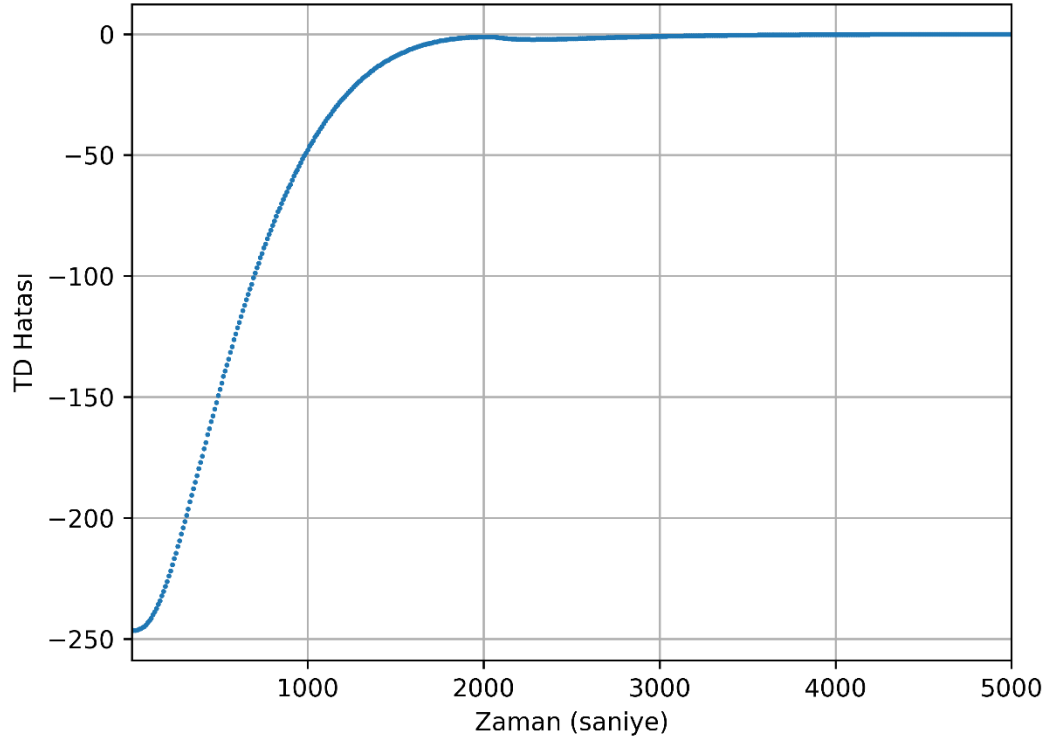
Hatalı bir model ile tasarlanan MPC sistem tarafından uygulanan kontrol sinyali Şekil 4.21’de gösterilmektedir. Öğrenme sürecinin başlarında salınım göstermesine rağmen, durum referanslarını takip ederken bozulmayı bastırabilecek şekilde bir başarımlı göstermiştir.



Şekil 4.21. a giriş sinyalinin zamana bağlı değişimi

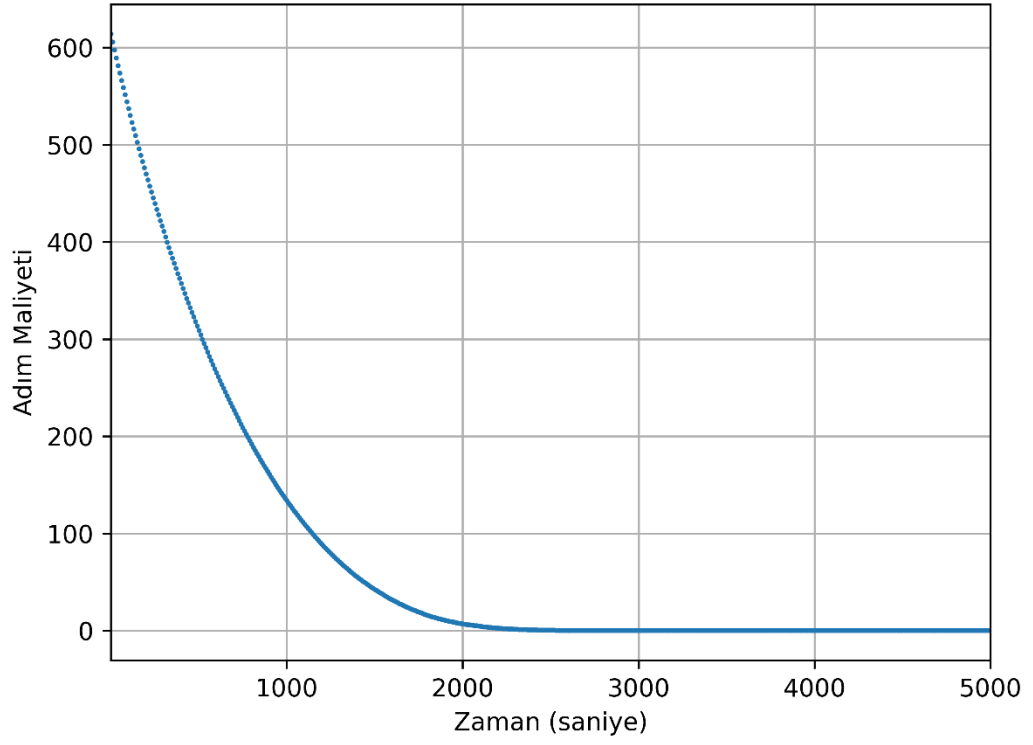
4.2.3. Zamansal Fark Hatası ve Adım Maliyeti

Pekiştirmeli öğrenme çevrimleri sırasında hesaplanan τ zamansal fark hatası ve L adım maliyeti, Şekil 4.22 ve Şekil 4.23'te gösterilmiştir. Sistemin öğrenme süreciyle birlikte durağan hale gelmesinden dolayı, zamansal fark hatası, belirli bir süre sonra 0'a yakınsamaktadır.



Şekil 4.22. Zamansal fark hatasının zamana bağlı değişimi

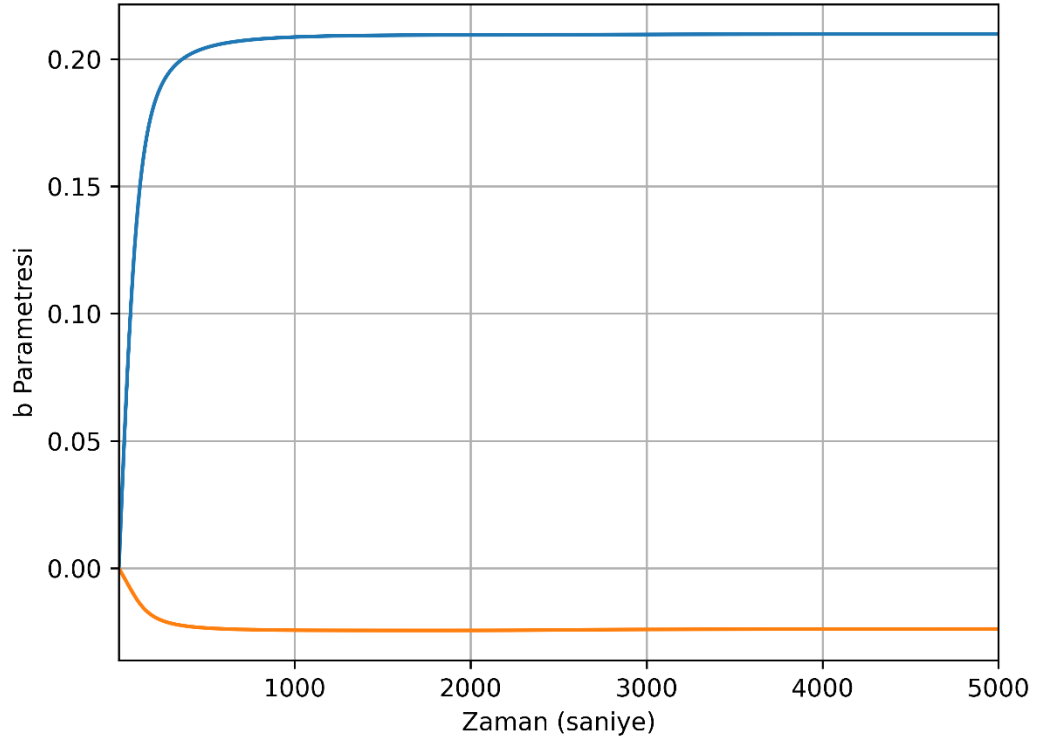
Adım maliyeti fonksiyonu, sadece referans ve anlık durum değerine bağlı olduğundan 0'a yakınsamaktadır.



Şekil 4.23. Adım maliyetinin zamana bağlı değişimi

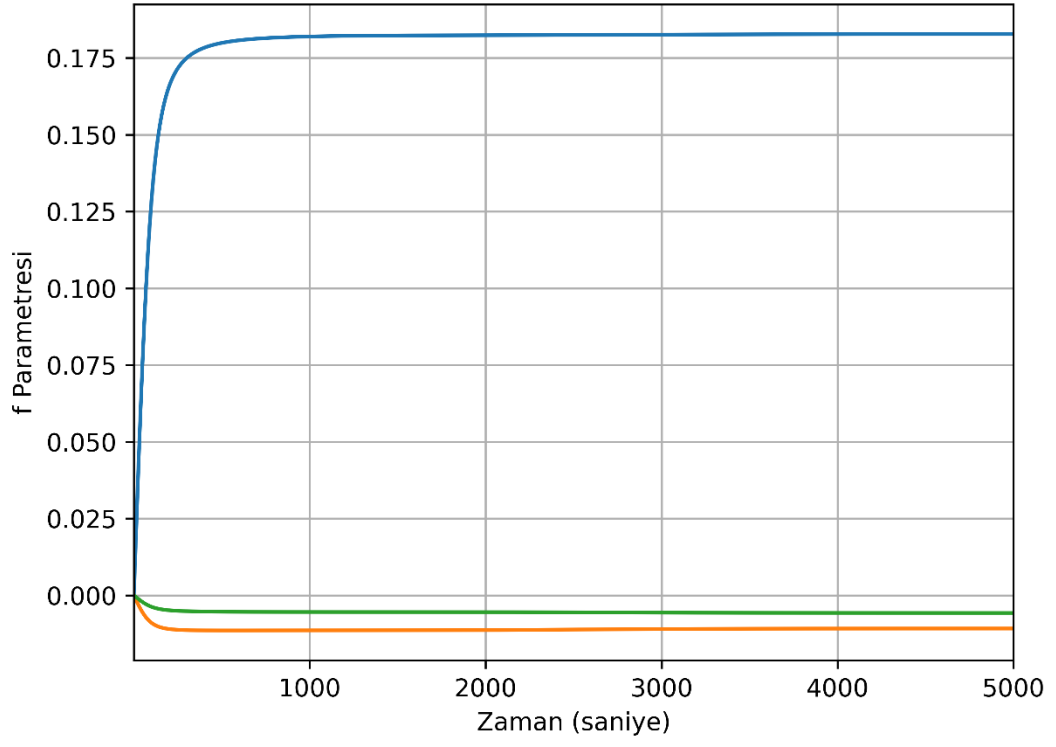
4.2.4. b , f ve V_0 Parametreleri

Durum geçiş matrisi ayar parametresi b , adım maliyet gradyanı parametresi f ve V_0 başlangıç maliyeti parametrelerinin zamana bağlı değişimi Şekil 4.24, Şekil 4.25 ve Şekil 4.26 üzerinde gösterilmiştir.



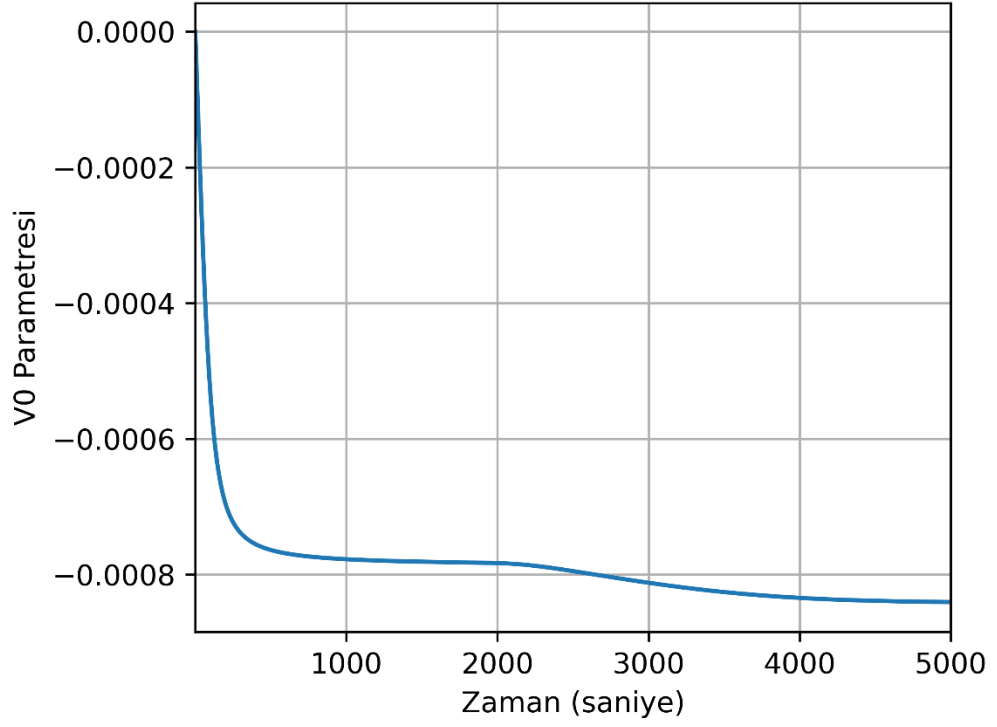
Şekil 4.24. b parametre vektörünün zamana bağlı değişimi

b parametre vektörünün $[0,20984716, -0,0238836]$ değerine yakınsadığı görülmektedir.



Şekil 4.25. f parametre vektörünün zamana bağlı değişimi

f parametre vektörünün $[0,18277509, -0,01081207, -0,00576989]$ değerine yakınsadığı görülmektedir.

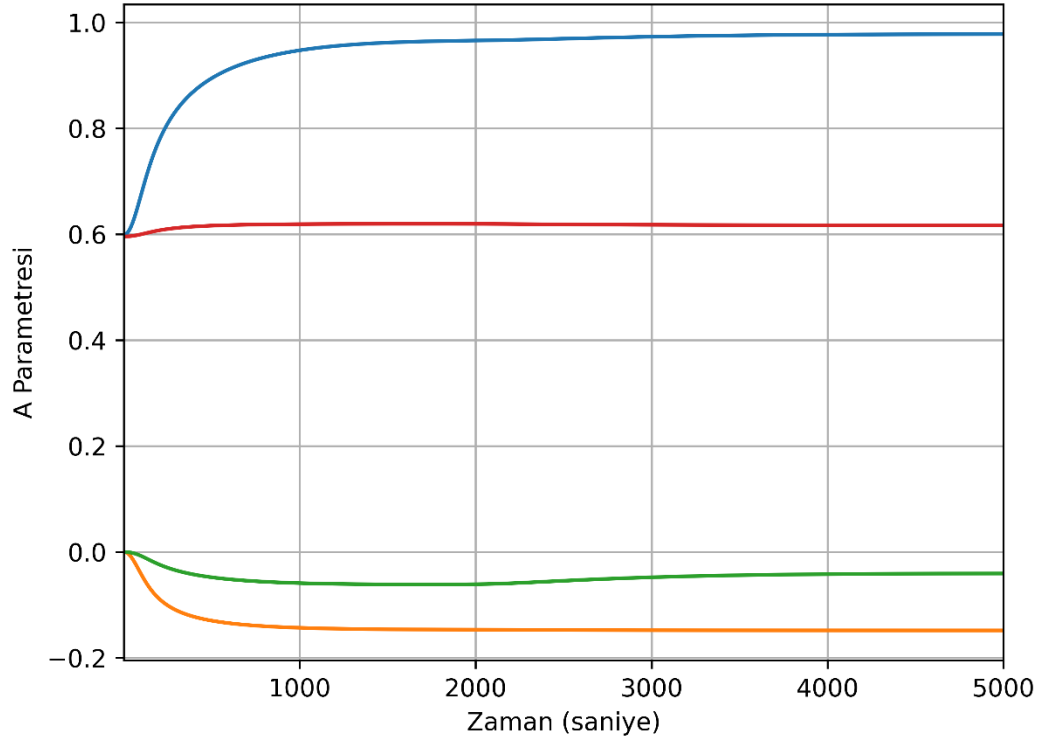


Şekil 4.26. V0 parametresinin zamana bağlı değişimi

V0 parametresinin $-0,00084202$ değerine yakınsadığı görülmektedir. Parametre değerlerinin sabit değerlere yakınsadığı gözlemlenmiştir.

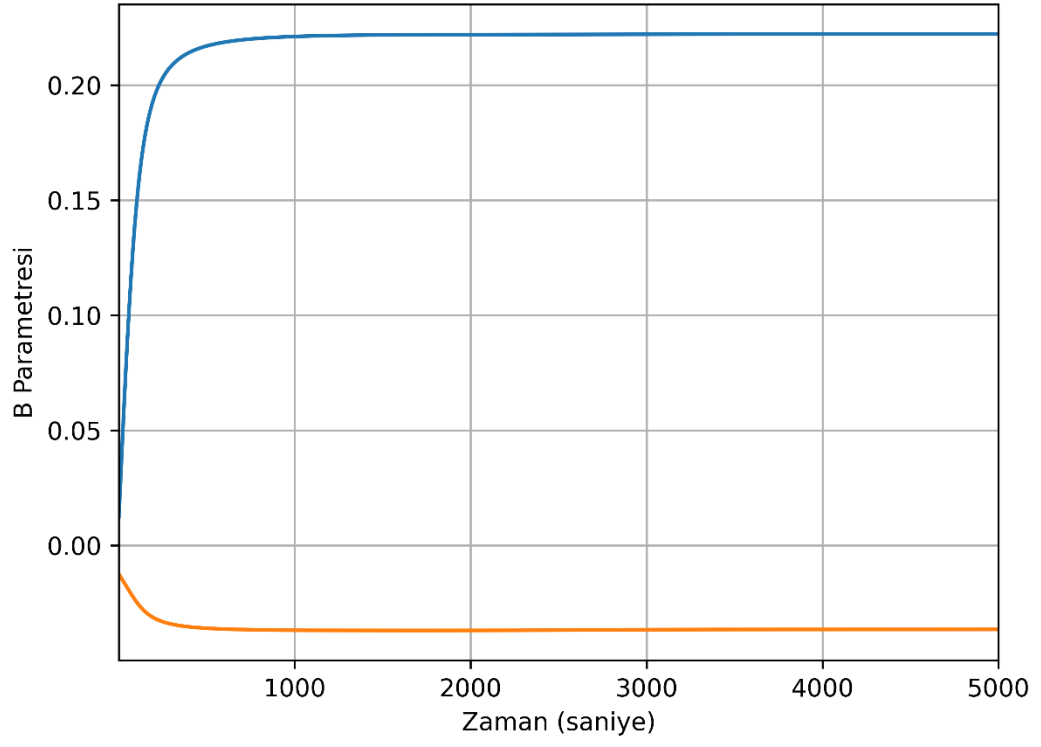
4.2.5. A ve B Parametreleri

A ve B parametre vektörlerinin zamana bağlı değişimi Şekil 4.27 ve Şekil 4.28’de gösterilmektedir. Görüldüğü gibi, parametre vektörleri gerçek sistemin durum ve giriş vektörlerine yakınsamadığından, MPC tabanlı RL yaklaşımı sistem tanımlama yapmamaktadır (Gros ve Zanon, 2020).



Şekil 4.27. A parametre vektörünün zamana bağlı değişimi

A parametre vektörünün $[[0,97802793 \ -0,14814163],[-0,04029689 \ 0,61703898]]$ değerine yakınsadığı görülmektedir.



Şekil 4.28. B parametre vektörünün zamana bağlı değişimi

B parametre vektörünün $[[0,22224674],[-0,0364714]]$ değerine yakınsadığı görülmektedir.

5. SONUÇ

Bu tez çalışmasında, laboratuvar inkübatör cihazının ekonomik maliyet fonksiyonu ve çoklu kısıtlar dikkate alınarak, model öngörülü kontrol (MPC) tabanlı pekiştirmeli öğrenme (RL) algoritması ile kontrol edilmesi amaçlanmıştır. MPC sisteminin, yanlış bir model kullanılarak tasarlanırsa bile, makine öğrenmesi temelli yöntemler ile gerçek sistemin optimal kontrol politikasını sağlayacak şekilde ayarlanabileceği gösterilmiştir.

Pekiştirmeli öğrenme sisteminin değer fonksiyonları, fonksiyon yaklaşımıcısı olarak doğrusal olmayan model öngörülü kontrol (NMPC) sistemi kullanılarak ifade edilmiştir. Aşama maliyeti, terminal maliyeti, kısıtlar ve inkübatör cihazının tahmini sistem modeli, MPC sistemi içerisinde parametrik olarak tanımlanmıştır. MPC sistemi, doğrudan çoklu atış yöntemi ile doğrusal olmayan programlama (NLP) problemi olarak ifade edilmiş ve elde edilen optimizasyon problemi iç nokta yöntemi ve otomatik türev alma yazılımı ile çözülmüştür. MPC sisteminin parametreleri, en küçük kareler zamansal fark (LSTD) Q-öğrenme yöntemi ile veriye dayalı olarak ayarlanmıştır.

MPC sisteminin tasarımında ve benzetim ortamında kullanılmak üzere, inkübatör cihazının matematiksel modeli sistem tanımlama yöntemi kullanılarak elde edilmiştir. İnkübatör, doğrusal zamanla değişmeyen tek giriş-tek çıkış (LTI SISO) bir sistem olarak ele alınmış, modelleme için ARX, ARMAX, OE ve Box-Jenkins (BJ) olmak üzere dört farklı model yapısı uygulanmıştır. Bu modellerin parametreleri, iki farklı senaryoda elde edilen tanımlama veri setlerinde tahmin hatası yöntemi (PEM) kullanılarak tahmin edilmiştir. Model yapılarını değerlendirmek için ortalama karesel hata (MSE), ortalama mutlak hata (MAE) ve uyum iyiliği ölçütleri kullanılmış, ayrıca otokorelasyon ve çapraz korelasyon grafikleri içeren kalıntı analizi yapılmıştır.

Deneyler, sisteme birim basamak girişi ve sözde rastgele ikili dizi (PRBS) girişi uygulanarak iki farklı senaryoda gerçekleştirilmiştir. Deneysel sonuçlar, hem MSE ve MAE hem de uyum iyiliği ölçütleri açısından, BJ modelinin her iki senaryoda da en iyi sonuçları verdiğini göstermektedir. BJ modeli, birinci senaryo için %90'ın üzerinde, ikinci senaryo için ise %95'in üzerinde uyum yüzdesine sahip olup, yüksek performans

sergilemiştir. Ayrıca, kalıntı analizi, BJ modelinin her iki senaryoda da kayda değer bir güven aralığına sahip olduğunu göstermektedir.

Bu sonuçlara dayanarak, çalışmada belirlenen BJ modelinin, benzer konfigürasyonlara sahip laboratuvar inkübatörleri için başarılı bir model olarak kullanılabileceği ve kontrol sistemlerinin tasarım aşamasında bir benzetim modeli olarak katkı sağlayabileceği sonucuna varılmıştır. Bu çalışmanın gelecekteki araştırma yönü, BJ modelinin daha geniş sıcaklık aralıklarına sahip laboratuvar inkübatörlerinin kontrol sistemi tasarımı için bir benzetim modeli olarak kullanılmasıdır.

Elde edilen sonuçlar, MPC tabanlı RL yönteminin endüstriyel kontrol sistemlerinde karşılaşılan zorlukların üstesinden gelmek için güçlü bir araç olduğunu göstermektedir. Özellikle sistem belirsizliklerinin ve kısıtlamalarının olduğu durumlarda, bu yöntem sistem performansını optimize etmek ve istenen referans değerini takip etmek için etkili bir çözüm sunar. Çevrimiçi öğrenme yeteneği sayesinde, sistem dinamiklerindeki değişikliklere hızlı bir şekilde uyum sağlayabilir ve kontrol performansını koruyabilir.

Sonuç olarak, bu tez çalışması, MPC tabanlı RL yönteminin endüstriyel kontrol sistemlerinin tasarımında ve uygulamasında yeni bir perspektif sunduğunu göstermektedir. Gelecekte, bu yöntemin daha karmaşık sistemlere ve farklı endüstriyel uygulamalara uyarlanması beklenmektedir. Özellikle enerji verimliliği, üretim maliyeti ve ürün kalitesi gibi performans kriterlerini optimize etmek için MPC tabanlı RL yönteminin kullanılması, endüstriyel süreçlerin verimliliğini ve sürdürülebilirliğini artırmada önemli bir potansiyele sahiptir.

KAYNAKLAR

- Abbeel, P., Coates, A., Quigley, M., Ng, A. 2006. "An application of reinforcement learning to aerobatic helicopter flight". *Advances in neural information processing systems*, 19.
- Akaike, H. 1976. "Canonical correlation analysis of time series and the use of an information criterion". İçinde *Mathematics in science and engineering* (C. 126, ss. 27–96). Elsevier.
- Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., Diehl, M. 2019. "{CasADi} -- {A} software framework for nonlinear optimization and optimal control". *Mathematical Programming Computation*, 11(1), 1–36.
- Andersson, J., Åkesson, J., Diehl, M. 2012. "Dynamic optimization with CasADi". İçinde *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (ss. 681–686).
- Armenise, G., Vaccari, M., Di Capaci, R. B., Pannocchia, G. 2018. "An Open-Source System Identification Package for Multivariable Processes". 2018 UKACC 12th International Conference on Control, CONTROL 2018, 152–157.
- Åström, K. J., Bohlin, T., Wensmark, S. 1965. "Automatic construction of linear stochastic dynamic models for stationary industrial processes with random disturbances using operating records".
- Aswani, A., Gonzalez, H., Sastry, S. S., Tomlin, C. 2013. "Provably safe and robust learning-based model predictive control". *Automatica*, 49(5), 1216–1226.
- Bagherian, D., Gornet, J., Bernstein, J., Ni, Y. L., Yue, Y., Meister, M. 2021. "Fine-Grained System Identification of Nonlinear Neural Circuits". *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 14–24.
- Beintema, G. I., Schoukens, M., Tóth, R. 2023. "Deep subspace encoders for nonlinear system identification". *Automatica*, 156, 111210.
- Bellman, R. 1952. "On the Theory of Dynamic Programming". *Proceedings of the National Academy of Sciences*, 38(8), 716–719.
- Bellman, R. 1957. "Dynamic Programming". İçinde *Mathematics in Science and Engineering* (C. 40, ss. 101–137). Princeton University Press.
- Bnhamdoon, O. A. A., Mohamad Hanif, N. H. H., Akmeliawati, R. 2020. "Identification of a quadcopter autopilot system via Box–Jenkins structure". *International Journal of Dynamics and Control*, 8(3), 835–850.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W. 2016. "Openai gym". arXiv preprint arXiv:1606.01540.
- Büskens, C., Maurer, H. 2001. "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems". İçinde *Online Optimization of Large Scale Systems* (ss. 3–16). Springer.
- Cappuccino, J. G., Welsh, C. T. 2016. "*Microbiology: A Laboratory Manual*". Pearson Education. Tarihiinde adresinden erişildi
<https://books.google.com.tr/books?id=OtF5CwAAQBAJ>

- Chow, Y., Nachum, O., Duenez-Guzman, E., Ghavamzadeh, M. 2018. "A lyapunov-based approach to safe reinforcement learning". *Advances in Neural Information Processing Systems, 2018-Decem(NeurIPS)*, 8092–8101.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. 2001. "*Introduction to Algorithms*" (2nd baskı). MIT Press & McGraw–Hill.
- Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., ... Riedmiller, M. 2022. "Magnetic control of tokamak plasmas through deep reinforcement learning". *Nature*, 602(7897), 414–419.
- Desborough, L., Miller, R. 2002. "Increasing customer value of industrial control performance monitoring - Honeywell's experience". *AIChE Symposium Series*, 98(326), 169–189.
- Diehl, M., Gros, S. 2011. "Numerical optimal control". *Optimization in Engineering Center (OPTEC)*.
- Ellis, M., Durand, H., Christofides, P. D. 2014. "A tutorial review of economic model predictive control methods". *Journal of Process Control*, 24(8), 1156–1178.
- Esfahani, H. N., Kordabad, A. B., Gros, S. 2021. "Approximate robust NMPC using reinforcement learning". İçinde *2021 European Control Conference (ECC)* (ss. 132–137).
- Fatima, S. K., Abbas, S. M., Mir, I., Gul, F., Mir, S., Saeed, N., ... Abualigah, L. 2022. "Data Driven Model Estimation for Aerial Vehicles: A Perspective Analysis". *Processes*, 10(7), 1–23.
- Fotouhi, A., Auger, D. J., Propp, K., Longo, S. 2017. "Electric vehicle battery parameter identification and SOC observability analysis: NiMH and Li-S case studies". *IET Power Electronics*, 10(11), 1289–1297.
- Gedon, D., Wahlström, N., Schön, T. B., Ljung, L. 2021. "Deep state space models for nonlinear system identification". *IFAC-PapersOnLine*, 54(7), 481–486.
- Gosavi, A. 2009. "Reinforcement learning: A tutorial survey and recent advances". *INFORMS Journal on Computing*, 21(2), 178–192.
- Gros, S., Zanon, M. 2020. "Data-driven economic NMPC using reinforcement learning". *IEEE Transactions on Automatic Control*, 65(2), 636–648.
- Gupta, S., Gupta, R., Padhee, S. 2020. "Stability and weighted sensitivity analysis of robust controller for heat exchanger". *Control Theory and Technology*, 18(1), 56–71.
- Haarnoja, T., Zhou, A., Levine, S., Abbeel, P. 2018. "*Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*". Tarihinde adresinden erişildi <https://arxiv.org/abs/1801.01290>
- Ho, B. L., Kalman, R. E. 1966. "Die Konstruktion von linearen Modeilen in der Darstellung durch Zustandsvariable aus den Beziehungen für Ein- und Ausgangsgrößen". at - Automatisierungstechnik, 14(1–12), 545–548.
- Kaelbling, L. P., Littman, M. L., Moore, A. W. 1996. "Reinforcement learning: A survey". *Journal of Artificial Intelligence Research*, 4, 237–285.
- Kaya, O., Abedinifar, M., Feldhaus, D., Diaz, F., Ertuğrul, Ş., Friedrich, B. 2023. "System

- identification and artificial intelligent (AI) modelling of the molten salt electrolysis process for prediction of the anode effect". *Computational Materials Science*, 230(September).
- Lagoudakis, M. G., Parr, R., Littman, M. L. 2002. "Least-squares methods in reinforcement learning for control". İçinde *Methods and Applications of Artificial Intelligence: Second Hellenic Conference on AI, SETN 2002 Thessaloniki, Greece, April 11--12, 2002 Proceedings 2* (ss. 249–260).
- Larimore, W. E. 1983. "Predictive inference, sufficiency, entropy and an asymptotic likelihood principle". *Biometrika*, 70(1), 175–181.
- Lazic, N., Lu, T., Boutilier, C., Ryu, M., Wong, E., Roy, B., Imwalle, G. 2018. "Data center cooling using model-predictive control". *Advances in Neural Information Processing Systems, 2018-Decem(NeurIPS)*, 3814–3823.
- Lewis, F. L., Vrabie, D., Syrmos, V. L. 2012. *"Optimal control"*. John Wiley & Sons.
- Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... Wierstra, D. 2016. "Continuous control with deep reinforcement learning". Tarihinde adresinden erişildi <https://arxiv.org/abs/1509.02971>
- Ljung, L. 1999. "System identification : theory for the user", 609.
- Ljung, L. 2008. "Perspectives on system identification". *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 17(1 PART 1).
- Mantar, S., Yılmaz, E. 2024. "Offline Identification of a Laboratory Incubator". *Applied Sciences*, 14(8), 3466.
- Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., Hutter, M. 2022. "Learning robust perceptive locomotion for quadrupedal robots in the wild". *Science Robotics*, 7(62).
- Miller, A. K., Ghionea, S., Vongsouvath, M., Davong, V., Mayxay, M., Somoskovi, A., ... Friend, M. 2019. "A robust incubator to improve access to microbiological culture in low resource environments". *Journal of Medical Devices, Transactions of the ASME*, 13(1), 1–7.
- Mnih, V., Badia, A. P., Mirza, L., Graves, A., Harley, T., Lillicrap, T. P., ... Kavukcuoglu, K. 2016. "Asynchronous methods for deep reinforcement learning". İçinde *33rd International Conference on Machine Learning, ICML 2016* (C. 4, ss. 2850–2869). Tarihinde adresinden erişildi <https://arxiv.org/abs/1602.01783>
- Moradimaryamnegari, H., Frego, M., Peer, A. 2022. "Model Predictive Control-Based Reinforcement Learning Using Expected Sarsa". *IEEE Access*, 10(July), 81177–81191.
- Mustafaraj, G., Chen, J., Lowry, G. 2010. "Development of room temperature and relative humidity linear parametric models for an open office using BMS data". *Energy and Buildings*, 42(3), 348–356.
- Nian, R., Liu, J., Huang, B. 2020. "A review On reinforcement learning: Introduction and applications in industrial process control". *Computers and Chemical Engineering*, 139, 106886.
- Oymak, S., Ozay, N. 2019. "Non-asymptotic identification of LTI systems from a single

- trajectory". Proceedings of the American Control Conference, 2019-July, 5655–5661.
- Paschke, F., Zaiczek, T., Robenack, K. 2019. "Identification of room temperature models using k-step PEM for Hammerstein systems". 2019 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019 - Proceedings, 320–325.
- Perrault, R., Clark, J. 2024. "Artificial Intelligence Index Report 2024".
- Pillonetto, G., Ljung, L. 2023. "Full Bayesian identification of linear dynamic systems using stable kernels". Proceedings of the National Academy of Sciences of the United States of America, 120(118), 1–11.
- Pillonetto, Gianluigi, Dinuzzo, F., Chen, T., De Nicolao, G., Ljung, L. 2014. "Kernel methods in system identification, machine learning and function estimation: A survey". Automatica, 50(3), 657–682.
- Pramuditha, K. A. S., Hapuarachchi, H. P., Nanayakkara, N. N., Senanayaka, P. R., De Silva, A. C. 2016. "Drawbacks of current IVF incubators and novel minimal embryo stress incubator design". 2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015 - Conference Proceedings, 100–105.
- Rawlings, J. B., Mayne, D. Q., Diehl, M., others. 2017. *Model predictive control: theory, computation, and design* (C. 2). Nob Hill Publishing Madison, WI.
- Rawlings, James B, Amrit, R. 2009. "Optimizing process economic performance using model predictive control". Nonlinear Model Predictive Control: Towards New Challenging Applications, 119–138.
- Richalet, J. 1976. "Algorithmic control of industrial processes". Proc. of the 4th IFAC Sympo. on Identification and System Parameter Estimation, 1119–1167.
- Richalet, J., Rault, A., Testud, J. L., Papon, J. 1978. "Model predictive heuristic control". Automatica (journal of IFAC), 14(5), 413–428.
- Ross, I. M. 2009. *A Primer on Pontryagin's Principle in Optimal Control*. Collegiate Publishers.
- Schulman, J., Levine, S., Moritz, P., Jordan, M., Abbeel, P. 2015. "Trust Region Policy Optimization". Tarihinde adresinden erişildi <https://arxiv.org/abs/1502.05477>
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. 2017. "Proximal Policy Optimization Algorithms". Tarihinde adresinden erişildi <https://arxiv.org/abs/1707.06347>
- Shen, H., Xu, M., Guez, A., Li, A., Ran, F. 2019. "An accurate sleep stages classification method based on state space model". IEEE Access, 7, 125268–125279.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... others. 2018. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". Science, 362(6419), 1140–1144.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M. 2014. "Deterministic policy gradient algorithms". İçinde *International conference on machine learning* (ss. 387–395).
- Söderström, T., Stoica, P. 1989. *System identification*. Prentice-Hall International.
- Sutton, R. S., Barto, A. G. 2017. *Reinforcement Learning: An Introduction*. Cambridge,

MA: The MIT Press.

- Van Otterlo, M., Wiering, M. 2012. "Reinforcement learning and markov decision processes". İçinde *Adaptation, Learning, and Optimization* (C. 12, ss. 3–42). Springer Verlag.
- Van Overschee, P., De Moor, B. 1994. "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems". *Automatica*, 30(1), 75–93.
- Van Overschee, P., De Moor, B. 1996. "Continuous-time frequency domain subspace system identification". *Signal Processing*, 52(2), 179–194.
- Wächter, A., Biegler, L. 2009. "IPOPT-an interior point OPTimizer".
- Wächter, A., Biegler, L. T. 2006. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical programming*, 106, 25–57.
- Wang, S., Chaovalitwongse, W., Babuska, R. 2012. "Machine learning algorithms in bipedal robot control". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(5), 728–743.
- Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*. Cambridge University.
- Zadeh, L. 1956. "On the identification problem". *IRE Transactions on Circuit Theory*, 3(4), 277–281.
- Zanon, M., Gros, S. 2020. "Safe reinforcement learning using robust MPC". *IEEE Transactions on Automatic Control*, 66(8), 3638–3652.