

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**PLC ile NMPC Uygulaması**

**YÜKSEK LİSANS TEZİ**

**Murat Erhan ÇİMEN**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**Tez Danışmanı: Doç.Dr. Yaprak YALÇIN**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**PLC ile NMPC Uygulaması**

**YÜKSEK LİSANS TEZİ**

**Murat Erhan ÇİMEN  
(504151123)**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**Tez Danışmanı: Doç.Dr. Yaprak YALÇIN**



İTÜ, Fen Bilimleri Enstitüsü'nün 504151123 numaralı Yüksek Lisans Öğrencisi Murat Erhan ÇİMEN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "PLC ile NMPC Uygulaması" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :** **Doç. Dr. Yaprak YALÇIN** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :** **Prof. Dr. Afife Leyla GÖREN SÜMER** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :** **Özgür Turay KAYMAKÇI** .....  
Yıldız Teknik Üniversitesi

**Teslim Tarihi** : **19 Ocak 2018**  
**Savunma Tarihi** : **20 Aralık 2017**



*Aileme,*





## ÖNSÖZ

Gerek bu tezin hazırlanması sırasında gerekse öğrencilik hayatımın her döneminde yardım ve katkıları ile bana her zaman destek olan değerli hocalarım Doç. Dr. Yaprak YALÇIN'a, Prof. Dr. Ali Fuat BOZ ve Yük. Müh. Mehmet Cüneyt HASBOLAT'a ve Yük. Müh. Ömer Faruk BOYRAZ'a, sonsuz teşekkürlerimi sunarım.

Her zaman maddi manevi destek vererek beni bugünlere getiren annem Ayfer ÇİMEN'E, babam Fadıl ÇİMEN'e, kardeşlerime ve yiğenime, her konuda manevi desteklerini esirgemeyen arkadaşlarıma sevgilerimi sunarım.

Kasım 2017

Murat Erhan ÇİMEN  
(Mühendis)



## İÇİNDEKİLER

### Sayfa

<b>ÖNSÖZ</b> .....	<b>vii</b>
<b>İÇİNDEKİLER</b> .....	<b>ix</b>
<b>KISALTMALAR</b> .....	<b>xi</b>
<b>SEMBOLLER</b> .....	<b>xiii</b>
<b>ÇİZELGE LİSTESİ</b> .....	<b>xv</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>xvii</b>
<b>ÖZET</b> .....	<b>xix</b>
<b>SUMMARY</b> .....	<b>xxi</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Modele Dayalı Öngörülü Kontrolün Temel Prensipleri .....	2
1.2 Literatür Araştırması .....	8
<b>2. DOĞRUSAL OLMAYAN MODELE DAYALI ÖNGÖRÜLÜ KONTROL..</b> <b>13</b>	
2.1 Doğrusal Olmayan Modele Dayalı Öngörülü Kontrolün Temel Bileşenleri.....	13
2.2 Modele dayalı öngörüler.....	14
2.2.1 Amaç fonksiyonu .....	14
2.2.2 Kontrol işaretinin seçilmesi.....	15
2.2.3 Öngörü ufku ve kontrol ufku.....	15
2.2.4 Referans yörüngesi ve işareti .....	17
2.2.5 Kısıtlar .....	17
2.3 Matematiksel Kabuller ve Problemin Formülizasyonu .....	18
2.4 Doğrusal Olmayan Model Öngörülü Kontrolün Avantaj ve Dezavantajları....	19
<b>3. NMPC'DE KULLANILAN TEKNİKLERİN FARKLILAŞMASI</b> .....	<b>23</b>
3.1 Sistemin Ayırıklaştırılması .....	24
3.1.1 Euler metodu .....	24
3.1.2 Runge Kutta metodu.....	24
3.1.2.1 Runge Kutta2 metodu.....	25
3.1.2.2 Runge Kutta4 metodu.....	25
3.2 İteratif Çözümleme .....	25
3.2.1 Anlık metod.....	26
3.2.2 Ardışıl metod.....	26
3.3 Çözümleme Tekniği .....	26
3.3.1 Ardışıl karesel programlama (SQP) .....	27
3.3.2 Genelleştirilmiş indirgenmiş gradyan metodu (GRG) .....	29
3.3.3 Penalty metodu .....	33
3.3.3.1 Interior penalty metodu .....	34
3.3.3.1.1 Primal-Dual interior metodu .....	36
3.3.3.1.2 Exterior penalty metodu .....	39
3.4 Doğrusal Olmayan Modele Dayalı Öngörülü Kontrolde Kararlılık.....	41

3.4.1 Sonsuz ufuklu doğrusal olmayan modele dayalı öngörülü kontrol .....	41
3.4.2 Sonlu ufuklu kesin kararlı doğrusal olmayan modele dayalı öngörülü kontrol	42
<b>4. FAN VE LEVHA SİSTEMİNİN NMPC İLE KONTROL UYGULANMASI</b>	<b>45</b>
4.1 Fan ve Levha Sistemi .....	45
4.2 NMPC'nin Matlab Uygulanması.....	50
4.2.1 Ölü zamanlı sistemin ölü zamansız nmpc ile kontrolü.....	50
4.2.2 Ölü zamanlı sistemin ölü zamanlı nmpc ile kontrolü.....	56
4.2.2.1 Bozucuların giderilmesi.....	64
4.3 NMPC'nin PLC Uygulaması.....	67
4.3.1 PLC yazılımının oluşturulması.....	68
4.3.2 Fan and levha sisteminin plc de nmpc ile simulasyon sonuçları.....	71
4.3.3 Fan and levha sisteminin plc de nmpc ile uygulama sonuçları .....	74
<b>5. SONUÇLAR VE ÖNERİLERİ .....</b>	<b>77</b>
<b>KAYNAKLAR.....</b>	<b>81</b>
<b>EKLER .....</b>	<b>85</b>
<b>ÖZGEÇMİŞ.....</b>	<b>89</b>

## KISALTMALAR

<b>NMPC</b>	: Doğrusal Olmayan Model Öngörülü Kontrol, Nonlinear Model Öngörülü Kontrol
<b>MPC</b>	: Model Öngörülü Kontrol
<b>BFGS</b>	: Broyden–Fletcher–Goldfarb–Shanno Algoritması
<b>PLC</b>	: Programlanabilir Mantık Denetleyicisi
<b>STL</b>	: Deyim Listesi Programlama Dili
<b>SCL</b>	: Yapısal Kontrol Dili
<b>MPHC</b>	: Model Öngörülü Sezgisel Kontrol
<b>MAC</b>	: Model Algoritmik Kontrol
<b>SQP</b>	: Ardışıl Karesel Programlama
<b>GRG</b>	: Genelleştirilmiş İndirgenmiş Gradyan Metodu
<b>KKT</b>	: Karush Kuhn Tucker Koşulları
<b>PEM</b>	: Proton exchange membrane



## SEMBOLLER

$A, B, C$	:Sistemin durum uzay parametreleri
$m, l_g, l, a, b$	:Sistemin parametreleri
$L, k, T, g, J_m$	:Sistemin parametreleri
$K, \tau, R_a, D_m$	:Sistemin parametreleri
$K_b, K_T, L_a$	:Sistem parametreleri
$e_a$	:Motor giriş gerilimi
$\phi$	:Amaç fonksiyonunun yeni formu
$\varphi, \dot{\varphi}, x, \dot{x}$	:Fan ve Levha model durumları
$\theta, \dot{\theta}, w, y, v$	:Fan ve Levha model durumları
$x_{ref}$	:Referans Değeri
$x^*$	:Optimal durum
$L$	:Lagrange fonksiyonu
$H$	:Hessien Matrisi
$u$	:Kontrol işareti
$y_{min}, y_{max}$	:Sistem çıkış kısıtları
$u_{min}, u_{max}$	:Kontrol işareti kısıtları
$i, k, j$	:Değişken isimleri
$t$	:Zaman
$T_c, T_p$	:Kontrol ve öngörü ufku
$E$	:Terminal ceza terimi
$J$	:Amaç Fonksiyonu
$X, U$	:Tanım Uzayı
$g_i$	:Eşitsizlik kısıt fonksiyonu
$h_i, l_i$	:Eşitsiz kısıt fonksiyonu
$G_i$	:Eşitsizlik kısıt fonksiyonu
$\mu$	:Primal-Dual parametresi
$s$	:slack değişken, Yön değeri
$r_k$	:Penalty ağırlığı
$\Psi$	:Kısıtların fonksiyonları
$z$	:Bağımsız kontrol değişkeni
$y$	:Bağımlı kontrol değişkeni
$e_a$	:Motora uygulanan gerilim
$NLP$	:Nonlinear programming değişkeni
$q$	:Sınır fonksiyonunun üstel değeri



## ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1: Sistem Parametreleri..... 49





## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1 : Model Öngörülü Kontrolün Genel Yapısı.....	2
Şekil 1.2 : Modele dayalı öngörülü kontrolün temel prensibi .....	3
Şekil 1.3 : Modele dayalı öngörülü kontrolün temel prensibi .....	5
Şekil 1.4 : NPMPC'nin Uygulanmasına İlişkin Zaman Grafiği .....	5
Şekil 1.5 : Lineer ve Nonlineer MPC .....	6
Şekil 1.6 : Bir Çevrimde İşlem Süreleri .....	8
Şekil 2.1 : Temel NMPC kontrol Döngüsü .....	13
Şekil 3.1 : NMPC'de farklı tekniklerin kullanılması.....	23
Şekil 3.2 : GRG'nin İzlediği yol.....	30
Şekil 3.3 : Interir Point Penalty Fonksiyonu Yaklaşımı .....	35
Şekil 3.4 : Primal-Dual Yol Takibi.....	37
Şekil 3.5 : Exterior Point Penalty Fonksiyonu Yaklaşımı .....	40
Şekil 4.1 : Fan ve Levha Sistemi .....	45
Şekil 4.2 : Sistemin Modeli .....	46
Şekil 4.3 : Levhanın Model .....	46
Şekil 4.4 : DC Motor Modeli.....	48
Şekil 4.5 : Ölü Zamansız Sistem ve Ölü Zamansız NMPC.....	50
Şekil 4.6 : Ölü Zamansız Sistem.....	51
Şekil 4.7 : Sistemin Ayrıklaştırılması.....	51
Şekil 4.8 : Amaç Fonksiyonu .....	52
Şekil 4.9 : Gradyan Hesabı .....	53
Şekil 4.10 : Hessien Matrisinin Oluşturulması .....	54
Şekil 4.11 : Adım Uzunluğunun Bulunması.....	55
Şekil 4.12 : NMPC ile Sistem Kontrolü .....	55
Şekil 4.13 : Ölü Zamansız NMPC ile Ölü Zamanlı Sistem Kontrolü .....	56
Şekil 4.14 : Ölü Zamanın Kaydırılması.....	56
Şekil 4.15 : Ölü Zamanlı Sistemin NMPC ile Kontrolü.....	57
Şekil 4.16 : NMPC Bloğunun İç Yapısı .....	58
Şekil 4.17 : Euler Model.....	59
Şekil 4.18 : Amaç Fonksiyonu_1 .....	60
Şekil 4.19 : Amaç Fonksiyonu_2 .....	61
Şekil 4.20 : Amaç Fonksiyonu_3 .....	61
Şekil 4.21 : NMPC Kontrollü Sistem Çıkışı.....	62
Şekil 4.22 : Amaç Fonsiyonunun Zamana Göre Değeri.....	64
Şekil 4.23 : Çıkış Bozuculu Sistem Yanıtı .....	65
Şekil 4.24 : Çıkış Bozuculu Sistem Yanıtı .....	65
Şekil 4.25 : Giriş Bozuculu Sistem Yanıtı.....	66
Şekil 4.26 : Giriş Bozuculu Sistem Yanıtı.....	66
Şekil 4.27 : Amaç fonksiyonunu Zamana Göre Değeri.....	67

Şekil 4.28 : Blok Ekleme .....	68
Şekil 4.29 : Euler Model ve NMPC Blokları.....	69
Şekil 4.30 : Euler Modelin SCL Kodu.....	70
Şekil 4.31 : Kontrol İşareti Üret Fonksiyonu.....	71
Şekil 4.32 : Sistem Cevabı.....	71
Şekil 4.33 : Sistem Cevabı ve Referans İşareti ve Bozucular.....	72
Şekil 4.34 : Kontrol İşareti Referans İşareti ve Bozucular .....	72
Şekil 4.35 : Sistem Cevabı ve Referans İşareti ve Bozucular.....	72
Şekil 4.36 : Kontrol İşareti Referans İşareti ve Bozucular .....	73
Şekil 4.37 : Sistem Cevabı ve Referans İşareti ve Bozucular.....	73
Şekil 4.38 : Kontrol İşareti Referans İşareti ve Bozucular .....	74
Şekil 4.39 : Sınırlandırılmış Kontrol İşareti.....	74
Şekil 4.40 : PLC ile NMPC Uygulaması Referans ve Sistem Cevabı.....	75
Şekil 4.41 : PLC ile NMPC Uygulaması Referans ve Kontrol İşareti.....	75



## PLC İLE NMPC UYGULAMASI

### ÖZET

Model öngörülü kontrol tekniği ilk olarak endüstride birbirlerinden bağımsız olarak Shell firmasındaki Cutler ve Ramaker ve Adersa firmasında çalışan Richalet ve arkadaşları tarafından 1978 yılında ortaya atılmıştır. Bu kontrol tekniği o zaman çok ilgi görmüştür ve şimdi başta petrol rafineleri olmak üzere endüstride 4500'ün üzerinde uygulaması bulunmaktadır. Model öngörülü kontrolün en büyük avantajı endüstride bulunan bir çok problemde olduğu gibi eşitlik veya eşitsizlik kısıtları bulunduğu kısıtları ihlal etmeden istenilen amaç fonksiyonunu minimize edecek şekilde kontrol işareti üretebilmektedir. Endüstriyel uygulamalarda kaçınılmaz olan ölü zamanlı sistemlerin kontrolünde de kullanılabilir. Ölü-zaman etkisini de öngörülerde göz önünde bulundurup ölüzamanlı sistemleri de başarı ile kontrol edebilmektedir Hatta bu tezde ölü zamanı duruma bağlı olarak değişen bir sistem için öngörü yapılmıştır.

Endüstride kullanılan sistemlerin çoğusu nonlineer sistemlerdir veya nonlinear kısıtlar içermektedir. Bu sistemleri lineer olan modeller üzerinden kontrol etmek istendiğinde beklenen performansın sağlanamamaktadır. Bunun yerine her çalışma koşulunda daha önceden tanımlanmış modeller kullanmak bir çözüm olabilir. Lakin bu durumda da süreksizlikler nedeni ile kararlılık problem oluşabilir, doğrudan Nonlineer Model Öngörülü Kontrol kullanımı daha akıllıca olacaktır. Çünkü Nonlinear Model Öngörülü Kontrol aslında sistemin durumlarına, giriş büyüklüklerine vb. nedenlerle değişen dinamiklerini yakalayabilmeyi sağlamaktadır.

NMPC'nin dezavantajlarından biri karmaşıklığı yüksek çevrim-içi sayısal optimizasyon işlemlerinin yapılmasını gerektirdiği için hızlı dinamiğe sahip sistemlerin kontrolü için uygulanabilir olmamasıdır.

Bu tezde NMPC'nin temel prensiplerinin ve temel bileşenlerinin neler olduğu araştırılıp öğrenilmiştir. NMPC'de kullanılabilecek olan farklı teknikler incelenmiştir. Bu tez çalışmasının amaçlarından biri temel prensipler aynı kalmak şartı ile NMPC'de yeni teknikler geliştirilebilmenin alt yapısını oluşturmaktır. Diğer amaç ise endüstriyel kontrol uygulamaları için giriş-çıkış arabirimleri ve işletim sistemleri özelleşmiş (MATLAB simülasyonlarının yaptığı güncel yüksek hızlı kişisel bilgisayarlara göre göre daha yavaş kod işlemektedirler ve türev alma gibi bazı matematik fonksiyonları bulunmamaktadır) ve örnekleme zamanına uygun biçimde gerçek zamanlı simülasyon ve gerçekleştirme yapmaya olanak tanıyan PLC'ye uygun NMPC algoritması oluşturmaktır.

NMPC'nin başarımını ve uygulanabilirliğini test edebilmek için sistemde eyleyici olan hava dinamiği ve sistemin kendi dinamiği gereği ağır nonlineerliği olan ve değişken ölü-zamanı bulunan Fan ve Levha düzeneğinin NMPC ile kontrolü problemi ele alınmıştır.

Fan ve Levha sisteminin Doğrusal Olmayan Model Öngörülü Kontrol kullanılarak PLC ile kontrolü için ilk olarak kısıtlarda göz önünde bulundurularak Matlab/Simulink'te benzetim çalışmaları yapılmıştır. Bu çalışmalarda Matlab'da bulunan hazır MPC fonksiyonları ve karesel programlama çözümleri kullanılmamış, PLC'de de gerçekleştirilmesine uygun olarak seçilen sayısal optimizasyon algoritması yazımı ve öngörülerin inşası gerçekleştirilmiştir. Sayısal optimizasyon algoritmasındaki Gradyan ve Hessian hesapları için matematik literatürde bulunan yaklaşımlar kullanılmıştır. Ardından bu kontrol tekniğinin PLC'de uygulaması TIA PORTAL V12 program geliştirme ortamında Nonlineer Model Öngörülü Kontrol Ladder ve SCL dili kullanılarak oluşturulmuş, başarımı test edilmiştir. Sistemde oluşan sürekli hal hatası ve aşım ilgili durumlarda ağırlıklar değiştirilerek çözülmüştür, yani sistemin yanıtının değişimi boyunca duruma göre ağırlıklar değiştirilmiştir.

Fan ve Levha sistemi üzerinde yapılan Matlab simülasyonlarında ve PLC'de yapılan simülasyon uygulamalarında giriş ve çıkış bozucularının etkisi de incelenmiştir. Bozucu etkiyi giderebilmek için sistem çıkışı ile öngörüler arasındaki fark göz önünde bulundurularak ölçüt fonksiyonunda uygun bir değişiklik yapılmıştır ve başarılı sonuçlar elde edilmiştir. Ayrıca S71200 ile gerçek zamanlı uygulama yapılmıştır.

## NMPC APPLICATION USING PLC

### SUMMARY

The model predictive control technique was first put forward as an industry. This technique was invented by different people who work study independently, Cutler-Ramaker who worked at Shell Oil and Richalet etc. worked at Adersa. After some symposium, which was organized by Shell Oil, about model predictive control was noticed by the this field. A consequence of this, model predictive control was began to study by academicians. That wasn't not just for academic level but also industrial applications. Thus, model predictive control has gained importance in the 80's and especially in the 90's. Until now, this technique has implemented to more than 4500 application and it has been going on.

The greatest advantage of the model predictive control was that it could generate a control signal to minimize the desired objective function without violating the constraints when there were equality or inequality constraints as in many problems in the industry. So, model predictive control technique allows the system to operate in the desired conditions at desired constraints and at desired constraints or in regions close to inequality constraints. Model predictive control could also be used to control time delay systems which are inevitable in industrial applications. Model predictive control can successfully control time delay systems by considering the time delay effect for prediction. Even in this thesis, a dead time is predicted for a system that changes depending on the situation.

Basically, Model Predictive Control solves optimal control problem. Its components makes it happen. Also these components specify the performance of controller. Those components are cost function, model of system, equality and inequality constraints of system and control and prediction horizon.

Cost function is also known as objective function or fitness objective function. It is used to obtain appropriate solution for problem. These cost functions may have changes from problems to problems. But generally main purpose is that predicted output of the system follows the reference signal with optimal input. That's why, difference of reference, predicted output signal and input changes multiply with coefficients accordig to their importance. The intention is that the predicted output follows reference signal along prediction horizon and both control and states stay in certain limits.

Describing a system is very improtant in model predictive control. Actually it is heart of this technique. Because prediction are made by using of model of system. If model doesn't present the system accurately and precisely then predictions of model never match with system answer. That is the point of distinction of lineer and nonlinear

model predictive control. Linear model predictive control uses linear models and nonlinear model predictive control uses nonlinear models or nonlinear constraints.

One of the main component is constraints. These could be equality and inequality and increase depend on problem. In the application almost every system has these constraints. These are used because constraints might be system equations or system limits such as actuator operating rates specified by manufacturer and actuators can't work out of these operating limits. So model predictive control guarantees that optimal solution remains within these limitations

The another component is control and prediction horizon. Control horizon is about how forward we can control. And prediction horizon is about how forward we can predict. Prediction horizon consist of control horizon. Because control horizon is smaller than prediction horizon. The main logic is that we can predict future but we can control less. So after control sequence is acquired the last control signal is applied rest of the predictions in one sampling time. If control prediction and control horizons are big that would need some conservative solutions but give satisfactory solution. If control prediction and control horizons are small that would cause aggressive and ineffective solution. Because of that this parameters should be selected system behaviors.

In this technique main idea is to minimize the the performance index in order to obtain control sequence. By doing that future of the system can be know at the current condions. But this can calculated by using of input's past value and past value of the output.

There are a lot of application about model predictive control. For example crude oil distillation, direct current motor, pH balance of wastewater, energy consumption ,electricity distribution, network control, flow control of fuel batteries, fuel cells control, PEM electrolyser control.

Most of the systems used in the industry include nonlinear systems or nonlinear constraints. If the system was desired to control through linear models, this could cause the expected value performance not to be achieved. Instead of that, using previously defined models in each working condition can be a solution. In this case, stability problems may arise due to discontinuities. Instead, it is wise to use nonlinear model to predictive control directly. Because the nonlinear model predictive control actually makes it possible to capture the dynamics of the system's states, input magnitudes, time-varying reasons, etc.

One of the disadvantages of the NMPC is that it is applicable to the control of systems with fast dynamics, as complexity requires high on-line numerical optimization operations

In this thesis, what basic presses and basic components of NMPC have beend studied and learned. Different techniques that can be used in NMPC have been studied. One of the aims of this thesis is to form the substructure of developing new techniques in NMPC with the basic principles being the same. The other goal is to provide real-time simulation and realization of the input-output interfaces and operating systems for industrial control applications (including some math functions like slower code operations and differentiation than current high-speed personal computers with MATLAB simulations) to create an NMPC algorithm suitable for the PLC

In order to test the performance and feasibility of the NMPC, the problem of control of the fan and plate arrangement with the NMPC, which has heavy nonlinearity and

variable time delay due to the air dynamism acting as the actuator in the system and its own dynamics, is discussed.

In order to control the Fan and Plate system with PLC using Nonlinear Model Predictive Control, simulation studies were first carried out in Matlab / Simulink considering limitations first. In these studies, ready MPC functions in Matlab and quadratic programming solvers were not used. A numerical optimization algorithm has been written in the PLC in accordance with the implementation, also construction of predictions has been realized. The approximations found in the mathematical literature for the Gradyan and Hessian calculations in the numerical optimization algorithm were used. Subsequently, this control technique was created using the Nonlinear Model Predictive Control ladder and SCL language in the TIA PORTAL V12 program development environment for application in the PLC and performance tested. The steady state error and overshoot that occurs in the system are solved by changing the weights in the related situations, i.e. the weights are changed according to the situation throughout the response of the system.

The effects of input and output disturbances in real-time applications in Matlab simulations and PLCs on the Fan and Plate system have also been examined. The fitness function for disturbance effect was arranged according to the difference between the system and the prediction and successful results have been achieved. Furthermore a real time application was implemented by using Siemens S71200 PLC on Fan and Plate System.

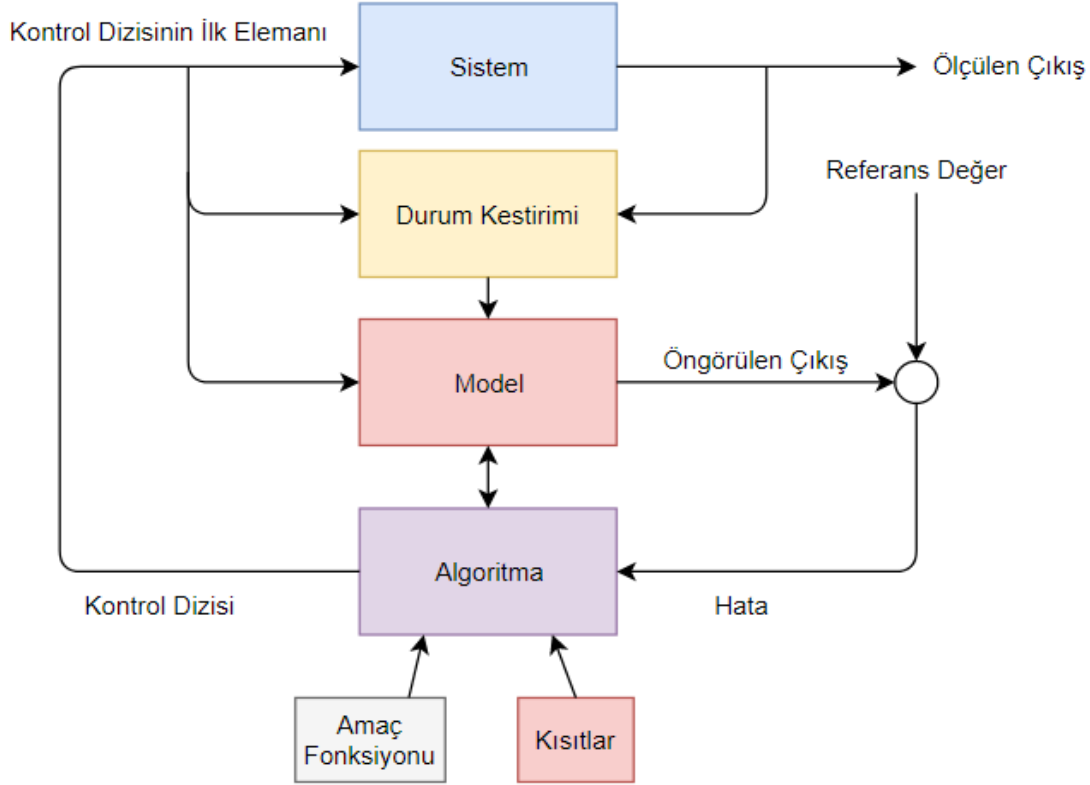


## 1. GİRİŞ

Modele Dayalı Öngörülü Kontrol 1970'lerden sonra çeşitli makaleler ve seminerlerde ortaya atılmıştır ve ilk ortaya çıktığı yıllardaki algoritmalar Model Öngörülü Sezgisel Kontrol (MPHC) veya Model Algoritmik Kontrol (MAC) olarak adlandırılmıştır. Teorik kısmının anlaşılmasından önce endüstride kullanılmaya başlanmış, sonrasında özellikle Shell firmasının yapmış olduğu seminerler ile Model Öngörülü Kontrole akademik ilgi artmıştır. Lakin bilgisayar sistemlerinin hızlı olmaması nedeni ile zaman sabiti yüksek sistemlere uygulanabilmiştir. Özellikle yapısının kolay olması ve kısıtlı-kısıtsız tek veya çok değişkenli sistemleri kontrol edebildiği için otomatik, kimyasal süreçlerin kontrolü, uçak uzay sistemleri, kağıt endüstrisi ve ağ sistemleri gibi pek çok alanda kullanılmaktadır (Eduardo F. Camacho & Bordons, 2007a; Kömürcü, 2009; Richalet, Rault, Testud, & Papon, 1976; Scokaert, Mayne, & Rawlings, 1999).

Model öngörülü kontrol sistemin modelini kullanarak sadece o anki sistem yanıtını değil gelecekteki sistem davranışını da göz önünde bulunduran bir kontrol yöntemidir. Her adımda MPC o anki sistem durumlarına ilişkin geribesleme bilgisini alıp, o andan sonraki sistem çıkışlarını bir öngörü ufku boyunca bir sistem modeli kullanarak hesaplayıp, öngörü ufku boyunca oluşan hataların bir fonksiyonu olan amaç ölçütünü enazlayacak şekilde açık çevrim optimizasyonla bir kontrol ufku boyunca gelecekteki optimal kontrol dizisini hesaplamaktadır. Bu tekniğe Kayan Ufuk (moving Horizon, receding) denilmesinin bir nedeni belirlenen kontrol dizisindeki sadece ilk kontrol değişkeninin sisteme uygunlanması ve ardından bu kontrol değişkeninin neden olduğu çıkış ölçülerek bir sonraki adımda ufuk bir kaydırılmış olarak yine amaç fonksiyonuna göre kontrol ufku bozunca yeni kontrol dizi elde edilmesidir. Ayrıca sistemin kısıtlarında meydana gelen değişiklikler gerçek zamanlı olarak güncellenip fiziksel sistemin kısıtlarına yaklaştırılabilmektedir. Bu nedenle klasik kontrol tekniklerine göre daha avantajlı olduğu görülmektedir. Bu tekniğin ana kalbi modeldir. Çünkü sadece model kullanılarak ölçülen değerlerle sistem durumlarının öngörü ufku boyunca kestirimi yapılmakta ve sisteme uygulanması gereken kontrol dizisi elde edilmektedir.

Buna ilave olarak sistem modelinin yanında sisteme paralel bir gözleci ile sistem parametreleri ve durumları kestirimi yapılabilir. Şekil 1.1’de Model Öngörülü Kontrol sisteminin genel yapısı verilmektedir (Seyab, 2006).



Şekil 1.1 : Model Öngörülü Kontrolün Genel Yapısı

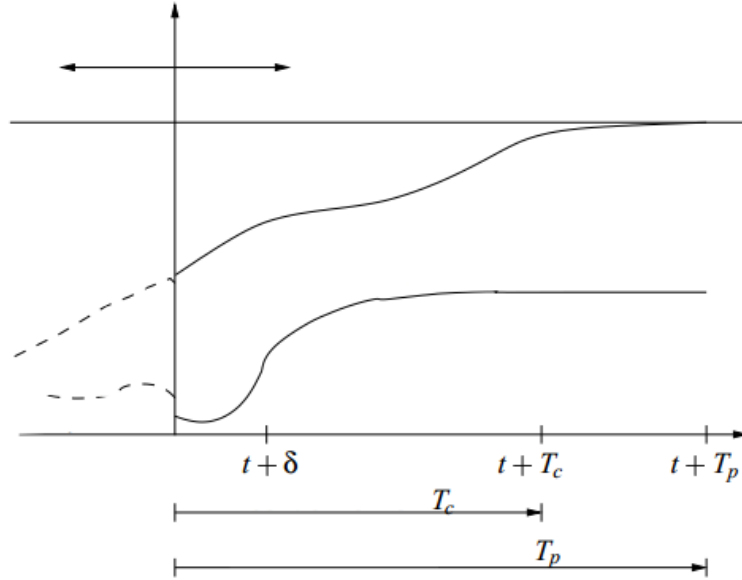
### 1.1 Modele Dayalı Öngörülü Kontrolün Temel Prensipleri

Model Öngörülü Kontrol; bir örnekleme zamanında sistemin matematiksel modelini kullanarak belirli bir kontrol ve öngörü ufku boyunca gelecekteki sistem durumlarını bir maliyet fonksiyonuna göre minimize edip sisteme uygulanması gereken optimal giriş sinyali olan kontrol ufku boyunca elde edilen kontrol dizisinin hesaplanmasıdır. Hesaplanan bu kontrol dizisinin ilk elemanı sisteme uygulanıp tekrardan bir sonraki örnekleme zamanı kontrol dizisi elde edilir ve ilk elemanı sisteme uygulanır.

Genel bir çerçeveden bakıldığında model öngörülü kontrol, sistem kısıtları, sistem dinamikler ve amaç fonksiyonu göz önünde bulundurularak sonlu bir ufuk boyunca

başlangıcı ve sonu bilinen optimal kontrol probleminin gerçek zamanlı bir şekilde çözülmesidir.

Gerçek zamanlı veya simülasyon yapıldığında kontrol işlemi yapılırken sistem durumları ayrık olarak alınmaktadır. Sistemin gelecek durumlarını etkileyecek olan kontrol dizisi bundan sonra amaç fonksiyonuna göre seçilebilmektedir. Genellikle öngörüğü belirli bir referans ile sistem çıkışı arasındaki hatanın farkı minimum yapılmaya çalışıldığı için bu problem bir referans takip problemine dönüşmektedir. Sonuç olarak sisteme uygulanacak kontrol işaretleri ile sistemin çıkışı istenilen bir seviyeye gelmektedir. Sistem çıkışı referans işaretinden uzaksa referansa yaklaştıracak şekilde kontrol işareti üretilirken, sistem çıkışı referans işaretine yakınsa sistem durumlarını koruyacak şekilde kontrol işareti üretilmelidir.



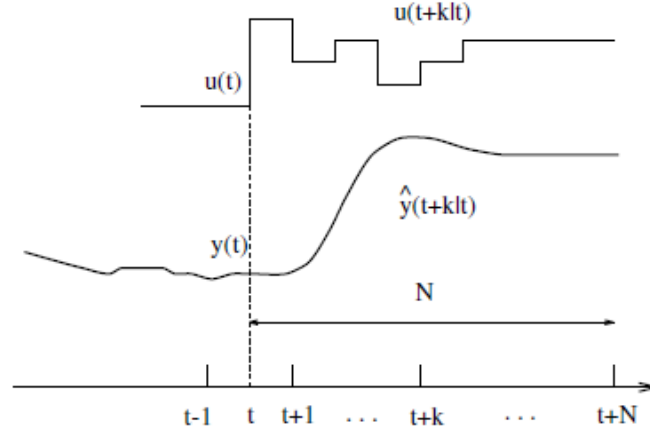
**Şekil 1.2 :** Modele dayalı öngörülü kontrolün temel prensibi

Genel olarak modele dayalı öngörülü kontrolün temel prensibi Şekil 1.2 'de verilmiştir. Sistem girişi ve sisteme uygulanan kontrol işareti verilmiştir. Denetleyici,  $t$  anında yapılan ölçümlere göre  $T_p$  süresince gelecekteki sistemin dinamik davranışını giriş işaretine göre ne olduğunu öngörür. Ayrıca bu algoritma belirlenmiş olan bir performans kriterini minimize etmek üzere, kontrol ufkü  $T_c$  boyunca kontrol dizisinin elemanları uygulanır ardından kalan öngörü ufkü  $T_p$  boyunca kontrol dizisinin son elemanı uygulanarak performans değeri belirlenir (Findeisen & Allgöwer, 2002).

Sistem üzerinde bir bozucu veya model hatası olmadığı durumda ve kontrol ufku sonsuz bir değer için çözümlenseydi o zaman elde edilen kontrol dizisi sisteme gerektiği t anından başlanarak sonsuz uzunluktaki kontrol dizisinin elemanları zamanı zamanı geldiğinde uygulanabilirdi. Lakin kontrol problemlerinde böyle bir durumun olmadığı açıktır. Sistem üzerinde dışarıdan muhakkak bir bozucu etki edecek ve bu doğrusal olmayan sistemin çalışmış olduğu durumları değiştirecek ve öngörü ufku sonsuz bir büyüklük alınmayacaktır. Bunun için bu öngörü yapılırken belirli bir öngörü zamanı üzerinden her t anında elde edilen çözüm sadece bir sonraki t+1 anına kadar geçerli sayılacak ve bu zaman zarfında kontrol dizisinin ilk elemanı sisteme uygulanacaktır. Bir sonraki zaman olan t+1 anında tekrardan durumlara ve amaç fonksiyonuna göre sisteme uygulanması gereken kontrol dizisi öngörü ufku boyunca elde edilmelidir (Rolf Findeisen & Frank Allgöwer, 2002).

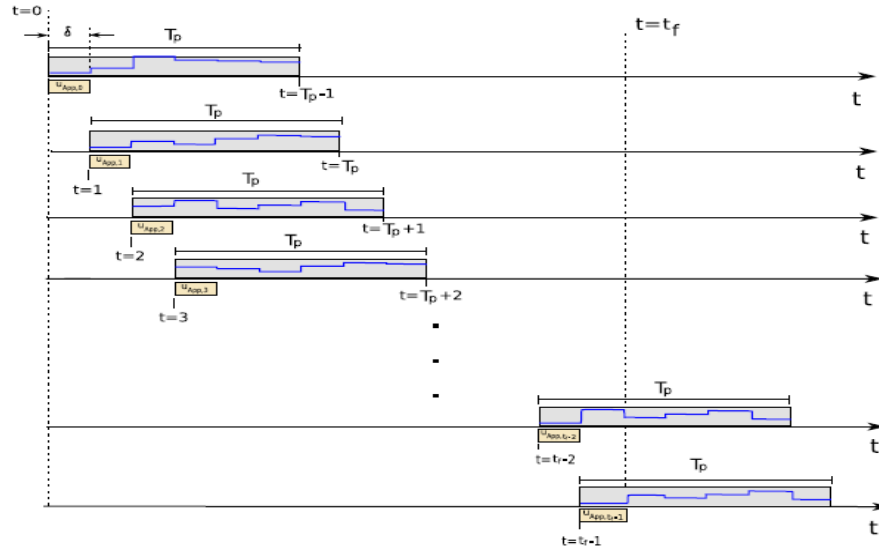
Lineer sistemler için geliştirilmiş basamak yanıtı yöntemi, darbe yanıtı yöntemi, transfer fonksiyonu, durum uzayı modeli gibi pek çok teknik bulunmaktadır. Genel olarak MPC yöntemlerinin temel adımları Camacho ve Bordons'da şu şekilde belirtilmiştir.

1. Sistemi ifade eden model kullanılarak o andaki sistem cevabı ve geçmiş giriş ve çıkış bilgileri kullanılarak öngörü ufku boyunca gelecek olan kontrol girişleri hesaplanır.
2. Gelecekteki sistem yanıtının bulunmasında kullanılan gelecekteki kontrol işareti, karesel fonksiyon olan başarımlı ölçütü minimize edilerek çoğunlukla öngörülen sistem çıkışı ile referansı arasındaki hatayı minimize ederek bulunur.
3. Kontrol işareti sisteme uygulandıktan sonra bir sonraki örnekleme zamanında adım 1 ve 2 tekrarlanarak yeni kontrol işareti dizisi hesaplanır. Bu kaydırma yöntemi ile gelecekte uygulanacak olan kontrol işareti hesaplanmaktadır. Lineer sistemler için model doğrusal olduğu zaman ölçüt karesel ve kısıtların olmadığı durumda çözüm için analitik bir cevap bulunabilmektedir. Aksi takdirde numerik yöntemler kullanılmalıdır.



**Şekil 1.3 :** Modele dayalı öngörülü kontrolün temel prensibi

Her örnekleme zamanında sistemden alınan veriler güncellenerek Şekil 1.3'deki gibi tekrardan kontrol dizisi hesaplanır (Yalçın, 2003). Yalnız üretilen kontrol dizisinin ilk işareti sisteme uygulanmaktadır. Bu işlemde Şekil 1.4'deki gibi her örnekleme zamanının yapılmaktadır (Kaya, 2007).



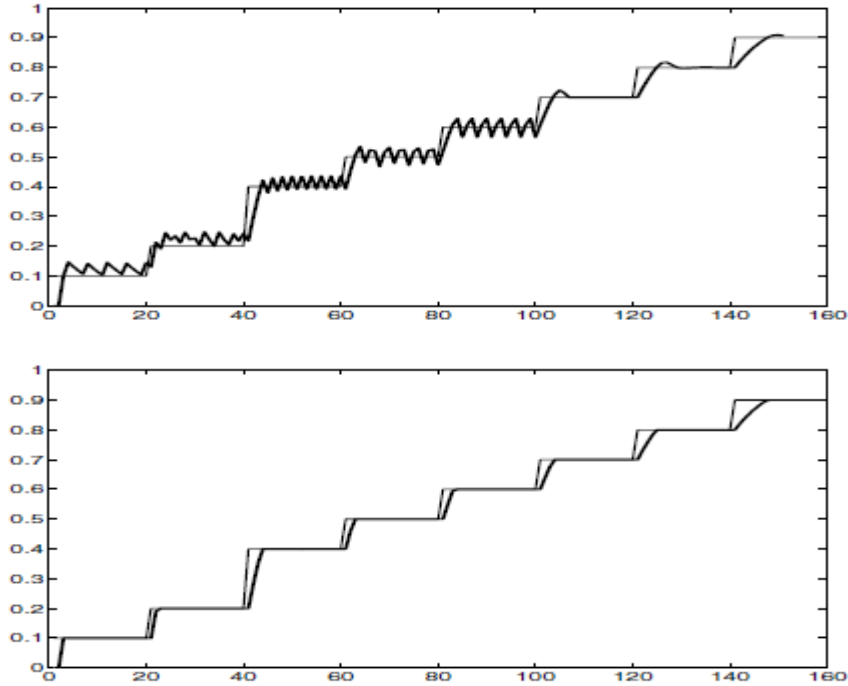
**Şekil 1.4 :** NPMC'nin Uygulanmasına İlişkin Zaman Grafiği

Modele dayalı öngörülü kontrol hareket eden ufuk veya kayan ufuk kontrolü olarak da bilinir. Bu tekniğin çekici hale gelmesinin nedeni geribesleme bilgisinin kullanılmasından kaynaklanmaktadır. Lineer modele dayalı öngörülü kontrol sistem dinamiğini (modelini) öngörülerini belirlemede kullanılmaya başlandığı için modele dayalı öngörülü kontrolün öncülüğünü yapmıştır (Findeisen & Allgöwer, 2002). Şimdilerde ise lineer modele dayalı öngörülü kontrol konusunda sistem tanıma, adatif

sistemler ile birlikte MPC kullanımı ve ilintili konularda kararlılık analizi ve ispatları gibi önemli konularda çalışmalar yürütülmüştür. (Mayne, Rawlings, Rao, & Scokaert, 2000).

Son zamanlara kadar endüstride model öngörülü kontrol uygulamaları genel olarak sistemler lineer oldukları varsayımı yapılarak kontrol edilmeye çalışılmıştır. Ancak gerçekte sistemlerin çoğunluğu doğrusal değildir yani nonlineerdir. Diğer taraftan sistemler herhangi bir denge noktasında lineerleştirildiğinde lineer MPC teknikleri kullanılabilir. Eşitlik (1.1)'de verilen örnek bir sistem için lineer ve doğrusal olmayan nonlinear model öngörülü kontrole yönelik sistem çıkışları ve referans cevapları Şekil 1.5'de verilmiştir. Ayrıca lineerleştirme yapılırken sistemin üzerindeki bozucular lineerleştirme noktasına etki edecektir. Bunun için bu lineerleştirme yapılırken bozucuların giderilmesi gerekmektedir. Bu nedenle lineer model öngörülü kontrolün genişletilmiş hali olan nonlinear model öngörülü kontrol nonlinear dünyasında yerini bulmuştur (P. E. F. Camacho, 2007; Rolf Findeisen & Frank Allgöwer, 2002).

$$y(t+1) = 0.9y(t) + u(t)^{\frac{1}{4}} \quad (1.1)$$

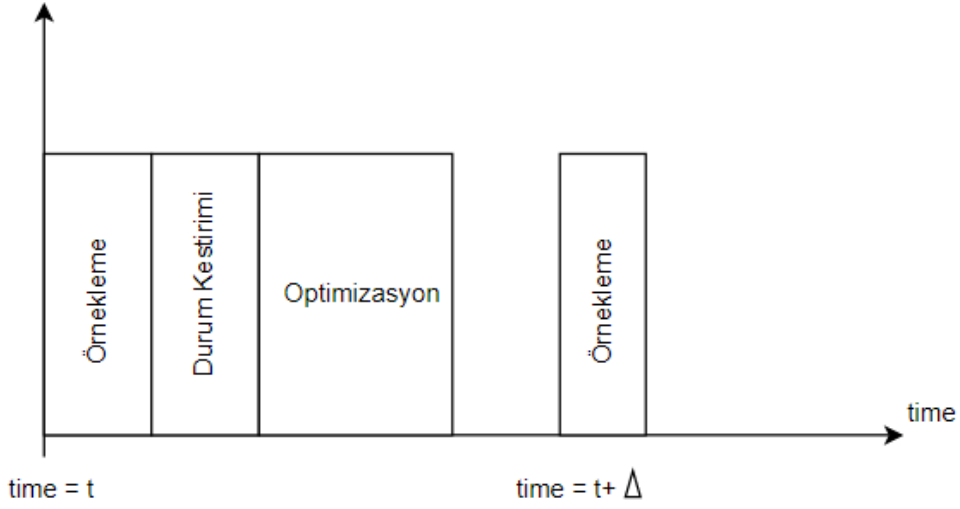


Şekil 1.5 : Lineer ve Nonlinear MPC

Doğrusal Olmayan Modele Öngörülü Kontrolün avantajı sistemin istenilen amaç fonksiyonuna göre optimize edebilmektedir. Fakat bu optimize işlemi zaman almaktadır. Ayrıca çözümlenen optimizasyon bazen suboptimal çözümler olabilmektedir. Bu teknikte;

- Gerçek zamanlı, hızlı ve güvenilir Nonlinear Optimizasyon Algoritmaları gerekmektedir. Çünkü kontrol ufkunun artması ile oluşan değişken sayısı ve sistemin yapısından dolayı oluşan sistem kısıtları çözülmesi zor bir problem olabilmektedir.
- Doğrusal olmayan nonlinear modelin geliştirilmesi gerekmektedir. Eğer sistem üzerinden veri toplanabiliyorsa yapay sinir ağı, bulanık model veya volterra serileri gibi farklı nonlinear model yapıları inşa edilebilmektedir. Bu modeller genellikle veriye bağımlı olmakta ve sistem dinamiklerini iyi yakalayabilmesi için sistemin tüm çalışma noktalarını içeren veri seti ile eğitilmeleri gerekmektedir. Diğer taraftan örnek olarak mekanik sistemlere yönelik klasik Newton yasaları veya elektriksel sistemlere yönelik gerilim yasaları gibi matematiksel yasalar kullanarak da model inşa edilebilmektedir.
- Doğrusal Olmayan Model öngörüsü gerektirmektedir.

Doğrusal Olmayan Model Öngörülü Kontrol de sistemden ölçümler yapıldıktan sonra durumların kestirilmesi ve ardından optimizasyon işlemi gelmektedir. Bundan sonra elde edilen kontrol işareti sisteme uygulanmaktadır. İki örnekleme zamanı içerisinde bu işlemlerin süreleri kıyas edildiğinde en fazla süreyi optimum kontrol dizisinin üretilmesini sağlayan optimizasyon işlemi almaktadır. Dolayısıyla kullanılacak optimizasyon tekniği için NMPC'nin kontrol ufku, öngörü ufku gibi parametreleride optimum olması gerekmektedir. Eğer sistem yüksek bir zaman sabitine sahipse yavaş hareket edeceği için optimizasyon işleminin bir çevrimde kısa veya uzun sürmesi sorun teşkil etmeyecektir. Lakin sistem düşük bir zaman sabitine sahipse hızlı bir dinamiği olacağı aşikardır. Bunun için hesaplama süreleri yaklaşık olarak Şekil 1.6'de verilmiştir. Sisteme farklı bir kontrol işareti uygulanmadan önce optimizasyon işlemi kontrol dizisini üretmiş olması gerekmektedir. Çünkü üretilen bu kontrol dizisinin ilk işareti kontrol işareti olarak uygulanacaktır. Ayrıca bu hızlı dinamiği göz önünde bulundurarak kontrol ufku ve öngörü ufku seçilmelidir. Çünkü seçilecek olan öngörü ufkunun uzunluğuna bağlı olarak sınırlı kararlılık garanti altına alınmaktadır.



**Şekil 1.6 : Bir Çevrimde İşlem Süreleri**

## 1.2 Literatür Araştırması

Öngörülü kontrol üzerindeki çalışmalar literatürde çok geniş bir yere sahiptir. Bu teknik tabii ilk olarak lineer sistemler üzerinden başlanarak onun üzerinde çeşitlenmiştir. Ardından numerik olarak çözümlemenin kolay olduğu bilgisayar sistemlerinin gelişmesi ile doğrusal olmayan sistemler üzerinde uygulanmaya başlamıştır.

Sanchez ve arkadaşları modele dayalı öngörülü kontrolün nasıl çalıştığını, prensiblerinin ne olduğunu anlatmışlar ve sistemin ayrık zamanlı modelini kullanarak kontrol tasarımını gerçekleştirmişlerdir.

Camacho ve Bordons modele dayalı öngörülü kontrole uygulanabilecek yöntemler ve algoritmalar üzerine çalışmışlar (Eduardo F. Camacho & Bordons, 2007a).

Clarke ve arkadaşları modele dayalı öngörülü kontrol içinde genelleştirilmiş öngörülü kontrol algoritması üzerine çalışmışlardır (D.W. Clarke, C. Mohtadi, & P.S. Tuffs, 1987a).

Richalet endüstriyel alanlardaki modele dayalı öngörülü kontrolü incelemiştir. Ayrıca kısıtlamaları bulunan ham petrol damıtma sistemi ile ilgili çalışma yapmıştır (Richalet, 1993).

Holkar ve Waghmare ise genelleştirilmiş öngörülü kontrol, dinamik matris kontrolü, model algoritmik kontrol, öngörülü fonksiyonel kontrol, genişletilmiş kendinden uyarımlı kontrol ve genişletilmiş uyarımlı kayan ufuk kontrol gibi pek çok konu üzerine çalışmışlardır. Ayrıca bu tekniklerin formülasyonunu yaparak bu teknikleri ve temel prensiplerini açıklamışlardır (Holkar & Waghmare, 2010).

Morari ve Garcia model öngörülü kontrolü durum uzayı üzerinden ifade etmişler ve kestirim yaparak basamak cevabını temel alan bir yöntem geliştirmişleridir (Morari, Lee, & Garcia, 1994).

Clarke ve Mohtadi genelleştirilmiş öngörülü kontrol ile lineer kuadratik programlama arasındaki ilişki üzerine çalışmışlardır. Dahası öngörü ve kontrol ufkunun model öngörülü kontrol üzerindeki etkilerini incelemişlerdir (D.W. Clarke, C. Mohtadi, & P.S Tuffs, 1987b).

Morari ve Lee modele dayalı öngörülü kontrolün kullanmış olduğu algoritmaların içerisindeki optimizasyon teknikleri üzerine çalışmışlar ve öngörülü kontrolün gelişimini incelemişler (Morari & Lee, 1999).

Morari ve arkadaşları kısıtlamalı model öngörülü kontrol tekniği için lineer matris eşitsizliği üzerine çalışmışlardır. Zaman gecikmesini, bozucu ve referans takibi gibi konuları tasarladıkları model öngörülü kontrol üzerinde incelemişlerdir (Kothare, Balakrishnan, & Morari, 1996).

Findeisen ve Allgöwer yapmış oldukları çalışmada NMPC'nin yapısını nasıl olduğu formülasyonunun nasıl yapıldığı hakkında bilgi verilmiştir. Aslında bu yapının sonlu ufuklu optimal kontrol problemi olduğu ve dolayısıyla bir optimal kontrol problemi olduğu ifade edilmektedir (Rolf Findeisen & Frank Allgöwer, 2002).

Foss ve arkadaşları sistemin farklı çalışma noktalarına göre lokal durum uzayı modelleri önermiştir. Bu modeller arasında ara değer uydurma yapılarak global bir model elde etmişlerdir. Bu model yapısı için MPC tasarımı gerçekleştirmiştir (Foss, Johansen, & Sørensen, 1995).

Mills ve arkadaşları yapmış oldukları çalışmada 61 değişkenli ve 200 üzerinde kısıtı bulunan ters sarkaç sistemini NMPC kullanarak gerçek zamanlı olarak kontrol

etmişler. Çözülmesi zor bir problem olan bu non-convex problemi SQP algoritması kullanarak çözülmüşlerdir (Mills, Wills, & Ninness, 2009).

Camacho ve Bordons yapmış oldukları review makalede Lineer MPC ve NMPC'nin özelliklerini ve farklarını vurgulamışlar. Özellikle ikisinin de hata fonksiyonunu minimum yapmaya çalıştığı fakat NMPC'nin daha kompleks bir yapısının olduğunu ifade etmişlerdir. Ayrıca NMPC için farklı tipte modellerin kullanılabilceğini ve bu modeller üzerinden kontrol dizisinin elde edilebileceğini ifade etmişlerdir. Sonuçta önerilen tekniklerde NMPC için suboptimal NMPC, Anlık yaklaşım, Kısa öngörü ufuklu kullanımı, kontrol ufkunun ayrıştırılması, geri beslemenin lineerleştirilmesi, yapay sinir ağları ve Volterra serileri kullanılarak MPC gibi konular hakkında bilgi vermiştir (Eduardo F. Camacho & Bordons, 2007b).

Quiryren ve ark. yapmış oldukları çalışmada SQP algoritmasını kullanarak NMPC ile her iterasyonda optimal kontrol problemi olarak çözdürmüşler. Multiple Shooting tekniğini kullanarak ayrıklaştırılmış olan sisteme ortalama 1 mikro saniyede kontrol işareti üretmişler. Dolayısıyla bu kadar hızlı cevap vermesi sayesinde sistem dinamiklerini çok iyi yakalayabildiği için gerçek zamanlı sistemlere uygulanabilmesine olanak tanımaktadır (Rien Quiryren, Milan Vukov, & Diehl, 2015).

Diehl ve arkadaşları ise optimal kontrol problemleri için numerik metodlardan olan “simultaneous” ve sequential ve bunun gibi farklı bir aile olan Newton tipi algoritmalarından sequential quadratic programming (SQP) ve interior point çalışmasından bahsedilmişlerdir. İkinci kısımdan ise bu çözüm tekniklerinin NMPC’de, Kayan Ufuk Yapısında ve gerçek zamanlı olarak uygulanmasındaki zorluklardan bahsedilmiştir (Moritz Diehl, Hans Joachim Ferreau, & Haverbeke, 2009).

Martinsen ve arkadaşları yapmış oldukları çalışmada Sürekli Karıştırılan Tank Reaktörünün NMPC ile kontrorlünü gerçekleştirmişler. Sonuçta farklı ayrıklaştırma metodlarının bulunduğunu, bununla birlikte farklı çözüm yapan optimizasyon tekniklerinin de olduğunu ifade etmişler. Bunlardan kısıtsız olarak SQP, penalty fonksiyonu olan rSQP ve kısıtların olduğu fakat penalty fonksiyonu olmayan sSQP algoritmasından bahsetmişler. Ayrıca “sequential” metodun “simultaneous” metoda

göre daha kolay olduğunu ifade etmişlerdir. Nedenin ise Hessian matrisinin oluşturulmasının daha kolay olması olduğunu belirtmişlerdir (Martinsen, Biegler, & Foss, 2002).

Kuo ve Melsheimer modele dayalı öngörülü kontrol için yapay sınır ağlarını model olarak belirlemişlerdi. Bu modeli atık suların pH değerini öngörüp nötralizasyonunu sağlamada kullanmışlardır (Kuo & Melsheimer, 1998).

Florin ve Lazar ise yaptıkları çalışmada yapay sinir ağlarını kullanarak modele dayalı öngörülü kontrol gerçekleştirmişler. Ayrıca bu yöntemi DC motor da kullanmışlardır (Caruntu & Lazar, 2011).

Yanhui ve arkadaşları ise bir eviricinin kontrolü için öngörülü kontrol algoritmasını kullanmışlardır. Amaçları evirici çıkışındaki gerilim dalgalanmalarını azaltarak harmoniklerin azaltılmasını sağlamaktır. Bunun için bu uygulamayı deneysel olarak gerçekleştirmişler ve dalgalanmanın azaldığını gözlemlemişler (Xie, Ghaemi, Sun, & Freudenberg, 2009).

Jianfend ve arkadaşları lokomotif fren sistemi için bulanık mantık model öngörülü kontrol yöntemini kullanmışlardır. Ayrıca bu yapıyı hem simülasyon hem de uygulama gerçekleştirmişlerdir (Liu, Huang, Liu, Yang, & Tong, 2008).

Koerber ve Rudibert modele dayalı öngörülü kontrolü rüzgar türbinlerinin kontrolünü gerçekleştirmişlerdir. Yaptıkları çalışmada dışarıdan gelen ani rüzgar değişimlerini, şebeke durumu gibi etkenleri durum kısıtlamaları içerisine almışlar. Bu problem için öngörülü kontrol kullanarak uygulama gerçekleştirmişlerdir (Koerber & King, 2013).

Yudong ve arkadaşları ise sıcaklık kontrolünü modele dayalı öngörülü kontrol kullanarak gerçekleştirmişlerdir. Model içerisinde hava şartları da göz önünde bulundurularak sisteme uygulanacak kontrol değerleri hesaplanmıştır. Yapılan bu çalışmada enerjinin optimum şekilde kullanımı sağlandığı için maliyette büyük oranda düşüş olmuş ve tasarruf sağlanmıştır (Ma et al., 2012).

Hredzak ve arkadaşları son zamanlarda popüler olan yakıt pili ve ultrakapasitör güç kaynağının kontrolü için model öngörülü kontrolör kullanmışlardır. Bu işlemi yaparken batarya ve ultrakapasitörün akımı ve gerilimlerinin belirlenen sınırlar içinde kalmasıdır (Hredzak, Jang, & Agelidis, 2013).

Lazar ve arkadaşları yapmış oldukları çalışmada suboptimal çözüm üreten ve robust ayrıca Lypunov tabanlı bir NMPC algoritmasını yükseltici ve alçaltıcı olarak çalışan bir DC-DC dönüştürücüye uygulamışlardır (Lazar, Heemels, Roset, Nijmeijer, & van den Bosch, 2008).

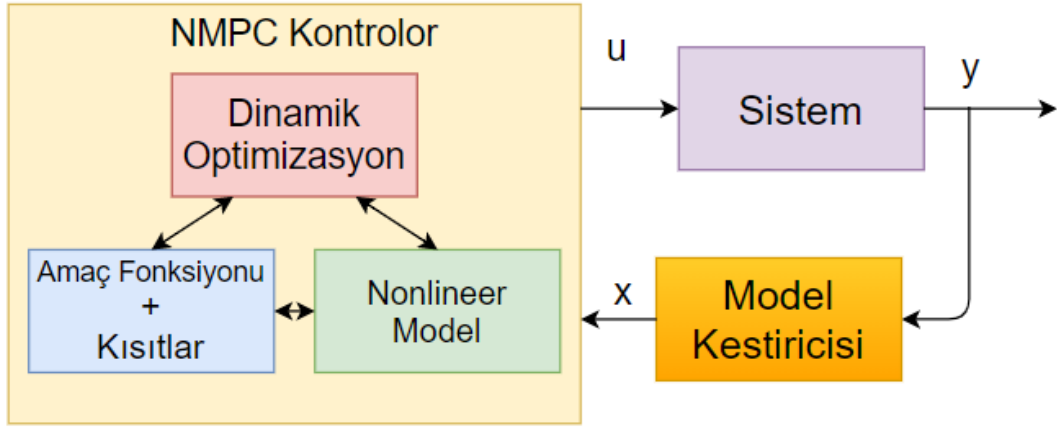
Krüger ve arkadaşları öngörülü kontrolü nonlinear olan ve pek çok katmanı olan ısıtıcı kazanlarına uygulamışlardır. Sıkı ve yumuşak kısıtlar içeren bu uygulamada amaç fonksiyonu ayrıca yakıt tüketimi ve ısıtma zamanını içermektedir. Bu problemi SQP algoritması ile çözümleyerek bu tekniğin önemini vurgulamışlardır (Kruger, 2004).



## 2 DOĞRUSAL OLMAYAN MODELE DAYALI ÖNGÖRÜLÜ KONTROL

Doğada bulunan bir çok sistem yapısı gereği nonlineerdir. Öte yandan endüstride bulunan bir çok sistem değişken dinamiğe ve değişken sınır koşullarına sahiptir. Böyle sistemler lineer model ile kontrol edilmek istendiğinde dinamikler sürekli değiştiği için sistem istenilen şekilde kontrol edilemeyebilmektedir. Ayrıca değişen sınırlarda sürekli sınıra yakın yerde sistemi çalıştırmakta ayrı bir sorun teşkil etmektedir. Bu nedenle böyle sistemler için doğrusal olmayan nonlineer modellerin kullanımı ortaya çıkmış ve doğrusal olmayan model öngörülü kontrol üzerine çalışmalar artmıştır (Rolf Findeisen & Frank Allgöwer, 2002).

### 2.1 Doğrusal Olmayan Modele Dayalı Öngörülü Kontrolün Temel Bileşeler



Şekil 2.1 Temel NMPC kontrol Döngüsü

NMPC'nin işleyişi lineer MPC'ye benzerdir Şekil 2.1'de verilmiştir (Rolf Findeisen & Frank Allgöwer, 2002). Buradaki aşamalar;

1. Sistem çıkışı ölçülür ve buradan elde edilen çıkışa göre durumların kestirimi veya ölçümü yapılır
2. Doğrusal Olmayan Model kullanılarak amaç fonksiyonunu minimum yapan kontrol dizisi dinamik optimizasyon ile elde edilir

3. Sistem çıkışının ve durumlarının yeni ölçümleri veya kestirimler gelene kadar kontrol dizisinin ilk elemanı sisteme uygulanır.
4. Bu işlemten sonra 1. Adıma tekrar geri dönülür.

NMPC için bu temel bileşenleri biraz ayrıntılandırılalım.

## 2.2 Modele dayalı öngörüler

Sistemin çalışmasını öngörebilmek için sistem davranışlarını belirleyebilmek için bir modele ihtiyaç bulunmaktadır. Bir sistem ne kadar iyi modelliyorsa o derece iyi kontrol edebilir. Çünkü yanlış bir öngörü bir amaç fonksiyonunun aldığı değerlerin yanlış bir şekilde üretilmesine neden olabilmektedir. Bu durumda sistem, klasik yöntemler ile modellenmek istendiğinde mekanik sistemler için Newton yasaları veya Langrangian'ın enerji prensipleri, elektriksel sistemler için kirşoff gerilim ve enerji denklemleri kullanılmaktadır. Farklı bir açıdan bakıldığında eğer sistemden veri toplanabiliyorsa analitik olmayan modeller de oluşturulup kullanılabilir. Dolayısıyla sistemin girişi ile çıkışı arasında ilişki iyi derecede biliniyorsa veya sistem girişi için her çalışma koşulunda veri toplandıysa sistemi yapay sinir ağları, bulanık modelleme veya Volterra serileri... v.b model yapıları ile sistem modellenebilir. Bununla birlikte bu yapıların devamlı güncellenen yani sürekli öğrenen bir yapı haline getirilerek adaptif modeller de oluşturmak mümkündür. Bu modeller ile sadece sistem girişini ve çıkışını kullanılarak elde edilen modellerdir.

Öngörü yapılırken model sadece sistemin çıkışın öngörmek için kullanılmaktadır. Kontrol açısından bakıldığında aslında sistemi en iyi temsil eden en basit model seçilmesi en uygun olan olmaktadır (Rossiter, 2004).

### 2.2.1 Amaç fonksiyonu

Öngörülü yapılırken kontrol dizisinin en optimum sonucu vererek sistemi istenilen referans işarete taşınmasını sağlayan bu yapının en önemli unsurlarından biridir. Amaç fonksiyonu sistem çıkışının referans işaretine yakınsamasını sağlayacak olan terime ek olarak bazı çevresel, sistemsel olarak değiştirilemeyen sistem kısıtlarını ve kontrol işaretine ilişkin kısıtları temsil eden terimleri de içeren bir fonksiyondur. Bu fonksiyon sistemin referansı takip etmesini sağlarken de istenen kısıtlar içinde kalmasını sağlamaktadır. Genel olarak amaç-maliyet fonksiyonunun ifadesi eşitlik (1.2) 'de verilmiştir (E.F. Camacho & Bordons, 2004).

$$\begin{aligned}
f(x, x_{bas}, u, u_{last}, T, T_c, T_p) = & Q(x_1(i) - ref)^2 + R(u_1(i) - u_{last})^2 + \\
& R(u_1(j) - u_1(j-1))^2 \\
& + Q(x_1(k) - ref)^2 \left\{ \begin{array}{l} i = 1, ..T_c \\ j = 2, ..T_c \\ k = T_c + 1, ..T_p \end{array} \right\}
\end{aligned} \tag{1.2}$$

NMPC’de maliyet fonksiyonunda bulunan  $T_c$  kontrol ufkunu ifade ederken  $T_p$  öngörü ufkunu temsil etmektedir.  $Q$  referans takibinin kazanç katsayısı iken  $R$  kontrol işaretinin değişim kazancıdır. Bu değer yüksek olduğunda kontrol işaretinin değişimi küçültülerek sistemin referans işarete yaklaşmasını zorlaştırmaktadır. Çok küçük olduğunda ise kontrol işareti çok hızlı değişmekte ve buda sistemi zorlayabilmektedir.

### 2.2.2 Kontrol işaretinin seçilmesi

NMPC’de amaç fonksiyonunu minimum yapacak şekilde optimal kontrol dizisi elde edilmektedir. Elde edilen kontrol dizisinin ilk elemanı sisteme uygulanacak kontrol işaretidir. Amaç fonksiyonu kısıt ve nonlinear denklemler içerdiğinden kontrol işaretinin elde edilebilmesi için sayısal çözüm yöntemleri kullanılmaktadır.

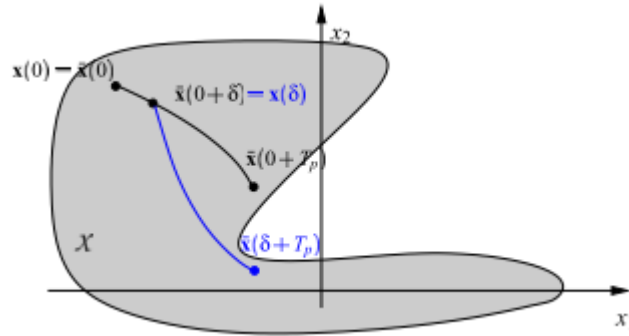
Pratik uygulamalara bakıldığında amaç fonksiyonu kapalı çevrimi az da olsa etkilemektedir. Amaç fonksiyonunun değeri gelecek kontrol değişkenlerine bağlıdır ve minimum yapılmaya çalışılan amaç fonksiyonunun değeri düşük olduğunda ise iyi bir kapalı döngü performansına elde edildiğini göstermektedir (Rossiter, 2004).

Sonuç olarak kontrol işareti amaç fonksiyonunun minimize olmasına bağlı olduğu gibi sistemin istenilen performansı sağlamasına, süreksizliklerin olmaması ve sistemin istenilen koşullarda çalışması için önemlidir.

### 2.2.3 Öngörü ufku ve kontrol ufku

Öngörü ve kontrol ufku aslında bir MPC kontrol sisteminin ne kadar uzağı kontrol edeceğini ve öngöreceğini belirlemektedir. Bu parametrelerin seçimi pek çok konu bakımından önem arz etmektedir. Aslında öngörü ufku denilen şey modelin  $t$  anından itibaren ne kadar uzaklığı gözlemleyebildiğiyle alakalıdır. Kontrol ufku da  $t$  anından itibaren gelecekte sisteme uygulanacak kontrol dizisinin değişken (eleman) sayına karşı düşmektedir. Eğer model çok iyi belirlendi ise dışarıdan bozucu etki etmiyorsa, sistem kısıtları farklı noktalara taşınmıyorsa, parametreler değişmiyorsa bu öngörüler

çok uzun seçilebilir. Bu işlem kararlı bir sonuç vermekle birlikte çok fazla işlem yükü getirektirmektedir. Diğer taraftan eğer çok kısa bir öngörü ufku seçildiyse bu seferde gelecekte meydana gelecek olan kısıtlardaki ve durumlardaki değişimleri göz ardı etmeye neden olacağından çok agrasif bir kontrol dizisi elde edilebilmektedir. Ampirik bir değer olarak öngörü ufku sistemin oturma zamanından büyük için bir değer seçilmelidir. Kontrol ufkunun öngörü ufkuna eşit seçilebileceği gibi kontrol ufkunu küçük değerde seçmek işlem yükü bakımından daha kolaylık sağlamaktadır. Bu seçimin nedeni ise nonlinear açık döngü optimal kontrol probleminin gerçek zamanlı olarak numerik çözümünü mümkün kılmasındandır. Dolayısıyla kısa öngörü ufuklarında minimize edilen amaç fonksiyonun değeri daha az değer alacağı aşikardır. Bu nedenle işlem yükü bakımından öngörülü kontrolde kısa ufuk kullanılması istenmektedir. Dahası herhangi bir uyumsuzluk, bozucu olmadığında bile model nonlinear olduğundan dolayı açık döngü yörüngesi gerçek sisteminkinden farklı olacaktır. Yani belirli gerçek sistemin kapalı döngü sistem durumları ve cevabı ile öngörülen açık döngü yanıtları farklı olacaktır (Allgöwer, Badgwell, Qin, Rawlings, & Wright, 1999). Bu durumda bile Şekil 2.2'deki gibi belirli kısıtlar arasında kapalı bölgede kalarak sistemin istenilen sınırdı tuttuğu var sayımı yapılır (Rolf Findeisen & Frank Allgöwer, 2002).



**Şekil 2.2** : Açık ve Kapalı döngü davranışları

Bu ise iki sonuç oluşturmaktadır. İlki sonsuz öngörü ufkuna göre minimize edilen amaç fonksiyonunun değeri elde edilemeyecektir. Yalnız sürekli olarak sonlu ufukta minimize edilen amaç fonksiyonu sonsuz ufuk için kısmen optimal bir çözüm sunacaktır. Ama bu iki sonuç elbetteki birbirlerinden farklı olacağı aşikardır. İkincisi ise öngörülen ile gerçek sistem durumları ve çıkışları birbirlerinden farklı oldukları

için kararlılığın bir garantisi yoktur (Bitmead, Gevers, & Wertz, 1993; Rolf Findeisen & Frank Allgöwer, 2002).

#### 2.2.4 Referans yörüngesi ve işareti

Öngörülü kontrolün diğer bir önemli unsur referans işaretidir. Bu işarete göre amaç fonksiyonu minimize edilerek sistem çıkışının bu referansa gelmesi sağlanmaktadır. Tabii referans işaretine yaklaşılrken ilk hızlı geçişler olabileceği gibi belirli bir fonksiyonu izleyen yörüngelerde oluşturulabilir. Buda sisteme uygulanacak olan kontrol dizisinin ona göre önceden değiştirilebilmesini sağlayarak sistem çıkışında daha yumuşak geçişlerin oluşmasını sağlayabilmektedir. Ayrıca ölü zamanlı sistemler için önem arz eden bu konu gelecekteki referans işareti biliniyorsa ölü zaman hesap edilerek sistemin istenilen referans işaretine gelmesi sağlanabilmektedir (E.F. Camacho & Bordons, 2004).

#### 2.2.5 Kısıtlar

Endüstriyel uygulamalarda hemen hemen tüm sistemlerin kısıtları vardır. Eyleyicilerin yapıları itibari ile belirli bir çalışma aralığı mevcuttur. Dolayısıyla kullanılan eyleyici bu kısıtların dışına çıkamaz bu aralıklarda çalışabilmektedir. Örnek olarak bir motor sürücüsünün belirli bir güç çıkışı verebilmesi, bir aracın belirli bir hız limitinin olması, elektrik motorunun belirli bir tork üretebilmesi, direncin belirli güce kadar çalışması, deponun belirli miktarda sıvı alabilmesi, analog haberleşmede sensörlerin 4-20mA arasında akım verilebilmesi, dijital haberleşmede haberleşmeyi sağlayan bant genişliği gibi sistem kısıtlarıdır (E.F. Camacho & Bordons, 2004).

Öngörülü kontrolde kısıtlar tasarım yapılırken belirlenir. Belirlenen kısıtlar optimize edilecek olan amaç fonksiyonu içerisine eklenir. Bu kısıtlar eşitlik (1.3)'de verilmiştir (E.F. Camacho & Bordons, 2004).

$$\begin{aligned} y_{min} &\leq y(t) \leq y_{max} \\ u_{min} &\leq u(t) \leq u_{max} \\ d_{min} &\leq u(t) - u(t-1) \leq d_{max} \end{aligned} \quad (1.3)$$

### 2.3 Matematiksel Kabuller ve Problemin Formülizasyonu

Sistem dinamikleri eşitlik (1.4)'deki gibi ifade edilebilir. Sistem ve durum kısıtları da eşitlik (1.5)'de verilmiştir. Burada verilen durumların ve girişlerin vektör olduğu kabulü yapılmıştır.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (1.4)$$

$$\begin{aligned} \mathbf{u}(t) &\in U \subseteq \mathfrak{R}^n \quad \forall t \geq 0 \\ \mathbf{x}(t) &\in X \subseteq \mathfrak{R}^m \quad \forall t \geq 0 \end{aligned} \quad (1.5)$$

Burada yapılan işlemde  $X$  ve  $U$ 'nun belirli aksiyomları sağladığı varsayımı yapılmıştır.

Varsayım 1: Giriş belirli bir aralıktadır ve buna bağlı olarak  $X$  de belirli bir aralıkta değişmektedir. Bu aslında en basit form olarak eşitlik (1.6)'de verilmiştir.

$$\begin{aligned} U &:= \{ \mathbf{u} \in \mathfrak{R}^m \mid \mathbf{u}_{\min} < \mathbf{u} < \mathbf{u}_{\max} \} \\ X &:= \{ \mathbf{x} \in \mathfrak{R}^m \mid \mathbf{x}_{\min} < \mathbf{x} < \mathbf{x}_{\max} \} \end{aligned} \quad (1.6)$$

Varsayım 2: Vektör alanı sürekli olmalı ve  $f(0,0)=0$  denge noktası vardır. Bununla birlikte lokal olarak  $x$ 'e bağlı olarak sistem Lipschitz sürekli olmalıdır.

Varsayım 3: Nonlineer sistem herhangi bir başlangıç koşulu için ilgili alanda sürekli tekil bir çözüme sahip olmalıdır. Ayrıca sistem parçalı sürekli olabilir. Ayrıca  $u$  verilen aralıkta sürekli ve tanımlı olmalıdır.

Minimize edilmek istenilen amaç fonksiyonu eşitlik (1.7)'de verilmiştir.

$$\min_{\bar{\mathbf{u}}(\cdot)} J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c, T_p) \quad (1.7)$$

Eşitlik (1.7)'deki form aslında meydana gelen hataların toplamını minimize etmeye çalışmaktadır. Bunun için bu form eşitlik (1.8)'daki gibi ifade edilebilir.

$$J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c, T_p) := \int_t^{t+T_p} F(\mathbf{x}(\tau), \bar{\mathbf{u}}(\tau)) d\tau \quad (1.8)$$

Minimize edilmek istenen sistemin kısıtlarında eşitlik (1.9)'de verilmiştir.

$$\begin{aligned}
\mathbf{x}(t) &= f(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot)) \\
\mathbf{u}(\tau) &\in U \quad \forall \tau \in [t, t+T_c] \\
\bar{\mathbf{u}}(\tau) &= \bar{\mathbf{u}}(\tau+T_c) \quad \forall \tau \in [t+T_c, t+T_p] \\
\bar{\mathbf{x}}(t) &\in X \quad \forall \tau \in [t, t+T_p] \\
T_c &\leq T_p
\end{aligned} \tag{1.9}$$

Eşitlik (1.8)'de verilen F amaç fonksiyonunun genel yapısı eşitlik (2.9) da verilmiştir. Eşitlik (2.9)'da  $\mathbf{x}_s$  sistemin durumlarının takip etmesi istenilen referans durum,  $\mathbf{u}_s$  sürekli halde durum hatasını sıfırlayan kontrol işareti değeridir. Öte yandan Q matrisi durumlara ilişkin hataların ağırlıklarını ifade ederken R ise kontrol dizisinin ağırlığını ifade etmektedir. Bu matrisler pozitif tanımlı ve simetriktir.

$$F(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_s)^T Q (\mathbf{x} - \mathbf{x}_s) + (\mathbf{u} - \mathbf{u}_s)^T R (\mathbf{u} - \mathbf{u}_s) \tag{1.10}$$

Bu formülüzasyonun istenilen çıkışı verebilmesi ve bir çözümü oluşturulabilmesi için yukarıda verilen varsayımların sağlanması gerekmektedir. Açık döngü optimal çözümler yapılırken sürekli olarak her örnekleme zamanı için yapılmaktadır.

#### 2.4 Doğrusal Olmayan Model Öngörülü Kontrolün Avantaj ve Dezavantajları

NMPC'nin üstünlükleri aşağıdaki gibi verilebilir (E.F. Camacho & Bordons, 2004; Clarke et al., 1987a):

- Nonlineer modellerin kullanılmasına olanak verir.
- NMPC, kısıtların rahat bir şekilde problemin içine katılmasına olanak sağlamaktadır.
- Sistemin değişen dinamiklerini öngörerek sisteme kontrol işareti uygulanmaktadır.
- Çok değişkenli sistemlere uygulanabilir.
- Ölü zamanı işleme katarak kontrol işareti üretebilmeyi sağlamaktadır.
- Ölü zamanı zamanla veya durumlara bağlı değişen sistemlere uygulanabilmektedir.
- Geleceğe ilişkin durumlar ve kontrol işareti bilinebilir.
- Bu teknik ile belirlenen performan kriteri online olarak minimum yapılmaktadır.

- Sistem zamanla ve durumlara bağı olarak deęişen nonlinear kısıtlara yakın kořullarda alıřtırılabilmektedir.
- Problem özömlenirken kısıtlardaki deęişimler online olarak problemin ierisine katılabilmektedir.
- Temel prensibler aynı kalması kaidesiyle gelişmeye ok aık bir alandır.

NMPC'nin avantajların yanında dezavantajlarıda ařaęıdaki gibi verilebilir (Bahtiyar, 2015; E.F. Camacho & Bordons, 2004):

- Modelin sistemi iyi bir şekilde ifade etmesi gerekmektedir.
- Öngörü yapabilmek için sistem durumlarının ölçülmesi veya kestiriminin yapılması gerekir.
- Numerik işlem yapıldığı için işlem yükü fazladır.
- Sistemin tüm dinamiklerini ieren bir örnekleme zamanının alınması gerekmektedir.
- Optimal problemin aık döngü problemi her örnekleme zamanında özölmektedir.
- İşlemlerin bir sonraki örnekleme zamanına kadar kontrol dizisinin oluşturulmuş olması gerekmektedir.

NMPC'nin MPC'ye üstünlükleri ařaęıdaki gibi verilebilir

- Endüstriyel olarak oęu doğrusal olmayan sistemlerdir.
- Deęişen dinamikleri hesaba katarak kontrol işaretinin bulunabilmesidir.
- Deęişen sınır kořullarında sınırlara yakın alıřabilmesidir.
- Linear sistemlerde farklı alıřma kořullarına göre deęişik modeller kullanılmaktadır.
- Sınırlı kararlılık ile istenilen kořullar rahatlıkla sağlanabilmektedir.
- Farklı model yapılarını kullanmaya daha müsaittir.
- Doğrusal olmayan bir sistemi linear model tabanlı öngörü ile kontrol etmek istendiğinde her zaman model ve sistem uyumsuzluğu olacaktır.
- NMPC sürekli deęişen dinamikler konusunda yada sürekli deęişen geçici rejim konusunda daha başarılıdır.

MPC'nin NMPC'ye üstünlükleri

- Kontrol kuralının daha kolay elde edilebilmesidir.

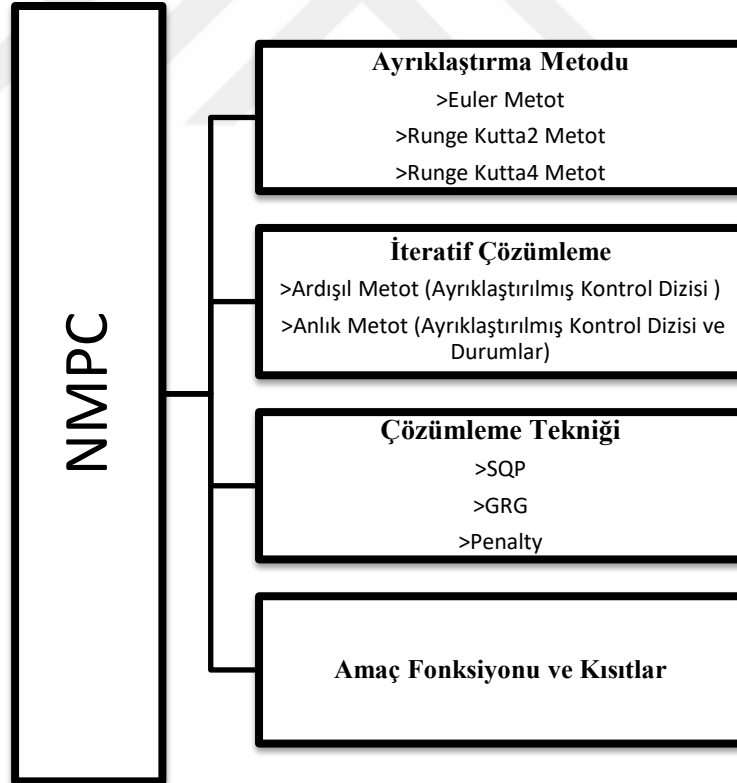
- Çözülmesi gereken büyük bir doğrusal olmayan optimizasyon probleminin olmamasıdır.
- Doğrusal olmayan modele göre daha rahat modelin oluşturulabilmesidir.
- İşlem yükü daha az olduğu için basit sistemlere çok rahatlıkla uygulanabilmesidir.
- Kısıtsız problemler için analitik bir çözümün olmasıdır.
- Kararlılığının daha kesin olmasıdır.
- İteratif bir yöntemden ziyade matrisin tersi ile direk olarak kontrol kuralının oluşturulmasıdır.
- NMPC’de kullanılan bazı modeller gerçek hayattan bağımsızdır. Bu nedenle giriş çıkış verilerine göre eğitilen bu modeller her çalışma koşulundan veri içermelidir.



### 3 NMPC'DE KULLANILAN TEKNİKLERİN FARKLILAŞMASI

Model tabanlı veya modele dayalı öngörülü kontrol, üzerine çalışılan sistemin gelecekteki davranışını tahmin edip bu tahmini kullanarak gelecekteki davranışı optimize etmek amacıyla uygun kontrol dizisini hesaplayan bir kontrolör sınıfını temsil etmektedir.

Bir problemin çözümünde farklı tekniklerin ve farklı yaklaşımların olması çözüm havuzunu arttırdığı gibi bazı dezavantajları da vardır. Kullanılacak teknik belirlenirken hassasiyet ve hesaplama yükü konusunda karşılaştırma ve değerlendirme yapmak gerekmektedir. Doğrusal Olmayan Modele Dayalı Öngörülü Kontrol yöntemlerinde değişiklik gösteren temel bileşenlerin bazıları Şekil 3.1'de şekildedir.



Şekil 3.1 : NMPC'de farklı tekniklerin kullanılması

### 3.1 Sistemin Ayrıklaştırılması

Sistemin durumlarına bağılı olarak bir sonraki durumunu tahmin edilebilmesi için sayısal işlemcilerde sistemin modelinin ayrıklaştırılması gerekmektedir. Bunun için Euler ve Runge Kutta Metodları kullanılmaktadır. Bu metodlar için eşitlik(3.1)'deki Taylor serisi kullanılmaktadır.

$$y(x+h) = y(x) + h\dot{y}(x) + \frac{h^2}{2!} \ddot{y}(x) + \dots + \frac{h^n}{n!} y^n(x) \quad (3.1)$$

#### 3.1.1 Euler metodu

Euler metodu eşitlik (3.1)'de ilk türev alınarak elde edilen bir ayrıklaştırma işlemi ile elde edilmektedir. Seride Eşitlik (3.2)'deki gibi bir yaklaşım yapılır.

$$y(x+T) \cong y(x) + T * \dot{y}(x) \quad (3.2)$$

Doğrusal olmayan sistem yapısından kaynaklı olarak ayrıklaştırma işleminde ara değerler kayıp olabilmektedir. Bunu önlemek için T örnekleme zamanından daha küçük seçilip örnekleme zamanı boyunca durumların ara değerleri hesaplanarak işlem yapılabilmektedir (Arifoğlu, 2005).

#### 3.1.2 Runge Kutta metodu

Euler Metodunun geliştirilmesinden sonra geliştirilen Runge Kutta yöntemi yine Taylor serisini kullanarak işlem yapmaktadır. Eşitlik (3.3)'deki ilk kabul yapılında açılımın türevleri elde edilecektir.

$$\begin{aligned} \dot{y}(x) &= f(x, y(x)) \\ \ddot{y}(x) &= \frac{\partial f(x, y(x))}{\partial x} + \frac{\partial f(x, y(x))}{\partial y} \frac{\partial y(x)}{\partial x} \\ \vdots &= \vdots \end{aligned} \quad (3.3)$$

Bu türevlerde eşitlik (3.1)'de yerine yazılarak eşitlik (3.4) gibi bir açılım elde edilir. Bu seri ise mertebelerine göre kullanıldığında Runge Kutta'nın mertebeleri olarak isimlendirilmektedir (Arifoğlu, 2005).

$$y(x+h) = y(x) + hf(x, y(x)) + \frac{h^2}{2!} \left[ \frac{\partial f}{\partial x}(x, y(x)) + \frac{\partial f}{\partial y}(x, y(x)) \frac{dy}{dx} \right] + \dots \quad (3.4)$$

### 3.1.2.1 Runge Kutta2 metodu

Taylor serisinin açılımı 2. Derece için düzenlendiğinde eşitlik (3.5) elde edilmektedir (Arifoğlu, 2005).

$$\begin{aligned} y(x+h) &= y(x) + \frac{(f_1 + f_2)}{2} \\ f_1 &= h * f(x, y(x)) \\ f_2 &= h * f(x+h, y(x) + h * f_1) \end{aligned} \quad (3.5)$$

### 3.1.2.2 Runge Kutta4 metodu

Taylor serisinin açılımı 4. Derece için düzenlendiğinde eşitlik (3.6) elde edilmektedir (Arifoğlu, 2005).

$$\begin{aligned} y(x+h) &= y(x) + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} \\ f_1 &= h * f(x, y(x)) \\ f_2 &= h * f(x+0.5h, y(x) + 0.5h * f_1) \\ f_3 &= h * f(x+0.5h, y(x) + 0.5h * f_2) \\ f_4 &= h * f(x+h, y(x) + 0.5h * f_3) \end{aligned} \quad (3.6)$$

Adım sayısı olarak verilen h parametresi yine örnekleme zamanı olabileceği gibi daha küçük bir değer seçilerek örnekleme zamanı boyunca bir döngü ile sürekli olarak yinelenabilir. Lakin işlem yükü çok arttığı için bu bazen sıkıntı oluşturabilmektedir.

Dolayısıyla Runge Kutta, Euler Metodu'nda olduğundan farklı olarak ara değerlerin bulunması söz konusudur. Bu sayede daha hassas bir çözüm yapabilmektedir.

## 3.2 İteratif Çözümleme

NMPC'de amaç ölçütünü minimize edecek optimal kontrol girişini belirleme optimal kontrol problemini çözecek çeşitli iteratif yöntemler bulunmaktadır. Daha doğrusu

algoritmaların çözüm yaparken her durumlar arası geçiş yaparken bu değerlere değişken ataması yaparak değişken sayısının artması ile çözüm yapılabilceği gibi durumları birbirine bağlayarak bu değişkenlerin ortadan kaldırılması ile de çözüm yapılabilir. Dolayısıyla problemin aynı anda çok boyutlu optimizasyon problemine dönüşmesi konusunda çok önem arz etmektedir.

### **3.2.1 Anlık metod**

Anlık metotta durumlar ve kontrol işareti ayrıklaştırılarak aynı anda durumlar ve kontrol dizisi bulunmaya çalışılmaktadır. Bu da çözüm esnasında her durum ve kontrol dizisi için kontrol ufku ve öngörü ufku boyunca oluşan değişkenin çözümlenmesini gerektirmektedir. Bu değişkenler ile üretilen amaç fonksiyon değerinin minimum olabilmesi içinde ayrıca amaç fonksiyonuna ilave olarak gelen kısıtların sağlanması gerekmektedir. Aksi takdirde durumların geçişleri arasında bir kopukluk meydana getirebilmektedir. Lakin bu problem çözümlendiğinde en optimum sonuç alınabilmektedir(Martinsen et al., 2002).

### **3.2.2 Ardışıl metod**

Ardışıl metotta ayrıklaştırılmış kontrol dizisi sisteme uygulanarak sistemin vermiş olduğu tepkiye bağlı olarak kontrol dizisi bulunmaya çalışılmaktadır. Bu sayede durumlara değer ataması yapılmadığı için durumlar arasında oluşan geçiş hataları oluşmamaktadır. Aksi takdirde bu geçiş hatalarında minimize yapılmaya çalışılacaktır. Sonuç olarak ayrıklaştırılmış olan kontrol dizisi elde edilmiş olmaktadır (Martinsen et al., 2002).

## **3.3 Çözümleme Tekniği**

Quadratik bir problemin çözümü için pek çok teknik bulunmaktadır. Genellikle bu problemlerin çözümünde en iyi çalışan türev tabanlı olan Newton tipi algoritmalarıdır. Lakin problemin yapısında eşitlikli ve eşitsizlik gibi kısıtlar olduğunda direk türev alınarak sistemin istenilen optimum noktası bulunamamaktadır. Bu nedenle bu Newton yapısı geliştirilerek oluşturulmuş algoritmalar vardır. 1960'larda ilk olarak Zoutendijk feasible yön yöntemini doğrusal olmayan problemler için önermiş. Ardından Rosen, Wolfe ve Powell gibi bilim insanları da bu problem üzerine çalışmışlardır. Ardından bu yöntem direk çözüm yöntemi olarak genelleştirilmiştir. Ardışıl Karesel Programlama ise 1963 yılında Wilson tarafından PhD teziyle önerilmiş

ve çok geniş anlamda kısıtlı doğrusal olmayan problemlerin çözümünde güçlü bir yöntem olmuştur. Farklı bir bakış açısı ile bu problemin çözümünü sağlayan “penalty” tekniğini 1968 yılında Fiacco ve McCormick klasik penalty yaklaşımını önermişlerdir.

### 3.3.1 Ardışıl karesel programlama (SQP)

Ardışıl Karesel Programlama son zamanlarda geliştirilen en iyi algoritmalarından biridir. Temel yapısı Newton Metoduna dayanmasına rağmen doğrusal olmayan nonlinear denklem kümesinin çözümü Kuhn-Tucker şartını sağlayarak Lagrangian’ın eşitlik Kısıtlı optimizasyon problemine dönüştürülmesi ile elde edilmektedir (Rao, 2009b).

$$\min f(X) \quad (3.7)$$

$$\text{Subject to } h_k(X) = 0 \quad k = 1, 2, 3, \dots, p$$

Eşitlik kısıtlı problemin Lagrangian fonksiyonu eşitlik (3.8)’deki gibi oluşturulmaktadır. Eşitsizlik kısıtlısı ise eşitlik kısıtı içeren problemin çözümü yapıldıktan sonra ilave yapılacaktır.

$$L(X, \lambda) = f(X) + \sum_{k=1}^p \lambda_k h_k(X) \quad (3.8)$$

Sağlanması gereken Kuhn-Tucker şartları eşitlik (3.9)’de verilmiştir.

$$\begin{aligned} \nabla L = 0 \quad \text{veya} \quad \nabla f + \sum_{k=1}^p \lambda_k \nabla h_k = 0 \quad \text{veya} \quad \nabla f + [A]\lambda^T = 0 \\ h_k(X) = 0 \quad k = 1, 2, \dots, p \end{aligned} \quad (3.9)$$

A matrisi  $n \times p$  boyutunda kısıtların türevlerinin içeren bir matristir. Bu denklem ve kısıtlar  $n+p$  adet bilinmeyen ( $x_i, i=1, \dots, n$  ve  $\lambda_k, k=1, \dots, p$ ) bir doğrusal olmayan denklem takımı oluşturmaktadır. Bu bilinmeyen değişkenler ise Newton Tekniği ile çözülebilmektedir.

İteratif bir yapı sağlayabilmek için eşitlik (3.10) dönüşümü yapıldığında geçici Y’ye bağımlı bir değişken haline olarak dönüştürülecektir.

$$\begin{aligned} F(Y) = 0 \\ F = \begin{Bmatrix} \nabla L \\ h \end{Bmatrix}_{(n+p) \times 1} \quad Y = \begin{Bmatrix} X \\ \lambda \end{Bmatrix}_{(n+p) \times 1} \quad 0 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}_{(n+p) \times 1} \end{aligned} \quad (3.10)$$

Her iterasyonda eşitlik (3.11)'daki gibi Newton'dan gelen değişim bir sonraki değere eklenerek güncelenecektir.

$$Y_{j+1} = Y_j + \Delta Y_j \quad (3.11)$$

Y'ye bağlı değişimin Newton'dan gelmesi ise eşitlik (3.12)'de olmaktadır.

$$[\nabla F]_j^T \Delta Y_j = -F(Y_j) \quad (3.12)$$

Bu yapı içerisinde geçici Y değişkeni yerine X cinsinden yazılırsa eşitlik (3.13) oluşmaktadır.

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H] & [0] \end{bmatrix}_j \begin{bmatrix} \Delta X \\ \Delta \lambda \end{bmatrix}_j = - \begin{bmatrix} \Delta L \\ h \end{bmatrix}_j \quad (3.13)$$

$$\begin{aligned} \Delta X_j &= X_{j+1} - X_j \\ \Delta \lambda_j &= \lambda_{j+1} - \lambda_j \end{aligned} \quad (3.14)$$

Eşitlik (3.13)'deki matris içerisinde bulunan  $\nabla^2 L$  aslında Hessian matrisini oluşturmaktadır. Ayrıca bu matris çarpımında bulunan  $\lambda$  parametresini değişim ifadeleri eşitlik (3.14)'de verilmiştir. Bu eşitliğin  $\lambda$  parametresini içeren 2. satırı eşitlik (3.13)'de yerine yazıldığında f(x) cinsinden eşitlik (3.15) elde edilmektedir.

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H]^T & [0] \end{bmatrix}_j \begin{bmatrix} \Delta X_j \\ \lambda_{j+1} \end{bmatrix} = - \begin{bmatrix} \Delta f_j \\ h_j \end{bmatrix} \quad (3.15)$$

Bu işlem ile aslında türeğe bağlı olarak elde edilen kvadratik bir problem olan eşitlik (3.16) minimize edilmeye çalışılmaktadır.

$$\begin{aligned} Q &= \nabla f^T \Delta X + \frac{1}{2} \Delta X [\Delta^2 L] \Delta X \\ h_k + \nabla h_k^T \Delta X &= 0 \quad k = 1, 2, \dots, p \quad \text{veya} \quad h + [H]^T \Delta X = 0 \end{aligned} \quad (3.16)$$

Sistemde kısıt olduğu takdirde eşitlik (3.16)'daki minimize edilmek istenen yapı eşitlik (3.17)'e dönüşmektedir.

$$\begin{aligned} Q &= \nabla f^T \Delta X + \frac{1}{2} \Delta X [\Delta^2 L] \Delta X \\ h_k + \nabla h_k^T \Delta X &= 0 \quad k = 1, 2, \dots, p \\ g_j + \nabla g_j^T \Delta X &\leq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (3.17)$$

Bu yapıda minimize edilecek olan  $\Delta X$  vektöründen ziyade arama yönü belirlenmeye çalışılacağı için  $S$  tercih edilmektedir. Sonuç olarak problem eşitlik (3.18)'e dönüşmektedir.

$$\begin{aligned} Q(S) &= \nabla f(X)^T S + \frac{1}{2} S^T [H] S \\ \bar{\beta} h_k(X) + \nabla h_k(X)^T S &= 0 \quad k=1,2,\dots,p \\ \beta g_j(X) + \nabla g_j(X)^T S &\leq 0 \quad j=1,2,\dots,m \end{aligned} \quad (3.18)$$

İşlemler eşitlik kısıtlı ile aynı olup tek farkı kontrol değişkenlerinin eşitlik (3.19)'daki gibi optimal bir adım uzunluğu ile her adımda güncellenmesidir. Ayrıca (3.18)'de denklemlerin içerisinde bulunan Beta parametresi bir sabit olup “feasible” bölge de lineerleştirmenin olduğundan emin olunması için yapılmış bir düzenlemedir. Bunun koşulları ise eşitlik (3.20)'de verilmiştir.

$$X_{j+1} = X_j + \alpha * S \quad (3.19)$$

$$\bar{\beta} \approx 0.9; \quad \beta_j = \begin{cases} 1 & g_j(X) \leq 0 \\ \bar{\beta} & g_j(X) > 0 \end{cases} \quad (3.20)$$

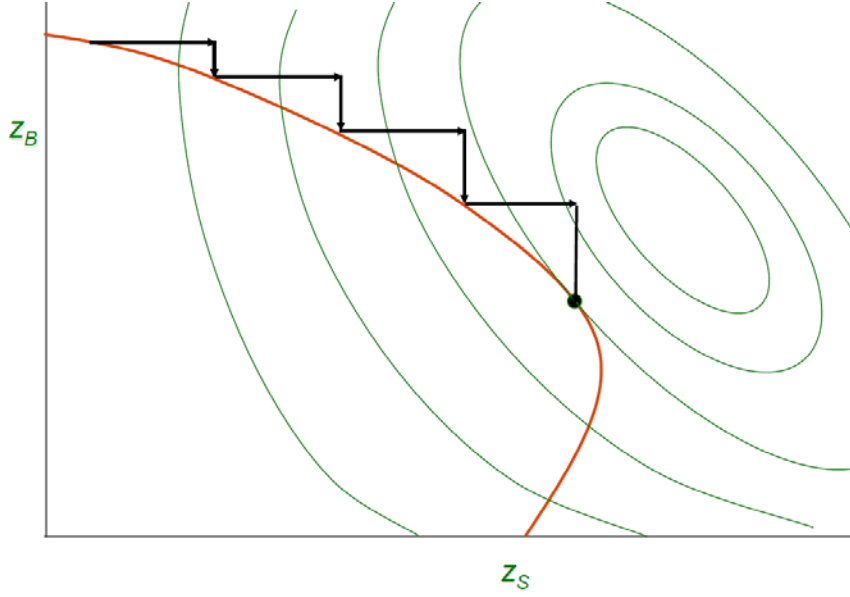
Sequential Quadratic Programming'de Hessien matrisi hesaplanmaktadır. Lakin ilk olarak hesaplamalara başlandığında birinci iterasyonda Hessien matrise birim matris ataması yapılmaktadır. Diğer iterasyonlarda da Hessien matrisinin güncellenmesi için modifiye Broyden–Fletcher–Goldfarb–Shanno (BFGS) algoritması kullanılır (Rao, 2009b).

### 3.3.2 Genelleştirilmiş indirgenmiş gradyan metodu (GRG)

Genelleştirilmiş yöntemde eşitlik kısıtlı olan Lagrangian yöntemine ilave olarak eşitsizlik kısıtı eklenmektedir. Bu eşitsizlik kısıtına slack denilen değişkenler eklenerek eşitsizlikler eşitlik haline dönüştürülerek çözüm yapılır. Bu teknikle problem çözümünün 2 yöntemi vardır. Birincisi kısıtları diğer değişkenler cinsinden yazarak değişken sayısı azaltılarak çözüm yapılabilir. İkinci yöntem ise kısıt sayısı kadar bağımlı değişken sayısı tanımlayarak bağımlı ve bağımsız değişkenler ile çözüm yapılıp optimum çözüm bulunmaya çalışılmaktadır (Rao, 2009b).

Geçen bölümde SQP'nin nasıl çalıştığı ve Kuhn-Tuhker şartını sağladığından bahsedilmişti. Temel olarak GRG eğer sistem değişkenleri “feasible” uzayın içinde

ise yapısı gereği çözümleri kısıtların üzerine taşımaya çalışmaktadır. Ardından bu sınırlar üzerinde ilerliyerek optimum nokta bulunmaya çalışılmaktadır. Şekil 3.2’de örnek tek kısıtlı bir problem için GRG’nin izlemiş olduğu yol verilmiştir.



**Şekil 3.2 :** GRG'nin İzlediği yol

Genel olarak minimize edilmek istenen bir fonksiyonun ve m tane eşitlik kısıtı bulunan bir problem eşitlik (3.21)'de verilmiştir.

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{s.t} \quad & g_i(x) - b_i = 0 \quad i = 1, 2, \dots, m \end{aligned} \quad (3.21)$$

Bu problem içerisinde n tane design parametresi ve m tane eşitlik kısıtı olarak barındırmaktadır. GRG geriği bu değişkenlere bağımlı ve bağımsız değişkenlere dönüştürülecektir. Kısıtlar m tane olduğu için m adet bağımlı değişkene (y) ve n-m adet bağımsız değişkene (z) dönüştürülecektedir. Bağımlı değişkenler kısıtların sağlanması için kullanılırken bağımsız değişkenler ise amaç fonksiyonunu minimize etmeye yönelecektir. Lakin her halükarda çözüm amaç fonksiyonu ve kısıtlar işin içine

katılarak iteratif olarak çözüm yapılmaktadır. Eşitlik (3.22)'de bağımlı ve bağımsız değişkenlere bağlı olarak amaç fonksiyonunun türevi alınmıştır.

$$\begin{aligned}\nabla f(z)^T &= \begin{bmatrix} \frac{\partial f(x)}{\partial z_1} & \frac{\partial f(x)}{\partial z_2} & \dots & \frac{\partial f(x)}{\partial z_{n-m}} \end{bmatrix} \\ \nabla f(y)^T &= \begin{bmatrix} \frac{\partial f(x)}{\partial y_1} & \frac{\partial f(x)}{\partial y_2} & \dots & \frac{\partial f(x)}{\partial y_{n-m}} \end{bmatrix}\end{aligned}\quad (3.22)$$

Bu işlemlerden sonra kısıtlarında bağımsız ve bağımlı değişkenlere bağlı olarak türevleri eşitlik (3.23)'de verilmiştir.

$$\begin{aligned}\frac{\partial \psi}{\partial z} &= \begin{bmatrix} \frac{\partial g_1}{\partial z_1} & \frac{\partial g_1}{\partial z_2} & \dots & \frac{\partial g_1}{\partial z_{n-m}} \\ \frac{\partial g_m}{\partial z_1} & \frac{\partial g_m}{\partial z_2} & \dots & \frac{\partial g_m}{\partial z_{n-m}} \end{bmatrix} \\ \frac{\partial \psi}{\partial y} &= \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & \frac{\partial g_1}{\partial y_2} & \dots & \frac{\partial g_1}{\partial y_{n-m}} \\ \frac{\partial g_m}{\partial y_1} & \frac{\partial g_m}{\partial y_2} & \dots & \frac{\partial g_m}{\partial y_{n-m}} \end{bmatrix}\end{aligned}\quad (3.23)$$

Amaç fonksiyonundaki ve kısıtlardaki değişimler eşitlik (3.24)'deki gibi yazılabilir.

$$\begin{aligned}df &= f(z)^T dz + f(y)^T dy \\ d\psi &= \frac{\partial \psi}{\partial z} dz + \frac{\partial \psi}{\partial y} dy = 0\end{aligned}\quad (3.24)$$

$\frac{\partial \psi}{\partial y}$  sonucu kare bir matristir. Bu nedenle bağımlı değişkenin değişimi bulunmak istendiğinde eşitlik (3.25)'den bulunabilir.

$$dy = -\left(\frac{\partial \psi}{\partial y}\right)^{-1} \frac{\partial \psi}{\partial z} dz \quad (3.25)$$

Eşitlik (3.25) ile elde edilen bağımlı değişkenin türevi eşitlik (3.24)'de yerine yazılırsa eşitlik (3.26) elde edilir.

$$df = \nabla f(z)^T dz - \nabla f(y)^T \left(\frac{\partial \psi}{\partial y}\right)^{-1} \frac{\partial \psi}{\partial z} dz \quad (3.26)$$

Eşitlik (3.26) farklı bir ifade ile ifade edilmek istendiğinde eşitlik (3.27) elde edilir. Bu ifade aslında azaltılmış gradyent olup aslında kısıtlara teget bir yön bilgisi vermektedir.

$$\nabla f_R^T = \nabla f(z)^T - \nabla f(y)^T \left( \frac{\partial \psi}{\partial y} \right)^{-1} \frac{\partial \psi}{\partial z} \quad (3.27)$$

$$s = \nabla f_R^T$$

Buradan elde edilen yön bilgisi normalize edilir. Ardından eşitlik (3.28) ile bağımsız değişkenlerin yeni değerleri hesaplanır. Bu işlemde “line search” yapılabileceği gibi sabit bir değerde alınabilir. “Line search” ile elde edilen adım uzunluğu ile elde edilen sonuçlar daha çabuk çözüm verebilir, lakin işlem yükünü arttırabilir.

$$z^{yeni} = z^{eski} + \alpha s \quad (3.28)$$

Bağımsız değişkenler kullanılarak bağımlı değişkenlerdeki değişim eşitlik (3.29) ile hesaplanır.

$$\Delta y = - \left( \frac{\partial \psi}{\partial y} \right)^{-1} \frac{\partial \psi}{\partial z} \Delta z \quad (3.29)$$

Eşitlik (3.29)’dan elde edilen bağımlı değişkendeki değişimde eşitlik (3.30)’deki gibi bağımlı değişkenlere eklenir.

$$y^{yeni} = y^{eski} + \Delta y \quad (3.30)$$

Sistemin amaç fonksiyonu eşitlik (3.31)’deki gibi olduğu zaman sistem içerisinde eşitsizlik kısıtı da içermektedir.

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) - b_i = 0 \quad i = 1, 2, \dots, m \\ & g_j(x) - b_j \leq 0 \quad j = 1, 2, \dots, k \end{aligned} \quad (3.31)$$

Bu tip bir eşitlik (3.32)’deki gibi eşitsizlik kısıtına slack değişken denilen bir değişken eklenerek eşitsizlik eşitlik haline dönüştürülür. Eğer feasible bölgede ise o zaman slack değişkenin değeri sıfırdan büyük olacaktır. Eğer slack değişken sıfır ise o zaman eşitsizlik kısıtı sınır üzerinde bir değere sahiptir.

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) - b_i = 0 \quad i = 1, 2, \dots, m \\ & g_j(x) - b_j + s_j^2 = 0 \quad j = 1, 2, \dots, k \end{aligned} \quad (3.32)$$

GRG için yapılması gerekenler;

1. Amaç fonksiyonunu ve kısıtları değerlendirilir.
2. Eğer eşitsizlik varsa slack değişken ataması yapılır.
3. Bağımlı ve bağımsız değişkenler belirlenir. Ancak her kısıt için bir bağımlı değişken ataması yapılması gerekmektedir.
4. Eşitlik (3.27) hesaplanır.
5. “Line search“ ile veya sabit bir katsayı ile bağımlı ve bağımsız değişkenlerin değişimleri hesaplanır
6. Tüm kısıtlar kontrol edilir. Herhangi bir sınır aşımı olduğu zaman  $\Delta y = -\frac{\partial \psi}{\partial y}(g - b)$  kullanılarak bağımlı değişkenlerdeki değişim tekrar hesaplanarak kısıtlar sağlanmaya zorlanır.
7. Bu iterasyon aşağıdaki durumlarda bırakılabilir
  - a. Minimum arama yönü elde edildiğinde
  - b. Bağımlı değişkenler sınıra geldiğinde
  - c. Eşitsizlikler sınırı aştığında
  - d. Yakınsamanın aksine uzaklaşma olduğunda
8. Eğer 7. Adım sağlanmıyorsa 1. Adıma tekrar geri dönülür.

### 3.3.3 Penalty metodu

Bu teknikle aslında kısıtlı optimizasyon problemi kısıtsız optimizasyon problemine dönüştürülerek ardışıl olarak minimizasyon işlemi yapmaktadır. Eşitlik (3.33) minimize edilmek istendiğinde bu problem eşitlik (3.34)'ye dönüştürülmektedir (Rao, 2009b).

$$\begin{aligned} \min \quad & f(x) \\ & g_j(x) \leq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (3.33)$$

$$\phi(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m G_j [g_j(\mathbf{x})] \quad (3.34)$$

Bu formadaki  $G_j$  bir kısıtın bir fonksiyonu ve  $r_k$  ise penalty denilen pozitif sabit bir katsayıdır. Orjinalliği ise eğer fonksiyonun yapısına göre buradaki “penalty” parametresi artan veya azalan penalty değerleri için tekrar tekrar çözümlendiğinde eşitlik (3.33)’deki kısıt sağlanabilmektedir. Bu nedenle bu penalty teriminden dolayı Ardışıl Kısıtsız Minimizasyon Tekniği (SMUTs) denilmektedir.

Problem içerisine eşitlik kısıtıda eklendiğinde problem eşitlik (3.35)’e dönüşmektedir. Dolayısıyla bu sefer yeni minimize edilmek istenen form eşitlik (3.36)’e dönüşmektedir.

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ & g_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \dots, m \\ & l_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \dots, n \end{aligned} \quad (3.35)$$

$$\phi(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m G_j [g_j(\mathbf{x})] + H(r_k) \sum_{j=1}^n l_j(\mathbf{x})^2 \quad (3.36)$$

İteratif olarak çalışan bu sistemlerin durdurulması için bir kaç kriter tipi belirlenebilir. Bunlardan bazıları şöyle sıralanabilir;

1. Optimum  $f(\mathbf{x})$  fonksiyonunu değişiminin  $\varepsilon$ ’dan küçük olmasıdır.

$$|f(\mathbf{x}_k^*) - f(\mathbf{x}_{k-1}^*)| / f(\mathbf{x}_{k-1}^*) < \varepsilon$$

2. Optimum noktalar arasındaki farkın  $\varepsilon$ ’dan küçük olması

$$|\mathbf{x}_k^* - \mathbf{x}_{k-1}^*| < \varepsilon$$

3. Belirli bir iterasyon sayısına erişildiğinden durdurma

$$k < k_{\max}$$

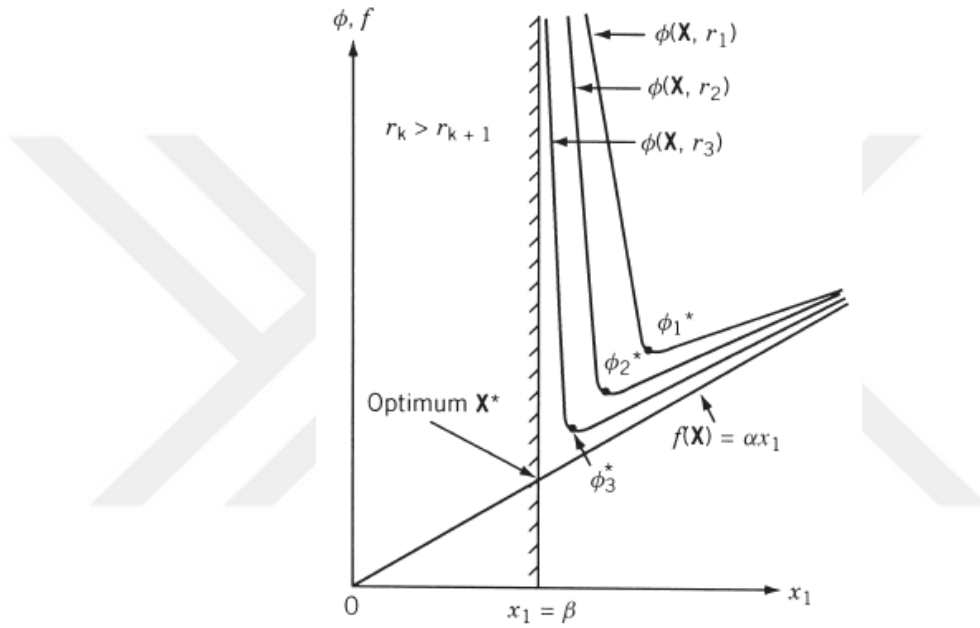
### 3.3.3.1 Interior penalty metodu

Bu teknikte amaç fonksiyonuna eklenen “penalty” terimi ile minimize edilmek istenen fonksiyon oluşturulmaktadır. Burada seçilecek ceza fonksiyonu muhtevası gereği bir takım özelliği sağlamalıdır. Örnek olarak eğer sistemin başlangıç koşulları kısıtlardan uzaksa o zaman bu fonksiyon çok küçük bir değer üretmelidir. Eğer bu başlangıç noktası kısıtlara yaklaşıyorsa o zaman bu fonksiyon çok büyük değerler üretmesi gerekmektedir. Bunu sağlayacak olan fonksiyonlar eşitlik (3.37) ve eşitlik (3.38)’de verilmiştir (Rao, 2009b).

$$G_j = -\frac{1}{g_j(\mathbf{x})} \quad (3.37)$$

$$G_j = -\log(g_j(\mathbf{x})) \quad (3.38)$$

Şekil 3.3’de bir minimizasyon işlemi için verilen bir fonksiyonun optimum noktasına yaklaşılrken penalty fonksiyonun etkisi verilmiştir (Rao, 2009a). Optimum noktaya yaklaşılrken sınırların aşılması için hata sonsuza yaklaşmaktadır.



**Şekil 3.3 :** Interir Point Penalty Fonksiyonu Yaklaşımı

Burada penalty metodu bariyer gibi davrandığından bu metod aynı zamanda bariyer method olarakda bilinir. Bu metodun en büyük sıkıntısı feasible bir başlangıç noktası olmazsa yapısı gereği sonsuz bir hatadan başlamak zorunda kalacaktır. Bu nedenle bu metod kullanılırken ilk olarak kısıtları sağlayan bir başlangıç noktasının belirlenmesi gerekmektedir.

İteratif olarak yapılan bu işlemin adımları aşağıdaki gibidir;

1.  $r > 0$  ve  $k=1$  olarak belirlenir. Tüm kısıtları sağlayan feasible bir başlangıç noktası olan  $\mathbf{x}$  seçilir.
2.  $\Phi(\mathbf{x}, r_k)$  herhangi bir kısıtsız optimizasyon tekniği ile minimize edilerek optimum  $\mathbf{x}^*$  değeri bulunur.
3. Eğer bulunan optimum  $\mathbf{x}^*$  istenilen şartı sağlıyorsa yada durdurma kriteri sağlanıyorsa algoritma durdurulur. Aksi takdirde 4. adıma geçilir.

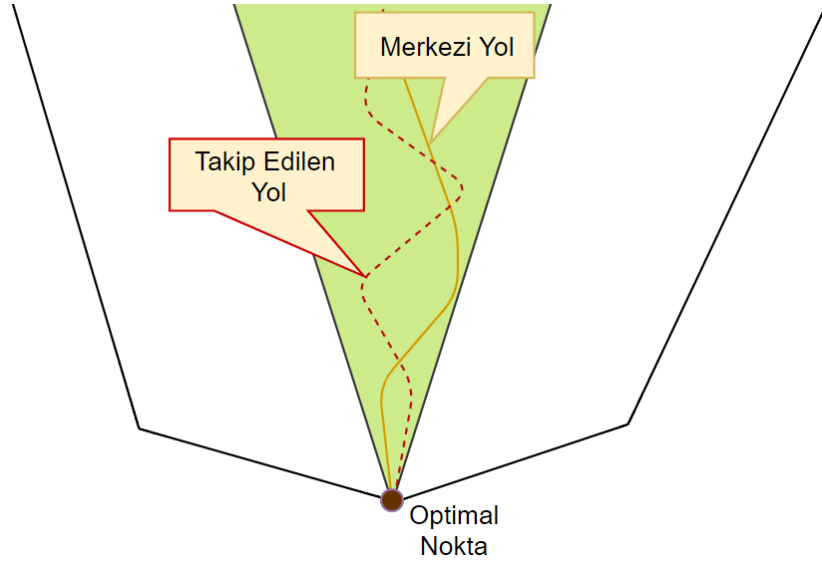
4. r katsayısı güncellenir. Bunun için  $c < 1$  olmak üzere  $r_{k+1} = c \cdot r_k$  ile yeni değer belirlenir.
5.  $k=k+1$  olarak belirlendikten sonra  $\mathbf{x}=\mathbf{x}^*$  yapılarak yeni başlangıç noktası optimum noktası olarak atanır. 2. Adıma gidilir.

Eşitlik kısıtı içeren bir problem çözümlenmeye çalışıldığında ise bu belirlenen  $H(r_k)$  fonksiyonu sonsuza yaklaşan bir fonksiyon seçilebilir. Böyle olunca eşitlik ifadesinin hata fonksiyonu penalty değeri 0'a yakınsaması sağlanacaktır. Örnek olarak Fiacco ve McCormick'in önerdiği eşitlik (3.39) kullanılabilir. Ardından yukarıdaki işlem adımları yapılarak bu problem çözümlenebilir

$$\phi(\mathbf{x}, r_k) = f(\mathbf{x}) - r_k \sum_{j=1}^m \log(g_j(\mathbf{x})) + \frac{1}{\sqrt{r_k}} \sum_{j=1}^n l_j(\mathbf{x})^2 \quad (3.39)$$

### 3.3.3.1.1 Primal-Dual interior metodu

Bu teknikte problemin minimizasyonu ve maksimizasyonu birbirine eşitlendiğinde ikisinin optimum noktalarının çakışması ile ortaya çıkmıştır. Lakin bu ikisi arasında yeni bir form oluşmuş ve Karush Kuhn Tucker koşulunun (KKT) tamamlayıcı olan şartına küçük bir katsayı eklenmiştir. Aslında bu katsayı 0 olduğunda KKT koşulu olmaktadır. Ardından Newton tekniği ile yön ve ardından adım uzunluğu belirlenmektedir. Bu teknikte takip edilen yol KKT ile elde edilen merkezi yolun etrafından belirli bir sınırlı alanla optimal noktaya ulaşmaktadır. Şekil 3.4'da bu verilmiştir (Wright, 1997).



**Şekil 3.4 : Primal-Dual Yol Takibi**

Lineer programlar için düşünüldüğünde minimizasyon yapılmak istenen primal probleme karşılık ve bunun duali olan problemde maximizasyon yapılmak istenmektedir. Basit bir problem eşitlik (3.40)'de verilmiştir. Burada verilen  $s$  slack değişkendir

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \quad \begin{array}{ll} \max & \mathbf{b}^T \boldsymbol{\lambda} \\ \text{s.t.} & \mathbf{A}\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c} \end{array} \quad (3.40)$$

Lagrange fonksiyonu uygulandığında eşitlik (3.41) elde edilmektedir.

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) = \mathbf{c}^T \mathbf{x} - \boldsymbol{\lambda}^T (\mathbf{Ax} - \mathbf{b}) - \sum_{i=1}^m s_i x_i \quad (3.41)$$

Optimal bir sonuç elde edilebilmesi için Lagrange fonksiyonu eşitlik (3.42)'de verilmiştir.

$$\begin{array}{ll} \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda} - \mathbf{s} & = 0 \\ \mathbf{Ax} & = 0 \\ \mathbf{xs} & = 0 \\ \mathbf{x}, \mathbf{s} & \geq 0 \end{array} \quad (3.42)$$

“Barrier” fonksiyonu ise eşitlik (3.43)'de verilmiştir. Burada  $\mu$ 'ye bağlı olarak problem eşitlik (3.44)'deki gibi olacaktır.

$$\phi(x, \mu) = f(x) - \mu \sum_{i=1}^m \log(x_i) \quad (3.43)$$

$$NLP_{\mu} = \begin{array}{ll} \min & f(x) - \mu \sum_{i=1}^m \log(x_i) \\ \text{st} & Ax = b \end{array} \quad (3.44)$$

Lagrangen'in "barrier" fonksiyonu da eşitlik (3.45)'deki gibi elde edilir.

$$L_{\mu}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T \mathbf{x} - \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{x} - \mathbf{b}) - \sum_{i=1}^m \log(x_i) \quad (3.45)$$

Sabit bir  $\mu$  değeri için  $NLP_{\mu}$  çözümlendiğinde KKT koşulları sağlanması gerekmektedir. Dolayısıyla KKT optimal şartı için eşitlik (3.46) uygulandığında eşitlik (3.47) elde edilmektedir

$$\begin{array}{l} \nabla_x L_{\mu}(x, \lambda) = 0 \\ \nabla_{\lambda} L_{\mu}(x, \lambda) = 0 \\ x > 0 \end{array} \quad (3.46)$$

$$\begin{array}{l} c - A^T \lambda - \mu X^{-1} e = 0 \\ Ax = b \\ x > 0 \end{array} \rightarrow \begin{array}{l} \mu X^{-1} e = s \\ Ax = b \\ (x, s) > 0 \end{array} \Rightarrow \begin{array}{l} c - A^T \lambda - \mu X^{-1} e = 0 \\ Ax = b \\ s = \mu X^{-1} e \\ (x, s) > 0 \end{array} \quad (3.47)$$

Bu denklem düzenlendiğinde sonuç olarak eşitlik (3.48) elde edilir

$$\begin{array}{l} A^T \lambda + s = c \\ Ax = b \\ sx = \mu e \\ (x, s) > 0 \end{array} \quad (3.48)$$

Bu denklem içerisindeki  $\mu$  aslında "central path" denilen asıl yola gidişe olan uzaklığının bir parametresidir. Merkezi nokta ise Eşitlik (3.49)'deki gibi verilebilir. Ayrıca bu parametre "feasible" çözümün içerisinde kalmasını sağlamaktadır.

$$C = \{x(\mu), \lambda(\mu), s(\mu) \mid F_{\mu}(x(\mu), \lambda(\mu), s(\mu)) = 0, (x(\mu), s(\mu)) > 0\}$$

$$F_{\mu}(x(\mu), \lambda(\mu), s(\mu)) = \begin{bmatrix} Ax - b \\ A\lambda + s - c \\ xs - \mu e \end{bmatrix} = 0 \quad (3.49)$$

Eşitlik (3.49)'de verilen yön bilgisi Newton tekniği uygulanarak elde edilebilir. Buda "feasible" çözümün değişim yönlerini bulmakla mümkün olmaktadır. Bunun için eşitlik (3.50) kullanılır. Buradan da daha açık bir form olan eşitlik (3.51) elde edilir.

$$J_{\mu}(x, \lambda, s)d = -F_{\mu}(x, \lambda, s) \quad (3.50)$$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ X & 0 & S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} Ax - b \\ A\lambda + s - c \\ xs - \mu e \end{bmatrix} \quad (3.51)$$

Bu denklemden elde edilen adım uzunlukları kullanılarak değişkenlerin değerleri güncellenir.

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k \Delta x_k \\ \lambda_{k+1} &= \lambda_k + \alpha_k \Delta \lambda_k \\ s_{k+1} &= s_k + \alpha_k \Delta s_k \end{aligned} \quad (3.52)$$

Buradaki adım uzunluğu line search veya farklı teknikler ile de belirlenebilir. Bu tekniği bir parametresi olan  $\mu$  ise eşitlik (3.53) kullanıldığında zamanla 0'a yakınsayacaktır. Bu denklemde kullanılan  $\sigma$  bir parametre olup çoğunlukla 0.01 seçilmektedir.

$$\mu(x, s) = 1 - \sigma \frac{\mu(x, s)}{\sqrt{n}} \quad (3.53)$$

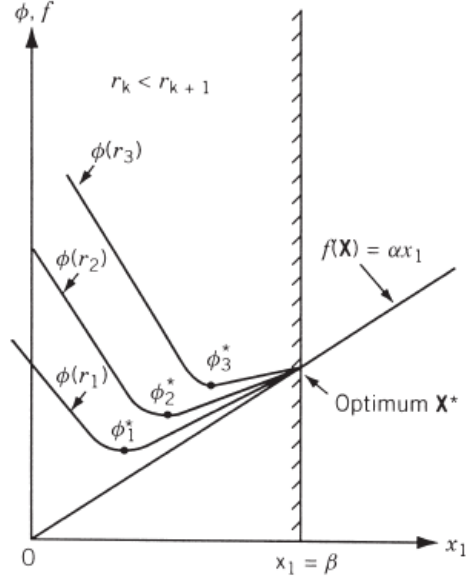
### 3.3.3.2 Exterior penalty metodu

Bu teknikte de kullanılan amaç fonksiyonu eşitlik (3.34) de bulunan fonksiyonun içerisindeki  $G_j$  fonksiyonu eşitlik (3.54)'de verilmiştir. Bu fonksiyon içerisindeki  $q$  ise negatif olmayan reel bir sayıdır.

$$G_j(\mathbf{x}) = \left\langle [g_j(\mathbf{x})] \right\rangle^q \quad (3.54)$$

$$\begin{aligned} \langle g_j(\mathbf{x}) \rangle &= \max(g_j(\mathbf{x}), 0) \\ &= \begin{cases} g_j(\mathbf{x}) & g_j(\mathbf{x}) > 0 \\ 0 & g_j(\mathbf{x}) < 0 \end{cases} \end{aligned} \quad (3.55)$$

Eşitlik (3.54)'deki terimden anlaşılacağı üzere  $q$  üssel bir ifade olduğundan bu eşitsizlik kısıtı ihlal edildiğinde anda bu amaç fonksiyonundaki etkisi üstel bir şekilde artacaktır. Yani ihlal durumunda amaç fonksiyonu değerini üstel artıracığından bu terim ihlal çözümünü cezalandırma işlevi görür Bu nedenden ötürü bu tekniğe penalty tekniği denilmektedir (Rao, 2009b).



**Şekil 3.5 :** Exterior Point Penalty Fonksiyonu Yaklaşımı

Şekil 3.5’da gösterildiği gibi penalty fonksiyonunun değeri arttıkça kısıt ihlali azalmakta ve cevap optimum noktaya yaklaşmaktadır (Rao, 2009a).

İteratif olarak yapılan bu işlemin adımları;

1. Herhangi bir  $\mathbf{x}$  başlangıç noktası belirlenir. Uygun bir  $r_1$  değeri ataması ve  $k=1$  yapılır.
2.  $\Phi(\mathbf{x}, r_k)$  herhangi bir kısıtsız optimizasyon tekniği ile minimize edilerek optimum  $\mathbf{x}^*$  değeri bulunur.
3. Eğer bulunan optimum  $\mathbf{x}^*$  istenilen şartı sağlıyorsa, “feasible” bölgede ise ve durdurma kriteri sağlanıyorsa algoritma durdurulur. Aksi takdirde 4. adıma geçilir.
4.  $r$  katsayısı güncellenir. Bunun için  $c>1$  olmak üzere  $r_{k+1} = c \cdot r_k$  ile yeni değer belirlenir.
5.  $k=k+1$  olarak belirlendikten sonra  $\mathbf{x} = \mathbf{x}^*$  yapılarak yeni başlangıç noktası optimum noktası olarak atanır. 2. Adıma gidilir.

Eşitlik kısıtı içeren bir problem çözümlenmeye çalışıldığında ise bu belirlenen  $H(r_k)$  fonksiyonu sonsuza yaklaşan bir fonksiyon seçilebilir. Dolayısıyla direk olarak  $r_k$  bu değere yakınsadığı için bu katsayı seçilebilir. Bunun için eşitlik (3.56) kullanılabilir. Yukarıdaki işlem adımları yapılarak bu problem çözümlenebilir

$$\phi(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m \left\langle \left[ g_j(\mathbf{x}) \right] \right\rangle^q + r_k \sum_{j=1}^n l_j(\mathbf{x})^2 \quad (3.56)$$

### 3.4 Doğrusal Olmayan Modele Dayalı Öngörülü Kontrolde Kararlılık

Doğrusal olmayan modele dayalı öngörülü kontroldeki ana problemlerinden biri, sınırlı bir ufukta modele dayalı öngörülü kontrol stratejisinin kapalı çevrimin kararsızlığa neden olup olmayacağı konusudur. Anlaşılacağı üzere, sınırlı bir tahmin ve kontrol ufku ile en önemli sıkıntı, tahmin edilen açık döngü ve sonunda ortaya çıkan kapalı döngü davranışlarının çoğunlukla birbirlerinden farklı olmasıdır.

NMPC için kapalı çevrim kararlılığı sağlamak için sınırlı ufuk uzunluğu kullanılarak farklı methodlar sunulmuştur. Doğrusal olmayan modele dayalı öngörülü kontrole sonsuz ufuk yaklaşımı olarak isimlendirilen (quasi-infinite horizon), sonuç odaklı bir kararlılık sağlayan bir method bulunmaktadır (Rolf Findeisen & Frank Allgöwer, 2002).

Bu yaklaşım sayesinde sistem hesaplanabilir bir şekilde uygulanabilir ayrıca kapalı çevrim kararlılığı da sağlamış olur. Ayrıca bu yöntemde tahmin ufkunun kontrol ufkuna  $T_p = T_c$  'ye eşit olduğu kabul edilmiştir.

#### 3.4.1 Sonsuz ufuklu doğrusal olmayan modele dayalı öngörülü kontrol

Kararlılığı sağlamak amaçlı en sezgisel yol sonsuz öngörü ufkunun kullanılmasıdır (Keerthi & Gilbert, 1988). Bellman'ın optimal prensibine göre  $T_p$  sonsuza ayarlanır. İlk anda nominal durumda bir örnekleme anındaki uygulanabilirlik, bir sonraki örneklemede uygulanabilirliği ve optimalliği beraberinde getirir (Bellman, 1957).

Örnek olarak belirli bir anda NMPC ile çözümlene yapıldığında sistemee uygulanan giriş ve durum yörüngeleri, gerçek doğrusal olmayan sistemin kapalı çevrim yörüngelerine eşittir. Sonraki örnekleme anından yörüngelerin kalan kısımları için diğer örnekleme sırasında optimal çözüm uygulanmaktadır. Bu ise kapalı döngü kararlılığını beraberinde getirir (Rolf Findeisen & Frank Allgöwer, 2002).

Doğrusal olmayan modele dayalı öngörülü kontrol için hemen hemen bütün kararlılık kanıtları, sonsuz ufuk kanıtıyla benzer başlıca adımları takip dolayı temel fikirler kısaca özetlenebilir. İlke olarak ispat ise bir fonksiyonunun Lyapunov fonksiyonu olarak kullanılmasını esas alır.

Birinci adımda, bir örnekleme anındaki uygulanabilirlik, nominal durum için bir sonraki örnekleme anındaki uygulanabilirlik sonucuna ulaşılır.

İkinci adımda, bu fonksiyonunun azaldığı ve bu sayede de durum ve girişin orijine yaklaştığı belirlenir (Findeisen & Allgöwer, 2002).

Üçüncü adımda bu fonksiyonunun sürekli ve monoton özellikleri ile asimptotik kararlılık sağlanmıştır.

Son ve dördüncü adımda ise uygulanabilirlik asimptotik kararlılığı kullanmayı gerektirdiğinden, açık döngü optimal kontrol probleminin çözümü olan bütün durumlar orijin bölgesinde kabulü yapılır.

### 3.4.2 Sonlu ufuklu kesin kararlı doğrusal olmayan modele dayalı öngörülü kontrol

Kesin kararlı durum için amaç fonksiyonuna ek penalty fonksiyonu eklenir. Bu kapalı çevrim kararlılığı arttırabileceği bu penalty fonksiyonu sistem kısıtları veya sistem durumlarında biri olabilir. Dolayısıyla kararlılık öngörü ufkunun sonundaki termal kısıtının eklenmesiyle sağlanabilir. Terminal kısıtının eklemesi aslında son noktanın istenilen referans değerine yakınsayıp yakınsamadığını da sorgulamaktadır (Keerthi & Gilbert, 1988; Mayne et al., 2000).

Takip problemi için kesin kararlılık sağlayacak olan durum kısıtı eşitlik (3.57)'de verilmiştir.

$$x(t + T_p; x(t), t, u) = x_{ref} \quad (3.57)$$

Böyle bir çözüm ise durum sayısı fazla olan sistemler için problem teşkil edebilir. Çünkü böyle bir çözüm sonlu bir zaman diliminde istenen çıkışı referans değerine yakınsamasına ve diğer durumları göz ardı etmesine neden olabilecektir.. Eğer öngörü ufku kısa seçilirse söz edilen sorun daha baskın hale gelecektir. Durumların geçişleri çok agresiff olacaktır. Lakin bu yöntemi uygulaması ve anlaması oldukça basittir (Chen & Allgöwer, 1998).

Terminal kısıtı gevşetilerek, eşitlik (3.58)'de belirtildiği gibi terminal durumun kararlılığın garanti olduğu küme içerisinde kalması koşulu terminal kısıt olarak belirlenebilir.

$$x(t + T_p) \subseteq X \quad (3.58)$$

Terminal fonksiyonun probleme eklenmesi ise oldukça basittir. Bu fonksiyonun performans özelliği olmayıp serbestçe seçilebilmektedir. E fonksiyonu olarak

tanımlanırsa amaç fonksiyonu eşitlik (3.59)'daki gibi revize edilebilir. Bu fonksiyon sayesinde belirlenmiş olan terminal kümesi içinde kararlılık sağlanmış olmaktadır.

$$J(x(t), u, T_c, T_p) = E(x(t + T_p)) \int_t^{t+T_p} F(x(\tau), u(\tau)) d\tau \quad (3.59)$$





## 4 FAN VE LEVHA SİSTEMİNİN NMPC İLE KONTROL UYGULANMASI

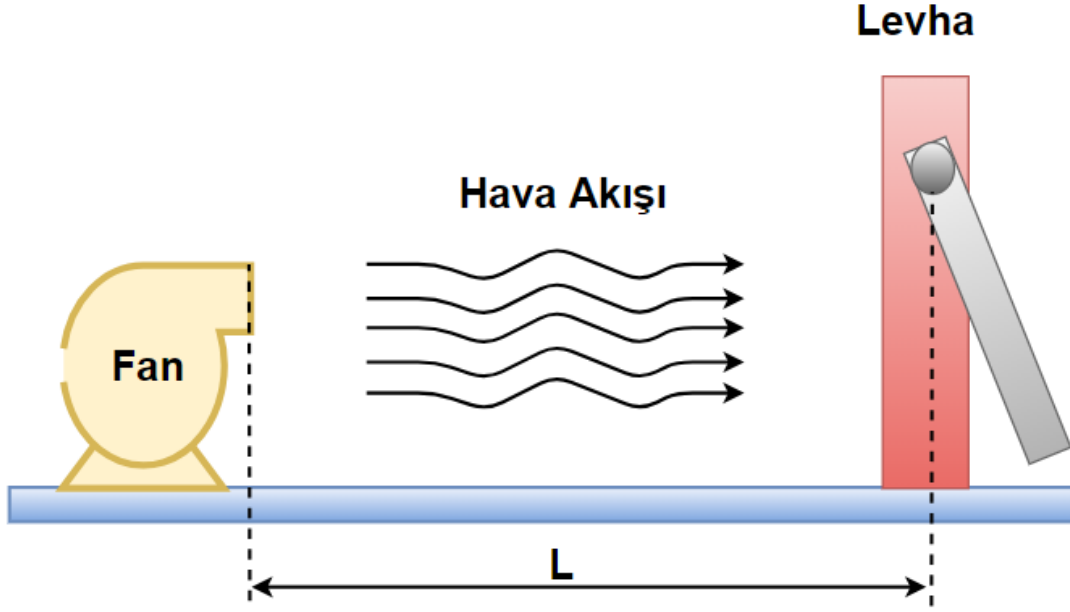
### 4.1 Fan ve Levha Sistemi

Fan ve Levha düzeneği yapısında bulundurduğu belirsizlikler ve doğrusal olmayan davranışı ile önemli bir kontrol problemidir. Bu sistemi kontrol edebilen bir kontrolör, model belirsizliklerinin ve doğrusal olmayan sistem davranışlarının üstesinden gelebildiği gösterir. Dahası farklı kontrol yapıları arasında, bu sistem üzerinden yapılan testlerin sonuçlarına bakılarak, karşılaştırma yapılabilir. Bu nedenle yeni kontrol yapılarının eskilere göre başarılı olup olmadığını gösterebilecek bir deney düzeneğidir. Bununla birlikte Şekil 4.1’de gösterilen Fan ve Levha test düzeneğinin resmi verilmiştir. Daha iyi anlaşılması için şematik çizimide Şekil 4.2’de verilmiştir.

Fan ve Levha sisteminin gayet basit bir yapısı bulunmaktadır. Fan vasıtası ile oluşturulan hava akımı ile levhanın açılma hareketi sağlanmaktadır (Dincel, Yalçın, & Kurtulan, 2014).

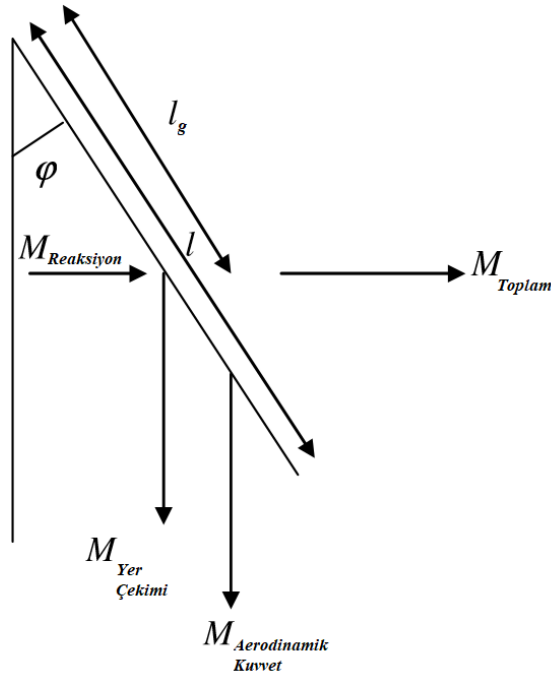


Şekil 4.1 : Fan ve Levha Sistemi



Şekil 4.2 : Sistemin Modeli

Levhanın hareket denklemlerini çıkarabilmek için Şekil 4.3’de verilen model kullanılmıştır. Levhaya Rüzgarın uygulamış olduğu kuvvet, Yerçekimi kuvveti, havanın aerodinamik yapısından dolayı oluşan kuvvetler etmektedir ve bu kuvvetlerin toplamı levhaya net bir kuvvet uygulanmasını sağlamaktadır.



Şekil 4.3 : Levhanın Model

Ayrıca, motora elektriksel enerjinin verilmesi ardından hava akımının oluşup levhaya ulaşip levhayı hareket ettirebilen düzeye ulaşması bir süre almaktadır Ölü Zaman olarak eşitlik (4.1)'den bulunabilir (Sharma, 2007).

$$T(t) = \frac{L + l_g \sin(\varphi(t))}{v(t)} \quad (4.1)$$

Newton yasaları kullanılarak sistem eşitlik (4.2) ile ifade edilebilir. Durumlar yazılmak istendiğinde de kolay anlaşılması için eşitlik (4.3) ile isimleri değiştirilebilir (Sharma, 2007).

$$J\ddot{\varphi}(t) = -mgl_g \sin(\varphi(t)) - b\dot{\varphi}(t) + kv^2(t)al \cos(\varphi(t)) \quad (4.2)$$

$$\varphi(t) = x_1(t)$$

$$\dot{\varphi}(t) = x_2(t)$$

$$v(t) = u(t)$$

(4.3)

Durumların denklemleri eşitlik (4.4)'de ifade edilmiştir (Sharma, 2007).

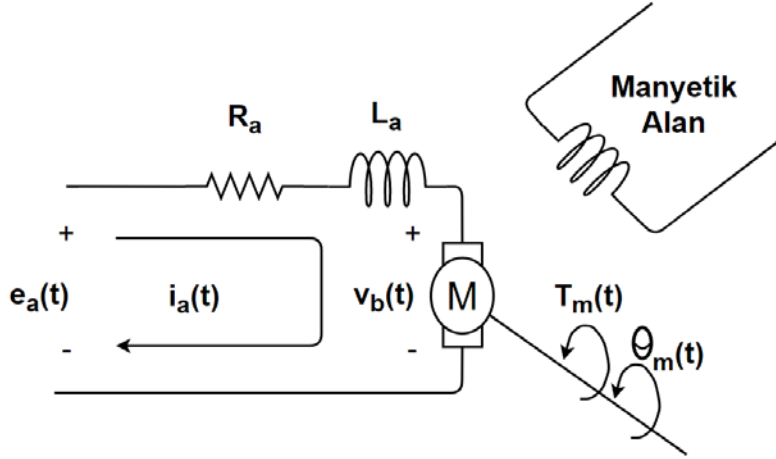
$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -A \sin(x_1(t)) - Bx_2(t) + Cu^2(t-T) \cos(x_1(t)) \quad (4.4)$$

$$y(t) = x_1(t)$$

$$A = \frac{mgl_g}{J}; \quad B = \frac{b}{J} \quad C = \frac{kal}{J} \quad (4.5)$$

Sistemi harekete geçiren sistemin bir DC motor olduğu varsayımı yapılarak işlemler yapılırsa motorun modelinin elde edilmesi gerekmektedir. DC motorun yapısı Şekil 4.4'de verilmiştir (Nise, 2011).



Şekil 4.4 : DC Motor Modeli

Motorun elektromekanik dinamiklerine ilişkin Transfer Fonksiyonu eşitlik (4.6)'da verilmiştir. Endüktans değeri çok küçük olarak kabul edildiğinde eşitlik (4.7) elde edilir (Nise, 2011).

$$\frac{\theta(s)}{e_a} = \frac{K_T}{(R_a + L_a s)(J_m s^2 + D_m s) + K_b K_T} \quad (4.6)$$

$$L_a \rightarrow 0 \Rightarrow \frac{\theta(s)}{e_a} = \frac{K_T}{R_a J_m s^2 + D_m s + K_b K_T s} \quad (4.7)$$

Hız ile gerilim arasında bir ilişki kurulmak istendiğinde eşitlik (4.8) elde edilmektedir.

Görüldüğü gibi bu ilişki kazançlı  $\frac{K_T}{D_m + K_b K_T}$ , zaman sabiti  $\frac{R_a J_m}{D_m + K_b K_T}$  olan birinci dereceden bir sisteme karşı düşmektedir (Nise, 2011).

$$w(t) = \dot{\theta}(t) \xrightarrow{L} w(s) = \theta(s)s \Rightarrow \frac{w(s)}{e_a} = \frac{K_T}{R_a J_m s + D_m + K_b K_T} \quad (4.8)$$

$$\frac{w(s)}{e_a} = \frac{K_T / (R_a J_m)}{s + \frac{(D_m + K_b K_T)}{R_a J_m}} = \frac{K}{\tau s + 1}$$

Sistemi durum denklemleri halinde yazmak istendiğinde eşitlik (4.9) elde edilmektedir.

Bu durumda  $w = x_3$  olarak değerlendirildiğinde sistemin toplam denklemi eşitlik (4.10) elde edilmektedir. Sistemdeki ölü zaman eşitlik(4.1)'de verilmiştir.

$$\frac{w}{e_a} = \frac{K}{\tau s + 1} \Rightarrow w(\tau s + 1) = K e_a \xrightarrow{L^{-1}} \dot{w} = \frac{-w + K e_a}{\tau} \quad (4.9)$$

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -A \sin(x_1(t)) - Bx_2(t) + Cx_3^2(t-T) \cos(x_1(t)) \\
\dot{x}_3(t) &= -x_3(t)/\tau + Ke_a/\tau \\
y(t) &= x_1(t)
\end{aligned} \tag{4.10}$$

$$A = \frac{mgl_g}{J}; \quad B = \frac{b}{J} \quad C = \frac{k a l}{J}$$

Sistem parametrelerinin belirlenmesi için Fan ve Levha sistemine farklı genliklerde giriş işaretleri uygulanmıştır. Ardından sisteme uygulanan ve sistemden ölçülen veriler alınmıştır. Bu veriler ile sistemi temsil eden eşitlik (4.10)'un parametrelerini belirlenmesi için Ateş böceği algoritması kullanılmıştır. Buradaki amaç model ile sistem arasındaki hatayı minimum yapan parametrelerin belirlenmesidir. Ama bazı parametreler ölçülebilmektedir. Bu nedenle ölçülmesi mümkün olmayan parametreler uzaklıklar ile belirlendi diğer parametrelerin belirlenmesi sağlandı. Buradan elde edilen sonuçlar çizelge 4.1'de verilmiştir.

**Çizelge 4.1: Sistem Parametreleri**

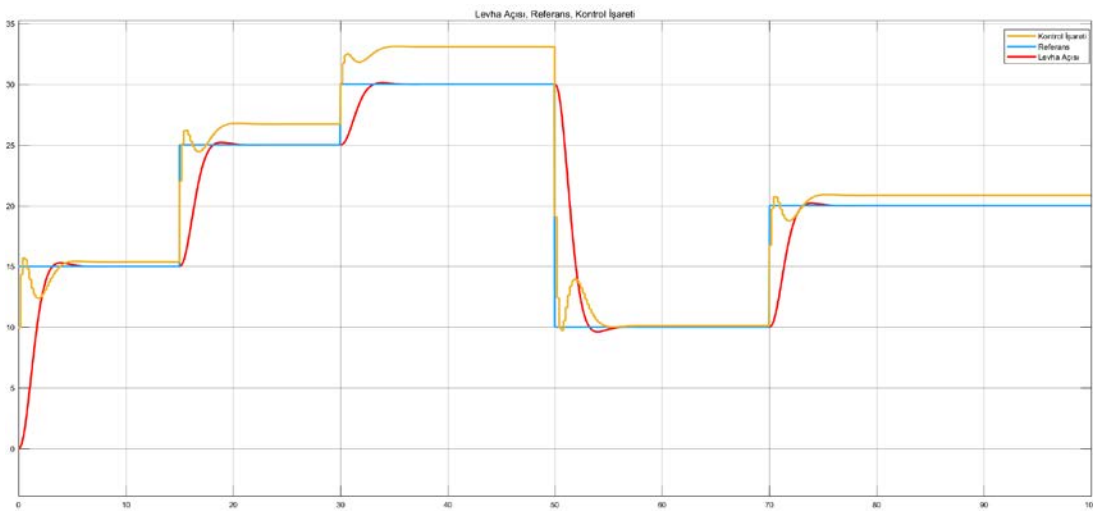
Parametre	Sembol	Değer	Birim
Levhanın kütlesi	m	0,566	kg
Levhanın Ağırlık Merkezine olan uzaklığı	$l_g$	0,105	kg m
Levhanın uzunluğu	l	0,2	m
Levhanın alanı	a	0,03	m <sup>2</sup>
Sürtünme katsayısı	b	0.5641	kg/sn
Eylemsizlik	J	0.0155	m <sup>4</sup>
Levha ile Fan arası uzaklık	L	0,21	m
Aerodinamik direnç	k	0.5641	-
Ölü Zaman değer	T	Değişken	sn
Yerçekimi	g	9,81	m/s <sup>2</sup>
Motor kazancı	K	0.22e-3	
Motor zaman sabiti	$\tau$	0.0108	

## 4.2 NMPC'nin Matlab Uygulanması

Fan ve Levha sisteminin modelini ayrıklaştırmak için Euler metodu tercih edildi. Ayrık zamanlı NMPC optimizasyonu iteratif çözümü için ardışıl yöntem seçilmiştir. Bu yöntemlerin birleşimi ile çözüm yapılmak istendiğinde öngörü ufku boyutu ile sistemin durum sayısının çarpımı kadar durumlara ilişkin değişken ve kontrol ufku kadar da kontrol değişkeni tanımlamak gerekmektedir. Bunların yanı sıra eşitsizlik kısıtlarının slack değişkenleride eklendiği takdirde çok fazla değişkenin işlenmesi gerekmektedir. Bu tip bir problemi Matlab ortamında benzetim çalışmaları yapılarak çözümlenmesi mümkün ancak çözüm çok sayıda değişkeni işleyen karmaşıklığı yüksek bir algoritma tarafından üretildiğinden çözümün oluşturulması sistem dinamiklerinin ayrıklaştırıldığı örnekleme zamanından uzun sürebileceği için çözümün gerçek zamanlı olarak uygulanması pek mümkün olmayabilir. En azından eşitlik kısıtlamalarını ortadan kaldırarak durumların birbirine bağımlı bir şekilde hareket etmesi için ardışıl ve Euler Metotlarının birleşimi tercih edildi.

### 4.2.1 Ölü zamanlı sistemin ölü zamansız nmpc ile kontrolü

Eğer sistem muhtevası gereği ölü zaman içermemiş olsaydı o zaman kontrolü daha doğru olabilirdi. Sistemde ölü zamanın bulunmadığı durum için Şekil 4.5'de görüldüğü gibi MATLAB da NMPC uygulaması yapıldığında aşımı çok az her çalışma noktasında aynı davranışı gösteren iyi bir sonuç elde edilmektedir (Belirtilen simulasyonda sistemde de model de de ölü zaman bulunmamaktadır).



Şekil 4.5 : Ölü Zamansız Sistem ve Ölü Zamansız NMPC

Modelde ölü zaman ihmal edildiği durum (ölü zamanı içermeyen modelin kullanıldığı durum) için NMPC'nin oluşturulması anlacaktır İlk olarak ölü zamansız sistemin denklemleri matlab kodu Şekil 4.6'de verilmiştir

```
function [dxdt y]=Sis3Denk(x,u)
dxdt=[x(2);
      -37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x(3)^2;
      -0.0108*x(3)+0.0203*u];
y=x(1);
end
```

Şekil 4.6 : Ölü Zamansız Sistem

Eğer sistem lineer denklem takımı olmuş olsaydı analitik olarak çözmek mümkün olurdu. Lakin sistem lineer olmadığı için durumların bir noktasında diğer noktaya gitmesi için numerik yöntemler kullanılmaktadır. Bunun için Euler Metodu tercih edilmiştir. Bu işlemin denklemleri (4.11)'de verilmiştir. Bu işlemi verilen bu sisteme uygulamak istendiğinde sistemin Matlab Alt fonksiyonu Şekil 4.7'de verilmiştir.

$$x(t+h) = x(t) + h \cdot \dot{x}(t) \quad (4.11)$$

```
function [x y]=Sistem_DT(x,u,T)
h=0.01;
y=0;
for i=0:h:T
dxdt=[x(2);
      -37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x(3)^2;
      -0.0108*x(3)+0.0203*u];
x=x+h*dxdt';
end
y=x(1);
end
```

Şekil 4.7 : Sistemin Ayırıklaştırılması

Problem için ilk olarak amaç fonksiyonu belirlenmiştir. Bu amaç fonksiyonunda bir önceki durumlar bir sonraki durumlara bağlı oldukları için amaç fonksiyonu içi içe bir döngü ile oluşturulmuştur. Bu fonksiyon eşitlik (4.12)'de verilmiştir. Nitekim bu amaç

fonksiyonu kullanılarak gradyent ve Hessian Matrisleri oluşturulacaktır. Bu denklemden verilen “ $x_{bas}$ ” daha önceki durumlar için, “ $mvseq$ ” sisteme uygulanmak istenen kontrol işaretidir. Diğer bir giriş olan “ $mvlast$ ” ise sisteme uygulanmış olan son kontrol işareti,  $T$  örnekleme zamanı, “ $ref$ ” referans işareti, “ $CP$ ” kontrol ufku iken “ $HP$ ” öngörü ufkudur. Öngörü ufkunun bittikten sonra sisteme en son uygulanmış olan kontrol işaretinin öngörü bitene kadar tekrardan öngörü bitene kadar uygulanılan kısımdır.

$$f(x_{bas}, mvseq, mvlast, T, ref, CP, HP, r) = Q*(x_1(i) - ref)^2 + R*(mvseq(1) - mvlast)^2 + R*(mvseq(j) - mvseq(j-1))^2 + \quad (4.12)$$

$$Q*(x_1(k) - ref)^2 - r*\log(g_p(x_1(l))) \quad \left\{ \begin{array}{l} i = 1, \dots, CP \\ j = 2, \dots, CP \\ k = CP + 1, \dots, HP \\ p = 1, \dots, m \\ l = 1, \dots, HP \end{array} \right.$$

```
function J=f(x_bas,mvseq,mv_last,T,ref,CP,HP,offset,r)
    dxdt=x_bas';u=mv_last;x=[0; 0];x_bas;x=x_bas(:);x=x';h=0.01;
    for i=0:h:T
        dxdt=[x(2);-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x(3)^2;-0.0108*x(3)+0.0203*mv_last;];
        x=x+h*dxdt';
    end
    J=0;
    for i=1:CP
        if mvseq(i)<0
            J=J-r*log(mvseq(i)-1);
        end
        if mvseq(i)>1
            J=J-log(mvseq(i)-1);
        end
    end
    for i=1:CP
        for ii=0:h:T
            dxdt=[x(2);-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x(3)^2;-0.0108*x(3)+0.0203*mv_last;];
            x=x+h*dxdt';
        end
        if ii==1
            J=J+200*((x(1)-ref+offset)^2+200*(mvseq(i)-mv_last)^2);
        else
            J=J+200*(x(1)-ref+offset)^2+10*(mvseq(i)-mvseq(i-1))^2;
        end
    end
    for ii=1:HP
        for iii=0:h:T
            dxdt=[x(2);-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x(3)^2;-0.0108*x(3)+0.0203*mvseq(CP);];
            x=x+h*dxdt';
        end
        J=J+2*((x(1)-ref+offset)^2);
    end
end
```

Şekil 4.8 : Amaç Fonksiyonu

Gradyent Matrisinin oluşturulmasında ileri farklar yöntemi kullanılmıştır. Gradyent fonksiyonu amaç fonksiyonunun gradyanı olduğu için bu fonksiyon bu yapıya dönüştürülmüştür. İlk olarak gradyanı yapısı eşitlik (4.13) yapısı kullanılacaktır.

$$f'(x_i) = y_i' \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (4.13)$$

Gradyan oluşturulurken kontrol değişkenleri x durumları olduğu için her bir kontrol işareti için fonksiyondaki değişimin bulunması eşitlik (4.14)'de verilmiştir.

$$\nabla f = \sum_{i=1}^{CP} \frac{f(x_{bas}, mvseq2(i), mvlast, T, ref, CP, HP, r) - f(x_{bas}, mvseq(i), mvlast, T, ref, CP, HP, r)}{Eps} \quad (4.14)$$

{mvseq2(i) = mvseq(i) + Eps}

```
function fx=Df(x_bas,mvseq,mv_last,T,ref,CP,HP,offset,r)
    Eps=5e-3;
    fx=zeros(CP,1);
    for i=1:CP
        mvseq2=mvseq;
        mvseq2(i)=mvseq(i)+Eps;
        fx(i)=(f(x_bas,mvseq2,mv_last,T,ref,CP,HP,offset,r) - f(x_bas,mvseq,mv_last,T,ref,CP,HP,offset,r))/Eps;
    end
    fx;
end
```

**Şekil 4.9 :** Gradyan Hesabı

Hessien Matrisi verilen amaç fonksiyonun durumlara göre ikinci türevi olarak düşünülebilir. Bunun için SQP algoritmasının yapısında ilk olarak diyagonal bir yaklaşımla Hessien Matrisine birim Matrisi ataması yapılmaktadır. Ardından bu durum matrisi üzerinden BFGS algoritması ile sürekli olarak bu matrisi güncellemektedir. Lakin burada yapılan işlem dinamik optimizasyon olduğu için zamana bağlı türevin durumlara göre türevi alınmak yerine yine ileri yönlü türev yaklaşımı ile bu matris bulunmuştur. Bunun temel ifadesi eşitlik(4.15)'de yaklaşım

verilmiştir. Bu yaklaşım her bir durum için kullanıldığında eşitlik (4.16)'daki Hessien Matrisi elde edilmektedir. Eğer bu sistem n adet kontrol değişkenine sahipse o zaman bu matris nxn adet olmaktadır (Thomas, 2013).

$$\frac{d^2 f}{dx^2} = \frac{f(x + Eps, y + Eps) - f(x + Eps, y - Eps)}{4 * Eps * Eps} - \frac{f(x + Eps, y - Eps) - f(x - Eps, y - Eps)}{4 * Eps * Eps} \quad (4.15)$$

$$H = \begin{bmatrix} \frac{d^2 f}{dx^2} & \frac{d^2 f}{dx dy} \\ \frac{d^2 f}{dy dx} & \frac{d^2 f}{dy^2} \end{bmatrix} \quad (4.16)$$

```
function H=H_Belirle(x_bas,mvseq,mv_last,T,ref,CP,HP,offset,r)
h1=5e-3;h2=5e-3;H=zeros(CP,CP);a1=0;a2=0;a3=0;a4=0;
for i=1:CP
    for j=1:CP
        x0=mvseq;
        x0(i)=mvseq(i)+h1;
        x0(j)=x0(j)+h2;
        a1=f(x_bas,x0,mv_last,T,ref,CP,HP,offset,r);
        % f2([x(1)+h1 x(1)-h2])
        x0=mvseq;
        x0(i)=mvseq(i)+h1;
        x0(j)=x0(j)-h2;
        a2=f(x_bas,x0,mv_last,T,ref,CP,HP,offset,r);
        % % f2([x(1)-h1 x(1)+h2])
        x0=mvseq;
        x0(i)=mvseq(i)-h1;
        x0(j)=x0(j)+h2;
        a3=f(x_bas,x0,mv_last,T,ref,CP,HP,offset,r);
        % % f2([x(1)-h1 x(1)-h2])
        x0=mvseq;
        x0(i)=mvseq(i)-h1;
        x0(j)=x0(j)-h2;
        a4=f(x_bas,x0,mv_last,T,ref,CP,HP,offset,r);
        H(i,j)=(a1-a2-a3+a4)/(4*h1*h2);
    end
end
end
```

Şekil 4.10 : Hessien Matrisinin Oluşturulması

Tüm bu bileşenler belirlendikten sonra eşitlik (4.17)'deki gibi SQP algoritmasının Matrisine yerleştirilecektir. Bu işlem yapıldıktan sonra H matrisinin tersi alınarak eşitlik (4.18)'deki gibi adım uzunlukları belirlenecektir. Bu işlemi yapan Matlab Kodu Şekil 4.11'de verilmiştir. Burada kontrol ufku büyüdükçe matris büyümektedir. Dolayısıyla Matrisin tersini almak zor olabilmektedir.

$$SQP \Rightarrow H \Delta x + \Delta f$$

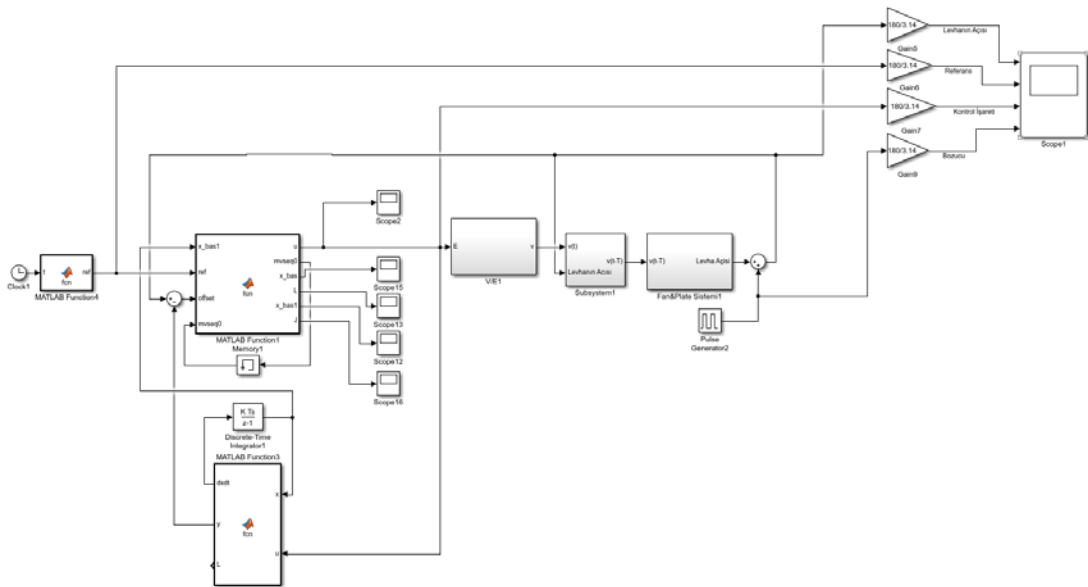
$$\Rightarrow \begin{bmatrix} H_{11} & \dots & H_{1 \times N} \\ \vdots & \ddots & \vdots \\ H_{N \times 1} & \dots & H_{N \times N} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_N \end{bmatrix} + \begin{bmatrix} \frac{df}{dx_1} \\ \vdots \\ \frac{df}{dx_N} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.17)$$

$$SQP \Rightarrow \Delta x = H^{-1} \Delta f$$

$$\Rightarrow \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_N \end{bmatrix} = \begin{bmatrix} H_{11} & \dots & H_{1 \times N} \\ \vdots & \ddots & \vdots \\ H_{N \times 1} & \dots & H_{N \times N} \end{bmatrix}^{-1} \begin{bmatrix} \frac{df}{dx_1} \\ \vdots \\ \frac{df}{dx_N} \end{bmatrix} \quad (4.18)$$

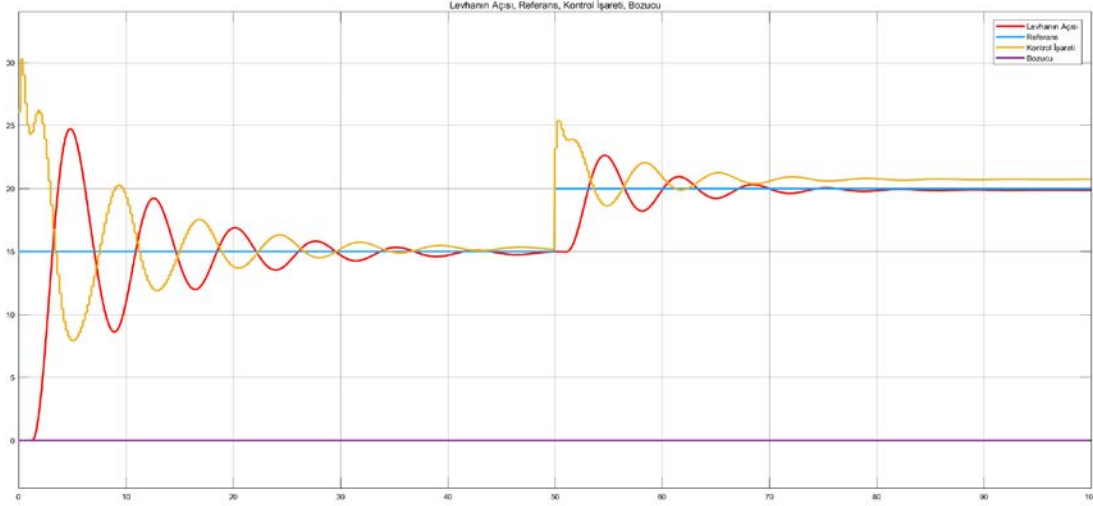
```
dx=inv(H)*(-Df(x_bas,mvseq,mv_last,T,ref,CP,HP)');
```

Şekil 4.11 : Adım Uzunluğunun Bulunması



Şekil 4.12 : NMPC ile Sistem Kontrolü

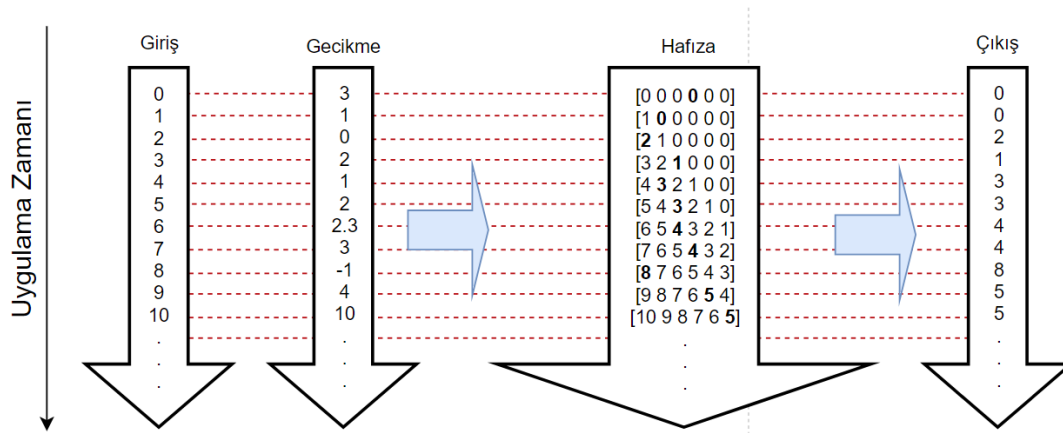
Bu sistem çalıştırıldığında sistem cevabı Şekil 4.13’de verilmiştir. Bu şekilde sistem cevabına bakıldığında aslında sistem ve model uyumsuzluğunda ölü zaman içermeyen NMPC ile kontrolü düzgün bir sonuç vermediği görülmüştür.



Şekil 4.13 : Ölü Zamansız NMPC ile Ölü Zamanlı Sistem Kontrolü

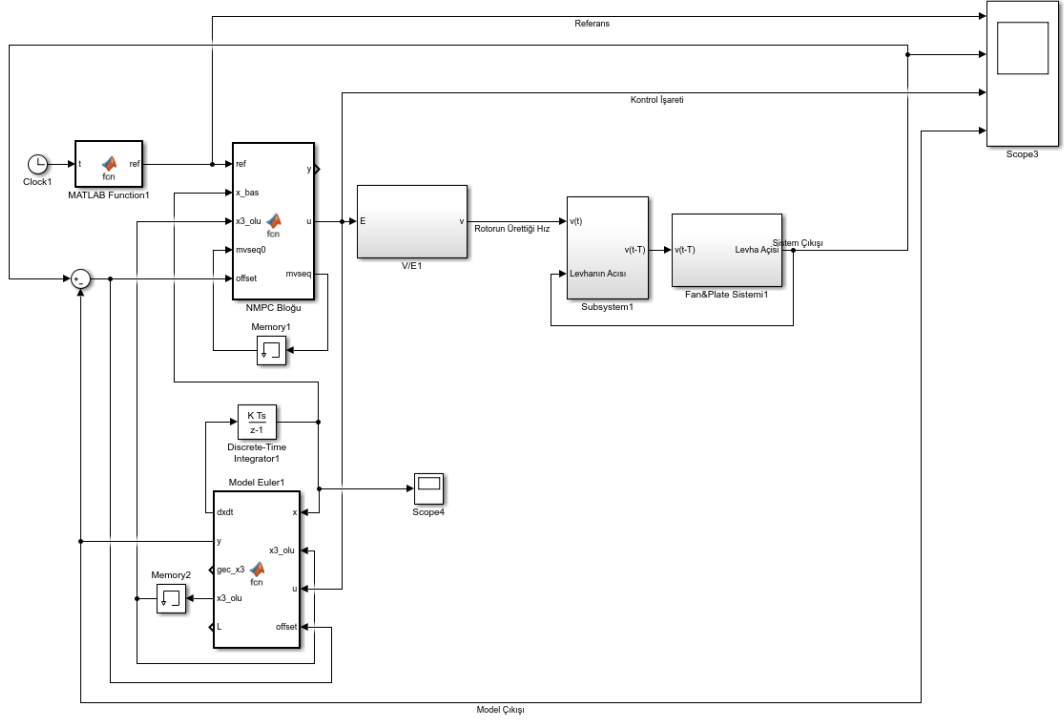
#### 4.2.2 Ölü zamanlı sistemin ölü zamanlı nmpc ile kontrolü

Daha doğru bir kontrol için yani bozucu olmadığı durumda öngörülerle sistem çıkışı arasında fark oluşmaması için modelde ölü zaman yer almalıdır. Bunu algoritmik olarak gerçekleyebilmek için önceki kontrol girişleri bir dizide saklanmış, değişken ölüzaman değeri eşitlik (4.1)’e göre hesaplanıp, hesaplanan ölü zamana ilişkin dizi elemanındaki kontrol değeri modele uygulanmıştır. Şekil 4.13’de görüldüğü gibi her örnekleme zamanında dizi bir kaydırılmış, o anki kontrol işareti ilk elemana yazılmıştır



Şekil 4.14 : Ölü Zamanın Kaydırılması

NMPC'nin içerisine aktarılan bu işlemi yapan sistemin blok diyagramı Şekil 4.14'de verilmiştir.



**Şekil 4.15 : Ölü Zamanlı Sistemin NMPC ile Kontrolü**

Şekil 4.15 deki içerisindeki sistemde ölü zaman Model Euler1 bloğunun çıkışındaki x3\_olu çıkışından alınmaktadır. Ardından Memory2 bloğu ile her örnekleme zamanında veriler bir adım kaydırılır ve ilk gelen veri dizinin 1. elemanının yerine yazılır. Bu veriler ve diğer veriler NMPC bloğu içerisinde kullanılarak öngörü yapmak için kullanılmaktadır.

```

function [y,u,mvseq,J] = fcn(ref,x_bas,x3_olu,mvseq0,offset)
y=0;u=0;mvseq=mvseq0;mv_last=mvseq(1);T=0.2;CP=1;HP=35;
L=(0.2+0.1*sin(x_bas(1)))/x_bas(3);
if L>3
    L=3;
elseif L<0
    L=T;
end
L2=max(round(L/T),1);L2=min(L2,15);
gec_x3=x3_olu(L2,1);
x3_olu(2:15,1)=x3_olu(1:14,1);
x3_olu(1,1)=x_bas(3);
for r=1000:-100:1
    H=H_Belirle(x_bas,x3_olu,mvseq,mv_last,T,ref,CP,HP,offset,r);
    dx=inv(H)*(-Df(x_bas,x3_olu,mvseq,mv_last,T,ref,CP,HP,offset,r));
    alfa=1;
    mvseq=mvseq+alfa*dx;
end
u=mvseq(1);
J=f(x_bas,x3_olu,mvseq,mv_last,T,ref,CP,HP,offset,r);
end

```

**Şekil 4.16 : NMPC Bloğunun İç Yapısı**

Şekil 4.16’de verilen Matlab kodu aslında NMPC’nin yapısına ilave olarak zaman gecikmesi ilave edilmiştir.

Ölü zamanlı olan bu sistem ayrıklaştırarak kullanılmasını sağlayan Model Euler Bloğunun içersindeki işlem Şekil 4.17 da verilmiştir. Bu modele gelen verilere ölü zaman bilgisine bağlı olarak bir diğer durumların kestirimi yapılmaktadır.

```

function [dxdt,y,gec_x3,x3_olu,L] = fcn(x,x3_olu,u,offset)
T=0.2;
L=(0.2+0.105*sin(x(1)))/x(3);
if L>3
    L=3;
elseif L<0
    L=T;
end
L2=max(round(L/T),1);
h=0.1;
for i=0:h:T
x=x+h*Sistem(x,x3_olu,u,T);
end
dxdt=[x(2);
-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x3_olu(L2)^2;
-0.0108*x(3)+0.0203*u];
y = x(1);
gec_x3=x3_olu(L2,1);x3_olu(2:15,1)=x3_olu(1:14,1);x3_olu(1,1)=x(3);
end
function dxdt=Sistem(x,x3_olu,u,T)
L=(0.2+0.105*sin(x(1)))/x(3);
if L>3
    L=3;
elseif L<0
    L=T;
end
L2=max(round(L/T),1);
dxdt=[x(2);
-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x3_olu(L2)^2;
-0.0108*x(3)+0.0203*u];
end

```

**Şekil 4.17** : Euler Model

NMPC bloğunun kullandığı amaç fonksiyonu ise ilk bölümleri Şekil 4.18, Şekil 4.19 ve Şekil 4.20’de verilmiştir.

```

function J=f(x,x3_olu,mvseq,mv_last,T,ref,CP,HP,offset,r)
x=x;h=0.01;L=(0.2+0.1*sin(x(1)))/x(3);
    if L>3
        L=3;
    elseif L<0
        L=T;
    end
L2=max(round(L/T),1);
L2=min(L2,15);
x3_olu(2:15,1)=x3_olu(1:14,1);
x3_olu(1,1)=x(3);
    for ii=0:h:T
        dxdt=[x(2);
-37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x3_olu(L2,1)^2;
-0.0108*x(3)+0.0203*mv_last;];
        x=x+dxdt*h;
    end
J=0;
for i=1:CP
    if mvseq(i)>1
        J=J-r*(mvseq(i)-1)^2;
    end
    if mvseq(i)<-0.001
        J=J-r*(1-mvseq(i));
    end
end
end

```

Şekil 4.18 : Amaç Fonksiyonu\_1

```

for i=1:CP
    L=(0.2+0.1*sin(x(1)))/x(3);
    if L>3
        L=3;
    elseif L<0
        L=T;
    end
    x3_olu(2:15,1)=x3_olu(1:14,1);x3_olu(1,1)=x(3);
    L2=max(round(L/T),1);L2=min(L2,15);
    for ii=0:h:T
        dxdt=[x(2);
            -37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x3_olu(L2,1)^2;
            -0.0108*x(3)+0.0203*mvseq(ii)];
        x=x+dxdt*h;
    end
    if i==1
        J=J+100*((x(1)-ref+offset))^2+100*(mvseq(ii)-mv_last)^2;
    else
        J=J+10*(x(1)-ref+offset)^2+100*(mvseq(ii)-mvseq(ii-1))^2;
    end
end
end

```

Şekil 4.19 : Amaç Fonksiyonu\_2

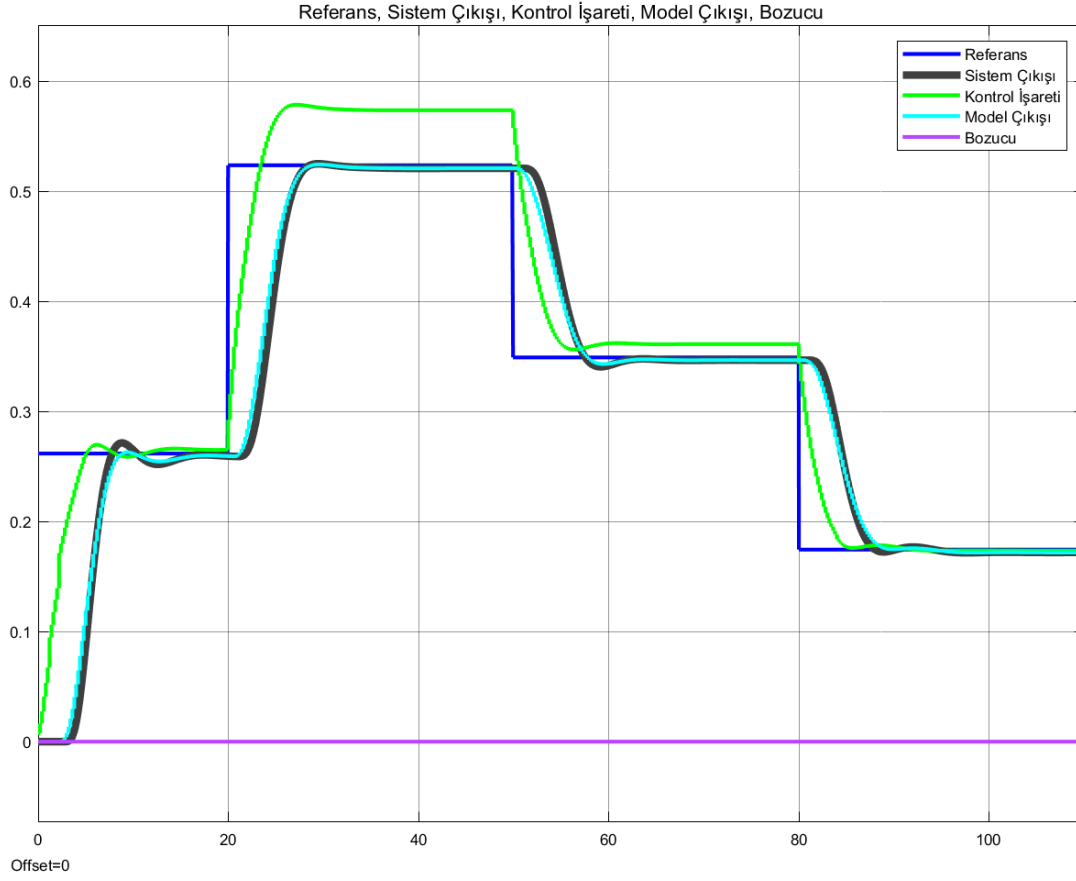
```

%Kontrol Ufkundan Sonraki Kısım Öngörü Ufkuna dayanıyor
for i=1:HP
    L=(0.2+0.1*sin(x(1)))/x(3);
    if L>3
        L=3;
    elseif L<0
        L=T;
    end
    L2=max(round(L/T),1);
    L2=min(L2,15);
    x3_olu(2:15,1)=x3_olu(1:14,1);
    x3_olu(1,1)=x(3);
    for ii=0:h:T
        dxdt=[x(2);
            -37.62*sin(x(1))-6.58*x(2)+0.218*cos(x(1))*x3_olu(L2,1)^2;
            -0.0108*x(3)+0.0203*mvseq(CP)];
        x=x+dxdt*h;
    end
    y=x(1);
    J=J+0.5*((x(1)-ref+offset))^2;
end
end

```

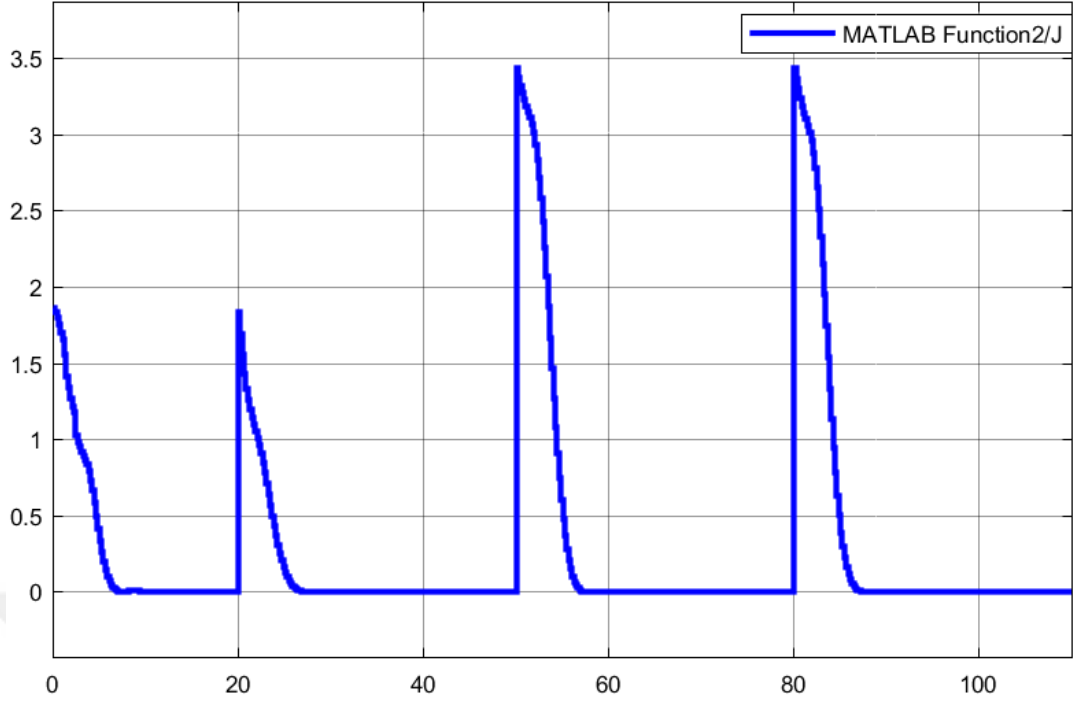
Şekil 4.20 : Amaç Fonksiyonu\_3

Bu amaç fonksiyonu içerisine yerleştirilmiş kaydırma işlemi ve eşitsizlik kısıtları ile oluşturulan Hessian ve Gradyent fonksiyonları ile sistem istenilen çıkışı verebilmiştir. Bu yapı sisteme uygulandığında sistem Şekil 4.21’de verilmiştir.

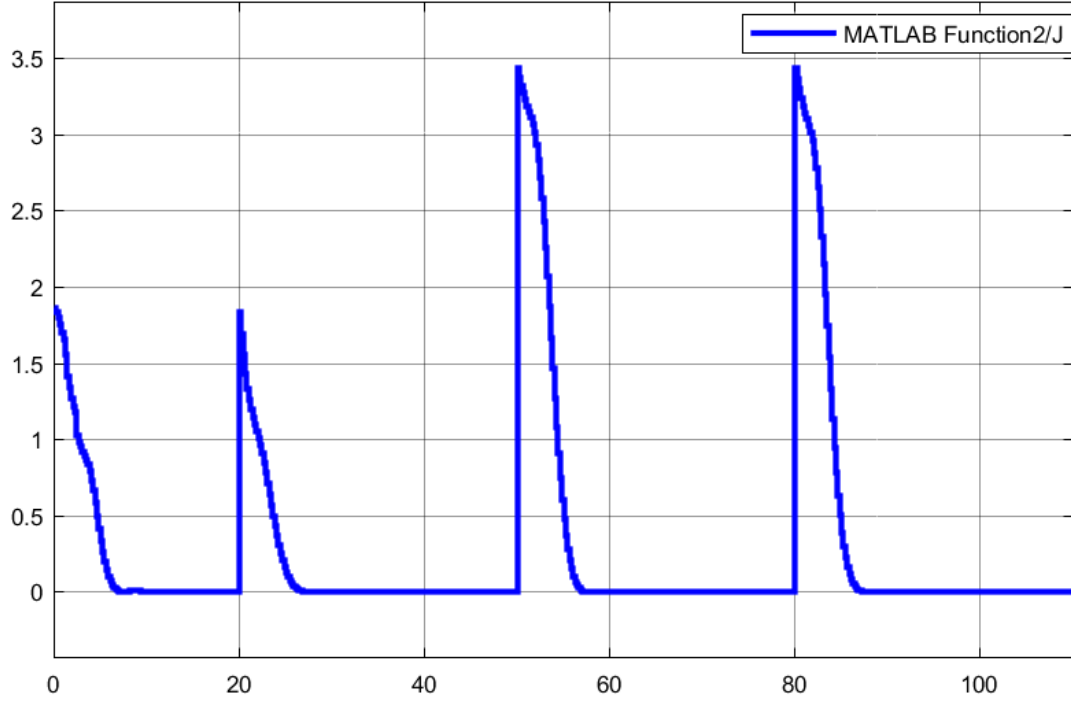


Şekil 4.21 : NMPC Kontrollü Sistem Çıkışı

Şekil 4.21’de uygulanmış referans işaretine bağlı olarak değişen sistem yanıtının zamana bağlı olarak amaç fonksiyon değeri



Şekil 4.22'de verilmiştir.

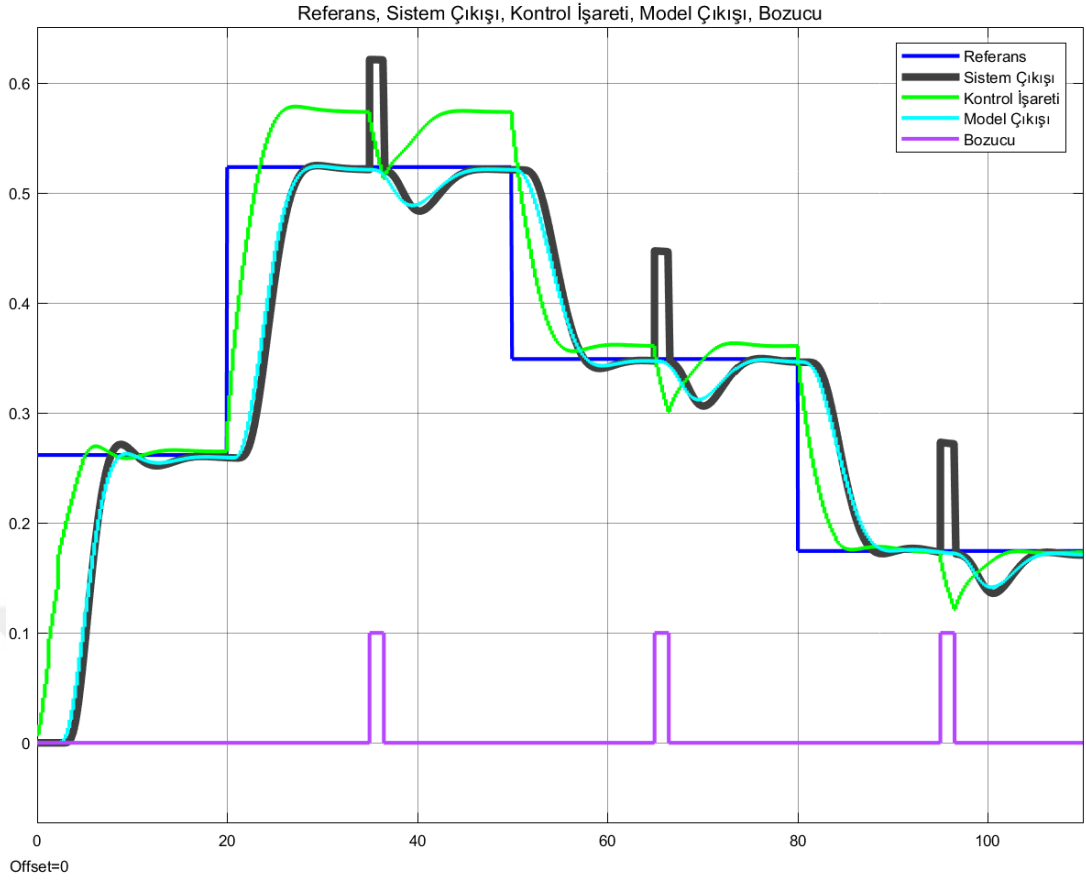


Şekil 4.22 : Amaç Fonsiyonunun Zamana Göre Değeri

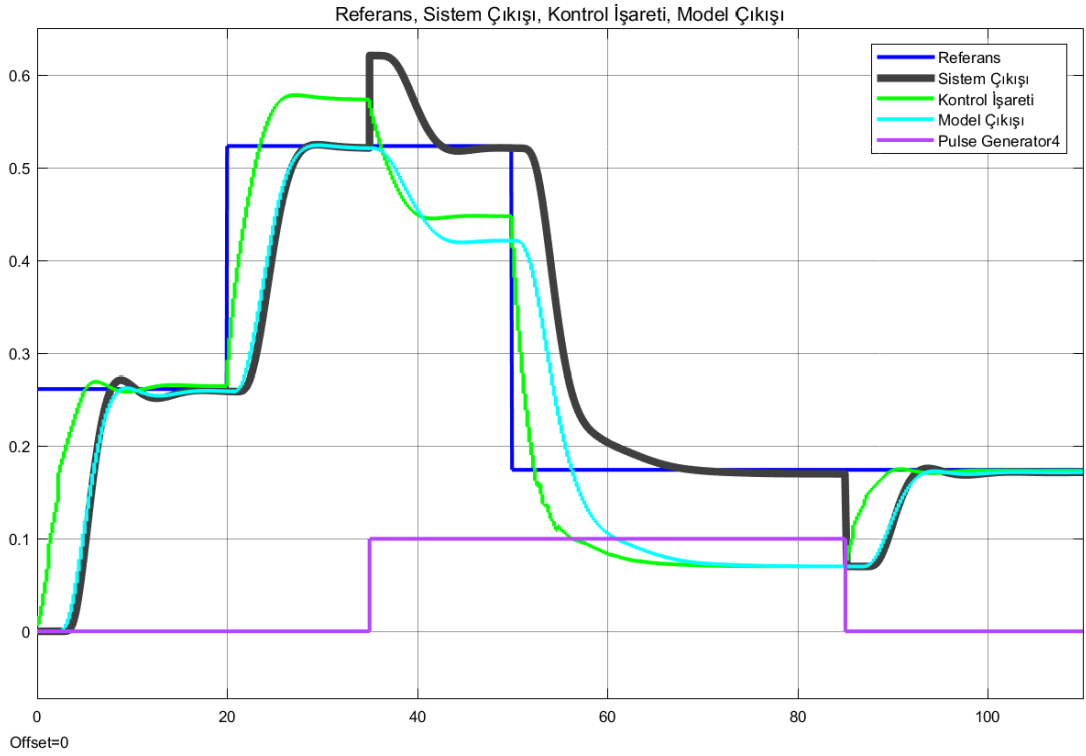
#### 4.2.2.1 Bozucuların giderilmesi

Sisteme dışarıdan uygulanan bozucular sistemin referans işareti ile sistem çıkışı arasında muhakkak bir değişime neden olacaktır. Klasik PID kontrolörde bu hata integratör ile giderilirken burada oluşan sistem çıkışı ile model çıkışı arasındaki hata Şekil 4.20’de olduğu gibi amaç fonksiyonunun içerisine eklenmektedir.

Bu sistem çıkışına uygulanan farklı gürültü yanıtları Şekil 4.23 ve Şekil 4.24’de verilmiştir.

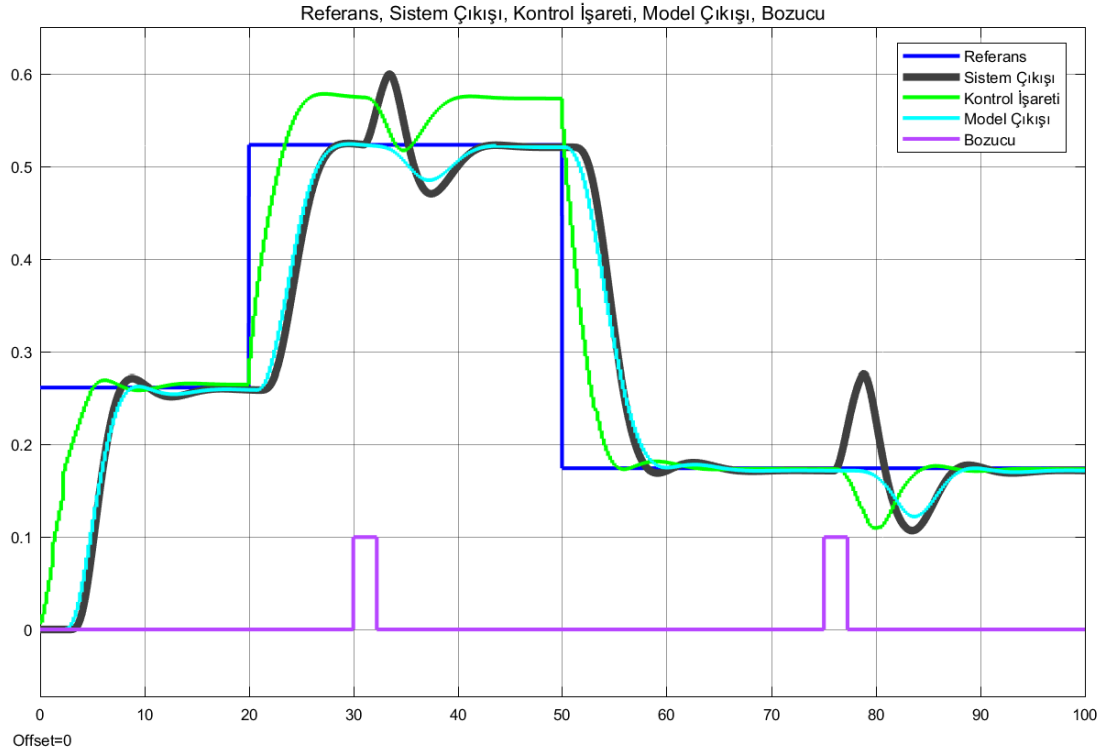


Şekil 4.23 : Çıkış Bozuculu Sistem Yanıtı

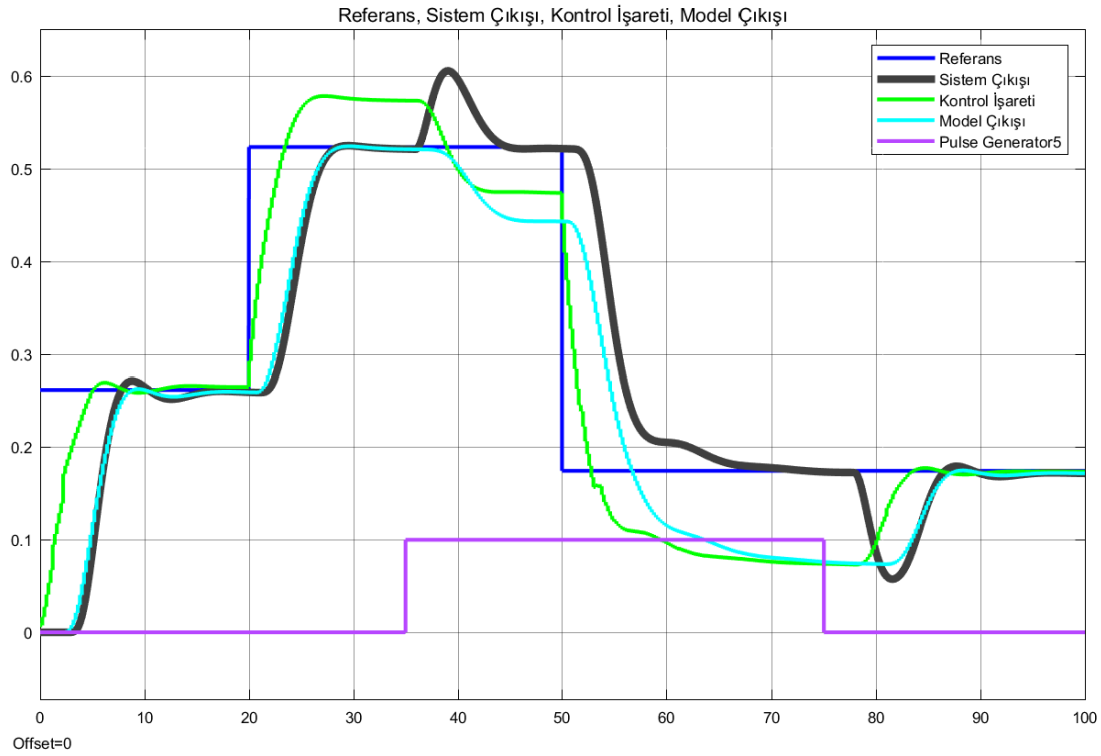


Şekil 4.24 : Çıkış Bozuculu Sistem Yanıtı

Sistemin girişten uygulanan farklı bozuculara karşı verdiği yanıtlar Şekil 4.25 ve Şekil 4.26’da verilmiştir.

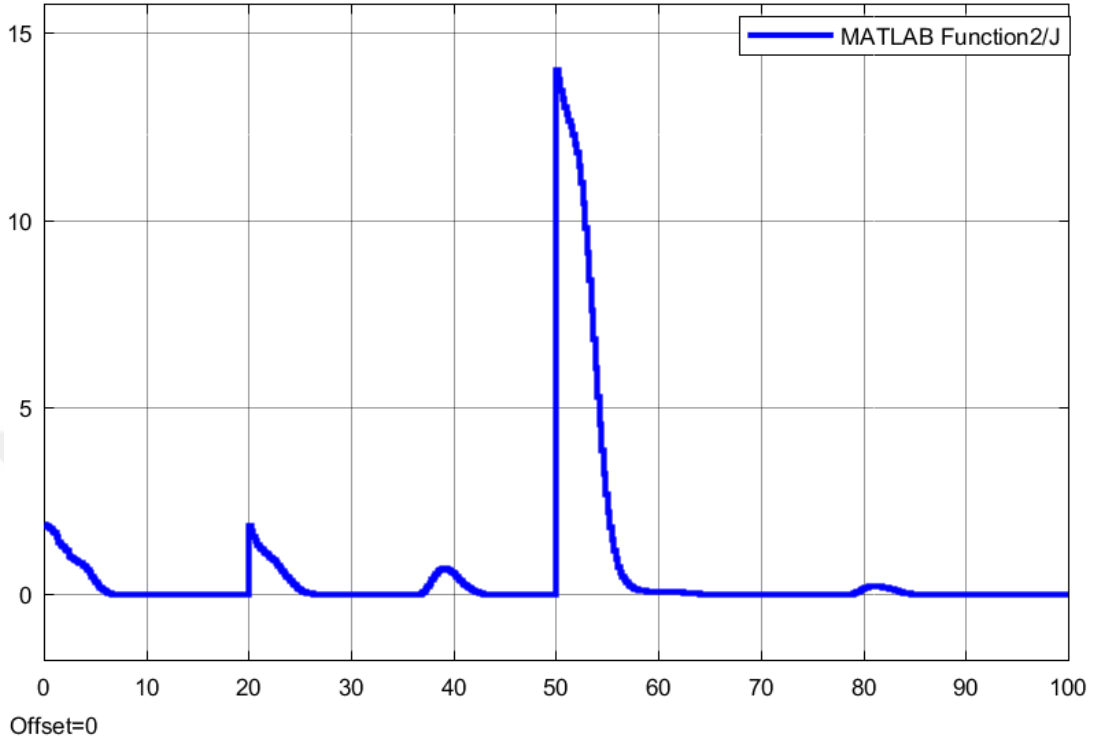


Şekil 4.25 : Giriş Bozuculu Sistem Yanıtı



Şekil 4.26 : Giriş Bozuculu Sistem Yanıtı

Şekil 4.26’da sistem cevabının oluşturmuş olduğu hata değeri yani minize edilmeye çalışılan amaç fonksiyonunu zamana göre grafiği Şekil 4.27’de verilmiştir.



Şekil 4.27 : Amaç fonksiyonunu Zamana Göre Değeri

### 4.3 NMPC'nin PLC Uygulaması

Doğrusal Olmayan Model Öngörülü Kontrol'ün PLC kullanarak Fan ve Levha sisteminin kontrol edilebilmesi için sistemin Matlab/Simulink içerisinde bulunan NMPC bloğunun PLC içerisinde oluşturulması gerekmektedir. Dolayısıyla PLC dışarıdan gelen veriler bu bloğa aktarılacaktır. Ardında bu blok vasıtasıyla amaç fonksiyonuna göre kontrol dizisi hesaplanacaktır. Hesaplanan kontrol dizisinin ilk elemanı sisteme uygulanarak sistemden tekrardan ölçümler yapıлып tekrardan bu işlemler yapılacaktır.

NMPC ile kontrol edilecek olan Fan ve Levha sistemi Siemens firmasının üretmiş olduğu TIA PORTAL V12 yazılımında kullanılarak gerçekleştirilecek ve uygulama bu program üzerinden yapılacaktır.

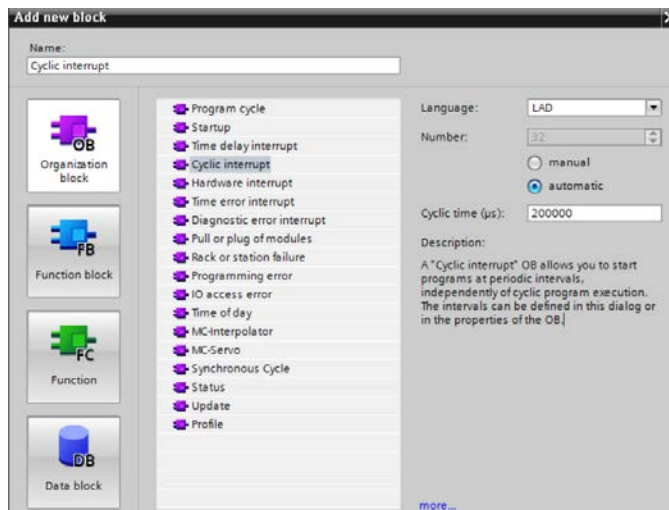
PLC yazılımları programlama teknikleri;

- Ladder Merdiven Diyagramı: En temel kontaktör mantığı olup kontaktör mantığı ile programlamayı sağlamaktadır.
- Fonksiyon Plan: Bu yapı grafik blokları ile tanımlanmış veri akışını gösteren mantık devrelerine dayanan şematik bir programlama dilidir. Bu şema içinde bulunan blokların içi yazılımcı tarafından oluşturulabildiği gibi hazır fonksiyonlarda kullanılabilir.
- Deyim Listesi (STL): Bu programlama dili assembly dili gibi bir programla dili olup takibi biraz zordur. Bunu yazılıma yatkın kişiler daha rahat kullanabilirler.

Bu çalışmada ise SCL programlama dili kullanılmış olup bu programlama dili STL, fonksiyon plan ve merdiven dilinin beraber kullanıldığı bir yapısal bir dildir. Bu dilin sağlamış olduğu karşılaştırma, matematiksel, döngüsel işlemlerinin kolaylığı nedeni ile genellikle Ladder diyagramı içerisinde bu dil fonksiyon blokları haline çağrılarak kullanıldı.

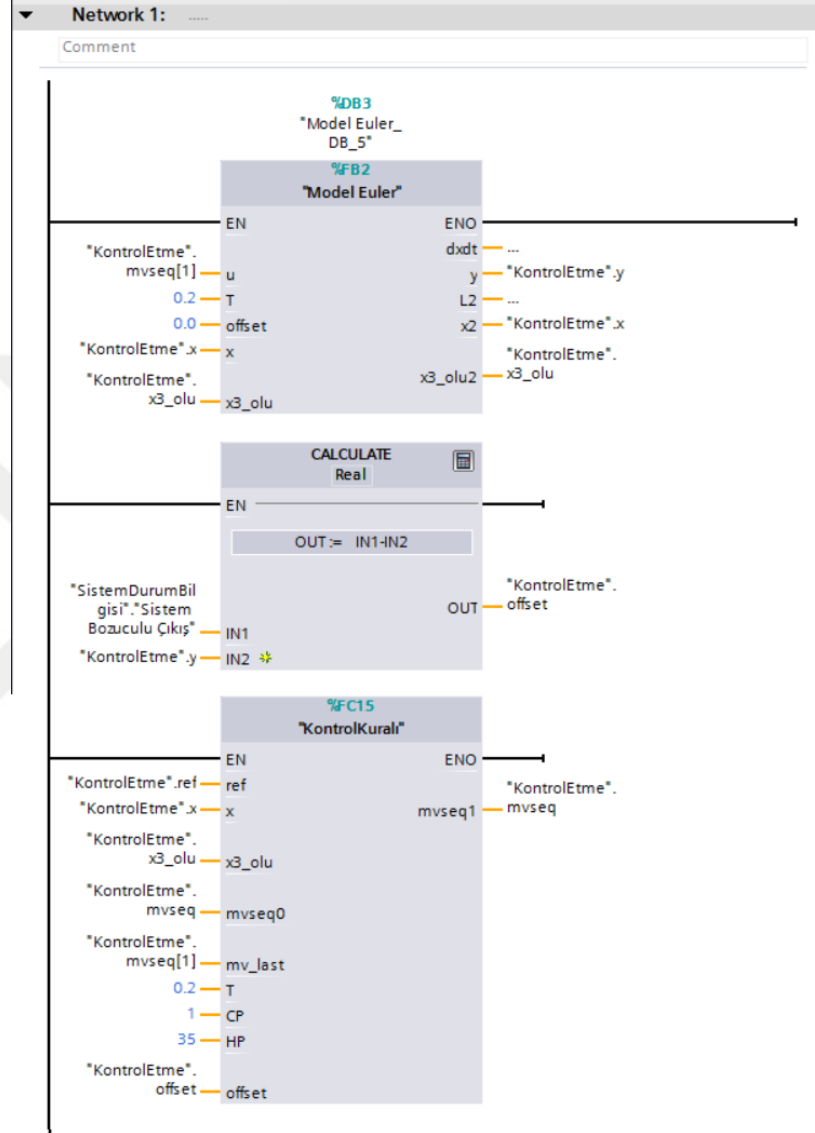
#### 4.3.1 PLC yazılımının oluşturulması

PLC’de seçilen örnekleme zamanına karşı düşen aralıklarla tekrarlı olarak koşturululan “Cyclic Interrupt” adı verilen Organizasyon blokları bulunmaktadır: Bu sayede gerçek zamanlı simülasyon yapmak mümkündür. böyle bir bloğun örnekleme zamanı belirtilerek oluşturulması Şekil 4.28’de gösterilmiştir.



Şekil 4.28 : Blok Ekleme

NMPC algoritması FC ve FB blokları halinde SCL dilinde oluşturulmuştur. Oluşturulan program blokları Şekil 4.31’de görüldüğü gibi “Cyclic Interrupt” bloğu içinden çağrılmıştır



Şekil 4.29 : Euler Model ve NMPC Blokları

Şekil 4.29’de verilen Euler Modelin içerisindeki kod Şekil 4.30’de verilmiştir.

```

1 #x_1:=#x; //Başlangıç Ataması yapılmakta
2 IF #x2[3]=0 THEN //Sıfıra bölüm engellenmektedir
3 #L:=(0.2+0.1*SIN(#x[1]))/(#x[3]+0.01);
4 ELSE
5 #L:=(0.2+0.1*SIN(#x[1]))/(#x[3]);
6 END_IF;
7 #L2:=REAL_TO_INT(MAX(IN1:=#L/#T, IN2:=1)); //Değişken dönüşümü yapılarak ayrık ölü zaman belirleniyor
8 #L2:=MIN(IN1:=#L2, IN2:=15);
9 #L2:=MAX(IN1:=#L2, IN2:=1);
10 #sayac:=0;
11 //Durumlara değer ataması yapılıyor
12 #dxdt[1]:=#x_1[2];
13 #dxdt[2]:=-37.62*SIN(#x_1[1])-6.58*#x_1[2]+0.218*COS(#x_1[1])*#x3_olu[#L2];
14 #dxdt[3]:=-0.108*#x_1[3]+0.0203*#u;
15 //Bir kaydırma işlemi yapılıyor
16 FOR #i := 2 TO 15 DO
17 // Statement section FOR
18 #x3_olu2[#i]:=#x3_olu[#i-1];
19 END_FOR;
20 #x3_olu2[1]:=#x2[3];
21 #x2[1]:=#x_1[1]+0.2*#dxdt[1]; //Durumlar güncelleniyor
22 #x2[2]:=#x_1[2]+0.2*#dxdt[2]; //Durumlar güncelleniyor
23 #x2[3]:=#x_1[3]+0.2*#dxdt[3]; //Durumlar güncelleniyor
24 #y:=#x2[1]; //Çıkış belirleniyor

```

**Şekil 4.30 : Euler Modelin SCL Kodu**

Modelden elde edilen çıkış ile gerçek sisteme arasındaki farka bakılıyor. Çünkü bir gerçek sisteme dışarıdan bozucu etki etti ise o zaman bu modelden çıkartarak bir “offset” değeri hesaplanmaktadır. Bu şekilde dışarıdan gelen bozucular bastırılmaktadır.

Kontrol işaretinin üretilebilmesi için bu Matlab/Simulink’de olduğu gibi Şekil 4.31’deki gibi Hessien ve gradyentin hesaplanması gerekmektedir. Bu değerleri FC bloğu olarak oluşturulduğu için Şekil 4.29’da bulunan Kontrol Kuralı fonksiyonunun içinde bir fonksiyon olarak çağırılmıştır. Bu fonksiyonlardan elde edilen verilere bağlı olarak çıkış güncellenmiştir. Burada alt fonksiyon olarak kullanılan Hessien ve Gradyan Matrislerinin alt fonksiyonlarının kodları EK A ve EK B’de verilmiştir. Bu fonksiyonların giriş ve çıkış verileri tanımlandıktan sonra bu kodlar bu fonksiyonlar kullanılabilir. Bu fonksiyonların giriş ve çıkış verileri tanımlandıktan sonra bu kodlar bu fonksiyonlar kullanılabilir.

```

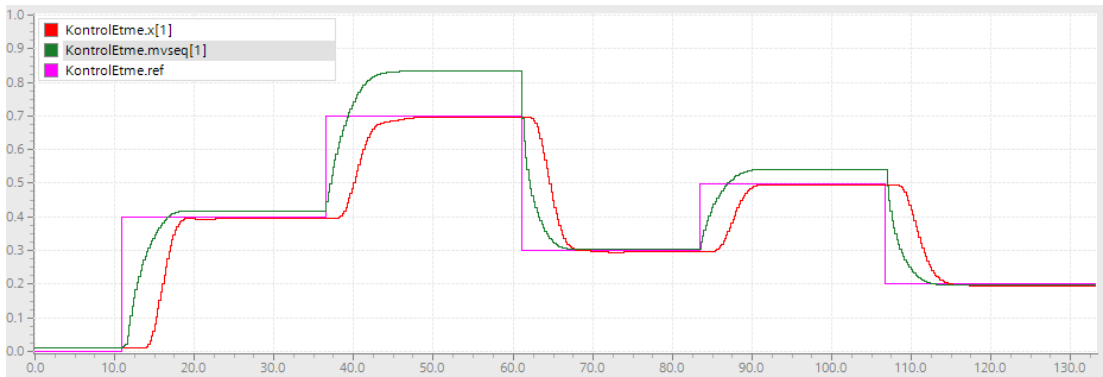
1 #mvseq_2[1]:=#mvseq0[1]; //Başlangıç değeri ataması yapılır
2 "HessienMatrisi"(ref:=#ref, //Hessien hesaplanır
3     x:=#x,
4     x3_olu:=#x3_olu,
5     mvseq:=#mvseq_2,
6     mv_last:=#mv_last,
7     T:=0.2,
8     CP:=3,
9     HP:=#HP,
10    offset:=#offset,
11    Hes=>#Hes);
12 "Gradyent_Hesaplama"(ref:=#ref, //Gradyent hesaplanır
13     x:=#x,
14     x3_olu:=#x3_olu,
15     mvseq:=#mvseq_2,
16     mv_last:=#mv_last,
17     T:=0.2,
18     CP:=3,
19     HP:=#HP,
20     offset:=#offset,
21     fx=>#fx);
22 IF #Hes[1,1]=0 THEN //Hessienin Tersisi hesaplanır
23 // Statement section IF
24 #THes[1,1]:=1/(#Hes[1,1]+0.001);
25 ELSE
26 #THes[1,1]:=1/#Hes[1,1];
27 END_IF;
28 #dx[1]:=#THes[1,1]*#fx[1]; //Adım uzunluğu hesaplanır
29 #mvseq_2[1]:=#mvseq_2[1]+#dx[1]; //Yeni kontrol işareti elde edilir
30 #mvseq1:=#mvseq_2; //Çıkış güncellenir

```

Şekil 4.31 : Kontrol İşareti Üret Fonksiyonu

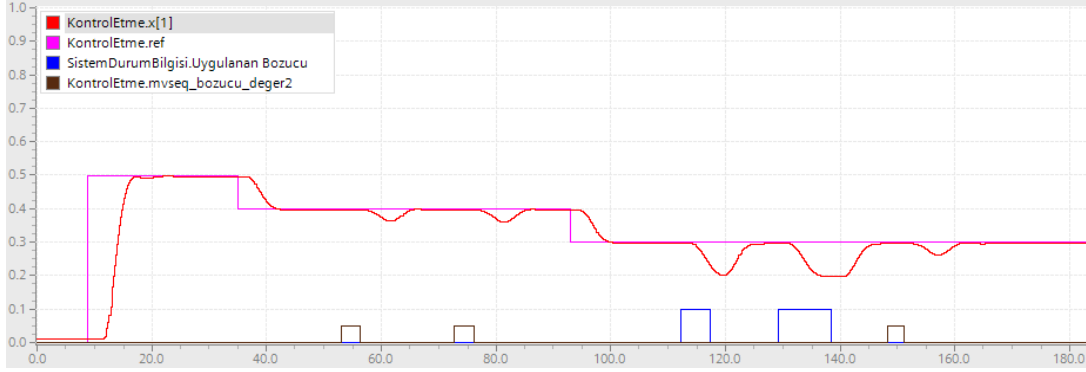
### 4.3.2 Fan and levha sisteminin plc de nmpc ile simulasyon sonuçları

TAI PORTAL V12 ile yazılan Ladder ve SCL kodu paket olarak PROFIBUS üzerinden PLC'ye yüklenmiştir. Örnekleme zamanı 0.2 sn seçilmiştir. Ardından sistem PLC üzerinden çalıştırıldığında Şekil 4.32'deki gibi bir cevap vermektedir.



Şekil 4.32 : Sistem Referans, Kontrol ve Çıkış İşaretleri

Şekil 4.33’de farklı referanslarda giriş ve çıkışlardan gürültüler uygulanmış ve sistem yanıtı yine referans değerine yaklaştığı görülmektedir. Şekil 4.34’de ise aynı uygulamanın kontrol işaretinin nasıl davrandığı gösterilmiştir.

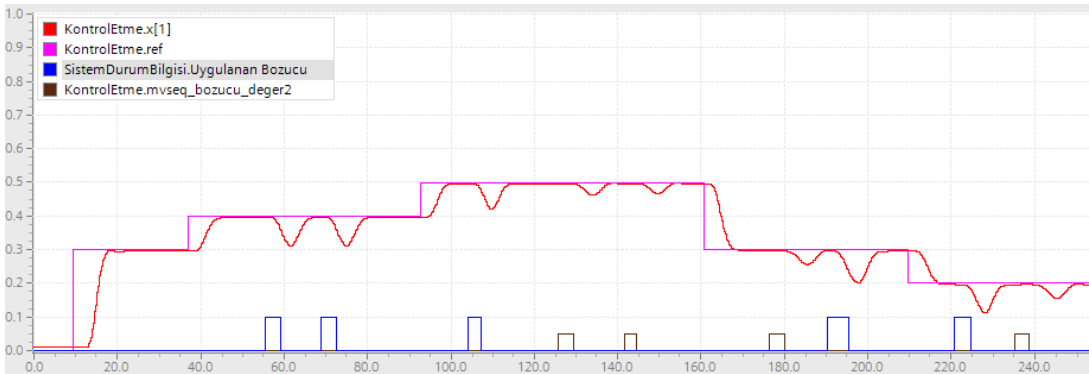


**Şekil 4.33 : Sistem Cevabı ve Referans İşareti ve Bozucular**

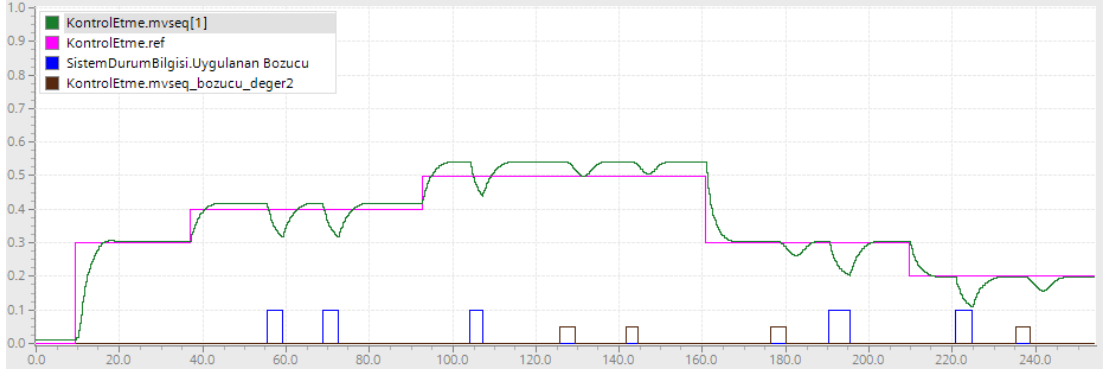


**Şekil 4.34 : Kontrol İşareti Referans İşareti ve Bozucular**

Şekil 4.35’de farklı referanslarda giriş ve çıkışlardan gürültüler uygulanmış ve sistem yanıtı yine referans değerine yaklaştığı görülmektedir. Şekil 4.36’da ise kontrol işaretinin davranışı vermiştir.

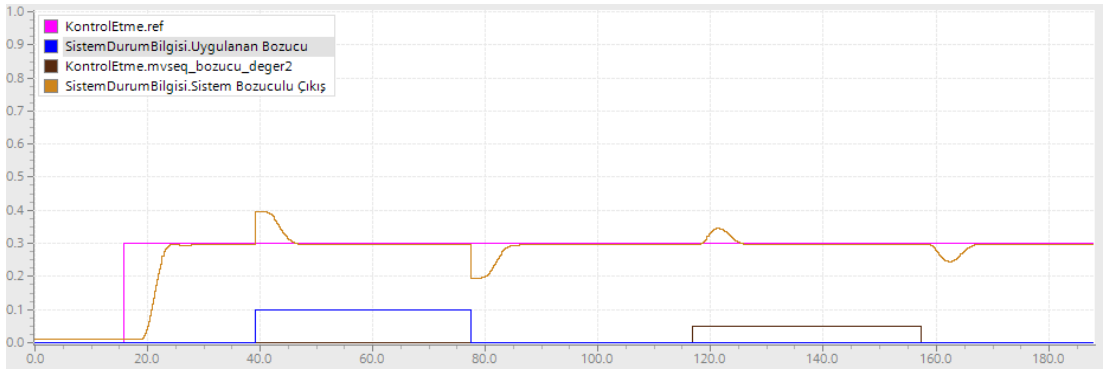


**Şekil 4.35 : Sistem Cevabı ve Referans İşareti ve Bozucular**

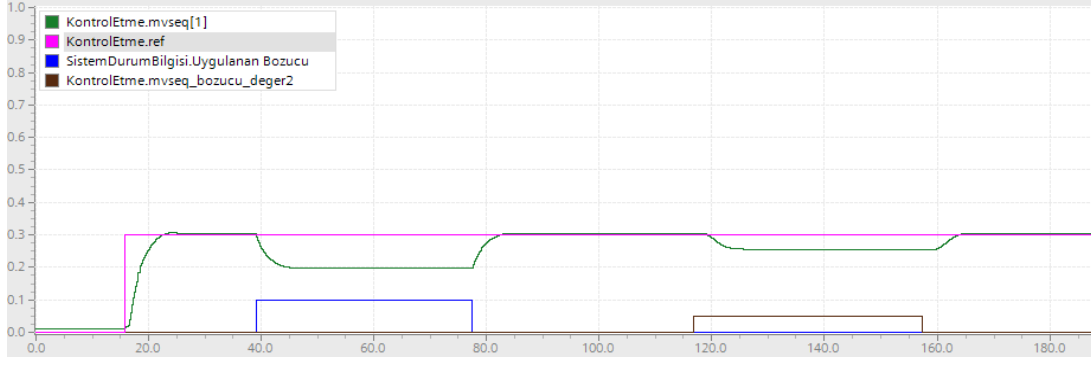


**Şekil 4.36 : Kontrol İşareti Referans İşareti ve Bozucular**

Şekil 4.37’de verilen sistem yanıtında sisteme girişinden ve çıkışından görüntüler uzun süreli uygulanmıştır. İlk olarak referans sinyali olan pembeyi turuncu olan sistem çıkışı takip etmektedir. Ardından sisteme 40 sn’den sonra sistem girişine mavi olan bozucusu uygulanmıştır. Şekil 4.38’de verilen aynı sistemin kontrol işareti verilmiştir. Sisteme bozucu geldiğinde NMPC bunu azaltacak şekilde kontrol işaretini düşürmüştür ve Şekil 4.37’de turuncu olan sistem çıkışı pembe olan referans değerine yaklaştırmıştır. Şekil 4.38’de de görüleceği üzere bozucu kalktığına da yeşil renkte olan kontrol işareti tekrardan yükselmiştir. Ardında 115 sn’den sonra sistem girişine bozucu uygulanmış ve yine NMPC bunu engelleyecek şekilde kontrol sinyalini düşürmüştür ve sistemi istenilen referans işaretine getirmiştir. Bozucu kalktığına da tekrardan kontrol işareti yükselmiş ve sistem istenilen referansa geldiği görülmüştür.

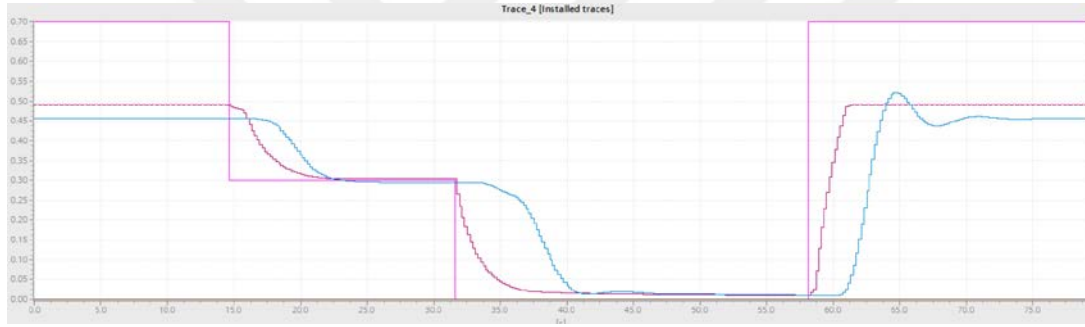


**Şekil 4.37 : Sistem Cevabı ve Referans İşareti ve Bozucular**



**Şekil 4.38 : Kontrol İşareti Referans İşareti ve Bozucular**

Kontrol işareti 0.5 ile sınırlandırıldığında ise sistemin çıkışı Şekil 4.39’da verilmiştir. Referans işareti bu değerden yüksek olduğunda kontrol işareti bu değeri aşamadığı için sistem yanıtı bu değer altında kaldığı görülmüştür.

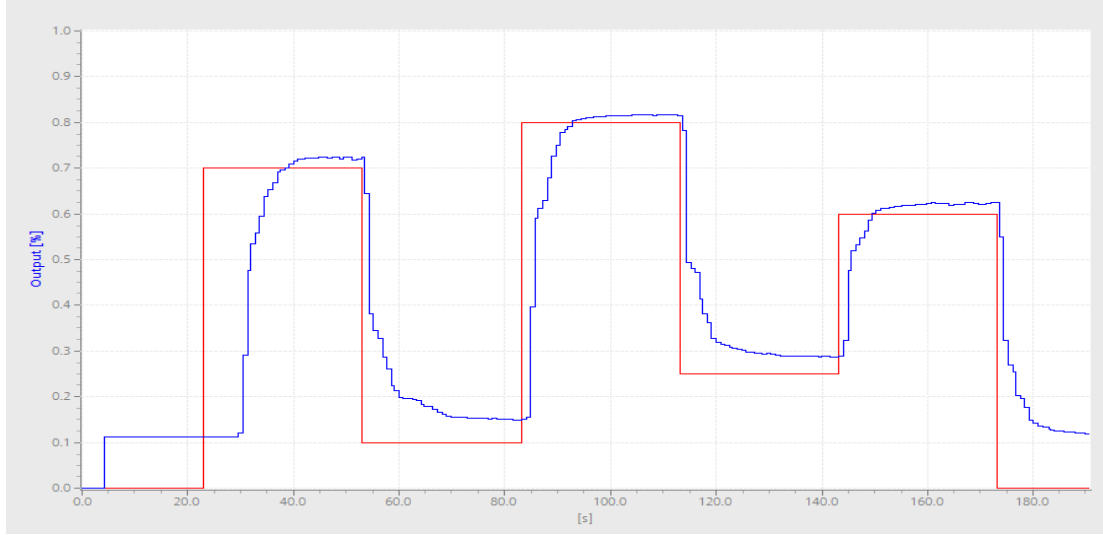


**Şekil 4.39 : Sınırlanmış Kontrol İşareti**

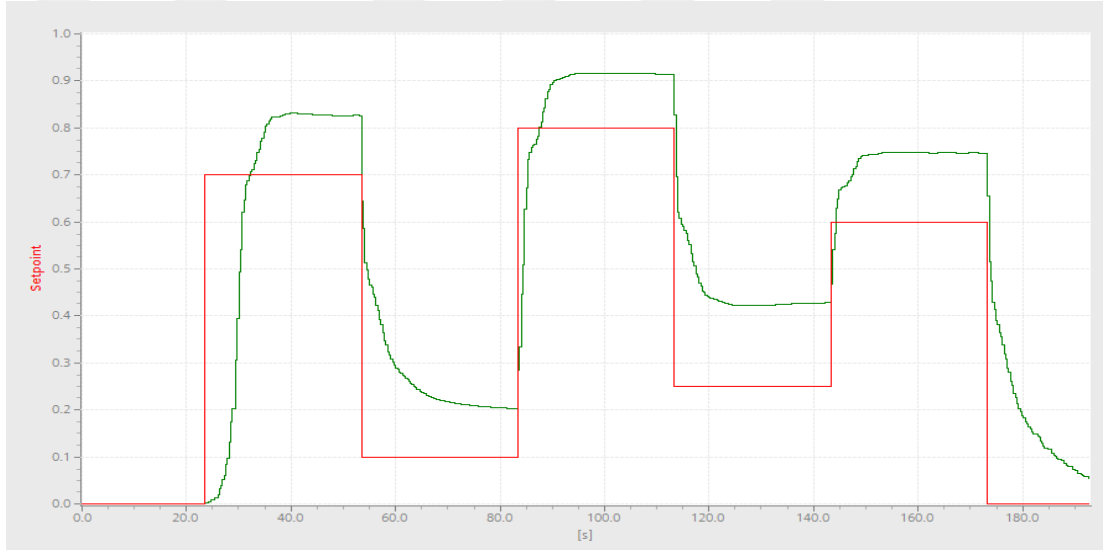
### 4.3.3 Fan and levha sisteminin plc de nmpc ile uygulama sonuçları

TAI PORTAL V12 ile yazılan Ladder ve SCL kodu PROFIBUS üzerinden Siemens S71200 PLC’ye yüklenmiştir. Ardından Şekil 4.40’deki gibi sisteme farklı referans değerleri verilmiş ve NMPC denetleyicisi ölçüme ve modele bağlı olarak Şekil 4.41’deki gibi kontrol işareti üretmiştir.

Sistem yanıtına her bölgede hemen hemen aynı performansı sergilediği görülmektedir. Fakat cevabında sürekli durum hatasının olduğu görülmektedir. Bunun nedeni ise modele dahil edilemeyen kullanılan ekipmanlardan dolayı sistemde süreksizliğin veya nonlineerliğin olduğunu göstermektedir. Eğer sistem ile model moda mod eşleşmiş olsaydı simülasyonlardaki gibi daha az kararlı durum hatası çıkmış olacaktı.



Şekil 4.40 : PLC ile NMPC Uygulaması Referans ve Sistem Cevabı



Şekil 4.41 : PLC ile NMPC Uygulaması Referans ve Kontrol İşareti



## 5 SONUÇLAR VE ÖNERİLERİ

Bu tezde NMPC'nin temel prensipleri, temel bileşenleri ve NMPC'de kullanılabilecek olan farklı teknikler araştırılmış ve incelenmiştir. Temelde NMPC teknikleri ardışıl veya anlık çözümleme teknikleri olarak farklılık göstermektedir. Sistemin ayrıklaştırılması için ise Taylor Serisinden elde edilmiş olan Euler veya Runge Kutta gibi metodlardan biri kullanılabilir. Ayrıca doğrusal ve doğrusal olmayan eşitlik-eşitsizlik kısıtlarına sahip, bu doğrusal olmayan optimizasyon probleminin çözümü için SQP, GRG, Interior Point, Exterior Point, Primal-Dual gibi farklı çözüm metodları bulunmaktadır.

Tezde çalışmada endüstriyel kontrol uygulamaları için giriş-çıkış arabirimleri ve işletim sistemleri özelleşmiş PLC'lerde uygulanabilirliği araştırılmıştır. Bu amaçla test sistemi olarak havanın dinamiği ve ters sarkaç özelliğindeki levha askı sisteminin dinamiği gereği ağır nonlineerliği olan ve değişken ölü-zamanı bulunan Fan ve Levha düzeneğinin NMPC ile kontrolü problemi ele alınmıştır. Belirtmek gerekir ki PLC'ler MATLAB simülasyonlarının yapıldığı güncel yüksek hızlı kişisel bilgisayarlara göre daha yavaş kod işlemektedirler ve türev alma gibi bazı matematik fonksiyonları bulunmamaktadır.

Öncelikle PLC'de de gerçekleştirilmesine uygun olarak seçilen sayısal optimizasyon algoritması yazımı ve öngörülerin inşası Matlab/simulinkte gerçekleştirilmiştir. Matlab'da bulunan hazır MPC fonksiyonları ve karesel programlama çözümleri kullanılmamıştır. Sayısal optimizasyon yöntemi olarak SQP algoritması seçilmiştir. Gradyan ve Hessian hesapları için matematik literatürde bulunan eşitlik (4.14) ve eşitlik 4.16'da verilen yaklaşımlar kullanılmıştır. Öngörülerini oluşturmak için kullanılan doğrusal olmayan modelin ayrıklaştırılmasında basitliği nedeniyle Euler method kullanılmıştır. Ölçülemeyen durumlar sistem modelinden elde edilmiştir. Kısıtlar "Interior Penalty" metodu kullanılarak problem dahil edilmiştir.

Yapılan simülasyonlar da sabit ağırlıklı ölçüt kullanıldığında aşım ve sürekli hal hatası bulunan sonuçlar elde edilmiştir. Bu sorunlar sürekli hal hatası ve aşım ilgili

durumlarda ağırlıklar değiştirilerek çözülmüştür, yani sistemin yanıtının değişimi boyunca duruma göre ağırlıklar değiştirilmiştir

Simülasyonlar önce ölü zaman içermeyen model kullanılarak ölü zaman içermeyen system üzerinde gerçekleştirilmiştir. Her çalışma noktasında aynı kapalı-çevrim davranışı sergileyen, yeterli hızda, aşısız ve sürekli hal hatasız sonuçlar elde edilmiştir. Ancak ölü-zaman içermeyen model kullanılarak ölü-zaman içeren system üzerinde yapılan simülasyonlarda, kapalı çevrimin farklı çalışma noktalarında farklı ve aşımli davranış sergilediği görülmüştür. Değişken ölü-zamanın modelle dahil edilmesi gerekliliği görülmüştür. Değişken ölü zaman her bir öngörü durumu için eşitlik (4.1) kullanılarak hesaplanmıştır. Kontrol işaretinin önceki değerleri bir dizide saklanarak hesaplanan ölü zamana göre gereken önceki değeri bu dizi üzerinden çekilip modele uygulanarak öngörülerin elde edilmesi ile ölü zamanın problemi ortadan kaldırılmıştır.

Simülasyonlarda giriş ve çıkış bozucularının etkisi de incelenmiştir. NMPC'nin yalın haliyle bozucu bastırma özelliğine sahip olmadığı görülmüştür. Bozucu etkiyi giderebilmek için sistem çıkışı ile öngürüler arasındaki fark göz önünde bulundurularak ölçüt fonksiyonunda uygun bir değişiklik yapılmıştır ve başarılı sonuçlar elde edilmiştir.

PLC'de uygulaması TIA PORTAL V12 program geliştirme ortamında gerçekleştirilmiştir. Algoritmaların yazımında Ladder ve SCL dilleri kullanılmıştır. Algoritmalar basitleştirilerek gerçeklenmeye çalışılsa da NMPC işlem yükü fazla olan bir yapıya sahip olduğundan PLC uygulamasında bazı sıkıntılar yaşanmıştır. PLC için kontrol ufku arttıkça tersi alınan Hessien Matrisin boyutu arttığı için PLC çevrimi bu işlem örnekleme zamanını olan 0,2 sn'yi aştığı görüldü. Bu nedenle 1 kontrol ufku ve 35 öngörü ufku seçilmiştir. Bununla birlikte yine de aşısız, yeterince hızlı ve sürekli hal hatasız sonuçlar elde edilmiştir. Algoritmanın başarı ile sistemi kısıtlar içinde tuttuğu da görülmüştür.

Ayrıca NMPC denetleyicisi S71200 PLC'sine yüklenerek gerçek sistem üzerinden çalışılmıştır. Sistemde modellenemeyen nonlineerlikler olduğundan uygulamada sürekli hal hatası oluştuğu görülmüştür. Bu ise kontrolöre integratör eklenerek çözümlenebilir. Bu ise bir sonraki çalışmamızda gerçekleştirilecektir. Bu çalışmada ise

yalın Nonlinear MPC'nin uygulanabilirliđi arařtırılmıř ve her blgede aynı performans elde edilmiřtir.

Bu alıřmada en uygun Q ve R'nin belirlenmesi iin bir yntem kullanılmamıřtır. Ama fonksiyonundanki Q ve R kazanlarının belirlenmesinde sezgisel algoritmalar kullanılabilir. Bu alıřma da ıkıř hatasını da deđerlendiren durum kestirimi yapılmamıřtır, daha dođru sonular elde etmek iin Lumberger veya Kalman kestiricileri kullanılabilir.

te yandan, optimizasyon iin Newton tipi algoritmaların yerine sezgisel algoritmalar da kullanılabilir. PLC'de gerekleme karmařıklıđı tam kestirilemekle birlikte matematiksel modelle ngr yapmak yerine verilere bađımlı olan yapay sinir ađı, bulanık model gibi farklı modeller kullanılabilir ve bařarımları test edilebilir.



## KAYNAKLAR

- Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., & Wright, S. J.** (1999). Nonlinear predictive control and moving horizon estimation. *Springer*, 391-449.
- Arifoğlu, U.** (2005). *Matlab 7.04 Simulink ve mühendislik uygulamaları*: Alfa Yayınları.
- Bahtiyar, B.** (2015). *FPGA ile gerçek-zamanlı runge-kutta model öngörülü kontrol*, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Pamukkale FBE.
- Bellman, R.** (1957). Dynamic Programming [Press release]
- Bitmead, R. R., Gevers, M., & Wertz, V.** (1993). Adaptive Optimal Control – The Thinking Man’s GPC. *Automatica*, 29(3), 798-800.
- Camacho, E. F., & Bordons, C.** (2004). *Model Predictive Control Second Edition*.
- Camacho, E. F., & Bordons, C.** (2007a). Nonlinear Model Predictive Control: An Introductory Review. *Assessment and future directions of nonlinear model, 2007 - Springer*.
- Camacho, E. F., & Bordons, C.** (2007b). Nonlinear model predictive control: An introductory review. *Assessment and future directions of nonlinear model predictive control*, 1-16.
- Camacho, P. E. F.** (2007). *Model Predictive Control Second Edition*. Springer.
- Caruntu, C. F., & Lazar, C.** (2011). *IEEE International Conference on Mechatronics*. Paper presented at the Stabilizing MPC for Network-controlled Systems With An Application to DC Motors, Istanbul.
- Chen, H., & Allgöwer, F.** (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205-1218.
- Clarke, D. W., Mohtadi, C., & Tuffs, P. S.** (1987a). Generalized Predictive Control-Part I: The Basic Algorithm *Automatica*, 23(2), 137-148.
- Clarke, D. W., Mohtadi, C., & Tuffs, P. S.** (1987b). Generalized Predictive Control-Part II: Extensions and Interpretations *Automatica*, 146-160.
- Dincel, E., Yalçın, Y., & Kurtulan, S.** (2014). *A new approach on angular position control of fan and plate system*. Paper presented at the Control, Decision and Information Technologies (CoDIT), 2014 International Conference on.
- Findeisen, R., & Allgöwer, F.** (2002). *An introduction to nonlinear model predictive control*. Paper presented at the 21st Benelux Meeting on Systems and Control.
- Foss, B. A., Johansen, T. A., & Sørensen, A. V.** (1995). Nonlinear predictive control using local models applied to a batch fermentation process. *Control Engineering Practice*, 289-293.
- Holkar, K. S., & Waghmare, L. M.** (2010). An Overview of Model Predictive Control. *International Journal of Control and Automation*.

- Hredzak, B., Jang, M., & Agelidis, G. V.** (2013). Model Predictive Control Systems for a Hybrid Battery-Ultracapacitor Power Source. *IEEE Transactions on Power Electronics*, 29(3), 1469-1479.
- Kaya, A. Ş.** (2007). *On the Optimal Operation of the Open Plate Reactor.*, Dalarna University.
- Keerthi, S. S., & Gilbert, E. G.** (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2), 265-293.
- Koerber, A., & King, R.** (2013). Combined Feedback-Feedforward Control of Wind Turbines Using State-Constrained Model Predictive Control. *IEEE Transactions on Control System Technology*, 21(4), 1117-1128.
- Kothare, M. V., Balakrishnan, V., & Morari, M.** (1996). Robust Constrained Model Predictive Control using Linear Matrix Inequalities. *Automatica*, 1361-1379.
- Kömürcü, E.** (2009). *Bulanık Model Öngörülü Kontrol*. (Yüksek Lisans Tezi), İTÜ FBE.
- Kruger, K.** (2004). Optimization of boiler start-up using a nonlinear boiler model and hard constraints. *Energy*, 29(12-15), 2239-2251. doi:10.1016/j.energy.2004.03.022
- Kuo, L. E., & Melsheimer, S. S.** (1998). *Wastewater Neutralisation Control Using a Neural Network Based Model Predictive Controller*. Paper presented at the American Control Conference, Philadelphia.
- Lazar, M., Heemels, W. P. M. H., Roset, B. J. P., Nijmeijer, H., & van den Bosch, P. P. J.** (2008). Input-to-state stabilizing sub-optimal NMPC with an application to DC-DC converters. *International Journal of Robust and Nonlinear Control*, 18(8), 890-904. doi:10.1002/rnc.1249
- Liu, J., Huang, Z., Liu, W., Yang, Y., & Tong, H.** (2008). *Locomotive Brake Control Method based on T-S Fuzzy Modeling Predictive Control*. Paper presented at the 2nd International Symposium on Intelligent Information Technology Application, Shanghai.
- Ma, Y., Borrelli, F., Hancey, B., Coffey, B., Bengea, S., & Haves, P.** (2012). Predictive Control for the Operation of Building Cooling Systems. *IEEE Transactions on Control Systems Technology*, 20(3), 797-803.
- Martinsen, F., Biegler, L. T., & Foss, B. A.** (2002). Application of optimization algorithms to nonlinear MPC. *IFAC Proceedings Volumes*, 35(1), 369-374.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. M.** (2000). Constrained model predictive control: stability and optimality. *Automatica*, 26(6), 789-814.
- Mills, A., Wills, A., & Ninness, B.** (2009). *Nonlinear Model Predictive Control of an Inverted Pendulum*. Paper presented at the American Control Conference, 2009. ACC'09.
- Morari, M., & Lee, J. H.** (1999). Model Predictive Control: Past, Present and Future. *Computer and Chemical Engineering*, 667-682.
- Morari, M., Lee, J. H., & Garcia, C. E.** (1994). State-Space Interpretation of Model Predictive Control. *Automatica*, 707-717.
- Moritz Diehl, Hans Joachim Ferreau, & Haverbeke, N.** (2009). Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. 391-417.
- Nise, N. S.** (2011). *Control Systems Engineering* (6th ed.): John Wiley & Sons.

- Rao, S. S.** (2009a). *Engineering Optimization Theory 4th Edition*: John Wiley & Sons.
- Rao, S. S.** (2009b). *Engineering optimization: theory and practice*: John Wiley & Sons.
- Richalet, J.** (1993). Industrial Application of Model Based Predictive Control. *Automatica*, 29(5), 1251-1274.
- Richalet, J., Rault, A., Testud, J. L., & Papon, J.** (1976). Algorithmic control of industrial processes.
- Rien Quirynen, Milan Vukov, & Diehl, M.** (2015). Multiple Shooting in a Microsecond *Multiple Shooting and Time Domain Decomposition Methods* (pp. 183-201): Springer.
- Rolf Findeisen, & Frank Allgöwer.** (2002). *An Introduction to Nonlinear Model Predictive Control*. Paper presented at the 21st Benelux Meeting on Systems and Control.
- Rossiter, J. A.** (2004). *Model-Based Predictive Control A Practical Approach*.
- Scokaert, P. O. M., Mayne, D. Q., & Rawlings, J. B.** (1999). Suboptimal Model Predictive Control (Feasibility Implies Stability). *IEEE Transactions on Automatic Control* 44(3):648 - 654 · April 1999.
- Seyab, R. K. A.** (2006). *Nonlinear Model Predictive Control Using Automatic Differentiation*. (Doktora Tezi), CRANFIELD UNIVERSITY.
- Sharma, V.** (2007). *Development of Control Lab Interface for Data Acquisition using Lab VIEW*. Retrieved from Stockholm, Sweden:
- Thomas, J. W.** (2013). *Numerical partial differential equations: finite difference methods* (Vol. 22): Springer Science & Business Media.
- Wright, S. J.** (1997). *Primal-dual interior-point methods*: SIAM.
- Xie, Y., Ghaemi, R., Sun, J., & Freudenberg, J. S.** (2009). Implicit Model Predictive Control of a Full Bridge DC-DC Converters. *IEEE Transactions on Power Electronics*, 24(12), 2704-2713.
- Yalçın, Y.** (2003). *Modele Dayalı Öngörülü Endüstriyel Kontrollerin Gerçeklenmesi*. (Yüksek Lisans), İTÜ, İTÜ FBE.



## **EKLER**

**EK A:** Hessien Matrisin SCL Kodu

**EK B:** Gradient Matrisin SCL Kodu



## EK A

//Hessien Matrisinin Elde Edilmesi

```
#h1:=0.01;
#h2:=0.01;
#a1:=0;
#a2:=0;
#a3:=0;
#a4:=0;
#i:=1;
#j:=1;
#x0:=#mvseq;
#x0[#i]:=#mvseq[#i]+#h1;
#x0[#j]:=#x0[#j]+#h2;
"Amac Fonksiyonu"(ref:=#ref,
  x:=#x,
  x3_olu:=#x3_olu,
  mvseq:=#x0,
  mv_last:=#mv_last,
  T:=#T,
  CP:=#CP,
  HP:=#HP,
  offset:=#offset,
  J=>#a1,
  y=>#y);
#x0:=#mvseq;
#x0[#i]:=#mvseq[#i]+#h1;
#x0[#j]:=#x0[#j]-#h2;
"Amac Fonksiyonu"(ref:=#ref,
  x:=#x,
  x3_olu:=#x3_olu,
  mvseq:=#x0,
  mv_last:=#mv_last,
  T:=#T,
  CP:=#CP,
  HP:=#HP,
  offset:=#offset,
  J=>#a2,
  y=>#y);
#x0:=#mvseq;
#x0[#i]:=#mvseq[#i]-#h1;
#x0[#j]:=#x0[#j]+#h2;
"Amac Fonksiyonu"(ref:=#ref,
  x:=#x,
  x3_olu:=#x3_olu,
  mvseq:=#x0,
  mv_last:=#mv_last,
  T:=#T,
  CP:=#CP,
  HP:=#HP,
  offset:=#offset,
  J=>#a3,
  y=>#y);
#x0:=#mvseq;
```

```

#x0[#i]:=#mvseq[#i]-#h1;
#x0[#j]:=#x0[#j]-#h2;
"Amac Fonksiyonu"(ref:=#ref,
    x:=#x,
    x3_olu:=#x3_olu,
    mvseq:=#x0,
    mv_last:=#mv_last,
    T:=#T,
    CP:=#CP,
    HP:=#HP,
    offset:=#offset,
    J=>#a4,
    y=>#y);
#Hes[#i,#j]:=(#a1-#a2-#a3+#a4)/(4*#h1*#h2);

```

## EK B

```

#Eps:=0.01;
#fx[1]:=0;
#i:=1;
#mvseq_gec:=#mvseq;
#mvseq_gec2:=#mvseq;
#mvseq_gec2[#i]:=#mvseq[#i]+#Eps;
"Amac Fonksiyonu"(ref:=#ref,
    x:=#x,
    x3_olu:=#x3_olu,
    mvseq:=#mvseq_gec2,
    mv_last:=#mv_last,
    T:=#T,
    CP:=#CP,
    HP:=#HP,
    offset:=#offset,
    J=>#j,
    y=>#y);

"Amac Fonksiyonu"(ref:=#ref,
    x:=#x,
    x3_olu:=#x3_olu,
    mvseq:=#mvseq_gec,
    mv_last:=#mv_last,
    T:=#T,
    CP:=#CP,
    HP:=#HP,
    offset:=#offset,

```

```
J=>#j_1,  
y=>#y_1);  
#fx[#i]:=(#j-#j_1)/0.01;
```



## ÖZGEÇMİŞ

**Ad-Soyad** : Murat Erhan ÇİMEN

**Doğum Tarihi ve Yeri** : 04.06.1992 Bakırköy

**E-posta** :

- muraterhan92@gmail.com
- muratcimen@sakarya.edu.tr

### ÖĞRENİM DURUMU:

• **Lise** : Bağcılar Teknik Lisesi, Makine Teknolojisi Bölümü

- Okul Birincisi

• **Lisans** :

- 2014, Sakarya Üniversite, Teknoloji Fakültesi, Mekatronik Mühendisliği 3.9/4
  - Bölüm Birincisi
  - Fakülte Birincisi
  - Üniversite İkincisi

➤ 2015, Sakarya Üniversite, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği 3.71/4

• **Yükseklisans** :

- 2018, İstanbul Teknik Üniversitesi, Fen Bilimleri, Kontrol ve Otomasyon Mühendisliği
- 2018, Sakarya Üniversitesi, Fen Bilimleri, Elektrik-Elektronik Mühendisliği

### MESLEKİ DENEYİM VE ÖDÜLLER:

- Ömer Halisdemir Üniversitesi Mekatronik Müh. Bölümü Arş Gör. olarak çalıştı.
- Sakarya Üniversitesi Elektrik Elektronik Müh. Bölümü Arş Gör. olarak çalışmaya devam etmektedir.

