

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**LİNEER OLMAYAN SİSTEMLERE KUADRATİK OPTİMAL KONTROL
YÖNTEMLERİ İLE MODEL ÖNGÖRÜLÜ KONTROL
UYGULANMASI ve KARŞILAŞTIRILMASI**

**YÜKSEK LİSANS TEZİ
Turgay ERAY**

Anabilim Dalı : Makina Mühendisliği

Programı : Sistem Dinamiği ve Kontrol

HAZİRAN 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**LİNEER OLMAYAN SİSTEMLERE KUADRATİK OPTİMAL KONTROL
YÖNTEMLERİ İLE MODEL ÖNGÖRÜLÜ KONTROL
UYGULANMASI ve KARŞILAŞTIRILMASI**

**YÜKSEK LİSANS TEZİ
Turgay ERAY
(503081618)**

Tezin Enstitüye Verildiği Tarih : 06 Mayıs 2011

Tezin Savunulduğu Tarih : 08 Haziran 2011

**Tez Danışmanı : Prof. Dr. Can ÖZSOY (İTÜ)
Diğer Jüri Üyeleri : Yrd. Doç. Dr. Ayhan KURAL (İTÜ)
Yrd. Doç. Dr. Cüneyt FETVACI (İÜ)**

HAZİRAN 2011

Anneme ve Babama,

ÖNSÖZ

Bu çalışmanın uygulamaya yarar getirmesi amacıyla bilgi ve tecrübeleriyle çalışmama ışık tutan ve değerli yardım ve katkılarıyla beni yönlendiren Sayın Hocam ve tez danışmanım Prof. Dr. Can Özsoy'a teşekkürlerimi sunarım.

Tez aşamasında her sorularıma ışık tutan arkadaşlarıma da teşekkürü bir borç bilirim.

Son olarak her türlü özveriye göstererek bugünlere gelmemi sağlayan, benden desteklerini ve sevgilerini hiç eksik etmeyen aileme şükranlarımı sunarım.

Haziran 2011

Turgay ERAY
(Makina Mühendisi)

İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ.....	v
İÇİNDEKİLER.....	vii
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY.....	xviii
1. GİRİŞ	1
1.1 Amaç ve Kapsam.....	2
1.2 Tezin İçeriği.....	2
2. NONLİNEER OPTİMİZASYON.....	3
2.1 Kısıtsız Optimizasyon.....	3
2.1.1 Lokal ve Global minimum.....	3
2.1.2 Optimal için gerekli koşullar.....	4
2.1.3 Optimal için yeterli koşullar.....	5
2.1.4 Kuadratik amaç fonksiyonu.....	6
2.1.5 Kısıtsız optimizasyon algoritmaları.....	7
2.1.5.1 Azalan yön ve adım büyüklüğü.....	7
2.1.5.2 Gradient metodu.....	8
2.1.5.3 Azalma yönü seçimi.....	10
2.1.5.4 Zigzag azalma (steepest descent) metodu.....	10
2.1.5.5 Newton metodu.....	11
2.1.5.6 Modifiye edilmiş Newton metodu.....	12
2.1.6 Adım boyutu seçimi.....	12
2.1.6.1 Minimize kuralı.....	12
2.1.6.2 Limitli minimize kuralı.....	12
2.1.6.3 Armijo kuralı.....	13
2.1.6.4 Goldstein kuralı.....	13
2.1.6.5 Sabit adım büyüklüğü.....	13
2.1.6.6 Azalan adım büyüklüğü.....	14
2.1.7 Newton metodu.....	14
2.1.7.1 Modifiye edilmiş Cholesky çarpanlarına ayırma.....	15
2.1.7.2 Güvenilir alan metodu.....	17
2.1.8. Eşlenik yön metodu.....	18
2.1.8.1 Eşlenik gradient metodu.....	19
2.1.8.2 Önkoşullu eşlenik gradient metodu.....	20
2.1.9 Quasi-Newton metodu.....	21
2.1.10 Ayırık-zamanlı optimal kontrol problemi.....	24
2.1.10.1 İlk mertebe gerekli optimal koşulu.....	25
2.1.10.2 Optimal kontrol için gradient metodu.....	25
2.1.10.3 Optimal kontrol için eşlenik gradient metodu.....	26
2.1.10.4 Optimal kontrol için Newton metodu.....	26
2.1.10.5 Optimal kontrol problemi, tek link manipulatör.....	26

Benzetim sonuçları.....	29
2.2 Kısıtlı Optimizasyon	31
2.2.1 Eşitlik kısıtı.....	31
2.2.1.1 Optimal koşulu	32
2.2.1.2 Uygun yön ve ikinci merteye koşulu.....	33
2.2.1.3 Kuadratik programlama	33
2.2.1.4 İndirilmiş gradient metodu	34
2.2.1.5 Menzil ve boş-uzay	35
2.2.1.6 İndirilmiş gradient ve QP.....	36
2.2.1.7 Penaltı fonksiyonu metodu	38
İndirilmiş gradient ve nonlinear kısıtlar	38
Penaltı fonksiyonları	39
Arttırılmış Langrangian.....	41
Tam penaltı fonksiyonu.....	42
2.2.1.8 Ardışık kuadratik programlama	43
Kuadratik / lineer model.....	43
Penaltı fonksiyonu ile SQP.....	45
2.2.2 Eşitsizlik Kısıtı.....	46
2.2.2.1 Optimal koşulu	47
2.2.2.2 Eşitsizliklerin eşitlik haline getirilmesi	47
2.2.2.3 Kuadratik programlama	48
2.2.2.4 İndirilmiş gradient metodu.....	49
2.2.2.5 Basit sınırlar için indirilmiş gradient metodu	49
2.2.2.6 Eşitsizlik kısıtı için penaltı fonksiyonu	50
2.2.2.7 Eşitsizlik kısıtı için arttırılmış Langrangian fonksiyonu.....	51
2.2.2.8 Eşitsizlik kısıtı için SQP.....	51
2.2.2.9 Kısıtlı optimal kontrol problemi, tek link manipulator	51
Benzetim sonuçları.....	54
3. LİNEER OPTİMİZASYON.....	57
3.1 Doğrusal Kuadratik Regulator	57
3.2 Model Öngörülü Kontrol.....	58
3.2.1 Kayan ufuk prensibi	59
3.2.2 MPC'de kullanılan modeller	62
3.2.2.1 Belirsizliklerin modellenmesi	63
3.2.2.2 Gürültünün modellenmesi.....	63
3.2.2.3 Proses modelleri	63
Darbe cevabı modeli	64
Bozucu modeli	65
Basamak cevabı modeli.....	65
Bozucu nodeli	66
Transfer fonksiyonu modeli.....	66
Bozucu modeli	67
$T(z^{-1})$ 'in seçimi.....	68
Durum-uzayı modeli	68
Artımsal model	68
Bozucu modelleri.....	69
Çıkış bozucu modeli.....	69
Giriş bozucu modeli	69
Durum ve çıkış bozucu modeli	70
Gürültü modeli.....	71

3.2.2.4 Öngörü modeli.....	71
Darbe cevabı öngörü modeli.....	72
Basamak cevabı öngörü modeli.....	72
Transfer fonksiyonu öngörü modeli.....	73
Durum-uzayı öngörü modeli.....	75
Nominal durum-uzayı öngörü modeli.....	75
Durum gözlemcisi.....	77
Sürekli rejim Kalman filtresi.....	78
Durum-uzayı öngörü modeli.....	80
Gözlemci modeli.....	81
3.2.2.5 Amaç fonksiyonu.....	82
3.2.2.6 Kısıtsız MPC problemi.....	83
3.2.2.7 Kısıtlı MPC problemi.....	84
3.2.2.8 MPC algoritması.....	84
3.2.2.9 Kontrol kanununun bulunması.....	85
Kısıtsız MPC kontrol kanununun bulunması.....	85
Kısıtlı MPC kontrol kanununun bulunması.....	87
3.2.2.10 Kuadratik programlama.....	89
3.2.2.11 Kararlılık.....	90
3.2.2.12 Kontrol parametrelerinin ayarlanması.....	91
Öngörü ufkunun etkisi.....	91
Kontrol ufkunun etkisi.....	92
Lambda ağırlık katsayısı etkisi.....	93
Parametrelerin ayarlanması.....	93
3.2.2.13 Nonlineer sistemler için MPC uygulamaları.....	93
4. BENZETİM SONUÇLARI.....	95
4.1 Magnetik Tahrikli Kütle-Yay-Damper Sistemi.....	95
4.1.1 Amaç ve kapsam.....	96
4.1.2 Sistem modeli.....	96
4.1.3 Sistemin kontrolü.....	97
4.1.4 Sonuçlar.....	98
4.2 Van Der Vusse Tepkimesi.....	98
4.2.1 Amaç ve kapsam.....	99
4.2.2 Sistemin kontrolü.....	99
4.2.3 Sonuçlar.....	100
4.3 Tek Link Manipulator.....	100
4.3.1 Amaç ve kapsam.....	101
4.3.2 Sistemin kontrolü.....	101
4.3.3 Sonuçlar.....	102
5. SONUÇLAR VE DEĞERLENDİRME.....	103
KAYNAKLAR.....	105
EKLER.....	107
ÖZGEÇMİŞ.....	113

KISALTMALAR

Q-N	: Quasi-Newton
BFGS	: Broyden – Fletcher – Goldfarb - Shanno
DFP	: Davidon – Fletcher – Powell
TLM	: Tek Link Manipulator
ZOH	: Zero-Order Holder
KKT	: Karush-Kuhn-Tucker
QP	: Quadratic Programming
EQP	: Equality Quadratic Programming
SQP	: Sequential Quadratic Programming
MPC	: Model Predictive Control
NLP	: Nonlinear Programming
NMPC	: Nonlinear Model Predictive Control
LQR	: Linear Quadratic Regulator
RH	: Receding Horizon
DCM	: Darbe Cevabı Modeli
BCM	: Basamak Cevabı Modeli
TFM	: Transfer Fonksiyonu Modeli
EMKYD	: Elektromagnetik Kütle Yay Damper
VDV	: Van Der Vusse
s.t.	: subject to

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1: Algoritmaların karşılaştırılması.....	30
Çizelge 4.1: EMKDY sistem parametreleri.....	96

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Lokal ve global minimum	3
Şekil 2.2: İteratif azalma	8
Şekil 2.3: Zigzag azalma metodu	10
Şekil 2.4: Tek link manipulatör	27
Şekil 2.5: TLM blok diyagramı	28
Şekil 2.6 Sistemin durumları	29
Şekil 2.7: Gradient	29
Şekil 2.8: Optimal giriş dizini u^*	29
Şekil 2.9: Geri beslemeli kapalı çevrim	30
Şekil 2.10: Sistemin durumları	30
Şekil 2.11: Optimal giriş dizini u^* ve K	31
Şekil 2.12: Tek link manipulatör	52
Şekil 2.13: TLM blok diyagramı	54
Şekil 2.14: Sistemin durumları	54
Şekil 2.15: Gradient	55
Şekil 2.16: Optimal giriş dizini u^* ve du^*	55
Şekil 3.1: Klasik kontrol ve RH'deki kısıtlar	60
Şekil 3.2: RH başlangıç durumu ($t=k$)	60
Şekil 3.3: RH'deki optimum kontrol dizini ($t=k$)	61
Şekil 3.4: RH kontrol girişi ve bir sonraki hesaplama	61
Şekil 3.5: ($t=k$) ve ($t=k+1$) hesaplanan kontrol dizini	62
Şekil 3.6: Entegratör içermeyen ve içeren sistem	64
Şekil 3.7: Darbe ve basamak cevabı	66
Şekil 3.8: Bozucu ve gürültü modelleri	71
Şekil 3.9: Durum gözlemcisi	77
Şekil 3.10: Sürekli rejim Kalman filtresi	79
Şekil 3.11: Sistem modeli	82
Şekil 3.12: MPC kapalı çevrim	84
Şekil 3.13: MPC iç yapısı	85
Şekil 3.14: Öngörü ufkun etkisi	92
Şekil 3.15: Kontrol ufkunun etkisi	92
Şekil 3.16: Lamba ağırlık katsayı etkisi	93
Şekil 4.1: EKMYD sistemi	95
Şekil 4.2: EKMYD sistem modeli	97
Şekil 4.3: Blok diyagramı	97
Şekil 4.4 Çıkış ve giriş değerleri	98
Şekil 4.5: Çıkış değeri	99
Şekil 4.6: Optimal giriş değeri	100
Şekil 4.7: Tek link manipulator	100
Şekil 4.8: Bozucu etki ve ölçüm gürültüsü	101

Şekil 4.9: Sistemin durumları	102
Şekil 4.10: Optimal u^*	102

LİNEER OLMAYAN SİSTEMLERE KUADRATİK OPTİMAL KONTROL YÖNTEMLERİ İLE MODEL ÖNGÖRÜLÜ KONTROL UYGULANMASI ve KARŞILAŞTIRILMASI

ÖZET

Optimizasyon konusu yıllardır mühendisler, bilimadamları, yöneticiler ve birçok kişi tarafından çok ilgi görür. Üretim maliyeti, kalite sağlanması, rekabet ve en önemli olarakta iş dünyasındaki başarı bunun sebepleridir. Optimizasyon konusunu görmezden gelmek günümüzde tercih edilebilecek seçenek değildir.

Günümüzde optimizasyon problemlerinden biri olan kontrol sistemi tasarımı da önemli yer teşkil etmektedir. Sistemler hemen hemen lineer olmayan davranış gösterir. Lineer sistem teorisine dayanan kontrolcülerin tasarımı her zaman lineer olmayan durumlara göre daha kolay ve basit olmuştur. Pratikte ise tasarlanan kontrolcü, gerçek lineer olmayan davranış gösteren gerçek fiziksel sisteme uygulanır.

Bu çalışmada kısıtlı, kısıtsız, lineer ve lineer olmayan durumlara ait optimizasyon problemleri tanıtılacak, bu problemlere yönelik çözüm metodları ve algoritmaları gösterilecektir. Optimal kontrolcü tasarımı lineer veya lineer olmayan programlama kapsamında ele alınıp, programlama algoritmaları ile lineer sistem teorisine göre tasarlanan model öngörülü kontrol, lineer olmayan sistemlere uygulanıp performansları karşılaştırılacaktır.

APPLICATION AND COMPARISON OF QUADRATIC OPTIMAL CONTROL METHODS AND MODEL PREDICTIVE CONTROLLER TO NONLINEAR SYSTEMS

SUMMARY

The subject of optimization has been receiving attentions from engineers, scientists, managers, and most everybody else. This is driven by competition, quality assurance, cost of production, and most importantly, the success of business enterprises. Ignoring the practice of optimization is not an option during current times.

In nowadays, one of the most important optimization problems is constructing the controller. Physical systems almost behave with nonlinearity. The controllers designed based on linear system theory has been easier than the nonlinear one. In practice, the designed controllers are used on the real systems which behave with nonlinearity.

In this project, building the optimization problems with different kinds such as constrained, unconstrained, linear and nonlinear, reviewing the solving methods to these kind of optimization problems are considered. Designing an optimal controller with the perspective of linear or nonlinear programming solving algorithms and based on the linear system theory the model predictive controller are analyzed and compared of their performance.

1. GİRİŞ

Optimizasyon konusu yıllardır mühendisler, bilimadamları, yöneticiler ve birçok kişi tarafından çok ilgi görür. Üretim maliyeti, kalite sağlanması, rekabet ve en önemli olarakta iş dünyasında başarı bunun sebepleridir. Optimizasyon konusunu görmezden gelmek günümüzde tercih edilebilecek seçenek değildir.

Optimizasyon problemlerine ait çözümler yazılım ve yeterli bilgisayar kaynakları gerektirir. Belirtici biçimde son yıllarda optimizasyon teknikleri pek değişikliğe uğramamıştır. Gelişen teknoloji ile, yüksek bilgisayar hesaplama kapasitesi karmaşık lineer olmayan sistemler üzerinde de optimizasyon tekniklerini uygulamamızı sağlar. Bu çözüm yöntemleri optimizasyon problemlerinin kategorisine göre değişir.

Endüstride artan rekabet düzeyi geliştirilmekte olan ürünlerin her zamankinden daha ucuz, daha iyi ve daha hızlı müşteriyle buluşmasını gerektirir. Artan rekabet baskısından dolayı ürünün kalitesi ne kadar yüksek olmalı ise üretim maliyetide az olmalıdır. Her bir parametrenin optimize edilmesi artık kaçınılmazdır. Bu bağlamda optimizasyon çözüm metodları önemli bir yet tutar.

Günümüzde optimizasyon problemlerinden biri olan kontrol sistemi tasarımı da önemli yer teşkil etmektedir. Rekabet içinde olan şirketler, üretim sürecini kontrol veya ürünün kendisi bir sistem ise onun kontrolünde dahi optimal koşullar artık gün yüzüne çıkmaktadır. Bu optimizasyon hedefleri için elde edilen metodların, performanslarıda önemli yer tutar. Sistemler hemen hemen lineer olmayan davranış gösterir. Lineer sistem teorisine dayanan kontrolcülerin tasarımı her zaman lineer olmayan durumlara göre daha kolay ve basit olmuştur. Pratikte ise tasarlanan kontrolcü, gerçek lineer olmayan davranış gösteren gerçek fiziksel sisteme uygulanır.

Farklı yapıdaki kontrol stratejileri, optimizasyon çatısı adı altında hesaplanan değerlerinde, sistem çeşidine göre incelenmesi ve performansı iyice düzenlenmelidir. Bunun için kullanılan program ve iç yapısı değiştirilmelidir. Bu çalışmada kısıtlı, kısıtsız, lineer ve lineer olmayan durumlara ait optimizasyon problemleri tanıtılacak, bu problemlere yönelik çözüm metodları ve algoritmaları gösterilecektir. Optimal

kontrolcü tasarımı lineer veya lineer olmayan programlama kapsamında ele alınıp, programlama algoritmaları ile lineer sistem teorisine göre tasarlanan model öngörülü kontrol, lineer olmayan sistemler için performansları karşılaştırılacaktır.

1.1 Amaç ve Kapsam

Bu tezin çalışmasının amacı hiçbir şekilde lineerleştirme işlemi kullanılmadan kontrol problemlerinin iteratif yöntemlerle çözülmesidir. Ayrıca kuadratik olmayan bir amaç fonksiyonu dahi olsa, kuadratik olarak ele alınabileceğini ve doğrusal olmayan sistemlere ait optimizasyon metodları ile sistemin doğrusal modelini kullanan lineer optimizasyon kuramlardan biri olan model öngörülü kontrol, lineer olmayan sistemler üzerinde performanslarının karşılaştırılmasıdır. Sistem modelleri durum uzay formunda tanımlanarak, Kalman filtrelemesi, durum gözlemcisi anlatılacaktır. Ele alınan optimizasyon problemi, MATLAB ve SİMULİNK ortamında çözülmüştür. Tasarlanan kontrolcülerin, bozucu ve gürültü gibi olası etkilere karşı performansları irdelenecektir.

1.2 Tezin İçeri

Bölüm 2’de nonlinear optimizasyon problemleri tanıtılır, bu problemlere ait çözüm metodları gösterilir, kontrol problemleri nasıl bir programlama problemi olarak ele alındığı gösterilir ve parametrelerin seçimine ilişkin bilgi verilir.

Bölüm 3’de lineer optimal kontrol problemini tanıtılır. Buna ait çözüm metodları olan LQR ve MPC nin karakteristikleri söylenir. MPC tanıtılır, çözümün oluşturulmasında, parametre seçimi ve buna ait kararlılık ele alınır.

Bölüm 4’de, benzetimlerde kullanılacak sistemler ve senaryolar tanıtılır. Kontrol edilecek sistemlerin hepsi lineer olmayan davranış sergiler.

Bölüm 5’de ise tez özetlenir, ve çalışma ile ilgili yorum ve öneriler aktarılır.

2. NONLİNEER OPTİMİZASYON

2.1 Kısıtsız Optimizasyon

Kısıtsız optimizasyon problemleri, belirli bir $f(x)$ fonksiyonunu kısıtlar olmadan minimize veya maksimize etme problemidir. Kısıtsız problemler bir çok uygulamada karşımıza çıkar. Bu kısımda kısıtsız optimizasyon ile ilgili optimal koşulları, problemlerin çözümüne ilişkin kullanılan algoritmalar tanıtılacaktır.

2.1.1 Lokal ve global minimum

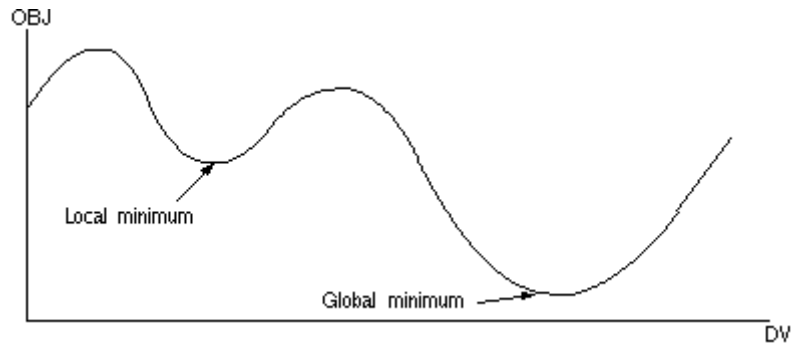
Eğer x^* vektörü çevresindekilerden daha büyük değil ise, bu x^* vektörü $f(x)$ fonksiyonunun kısıtsız lokal minimumudur, ve eğer öyle bir $\varepsilon > 0$ sayısı mevcut ise, aşağıdaki denklem ile tanımlıdır.

$$f(x^*) \leq f(x), \quad \forall x \quad \|x - x^*\| < \varepsilon \quad (2.1)$$

Eğer x^* vektörü bütün diğer vektörlerden daha büyük değil ise, bu x^* vektörü $f(x)$ fonksiyonunun kısıtsız global minimumudur.

$$f(x^*) \leq f(x), \quad \forall x \in \mathcal{R}^n \quad (2.2)$$

Eğer kısıtsız lokal veya global minimum $f(x^*) < f(x)$ eşitsizlik durumunda kesin global ve lokal minimumdur. Şekil 2.1 'de lokal ve global minimumlar gösterilmiştir.



Şekil 2.1 Lokal ve global minimum

2.1.2 Optimal için gerekli koşullar

Verilen bir x^* noktasının, mümkünse lokal veya global minimum olup olmadığını bulmamız gerekir. Bu yüzden optimal olma koşulları belirtilmiştir. Teorem (a) ilk-mertebeden gerekli koşulu, teorem (b) ikinci-mertebeden gerekli koşulu belirtir.

Teorem (a) : Bir f fonksiyonu, $f: \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ tanımlı, ve x^* noktasında türevi alınabilir olsun. Eğer öyle bir d vektörü tanımlı ve $\nabla f(x^*)^T d < 0$ ise, ayrıca öyle bir $\delta > 0$ sayısı var ise $f(x^* + \lambda d) < f(x^*)$ her $\lambda \in (0, \delta)$ için d vektörü x^* 'in azalan yönüdür denir.

Kanıt: $f: \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ 'de tanımlı f fonksiyonun x^* üzerinde türevi alınırsa

$$f(x^* + \lambda d) \approx f(x^*) + \lambda \nabla f(x^*)^T d + \lambda o(x^*; \lambda d) \quad (2.3)$$

Burada $\lambda \rightarrow 0$ gittikçe $o(x^*; \lambda d) \rightarrow 0$ gider. (2.3) denklemini tekrar düzenler ve λ ile bölersek ($\lambda \neq 0$) aşağıdaki gibi tanımlanır:

$$\frac{f(x^* + \lambda d) - f(x^*)}{\lambda} = \nabla f(x^*)^T d + o(x^*; \lambda d) \quad (2.4)$$

Tanımlı olan $\nabla f(x^*)^T d < 0$ ve $\lambda \rightarrow 0$ gittikçe $o(x^*; \lambda d) \rightarrow 0$, ayrıca öyle bir $\delta > 0$ sayısı mevcuttur ki her $\lambda \in (0, \delta)$ için $\nabla f(x^*)^T d + o(x^*; \lambda d) < 0$ bizi şu sonuca götürmektedir.

f fonksiyonu, $f: \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ tanımlı, ve x^* noktasında türevi alınabilir olsun. Eğer x^* lokal minimum ise, $\nabla f(x^*) = 0$ dır.

Kanıt: Farzedelim ki $\nabla f(x^*) \neq 0$, ve $d = -\nabla f(x^*)$ kabul edersek, $\nabla f(x^*)^T d = -\nabla f(x^*)^T \nabla f(x^*) = -\|\nabla f(x^*)\|^2 < 0$ olur ve Teorem (a) ile x^* 'nin lokal minimum olduğu varsayılır buda bizi $\nabla f(x^*) = 0$ sonucuna götürür.

Yukarıda tanıtılmış olan koşul, f fonksiyonun ilk türevi olan gradient vektörleri kullanmaktadır. Bu yüzden ilk-mertebe koşul diye adlandırılmaktadır. Gerekli koşul Hessian matrix bazında da tanımlanabilir, buna da ikinci-mertebe koşul denmektedir ve Teorem (b) 'de verilmiştir:

Teorem (b) Bir f fonksiyonu, $f : \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ tanımlı, ve x^* noktasında iki kez türevi alınabilir olsun. Eğer x^* lokal minimum ise, $\nabla f(x^*) = 0$ ve $H(x^*)$ pozitif tanımlıdır.

Kanıt: Gelişigüzel d vektörü yönü seçilir ve f fonksiyonun x^* etrafında türevi alınırsa :

$$f(x^* + \lambda d) \approx f(x^*) + \lambda \nabla f(x^*)^T d + \frac{1}{2} \lambda^2 d^T H(x^*) d + \lambda^2 o(x^*; \lambda d) \quad (2.5)$$

Burada $H(x^*) = \nabla^2 f(x^*)$ olarak tanımlıdır, ve x^* lokal minimum olduğundan $\nabla f(x^*) = 0$ 'dır. (2.5) denklemini tekrar düzenler ve λ^2 ile bölersek ($\lambda^2 > 0$) :

$$\frac{f(x^* + \lambda d) - f(x^*)}{\lambda^2} = \frac{1}{2} d^T H(x^*) d + o(x^*; \lambda d) \quad (2.6)$$

x^* lokal minimum olduğundan, küçük λ değerleri için $f(x^* + \lambda d) \geq f(x^*)$, (2.6)

denkleminde görülür ki, küçük λ değerleri için $\frac{1}{2} d^T H(x^*) d + o(x^*; \lambda d) \geq 0$ 'dir.

Limit işlemi uygularsak $\lambda \rightarrow 0$ yaklaştığında $o(x^*; \lambda d) \rightarrow 0$ yaklaşır ve burdan da şu sonucu elde edilir: $H(x^*)$ pozitif tanımlıdır.

2.1.3 Optimal için yeterli koşullar

Optimal için gerekli koşullardan yukarıda bahsedildi. Bu koşullar her lokal minimum için sağlanmalıdır, fakat bu koşulları sağlayan noktalar lokal minimum olmak zorunda değildir. Teorem (c) bununla ilgili yeterli koşulu vermektedir.

Teorem (c) Bir f fonksiyonu, $f : \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ tanımlı, ve x^* noktasında iki kez türevi alınabilir olsun. Eğer $\nabla f(x^*) = 0$ ve $H(x^*)$ pozitif tanımlı ise, x^* kesin lokal minimumdur.

Kanıt: f fonksiyonu x^* etrafında iki kez türevi alınabildiğinden, her $x \in \mathfrak{R}_n$ için

$$f(x) \approx f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^T H(x^*) (x - x^*) + \|x - x^*\|^2 o(x; x^*) \quad (2.7)$$

Burada $x \rightarrow x^*$ gittikçe $o(x; x^*) \rightarrow 0$ gider. Eğer öyle bir x^* 'e yaklaşan $\{x_k\}$ dizin var ise, her k için $f(x_k) \leq f(x^*)$, $x_k \neq x^*$ geçerlidir. Böyle bir $\{x_k\}$ dizini için

$\nabla f(x^*) = 0$, $f(x_k) \leq f(x^*)$ ele alınırsa ve $(x_k - x^*) / \|x_k - x^*\| = d_k$ şeklinde yazılırsa:

$$\frac{1}{2} d_k^T H(x^*) d_k + o(x; x_k - x^*) \leq 0 \quad \text{her } k \text{ için} \quad (2.8)$$

Fakat her k için $\|d_k\| = 1$, ve öyle bir K indexi mevcuttur ki, $\{d_k\}_K$ d 'ye yakınsar. Bu alt dizini göz önüne alıp, ayrıca $k \in K \rightarrow \infty$ yakınsarken, $o(x; x^*) \rightarrow 0$ gider. Buda $d^T H(x^*) d \leq 0$ sonucunu açığa çıkarır. $H(x^*)$ 'in pozitif tanımlı olduğu varsayımıyla , x^* kesin lokal minimumdur sonucu çıkar.

Teorem (d) Bir f fonksiyonu, $f : \mathfrak{R}_n \rightarrow \mathfrak{R}_1$ tanımlı, ve x^* noktasında iki kez türevi alınabilir olsun. Her x için eğer $\nabla f(x^*) = 0$ ve $H(x^*)$ pozitif tanımlı ise, f fonksiyonu konveks ve pseudokonvekstir. O zaman yalnız ve yalnız (ing. if and only if) $\nabla f(x^*) = 0$, x^* global minimumdur.

Kanıt. Eğer x^* global minimumsa, o zaman $\nabla f(x^*) = 0$. Varsayalım ki $\nabla f(x^*) = 0$ o zaman her $x \in \mathfrak{R}_n$ için $\nabla f(x^*)^T (x - x^*) = 0$ 'dir. x^* 'nin pseduokonveks oluşundan dolayı , her $x \in \mathfrak{R}_n$ için $f(x) \geq f(x^*)$ ve bu da teorem (d) yi kanıtlamış olur.

2.1.4 Kuadratik amaç fonksiyonu

Amaç fonksiyonunu şu şekilde olduğunu varsayalım:

$$f(x) = \frac{1}{2} x^T Q x + b^T x \quad (2.9)$$

Burada Q simetrik $n \times n$ noyutunda matris, b vektördür. Eğer x^* , f fonksiyonunun lokal mimimumu ise, aşağıdaki şartlar sağlanmalıdır:

$$\nabla f(x^*) = Q x + b = 0 \quad (2.10)$$

$$\nabla^2 f(x^*) = Q : \text{pozitif tanımlı olmalıdır.} \quad (2.11)$$

Eğer Q yarı-pozitif (ing. semi positive) tanımlı değil ise, f fonksiyonunun lokal mimimumu bulunmamaktadır. Eğer Q yarı-pozitif tanımlı ise, f fonksiyonu konvekstir.

Kuadratik amaç fonksiyonları, nonlinear programlamada önemlidirler. Birçok uygulamada karşımıza çıkmaktadırlar. Önemli olmasının başka bir nedeni ise, Taylor serisine açılmış olan herhangi bir fonksiyon aşağıdaki gibi tanımlanabilir.

$$f(x) \approx f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^T H(x^*) (x - x^*) + o(\|x - x^*\|^2) \quad (2.12)$$

(2.12) denkleminde görüldüğü üzere, herhangi bir non-kuadratik amaç fonksiyonu, tekil olmayan $[\nabla^2 f(x^*)]$ pozitif tanımlı lokal minimum noktası olan x^* noktası etrafında Taylor serisine açılarak kuadratik bir fonksiyon olarak tanımlanabilir. Buda bize her türlü analiz ve deneysel metodlarımızı pozitif tanımlı kuadratik amaç fonksiyonu üzerinde uygulayamamıza olanak verir. Fakat Hessian matrisin tekil olduğu veya mevcut olmadığı durumlarda, 2. mertebeden sonraki yüksek mertebeli terimleri $o(\|x - x^*\|^2)$ Taylor serisine açılırken, ihmal edilemez duruma gelir ve bu kuadratik amaç fonksiyon yaklaşımında kusurlu kılmaktadır.

2.1.5 Kısıtsız optimizasyon algoritmaları

Bu bölümde kısıtsız optimizasyon problemlerine ait çözüm yöntem ve algoritmaları anlatılacak ve en son kısımda kontrol problemine ait çözüm metodu verilecektir.

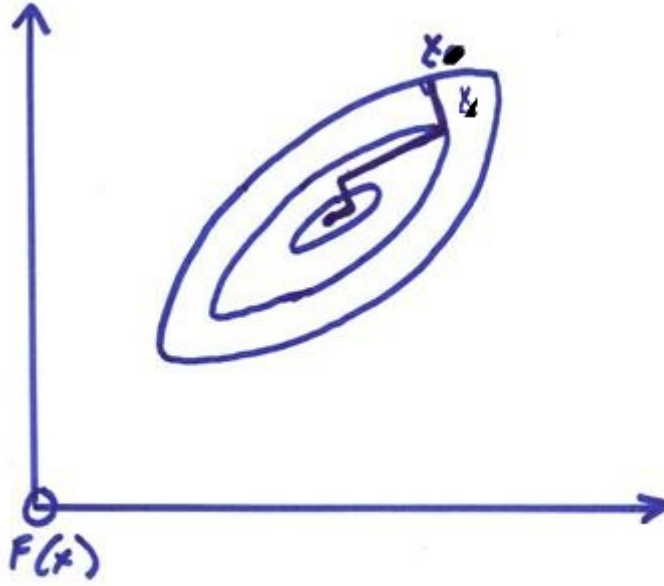
2.1.5.1 Azalan yön ve adım büyüklüğü

$f : \mathcal{R}_n \rightarrow \mathcal{R}_1$ 'de tanımlı türevi alınabilen kısıtsız f fonksiyonunun minimizasyonunu ele alalım. Bir çok kısıtsız minimizasyon probleminde kullanılan algoritmalar, iteratif azalma adı ile geçen önemli metodu kullanmaktadırlar. Şu şekilde tanımlıdır:

Başlangıç noktası x^0 (ilk tahmin) seçilerek, ardışık olarak her iterasyon adımında f fonksiyonunu azaltan x^1, x^2, \dots vektörleri bulunur, aşağıda şu şekilde tanımlanır:

$$f(x^{k+1}) < f(x^k) \quad k = 0, 1, \dots \quad (2.13)$$

Şekil 2.2'de bir f fonksiyonunun iteratif olarak azalması gösterilmiştir.



Şekil 2.2 İteratif azalma

2.1.5.2 Gradient metodu

$x \in \mathfrak{R}_b$ tanımlı ve $\nabla f(x) \neq 0$ olan bir x vektörünü ele alalım. Yarı çizgideki vektörleri:

$$x_a = x - a\nabla f(x) \quad \forall a \geq 0 \quad (2.14)$$

(2.14) denkleminin x noktası etrafında Taylor serisine açılımı aşağıdaki gibi tanımlanır:

$$f(x_a) = f(x) + a\nabla f(x)^T (x_a - x) + o(\|x_a - x\|^2) \quad (2.15)$$

Denklem (2.14) teki $x_a - x = -a\nabla f(x)$ değerini denklem (2.15) te yerine yazılırsa:

$$f(x_a) = f(x) - a\|\nabla f(x)\|^2 + o(a\|x_a - x\|^2) \quad (2.16)$$

elde edilir, ve Taylor serisine x noktası etrafında açılmış f fonksiyonu şu şekilde yazılabilir:

$$f(x_a) = f(x) - a\|\nabla f(x)\|^2 + o(a) \quad (2.17)$$

Buradaki $a\|\nabla f(x)\|^2$ terim, sıfır yakınlarında, pozitif ama küçük a değerleri için $o(a)$ değerini ihmal etmemizi sağlamaktadır. (2.17) denklemini tekrar düzenlenirse;

$$f(x_a) = f(x) - a\|\nabla f(x)\|^2 \quad (2.18)$$

(2.18) denkleminde de görüldüğü gibi $f(x_a)$ değerinin $f(x)$ 'ten küçük olduğu gözlemlenir.

Bir adım daha ileriye götürürsek işlemi;

$$x_a = x + ad \quad \forall a \geq 0 \quad (2.19)$$

Burada $d \in \mathfrak{R}_b$ 'de tanımlı, $\nabla f(x)$ ile 90 dereceden büyük açığa sahip yön vektörüdür. Şu şekilde tanımlıdır:

$$\nabla f(x)^T d < 0 \quad (2.20)$$

Taylor serisine açılmış olan f fonksiyonunu tekrar yazarsak;

$$f(x + ad) = f(x) + a\nabla f(x)^T d + o(a) \quad (2.21)$$

Buradaki $a\nabla f(x)^T d$ terim, sıfır yakınlarında , pozitif ama küçük a değerleri için o(a) değerini ihmal etmemizi sağlamaktadır. (2.21) denklemi tekrar düzenlenirse;

$$f(x + ad) = f(x) + a\nabla f(x)^T d \quad (2.22)$$

(2.22) denkleminde de görüldüğü üzere $f(x + ad)$ değeri, $f(x)$ değerinden küçüktür. Bı sayede şu genel algoritma ortaya çıkmaktadır:

$$x^{k+1} = x^k + a^k d^k \quad k = 0,1,\dots \quad (2.23)$$

Burada $\nabla f(x) \neq 0$ ve d^k öyle seçilmelidir ki, aşağıdaki eşitliği sağlamalıdır:

$$\nabla f(x^k)^T d^k < 0 \quad (2.24)$$

Adım büyüklüğü olan a^k pozitif olmalıdır. Eğer $\nabla f(x) = 0$ olursa, metod duracak ve $x^{k+1} = x^k$ olacaktır. Yön vektörü olan d^k ve $f(x)$ fonksiyonunun gradienti olan $\nabla f(x)$ arasındaki bağ yüzünden bu tarz algoritmalara gradient metodu denmektedir. Bazı yazarlar bu şekilde kabul etmiştir.[1 s.24]. Bu gradient metodunun önemli kısmı ayrıca azalan algoritmadır, adım büyüklüğü seçilmelidir ki, aşağıdaki koşul sağlanmalıdır:

$$f(x^k + a^k d^k) < f(x^k) , \quad k = 0,1,\dots \quad (2.25)$$

2.1.5.3 Azalma yönü seçimi

Birçok gradient metodu şu şekilde tanımlıdır:

$$x^{k+1} = x^k - a^k D^k \nabla f(x^k) \quad (2.26)$$

Burada D^k pozitif tanımlı simetrik matris ve d^k yön vektörü $d^k = -D^k \nabla f(x^k)$ olarak tanımlıdır. Azalma koşulu ise $\nabla f(x^k)^T d^k < 0$, D^k pozitif tanımlı matris olduğundan şu şekilde yazılabilir;

$$\nabla f(x^k)^T D^k \nabla f(x^k) > 0 \quad (2.27)$$

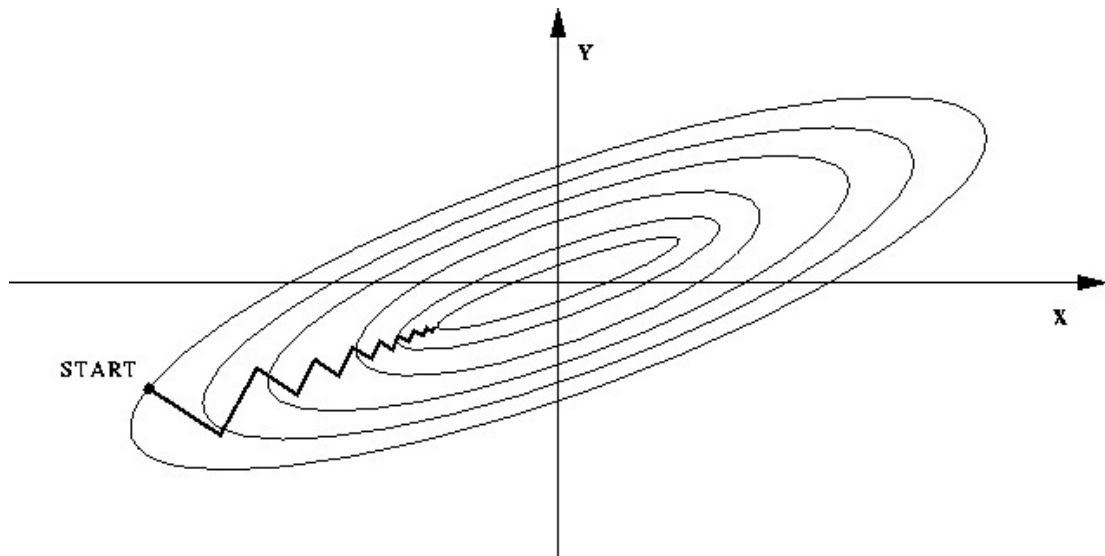
D^k matris seçimine ait metodlar aşağıdaki gibi verilmiştir.

2.1.5.4 Zigzag azalma (steepest descent) metodu

Zigzag azalma (ing. Steepest Descent) metodunda D^k seçimi şu şekilde olmaktadır;

$$D^k = I, \quad k = 0, 1, \dots \quad (2.28)$$

(2.28) denkleminde I , $n \times n$ birim matristir. Bu en basit seçim olup, fakat çözümü yavaştır. Zigzag azalma ismi, bu metodun ilginç bir özelliği olan bütün yönler içinde $d^k \in \mathfrak{R}_m$ normalize negatif gradient yönünden kaynaklıdır. $d^k = -\nabla f(x^k) / \|\nabla f(x^k)\|$ Bu gradient yönü normalize edilir ki, $\|d^k\| = 1$ olur. Bu da $a=0$ durumunda, $f(x^k + ad)$ fonksiyonunun d yönündeki $\nabla f(x^k)^T d$ eğimini minimize eder. Zigzag azalma metodu Şekil 2.3'te verilmiştir.



Şekil 2.3 Zigzag azalma metodu

2.1.5.5 Newton metodu

Newton Metodun'da D^k matrisi şu şekilde seçilmektedir:

$$D^k = (\nabla^2 f(x^k))^{-1}, \quad k = 0,1,\dots \quad (2.29)$$

$\nabla^2 f(x)$ pozitif tanımlıdır. Newton Methodunun ana felsefesi, verilen x^k noktası etrafında f fonksiyonunun kuadratik yakınsaması değerinin her iterasyonda minimize etmektir. f fonksiyonunun x^k noktası etrafında kuadratik yakınsaması şu şekilde tanımlanır:

$$f^k(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \left(\frac{1}{2} (x - x^k)^T \nabla^2 f(x^k) (x - x^k)\right) \quad (2.30)$$

Denklem (2.30) daki $f^k(x)$ fonksiyonunun türevini sıfıra eşitlersek:

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0 \quad (2.31)$$

(2.31) denklemini tekrar düzenlersek:

$$(x - x^k) = -(\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.32)$$

(2.32) denkleminde $f^k(x)$ 'yi minimum yapacak olan bir sonraki iterasyon değeri x^{k+1} şu şekilde bulunur.

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.33)$$

(2.33) denklemini Newton method'unun saf halidir (ing pure form). Daha genel iterasyon metodu ise aşağıda verilmiştir.

$$x^{k+1} = x^k - a^k (\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.34)$$

(2.34) denkleminde adım büyüklüğü $a^k = 1$ olması durumudur. Newton metod'u basit iterasyonla pozitif tanımlı kuadratik bir fonksiyonun global minimumunu bulur.

Fakat Hessian matrisin ($\nabla^2 f(x)$) tekil olduğu durumlar veya mevcut olmadığı durumlarda meydana gelebilir. Bunun için modifiye edilmiş Newton methodu bulunur.

2.1.5.6 Modifiye edilmiş Newton methodu

Bir üstte anlatıldığı gibi Newton Metod'u Hessian matrisi ($\nabla^2 f(x)$) eğer pozitif tanımlı ise, minimum nokta olan x^k 'ya kuadratik olarak sonuca varmaktadır. Fakat Hessian matrisin tekil olabilme durumu, yön vektörü $d^k = -\nabla^2 f(x^k)\nabla f(x^k)$ azalan vektör olamama durumu, veya $\nabla f(x^k)d^k < 0$, birim adım f fonksiyonunda azalma vermeyebilir Bu yüzden modifiye edilmiş Newton methodu kullanılır. Bu kısım bir sonraki kısımlarda detaylıca incelenecektir.

D^k matrisinin farklı seçimleri Quasi-Newton metodunda açıklanacaktır. d^k yön vektörünün $d^k = -D^k\nabla f(x^k)$ olarak seçilmediği, Congugate Gradient Metodu sonraki kısımlarda anlatılacaktır.

2.1.6 Adım boyutu seçimi

Bu bölümde Gradient metodunda adım boyutu olan a^k 'nin seçimine ilişkin pratikte kullanılan metodlar anlatılacaktır.

2.1.6.1 Minimize kuralı

d^k yönü boyunca f amaç fonksiyonu minimize edilecek şekilde a^k seçilir, ve şu şekilde tanımlanmıştır:

$$f(x^k + a^k d^k) = \min_{a \geq 0} f(x^k + a d^k) \quad (2.35)$$

2.1.6.2 Limitli minimize kuralı

Bir çok durumda uygulanması daha kolay olan bu metod, minimize metodunun bir versiyonudur. Öyle bir sabit skalar $s > 0$ sayısı seçilir ki, a^k adım büyüklüğü, $[0, s]$ bölgesinde tanımlı olan bütün adım büyüklükleri arasından, amaç fonksiyonunu en fazla minimize edilecek şekilde seçilir, ve şu şekilde tanımlanır:

$$f(x^k + a^k d^k) = \min_{a \in [0, s]} f(x^k + a d^k) \quad (2.36)$$

Minimize ve limitli minimize kuralı tek boyutta doğru arama (ing. one-dimensional line search) metodları ile kullanılmalıdır, ona rağmen a^k adım büyüklüğünü minimize etmek kolay bir şekilde hesaplanamaz. Pratikte, adım büyüklüğü (2.36) denkleminde tanımlı kriteri sağladığında doğru arama algoritması durur.

2.1.6.3 Armijo kuralı

Doğru arama yöntemlerindeki hesaplama sorunlarının üstesinden gelebilmek için, ardışık adım büyüklüğü azalamı yapılmalıdır. En basit olarak adım büyüklüğü için ilk tahmin s seçilir, ve eğer $x^k + sd^k$ terimi, f fonksiyonunu daha ileri bir noktaya getirmediği durumda $f(x^k + sd^k) \leq f(x^k)$, daha büyük değere gelesiye kadar adım boyutu azaltılır. Bu metod pratikte çok kullanışlı olmasına rağmen, teoride f amaç fonksiyonunun her iterasyonda daha büyük değerinin bulunması durumunda seçilecek olan adım büyüklüğü, f amaç fonksiyonunun minimum yapma garantisini vermez.

Armijo kuralı üç skalar parametre ile tanımlıdır. s , β , ve σ , burada $0 < \beta < 1$, $0 < \sigma < 1$, $s > 0$ olarak tanımlıdır. Adım büyüklüğünü m_k negatif tam sayı olmamak koşulu ile aşağıdaki eşitsizliğin tanımlı olduğu en küçük m değeri için, $a^k = \beta^{m_k} s$ olarak seçilir.

$$f(x^k) - f(x^k + \beta^m s d^k) \geq -\sigma \beta^m s \nabla f(x^k)^T d^k \quad (2.37)$$

Adım büyüklüğü olan $a^k = \beta^{m_k} s$, $m = 0, 1, 2, \dots$ yukarıdaki eşitsizlik sağlanıncaya kadar ardışık olarak hesaplanır ve $m = m_k$ bulunur.

Genellikle σ , sıfıra yakın değer seçilir. Örnek olarak $\sigma \in [10^{-5}, 10^{-1}]$. Azaltma faktörü olan β , genellikle 1/2'den 1/10'a kadar değerlerden seçilir.

2.1.6.4 Goldstein Kuralı

Bu seçim kuralında sabit skalar olan $\sigma \in (0, 1/2)$ arasında seçilir ve adım büyüklüğü aşağıda tanımlı eşitsizliğe göre seçilir.

$$\sigma \leq \frac{f(x^k + a^k d^k) - f(x^k)}{a^k \nabla f(x^k)^T d^k} \leq 1 - \sigma \quad (2.38)$$

2.1.6.5 Sabit adım büyüklüğü

Bu yöntemde sabit adım büyüklüğü $s > 0$ seçilir.

$$a^k = s, \quad k = 0, 1, \dots \quad (2.39)$$

Sabit adım büyüklüğü seçimi basittir, fakat eğer büyük değer seçilirse, uyuşmama (ing. diverge) meydana gelir, küçük değer seçilirse, f amaç fonksiyonunun minimum değerine yaklaşma yavaş olabilir.

2.1.6.6 Azalan adım büyüklüğü

Bu yöntemde adım büyüklüğü sifıra doğru yakınsar.

$$a^k \rightarrow 0, \quad k = 0,1,\dots \quad (2.40)$$

Bu yöntem diğerlerinden farklı olarak, her iterasyonda amaç fonksiyonunda azalmayı garanti etmemektedir. Bu yöntemin zor olan kısmı, o kadar küçük adım boyutuna ulaşılabilir ki, dayanıklı işlem yapılamaz hale gelir, bunun için aşağıdaki durum gereklidir:

$$\sum_{k=0}^{\infty} a^k = \infty \quad (2.41)$$

Son koşul $\{x^k\}$ 'nin yakınsamasını garanti etmektedir. Öyleki, eğer $x^k \rightarrow \tilde{x}$, m ve n gibi fazla ($m > n$) sayılar mevcut ise :

$$x^m \approx x^n \approx \tilde{x}, \quad x^m \approx x^n - \left(\sum_{k=n}^m a^k \right) \nabla f(\tilde{x}) \quad (2.42)$$

Bu kısımda adım büyüklüğü seçiminde kullanılan metodlar verildi. Genellikle yaygın olarak kullanılan Armijo kuralıdır..

2.1.7 Newton metodu

İlk bölümde incelenen gradient metodları arasında bulunan Zigzag azalma metodu, aralarında en yavaşı ve en basitidir. Tam tersi olan Newton metodu, bu bölümde detaylı olarak incelenecektir.

Newton metodu şu iterasyonları içermektedir:

$$x^{k+1} = x^k - a^k (\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.43)$$

Newton azamla yönü (ing. direction of descent) ise;

$$d^k = -(\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.44)$$

Tanımlıdır. Newton metodu'nu, optimal ölçek matrisi $(\nabla^2 f(x^k))^{-1}$ ölçeklendirilmiş Zigzag azalma metodu olarak düşünülebilir.

Adım boyutu seçiminde başlangıç değeri olarak $s=1$ Armijo kuralı kullanıldığında, tekil olmayan Hessian matris ile tanımlı durumda, adım büyüklüğünün azalmasına gerek yoktur, ve Newton metodu şu şekilde tanımlanır:

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.45)$$

(2.45) denklemini Newton metodu'nun saf formu (ing. Newton pure form) olarak tanımlanır. Kısıtsız optimizasyon problemlerinde Newton metodu'nun saf formu, birkaç engel yaratır;

- Hessian matrisin $(\nabla^2 f(x^k))$ tekil olma durumu, buda tersininin alınamamasına sebep olur.
- Saf form amaç fonksiyonunun azalma garantisini vermemektedir.
- Lokal minima problemini kadar lokal maxima probleminide çözmeye çalışır. $\nabla f(x^k) = 0$ problemini çözmeye çalışır.

Bu yüzden, Newton metodu'nu düzenlemek gerekmektedir. Bunun için birçok şema mevcuttur, bu sayede saf formu, gradient tabanlı yön dizini , gradient metoduna dönüştürülebilir.

En basit uygulanabilir yöntem, Newton yönünü, tanımlı olmadığı veya azalan yön olmadığı durumlarda Zigzag yönü ile değiştirmektir.

Genel anlamda, modifiye edilmiş Newton metodu , ilk iterasyonlarda yakınsama garantisi vermekle beraber, ikinci türev bilgisini içeren Hessian matrisin pozitif tanımlı olmadığı durumlar da dahi kullanılabilir. Bu şemalar, Hessian matris üzerinde diyagonal yönde değişimler yapılarak elde edilir, ve aşağıdaki denklem çözümlere yön d^k bulunur:

$$(\nabla^2 f(x^k) + \Delta x)d^k = -\nabla f(x^k) \quad (2.46)$$

Burada Δx diyagonal matristir ve aşağıdaki durum ile tanımlanır:

$$(\nabla^2 f(x^k) + \Delta x) : \text{pozitif tanımlı} \quad (2.47)$$

2.1.7.1 Modifiye edilmiş Cholesky çarpanlarına ayırma

Her pozitif tanımlı Q matrisi kendine ait tek çarpanları bulunmaktadır (ing. factorization) ve şu şekilde tanımlanır:

$$Q = LL^T \quad (2.48)$$

Burada L alt üçgen matris olarak tanımlıdır. $Qx = b$ olarak tanımlı sistem denklemlerini, $Ly = b$ y için ve $L^T x = b$ x için çözerek, çözülebilir. Newton yönüde şu şekilde tanımlı olduğundan:

$$(\nabla^2 f(x^k))d^k = -\nabla f(x^k) \quad (2.49)$$

$(\nabla^2 f(x^k))$ matrisine Cholesky çarpanlarına ayırma metodu kullanılarak (2.49) denkleminde d^k hesaplanabilir. Bu işlem boyunca $(\nabla^2 f(x^k))$ 'un pozitif tanımlı olmama veya tekil olma durumları ile karşılaşılabilir. Bunu engellemek için Hessian matrisin diyagonaldeki elemanlarına müdahalede bulunulur. Bu işlem çarpanlara ayırma yapılırken sırayla gerçekleştirilir, ve şu şekilde tanımlıdır:

$$L^k L^{kT} = \nabla^2 f(x^k) + \Delta x^k \quad (2.50)$$

Örnek olarak, $\nabla^2 f(x^k) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$ olarak tanımlansın, ve Cholesky çarpanlarına

ayırma metodunda, istenilen L matrisi ise şu şekilde tanımlı olsun:

$$LL^T = \begin{pmatrix} a & 0 \\ \gamma & \beta \end{pmatrix} \begin{pmatrix} a & \gamma \\ 0 & \beta \end{pmatrix} \quad (2.51)$$

a, γ ve β öyle seçilir ki, eğer $\nabla^2 f(x^k)$ pozitif tanımlı ise, şu koşulu sağlamalıdır.

$L^k L^{kT} = \nabla^2 f(x^k)$, eğer pozitif tanımlı değilse h_{11} ve h_{22} 'yi değiştirilir. (2.50) ve (2.51) denklemlerinden yola çıkarak a, γ ve β bulunur.

$$a = \begin{cases} \sqrt{h_{11}} & \text{Eğer } h_{11} > 0 \\ \sqrt{h_{11} + \delta_1} & \text{Tersi} \end{cases} \quad (2.52)$$

Burada δ_1 öyle seçilmelidir ki, $h_{11} + \delta_1 > 0$ sağlanmalıdır. a , bulunduktan sonra γ ve β şu şekilde bulunur:

$$\gamma a = h_{12} \Rightarrow \gamma = \frac{h_{12}}{a} \quad (2.53)$$

$$\beta = \begin{cases} \sqrt{h_{22} - \gamma^2} & \text{Eğer } h_{22} > \gamma^2 \\ \sqrt{h_{22} - \gamma^2 + \delta_{21}} & \text{Tersi} \end{cases} \quad (2.54)$$

Bulunan $L^k L^{kT}$ çarpanlar ile, Newton yönü d^k şu şekilde bulunur:

$$L^k L^{kT} d^k = -\nabla f(x^k) \quad (2.55)$$

Bir sonraki iterasyon;

$$x^{k+1} = x^k + a^k d^k \quad (2.56)$$

Burada a^k adım büyüklüğü bahsi geçen Armijo veya diğer yöntemlerle elde edilir.

2.1.7.2 Güvenilir alan metodu

Newton saf adımı, x^k etrafında f fonksiyonunun Taylor serisine ikinci mertebeden açılarak, d değerini minimize ederek elde edileceği, önceki kısımlarda gösterildi;

$$f^k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d \quad (2.57)$$

Biliyor ki $f(d)$, d değerinin sıfıra yakın değerler bölgesinde, $f(x^k + d)$ 'nin iyi bir yaklaşım değeridir, fakat kısıtsız minimizasyonda, bu alan dışına çıkmak zorunda kalınabilir. Bunu önlemek için, sınırlı Newton adımı (ing. restricted Newton step) d^k , $f^k(d)$ fonksiyonunu uygun olan sıfıra yakın alan etrafında minimize edilmesine, güvenilir alan metodu (ing. Trust Region) denir.

$$d^k \in \arg \min_{\|d\| \leq \gamma^k} f^k(d) \quad (2.58)$$

Burada γ^k skalar bir değerdir. Denklem (2.57) ve (2.58) 'dan gözlemleneceği gibi, $(\nabla^2 f(x^k))$ pozitif tanımlı olmasa dahi, genel anlamda Newton yönü, azalan yön olmasa dahi, sınırlı Newton adımı d^k , amaç fonksiyonunu arttır biçimdedir. $f^k(d^k)$, $f(x^k)$ 'dan daha küçük olduğu ve $f(x^k + d^k)$ nın $\|d^k\|$ küçük değerler olması durumunda, $f^k(d^k)$ 'nın ikinci mertebe değerine yakın olduğu görülür. $\|d^k\| \leq \gamma^k$ durumunda;

$$f(x^k + d) = f^k(d) + o((\gamma^k)^2) \quad (2.59)$$

$$f(x^k + d^k) = f^k(d^k) + o((\gamma^k)^2)$$

$$f(x^k + d^k) = f(x^k) + \min_{\|d\| \leq \gamma^k} \left\{ \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d \right\} + o((\gamma^k)^2) \quad (2.60)$$

Olarak seçilir. Burada γ^k skalar değerinin seçimi önemli yer tutmaktadır. Başlangıç değer olarak bir sayı seçilip, ardışık olarak $f(x^{k+1}) < f(x^k)$ eşitsizliği sağlanıyaya kadar azaltılmaya devam edilir. Başlangıç değerinin seçimi ise, çok büyük seçilirse, f amaç fonksiyonunun minimize edildiği ilk değere gelesiyeye kadar çok fazla γ^k azaltım meydana gelebilir. Eğer çok küçük seçilirse, işlem çok yavaş sürebilir.

2.1.8 Eşlenik yön metodu

Eşlenik yön (ing conjugate direction) metodları, zigzag azalma metodunun yavaş çözümünü hızlandırmak, ve Newton metodunda çıkan zorlukların üstesinden gelme amaçlıdır. İlk olarak kuadratik problemlerin çözümü için bulunmuşlardır.

$$\begin{aligned} \text{minimum } f(x) &= \frac{1}{2} x^T Qx + b^T x & (2.61) \\ \text{s.t } & x \in \mathbb{R}^n \end{aligned}$$

Q pozitif tanımlı matris olup, bu metod genel olarak $Ax=b$ lineer sistemlerin çözümünde de kullanılır.

Eşlenik yön metodları, bu tür problemleri maksimum n iterasyon sonunda çözebilir, ayrıca bu metod, kuadratik olmayan optimizasyon problemlerinde de kullanılabilir. Bunun hakkında detaylı bilgi [2.s139] elde edilebilir. Bu kısımda anlatılmayacaktır.

$n \times n$ boyutunda verilen pozitif tanımlı Q matrisin, sıfırdan farklı d^1, \dots, d^k vektörleri Q-Conjugate'i olsun. Bu durumda aşağıda verilen tanıma uymak zorundadırlar:

$$d^{iT} Qd^j = 0 \quad \text{her } i \text{ ve } j \text{ için } \quad i \neq j \quad (2.62)$$

Eğer d^1, \dots, d^k , Q-Conjugate ise, lineerli bağımsızdırlar;

$$d^k = a^1 d^1 + \dots + a^{k-1} d^{k-1} \quad (2.63)$$

(2.63) denklemini $d^{kT} Q$ ile çarparsak, Q-conjugacy elde edilmiş olur.

$$d^{kT} Qd^k = a^1 d^{1T} Qd^1 + \dots + a^{k-1} d^{k-1T} Qd^{k-1} = 0 \quad (2.64)$$

(2.64) denklemini Q pozitif tanımlı ve $d^k \neq 0$ olduğu için mümkün değildir, ve d^1, \dots, d^k lineerli bağımsızdır.

n adet d^0, \dots, d^{n-1} ile tanımlı Q-Conjugate yönüne ait, kuadratik tanımlı f amaç fonksiyonunun kısıtsız minimizasyon problemleri için eşlenik yön metodu şu şekilde tanımlıdır:

$$f(x) = \frac{1}{2} x^T Qx + b^T x \quad (2.65)$$

$$x^{k+1} = x^k + a^k d^k \quad k = 0, 1, \dots, n-1 \quad (2.66)$$

Burada x^0 rastgele bir vektör, ve adım büyüklüğü a^k 2. kısımda anlatılan doğru minimizasyonu kuralı ile bulunur;

$$f(x^k + a^k d^k) = \min_a f(x^k + a d^k) \quad (2.67)$$

(2.67) denklemini a ya göre türevini alıp sıfıra eşitlersek;

$$\frac{\partial f(x^k + a^k d^k)}{\partial a} = d^{kT} \nabla f(x^k + a^k d^k) = 0 \quad (2.68)$$

$$d^{kT} (Q(x^k + a^k d^k) + b) = 0 \quad (2.69)$$

(2.69) denkleminde adım büyüklüğünü ayırırsak:

$$a^k = \frac{d^{kT} (b - Qx^k)}{d^{kT} Q d^k} \quad (2.70)$$

Eşlenik yön metodunun prensibi, ardışık olarak, f fonksiyonunun global minimum olan değerinde içeren, gittikçe genişleyen lineer manifold alanı içinde iteratif olarak minimize etme işlemidir. Her k için x^0 başlangıç değerinden geçerek ve conjugate yönler olan d^0, \dots, d^{n-1} üzerinden yönelerek, x^{k+1} minimize etme işlemidir.

2.1.8.1 Eşlenik gradient metodu

Eşlenik gradient metodu, Gram-Schmidt prosedürün gradient vektörlere uygulanması ile elde edilir.

$$g^k = \nabla f(x^k) = Qx^k + b \quad (2.71)$$

Eşlenik gradient metodu şu şekilde tanımlanır:

$$x^{k+1} = x^k + a^k d^k \quad (2.72)$$

a^k adım büyüklüğü, çizgi minimizasyon kuralı ile, d^k yön vektöründe, k kadar Gram-Schmidt prosedürünün $-g^k$ vektörüne uygulanması ile bulunur.

$$d^k = -g^k + \sum_{j=0}^{k-1} \frac{g^{kT} Q d^j}{d^{jT} Q d^j} d^j \quad (2.73)$$

Burada $d^0 = -g^0$ olarak tanımlıdır, eğer $g^k = 0$ ise, metod optimal sonuç bulur ve durur.

(2.73) denklemi ile verilen yön denklemi daha da sadeleştirilebilir. Eğer (2.73) denklemi genişletilirse, bazı katsayıların toplamının sıfır olacağı gözlemlenir [1,s136]. Sadeleştirme işlemi yapıldığı takdirde, eşlenik gradient metodu şu şekilde tanımlıdır:

$$\begin{aligned}
 d^0 &= -g^0 \\
 d^k &= -g^k + \beta^k d^{k-1} \\
 \beta^k &= \frac{g^{kT} g^k}{g^{k-1T} g^{k-1}} \tag{2.74}
 \end{aligned}$$

n adım sonunda metod optimal sonuca ulaştığında durur.

2.1.8.2 Önkoşullu eşlenik gradient metodu

Önkoşullu eşlenik gradient metodu (ing. Preconditioned conjugate gradient method) farklı koordinat sisteminde uygulanan eşlenik gradient metodudur. Varsayalım ki, değişkenleri farklı değişkenler ile tanımlı olsun; $x = Sy$, burada S tersi alınabilen $n \times n$ boyutunda matristir. Değişken değişimi yapılmış sisteme eşlenik gradient metodu aşağıdaki şekilde uygulanır:

$$\begin{aligned}
 \text{mimumum } h(y) &= f(Sy) = \frac{1}{2} y^T S^T Q S y + b^T S y \\
 \text{s.t} \quad y &\in \mathfrak{R}^n \tag{2.75}
 \end{aligned}$$

$$y^{k+1} = y^k + a^k \tilde{d}^k \tag{2.76}$$

a^k adım büyüklüğü, çizgi minimizasyon kuralı kullanılarak bulunur. \tilde{d}^k yön vektöründe, eşlenik gradient metodu kullanılarak bulunur.

$$\begin{aligned}
 \tilde{d}^0 &= -\nabla h(y^0) \\
 \tilde{d}^k &= -\nabla h(y^k) + \beta^k \tilde{d}^{k-1} \quad k = 0,1,\dots,n-1 \tag{2.77}
 \end{aligned}$$

$$\beta^k = \frac{\nabla h(y^k)^T \nabla h(y^k)}{\nabla h(y^{k-1})^T \nabla h(y^{k-1})} \tag{2.78}$$

$x^k = S y^k$, $\nabla h(y^k) = S g^k$ ve $H = S^2$ olarak yazılırsa;

$$x^{k+1} = x^k + a^k d^k \tag{2.79}$$

$$d^0 = -H g^0, \quad d^k = -H g^k + \beta^k d^{k-1}, \quad k = 0,1,\dots,n-1 \tag{2.80}$$

$$\beta^k = \frac{g^{kT} H g^k}{g^{k-1T} H g^{k-1}} \quad (2.81)$$

Bu şekilde tanımlanmış metoda, H matrisi ile ölçeklendirilmiş önkoşullu eşlenik gradient metodu denir. Eşlenik gradient metodunda olduğu gibi, n adım sonunda f amaç fonksiyonunu minimize ederek durur.

2.1.9 Quasi-Newton metodu

Quasi-Newton metodu gradient metodu olup, aşağıdaki gibi tanımlıdır:

$$x^{k+1} = x^k + a^k d^k \quad (2.82)$$

$$d^k = -D^k \nabla f(x^k) \quad (2.83)$$

D^k pozitif tanımlı, d^k yön vektörünün Newton yön vektörüne yakınsaması için, bir iterarıftan diğerine geçerken değiştirilebilir matristir. Bu tür metodlar, Hessian matrisin hesaplama güçlüğünden, hesaplama işlemini görmezden gelmesi nedeni ile, yaygınca kullanılmaktadır. Bu metodun en önemli engeli ise, hafızasında D^k matrisini tutması gerekmektedir.

Quasi-Newton metodunun ana fikri, ardışık iki iterasyonun gradientleri ile beraber, amaç fonksiyon değerlerinin hakkında yaklaşık değerler vermektir:

$$q^k = \nabla^2 f(x^{k+1}) p^k \quad (2.84)$$

$$p^k = x^{k+1} - x^k \quad (2.85)$$

$$q^k = \nabla f(x^{k+1}) - \nabla f(x^k) \quad (2.86)$$

Verilen lineer bağımsız iterasyon artımlarının p^0, \dots, p^{n-1} , bunlara ait gradient artımlarının q^0, \dots, q^{n-1} kullanarak yaklaşık olarak Hessian matrisi hesaplanabilir:

$$\nabla^2 f(x^n) \approx [q^0, \dots, q^{n-1}] [p^0, \dots, p^{n-1}]^{-1} \quad (2.87)$$

Hessian matrisin tersi ise:

$$\nabla^2 f(x^n)^{-1} \approx [p^0, \dots, p^{n-1}] [q^0, \dots, q^{n-1}]^{-1} \quad (2.88)$$

Amaç fonksiyonu kuadratik olarak verilirse, bu bağlantı kesin olmaktadır. Bir çok Quasi-Newton metodunda, D^k matrisi üzerinde bilgileri saklayarak, Hessian matrisin tersini bulunmasına yönelik değişik metodları bulunmaktadır.

En genel halde D^{k+1} matrisi, D^k matrisinden elde edilmektedir:

$$D^{k+1} = D^k + \frac{p^k p^{kT}}{p^{kT} q^k} - \frac{D^k q^k q^{kT} D^k}{q^{kT} D^k q^k} + \zeta^k \tau^k v^k v^{kT} \quad (2.89)$$

Burada v^k ve τ^k aşağıdaki gibi tanımlanır:

$$v^k = D^k + \frac{p^k}{p^{kT} q^k} - \frac{D^k q^k}{\tau^k} \quad (2.90)$$

$$\tau^k = q^{kT} D^k q^k$$

Skalar sayı olan ζ^k , her k için aşağıdaki eşitsizliği sağlamalıdır:

$$0 \leq \zeta^k \leq 1 \quad (2.91)$$

D^0 pozitif tanımlı rastgele matristir. Skalar değer olan ζ^k değerine göre metod değişmektedir. Eğer her k için $\zeta^k = 0$ seçilirse, Davidon-Fletcher-Powell (DFP) metodu, her k için $\zeta^k = 1$ ise, Broyden-Fletcher-Goldfarb-Shanno (BFGS) metodu olmaktadır.

Bu algoritmanın en önemli özelliği ise, kuadratik olarak tanımlı amaç fonksiyonuna uygulandığında, n iterasyon sonucunda Hessian matrisin tersi bulunmaktadır.

$$D^{k+1} q^i = p^i \quad (2.92)$$

$$\nabla^2 f(x^n)^{-1} = D^n = [p^0, \dots, p^{n-1}][q^0, \dots, q^{n-1}]^{-1} \quad (2.93)$$

Tanım: $\{x^k\}$, $\{d^k\}$ ve $\{D^k\}$ Quasi-Newton algoritması ile elde edilmiş dizinler olsun. (2.82)-(2.83), (2.89)-(2.91) denklemleri, aşağıda verilmiş olan kuadratik olarak tanımlı amaç fonksiyonunun minimize edilmesinde kullanılsın:

$$f(x) = \frac{1}{2} x^T Qx + b^T x \quad (2.94)$$

Adım boyutu a^k ise, çizgi minimize kuralı ile bulunsun;

$$f(x^k + a^k d^k) = \min_a f(x^k + a d^k) \quad (2.95)$$

x^0, \dots, x^{n-1} vektörlerinden hiç biri optimal olmadığı varsayılınsın; o zaman;

- d^0, \dots, d^{n-1} vektörleri Q-conjugate

- Sonuç olarak $D^n = Q^{-1}$

Kanıt: Her k için

$$d^{iT} Q d^j = 0 \quad 0 \leq i \leq j \leq k \quad (2.96)$$

$$D^{k+1} Q p^i = p^i \quad 0 \leq i \leq k \quad (2.97)$$

(2.96) ile (2.97) denklemleri yukarıdaki iki koşulu kanıtlamaktadır. $i < n$ olduğu durumda, x^i vektörlerinden hiç biri optimal değildir ve d^i azalma yönünü verir. Ayrıca $p^i \neq 0$ ve $p^i = a^i d^i$, d^0, \dots, d^{n-1} Q-conjugate'tir ve buda p^0, \dots, p^{n-1} lineer bağımsız olduğunu gösterir, sonuç olarak $D^n Q = I$ eşitliği çıkar.

Kanıtlanması gereken ayrı eşitlik ise;

$$D^{k+1} Q p^k = p^k \quad \forall k \quad (2.98)$$

$Q p^k = q^k$ denklemi ve (2.98) denklemini yeniden düzenlenirse:

$$\begin{aligned} D^{k+1} Q p^k &= D^{k+1} q^k \\ &= D^k q^k + \frac{p^k p^{kT} q^k}{p^{kT} q^k} - \frac{D^k q^k q^{kT} D^k q^k}{q^{kT} D^k q^k} + \zeta^k \tau^k v^k v^{kT} q^k \\ &= D^k q^k + p^k - D^k q^k + \zeta^k \tau^k v^k v^{kT} q^k \\ &= p^k + \zeta^k \tau^k v^k v^{kT} q^k \end{aligned} \quad (2.99)$$

$i < k$ için (2.96) ve (2.97) denkleminde,

$$\nabla f(x^{k+1}) = \nabla f(x^{i+1}) + Q(p^{i+1}, \dots, p^k) \quad (2.100)$$

p^i vektörü diğer her vektöre diktir, ayrıca $Q p^0, \dots, Q p^k$ vektörlerinede diktir çünkü

p^0, \dots, p^k Q-Conjugatedir ($p^i = a^i d^i$) ve $\nabla f(x^{i+1})$ 'ye de diktir. Bu sebeple (2.100) denklemi şu duruma gelir:

$$p^{iT} \nabla f(x^{k+1}) = 0 \quad 0 \leq i \leq k \quad (2.101)$$

(2.98) denkleminde,

$$p^{iT} Q D^{k+1} \nabla f(x^{k+1}) = 0 \quad 0 \leq i \leq k \quad (2.102)$$

$p^i = a^i d^i$ ve $d^{k+1} = -D^{k+1} \nabla f(x^{k+1})$ yola çıkarak;

$$d^{iT} Q d^j = 0 \quad 0 \leq i \leq j \leq k \quad (2.103)$$

Quasi-Newton metodunun avantajlarından birisi, doğru arama metodu doğru olduğunda, algoritma eşlenik yön bulmakla beraber, Hessian matrisin tersinin yaklaşık değerini elde etmektedir. Buda hesaplama zamanı bakımından kolaylık sağlamaktadır.

Diğer bir avantajı ise, her iterasyonda n adımı büyük olduğunda hesaplamaya için gereken durum bakımından da iyilik göstermektedir. Eşlenik gradient metodu k. iterasyonda, amaç fonksiyonunun hesabı ve gradientinin bilinmesi gerekmektedir, fakat Hessian matrisin tekil olması durumunda ikinci mertebeden büyük terimler ihmal edilmeme durumu olursa, Q-N metodu burada daha hızlı olarak çalışmaktadır.

2.1.10 Ayrık-zamanlı optimal kontrol problemi

Bu bölümde, ayrık zamanda optimal kontrol probleminin çözüm metodları anlatılacaktır. Aşağıda tanımlı fonksiyonu minize eden, $x = \{x_1, \dots, x_N\}$ $u = \{u_0, \dots, u_{N-1}\}$ bulunması problemini ele alalım.

$$J(x, u) = g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i) \quad (2.104)$$

s.t.

$$x(i+1) = f(x(i), u(i)) \quad i = 0, \dots, N-1 \quad (2.105)$$

$$x_i \in X_i \subset \mathfrak{R}_n \quad i = 0, \dots, N \quad (2.106)$$

$$u_i \in U_i \subset \mathfrak{R}_m \quad i = 0, \dots, N-1 \quad (2.107)$$

Burada u_i kontrol vektörü, $u = \{u_0, \dots, u_{N-1}\}$ kontrol yörüngesi, x_i durum vektörü, $x = \{x_1, \dots, x_N\}$, durum yörüngesi olarak tanımlanır. $x(i+1) = f(x(i), u(i))$ sistem modelini tanımlar. $(x_i, u_i) \in \Omega_i \subset \mathfrak{R}_n \times \mathfrak{R}_m$ ise en genel halde bulunan kısıtlamaları ifade etmektedir. Bu kısımda, kısıtlamasız hal için yukarıda anlatılan algoritmaların nasıl kullanıldığı incelenecek, kısıtlamalı hal ise bir sonraki kısımda anlatılacaktır.

2.1.10.1 İlk merteye gerekli optimal koşulu

2.1.2 kısımda bir optimizasyon problemi için gerekli koşullar incelenmişti. Optimal kontrol problemine ait ise, aşağıdaki gibi tanımlıdır:

$$\nabla J(u^*) = 0 \quad (2.108)$$

$$u^* = \{u_0^*, \dots, u_{N-1}^*\} \quad (2.109)$$

u^* değerinin J amaç fonksiyonunun lokal minimumu olmalıdır. Hamiltonian fonksiyonu cinsinden yazılırsa:

$$H_i(x_i, u_i, p_{i+1}) = g_i(x_i, u_i) + p_{i+1}^T f_i(x_i, u_i) \quad (2.110)$$

Tanım. $u^* = \{u_0^*, \dots, u_{N-1}^*\}$ lokal minimum kontrol dizini, $x^* = \{x_0^*, \dots, x_{N-1}^*\}$ buna ait durum değerleri olsun. (2.110) denkleminde;

$$\nabla u_i H_i(x_i^*, u_i^*, p_{i+1}^*) = 0 \quad i = 0, \dots, N-1 \quad (2.111)$$

$$p_i^* = \nabla H_i(x_i^*, u_i^*, p_{i+1}^*) \quad i = 0, \dots, N-1 \quad (2.112)$$

Terminal kısıt ile;

$$p_N^* = \nabla g_N(x_N^*) \quad (2.113)$$

2.1.10.2 Optimal kontrol için gradient metodu

Kısıtsız optimal kontrol problemleri için uygulanan gradient metodları prensipte tek düzedir. Örnek olarak Zigzag azalma metodu şu formu alır;

$$u_i^{k+1} = u_i^k - a^k \nabla H_i(x_i^k, u_i^k, p_{i+1}^k) \quad i = 0, \dots, N-1 \quad (2.114)$$

Hamiltonian fonksiyonu şu şekilde tanımlıdır:

$$H_i(x_i, u_i, p_{i+1}) = g_i(x_i, u_i) + p_{i+1}^T f_i(x_i, u_i) \quad (2.115)$$

$u^k = \{u_0^k, \dots, u_{N-1}^k\}$ k. kontrol dizini, $x^k = \{x_0^k, \dots, x_N^k\}$ o giriş değerlerine karşılık gelen durum değerleri $p^k = \{p_0^k, \dots, p_N^k\}$ k. costate değerlerini vermektedir.

Verilen u^k için, x^k sistem denkleminde ileri değerleri ile, p^k ise adjoint denklemlerinden geriye doğru gelinerek hesaplanır. (2.114) deki Zigzag azalma metodunda, adım boyutu ise belirlenen metodlar arasından seçilir. Örnek verilirse Armijo kuralıdır.

Her ne kadar zigzag metodu çok basit olsa dahi, optimal kontrol probleminde çözüm hızı çok yavaştır. Buna sebep olabilecek nedenlerinden biriside sistemin unstabil olma olasılığıdır. Bu sebepten dolayı kontrol giriş değerlerini ölçeklendirmek veya, conjugate gradient veya Newton metodu kullanılmalıdır.

2.1.10.3 Optimal kontrol için eşlenik gradient metodu

Kısıtsız optimal kontrol problemleri için uygulanan eşlenik gradient metoduda direk uygulamalıdır. Yön bulma işlemi, o anki gradient ve önceki yönün lineer kombinasyonudur. Bu metodun performansı, diyagonal olarak Hessian matrisinin yakın değerinin öncelikli koşul matrisi olarak kullanılmasıdır.

2.1.10.4 Optimal kontrol için Newton metodu

Newton saf formundaki bir sonraki iterasyon, f amaç fonksiyonunun x^k noktasında etrafında ikinci derecen Taylor serisine açılmış değerinin minimize edilmesi ile elde edilir:

$$f(x^k) + \nabla f(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k) \quad (2.116)$$

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \cdot \nabla f(x^k) \quad (2.117)$$

(2.116) ve (2.117) denklemini optimal kontrol problemine dönüştürüldüğünde, verilen amaç fonksiyonu J kuadratik formda ise,

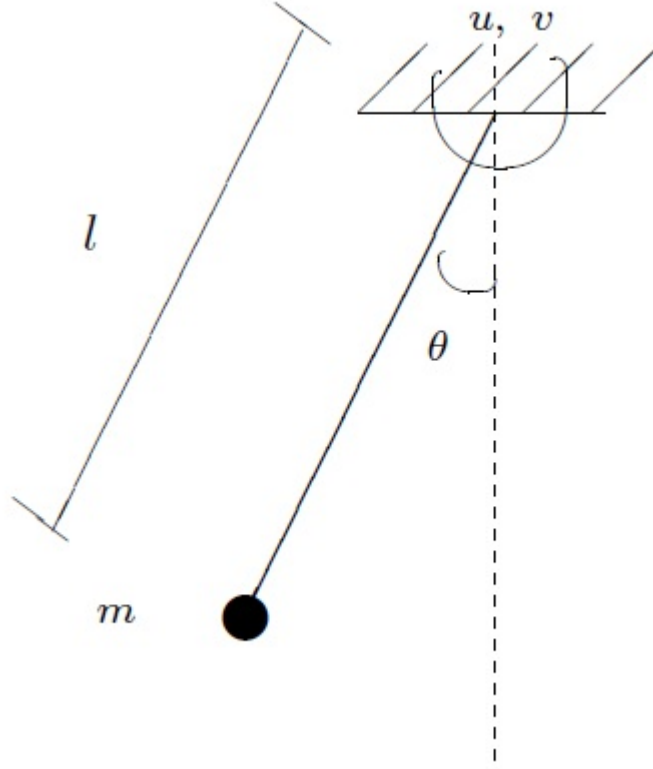
$$J(x, u) = J(x^k) + \nabla J(x^k)(u - u^k) + \frac{1}{2}(u - u^k)^T \nabla^2 J(u^k)(u - u^k) \quad (2.118)$$

$$u^{k+1} = x^k - (\nabla^2 J(x^k))^{-1} \cdot \nabla J(x^k) \quad (2.119)$$

(2.119) ile bir sonraki u optimal kontrol değeri hesaplanır.

2.1.10.5 Optimal kontrol problemi , tek link manipulatör

Günümüzde bir çok alanda kullanılan manipulatörler, kontrol edilmeleri gerekmektedir. Önceki kısımlarda anlatılan nonlinear optimizasyon algoritmalarının tek link manipulatör üzerinde optimal kontrol kanununun bulunması bu konuda işlenecektir.



Şekil 2.4 Tek link manipulatör

Sisteme ait dinamik denklemi yazarsak;

$$\ddot{\theta} = -\frac{g}{l}\sin(\theta) - \frac{v}{ml^2}\dot{\theta} + \frac{1}{ml^2}u \quad (2.120)$$

Burada θ açısal konumu, m linkin sonunda bulunan kütleyi, l linkin uzunluğunu, v bağlantı noktasındaki sürtünme katsayısını, u ise sistemin girişi olan torku belirtir. Durum denklemleri olarak yazarsak $x_1 = \theta$, $x_2 = \dot{\theta}$ olarak seçilir, ve parametre değerleri $m = 2$ kg, $l = 1$ m, $v = 6$ kg m²/s alınır;

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -9.81\sin(x_1) - 3x_2 + 0.5u \end{aligned} \quad (2.121)$$

Euler formülasyonunu kullanarak (2.121) denkleminde tanımlı sistem modelini, ayrık zamanlı hali aşağıda verilmiştir:

$$\begin{aligned} x_1(k+1) &= x_1(k) + hx_2(k) \\ x_2(k+1) &= x_2(k) + h[-9.81\sin(x_1(k)) - 3x_2(k) + 0.5u(k)] \end{aligned} \quad (2.122)$$

Burada k adım sayısını, h örnekleme zamanını göstermektedir.

Tek link manipulatörü, başlangıçta durağan halden, belli bir referans değerine getirmek istendiğini varsayalım, bunun için amaç fonksiyonu şu şekildedir:

$$J(x, u) = g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i) \quad (2.123)$$

s.t.

$$x(i+1) = f(x(i), u(i)) \quad i = 0, \dots, N-1 \quad (2.124)$$

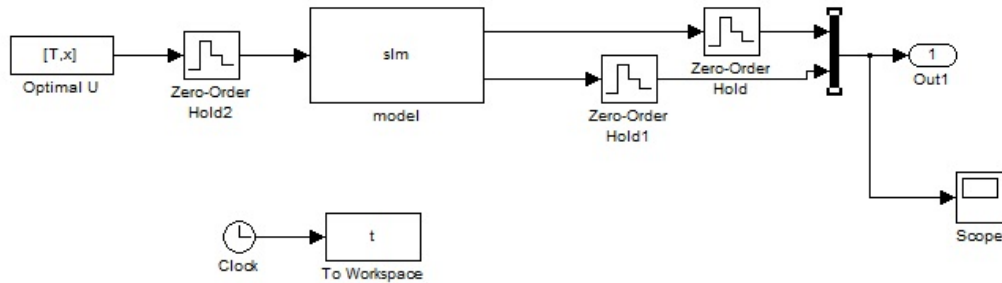
Burada N adım miktarını, $g_N(x_N)$ fonksiyonu, terminal penaltıyı ifade etmektedir. Başlangıç değeri $x_0 = [0,0]'$ olan ve N adım sonunda istenen referans değeri $x_N = [0.4,0]'$ olsun. (2.121) ile tanımlı sistem denkleminde $x_1 = 0.4, x_2 = 0$ çalışma noktasına ait sisteme ait giriş değeri $u=7.6404$ olarak bulunur ve $h=0.05$ ve $N=40$ seçilip, (2.123) deki amaç fonksiyonu tekrar düzenlenirse:

$$J = 50(x_1(40) - 0.4) + \frac{1}{2} \sum_{i=0}^{39} (x_1(k) - 0.4)^2 + x_2^2 + (u(k) - 7.6404)^2 \quad (2.125)$$

s.t.

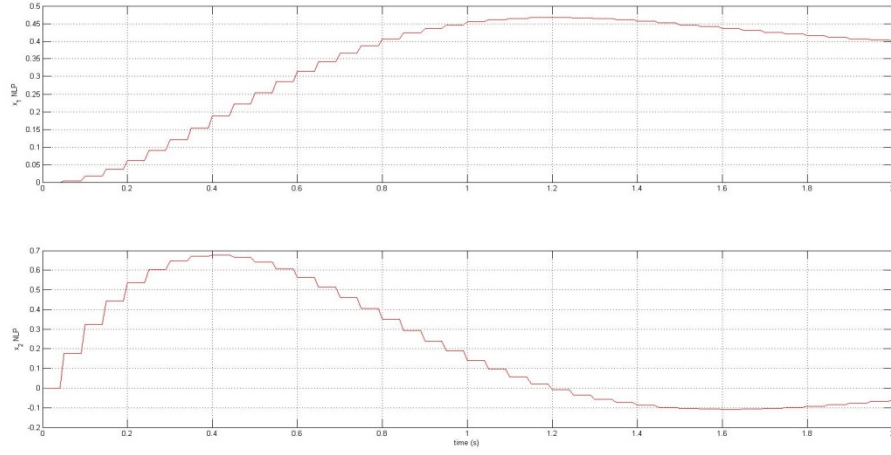
$$\begin{aligned} x_1(k+1) &= x_1(k) + 0.05x_2(k) \\ x_2(k+1) &= x_2(k) + 0.05[-9.81 \sin(x_1(k)) - 3x_2(k) + 0.5u(k)] \end{aligned} \quad (2.126)$$

Olarak seçilir. Sisteme ait blok diyagramı Şekil 2.5'te verilmiştir. Benzetim sonuçları MATLAB ve SİMULİNK programları kullanılarak yapılmıştır. Sistemin fiziksel olarak sürekli (ing. continious) sistem olduğundan, benzetimlerde sistemi ayrık-zamana dönüştürmek için ZOH (zero-order holder) kullanılmıştır.

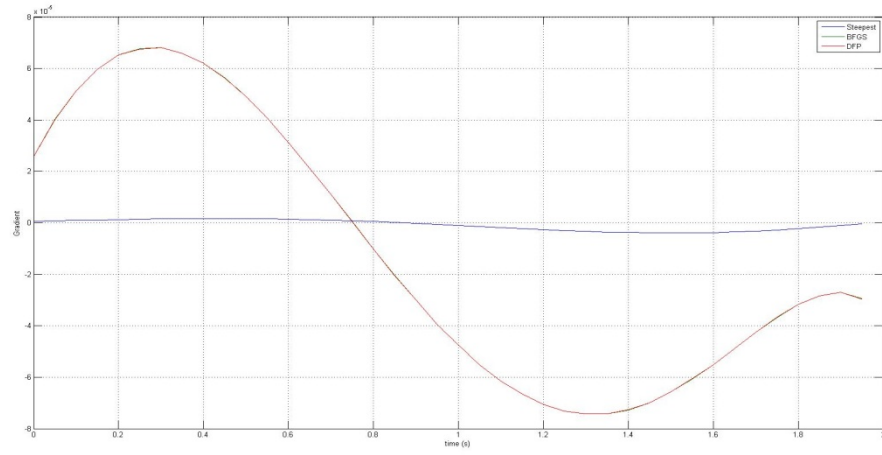


Şekil 2.5 TLM blok diyagramı

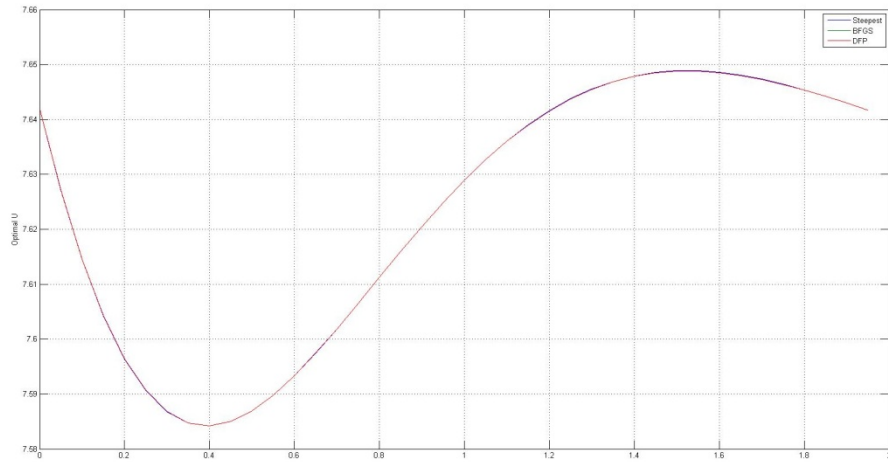
Benzetim Sonuçları



Şekil 2.6 Sistemin Durumları



Şekil 2.7 Gradient

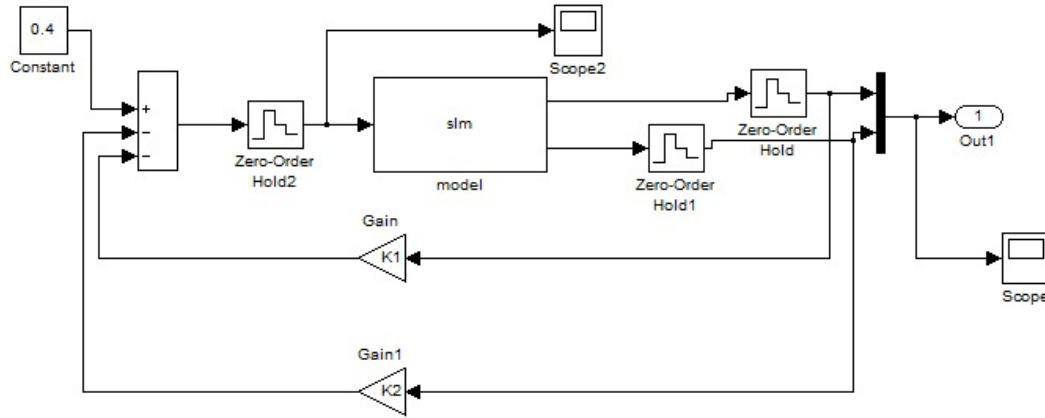


Şekil 2.8 Optimal Giriş Dizini u^*

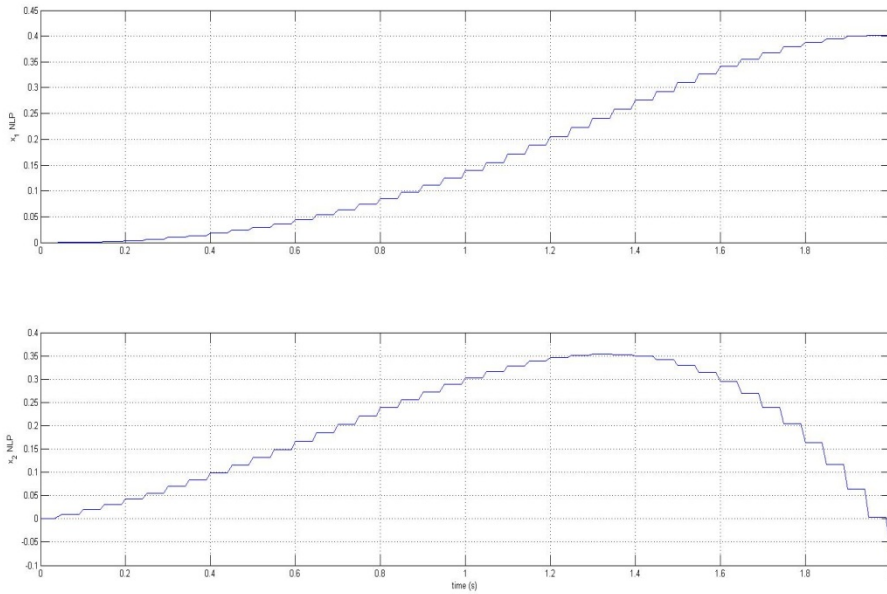
Çizelge 2.1 Algoritmaların karşılaştırılması

Algoritma	Quasi-Newton		Zigzag Metodu
	BFGS	DFP	
İterasyon Sayısı	3	3	5
f(x)	3.0517	3.0517	3.0517
Fun-Count	164	164	246
İlk Mertebe Opt.	7.4277e-005	7.4277e-005	3.9275e-006

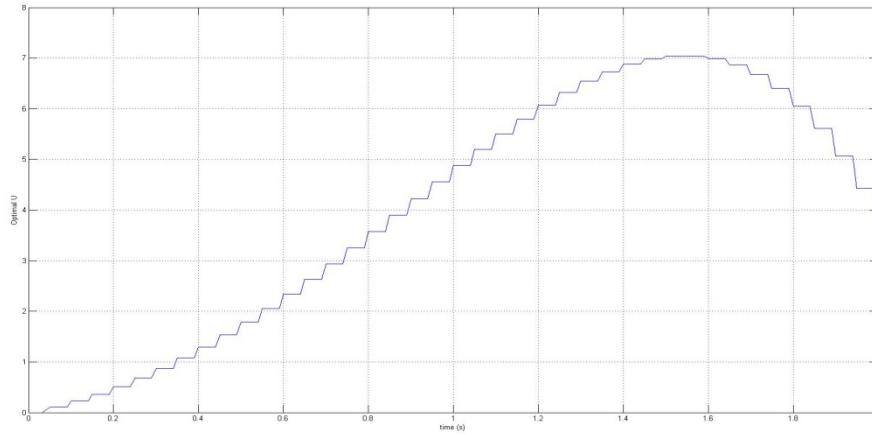
Amaç fonksiyonunu $U=-Kx$, geri besleme kontrol kanununu uygulandığı zaman, minimize edecek minimum K problemine dönüşmektedir. Aşağıda benzetim sonuçları ve blok diyagramı verilmiştir.



Şekil 2.9 Geri beslemeli kapalı çevrim



Şekil 2.10 Sistemin durumları



Şekil 2.11 Optimal giriş dizini u^* ve $K=[-10.9295,-11.0614]$

Benzetim sonuçlarından da görüleceği üzere, kuadratik biçimde verilen bir amaç fonksiyonunun Q matrisi pozitif tanımlı olduğu süreçte, Q-N metodu, Zigzag metodu da aynı sonuçları hemen hemen vermektedir. Kontrol dizini ve o kontrol dizinine ait durum değerleri BFGS ve DFP Hessian matris güncellemesinde aynı sonuçları vermektedir. Konularda da anlatıldığı üzere, Zigzag metodunun en önemli dezavantajı yavaş olmasıdır. **Çizelge 2.1**'deki iterasyon sayısından da görüleceği üzere, Q-N metodu Zigzag metoduna göre daha hızlı çözmektedir.

2.2 Kısıtlı Optimizasyon

Bu kısımda eşitlik veya eşitsizlik kısıtlarına sahip nonlinear optimizasyon problemleri incelenecektir. Genel anlamda, kısıtlı optimizasyon problemlerinde kullanılan algoritmalar, problemi kısıtsız hale getirme veya kısıtları basit ele almaktan geçer. Minimize edilmek istenen amaç fonksiyonu kuadratik olması durumunda, kısıtların lineer ise, bu probleme kuadratik programlama problemi denmektedir. Bu kısımda QP ile sequential QP ve bunlara ait versiyonlar anlatılacaktır.

2.2.1. Eşitlik kısıtı

Optimizasyon problemlerinde eşitlik kısıtı (ing. equality constraint) şu şekilde tanımlanmaktadır:

$$\text{minimum } f(x) \quad (2.127)$$

s.t.

$$c(x_i) = 0 \quad i = 0, \dots, l \quad (2.128)$$

2.2.1.1 Optimal koşulu

Tanım. Eğer x^* (2.128) ile tanımlı kısıtlamaları sağlıyorsa, bu nokta uygun (feasible) denir.

Eğer x^* , (2.127) ve (2.128) denklemlerinin bir çözümü ise, uygun nokta olmak zorundadır. x^* 'nın optimal olma koşulu, amaç fonksiyonu ile kısıtların bir dengesi olarak söylenebilir. x^* 'ya yakın bir nokta, kısıtları çiğnememeli ve f amaç fonksiyonunu değerini arttırmamalıdır. F amaç fonksiyonu ve kısıtların türevi alınabilir ise, Karush-Kuhn-Tucker (KKT) koşulu olarak adlandırılan optimal ilk-mertebe koşulu şu şekilde tanımlıdır:

Tanım. Eğer x^* (2.127) ve (2.128) ya ait lokal çözümü ise, x^* noktası uygun bir nokta olmalıdır ki;

$$c(x_i^*) = 0 \quad i = 0, \dots, l \quad (2.129)$$

Öyle bir skalar sayılar $\lambda_1^*, \dots, \lambda_l^*$ mevcut olmalıdır ki;

$$\nabla f(x^*) - \sum_{i=1}^l \lambda_i \nabla c_i(x^*) = 0 \quad i = 0, \dots, l \quad (2.130)$$

Bu $\lambda_1^*, \dots, \lambda_l^*$ skalar sayılarına Langrange Çarpanları (ing. Langrange Multipliers) denmektedir. $\nabla c(x_1), \dots, \nabla c(x_l)$ vektörü, kısıtların normali olarak adlandırılır.

Tanım. $l \times m$ boyutunda ve $\nabla c(x_1)^T, \dots, \nabla c(x_l)^T$ satırına sahip matris, kısıtların Jacobianı olarak tanımlanır. Eğer N , (2.128) daki kısıtların Jacobianı ise, (2.130) denklemini şu şekilde yazılabilir:

$$\nabla f(x^*) - N^T \lambda_i^* = 0 \quad (2.131)$$

(2.130) ve (2.131) daki koşullar, $\nabla f(x^*)$ 'in kısıtlamaların normali olan N ile lineer bağımlı olduğunu göstermektedir. Bu da, eğer kısıtlı minimum noktası bulunmuş ise, f fonksiyonunu gradientinin ve kısıtlar öyle birbirine bağlıdır ki, o noktadan itibaren amaç fonksiyonunu azaltmak için, kısıtları çiğnenmek zorundadır. (2.131) denklemini bir fonksiyonun gradienti olarak da tanımlanabilir:

$$L(x, \lambda^*) = f(x) - \sum_{i=1}^l \lambda_i c_i(x) \quad i = 0, \dots, l \quad (2.132)$$

$$L(x, \lambda^*) = f(x) - \lambda_i^T c_i(x) \quad i = 0, \dots, l \quad (2.133)$$

$L(x, \lambda^*)$ Langrangian fonksiyonu olarak adlandırılır.

2.2.1.2 Uygun yön ve ikinci merteye koşulu

Tanım. z , n boyutunda, x^* noktasında uygun yön olan n boyutunda bir vektör olsun, ve N (2.129) ile tanımlı kısıtların normali olsun. Eğer $Nz=0$ ise, normalleştirilmiş yönün faredelim ki z , $\|z\|=1$ olsun ve Taylor serisine açılırsa;

$$c(x^* + \varepsilon z) = c(x^*) + \varepsilon Nz + O(\|ez\|^2) \quad (2.134)$$

$c(x^*) = 0$ bilindiği üzere, (2.134) denklemi $c(x^* + \varepsilon z) = O(\|ez\|^2)$ duruma gelir. x^* noktasından z yönünde ilerleme durumunda, kısıtlar ile ilk-mertebeden koşullar sağlanmış olacaktır. Eğer bütün kısıtlar lineer ise, $x^* + \varepsilon z$ bütün ε değerleri için uygun noktadır.

(2.130) koşulu, herhangi bir uygun yön olan z durumunda;

$$z^T \nabla f(x^*) = 0 \quad (2.135)$$

(2.131) ve (2.135) koşulları ilk-mertebeden koşullardır. Eğer x^* noktasının f amaç fonksiyonunun minimum olma durumu için, ikinci-mertebe koşul gereklidir.

Tanım. Bütün kısıtlar lineer ise, ikinci-mertebe koşul, x^* minimum olduğunu garanti eder.

$$z^T \nabla f(x^*) z > 0 \quad (2.136)$$

Kısıtların nonlineer olduğu durumlarda, (2.136) koşulunda, Langrangian fonksiyonunun Hessian matrisi gelmektedir.

$$z^T \nabla L(x^*, \lambda^*) z > 0 \quad (2.137)$$

2.2.1.3 Kuadratik programlama

Kuadratik programlama, (2.127) ile tanımlı amaç fonksiyonunun kuadratik olarak verilmesi ve kısıtların lineer olması durumunda özel halidir. Eşitlik kısıtına sahip

kuadratik programlama (ing. equality constrained quadratic, EQP) problemi aşağıda tanımlanmıştır:

$$\text{minimum } f(x) = \frac{1}{2}x^T Gx + h^T x + c \quad (2.138)$$

s.t.

$$Ax - b = 0 \quad (2.139)$$

Bu probleme ait ilk-mertebe optimal olma koşulu şu şekilde tanımlıdır:

$$Ax^* - b = 0 \quad (2.140)$$

$$Gx^* + h - A^T \lambda^* = 0 \quad (2.141)$$

Tekrar düzenlersek, (2.140) ve (2.141) n+1 boyutunda lineer sistem olarak çözümlenir:

$$\begin{bmatrix} G & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -h \\ b \end{bmatrix} \quad (2.142)$$

(2.142) ile tanımlı lineer sistemin çözüme metodların biriside Gauss Eleme Metodudur.[2.s169]

Eğer G pozitif tanımlı ise, (2.142) 'ın çözümü bize uygun noktayı verecektir, fakat değil ise, x^* 'nın maksimum veya saddle noktası olmadığını kontrol etmek için ikinci-mertebe koşulunu da dikkate almamız gerekmektedir.

$$z^T Gz > 0 \quad \text{her } z \text{ için} \quad \text{st. } Az=0 \quad (2.143)$$

2.2.1.4 İndirilmiş gradient metodu

İndirilmiş gradient metodu (ing. Reduced gradient method) lineer kısıtları kullanarak, bazı değişkenleri egale etme metodudur. Lineer kısıtlamalar şu şekilde tanımlıdır:

$$Ax + b = 0 \quad (2.144)$$

A matrisini iki kısımdan olduğu varsayalım. ($A = [\tilde{A}, \hat{A}]$), \tilde{A} n-1 sütuna, \hat{A} 1 sütünuna sahiptir. x vektörlerini de iki kısma ayırırsak, (2.144) şu şekilde yazılır:

$$\tilde{A}\tilde{x} + \hat{A}\hat{x} = b \quad (2.145)$$

Eğer \hat{A} tekil değilse, tersi alınabilirse;

$$\hat{x} = \hat{A}^{-1}(b - \tilde{A}\tilde{x}) \quad (2.146)$$

Bu durumda, l kadar değişkeni, n-l biçinden yazmış olduk, bu sayede aşağıdaki eşitliğide yazabiliriz:

$$x = v + M\tilde{x} \quad (2.147)$$

v , n boyutunda sütun vektörü, M $n \times (n-l)$ boyutunda matris olup, şu şekilde tanımlıdır:

$$v = \begin{pmatrix} 0_{(n-l) \times 1} \\ \hat{A}^{-1}b \end{pmatrix}, \quad M = \begin{pmatrix} I_{(n-l) \times (n-l)} \\ -\hat{A}^{-1}\tilde{A} \end{pmatrix} \quad (2.148)$$

(2.138)'daki amaç fonksiyonunda yerine yazarsak:

$$\tilde{f}(x) = \frac{1}{2}(\tilde{x}^T M^T + v^T)G(v + M\tilde{x}) + h^T(v + M\tilde{x}) + c \quad (2.149)$$

Yeniden düzenlenirse:

$$\tilde{f}(x) = \frac{1}{2}\tilde{x}^T(M^T GM)\tilde{x} + (v^T GM + h^T M)\tilde{x} + v^T Gv + h^T v + c \quad (2.150)$$

(2.138) denkleminin çözümü, (2.150) ile tanımlı (n-l) kadar \tilde{x} cinsinden f amaç fonksiyonunun kısıtsız hali çözüm problemine dönüşmüştür.

2.2.1.5 Menzil ve boş-uzay

İndirilmiş gradient metodu, A matrisini tersi alınabilir kısımlara ayırmak yerine daha genel biçimde ele alınabilir. A matrisini kısıtlara normal ve teğetsel olarak iki alt bölüme ayrılacaktır.

Eğer n-değişkenli optimizasyon problemi l kadar lineer kısıt içeriyorsa, $Ax+b=0$, doğru yön (ing. feasible direction) $Az=0$ 'ı sağlayan (n-l) boyutundaki z vektörünün herhangi bir alt kümesinde olabilir.

Tanım. $Az=0$ koşulunu sağlayan (n-l) boyutundaki z vektörünün alt kümesi, A'nın boş-uzayı (ing. null-space) olarak adlandırılır.

Z matrisi A'nın null-space'ini sütununda içeren $n \times (n-l)$ matrisi olsun, buda Zw herhangi bir n-l vektörü için doğru yöndür denebilir. Z 'nin seçimi tek değildir, fakat bir metodu, kısıtların Jacobianın'ın orthogonal çarpanlarına ayırma metodudur. Bu çarpanlara ayırma $n \times n$ boyutunda Q, lxl boyutunda triangular L matrisi üretir.

$$AQ = R = (L : 0) \quad \text{ve} \quad Q^T Q = I \quad (2.151)$$

Eğer Y matrisini Q'nun ilk l sütünü ve Z 'nin (n-l) sütünü içeren matris seçilirse;

$$AZ = 0, AY = L \quad \text{ve} \quad Y^T Z = 0 \quad (2.152)$$

Tanım. Y'ye ait l boyutundaki alt küme, kısıt olan A 'nın Jacobian'ın menzil-uzayı (ing. range-space) denir.

Tanım. $Z^T \nabla f(x)$ ve $Z^T \nabla^2 f(x) Z$ sırasıyla, $f(x)$ 'in indirilmiş gradient ve indirilmiş Hessian'ı olarak adlandırılır.

2.2.1.6 İndirilmiş gradient ve QP

Eğer x^* (2.138)'nin çözümü ise, ona ait bileşenleri Y ve Z uzayında bulmamız mümkündür. \tilde{y} l boyutunda, \tilde{z} (n-l) boyutunda vektörler olduğunu varsayalım;

$$x^* = Y\tilde{y} + Z\tilde{z} \quad (2.153)$$

$Y\tilde{y}$ ve $Z\tilde{z}$, bazen dikey adım (ing. vertical step) ve yatay adım (ing. horizontal step) olarak adlandırılır. $AZ=0$ koşulundan dolayı;

$$AY\tilde{y} + b = 0 \quad (2.154)$$

(2.152) 'den $AY=L$, sonucunda \tilde{y} çarpanlarına ayırma metodu ile alt üçgensel sistemleri çözerek bulunabilir.

$$Z^T G Z \tilde{z} = -Z^T h - Z^T G Y \tilde{y} \quad (2.155)$$

Eğer EQP minimumu var ise, buda Hessian matrisi olan $Z^T G Z$ pozitif tanımlı olması demektir ve Cholesky çarpanlarına ayırma metodu ile (2.156) çözülebilir:

$$Y^T A^T \lambda^* = Y^T h + Y^T G x^* \quad (2.156)$$

Langrange çarpanları ile çözülebilir hale gelmiştir.

İndirilmiş gradient yaklaşımı, kısıtların lineer olduğu durumda kuadratik amaç fonksiyonunu minimize etmek için kullanılabilir. Kuadratik olmadığı durumda ise genel formu şu şekildedir:

$$\text{minimum } f(x) \quad (2.157)$$

s.t.

$$Ax + b = 0 \quad (2.158)$$

Kısıtsız optimizasyon problemlerindeki kullanılan algoritmalarda olduğu gibi, kuadratik olarak $f(x)$ fonksiyonunun yaklaşık değerini iteratif olarak çözümlenmelidir.

$$f(x + p) = f(x) + p^T \nabla f(x) + \frac{1}{2} p^T B p \quad (2.159)$$

Burada $B = \nabla^2 f(x)$ veya $B \approx \nabla^2 f(x)$ dır. Kuadratik olarak verilmeyen amaç fonksiyonlarında, fonksiyonunun kuadratik olarak yaklaşımında, Hessian matrisin her iterasyonda yaklaşık değerinin tahmin edilmesine dayanır. Hessian matris güncellenmesinde Q-N algoritmasında anlatıldığı gibi BFGS algoritması üzerine dayanmaktadır. Direk hesaplamak yerine, tahmini değerinin bulunması, ayrıca matrisin pozitif tanımlı olmasını da sağlamaktadır. Bu algoritma şu şekilde tanımlanmıştır:

- Uygun bir başlangıç x^0 değeri seçilir ve $\lambda^0 = 0$ alınır.
- $\nabla^2 f(x)$ 'nin tahmini değeri olan pozitif tanımlı B^0 seçilir.
- Y ve Z matrisleri elde edilir.

$k=0,1,2,\dots$ adım boyunca $g_k = \nabla f(x_k)$ olsun;

- \tilde{z} , $Z^T B_k Z \tilde{z} = -Z^T g_k$ çözümlenerek bulunur ve $d_k = Z \tilde{z}$ olsun;
- $Y^T A^T \lambda = Y^T g_k + Y^T B_k d_k$ çözümlenerek, λ_{k+1} bulunur.
- Uygun bir doğru arama algoritması kullanılarak $x_{k+1} = x_k + \alpha d_k$,
 $f(x_{k+1}) < f(x_k)$ olacak şekilde bulunur
- BFGS algoritması kullanılarak B_{k+1} , B_k 'dan bulunur.

Bu işlem $\|Z^T g_k\|$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

2.2.1.7 Penaltı fonksiyonu metodu

Bu kısımda kısıtların nonlinear olduğu durumlar incelenecektir. Kısıtların nonlinear olma durumu, amaç fonksiyonunun kuadratik olmamasından daha fazla zorluk çıkartmaktadır. Sebep olarak, her iterasyon, doğru alanda kalıp kalınmayacağı garantisini verilememektedir. Bunun için çözüm metodları bu kısımda irdelenmiştir.

İndirilmiş gradient ve nonlinear kısıtlar

Bir önceki bölümde açıklanan indirilmiş gradient metodu, nonlinear kısıtlarıda içeren problemlere uygulanabilmektedir. En zor kısmı ise, d adım yönü , kısıtların $c(x + ad) = 0$ sağlanması garantisini vermemektedir. Bunun için yeni bir straji üzerinde yoğunlaşılmalıdır.

Kısıtlamalara doğru dikey olarak hareketten sonra, yatay yönde hareket edilerek bunun üstesinden gelinebilir ve yeni tanımlanmış adımın ilk tahmini değeri şu şekilde tanımlanarak elde edilir:

$$\hat{c}_i = c_i(x + ad) \quad i = 1, \dots, l \quad (2.160)$$

$AYy = -\hat{c}$ çözümlenerek y bulunur ve $\hat{d} = Yy$ seçilir, ve kısıtlar yaklaşık olarak lineer ise;

$$x^+ = x + ad + \hat{d} \quad (2.161)$$

Başlangıç olarak bir sonraki iterasyon için uygun bir nokta olarak düşünülebilir. Fakat c_i yüksek nonlinearitesine sahipse, uygun adımı bulmak başlı başına bir iterasyon problemine dönüşebilir.

İndirilmiş gradient metodunun bir başka durumu ise, kuadratik olarak tanımlanmayan amaç fonksiyonunu ait minimizasyon probleminde (2.157) nonlinear kısıtların bulunmasında durumunda da kullanılabilir. İkinci-mertebe optimal koşulu olan amaç fonksiyonunun Hessianı yerine Langrangian fonksiyonunun Hessianını kullanma durumunda, B , $\nabla^2 f(x)$ yaklaşık değeri alınmalıdır:

$$\nabla^2 L^* = \nabla^2 f(x) - \sum_{i=1}^l \lambda_i^* \nabla^2 c_i \quad (2.162)$$

Q-N metodu ile B matrisi güncelenecek ise; uygun bir koşul seçilerek;

$$B_{k+1} \delta_k = \gamma_k \quad (2.163)$$

$$\gamma_k = \nabla L(x_{k+1}) - \nabla L(x_k) \quad (2.164)$$

Burada L , L^* değerinin Langrange çarpanlarına dayanan lokal yaklaşık değeridir.

$$L(x) = \nabla f(x) - \sum_{i=1}^l \lambda_{k_i} \nabla c_i(x) \quad (2.165)$$

Penaltı Fonksiyonları

Nonlinear kısıtlardan dolayı oluşabilecek, doğru yön bulamama durumun üstesinden gelmek için, kısıtlı optimizasyon problemleri, öncede belirtildiği üzere, kısıtsız hale dönüştürülerek yapılabilir.

Tanım. (2.157) ve (2.158) ile tanımlı optimizasyon problemine ait penaltı fonksiyonu şu şekilde tanımlı olsun:

$$P(x, r) = f(x) + \frac{1}{r} \sum_{i=1}^l c_i(x)^2, \quad r > 0 \quad (2.166)$$

Burada r , penaltı parametresi olarak tanımlanır. Eğer x doğru bir nokta ise $P(x, r) = f(x)$ olacaktır, eğer değil ise, P , f fonksiyon değerini, kısıtların ihlal edildiği miktarın karesi kadar fazla olacaktır.

Tanım. $F(x)$ fonksiyonu, her x için sınırlandırılmış olsun, ve x^* , kısıtların normalleri olan $\nabla c(x_1), \dots, \nabla c(x_l)$ lineerli bağımsız olmasını sağlayan çözüm olsun. ρ pozitif sayı, ve her $r_k < \rho$, Hessian matrisi olan $\nabla^2 P(x, r_k)$, her x için pozitif tanımlı olsun, bu durumda x_k aşağıda tanımlı kısıtsız optimizasyon probleminin çözümüdür:

$$\text{minimize } P(x, r_k), \quad (2.167)$$

Şu şekilde devam eder;

$$r_k \rightarrow 0, \quad x_k \rightarrow x^* \quad (2.168)$$

$$r_k \rightarrow 0, \quad -\frac{2c_i(x_k)}{r_k} \rightarrow \lambda_i^* \quad (2.169)$$

Kanıt. Yukarıda adı geçen tanımın geçerli olmadığını, ε pozitif sabit bir sayı olan ile varsayalım:

$$\sum_{i=1}^l c_i(x_k)^2 > \varepsilon, \forall r_k \quad (2.170)$$

Buda bize;

$$P(x_k, r_k) > f(x_k) + \frac{1}{r_k} \varepsilon, \quad (2.171)$$

f^* , $f(x)$ fonksiyonunun çözüm alanı içinde minimum değeri olduğunu varsayalım.

x_k , $P(x_k, r_k)$ 'nin tek çözümü olduğundan;

$$P(x_k, r_k) \leq f^* \quad (2.172)$$

$$f(x_k) + \frac{1}{r_k} \varepsilon < f^*, \quad (2.173)$$

Yeniden düzenlenirse;

$$f(x_k) < f^* - \frac{1}{r_k} \varepsilon, \quad (2.174)$$

(2.174) terimi bize gösterir ki, $r_k \rightarrow 0$ giderken, $f(x_k)$ değeri çok yüksek ve negatif olabileceğini göstermektedir, buda alt değerinde limit olarak kısıtlı olan $f(x)$ fonksiyonun kısıtı ile çakışmaktadır, bu öneri yanlıştır. $r_k \rightarrow 0$

$$c_i(x_k) \rightarrow 0 \quad i = 1, \dots, l \quad (2.175)$$

Her x_k kısıtsız minimum problemin çözümü ise;

$$\nabla P(x_k, r_k) = \nabla f(x_k) + \frac{1}{r_k} \sum_{i=1}^l 2c_i(x_k) \nabla c_i(x_k) = 0 \quad (2.176)$$

$$\lambda_i(x_k) = -\frac{2}{r_k} c_i(x_k) \quad (2.177)$$

Olarak tanımlarsak, (2.176) denklemini tekrar düzenlenirse:

$$\nabla P(x_k, r_k) = \nabla f(x_k) + \sum_{i=1}^l \lambda_i(x_k) \nabla c_i(x_k) = 0 \quad (2.178)$$

Olarak çözülebilir.

- Uygun bir başlangıç x^0 değeri seçilir.

- Penaltı parametresi r_1 ve sabit bir sayı $\beta < 1$ seçilir.

$k=1,2,\dots$

- x_{k-1} den başlayarak iteratif olarak x_k (2.166) denklemi için çözülür.
- $r_{k+1} = \beta r_k$ olarak seçillir.

Bu işlem $\|c(x_k)\|$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

Arttırılmış Langrangian

Penaltı fonksiyonunda r_1 değerini küçük seçerek, kısıtsız optimizasyon algoritmasının yaklaşık sonuç vereceğini umarak hızlandırmayı düşünebiliriz. Pratikte ise, bu mümkün olmamaktadır.

r sıfıra yakın değerler seçilirse, $P(x, r)$, ikinci terimi baskın hale gelmekte ve buda amaç fonksiyonu ile çakışma meydana getirebilmektedir. Ayrıca nümerik olarak elde edilen ∇P ve $\nabla^2 P$ hatalı olabilir.

Bu tür nümerik zorlukların üstesinden gelmek için, r çok küçük değerleri ile çalışırken $c_i(x)$ çoktan sıfıra yakın değerlerde olmalıdır. Bunun için arttırılmış Langrangian fonksiyonu kullanılmaktadır ve şu şekilde tanımlıdır:

$$M(x, r, v) = f(x) + \frac{1}{r} \sum_{i=1}^l (c_i(x) - \frac{r}{2} v_i)^2 \quad (2.179)$$

$P(x, r)$ ile kıyaslanırsa, fazladan v_1, \dots, v_l parametleri bulunmaktadır, (2.179) daha açık bir şekilde yazılırsa:

$$M(x, r, v) = f(x) - \sum_{i=1}^l v_i c_i(x) + \frac{1}{r} \sum_{i=1}^l c_i(x)^2 + \frac{r}{4} \sum_{i=1}^l v_i^2 \quad (2.180)$$

Arttırılmış Langrangian (ing. Augumented Langrangian) metodunu kısıtsız ardışık minimizasyonda (sequential unconstrained minimization) uygulanmasında, v parametrelerinin tayini gerekmektedir ki, bu sayede Langrange çarpanlarına yönelmesi sağlanmalıdır.

- Uygun bir başlangıç x^0 değeri seçilir.
- Penaltı parametresi r_1 ve sabit bir sayı $\beta < 1$ seçilir.

$k=1,2,\dots$

- x_{k-1} den başlayarak iteratif olarak x_k (2.179) denklemi için çözülür.
- $v_{k+1} = v_k - 2c(x_k)/r_k$ ve $r_{k+1} = \beta r_k$ olarak seçilir.

Bu işlem $\|c(x_k)\|$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

Tam penaltı fonksiyonu

Şu ana kadar anlatılan penaltı fonksiyon metodlarında, kısıtlı optimizasyon problemini, ardışık kısıtsız optimizasyon problemine dönüştürülmektedir. Bunun haricinde, bir tek kısıtsız minimizasyon ile bu problem çözülebilmektedir. Bu tür fonksiyona tam penaltı fonksiyonu (ing. Exact penalty function) adı verilmektedir, ve şu şekilde tanımlıdır:

$$E(x, r) = f(x) + \frac{1}{r} \left\{ \sum_{i=1}^l |c_i(x)| \right\} \quad (2.181)$$

Buna penaltı fonksiyonu adı verilir ve her r yeteri kadar küçük değeri için x^* minimum değeri bulunmaktadır. Bu kısımda M ve P deki gibi parametre ayarlaması yapmaya gerek yoktur.

(2.181) de tanımlı E fonksiyonu pürüzlü (ing. nonsmooth) bir fonksiyon olup, türevleri $c(x)=0$ olduğu yerlerde süreksizlik göstermektedir. Buda birçok kısıtsız minimizasyon algoritmalarında sorun yaratabilmektedir. Bunun için biraz daha düzgün (ing. smooth) bir fonksiyon olan şu denklem tanımlanmıştır:

$$E'(x, r) = f - c^T (AA^T)^{-1} Ag + \frac{1}{r} c^T c \quad (2.182)$$

c , kısıt vektörü, g , f fonksiyonuna ait gradient vektörü, A , kısıtların normal vektörünün Jacobianı olarak tanımlıdır. Bu şekilde tanımlanmış E' fonksiyonu pratikte daha avantajlıdır. Sağ taraf amaç fonksiyonunun ve kısıtların ilk türevlerini, ve ikinci türevlerini bulundurmaktadır. Bu sayede Gradient methodu kullanılabilir.

2.2.1.8 Ardışık kuadratik programlama

Ardışık kuadratik programlama (ing. sequential quadratic programming, SQP) da iki türlü aşama geliştirilmiştir. Birincisi, QP kullanarak c kısıt vektörünün lineerize edilmiş ve F fonksiyonunun kuadratik formu alınarak, (2.127) ve (2.128) problemlerine uygulanması işlemi, diğeri ise, QP alt problemlerini Augmented Langrangian minimize hesaplamalarına uygulamaktır.

Kuadratik / lineer model

$$g(x) = \nabla f(x), G(x) = \nabla^2 f(x), c = (c_1, \dots, c_m)^T \quad (2.183)$$

Ve A matriside i. sırası $\nabla c_i(x)^T$ olan matristir.

(2.127) ve (2.128) ' da tanımlı eşitlik kısıtı bulunan probleme ait ilk mertebe optimal koşulu:

$$g(x^*) - \sum_{i=1}^l \lambda_i^* \nabla c_i(x) = 0 \quad \text{ve} \quad c_i(x^*) = 0 \quad i = 1, \dots, l \quad (2.184)$$

Eğer x ve λ , x^* ve λ^* öngörü değerleri ise, ölçüm hatası eklenebilir;

$$T(x, \lambda) = \left\| g(x) - \sum_{i=1}^l \lambda_i \nabla c_i(x) \right\| + \aleph \|c_i(x)\| \quad (2.185)$$

Burada \aleph pozitif ağırlık parametresi olarak tanımlanır.

Farz edelim ki $\delta x = x^* - x$ olsun, δx ve λ^* şu denklemi sağlamalıdır:

$$g(x + \delta x) - \sum_{i=1}^l \lambda_i^* \nabla c_i(x + \delta x) = 0 \quad (2.186)$$

$$c_i(x + \delta x) = 0 \quad i = 1, \dots, l \quad (2.187)$$

Taylor açılımının ilk değerini kullanılırsa;

$$g(x) + G(x)\delta x - \sum_{i=1}^l \lambda_i^* \{ \nabla c_i(x) + \nabla^2 c_i(x)\delta x \} = 0 \quad (2.188)$$

$$c_i(x) + \nabla c_i(x)^T \delta x = 0 \quad i = 1, \dots, l \quad (2.189)$$

$$\tilde{G} = G(x) - \sum_{i=1}^l \lambda_i^* \nabla^2 c_i(x) \quad (2.190)$$

Olarak tanımlanırsa,

$$\tilde{G}\delta x - A^T \lambda^* = -g \quad (2.191)$$

$$-A\delta x = c \quad (2.192)$$

(2.191) ve (2.192) denklemlerini, kuadratik programlamada da gösterildiği gibi, optimal olma koşuludur, bu yüzden tekrar düzenlenirse;

$$\text{mimimize } \frac{1}{2}(\delta x^T \tilde{G} \delta x) + g^T \delta x \quad (2.193)$$

s.t.

$$A\delta x + c = 0 \quad (2.194)$$

δx ve λ^* EQP problemini çözerek bulunabilir. (2.193) 'de tanımlı amaç fonksiyonunda, normal amaç fonksiyonunun gradientini, kısıtların ikinci mertebeden türevininin yakınsama değerini içermektedir.

EQP algoritması genel eşitlik kısıtlı minimize problemlerinde, yön bulunması için kullanılmaktadır. Bu versiyonunda, ikinci türev olan Hessian matrisin hesaplanmasından ziyade, öngörüsü alınmaktadır.

- Uygun bir başlangıç x^0 ve B^0 değeri seçilir..

$k=0,1,2,\dots$

- d_k ve λ_{k+1} alt QP problemi çözülerek bulunur.

$$\text{mimimize } \frac{1}{2}(d^T B_k d) + \nabla f(x_k) d$$

s.t

$$c_i(x_k) + \nabla c_i(x_k)^T d = 0 \quad i = 1, \dots, l$$

- Bir sonraki çizgi arama metodları kullanılarak bir sonraki nokta bulunur.

$$x_{k+1} = x_k + ad_k$$

- B_{k+1} , Q-N metodu ile B_k yardımıyla bulunur.

Bu işlem (2.185)'de verilen $T(x_{k+1}, \lambda_{k+1})$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

Penalti fonksiyonu ile SQP

Bir önceki durumda anlatılan SQP algoritmasında, QP nin çözümü olan yön ile doğru arama metodunda kullanılan penalti fonksiyonu arasında bir bağlantı bulunmamaktaydı. Bu kısımda ise, QP alt problemi, **(2.180)**' da tanımlı arttırılmış Langrangian fonksiyonunun minimum değerini çözer. $\nabla M(x, r, v)$ x noktasında Taylor serisine açılırsa;

$$\nabla M(x + \delta x, r, v) = g - A^T v + \frac{2}{r} A^T c + \left(\tilde{G} + \frac{2}{r} A^T A \right) \delta x + O(\|\delta x\|^2) \quad (2.195)$$

Burada;

$$\tilde{G} = \nabla^2 f(x) - \sum_{i=1}^l \nabla^2 c_i(x) v_i + \frac{2}{r} \left[\sum_{i=1}^l \nabla^2 c_i(x) c_i(x) \right] \quad (2.196)$$

Eğer $x = x^*$ ve $v = \lambda^*$, bütün kısıtlar sıfır olacağından $c_i(x^*) = 0$;

$$\tilde{G} = \nabla^2 f(x^*) - \sum_{i=1}^l \lambda_i^* \nabla^2 c_i(x^*) \quad (2.197)$$

Eğer $x + \delta x$ M fonksiyonunu minimize ediyorsa, **(2.195)** ile tanımlı fonksiyonun sol tarafı sıfır olur, yüksek mertebeden terimleri ihmal edersek;

$$\left(\tilde{G} + \frac{2}{r} A^T A \right) \delta x = -g + A^T v + \frac{2}{r} A^T c \quad (2.198)$$

(2.198)'in çözümü δx , M'in minimum yönündeki giden Newton adımını vermektedir. Yeniden düzenlenirse;

$$u = v - \frac{2}{r} (A \delta x + c) ,$$

$$A \delta x = \frac{r}{2} (u - v) - c \quad (2.199)$$

Olarak yazılırsa, **(2.198)** denklemini daha sade hale getirilir;

$$\tilde{G} \delta x - A^T u = -g \quad (2.200)$$

(2.199) ve **(2.200)** incelenirse, δx ve u, EQP nin Langrange çarpanları olduğu görülmektedir. Aşağıdaki şekilde genel hali yazılabilir [Kanıt:2.s201];

$$\text{mimize } \frac{1}{2} (d^T H^{-1} d) + d^T \nabla f(x_k) \quad (2.201)$$

s.t.

$$A(x_k)d + c(x_k) = -\frac{r}{2}(u_k - \lambda_k) \quad (2.202)$$

- Uygun bir başlangıç x^0 , λ^0 r^0 değeri seçilir..
- Uygun bir H^0 matrisi seçilir.
- $\beta < 1$ olamk üzere uygun bir ölçeklendirme faktörü seçilir ve $\mu = 0, T^- = T(x, \lambda_0)$ seçilir.

k=0,1,...

- $c_k = c(x_k)$, $g_k = G(x_k)$, $A_k = A(x_k)$ hesaplanır.
- d_k ve u_k (2.203) ve (2.304) denklemlerinden hesaplanır.

$$\left(\frac{r_k}{2}I + A_k H_k A_k^T\right)u_k = A_k H_k g_k - c_k + \frac{r_k}{2} \lambda_k \quad (2.203)$$

$$d_k = H_k (A_k^T u_k - g_k) \quad (2.204)$$

- Çizgi arama metodları kullanılarak $M(x_{k+1}, \lambda_k, r_k) < M(x_k, \lambda_k, r_k)$ sağlayacak şekilde, bir sonraki nokta bulunur. $x_{k+1} = x_k + ad_k$
- Eğer $T(x_k, u_k) < T^-$ ise;

$$r_{k+1} = \beta r_k, \lambda_{k+1} = u_k \text{ ve } T^- = T(x_{k+1}, \lambda_{k+1})$$

- Değil ise;

$$r_{k+1} = r_k, \lambda_{k+1} = \lambda_k$$

- H_{k+1} , Q-N metodu ile H_k yardımıyla bulunur.

Bu işlem (2.185)'de verilen $T(x_{k+1}, \lambda_{k+1})$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

2.2.2 Eşitsizlik kısıtı

Bir önceki bölümde eşitlik kısıtı olduğunda çözüm yöntemleri üzerinde duruldu. Bu bölümde ise eşitsizlik kısıtlaması içeren optimizasyon problemleri incelenecektir, Genel hal şu şekilde tanımlıdır;

$$\text{mimimum } f(x) \quad (2.205)$$

s.t.

$$c_i(x) = 0 \quad i = 0, \dots, l \quad (2.206)$$

$$c_i(x) \geq 0 \quad i = l+1, \dots, m \quad (2.207)$$

Tanım. Eğer f amaç fonksiyonu ve c_i lineer ise, bu lineer programlama problemidir. Bu sonraki kısımlarda LQR ve MPC bakış açısıyla ele alınacaktır.

Tanım. Eğer f fonksiyonu kuadratik ve c_i lineer bir fonksiyon ise, bu kuadratik programlama problemidir, Bir önceki kısımlarda eşitlik kısıtı dahilinde anlatılmıştır.

2.2.2.1 Optimal koşulu

Tanım. Eğer x^* (2.206) ve (2.207) denklemlerini sağlıyorsa, düzgün bir nokta, değilse düzgün olmayan nokta denir.

Tanım. Eğer x^* (2.205)-(2.207) denklemlerinin lokal bir çözümü ise, birinci mertedeben optimal olma koşulu şu şekildedir:

$$c(x_i^*) = 0 \quad i = 0, \dots, l \quad (2.208)$$

$$c(x_i^*) \geq 0 \quad i = l+1, \dots, m \quad (2.209)$$

Öyle bir skalar sayılar $\lambda_1^*, \dots, \lambda_l^*$ mevcut olmalıdır ki;

$$\nabla L(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) = 0 \quad i = 0, \dots, m \quad (2.210)$$

$$\lambda_i^* c_i(x^*) = 0 \quad i = 0, \dots, m \quad (2.211)$$

$$\lambda_i^* \geq 0 \quad i = 0, \dots, m \quad (2.212)$$

2.2.2.2 Eşitsizlikleri eşitlik haline getirme

Eşitsizlik kısıtlara ait problem çözümlerine değinmeden, eşitsizlik kısıtlarının eşitlik kısıtlar haline nasıl getirebileceği bu konuda anlatılacaktır. Eşitsizlik kısıtına ait bir problem, fazladan bir değişken tanımlayarak eşitli kısıtlı hale getirilebilir.

$$\text{mimimum } f(x) \quad \text{s.t.} \quad c_i(x) \geq 0 \quad i = 0, \dots, m \quad (2.213)$$

Yeni bir deęişken tanımlayarak;

$$\text{mimimum } f(x) \quad \text{s.t.} \quad c_i(x) - w_i^2 = 0 \quad i = 0, \dots, m \quad (2.214)$$

2.2.2.3 Kuadratik programlama

(2.205)-(2.207) eşitsizlik kısıtına sahip kuadratik programlama (ing. inequality quadratic programming) problemidir. EQP için kullanılan optimal olma koşulu, bu problem içinde kullanılabilir. Problem aşağıdaki gibi tanımlı olsun:

$$\text{mimimum } \frac{1}{2} x^T G x + h^T x + c \quad (2.215)$$

s.t.

$$A x + b \geq 0 \quad (2.216)$$

Kısıtların aktif seti tanımlanmalıdır. Varsayalım ki, t kadar sıra A ve b'ye ait aktif kısıtlar olsun. Bu aktif kısıtlar, \tilde{A} ve \tilde{b} ile tanımlansın. Eğer bu aktif kısıtları, diğer kısıtları ihmal ederek, eşitlik kısıtı olarak varsayarak, (2.215) ile tanımlı amaç fonksiyonunu aşağıdaki eşitliğe göre minimize edersek, sonuç olan $(\tilde{x}, \tilde{\lambda})$ bulunur:

$$\tilde{A} \tilde{x} + \tilde{b} = 0 \quad (2.217)$$

Bu çözümü bir önceki bölümde anlatıldığı üzere eşitlik kısıtı bulunan EQP problemi olarak ele alınır ve çözülür. Eğer aşağıdaki eşitsizlik sağlanırsa;

$$A \tilde{x} + b \geq 0 \quad (2.218)$$

Yani \tilde{x} , hiçbir aktif olmayan kısıtları ihlal etmez ise ve;

$$\tilde{\lambda}^T (A \tilde{x} + b) = 0 \quad \text{ve} \quad \tilde{\lambda} \geq 0 \quad (2.219)$$

Durumunda, optimal olma koşulu sağlanmış olur ve çözüm şu şekildedir, $x^* = \tilde{x}$, $\lambda^* = \tilde{\lambda}$. Fakat eğer $(\tilde{x}, \tilde{\lambda})$ yukarıda ki kısıtlamaları sağlamıyorsa, aktif küme değiştirilip tekrar EQP problemi çözülmelidir.

Yeni aktif küme seçiminde ise, iki yöntem mevcuttur. İlki \tilde{x} değerinin ihlal ettiği kısıtlar, mevcut olan aktif kümeye eklenebilir. Diğerisi ise, $\tilde{\lambda}$ 'nin negatif değerlerine karşılık gelen aktif kısıtlar, mevcut aktif kümeden silinebilir.

2.2.2.4 İndirilmiş gradient metodu

Önceki kısımda anlatılan indirilmiş gradient metodunu, aktif küme stratejisi ile birleştirip, eşitsizlik kısıtlara sahip amaç fonksiyonu kuadratik tanımlı olmayan problemlere uygulanmasına ait algoritma aşağıda tanımlanmıştır:

- Uygun bir başlangıç x^0 değeri seçilir ve $\lambda^0 = 0$ alınır.
- $\nabla^2 f(x)$ 'nin tahmini değeri olan pozitif tanımlı B^0 seçilir.

$k=0,1,2,\dots$ adım boyunca $g_k = \nabla f(x_k)$ olsun;

- Aktik küme seçilir , $c_i(x_k) = 0$ ve $\lambda_{k_i} \geq 0$
- x_k daki aktif kısıtlamaları belirten A_k matris elde edilir;
- Y_k ve Z_k hesaplanır.
- \tilde{z} , $Z^T B_k Z \tilde{z} = -Z^T g_k$ çözümlenerek bulunur ve $d_k = Z \tilde{z}$ olsun;
- $Y^T A^T \lambda = Y^T g_k + Y^T B_k d_k$ çözümlenerek, λ_{k+1} bulunur.
- Uygun bir doğru arama algoritması kullanılarak $x_{k+1} = x_k + \alpha d_k$,
 $f(x_{k+1}) < f(x_k)$ olacak şekilde bulunur
- BFGS algoritması kullanılarak B_{k+1} , B_k 'dan bulunur.

Bu işlem $\|Z^T g_k\|$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

Bu algoritmada önemli olan, çizgi arama adımının, yeni kısıtların ihlaline yer vermemelidir. Adım boyutu bu yüzden bir üst limiti bulunmaktadır. Bu algoritma nonlinear eşitsizliklere de uygulanabilmektedir.

2.2.2.5 Basit sınırlar için indirilmiş gradient metodu

Özel durumlarda Z 'nin elde edilişi kolaydır. Bu durumlar eşitlik içermeyen ve eşitsizlikler sadece üst ve alt limit olarak tanımlanması durumunda geçerli olup aşağıda belirtilmiştir:

$$l_i \leq x_i \leq m_i \quad \text{ve} \quad i = 1, \dots, n \quad (2.220)$$

Yöntemin ana prensibi, x_k 'nın sabit seçilip, sınırlarda olduğu varsayılmasıdır.

$$x_i = l_i \text{ ve } g_{k_i} > 0 \quad \text{veya} \quad x_i = m_i \text{ ve } g_{k_i} < 0 \quad (2.221)$$

$$\sigma_i = \begin{cases} (m_i - x_{k_i}) / d_{k_i} & \text{eğer } d_{k_i} > 0 \\ (l_i - x_{k_i}) / d_{k_i} & \text{eğer } d_{k_i} < 0 \end{cases} \quad (2.222)$$

Bu yönteme ait algoritma şu şekildedir:

- Uygun bir başlangıç x^0 değeri seçilir.
- $\nabla^2 f(x)$ 'nin tahmini değeri olan pozitif tanımlı B^0 seçilir.

$k=0,1,2,\dots$ adım boyunca $g_k = \nabla f(x_k)$ ve Z_k n x n birim matris olsun.

$i=1,\dots,n$ için,

- x_k, Z_k 'ya ait i. sütunda yer alıyorsa,

$$\tilde{z}, Z^T B_k Z \tilde{z} = -Z^T g_k \text{ çözümlenerek bulunur ve } d_k = Z \tilde{z} \text{ olsun;}$$

Uygun bir doğru arama algoritması kullanılarak $x_{k+1} = x_k + \alpha d_k$,
 $f(x_{k+1}) < f(x_k)$ olacak şekilde bulunur

- Bu işlem her x_k için yapılır.
- (2.222)'ten σ_i hesaplanır ve $\alpha = \min(\alpha, \sigma_i)$ olarak alınır.

$$x_{k+1} = x_k + \alpha d_k$$

- BFGS algoritması kullanılarak B_{k+1}, B_k 'dan bulunur.

Bu işlem $\|Z^T g_k\|$ değeri kullanıcı tarafından belirtilen toleransdan küçük olasıya kadar devam eder.

2.2.2.6 Eşitsizlik kısıtı için penaltı fonksiyonları

Eğer penaltı fonksiyonunu değiştirilir ise, eşitlik kısıtı için uygulanan bu yöntem, aynı zamanda, eşitsizlik kısıtı içinde kullanılabilir. Bunun için düzenlenmiş penaltı fonksiyonu şu şekilde tanımlıdır:

$$P(x, r) = f(x) + \frac{1}{r} \left\{ \sum_{i=1}^l c_i(x)^2 + \sum_{i=l+1}^m \min[0, c_i(x)]^2 \right\} \quad (2.223)$$

İlk penaltı terimi, eşitlik kısıtları ile ilgilenmekte, ikinci kısım ise ihlal edilen kısıtlamalarla ilgilenmektedir.

2.2.2.7 Eşitsizlik kısıtı için arttırılmış Langrangian

r sıfıra yakın değerler seçilirse, $P(x, r)$, ikinci terimi baskın hale gelmekte ve buda amaç fonksiyonu ile çakışma meydana getirebilmektedir. Ayrıca nümerik olarak elde edilen ∇P ve $\nabla^2 P$ hatalı olabilir.

Bu tür nümerik zorlukların üstesinden gelmek için, r çok küçük değerleri ile çalışırken $c_i(x)$ çoktan sıfıra yakın değerlerde olmalıdır. Bunun için attrılmış Langrangian fonksiyonu kullanmaktadır ve eşitsizlik kısıtı olduğunda şu hali almaktadır:

$$M(x, r, v) = f(x) + \frac{1}{r} \left\{ \sum_{i=1}^l (c_i(x) - \frac{r}{2} v_i)^2 + \sum_{i=l+1}^m \min[0, c_i(x) - \frac{r}{2} v_i]^2 \right\} \quad (2.224)$$

$$v_{k+1,i} = \begin{cases} v_{k,i} - \frac{2}{r_k} c_i(x_k) & \text{eğer } c_i(x_k) < \frac{r_k}{2} v_{k,i} \\ 0 & \text{Diğer} \end{cases} \quad (2.225)$$

2.2.2.8 Eşitsizlik kısıtı için SQP

Bir önceki kısımda anlatılan SQP algoritmasına, lineerleştirerek eşitsizlik kısıtlarını da eklemek mümkündür. d_k ve λ_{k+1} QP alt problemini çözülerek bulunur:

$$\text{mimize } \frac{1}{2} (d^T H^{-1} d) + \nabla f(x_k)^T d \quad (2.226)$$

s.t.

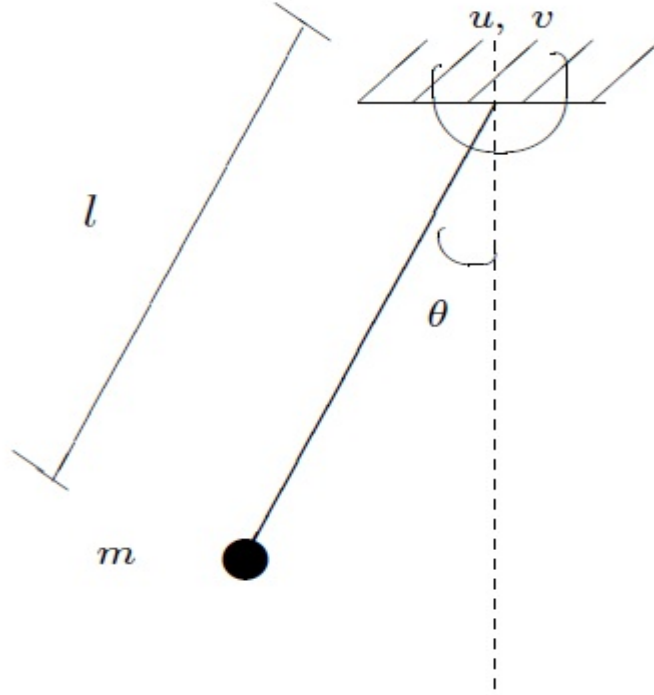
$$c_i(x_k) + \nabla c_i(x_k)^T d = 0 \quad i = 1, \dots, l \quad (2.227)$$

$$c_i(x_k) + \nabla c_i(x_k)^T d \geq 0 \quad i = l+1, \dots, l+m \quad (2.228)$$

Geri kalan hesaplamalar, 2.2.1.8 te anlatılan algoritması ile aynıdır.

2.2.2.9 Kısıtlı optimal kontrol problemi , tek link manipulatör

2.1.10.5 deki kısıtsız optimal kontrol problemine ait tek link manipulatör sistemine kısıtlar eklenmiş ve yukarıda anlatılan algoritmalarla SQP, MATLAB ve SİMULİNK programları aracılığıyla çözülmüştür.



Şekil 2.12 Tek Link Manipulatör

Sisteme ait dinamik denklemi yazarsak;

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) - \frac{v}{ml^2} \dot{\theta} + \frac{1}{ml^2} u \quad (2.229)$$

Burada θ açısal konumu, m linkin sonunda bulunan kütle, l linkin uzunluğunu, v bağlantı noktasındaki sürtünme katsayısını, u ise sistemin girişi olan torku belirtir. Durum denklemleri olarak yazarsak $x_1 = \theta$, $x_2 = \dot{\theta}$ olarak seçilir, ve parametre değerleri $m = 2 \text{ kg}$, $l = 1 \text{ m}$, $v = 6 \text{ kg m}^2/\text{s}$ alınır;

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -9.81 \sin(x_1) - 3x_2 + 0.5u \end{aligned} \quad (2.230)$$

Euler formülasyonunu kullanarak (2.229) denkleminde tanımlı sistem modelini, ayrık zamanlı hali aşağıda verilmiştir:

$$\begin{aligned} x_1(k+1) &= x_1(k) + hx_2(k) \\ x_2(k+1) &= x_2(k) + h[-9.81 \sin(x_1(k)) - 3x_2(k) + 0.5u(k)] \end{aligned} \quad (2.231)$$

Burada k adım sayısını, h örnekleme zamanını göstermektedir.

Tek link manipulatörü, başlangıçta durağan halden, belli bir referans değerine getirmek istendiğini varsayalım, bunun için amaç fonksiyonu şu şekildedir:

$$J(x, u) = g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i) \quad (2.232)$$

s.t.

$$x(i+1) = f(x(i), u(i)) \quad i = 0, \dots, N-1 \quad (2.233)$$

$$x_2(i) \leq 0.4 \quad i = 0, \dots, N-1 \quad (2.234)$$

$$|u| \leq 8 \quad i = 0, \dots, N-1 \quad (2.235)$$

$$|\Delta u| \leq 0.5 \quad i = 0, \dots, N-1 \quad (2.236)$$

$$\Delta u = u(k) - u(k-1) \quad i = 0, \dots, N-1 \quad (2.237)$$

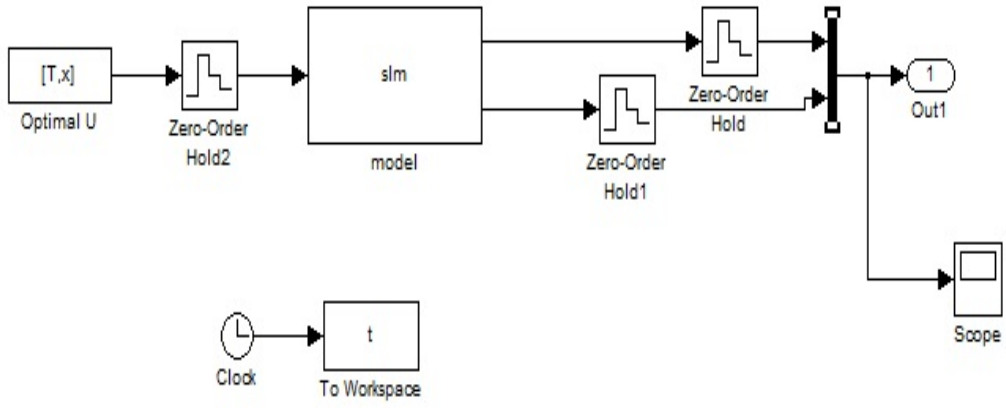
Burada N adım miktarını, $g_N(x_N)$ fonksiyonu, terminal penaltıyı ifade etmektedir. Başlangıç değeri $x_0 = [0,0]$ 'olan ve N adım sonunda istenen referans değeri $x_N = [0.4,0]$ ' olsun. (2.230) ile tanımlı sistem denkleminde $x_1 = 0.4, x_2 = 0$ çalışma noktasına ait sisteme ait giriş değeri $u=7.6404$ olarak bulunur ve $h=0.05$ ve $N=40$ seçilip, (2.232) deki amaç fonksiyonu tekrar düzenlenirse:

$$J = 50(x_1(40) - 0.4) + \frac{1}{2} \sum_{i=0}^{39} (x_1(k) - 0.4)^2 + x_2^2 + (u(k) - 7.6404)^2 \quad (2.238)$$

s.t.

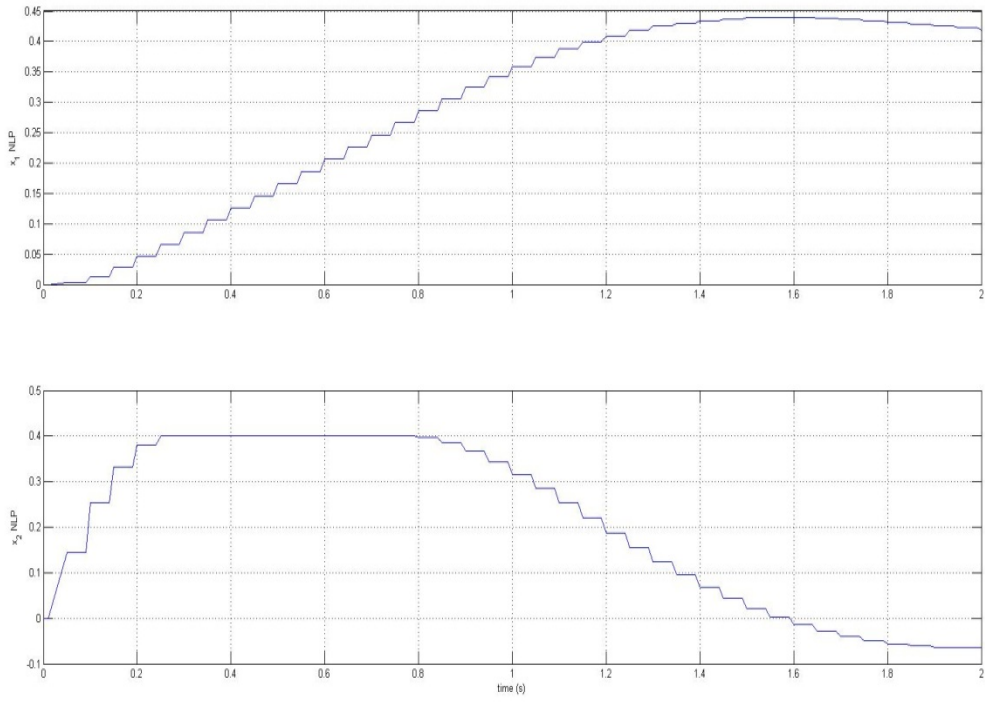
$$\begin{aligned} x_1(k+1) &= x_1(k) + 0.05x_2(k) \\ x_2(k+1) &= x_2(k) + 0.05[-9.81 \sin(x_1(k)) - 3x_2(k) + 0.5u(k)] \end{aligned} \quad (2.239)$$

Olarak seçilir. Sisteme ait blok diyagramı Şekil 2.13'te verilmiştir. Sistem fiziksel olarak sürekli (ing. continuous) sistem olduğundan, benzetimlerde sistemi ayrık-zamana ayırmak için ZOH (zero-order holder) kullanılmıştır.

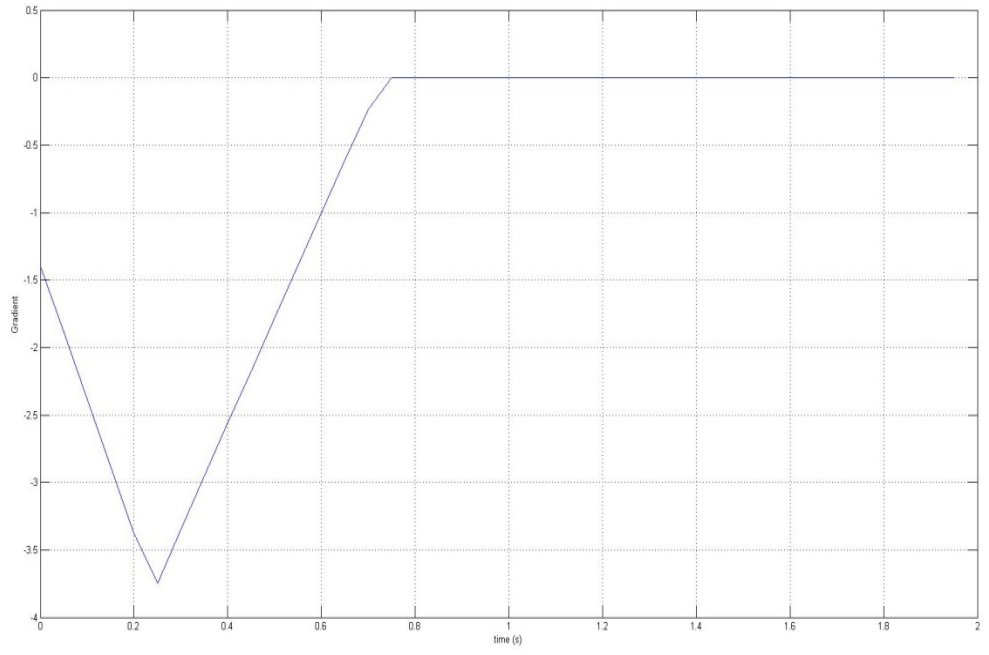


Şekil 2.13 TLM blok diyagramı

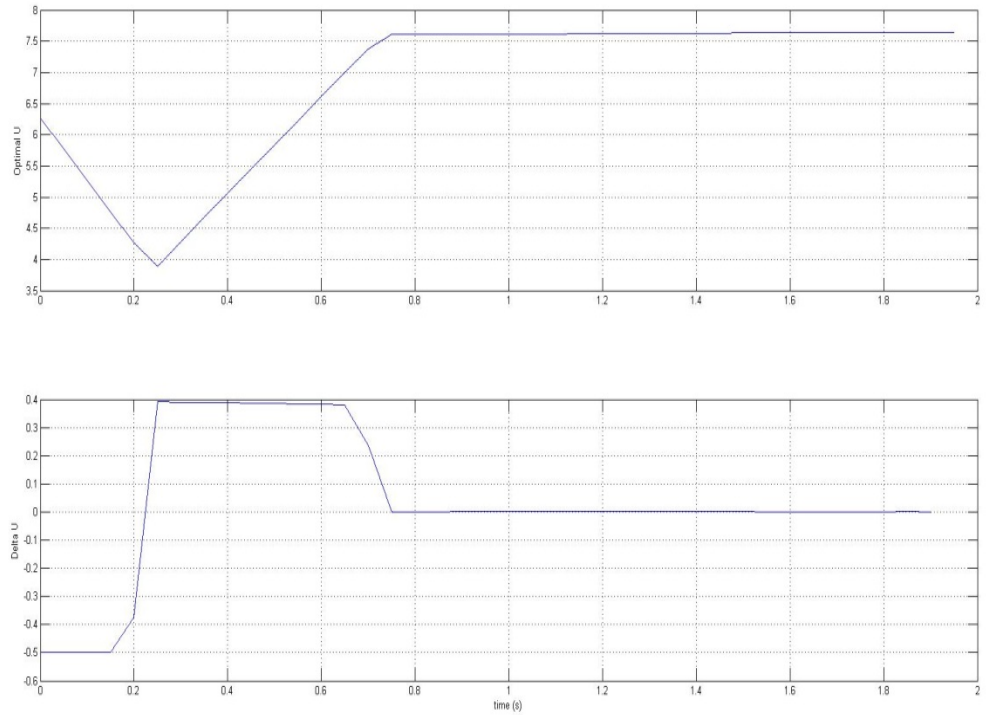
Benzetim Sonuçları



Şekil 2.14 Sistemin durumları



Şekil 2.15 Gradient



Şekil 2.16 Optimal giriş dizini u^* ve du^*

3. LİNEER OPTİMİZASYON

Bu bölümde kuadratik olarak verilen amaç fonksiyonuna ait lineer optimal kontrolcülerden olan Lineer Kuadratik Regulator (ing. Linear Quadratic Regulator, LQR) ve kısıtlı durumlar içinde geçerli olan sonlu ufuk prensibine dayanan Model Öngörülü Kontrol anlatılacaktır.

3.1 Doğrusal Kuadratik Regulator

Modern kontrol kavramları, Kalman'ın 1960'ların başlarında yaptığı çalışmalara dayanır. Kalman ve arkadaşlarının ele aldığı sürece ait sistem ayrık-zamanlı doğrusal durum-uzayı modelidir.

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\y(t) &= Cx(t) + Du(t)\end{aligned}\tag{3.1}$$

Kullanılan durum uzayı formunda olduğu için, en genel haliyle, çok değişkenli sistemleride kapsamaktadır. U sistem girişini, y sistem çıkışını, x sisteme ait durumları göstermektedir. Sistemin orijinde durağan olarak kaldığını varsayalım.

Amaç fonksiyonu J, giriş ve durumların sistemin orijinden sapmalarının karelerini cezalandıran ağırlık matrisleri Q ve R ile tanımlanır:

$$J = \sum_{j=1}^{\infty} \|x(t+j)\|_Q^2 + \|u(t+j)\|_R^2\tag{3.2}$$

Amaç fonksiyonunun norm terimi, aşağıdaki şekilde tanımlanır:

$$\|x\|_Q^2 = x^T Qx\tag{3.3}$$

Yukarıda belirtilen problemin çözümü LQR olarak adlandırılır Algebraic Riccati denkleminin çözümü ile $u=-Kx$ olarak tanımlanan kontrol kanununun, K matris değeri bulunur.

LQR probleminin amaç fonksiyonda (3.2) ilgi çekici bir nokta ise, sayısal integral işleminin sonsuza kadar (sonsuz ufuk ile) hesaplanmasıdır. LQR, kararlılaştırıcı özelliklere sahiptir. Modelleme hatası yokluğunda, Q pozitif yarı tanımlı ve R pozitif tanımlı seçilirse, kararsız sistemleri dahi kararsızlaştırır. En büyük dezavantajı LQR probleminin kısıtları alamaması, doğrusal olmayan davranışlar ve model

belirsizliklerini içermemesidir. LQR, kısıtların amaç fonksiyonuna eklenmesi ve kayan ufuk uygulamasını kullanan MPC nin temellerini oluşturmuştur.

3.2 Model Öngörülü Kontrol

Bu bölümde Model Öngörülü Kontrol (ing. Model Predictive Control, MPC) temel kavramlardan başlanarak, doğrusal sistemler için kısıtlı ve kısıtsız MPC problemleri tanımlanacak, problemin analitik ve yinelemeli çözüm yöntemleri sunulacak, bozucu, belirsizlik ve gürültü durumlarında kapalı çevrim davranışı incelenecek ve kontrolcünün ince ayarlamasına ilişkin önerilerde bulunulacaktır.

Son olarak ise en genel haldeki Nonlinear MPC (NMPC) problemi sunulacaktır. Bölüm, NMPC probleminin MPC bakış açısı ile ele alınmasında uygulanan yaklaşımların tanımı ile sona erecektir.

Model Öngörülü Kontrol, sistemin gelecekteki davranışını kuadratik bir başarı ölçütüne göre optimize etmek için sistemin ayrıntılı bir modelini kullanan yöntemlere verilen genel isimdir. Petrokimya endüstrisi için geliştirilmiş olmasına rağmen, proses endüstrisi, uçak ve askeri uygulamalarda kendisine birçok uygulama alanı bulmuştur. Sanayide en geniş uygulama alanı bulabilmiş gelişmiş kontrol yöntemidir. MPC'un bu başarısı şu şekilde sıralanabilir:

- 1.Çok değişkenli kontrol problemlerini kolaylıkla üstesinden gelme.
- 2.Eyleyicilerdeki kısıtları ele alması.
- 3.Kısıtlara yakın bölgelerde çalışma olanağı vermesi.
- 4.Bu uygulamalarda kontrol sinyalinin giriş sıklığı, göreceli olarak az olduğundan, online hesaplama için gerekli zamana sahip olması.

MPC'un temel elemanları aşağıdaki gibi sıralanabilir:

- 1.Sistemin gelecekteki davranışını öngörmek için, öngörü modeli
- 2.Öngörülen davranışı bir başarı ölçütüne göre optimizasyonu, bedel fonksiyonu
- 3.Optimum sistem girişlerinin her kontrol adımında hesaplanması, kayan ufuk prensibi

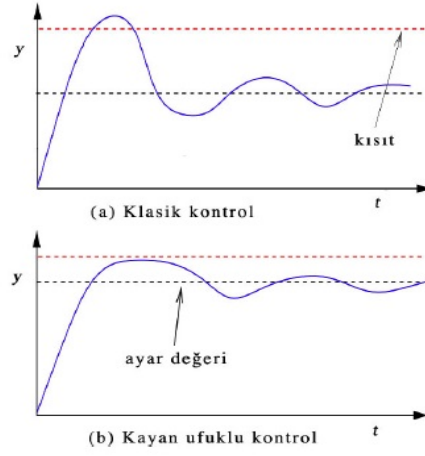
3.2.1 Kayan ufuk prensibi

Kayan ufuk (ing. Receding horizon, RH) prensibi doğrusal kuadratik (ing. Linear quadratic, LQ) kontrol kuramına dayanır. LQ kontrolcü, kuadratik olarak tanımlanmış bir bedel fonksiyonunu sonsuz zaman uzunluğu için minimize ederek optimum kontrol girişi üretir. Ele alınan problem için, başka bir optimal çözüm yoktur. Kayan ufuklu kontrolde, kontrol girişleri sonlu bir ufuk için bulunur, bu ufuktan sonraki optimal kontrol girişleri hesaplanmaz. Bu bağlamda RH kontrol alt-optimaldir denebilir, fakat bedel fonksiyonuna eklenebilecek fonksiyonuna eklenecek ek terimler ile bu aşılabılır.

Analitik ifadelere dayanan çözümler üreten LQR'nin aksine, kayan ufuklu kontrolde sayısal çözümlere gerekir. Bedel fonksiyonu kuadratik programlama ile çözümlenmelidir[4]. Kuadratik optimizasyon yapısı kısıtların formülasyonda yer almasına izin verdiğinden, RH kontrolcü kısıtları ihlal edilmeden önce ele alarak, klasik kontrole göre üstünlük sağlamaktadır. **Şekil 3.1**'de görüldüğü gibi klasik kontrolde kısıtlamalar, sadece ihlal edilmeleri halinde ele alınırlar. RH kontrolcü ise, gelecekteki çıkış değerlerini kısıtlamaları ihlal etmeyecek şekilde hesapladığı için, çıkış limitlerin çok yakınında, fakat sınırları geçmeden çalışabilmektedir. Bu önemli bir durumdur, çünkü kısıtlı problemlerde optimal çözüm çoğu zaman kısıtların kesişim noktalarında bulunur.

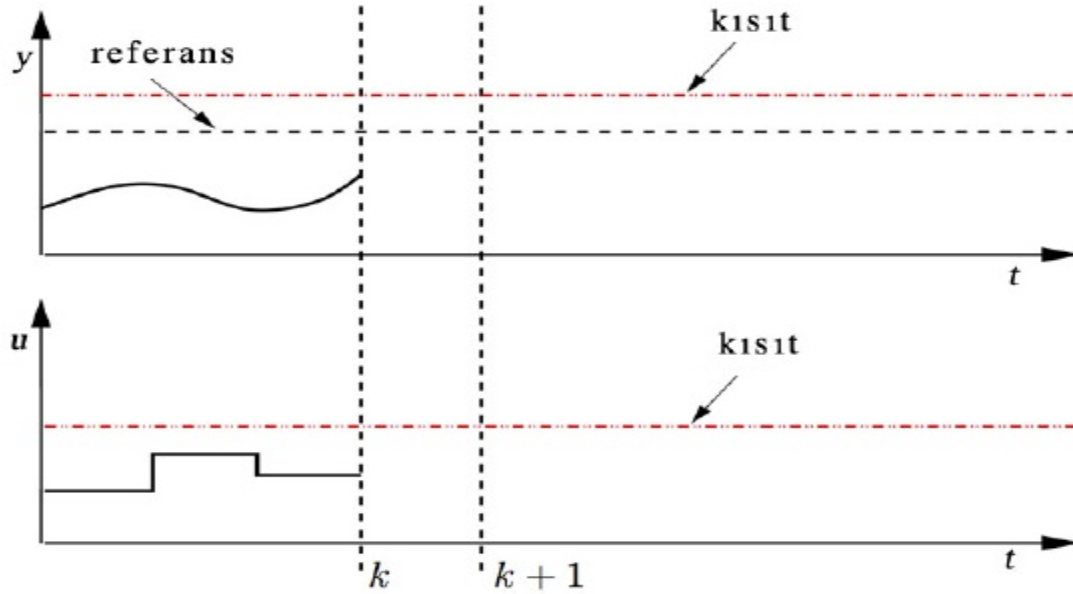
Kayan ufuk prensibi bir örnekleme adımı şu şekildedir:

- 1.Sistemin durumları ölçülür. Ölçülemeyen durumlar gözlemci ile kestirilir.
- 2.Sistemin gelecekteki davranışlarını öngören model kullanılarak, tanımlı bir kuadratik bedel fonksiyonunu kısıtlar dahilinde, mevcut ve gelecek zaman adımları için kontrol giriş dizisi hesaplanır.
- 3.Optimal kontrol giriş dizisinin mevcut zaman adımı için olan, ilk değeri, sisteme gönderilir.



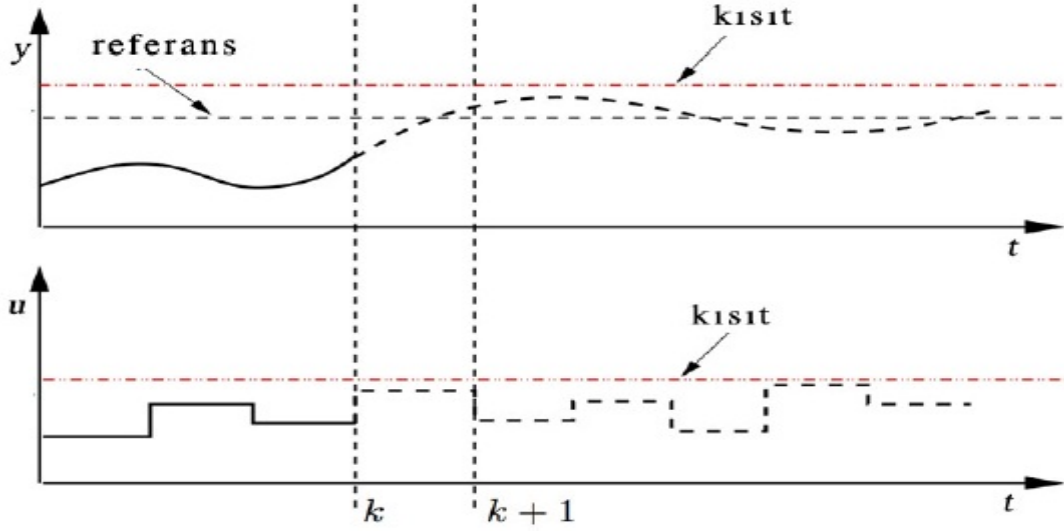
Şekil 3.1 Klasik kontrol ve RH'daki kısıtlar

RH kontrolün davranışını incelemek için, kontrol edilen sistemin bir k zaman adımında, sıfır olmayan giriş ve çıkış değerlerinde olduğunu varsayalım. Bu durum **Şekil 3.2**'de gösterilmiştir.



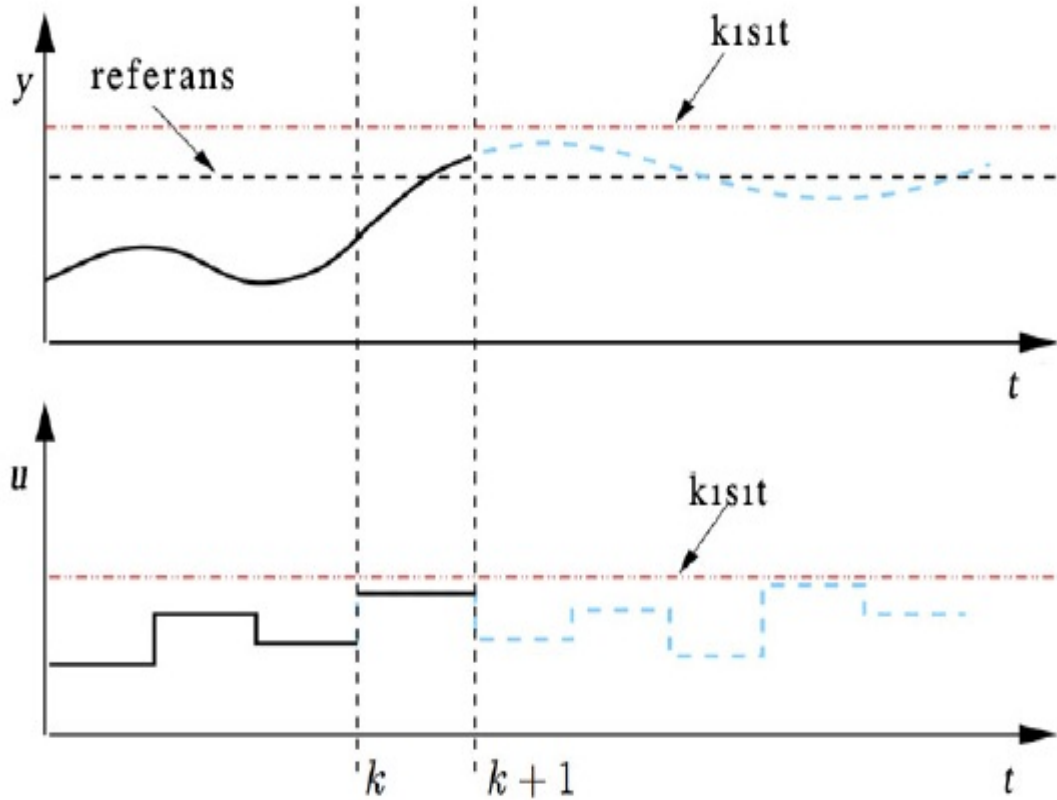
Şekil 3.2 RH başlangıç durumu ($t=k$)

RH prensibinde, ilk adımda sistemin o anki durumları ölçülür ve $x(k)$ durum vektörü elde edilir. İkinci adımda öngörü modeli ve bedel fonksiyonu kullanılarak, belirli sonlu ufuk için bir kontrol girişi dizini elde edilir. Bu dizi, **Şekil 3.3**'de görüldüğü gibi, ufuk bitmeden sistemi istenilen değere getirecek şekilde hesaplanır.



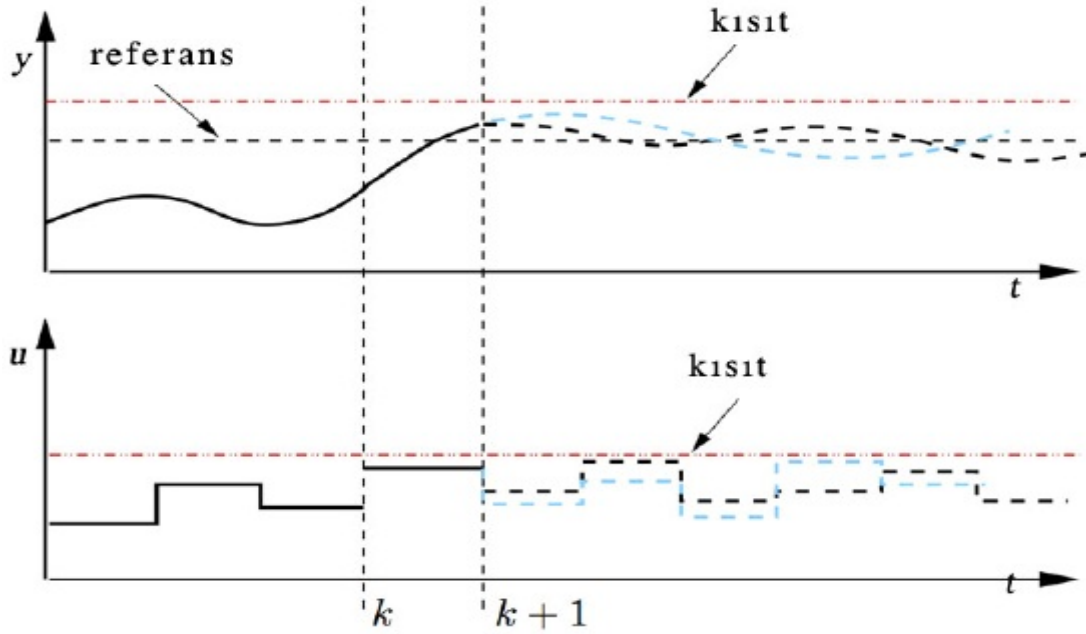
Şekil 3.3 RH'daki optimum kontrol dizini ($t=k$)

Son adımda ise hesaplanan optimal kontrol sinyal dizinin ilk elemanı $u(k)$ sisteme gönderilir ve Şekil 3.4'te gösterildiği gibi sonlu ufuklu optimizasyonunun ilk adımı sona ermiş olup, bir sonraki adıma geçilir. Bu adımda ölçülen durum vektörü $x(k+1)$ 'dir.



Şekil 3.4 RH kontrol girişi ve bir sonraki hesaplama

$(k+1)$ adımına gelindiğinde, k adımına göre kontrolcü, bir adım daha uzağı öngörebilmektedir, ufuk zamanda ileriye doğru kaymıştır. Kayan ufuk prensibi ismi de, ufkun her kontrol adımında bir adım ileriye doğru kaymasından gelmektedir. Gürültü ve bozucu olmayan durumlarda, (k) ve $(k+1)$ adımlarında hesaplanan optimal kontrol giriş dizininin aynı olması beklenir, ama pratikte bu etkiler hemen hemen bulunur. Şekil 3.5'te görüldüğü gibi, iki ayrı adımda hesaplanan optimal kontrol giriş dizini birbirinden farklı olabilir.



Şekil 3.5 $(t=k)$ ve $(t=k+1)$ hesaplanan kontrol dizini

Sisteme sadece hesaplanan optimal kontrol dizinin ilk bileşeni gönderilecekse, bir kontrol ufku boyunca hesaplanma gereği nedendir? Bir adım sonraki anlık bedel fonksiyonu (ing. Instantaneous cost function) hesaplaması gerçekleştirilerek yapılan kontrol stratejisine arsız kontrol (ing. Greedy control) adı verilir.[4] Böyle bir kontrolcü, bozucu etkileri altında kararsızlığa düşebilmektedir. Uzun ufuklu öngörü ile mevcut zaman adımında yapılan bir kontrol hamlesinin gelecekteki etkileri de ele alınmış olmaktadır.

3.2.2 MPC'de kullanılan modeller

MPC'de kullanılacak olan modelin seçimi MPC algoritma tasarımında önemli bir yer teşkil etmektedir. Düşük performansa sahip MP Kontrolcülerde, genellikle sistemin uygun modeli kullanmamasından kaynaklanmaktadır. Bu bölümde MPC'de

kullanılan sistem modelleri, belirsizliklerin modellenmesi, gürültü modeli, ve bunların sistemler ile beraber olma durumları ele alınacaktır.

3.2.2.1 Belirsizliklerin modellenmesi

Bir sistemi kontrol ederken birincil amaç, sistem çıkışının referansı sürekli rejim hatası (ofset, ing. offset) olmadan izlemesidir. Olasıl (ing. stochastic) etkilere ve parametre belirsizliklerine sıklıkla rastlanır. Olasıl etkilerin modele nasıl dahil edilmesi gerekliliği ele alınacaktır.

Klasik kontrol kuramından bilindiği gibi integral etki, ofset oluşmasını engeller. Belirsizliklerin ve bozucuların yokluğunda integral etki olmadan da ofset oluşturmadan kontrol sağlanabilir, fakat pratikte parametre belirsizlikleri ve bozucular sıklıkla karşılaşılan bir durumdur ve integral etki sağlamak gereklidir.

Bozucuları sistematik olarak reddetmenin en iyi yolunun, bozucuların bir modelinin kurulması olduğunu ele alalım [4. s18]. Olasıl bozucu $d(k)$ için, beyaz gürültü entegratörü olarak tanımlanan, aşağıdaki gibi bir (3.4) model seçilsin:

$$d(k+1) = d(k) + v(k) \quad (3.4)$$

Burada $v(k)$ sıfır ortalamaya sahip, değeri bilinmeyen beyaz gürültüyü ifade etmektedir. Beyaz gürültü, düz spektral yoğunluğa sahip özel bir sinyaldir.

3.2.2.2 Gürültünün modellenmesi

Geri beslemeli kontrolcülerde çıkış değeri ölçülürken, yüksek frekanslı gürültü sinyali, çıkış sinyaline etki eder ve ölçülmek istenen değeri bozarlar. Bu olguya ölçüm gürültüsü (ing. measurement noise) adı verilir. Ölçüm gürültüsü genellikle yüksek frekanslıdır ve alçak geçiren (ing. low-pass) filtreler ile sinyalden süzülür. Buna rağmen ölçülen büyüklükte bozucu etki yaratırlar. Bozucular için olduğu gibi, ölçüm gürültüsü etkisinden kurtulmak için sisteme gürültü modeli eklenebilir.

3.2.2.3 Proses modelleri

Bu bölümde sistemi en iyi şekilde tarif etmek için kullanılan modeller üzerinde durulacaktır. Bazı modellerin matematiksel ifadeleri aşağıda kısaca açıklanacak, durum uzay modeli üzerinde detaylı inceleme yapılacaktır.

Doğrusal Modeller:

- Basamak Cevabı Modeli (ing. Finite Impulse Response Model (FIR))
- Darbe Cevabı Modeli (ing. Finite Step Response Model (FSR))

- Transfer Fonksiyonu Modeli
- Durum uzayı Modeli
- Polinom Modelleri (ARX,ARMAX)

Doğrusal Olmayan Modeller:

- Yapay Sinir Ağları
- Bulanık Modeller,...

Darbe Cevabı Modeli

Durağan halde bulunan lineer bir sisteme darbe girişi uygulayalım.

$$u(0) = \{1, 0, \dots, 0, \dots\} \quad (3.5)$$

Ve sistemin çıkışını gözlemleyelim.

$$y(0) = \{h_0, h_1, \dots, h_n, h_{n+1}, \dots\} \quad (3.6)$$



Şekil 3.6 Entegratör içermeyen ve içeren Sistem

Sistemin davranışının şu şekilde olduğu varsayılacaktır.

- $h_0 = 0$ Sistem girişe anında cevap vermemekte
- $h_k = 0 \quad k > n$ Sistem n adım sonunda durağanlaşacaktır.

Bu tür sistemlere aşağıda verilen katsayılarla sahip Darbe Cevabı Modeli (DCM) denmektedir. (ing. Finite Impulse Response Model).

$$H = [h_1, h_2, \dots, h_n] \quad (3.7)$$

Bir çok sistem DCM ile tanımlanabilmektedir. DCM, MPC'e dayalı kontrolcülerde yaygın olarak kullanılmaktadır. Model Algoritmik Kontrol (MAK), özel olarak Genelleştirilmiş Öngörü Kontrol (ing. Generalized Predictive Control) ve Genişletilmiş Kendi Kendine Ayarlayan Öngörülü Kontrol (ing. Extended Prediction Self Adaptive Control) yöntemi bu modeli kullanmaktadır. Giriş çıkış ilişkisi bozucu etkisi olmadan (3.8) ile verilmektedir. Burada h_i 'ler sisteme darbe girişi uygulandığında elde edilen çıkışın örneklenmiş değerleridir. Görüldüğü gibi sistem çıkışına ilişkin N değer göz önüne alınmış, sonsuz bir toplama yapılmamıştır. Bu

nedenle bu model **Şekil 3.6** (a) ' da verilen entegratör içermeyen ve kararlı sistemler için uygundur. $H(z^{-1})$ sistemin ayrık transfer fonksiyonu ve z^{-1} geciktirme operatörüdür.

$$y(t) = \sum_{i=1}^N h_i u(t-i) = H(z^{-1})u(t) \quad (3.8)$$

Sanayide yaygın olarak kullanılan bu modelin en büyük avantajlarından birisi, sistem hakkında detaylı bilgiye gerek olmamasıdır, bu sayede sistem tanımlaması (ing. System Identification) kolay hale gelmiştir. Bozucu etkisi olmadan çok girişli çok çıkışlı bir sistem ise **(3.8)**, giriş çıkış ilişkisi **(3.9)** ile tanımlanmaktadır.

$$y_j(t) = \sum_{k=1}^m \sum_{i=1}^N h_i^{kj} u^k(t-i) \quad (3.9)$$

Bozucu Modeli

Genel olarak DCM 'de kullanılan bozucu modeli sabit çıkış bozucu modelidir [4]. Giriş çıkış ilişkisi **(3.10)** de verilmiştir.

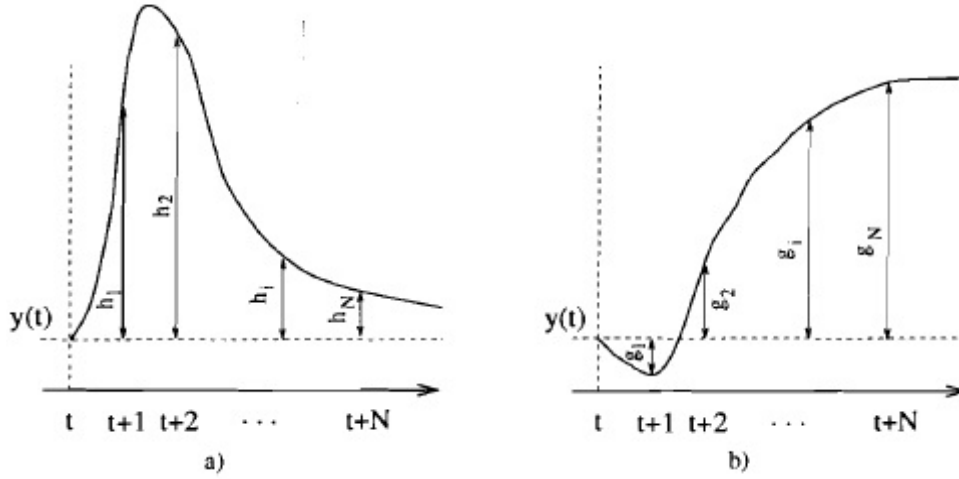
$$y(t) = \sum_{i=1}^N h_i u(t-i) + d_k = H(z^{-1})u(t-i) + d_k \quad (3.10)$$

Basamak Cevabı Modeli

Modele dayalı öngörülü kontrol algoritmalarında, yaygın olarak kullanılan modellerden birisi basamak cevabı modelidir (BCM). Dinamik Matris Kontrol ve varyasyonlarında kullanılmaktadır. Bozucu etkisi olmadan giriş çıkış ilişkisi **(3.11)** ile verilir. Stabil sistemler için, g_i 'ler sisteme basamak giriş uygulandığında elde edilen çıkışın örneklenmiş değerleridir.

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t-i) = y_0 + G(z^{-1})(1-z^{-1})u(t) \quad (3.11)$$

Bu denklemde **(3.11)**, girişin arttırmalarını barındırmaktadır (kontrol girişindeki değişimler), $\Delta u(t) = u(t) - u(t-1)$ olarak tanımlanır.



Şekil 3.7 Darbe ve basamak cevabı

Bozucu Modeli

Yaygın olarak BCM modelinde kullanılan bozucu modeli sabit çıkış bozucu modelidir. (ing constant output disturbance model) . Giriş çıkış ilişkisi (3.12) ile tanımlanmıştır. [4]

$$y(t) = \sum_{i=1}^N g_i \Delta u(t-i) + d_k \quad (3.12)$$

d_k her iki DCM ve BCM bozucu modellerinde, bozucuları temsil etmektedir.

Transfer Fonksiyonu Modeli

Parametre sayısı az olması ve her türlü lineer sisteme uygunluk avantajları sebebiyle yaygın olarak kullanılan modellerden birisidir. Genelleştirilmiş Öngörülü Kontrol, Birleşik Öngörülü Kontrol (ing. Unified Predictive Control), EPSAC , Genişletilmiş Ufuklu Adaptif Kontrol (ing. Extended Horizon Adaptive Control), Çok Adımlı Çok Değişkenli Adaptif Regülatör (ing. Multistep Multivariable Adaptive Regulator), Çok Öngörülü Kayan Ufuklu Adaptif Kontrol (ing. Multipredictor Receding Horizon Adaptive Control) algoritmalarında kullanılır.

Sistem çıkışı $y(t)$ ve sistem girişi $u(t)$ olarak verilen sistem, (3.13) giriş-çıkış ilişkisi ile tanımlanır.

$$A(z^{-1})y(t) = B(z^{-1})u(t) \quad (3.13)$$

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})}u(t)$$

Burada $A(z^{-1})$ ve $B(z^{-1})$ polinomları aşağıdaki gibi tanımlanır.

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na} \\ B(z^{-1}) &= b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb} \end{aligned} \quad (3.14)$$

Fark denklemleri olarak sistem (3.15) ile tanımlıdır.

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{nb} u(t-nb-1) \quad (3.15)$$

Yaygın olarak kullanılan Transfer Fonksiyonu modeli, stabil olmayan sistemlerde de uygun olsa bile, sistem hakkında ilk bilgilerin bilinmesi gerekmektedir. (A ve B polinomlarının mertebeleri)

Bozucu Modeli

Transfer Fonksiyon modelinde, var olan bozucu ve ölçmeden kaynaklı gürültüleri içeren giriş çıkış ilişkisi aşağıdaki gibi tanımlıdır.

Burada v_k sistemde bulunabilecek dış bozucu etkileri ve ölçmeden kaynaklı gürültüleri en iyi şekilde tanımlayabilecek sıfır ortalamaya sahip bilinmeyen gelişigüzel bir değişendir (ing. unknown zero mean random variable). $T(z^{-1}) = 1$ olması durumunda (3.15) denklemi farklar denklemi olarak yazılırsa:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{nb} u(t-nb-1) + d_k \quad (3.16)$$

Burada d_k , $\frac{T(z^{-1})}{\Delta(z^{-1})} v_k$ dan elde edilen bilinmeyen bozucu etkidir.

$T(z) = 1$ durumunda bozucu modeli:

$$d_k = \frac{1}{\Delta(z^{-1})} v_k \equiv d_{k+1} = d_k + v_k \quad (3.17)$$

$d = \frac{v}{\Delta}$ entegratör içeren beyaz gürültüyü temsil etmektedir. Bu TFM 'de yaygın

olarak kullanılan bozucu modelidir. Bu durumda önemli olan $T(z^{-1})$ uygun olarak seçilmesidir.

(3.13) 'daki giriş çıkış ilişkisini gösteren sistem modelinin , artımsal versiyonu şu şekilde tanımlıdır.

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})\Delta(z^{-1})} \Delta u(t) + T(z^{-1})v(t) \quad (3.18)$$

Artımsal modeli kullanarak, değeri sıfırdan farklı bilinmeyen bozucu modeli olan d elimine edilmiş ve sadece sıfır ortalamaya sahip v bozucu modeli ile tanımlanmıştır.

v 'nin gelecekteki deęerleri sıfır olarak kabul edilebilir, buda öngörü modeline etki etmemektedir.

$T(z^{-1})$ 'nin seçimi

$T(z^{-1})$ bozucu etkileri modellemek için kullanılan polinomdur. Seçimi ile ilgili aşağıda birkaç rehber belirtilmiştir:

1. $T = \hat{a}\hat{T}$, burada \hat{a} sistemin baskın kutup veya kutuplarını barındırır, \hat{T} bire yakın olan kutupları barındırır. Örnek olarak $T = [1 - 0.8z^{-1}]^n$
2. $1/T$ 'yi düşük frekansları geçiren bir filtre olarak seçilebilir (ing. low-pass filter). Buda yüksek frekanstaki gürültülerin etkisini azaltacaktır.
3. $T=1$ seçilebilir.
4. $T=a$, FIR modeline benzer bir yapıya sahip olmaktadır.
5. T nonlineer veya daha karmaşık bir yapıda seçilebilir.

Durum-uzayı modeli

Durum uzayı modeli, çok deęişkenli sistemlerin tanımlanmasında kolaylık sağladığından dolayı Öngörüs el Fonksiyonel Kontrol (ing. Predictive Function Control) gibi bazı Modele Dayalı Öngörülü Kontrol algoritmalarında kullanılmaktadır.

Sistem modeli (3.19) ile ifade edilmektedir.

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\y(t) &= Cx(t) + Du(t)\end{aligned}\tag{3.19}$$

Bu denklemde x durum deęişken vektörünü, A, B, C ve D sırasıyla sistem matrisi, giriş matrisi ve çıkış matrisi olarak ifade edilmektedir. Bu çalışmada $D=0$ durum-uzay modeli ele alınacaktır.

Artımsal model

MPC'de kullanılan sistem modelleri artımsal modellerdir. (3.19)'da tanımlanan durum-uzay modeli, aşağıda (3.20) artımsal model olarak tanımlanabilir.

$$\begin{aligned}x(k+1) &= \tilde{A}x(k) + \tilde{B}\Delta u(k) \\y(k) &= \tilde{C}x(k)\end{aligned}$$

$$x = \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}, \tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}, C = [C, 0] \quad (3.20)$$

$$u_k = u_{k-1} + \Delta u_k$$

Bozucu modelleri

Çıkış bozucu modeli

Belirsizliklerin modellenmesine doğrudan bir yaklaşım, bir bozucunun çıkış üzerine etkidiğini varsaymaktır. Çıkış bozucu modeli d_y ile ifade edilsin. (3.19)'a eklenen

(3.4) çıkış bozucu ile, sistem modeli aşağıdaki şekli alacaktır:

$$\begin{aligned} d_y(k+1) &= d_y(k) + v(k) \\ x_p(k+1) &= A_p x_p(k) + B_p u_p(k) \\ y(k) &= C_p x_p(k) + D_p u_p(k) + d_y(k) \end{aligned} \quad (3.21)$$

Bu şekilde tanımlanan bozucu, literatürde [4], genelde bir yeni durum olarak durum vektörüne dahil edilir.

$$x = \begin{bmatrix} x_p(k) \\ d_y(k) \end{bmatrix}, u(k) = u_p(k), A = \begin{bmatrix} A_p & 0 \\ 0 & I \end{bmatrix}, B = \begin{bmatrix} B_p \\ 0 \end{bmatrix}, C = [C_p, I], D = D_p$$

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned} \quad (3.22)$$

Aynı bozucu, bir giriş olarak da ele alınabilir. Benzetimlerde kullanılacak yazılım, bozucuların giriş olarak tanımlanmasını gerektirmektedir. İki ifadenin birbirine özdeş olduğu kolaylıkla gösterilebilir. Bozucu sistem giriş olarak tanımlandığında aşağıdaki durum-uzay modeli elde edilir.

$$x = x_p(k), u(k) = \begin{bmatrix} u_p(k) \\ d_y(k) \end{bmatrix}, A = A_p, B = [B_p, 0], C = C_p, D = [D_p, I]$$

$$\begin{aligned} d_y(k+1) &= d_y(k) + v(k) \\ x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (3.23)$$

Giriş bozucu modeli

Dinamik denklemi (3.4)' de verilen bozucunun, durumlar üzerinde etkili olduğu hal durumunda giriş bozucu modeli denir. Bozucu etkisi altındaki sistem modeli aşağıdaki gibidir:

$$\begin{aligned}
d_x(k+1) &= d_x(k) + v(k) \\
x_p(k+1) &= A_p x_p(k) + B_p u_p(k) + d_x(k) \\
y(k) &= C_p x_p(k) + D_p u_p(k)
\end{aligned} \tag{3.24}$$

Giriş bozucusu bir durum olarak modele eklenirse:

$$\begin{aligned}
x &= \begin{bmatrix} x_p(k) \\ d_x(k) \end{bmatrix}, u(k) = u_p(k), A = \begin{bmatrix} A_p & I \\ 0 & I \end{bmatrix}, B = \begin{bmatrix} B_p \\ 0 \end{bmatrix}, C = [C_p, 0], D = D_p \\
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k) + v(k)
\end{aligned} \tag{3.25}$$

Bozucu bir giriş olarak modele eklenirse:

$$\begin{aligned}
x &= x_p(k), u(k) = \begin{bmatrix} u_p(k) \\ d_x(k) \end{bmatrix}, A = A_p, B = [B_p, I], C = C_p, D = [D_p, 0] \\
d_x(k+1) &= d_x(k) + v(k) \\
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k)
\end{aligned} \tag{3.26}$$

Durum ve çıkış bozucu modeli

Belirsizliklerin her iki durumda hem duruma hem de çıkışa etki etmesi durumunda durum-uzay modeli aşağıdaki gibi elde edilir:

$$\begin{aligned}
x &= \begin{bmatrix} x_p(k) \\ d_x(k) \\ d_y(k) \end{bmatrix}, u(k) = u_p(k), A = \begin{bmatrix} A_p & I & 0 \\ 0 & I & 0 \\ 0 & I & I \end{bmatrix}, B = \begin{bmatrix} B_p \\ 0 \\ 0 \end{bmatrix}, C = [C_p, 0, I], D = D_p \\
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k) + v(k)
\end{aligned} \tag{3.27}$$

Bozucular, giriş olarak modele eklenirse;

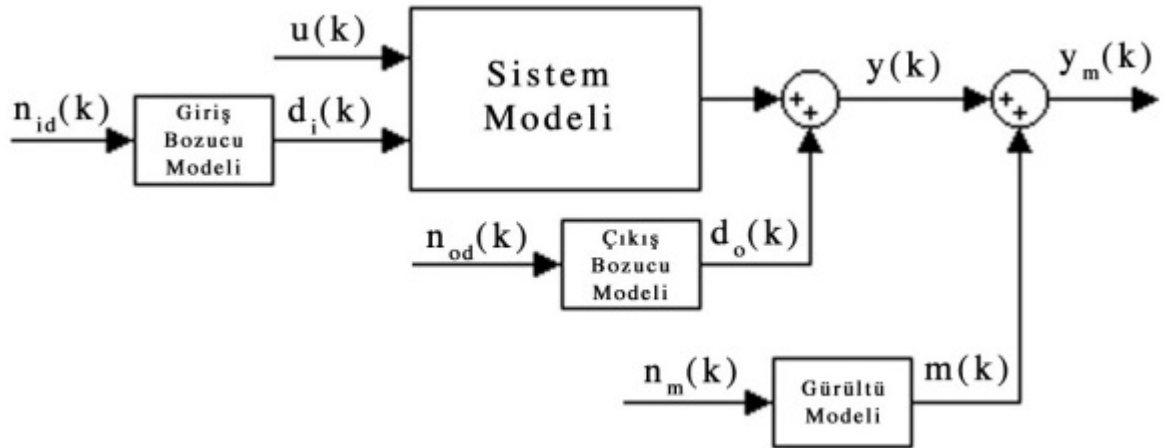
$$\begin{aligned}
x &= x_p(k), u(k) = \begin{bmatrix} u_p(k) \\ d_x(k) \\ d_y(k) \end{bmatrix}, d(k) = \begin{bmatrix} d_x(k) \\ d_y(k) \end{bmatrix} \\
A &= A_p, B = [B_p, I, 0], C = C_p, D = [D_p, 0, I] \\
d_x(k+1) &= d_x(k) + v(k) \\
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k)
\end{aligned} \tag{3.28}$$

Gürültü modeli

Yüksek frekanslı gürültü sinyali, çıkış değeri üzerine etki eder. Çıkış vektörü $y(k)$ 'nin ölçüm gürültüsü $m(k)$ tarafından etkilendiğini varsayalım. Ölçüm gürültüsü $m(k)$, doğrusal zamanla-değişmeyen bir sistemin çıkışı olarak modellenebilir. Aşağıda belirtilen gürültü modeli, sıfır ortalamalı ve birim varyanslı beyaz gürültü tarafından tahrik edilir. Denklemlerde x_m gürültüye ait durum vektörünü, m gürültü vektörünü, A,B,C,D ise gürültü modelini ifade eder.

$$\begin{aligned}x_m(k+1) &= Ax_m(k) + Bu_m(k) \\m(k) &= Cx_m(k) + Du_m(k)\end{aligned}\tag{3.29}$$

MPC algoritmalarına uygun sistemi en iyi şekilde ifade etmek için kullanılan modeller tanıttı. Şekil 3.8'de sistem ve sistemi etkileyecek bozucular ve ölçme gürültüsü şematik olarak verilmiştir. Bir sonraki bölümde, MPC temel elemanlarından olan öngörü modeli, tanıtilan sistemler için irdelenecektir.



Şekil 3.8 Bozucu ve gürültü modelleri

3.2.2.4 Öngörü modelleri

MÖK'ün temel taşıdır. Öngörülen çıkışları hesaplamak için kullanılır. Bu yüzden etkisiz bir dizayna sahip, süreç dinamiklerini tam yakalayabilen en iyi olası modelin kullanılması zorunludur. Sistemin doğrusal olup olmamasına, kısıtlara, bozucu etkiye göre farklı tipte öngörü modelleri mevcuttur. Bu bölümde, tanıtilan sistem modellerine ait öngörü modelleri irdelenecektir.

Darbe cevabı öngörü modeli

Nominal halde olan denklem (3.8)'de tanımlı darbe cevabı modeline ait öngörü modeli aşağıdaki gibi tanımlıdır:

$$\hat{y}(t+k|t) = \sum_{i=1}^N h_i u(t+k-i|t) = H(z^{-1})u(t+k|t) \quad (3.30)$$

Model çıkış değeri \hat{y}_k ve ölçülen değer y_k arasındaki ofset şu şekilde tanımlı olsun:

$$\hat{d}_k = y(k) - \hat{y}(k) \quad (3.31)$$

Bu ofset, bilinmeyen bozucunun en iyi kestirimi olduğu varsayımıyla, denklem (3.10)'de tanımlı bozucu etkiye sahip darbe cevabı modeline ait öngörü modeli aşağıdaki şekilde tanımlanır:

$$\begin{aligned} \hat{y}(t+k|t) &= \sum_{i=1}^N h_i u(t+k-i|t) + \hat{d}_k \\ \hat{y}(t+k|t) &= y(k) + \sum_{i=1}^N h_i u(t+k-i|t) - \sum_{i=1}^N h_i u(k-i|t) \end{aligned} \quad (3.32)$$

Basamak cevabı öngörü modeli

Nominal halde olan denklem (3.11)'de tanımlı basamak cevabı modeline ait öngörü modeli aşağıdaki gibi tanımlıdır:

$$\hat{y}(t+k|t) = \sum_{j=1}^N g_j \Delta u(t+k-j|t) + d_k \quad (3.33)$$

Model çıkış değeri \hat{y}_k ve ölçülen değer y_k arasındaki ofset şu şekilde tanımlı olsun:

$$\hat{d}_k = y(k) - \hat{y}(k) = y(k) - \sum_{j=1}^N g_j \Delta u(k-j|t) \quad (3.34)$$

Bu ofset, bilinmeyen bozucunun en iyi kestirimi olduğu varsayımıyla, denklem (3.12)'de tanımlı bozucu etkiye sahip darbe cevabı modeline ait öngörü modeli aşağıdaki şekilde tanımlanır:

$$\begin{aligned} \hat{y}(t+k|t) &= \sum_{j=1}^N h_j u(t+k-j|t) + \hat{d}_k \\ \hat{y}(t+k|t) &= y(k) + \sum_{j=1}^i h_{i-j} \Delta u(k+j|t) + \sum_{j=1}^N [h_{j+i} - h_j] \Delta u(k-j|t) \end{aligned} \quad (3.35)$$

Basamak cevabı modelinde öngörü modeli

$$\hat{y}_{\rightarrow k} = H \underline{\Delta u}_{k-1} + L \hat{y}_k + M \underline{\Delta u}_{k-1} \quad (3.36)$$

Burada H, M ve L şu şekilde tanımlıdır:

$$H = \frac{C_G}{\Delta}, M = \begin{bmatrix} H_2 - H_1 & H_{n+1} - H_n \\ H_3 - H_1 & H_{n+2} - H_n \\ \cdot & \dots \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}, L = \begin{bmatrix} I \\ I \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (3.37)$$

Transfer fonksiyonu öngörü modeli

Transfer fonksiyonu modele ait öngörü modeli oluşturmak için birden fazla yöntem bulunmaktadır.[3]. Birçok makalede Diophantine Denklemi kullanılarak öngörü modeli oluşturulmaktadır.. Bu bölümde ise, $T(z^{-1}) = 1$ ve $T(z^{-1}) \neq 1$ durumlarındaki öngörü modelleri irdelenecektir.

(3.13) denkleminde tanımlı bozucu etkisi olmadan nominal transfer fonksiyon modeli ve modele ait öngörü modeli aşağıdaki gibi tanımlıdır:

$$A(z^{-1})\Delta(z^{-1})y(t) = B(z^{-1})\Delta u(t) \quad (3.38)$$

Transfer fonksiyonu modelini A ve B polinomlarını genişleterek farklar denklemi olarak yazılırsa:

$$\begin{aligned} A(z^{-1})\Delta(z^{-1}) &= 1 + A_1z^{-1} + A_2z^{-2} + \dots + A_{n+1}z^{-n-1} \\ B(z^{-1}) &= b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n} \end{aligned} \quad (3.39)$$

Bir adım sonraki öngörü modeli:

$$y(k+1) = -A_1y(k) - \dots - A_{n+1}y(k-n) + b_1\Delta(k) + \dots + b_{n-1}\Delta u(k-n+1) \quad (3.40)$$

(3.40) öngörü modeli, geçmiş giriş, çıkış ve gelecekteki girişler biçimi ile yeniden düzenlenirse:

$$y(k+1) = -[A_1, \dots, A_{n+1}]y_{\leftarrow k} + [b_2, \dots, b_n]u_{\leftarrow k-1} + b_1\Delta(k) \quad (3.41)$$

(3.41) denklemi, $y(k+1)$ için verilen geçmiş ve o anki ölçülmüş çıkış ve giriş artımı ile bir adım sonraki öngörü modelidir.

N_y adım boyunca öngörü modeli:

(3.41) denklemi N adım boyunca yazılırsa:

$$\begin{aligned} y(k+1) + A_1y(k) + \dots + A_{n+1}y(k-n) &= b_1\Delta(k) + \dots + b_{n-1}\Delta u(k-n+1) \\ y(k+2) + A_1y(k+1) + \dots + A_{n+1}y(k-n+1) &= b_1\Delta(k+1) + \dots + b_{n-1}\Delta u(k-n+2) \\ \dots & \\ y(k+n_y) + \dots + A_{n+1}y(k+n_y+1-n) &= b_1\Delta(k+n_y-1) + \dots + b_{n-1}\Delta u(k+n_y-n) \end{aligned} \quad (3.41)$$

Vektör – Matris formatında yazılırsa:

$$\begin{aligned}
 & \begin{bmatrix} 1 & 0 & \dots & 0 \\ A_1 & 1 & \dots & 0 \\ A_2 & A_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ \vdots \\ y(k+n_y) \end{bmatrix} + \begin{bmatrix} A_1 & A_2 & \dots & A_{n+1} \\ A_2 & A_3 & \dots & 0 \\ A_3 & A_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ \vdots \\ y(k-n) \end{bmatrix} \\
 & = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ b_2 & b_1 & \dots & 0 \\ b_3 & b_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \vdots \\ \Delta u(k+n_y-1) \end{bmatrix} + \begin{bmatrix} b_2 & b_3 & \dots & b_n \\ b_3 & b_4 & \dots & 0 \\ b_4 & b_5 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \vdots \\ \Delta u(k-n+1) \end{bmatrix} \quad (3.42)
 \end{aligned}$$

(3.42) denklemini sadeleştirilip yazılırsa

$$C_A \underline{y}_k + H_A \underline{y}_{-k} = C_{zb} \underline{\Delta u}_{k-1} + H_{zb} \underline{\Delta u}_{k-1} \quad (3.43)$$

Öngörü denklemi

$$\underline{y}_{-k} = C_A^{-1} [C_{zb} \underline{\Delta u}_{k-1} + H_{zb} \underline{\Delta u}_{k-1} - H_A \underline{y}_{-k}] \quad (3.44)$$

$T(z^{-1}) = 1$ TFM öngörü modeli genel formu aşağıdaki gibi elde edilir:

$$\begin{aligned}
 \underline{y}_{-k} &= H \underline{\Delta u}_{k-1} + P \underline{\Delta u}_{k-1} + Q \underline{y}_{-k} \\
 H &= C_A^{-1} C_{zb}, P = C_A^{-1} H_{zb}, Q = -C_A^{-1} H_A \quad (3.45)
 \end{aligned}$$

$T(z^{-1}) \neq 1$ olması durumunda öngörü modeli;

T filtresi içeren transfer fonksiyonu modelini tekrar yazarsak:

$$A y_k = b \Delta u_k + T v \quad (3.46)$$

Sağ ve sol iki tarafı filtrelemeden geçirirsek;

$$\frac{A}{T} y_k = \frac{b}{T} \Delta u_k + v \quad (3.47)$$

Bunu tekrar düzenlersek;

$$A \frac{y_k}{T} = b \frac{\Delta u_k}{T} + v \Rightarrow A \tilde{y}_k = b \Delta \tilde{u}_k + v \quad (3.48)$$

Bu yöntemle giriş ve çıkış öngörü yapılmadan önce filtrelendir. Tipik olarak $1/T$ alt geçiren filtre bu sayede yüksek frekans gürültü etkisi ortadan kaldırılmış olur.

(3.44) öngörü modelinden tek fark, denkleme filtrelenmiş giriş ve çıkışlar girmektedir.

$$\underline{\tilde{y}}_{-k} = H \underline{\Delta \tilde{u}}_{k-1} + P \underline{\Delta \tilde{u}}_{k-1} + Q \underline{\tilde{y}}_{-k} \quad (3.49)$$

(3.49) denkleminde var olan girişin gelecekteki denklemlerinin filtrelenmesi işimize yaramayacağından yeniden düzenlenmelidir.

$$C_T \tilde{y} + H_T \tilde{y} = \underline{y}; C_T \Delta \tilde{u} + H_T \Delta \tilde{u} = \Delta \underline{u} \quad (3.50)$$

(3.49) denklemini (3.50) 'da yazılırsa;

$$\underline{C}_T^{-1} [y_k - H_T \tilde{y}] = H C_T^{-1} [\Delta u + H_T \Delta \tilde{u}] + P \Delta \tilde{u} + Q \tilde{y} \quad (3.51)$$

Her iki tarafı C_T ile çarpılır bu tekrar düzenlenirse;

$$\begin{aligned} y_k &= H \Delta u_{k-1} + \tilde{P} \Delta \tilde{u}_{k-1} + \tilde{Q} \tilde{y}_k \\ \tilde{P} &= [C_T P - H H_T], \tilde{Q} = [H_T + C_T Q] \end{aligned} \quad (3.52)$$

Bu denklemin $T(z^{-1}) = 1$ olmasından farkları:

- 1-Geçmiş datalar filtrelenmiştir.
- 2-Matrisler değişmiştir. $P \Rightarrow \tilde{P}, Q \Rightarrow \tilde{Q}$
- 3-Eğer P ve Q biliniyorsa, \tilde{P} ve \tilde{Q} kolayca hesaplanabilir.
- 4-Eğer $T=1$ ise, $H_T = 0, C_T = I, \Rightarrow P = \tilde{P}, Q = \tilde{Q}$

Durum-uzayı öngörü modeli

Bu bölümde durum-uzayı modeli ile verilmiş sistemin, nominal hal ve bozucu etkileri altındaki öngörü modelleri incelenecektir.

Nominal durum-uzayı öngörü modeli

Kontrol edilmek istenen sistemin modelinin tam olarak bilindiği ve dışarıdan ölçülmeyen başka bir etkinin bulunmadığı duruma nominal hal adı verilir. Bu durumda ele alınan problem belirgin bir problemdir ve çözümü sayısal yollarla kesin olarak yapılabilir. Nominal durum-uzay modeli aşağıda verilmiştir:

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ y(k) &= Cx(k) \end{aligned} \quad (3.53)$$

Bir sonraki adımda durum ve çıkışın alacağı değerler aşağıdaki gibi olur;

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ y(k+1) &= Cx(k+1) \end{aligned} \quad (3.54)$$

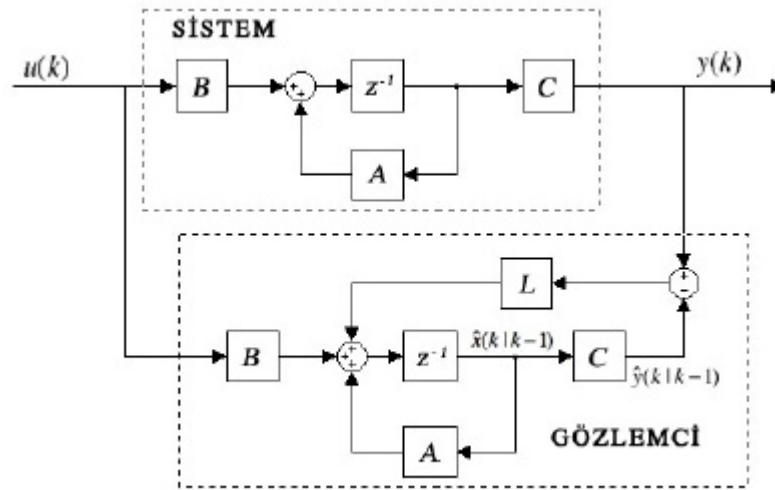
Bir adım daha ilerlenirse;

Bu kısımda, mevcut durum ve gelecekteki girişler bilindiği sürece, bozucu etkilerin olmadığı nominal haldeki durum-uzay formunda modellenmiş sistemin öngörü denklemleri gösterildi. Mevcut durumun bilinmesi için iki yol vardır: i) durumların hepsinin ölçüldüğü geri beslemesi, ii) ölçülmeyen durumların ölçülen çıkışlardan hesaplandığı durum gözlemcisi kullanımı.

Durum gözlemcisi

Durum vektörünün tamamı ölçülemiyorsa, gözlemlenebilirlik varsayımıyla, bir gözlemci ile kestirilebilir. Durum gözlemi problemi, ölçülen çıkış ve bilinen giriş değerlerinden, ölçülmeyen durum değerini hesaplamak olarak tanımlanır ve **Şekil 3.9**'de gösterilmiştir. **(3.53)** denklemi ile tanımlanmış doğrusal bir dinamik sistem için gözlemci denklemleri aşağıdaki gibidir;

$$\begin{aligned}\hat{x}(k+1|k) &= (A-LC)\hat{x}(k|k-1) + Bu(k) + Ly(k) \\ e(k+1) &= x(k) - \hat{x}(k|k-1) = (A-LC)e(k)\end{aligned}\tag{3.60}$$



Şekil 3.9 Durum Gözlemcisi

(3.60) daki eşitlikte, L gözlemci kazancı olarak adlandırılır. $\hat{x}(k+1|k)$; k. adıma kadar hesaplanan (k+1). adım tahmini, e gözlemcideki kestirim hatasını ifade etmektedir. (A-LC) matrisinin özdeğerleri z-düzleminde birim çemberin içindeyse, gözlemci kararlı olacaktır. Gözlemcinin kararlı olması durumunda, durum kestirim hatasının sıfıra yakınsayacağı belirtir. Gözlemci tasarımı, kutup yerleştirme ile yapılır. Kontrol edilmek istenen sistemin durum ve çıkış denklemlerine, bilinen kovaryans matrislerine sahip, beyaz gürültü bozucuları eklendiği takdirde, L kazancı

durum kestirim hatasının karelerinin toplamını (ing. mean square state) minimize edilecek şekilde seçilebilir. Gözlemci bu durumda Kalman Filtresi adını alır.

Sürekli rejim Kalman filtresi

Modelleme belirsizlikleri veya ölçüm gürültüsünün varlığı nedeniyle tüm sistemler belirgin modellerle ifade edilemezler. Durumu kestirimi yapılmak istenen sistemde olası etkiler var ise, proses (3.61)'de verilen zamanla değişmeyen, istasyonel, doğrusal, belirgin-olasıl durum uzay modeli (ing. determinist-stochastic model) ile ayrık zamanda ifade edilebilir.

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + Gw(k) \\y(k) &= Cx(k) + v(k)\end{aligned}\tag{3.61}$$

Burada $w(k)$ proses gürültüsü, nonlinear veya yüksek frekans etkilerin ihmalinden doğan modelleme hatalarından kaynaklanır. Diğer rastlantı değişkeni $v(k)$ ise ölçüm gürültüsünü ifade eder. Bu iki rastlantı değişkeninin de istasyonel, sıfır ortalamalı beyaz gürültüler olduğu varsayılır. G matrisi, proses gürültüsünün durumlar etkisini ifade eden matrisdir. Proses ve ölçüm gürültülerinin kovaryansları sırasıyla Q ve R olarak (3.58) te gibi tanımlansın

$$\begin{aligned}E\{w(k)w^T(k)\} &= Q \\E\{v(k)v^T(k)\} &= R\end{aligned}\tag{3.62}$$

(3.61) denklemindeki belirgin-olasıl sisteme göre kontrol sistemi tasarlanırken, durum geri beslemesine güvenilmez, çünkü ölçülen durum vektörü $x(k)$, proses gürültüsü $w(k)$ 'nin etkisi altındadır. Gerçek durum vektörü $x(k)$ 'nin bilinebilmesi için, bir durum gözlemcisine ihtiyaç vardır. Durum gözlemcisi kısmında belirtilen Luenberger gözlemcisi ile istenen kutuplara sahip gözlemci tasarlanabilir. Fakat bu şekilde tasarlanan bir gözlemci, proses ve ölçüm gürültülerinin güç spektrumlarını göz önüne almaz. Luenberger gözlemcisi, sürekli rejimde kestirim hatasını sıfıra götürecek şekilde, determinist bir şekilde tasarlanır. Olasıl sistemlerde rastlantı değişkenlerinin değerleri tam olarak bilinmediğinden, hatanın sıfırlanması iyi tanımlanmış bir amaç olarak görülmez. Ölçülen $y(k)$ ve sistemin durum vektörü $x(t)$ 'nin rastlantı vektörleri olduğunu hesaba katmak için, problemin istatistiksek bir şekilde tanımlanması gerekir [5]. Durum değişkenlerinin kestirimlerinin ortalamasının, gerçek durum değerlerinin ortalamasına eşit olması istenir. Kestirimin beklenen değerinin durumun beklenen değerine eşit olması gerekmektedir. Hatanın kovaryansını en küçük yapan kestirimci tasarlanmalıdır.

Kalman filtresi, kestirim hatasının kovaryansını minimize edecek şekilde tasarlanan bir gözlemcidir. Sistem ve gürültülerin durum vektörlerini, kontrol girişi ve sistem çıkışı vektöründen kestirir. Tüm gürültüler beyaz gürültüsü ise, filtre kestirilen parametrelerin hatasını minimize eder.

İstatiksel sürekli rejimde, hata kovaryansı $P(k)$, bir sürekli rejim değerine yakınsar, buda P matrisi ile gösterilir, algebraik Riccati denkleminin (3.63) bir çözümüdür.

$$P = A[P - PC^T(CPC^T + R)^{-1}CP]A^T + GQG^T \quad (3.63)$$

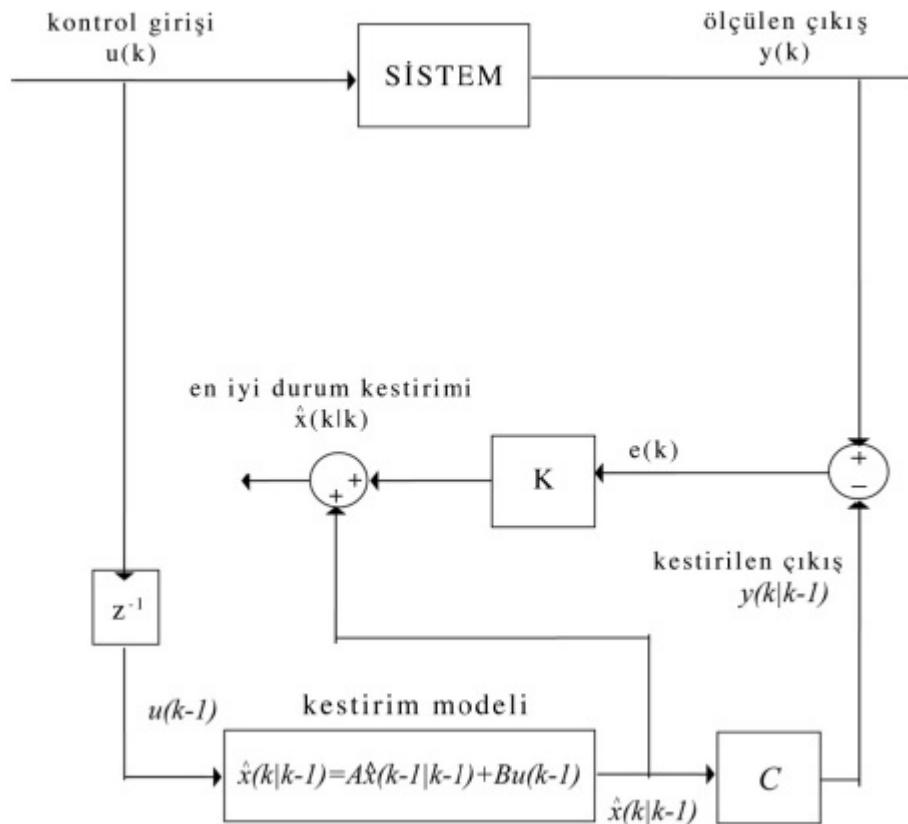
P sürekli rejimde, sabit bir matris olduğundan, gözlemci kazancı K 'da (3.64) bir sabit matris olacaktır, ve sürekli rejim Kalman kazancı adını alır.

$$K = PC^T(CPC^T + R)^{-1} \quad (3.64)$$

(3.61) de verilen sistem için sürekli rejim Kalman Filtresi, (3.65) aşağıda tanımlanmıştır.

$$\begin{aligned} \hat{x}(k|k-1) &= A\hat{x}(k-1|k-1) + Bu(k) \\ x(k|k) &= \hat{x}(k|k-1) + K[y(k) - C\hat{x}(k|k-1)] \end{aligned} \quad (3.65)$$

Sürekli rejim Kalman Filtresi'nin blok diyagramı Şekil 3.10'de verilmiştir.



Şekil 3.10 Sürekli rejim Kalman filtresi

Durum-uzayı öngörü modeli

(3.27) denklemi ile tanımlanan, çıkış, giriş bozucuları ve ölçüm gürültüsünü içeren durum-uzay modeline ait öngörü modeli aşağıda verilmiştir.

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + B_D d(k) \\y(k) &= Cx(k) + D_D d(k)\end{aligned}\quad (3.66)$$

(3.66) denkleminde, d , ölçülemeyen bozucu vektörünü, B_D , B sistem matrisinin u kontrol girişi ve bozucuya ait alt matrisleri ifade eder. D_D , D matrisinin d bozucusuna ait alt matrisini gösterir. $B_D = 0$ olması durumunda, sistemde sadece çıkış bozucuları, $D_D = 0$ olması durumunda ise sistemde sadece giriş bozucuları olduğunu gösterir. En genel halde $d(k)$ giriş ve çıkış bozucularının vektörüdür.

(3.66) denklemi bir adım sonrası hesaplanırsa;

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + B_D d(k) \\y(k+1) &= Cx(k+1) + D_D d(k+1)\end{aligned}\quad (3.67)$$

Aynı şekilde bir adım daha ileri gidilirse;

$$\begin{aligned}x(k+2) &= Ax(k+1) + Bu(k+1) + B_D d(k+1) \\x(k+2) &= A[Ax(k) + Bu(k) + B_D d(k)] + Bu(k+1) + B_D d(k+1) \\x(k+2) &= A^2 x(k) + ABu(k) + AB_D d(k) + Bu(k+1) + B_D d(k+1) \\y(k+2) &= Cx(k+2) + D_D d(k+2)\end{aligned}\quad (3.68)$$

(k+3). adıma gelirsek;

$$\begin{aligned}x(k+3) &= Ax(k+2) + Bu(k+2) + B_D d(k+2) \\x(k+3) &= A[A^2 x(k) + ABu(k) + AB_D d(k)] + Bu(k+1) + B_D d(k+1) + Bu(k+2) + B_D d(k+2) \\x(k+3) &= A^3 x(k) + A^2 Bu(k) + ABu(k+1) + Bu(k+2) + A^2 B_D d(k) + AB_D d(k+1) + B_D d(k+2) \\y(k+3) &= Cx(k+3) + D_D d(k+3)\end{aligned}\quad (3.69)$$

n-adım sonrası için öngörü modeli:

$$\begin{aligned}x(k+n) &= A^n x(k) + A^{n-1} Bu(k) + A^{n-2} Bu(k+1) + \dots + Bu(k+n-1) + A^{n-1} B_D d(k) + \dots + B_D d(k+n-1) \\y(k+n) &= C[A^n x(k) + A^{n-1} Bu(k) + \dots + Bu(k+n-1) + A^{n-1} B_D d(k) + \dots + B_D d(k+n-1)] + D_D d(k+n)\end{aligned}\quad (3.70)$$

Böylece n-adım sonrasının gelecek öngörülleri vektör-matris formunda yazılırsa;

$$\begin{aligned}
\underbrace{\begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+n) \end{bmatrix}}_{\underline{x}(k)} &= \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix}}_{P_{xx}} x(k) + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ A^2B & AB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B & A^{n-2}B & \dots & B \end{bmatrix}}_{H_x} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix}}_{\underline{u}(k-1)} + \underbrace{\begin{bmatrix} B_D & 0 & \dots & 0 \\ AB_D & B_D & \dots & 0 \\ A^2B_D & AB_D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B_D & A^{n-2}B_D & \dots & B_D \end{bmatrix}}_{H_D} \underbrace{\begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+n-1) \end{bmatrix}}_{\underline{d}(k)} \\
\underbrace{\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+n) \end{bmatrix}}_{\underline{y}(k)} &= \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^n \end{bmatrix}}_P x(k) + \underbrace{\begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ CA^2B & CAB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-1}B & CA^{n-2}B & \dots & B \end{bmatrix}}_H \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix}}_{\underline{u}(k-1)} + D_d \underbrace{\begin{bmatrix} d(k+1) \\ d(k+1) \\ \vdots \\ d(k+n) \end{bmatrix}}_{\underline{d}(k)} \quad (3.71)
\end{aligned}$$

Durum ve girişlerdeki sağa ok işaretleri, vektörün sadece gelecek değerleri içerdiğini göstermektedir. Şimdiki ve geçmiş değerleri içeren vektör ise sola doğru ok ile betimlenmektedir. Daha sadece bir gösterimle, öngörü modeli (3.72) denklemi ile tanımlanacaktır:

$$\begin{aligned}
\underline{x}(k) &= P_{xx} \underline{x}(k) + H_x \underline{u}(k-1) + H_D \underline{d}(k) \\
\underline{y}(k) &= P \underline{x}(k) + H \underline{u}(k-1) + D_d \underline{d}(k) \quad (3.72)
\end{aligned}$$

(3.72) denklemi ile tanımlı durum-uzay modeline ait öngörü modelinde, bozucu etkilerin mevcut değerleri Kalman Filtresi ile öngörülür, ayrıca gelecekteki bozucu etkilerin değerleri hakkında bir bilgi sahibi olamadığımızdan, en iyi yaklaşımla sabit olduğu, $d(k+n) = d(k)$ olduğu [6] varsayımı yapılarak, öngörü modelde yerlerine konur.

Gözlemci modeli

Öngörülü kontrol yöntemlerinde durum gözlemcisi, iki temel görev üstlenir. Ölçülemeyen durum vektörlerinin sayısal değerlerinin ölçülen vektörlerden kestirmek, sisteme etki eden bozucu veya ölçüm gürültüsü olması durumunda, bozucu ve gürültünün durum vektörlerinin kestirilmesini yerine getirir. Bozucu ve gürültü ile modellenen sistemin, kestirim denklemi aşağıda verilmiştir:

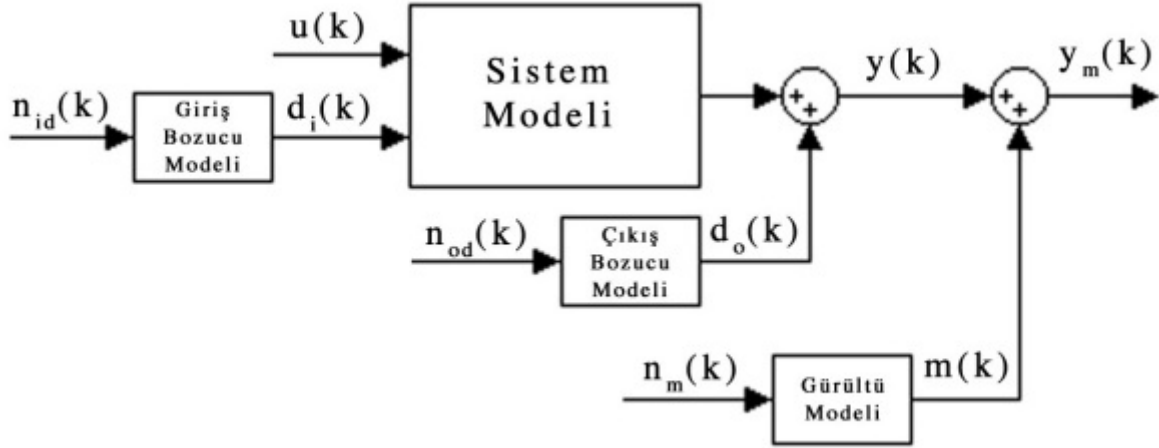
$$\begin{bmatrix} \hat{x}(k|k) \\ \hat{x}_d(k|k) \\ \hat{x}_m(k|k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k|k-1) \\ \hat{x}_d(k|k-1) \\ \hat{x}_m(k|k-1) \end{bmatrix} + K(y(k) - \hat{y}(k))$$

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}_d(k+1|k) \\ \hat{x}_m(k+1|k) \end{bmatrix} = \begin{bmatrix} A\hat{x}(k|k) + B_u u(k) + B_D \tilde{C}\hat{x}_d(k|k) \\ \tilde{A}\hat{x}_d(k|k) \\ \tilde{A}\hat{x}_m(k|k) \end{bmatrix}$$

$$y(k) = C\hat{x}(k|k-1) + D_d \tilde{C}\hat{x}_d(k|k-1) + C_m \hat{x}_m(k|k-1) \quad (3.73)$$

Alt indisler d ve m sırasıyla, bozucu ve gürültüye ait olduğunu gösterir. Ölçülen çıkış $y(k)$ ile gösterilir. K gözlemci kazancıdır.

Kullanılan sistem modelinin blok diyagramı Şekil 3.11’de verilmiştir.



Şekil 3.11 Sistem modeli

3.2.2.5 Amaç Fonksiyonu

Çeşitli MPC algoritmaları kontrol kanunu geçerli kılmak için farklı maliyet (ing. cost function) kullanır. Genel amaç, ilgili ufuktaki gelecek çıkışın $y(t)$, bir referans sinyali $r(t)$ takip ederken, aynı zamanda kontrol eforu pratikteki kısıtlamalar ve maliyet açısından önemli olduğu için enerji terimi olarak ifade edilebilir olmasıdır. Kontrol eforu, kontrol girişindeki değişim miktarı olarak adlandırılır. Kontrol değişimi (3.74) ile tanımlanmıştır. Amaç fonksiyonu, genel olarak (3.75) denklemi ile tanımlamak mümkündür:

$$\Delta u(k) = u(k) - u(k-1) \quad (3.74)$$

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta [y(t+j|t) - r(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 \quad (3.75)$$

Amaç fonksiyonunda ifade edilen N_1 ve N_2 sırasıyla minimum ve maksimum öngörü ufuklarına karşılık gelmekte olup, N_u kontrol ufku olarak adlandırılmaktadır. Kontrol edilmek istenen sistemde giriş ve çıkış arasında zaman gecikmesi varsa, minimum öngörü ufku bu gecikme kadar seçilmelidir. λ ve δ katsayıları ise gelecekteki sistem davranışı etkileyen ağırlık katsayılarıdır.

J amaç fonksiyonunun bir örnekleme adımında minimize edilmesi, öngörü ufku boyunca optimum performansı sağlayacak kontrol değişimlerinin hesaplanmasıyla sonuçlanır. Bu kontrol değişimlerinin vektörünün uzunluğu, kontrol ufkunun uzunluğuna eşittir.

3.2.2.6 Kısıtsız MPC problemi

Kontrol probleminde giriş, çıkış veya durumlar üzerinde hiçbir kısıtlama yoksa kapalı çevrim öngörülü kontrol problemi, kısıtsız MPC problemi olarak adlandırılır. Kısıtsız MPC probleminde kullanılan amaç fonksiyonu aşağıdaki gibi tanımlıdır:

$$J(k) = \sum_{j=N_1}^{N_2} [r(t+j) - y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 \quad (3.76)$$

$$\Delta u(k+j|k) = 0, j \geq N_u$$

s.t.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (3.77)$$

(3.77) ile tanımlı durum-uzay sistemi modeline ait, amaç fonksiyonu, öngörü ufku N_2 boyunca öngörülen çıkışların, istenilen referans değeri ile arasında farkının karelerinin toplamı ve kontrol ufku N_u boyunca kontrol değişimlerinin karelerinin λ katsayısı ile ağırlıklandırılmış toplamıdır. İkinci denklem kontrol ufkundan sonraki kontrol değişimlerinin sıfır olduğu varsayılır.

(3.77) ile tanımlı prosesin genel amaç fonksiyonu şu şekildedir:

$$J(k) = \sum_{k=1}^N \left\{ \sum_{j=N_1}^{N_2} [r(t+j) - y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 \right\} \quad (3.78)$$

Burada k adım sayısını, N ise prosesin çalışma süresini adım boyunca göstermektedir.

3.2.2.7 Kısıtlı MPC problemi

Endüstriyel prosesler pratikte belirli kısıtlamalar içerir. Bu yüzden optimizasyon problemini kısıtlı optimizasyon problemi olarak ele almak gerekmektedir. MPC'nin sanayide yaygın olarak kullanılmasının temel nedeni, proses kısıtlarını doğrudan hesaba katarak, optimizasyon problemini ele almasıdır.

Kontrol edilmek istenen prosesin giriş veya çıkış değerleri üzerinde kısıtlar olabilir. Fiziksel olarak ilgili sinyalin alt ve üst limitleri ile sınırlandırılmıştır. Girişe ait kısıtlar (3.79)'de , çıkışa ait kısıtlar (3.80)'de verilmiştir.

$$u_{\min} \leq u(k) \leq u_{\max} \quad (3.79)$$

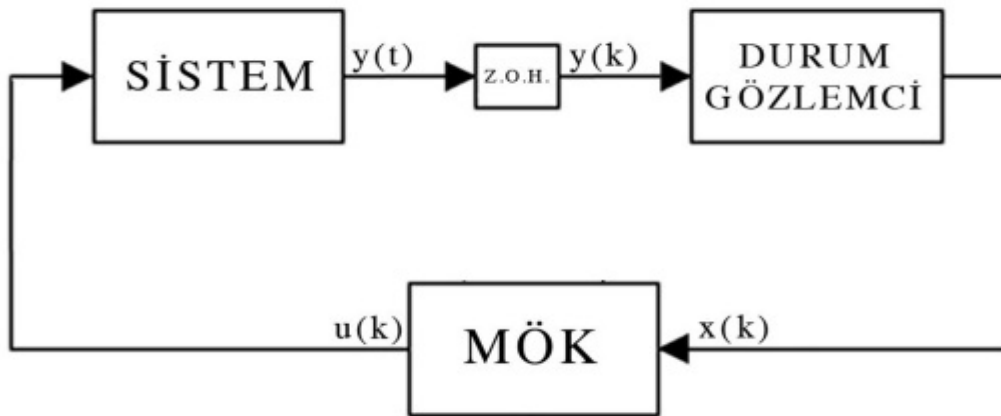
$$y_{\min} \leq y(k) \leq y_{\max} \quad (3.80)$$

3.2.2.8 MPC algoritması

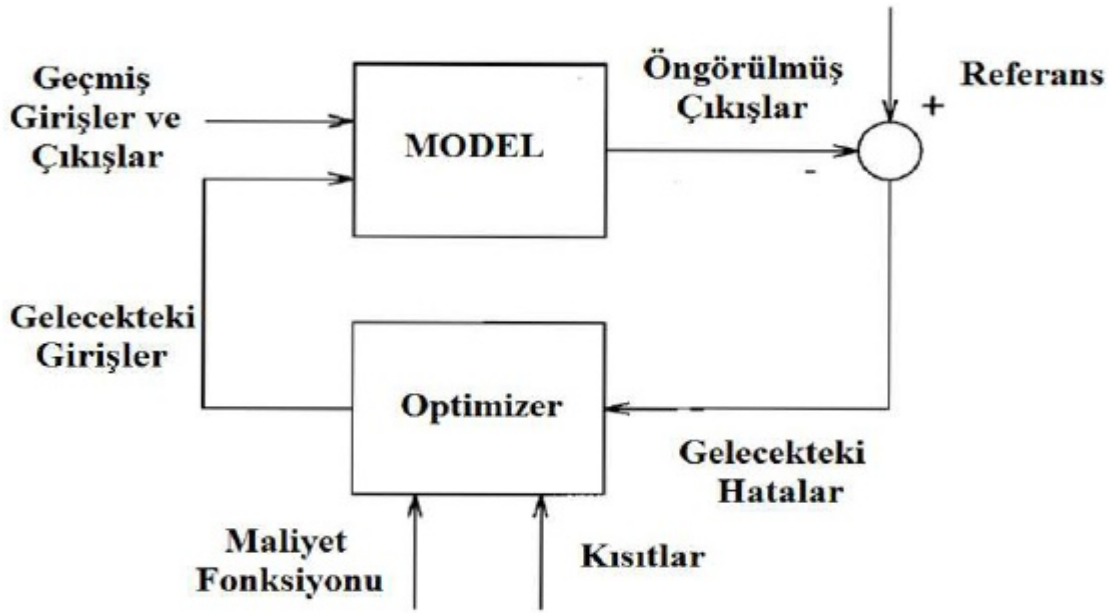
Ele alınan proses nonlinear ise, sistemi çalışma noktası etrafında doğrusallaştırılarak lineer MPC uygulanabilir. MPC algoritmasında üç ana adımdan oluşur:

- 1.x(k) durum değerlerinin ölçülmesi, ölçülemeyen durum vektörleri mevcut ise ölçülen çıkış y(k) 'dan gözlemlenebilirlik uygunsu kestirilmesi
- 2.Belirli olan amaç fonksiyonuna göre optimal kontrol kanunu bulunması
- 3.Bulunan optimal kontrol dizinin u(k), ilk değeri sisteme uygulanması

MPC algoritması kapalı çevrim ile MPC iç yapısı blok diyagramı Şekil 3.12 ve 3.13'de verilmiştir:



Şekil 3.12 MPC kapalı çevrimi



Şekil 3.13 MPC İç Yapısı

Kapalı çevrimde k . örnekleme adımında, mevcut adıma ait $y(k)$ çıkış değeri ölçülür. MPC algoritmalarında ayrık zamanlı sistemler kullanıldığından, gerçek prosesler ise sürekli (ing. continuous system) olduğundan, sistemden elde edilen verilerin ayrıklaştırılması (ing. discretization) gerekmektedir. Bunun için sıfırıncı mertebeli tutucu (ing. zero-order hold) sistem ölçüm değerlerinden sonra kullanılmaktadır. Çıkış değerinden ölçülemeyen durumlar veya sisteme ait bozucu etkiler elde edilir. MPC ana yapısı içerisinde, oluşturulan öngörü modele, bozucu etkiler, k . adıma ait sistem bilgileri optimizasyon rutini içerisinde belirli olan amaç fonksiyonunu kullanarak N_y öngörü ufku boyunca gelecekteki sistem çıkışlarını N_u kontrol ufku boyunca hesaplanan gelecekteki sistem girişlerine göre optimize eder. Bu sayede bulunan N_u kadar k . adımdaki optimal $[u(k), u(k+1), \dots, u(k+n)]$ kontrol dizininin ilk değeri sisteme gönderilir, ve bir sonraki adıma geçilir.

3.2.2.9 Kontrol kanununun bulunması

Kısıtsız MPC kontrol kanununun bulunması

Artımsal durum-uzay modeli ve kısıtsız hal için sistem ve amaç fonksiyonu aşağıda verilmiştir:

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$Y(k) = Cx(k)$$

$$J(k) = \sum_{j=N_1}^{N_2} [R(t+j) - Y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 \quad (3.81)$$

Bu sisteme ait öngörü modeli aşağıda verilmiştir:

$$\begin{aligned} \underline{x}(k) &= P_{xx} x(k) + H_x u(k-1) \\ \underline{Y}(k) &= P x(k) + H u(k-1) \end{aligned} \quad (3.82)$$

Burada;

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+n) \end{bmatrix} &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix} x(k) + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ A^2 B & AB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1} B & A^{n-2} B & \dots & B \end{bmatrix}}_{H_x} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix}}_{u^{(k-1)}} \\ \begin{bmatrix} Y(k+1) \\ Y(k+2) \\ \vdots \\ Y(k+n) \end{bmatrix} &= \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^n \end{bmatrix} x(k) + \underbrace{\begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ CA^2 B & CAB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-1} B & CA^{n-2} B & \dots & B \end{bmatrix}}_H \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix}}_{u^{(k-1)}} \end{aligned} \quad (3.83)$$

Sistemde herhangi bir bozucu etkinin olmadığı, modelleme hatasının olmadığı, ölçüm gürültüsünün olmadığı varsayalım.

Kısıtların olmadığı durumda, konveks optimizasyon probleminin çözümü analitik olarak mümkündür ve J(k) amaç fonksiyonun giriş değişkenine göre türevini sıfıra eşitlenerek elde edilir.

Referans değeri şu şekilde tanımlı olsun:

$$R^T = \underbrace{[1, 1, \dots, 1]}_{N_y} r(t) \quad (3.84)$$

Amaç fonksiyonunu tekrar düzenlersek

$$J = (R - Y)^T (R - Y) + \Delta U^T Q \Delta U \quad (3.85)$$

Q burada diagonal olarak tanımlıdır. Öngörü modelinden Y değerini J'de yerine yazarsak. $Q = \text{diag}(\lambda, \lambda, \dots, \lambda)$

$$J = (R - Px(k))^T (R - Px(k)) - 2\Delta U^T H^T (R - Px(k)) + \Delta U^T (H^T H + Q) \Delta U \quad (3.86)$$

Düzenlenen J amaç fonksiyonunu ΔU 'ya göre türevini alırsak

$$\frac{\partial J}{\partial \Delta U} = -2\Delta H^T (R - Px(k)) + 2(H^T H + Q) \Delta U \quad (3.87)$$

J'nin minimum durumu için sıfır olması gerekmektedir.

$$\frac{\partial J}{\partial \Delta U} = 0 \quad (3.88)$$

Buradan kontrol kanunu (3.89) denklemi ile bulunur.

$$\Delta U = (H^T H + Q)^{-1} H^T (R - Px(k)) \quad (3.89)$$

R referans sinyali içeren vektördür. Buradaki önemli nokta $(H^T H + Q)^{-1}$ var olup olmamasıdır. Optimizasyon algoritmalarında bu matrise Hessian matris denir. Sisteme bu hesaplanan kontrol kanunun ilk değeri gönderilir.

$$\Delta U = \underbrace{[1, 0, \dots, 0]}_{N_u} (H^T H + Q)^{-1} H^T (Rr(k) - Px(k)) \quad (3.90)$$

Buradan;

$$\Delta U = K_r r(k) - K_{MPC} x(k) \quad (3.91)$$

Çok giriş-çok çıkışlı sistemler içinde hesaplamalar aynı olmakta, dikkat edilmesi gereken durum matris ve vektörlerin boyutu olmaktadır.

Burada K_r ve K_{MPC} sırasıyla referans ve durum kazançlarını ifade eder. Belirsizliklerin olmadığı, kısıtlamaların bulunmadığı durumda analitik olarak ifade edilen MPC kanunu, durum geri beslemeli doğrusal kontrolcüye denktir.

Kısıtlı MPC kontrol kanununun bulunması

Kısıtlı haldeki MPC problemi farklı şekilde ele alınmalıdır.

Pratikte kısıtlar üç tipte bulunmaktadır. İlk ikisi kontrol sinyali üzerinde, diğeri sistem çıkışı veya durumları üzerinde bulunmaktadır. Kontrol sinyali üzerinde bulunan iki tip kısıt, kontrol sinyali değişimi ve değeri üzerindedir. Çıkış veya durumlardaki kısıtlarda değerleri üzerindedir. Bu durumda kısıtlı haldeki MÖK problemine ait amaç fonksiyonu ve kısıtlar şu şekilde tanımlıdır:

$$\min_{\Delta U} J(k) = \sum_{j=N_1}^{N_2} [R(t+j) - Y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 \quad (3.92)$$

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, u_{\min} \leq u(k) \leq u_{\max}, y_{\min} \leq y(k) \leq y_{\max}$$

Bu şekilde tanımlı, çok değişkenli, kuadratik amaç fonksiyonun kısıtlı optimasyonu Kuadratik Programlama (ing. Quadratic Programming, QP) ile çözülebilir.

Kısıtlı problem altında MPC algoritması, kontrol sinyalinin gelecekteki değişimlerinin değerleri ile ilgilendiğinden, gelecekteki giriş sinyalleri üzerinde de aynı kısıtlamalar bulunacaktır. Bunu ilk üç adım için yazılırsa:

$$\begin{aligned}
\Delta u_{\min} &\leq \Delta u(k) \leq \Delta u_{\max} \\
\Delta u_{\min} &\leq \Delta u(k+1) \leq \Delta u_{\max} \\
\Delta u_{\min} &\leq \Delta u(k+2) \leq \Delta u_{\max}
\end{aligned} \tag{3.93}$$

Bu yaklaşımla, kontrol sinyalindeki kısıtlar kontrol ufku boyunca tanımlanır. Hesaplama kolaylığı açısından tüm giriş datalarının yerine, daha küçük sayıda örnekler seçilebilir. Kısıtlamaların k . adımında $N_c = 4$ kontrol ufku boyunca ΔU cinsinden yazılırsa:

$$\begin{aligned}
\Delta U &= [\Delta u(k), \Delta u(k+1), \Delta u(k+2), \Delta u(k+3)] \\
u(k) &= u(k-1) + \Delta u(k) = u(k-1) + [1, 0, 0, 0] \Delta U \\
u(k+1) &= u(k) + \Delta u(k+1) = u(k) + \Delta u(k) + \Delta u(k+1) \\
u(k+1) &= u(k-1) + [1, 1, 0, 0] \Delta U
\end{aligned} \tag{3.94}$$

Çözüm önerisi için kullanılacak olan QP için, lineer eşitsizliklerin uygun formata dönüştürülmesi gerekir. Kontrol sinyalinin alt ve üst limit değerleri aşağıdaki gibi tanımlanır:

$$\begin{aligned}
\Delta U_{\min} &\leq \Delta U \leq \Delta U_{\max} \\
-\Delta U &\leq -\Delta U_{\min} \\
\Delta U &\leq \Delta U_{\max}
\end{aligned} \tag{3.95}$$

Matris formunda yazılırsa:

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U_{\min} \\ \Delta U_{\max} \end{bmatrix} \tag{3.96}$$

Giriş değerleri üzerinde kısıtlamaları da, değişim cinsinden ifade edilirse:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \cdot \\ \cdot \\ \cdot \\ u(k+N_c-1) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \cdot \\ \cdot \\ \cdot \\ I \end{bmatrix} u(k-1) + \begin{bmatrix} I & 0 & 0 & 0 \\ I & I & 0 & 0 \\ I & I & I & 0 \\ \cdot & \dots & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ I & I & I & I \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \cdot \\ \cdot \\ \cdot \\ \Delta u(k+N_c-1) \end{bmatrix} \tag{3.97}$$

(3.97)'te verilen giriş üzerindeki kısıtların, kuadratik programlamaya uygun formatı aşağıda verilmiştir:

$$\begin{aligned}
-(C_1 u(k-1) + C_2 \Delta U) &\leq -\Delta U_{\min} \\
(C_1 u(k-1) + C_2 \Delta U) &\leq \Delta U_{\max}
\end{aligned} \tag{3.98}$$

Yukarıda da bahsedildiği gibi, MPC algoritması gelecekteki datalarla ilgilendiğinden, çıkış değeri üzerinde kısıtlamalarda, gelecek değerleri açısından öngörü modelinden alınarak yazılır. Öngörü modeli ve kısıtlamalar aşağıda verilmiştir:

$$\begin{aligned} \underline{Y}(k) &= Px(k) + H\Delta U \\ Y_{\min} &\leq Px(k) + H\Delta U \leq Y_{\max} \end{aligned} \quad (3.99)$$

Kuadratik programlamaya uygun formatta tekrar yazılırsa:

$$\begin{aligned} Y_{\min} &\leq Px(k) + H\Delta U \leq Y_{\max} \\ -H\Delta U &\leq -Y_{\min} + Px(k) \\ H\Delta U &\leq Y_{\max} - Px(k) \end{aligned} \quad (3.100)$$

Minimize edilecek amaç fonksiyonu bu kısıtlar altında tekrar yazılırsa:

$$J = (R - Px(k))^T (R - Px(k)) - 2\Delta U^T H^T (R - Px(k)) + \Delta U^T (H^T H + Q)\Delta U \quad (3.101)$$

s.t

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \quad (3.102)$$

Burada

$$\begin{aligned} M_1 &= \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}; N_1 = \begin{bmatrix} -U_{\min} + C_1 u(k-1) \\ U_{\max} - C_1 u(k-1) \end{bmatrix} \\ M_2 &= \begin{bmatrix} -I \\ I \end{bmatrix}; N_2 = \begin{bmatrix} -\Delta U_{\min} \\ \Delta U_{\max} \end{bmatrix} \\ M_3 &= \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}; N_3 = \begin{bmatrix} -Y_{\min} + Px(k) \\ Y_{\max} - Px(k) \end{bmatrix} \end{aligned} \quad (3.103)$$

Denklemleri daha sade halde yazarsak:

$$M\Delta U \leq \gamma \quad (3.104)$$

3.2.2.10 Kuadratik Programlama

QP, standart bir optimizasyon problemidir, Bedel fonksiyonunun ikinci türevi olan Hessian matrisi, pozitif yarı-belirli bir matris ise problem konvektir. Bu durumda QP probleminin bir global optimumu vardır. Hessian matrisi pozitif tanımlı ise, bu global optimum tektir.

QP problemi nümerik olarak çözülür. Problemin çözümü için farklı yöntemler mevcuttur. İç nokta (ing. interior point), aktif küme (ing. active set), eşlenik gradyen

(ing. conjugate gradient) yaygın olarak kullanılmaktadır. Aşağıda QP'nin genel formu verilmiştir:

$$\begin{aligned} \min_x J &= \frac{1}{2} x^T E x + x^T F \\ s.t & \\ Mx &\leq \gamma \end{aligned} \quad (3.105)$$

Burada E, F, M ve γ kuadratik formdaki matris ve vektörleri temsil etmektedir. E simetrik ve pozitif tanımlıdır.

(3.101) ile tanımlı verilen amaç fonksiyonu ve kısıtlamalar tekrar yazılırsa:

$$\begin{aligned} \min_{\Delta U} J &= (R - Px(k))^T (R - Px(k)) - 2\Delta U^T H^T (R - Px(k)) + \Delta U^T (H^T H + Q)\Delta U \\ s.t & \\ M\Delta U &\leq \gamma \end{aligned} \quad (3.106)$$

Amaç fonksiyonunu minimize edecek ΔU bulunması durumu için, ilk terimler ΔU terimini içermemektedir. Amaç fonksiyonunu tekrar yazarsak:

$$\min_{\Delta U} J = -2\Delta U^T H^T (R - Px(k)) + \Delta U^T (H^T H + Q)\Delta U \quad (3.107)$$

(3.105) Kuadratik formda belirtilen duruma göre, amaç fonksiyonundaki değerleri aşağıda belirtilmiştir:

$$\begin{aligned} \min_{\Delta U} J &= -2\Delta U^T H^T (R - Px(k)) + \Delta U^T (H^T H + Q)\Delta U \\ \min_x J &= \frac{1}{2} x^T E x + x^T F \\ s.t & \\ Mx &\leq \gamma \\ x &= \Delta U, E = H^T H + Q, F = H^T (R - Px(k)) \end{aligned} \quad (3.108)$$

Bu durumda kuadratik programlama ile kısıtlı MPC algoritması çözülür.

3.2.2.11 Kararlılık

MPC kararlı sistemleri kontrol edebilir ve kararsızları kararlaştırabilir. Kısıtsız MPC probleminin analitik bir çözümü olduğundan ve sonuçta elde edilen kontrolcünün bir durum geri beslemeli kontrolcü olduğu ilgili bölümde gösterilmiştir. Kısıtsız MPC'de kararlılık analizi, bilinen doğrusal analiz yöntemleri kullanılarak yapılır. Sürekli zamanda kontrollü sistemin kutupların imajiner eksenin sol tarafında, ayrık zamanda tanımlı sistemlerin ise kutupların birim daire çemberin içinde kalması gerekmektedir. Kısıtlı MPC probleminde ise kontrollü sistemin kapalı çevrim davranışı nonlineer olduğundan, bu doğrusal metodlar kullanılamazlar.

Optimizasyon probleminin sonsuz ufuklar için çözülmesi, kararlılığı garanti edecektir. Kısıtsız halde, bu sonsuz ufuklu probleminin analitik çözümü Algebraic Riccati denkleminin çözümü olup LQR elde edilir. Kısıtlı halde analitik çözüm bulunmadığından sayısal çözümleme ele alınmalıdır. Aşağıda verilen problem için;

$$\min_{\Delta U} J(k) = \sum_{j=N_1}^{N_2} [r(t+j) - y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2$$

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, u_{\min} \leq u(k) \leq u_{\max}, y_{\min} \leq y(k) \leq y_{\max} \quad (3.109)$$

Bir QP problemidir ve çözülebilir. Fakat problem incelenince, deltau'nun sonsuz boyutlu olması, optimizasyon problemini sonsuz serbestlik dereceli yapar. Pratikte kısıtlı MPC problemi, sonsuz ufuklar için çözümlenemez.

Kısıtlı sistemlerde öngörülü kontrolünde kararlılığın sağlanması, amaç fonksiyonuna yeni terim eklenmesiyle gerçekleştirilebilir. Eklenen fonksiyon, öngörü ufkunun sonunda çıkış değerinin referanstan sapmasını cezalandırır. Bu şekilde tanımlı amaç fonksiyonu aşağıda verilmiştir:

$$\min_{\Delta U} J(k) = \sum_{j=N_1}^{N_2} [r(t+j) - y(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta(t+j-1)]^2 + P(x + N_2)$$

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, u_{\min} \leq u(k) \leq u_{\max}, y_{\min} \leq y(k) \leq y_{\max} \quad (3.110)$$

Bu eklenen P fonksiyonu, öngörü sonunda sistemin çıkışını cezalandırma durumuna, terminal cezası adı verilir ve sistemi referans değerine getirmeye zorlar.

3.2.2.12 Kontrol parametrelerinin ayarlanması

Optimizasyon probleminin kötü tanımlı olmaması için öngörülü kontrolde ayar parametreleri olan öngörü ufku, kontrol ufku ve giriş değişkeninin ağırlıklandırma faktörü ayarlanmalıdır. Çok giriş ve çıkışlı sistemlerde bu değişkenlere bir de çıkış ağırlıklandırma faktörü eklenir.

Bu kısımda değinilen konular, lineer sistem olan motor üzerinde gösterilecektir.

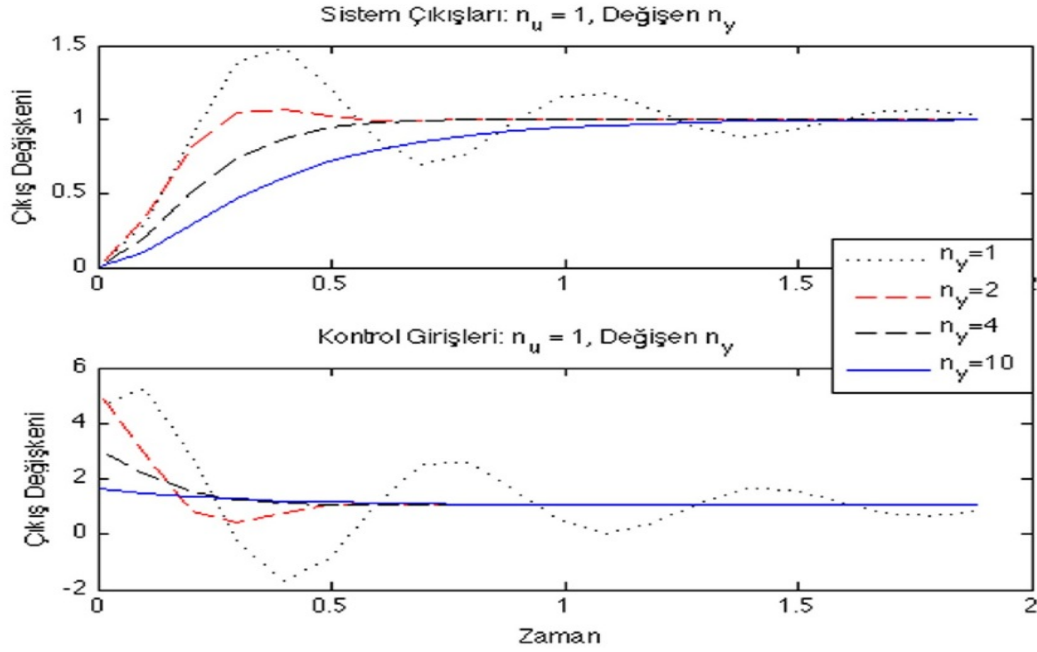
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -0.02 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$$

$$y = [9.549, 0]x \quad (3.111)$$

Öngörü ufkunun etkisi

Kontrol ufku 1 seçilmiş ve öngörü ufukları değiştirilmiştir. Öngörü ve kontrol ufuklarının birbirine eşit olduğu durumda, kontrolcü her adımda tanımlanan amaca

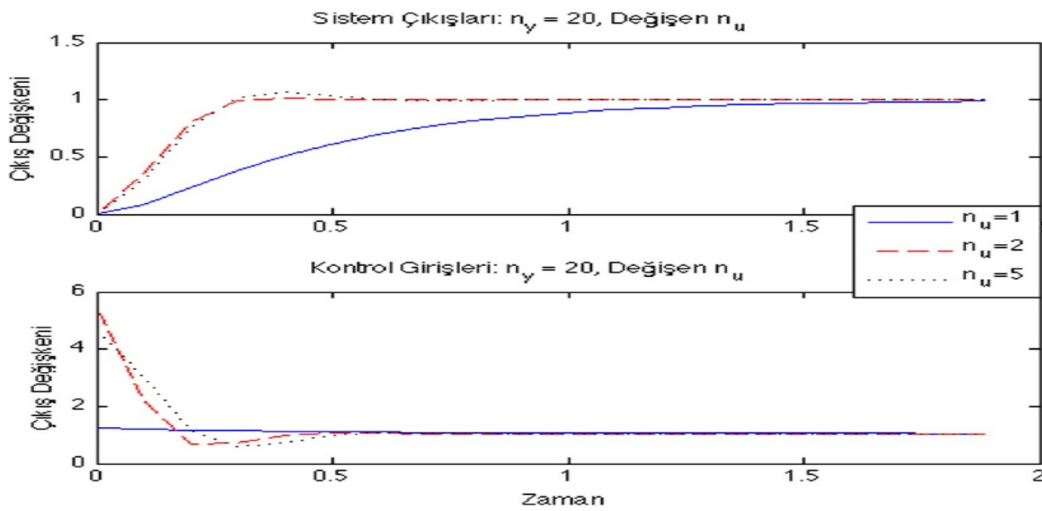
göre optimum kontrol girişini üretmekte, fakat istenen ayar değerine ulaşamamaktadır. Kısa kontrol ufku ve uzun öngörü ufku seçilmesi halinde sistem daha fazla sönümlü hale gelmektedir.



Şekil 3.14 Öngörü ufku etkisi

Kontrol ufkunun etkisi

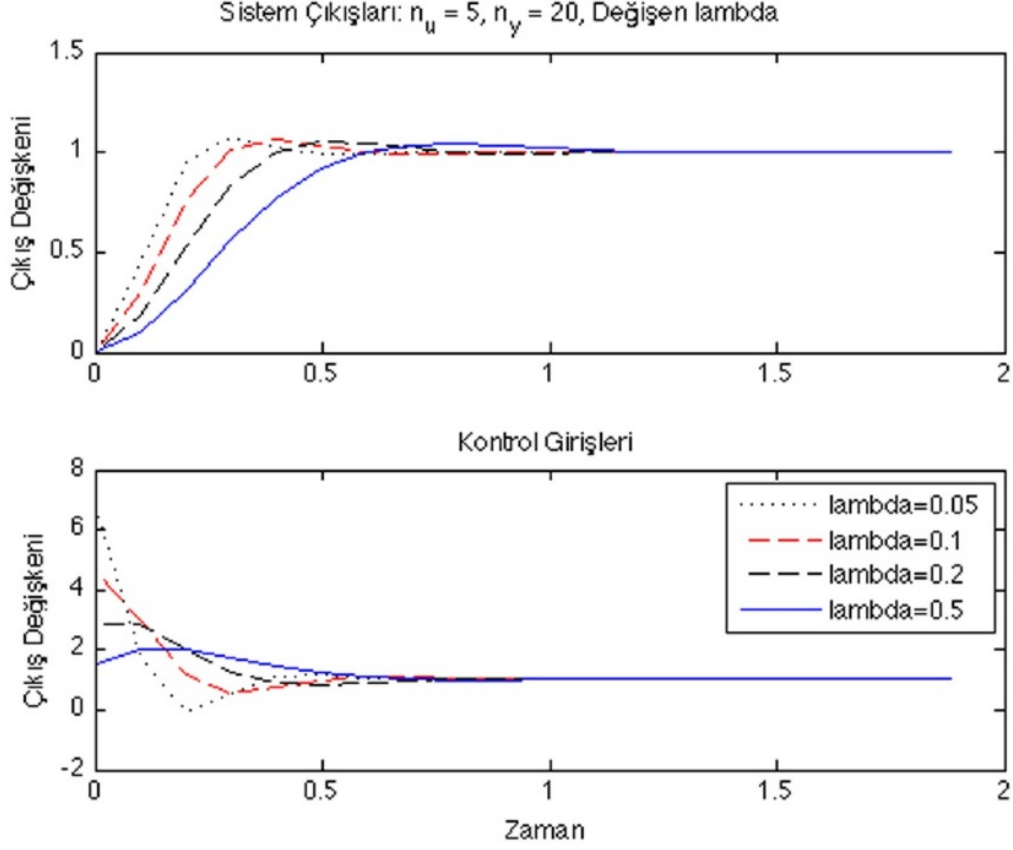
Öngörü ufku sabit tutulup, kontrol ufukları değişmiştir. Kontrol ufkunun artması daha agresif kontrol hamlelerine, azalması ise sönümlü kontrol değişimlerine neden olmaktadır.



Şekil 3.15 Kontrol ufku etkisi

Lambda ağırlık katsayısı etkisi

Sabit kontrol ve öngörü ufku seçilmiştir. Lambdanın büyüdükçe sistem yavaşlar ve kontrol sinyali daha fazla cezalandırıldığından sönümlü hale gelir.



Şekil 3.16 Lambda ağırlık katsayısı etkisi

Parametrelerin ayarlaması

- Öngörü ufku arttıkça nominal kapalı çevrim daranışı iyileşir
- Kontrol ufku arttıkça kapalı çevrim daranışı iyileşir.

Parametreler yeterince büyük seçilmelidir. İstenilen kontrol sinyalindeki sönüm değeri ise lambda ile seçilmelidir.

3.2.2.13 Nonlinear Sistemler için MPC Uygulaması

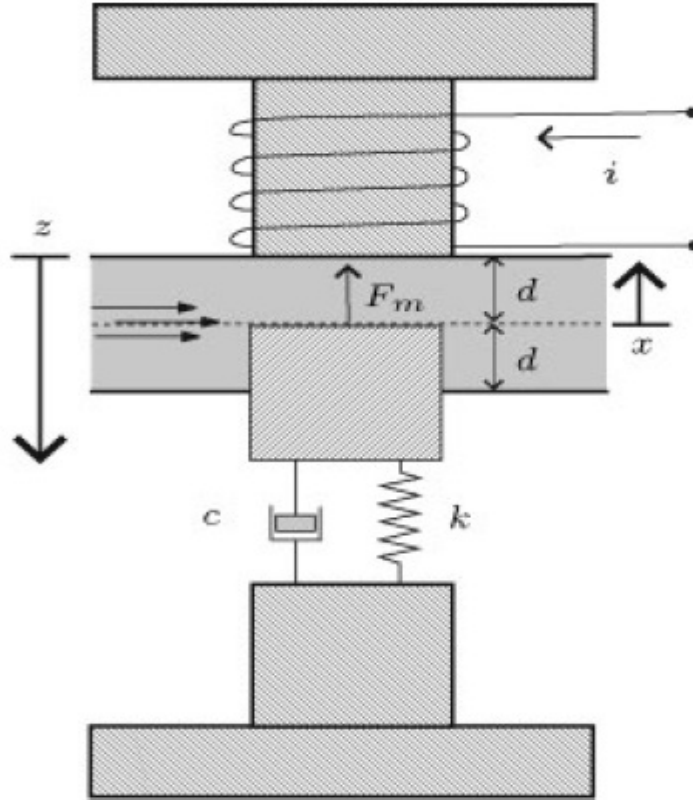
Şu ana kadar anlatılan MPC tasarımları, sistemin lineer olma durumunda geçerli idi. Gerçek hayatta ele alınan sistemler nonlinear davranış içermekte. Bunun için MPC tasarımında, sistemi çalışma noktası etrafında Taylor serisine açılarak lineer modeli elde edilmesi ve tasarımlarını o modele göre yapılması gerekmektedir.

Ayrı bir yöntem ise her ne kadar lineerleştirme işlemini gerçekleştiriyor olsada, bu işlem tek bir nokta da değil, sistemin durumlarının her adımda yeni değerleri üzerinden gerçekleşmektedir. Bu işleme ardışık lineerleştirme denir. (ing succesive linearization). Uygulamada kullanılan nonlinear modeller, MPC tasarımı yapılması için çalışma noktasında lineerleştirilmiştir.

4. BENZETİM SONUÇLARI

4.1 Elektromagnetik Tahrikli Kütle-Yay-Damper Sistemi

Elektromagnetik tahrikli kütle-yay-damper (EMKYD) sistemi kontrol edilmiştir. Bu tip sistemler, otomotiv sanayisinde, rulmanlarda ve bilim um eğlence için satılan oyuncaklarda (i.e. magnet ball) bulunabilir. Bu tip sistemde nonlineerlik ve kısıtlar mevcuttur. Şekil 4.1’de resmi verilmiştir:



Şekil 4.1 EKMYD sistemi

Sisteme ait matematik model şu şekildedir:

$$\ddot{x} = -\frac{k}{m}x - \frac{c}{m}\dot{x} + \frac{a}{m} \frac{C}{(d-x)^2}$$

$$-d < x < d$$

$$0 < C$$

(4.1)

Burada x , hareketli kütlenin bobine göre olan konumunu, \dot{x} hızını, \ddot{x} ise ivmesini göstermektedir. m, c, k, a ve C sistem parametreleridir. Kısıtlamalar ise, sistemin kontrol edilebilme durumu için, bobin ile temas halinde bulunmaması gerekmektedir. Bobin ile temas halinde olursa, magnetizmada ötürü, sistem ile bobin bir bütün olup, akım onun içinden de geçmeye başlayacak, bu durumda sistem üzerinde her hangi bir magnetik çekim kuvveti oluşturulmayacaktır. Bu da sistemin artık olmayacağı anlamına gelmektedir.

Bobine gelen akım (i), sıfırdan küçük olmamalıdır Elektrik sinyalinden, magnetik kuvvet oluşmakta, buda sisteme çekme kuvveti uygulaktadır. Kontrol edilmek istenilen durum ise, x konumunun belli bir referansta kalması sağlanmalıdır. Sisteme ait parametre değerleri **Çizelge 4.1**' de verilmiştir.

Çizelge 4.1 EMKYD sistem parametreleri

Pa rametre	eri	Deg
a		4.5e
γ	-5	1.99
c		.659
k	N/m.ms	38.9
m	4 N/m	1.54
	kg	

4.1.1 Amaç ve kapsam

Nonlinear ve kısıtlamalara sahip olan bu sistemin, kısıtlar ihlali gerçekleştirilmeden, nonlinear programlama algoritmaları ile MPC tasarımı gerçekleştirmek ve uygulamaktır. MPC için linear model tanıtılacak ve simulasyonlar gerçekleştirilecektir.

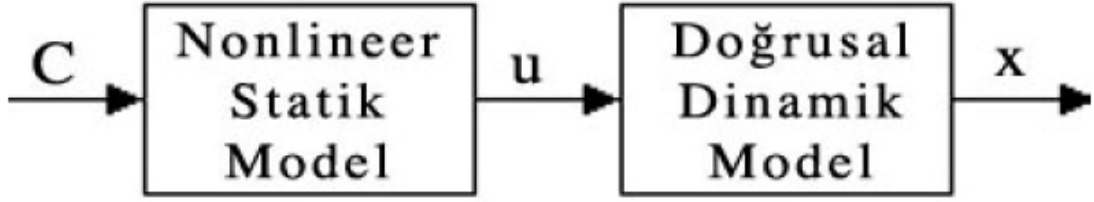
4.1.2 Sistem modeli

Giriş sinyalini ayrı bir parametre olarak ele alıp sistem denklemleri tekrar düzenlenirse;

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

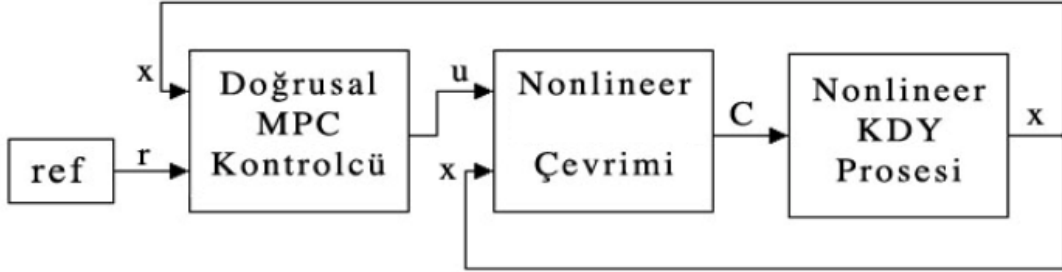
$$u = \frac{aC}{(d-x)^2} \quad (4.2)$$

Sistem bu şekilde ele alındığında lineer duruma gelmekte, gerekli giriş sinyali iste nonlineer bir denklem ile tanımlanmaktadır. Bu modellemeye ait şematik gösterim aşağıdaki şekilde verilmiştir:



Şekil 4.2 EMKYD sistem modeli

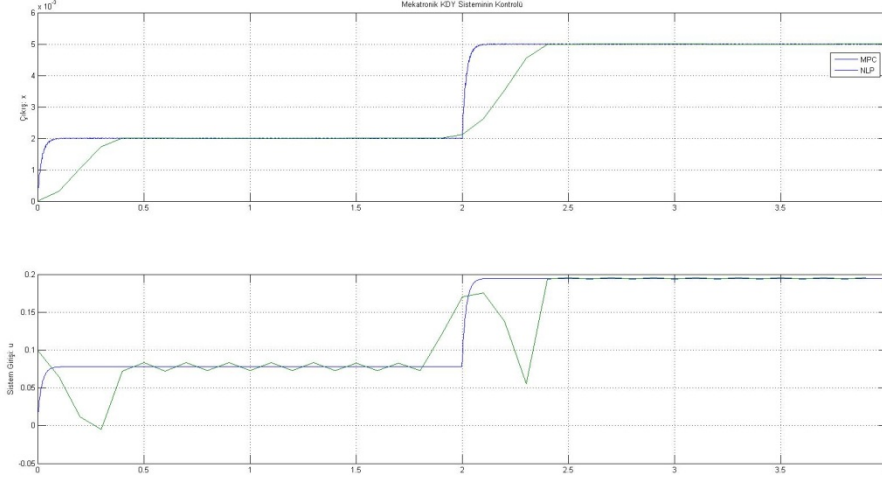
Bu yaklaşım ile, MPC tasarımı lineer sisteme uygulanır, hesaplanan u giriş sinyali, bir diğer çevrim ile hesaplanıp sisteme gönderilir. Buna ait blok diyagramı Şekil 4.3'te verilmiştir.



Şekil 4.3 Blok diyagramı

4.1.3 Sistemin kontrolü

Sisteme ait MPC için kullanılacak lineer model tanıldı ve kontrolü hem MPC hemde NLP algoritmaları kullanılarak yapılmıştır. NLP 'de sistemin lineerleştirilmesi gerekmediğinden, iki farklı çevrime gerek yoktur.



Şekil 4.4 Çıkış ve giriş değerleri

4.1.4 Sonuçlar

Sanayide ve eğlence endüstrisinde sıklıkla rastlanan bu sistem, lineerleştirme işlemi gerçekleştirilmeden uygulanarak kontrol kanunu NLP adı altında elde edilmiştir. MPC tasarımı için ise, lineer bir model önerilmiş ve onun üzerinden tasarlanmıştır. Şekil 4.4'te verilen giriş ve çıkışa ait simülasyon sonucundan da görüleceği üzere, MPC tasarımında lineer sistem kullanıldığından düzgün olarak çalışmaktadır. Öte yandan, NLP olarak ele alınan bu optimizasyon problemi, lineerleştirme işlemi yapılmadan uygulandığı için, ve kontrol sinyali üzerinde çıkışın karesi gibi bir değer olduğundan, giriş sinyali değişimleri çok düzgün olmamaktadır.

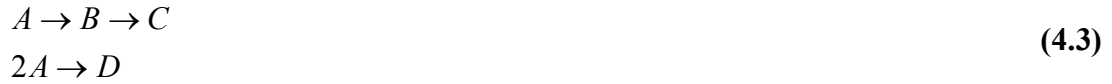
4.2 Van Der Vusse Tepkimesi

Nonlinear test problemlerinden sayılan, minimum faz olmayan, çok çıkışlı Van der Vusse tepkimesine NLP ve MPC algoritmaları uygulanacaktır.

Van Der Vusse Tepkimesine ait birkaç özellik aşağıda verilmiştir;

- Histeriz davranışı gösterirler
- Minimum faz değildir.
- Kaotik davranış sergilemekte, başlangıç koşullarındaki ufak değişiklik kararsızlık yaratabilmekte.
- Kaotik yapının kaynaklı sorununun iyi ele alınması gerekmektedir.

Van Der Vusse tepkimesine ait denklemler [7] şu şekilde tanımlıdır:



$$\begin{aligned}
 \frac{dC_a}{dt} &= (F/V)(C_{a0} - C_a) - k_1 C_a - k_3 C_a^2 \\
 \frac{dC_b}{dt} &= k_1 C_a - k_2 C_b - (F/V)C_b
 \end{aligned}
 \tag{4.4}$$

Burada F, A reaktörünü besleyen akış değeri, V sabit reaktör hacmi, Ca ve Cb ise reaktör içindeki bileşen miktarları, k, (4.3)'te bahsedilen üç reaksiyon için katsayılarıdır.

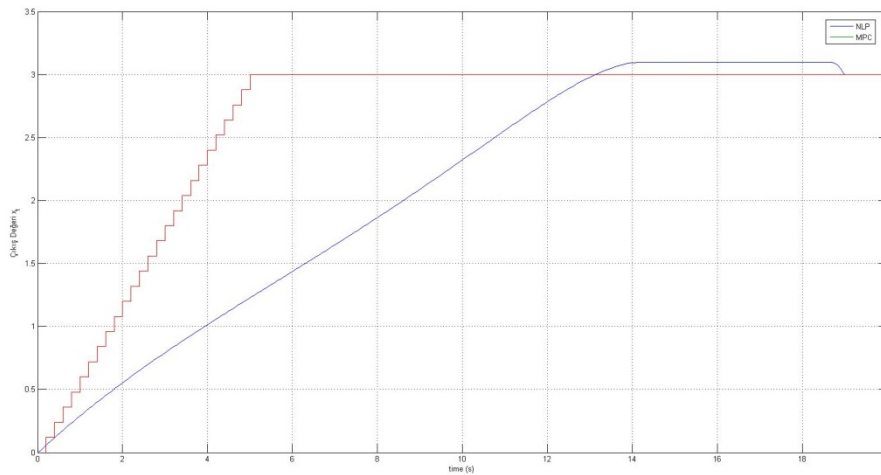
4.2.1 Amaç ve kapsam

Simülasyonun amacı, minimum faz olmayan nonlinear sistemi NLP ve MPC ile karşılaştırmaktır.

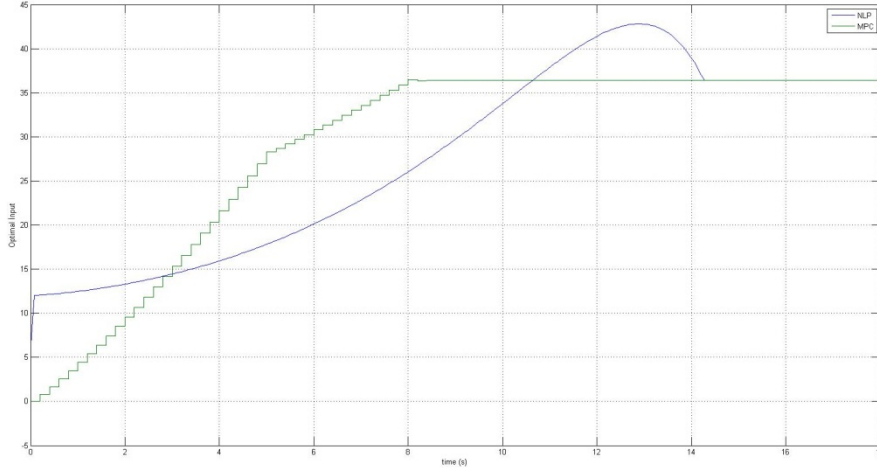
NLP bazında ele alındığında hiçbir işlem gerektirmeden SQP ile çözülmekte, MPC de ise lineerleştirme işlemi gerekmektedir. Zamanla değişen bir sistem olduğundan iteratif olarak lineerleştirme yapılmalıdır. Bu simülasyonda ardışık lineerleştirme kuralı uygulanmış olup, MPC bu multi noktada oluşan lineer sistemlere ait kontrolcülerdir.

4.2.2 Sistemin kontrolü

Simülasyonda belli bir referans değerine durağan halden getirilmesi hedeflenmiştir.



Şekil 4.5 Çıkış değeri

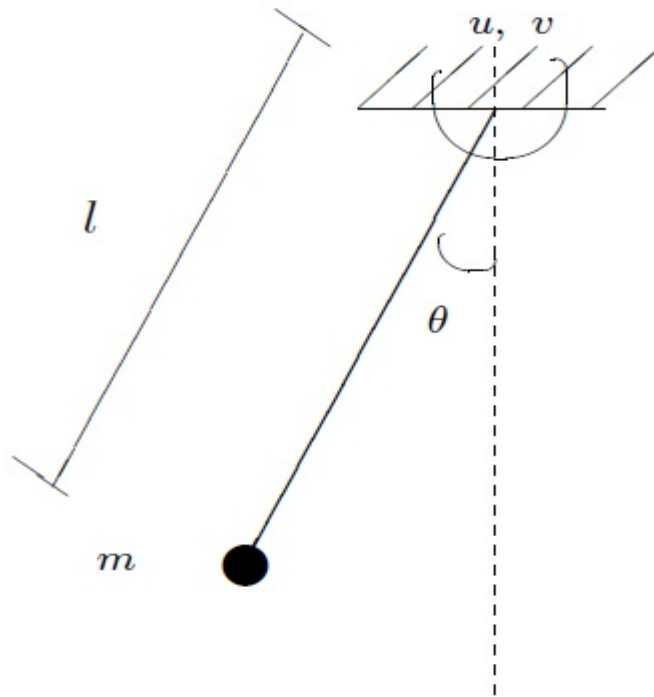


Şekil 4.6 Optimal Giriş Değerleri

4.2.3 Sonuçlar

Grafiklerinde görüldüğü üzere, MPC ve NLP minimum faz olmayan VDV sistemini kararlı bir şekilde hedeflenen referansa getirmiştir. MPC için ardışık linerizasyon yöntemi kullanıldığından NLP çözüm algoritmasına göre daha hızlı düzenli rejime oturtulabilmiştir, ve daha smooth bir yörüngeye sahiptir.

4.3 Tek Link Manipulator



Şekil 4.7 Tek Link Manipulatör

Sistemin girişte ve çıkışta bozucunun, sinyal ölçümünde de ölçüm gürültüsünün olsun. Sisteme ait durum-uzay modeli aşağıdaki gibidir. Parametre değerleri $m = 2 \text{ kg}$, $l = 1 \text{ m}$, $\nu = 6 \text{ kg m}^2 / \text{s}$ alınır;

$$\begin{aligned} \dot{x}_1 &= x_2 + d_k \\ \dot{x}_2 &= -9.81 \sin(x_1) - 3x_2 + 0.5u + d_k \\ y &= x_1 + w_k \end{aligned} \quad (4.5)$$

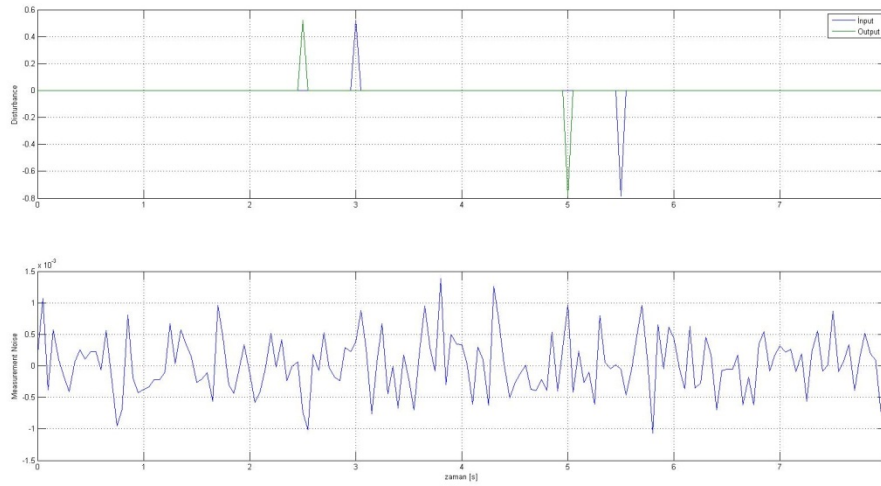
$$x_2 \leq 0.4$$

$$|u| \leq 8 \quad (4.6)$$

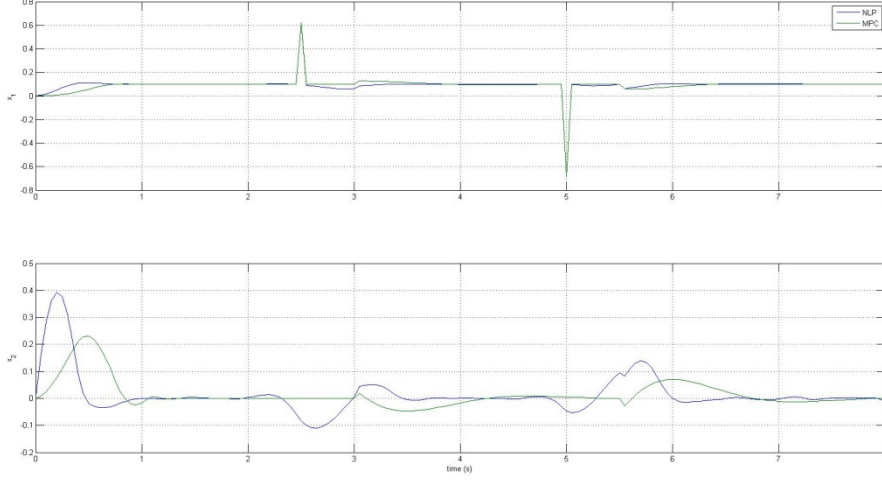
4.3.1 Amaç ve kapsam

Nonlinear, kısıtlamalara ve bozucu etkiye sahip olan bu fiziksel sistemi, kısıtların ihlal etmeden, nonlinear programlama algoritmaları ve MPC tasarımı gerçekleştirmek, uygulamaktır. MPC için lineerize edilmiş model kullanılmıştır ve simülasyonlar aşağıdaki şekildedir.

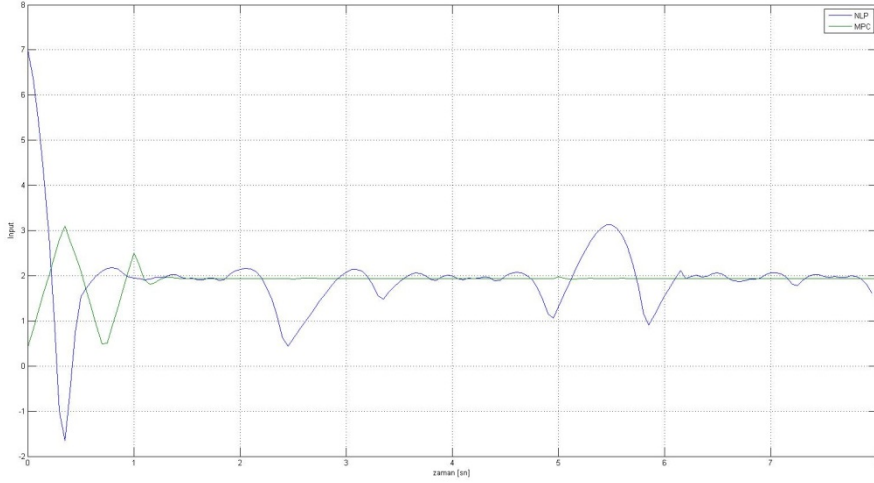
4.3.2 Sistem kontrolü



Şekil 4.8 Bozucu etki ve ölçüm gürültüsü



Şekil 4.9 Sistemin durumları



Şekil 4.10 Optimal u^*

4.3.3 Sonuçlar

Bozucu etki ve ölçüm gürültüsü etkisi altında çalışan tek link manipulator lineerleştirme işlemi gerçekleştirilmeden uygulanarak kontrol kanunu NLP adı altında elde edilmiştir. MPC tasarımı için ise, lineer bir model üzerinden tasarlanmıştır. Şekil 4.9 ve Şekil4.10'te verilen giriş ve çıkışa ait simulasyon sonucundan da görüleceği üzere, MPC tasarımında lineer sistem kullanıldığından düzgün olarak çalışmaktadır. Öte yandan, NLP bozucu etkisi yüzünden, sürekli bir giriş uygulamak durumunda kalmıştır.

5. SONUÇLAR ve DEĞERLENDİRME

Lineer olmayan programlama tarihte çok eskilere dayanmakta ve günümüzde dahi kullanılmaktadır. Artan teknoloji ile nonlineeritesi yüksek olan problemler artık hızla çözülebilmektedir. Buna nazaran gelişmiş kontrol metodları çıkmakta ve sanayide yaygın olarak kullanılmaktadır. Teorisi kanıtlanan programlama metodlarının ve model öngörülü kontrolün kısıtlı sistemlerde, kısıtları sistematik şekilde aldığı görülmektedir. Sanayide nonlineer sistemlerin kontrolünde bazen kararlılık sağlanamamaktadır.

Tez çalışmasında kısıtlı nonlineer sistemlerin kontrolü ele alınmış ve model öngörülü kontrol için doğrusal model ile çalışılmıştır. Her iki problemde, durum uzay modeli kullanılmıştır. Durum uzay modelinin, diğer modellere göre avantajları şöyle sıralanırsa, çok giriş-çok çıkışlı sistemler ile tek giriş-tek çıkış sistemler çözüm yöntemleri aynıdır, dikkat edilmesi gereken konu matrislerin büyüklüğüdür, ayrıca MPC için gözlemci tasarlanmasını mümkün kılmaktadır. NLP problemi için ise, iteratif yöntem kullanıldığından rahat olarak ele alınabilmektedir.

Bu çalışma farklı dinamik özelliklere sahip, kısıtlı nonlineer sistemlerin kontrolünde nonlineer programlama ve model öngörülü kontrolcülerin performansı incelenmiştir. Açık çevrim kararsız, zamanla değişen, minimum faz olmayan davranış gösteren nonlineer sistem de incelenmiştir.

Nonlinear optimal kontrol problemi, nonlineer programlama metodu şeklinde ele alındığında lineerleştirme işlemi gerçekleştirilmeden sonuç bulunabildiği görülmüştür. Bu da lineerleştirme ile doğabilecek sistem modelleme hatası, kaotik davranışlardan kaynaklı durumları irdelenmesini kolaylaştırmaktadır. MPC olayında durum lineerleştirmeye dayandığından, lineer sistemler üzerinde çalışmak, nonlineer sistemlere göre daha kolay ve hızlı olmuştur. MPC ' de en önemli kısıtlardan birisi de, gerçek sistem ile lineer sistemin, lineer bölge denilen, sistem davranışlarının birbirine çok yakın olduğu kısımda olması gerekmektedir. Nonlineer sistemlerin özelliği ise en ufak değişimin sistemde nasıl bir etki yaratacağı kesin olarak bilinmemektedir. Lineer bölgenin dışında MPC kontrol yöntemi kesin olarak çözüm bulabilir denememektedir.

NLP optimizasyon probleminde, günümüz teknolojinde MPC de bulunan kestirimci ve gözlemci modeli lineerleştirme işlemi yapılmadan gerçekleştirilememektedir. Nonlineer sistemlere uygulanan Genişletilmiş Kalman Filtresi, sistemin lineerleştirilmesine dayanmaktadır. Tez amacında hiçbir şekilde lineerleştirme olmadan kontrolcü tasarlanma olduğundan, ölçülemeyen sistem durumları var ise bu NLP metodunda kestirelemektedir.

Kısıtlı optimizasyon problemlerinde NLP ile alınması durumunda, bu kısıtların hiçbir şekilde bir sınırı olmamaktadır. MPC durumunda ise, kısıtların sadece ve sadece üst ve alt limit olarak tanımlanması gerekmektedir. MPC de lineer olmayan kısıtların tanımlanması kolay değildir. Buda iki kontrolcü yapısındaki önemli farklardan birisidir.

Gelecekte teorisi kesinleşildiğinde, lineerleştirme işlemi olmadan Nonlineer sistemlerde durum kestirimi metodları NLP ele alınabilir, ve MPC ' ye ait öngörü yeteneği NLP için genişletilebilir.

KAYNAKLAR

- [1] **Bertsekas , D.**,2003. Nonlinear Programming 2nd Edition, Athena Scientific
- [2] **Betts. T. J.**, 2009. Practical Methods for Optimal Control and Estimation Using Nonlinear Programming 2nd Edition, Siam
- [3] **Rossiter J.A.**,2005, Model Based Predictive Control, CRC Press
- [4] **Maciejowski, J. M.**,2002, Predictive Control With Constraints, Prentice Hall
- [5] **Chui, C.K.**, 2009. Kalman Filtering: with real-time applications, Springer Verlag
- [6] **Wang L.**, 2009, Model Predictive Control System Design and Implementation Using MATLAB, Springer
- [7] **Long C.E, Polisetty P.K, Gatzke E.P**,2008, Globally Optimal Nonlinear Model Predictive Control
- [8] **Bazaraa Z. M. Sherali D. H., Shetty M. C.**,1993, Nonlinear Programming Theory and Algorithms 2nd Edition. John Wiler & Sons Inc.
- [9] **Vincent L.T., Grentham J. W.**, 1997. Nonlinear and Optimal Control Systems, Springer
- [10] **Horst, R., Pardolos M. P., Thoai V. N.**, 2003, İntroduction to Global Optimization 2nd Edition, John Wiley & Sons Inc.
- [11] **Sun W., Yuan Y.**, 2006. Optimization Theory and Methods Nonlinear Programming, Springer
- [12] **Simon D.**, 2006. Optimal State Estimation Kalman, Hinf and Nonlinear Approaches, Wiley-Interscience
- [13] **Mangasarian L. O.**,1994. Nonlinear Programming, Siam
- [14] **Kouvaritakis B., Cannon M.**,2001. Nonlinear Predictive Control, theory and practice
- [15] **Biggs-Bartholomew M.**, 2000. Nonlinear Optimizastion With Engineering Applications, Springer
- [16] **Kalman, R. E.**,1960, Contributions to the theory of optimal control, *Bulletin de la Societe Mathematique de Mexicana*, 5, s102-119
- [17] **Kalman, R. E.**, 41960. A new approach to linear filtering and predictions problem, *Transactions of ASME, Journal of Basic Engineering*, 87,s.35-45
- [18] Matlab Control Systems Toolbox User's Guide,
<http://www.mathworks.com/products/control/>; Erisilen Tarih: 10.05.2011.
- [19] Matlab Model Predictive Control Toolbox User's Guide,
<http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/>
Erisilen Tarih: 11.05.2011

- [20] Matlab Optimization Toolbox User's Guide,
<http://www.mathworks.com/products/optimization>;
 Erisilen Tarih: 12.05.2011
- [21] **Tenny M.**, 2002. Computational Strategies for Nonlinear Model Predictive Control, Doktora Tezi, Wisconsin-Madison Universitei 233.s
- [22] **Rao S.S.**, 2009. Engineering Optimization, Theory and Practice, Wiley
- [23] **Lee H. J., Morari M., Garcia E.C.**, 2003. Model Predictive Control
- [24] **Griva I. Nash G. Stephen, Sofer A.**, 2009. Linear and Nonlinear Optimizastion 2nd Edition, Siam
- [25] **Luenberger G. D., Ye Y.**, 2008. Linear and Nonlinear Programming 3rd Edition, Springer
- [26] **Chong K. P. E., Zak H. S.**, 2001. An Introduction to Optimization , John Wiley & Sons
- [27] **Body S., Vandenberghe L.**, 2004. Convex Optimization, Cambridge University Press
- [28] **Isidori A.**, 1985. Nonlinear Control Systems: An Introduction, Springer-Verlag
- [29] **Khalil K. H.**, 1996. Nonlinear Systems, 2nd Edition, Prentice Hall
- [30] **Becarra V. M.**, Nonlinear Programming Lecture Notes
- [31] **Becarra V. M.**, Nonlinear Optimal Control Lecture Notes
- [32] **Olivieara S. L.**, 1996. Model Predictive Control for Constrained Nonlinear Systems
- [33] **Chui, C.K.**, 2009. Kalman Filtering: with real-time applications, Springer Verlag
- [34] **Badgwell, T.A., Muske, K. R.**, 2002 Disturbance model design for linear model predictive control, *Proceedings of the American Control Conference, 2002/2 p.1621-1626*
- [35] **Soderstrom T.**, 2002 Discrete-time stochastic systems : estimation and control, Springer Verlag
- [36] **Rothman D.H.**, Nonlinear Dynamics I : Chaos Lecture Notes, MIT
- [37] **Yurkevich D.V.**, 2004 Design of Nonlinear Control Systems with the Highest Derivate in Feedback, World Scientific
- [38] **YuMurli A, Toraldo G.**, 1997, Computational Issues in high performance software for Nonlinear Optimization, Kluwer Academic Publishers
- [39] **Tewari, A.**, 2002, Modern Control Design with Matlab and SIMULINK, Wiley.
- [40] **Prett, D.M., Gillette R.D.**, 1980. Model Predictive Control: Theory and practive : a survey, *Automatica*, 25,s335-348

EKLER

EK A.1 : KISITSIZ TLM DURUM GERİ BESLEME OPTIMAL KONTROL İÇİN MATLAB PROGRAMI

EK A.2 : KISITSIZ TLM OPTIMAL KONTROL İÇİN MATLAB PROGRAMI

EK A.3 : KISITLI TLM OPTIMAL KONTROL İÇİN MATLAB PROGRAMI

EK A.1

```
function f = simsfobj(x,h,Tfinal)

opt =
simset('solver','ode5','SrcWorkspace','Current','FixedStep',h);

K1 = x(1);
K2 = x(2);

[tout,xout,yout] = sim('simsf',[0 Tfinal],opt);

u=-K1*yout(:,1)-K2*yout(:,2);

im = length(yout(:,1));
w = ones(1,length(yout(:,1)));
w(length(w)) = 100;
D = diag(w);
f = 0.5*( (yout(:,1)-0.4)'*D*(yout(:,1)-
0.4)+(yout(:,2))'*(yout(:,2))+ (u-7.6404)'*(u-7.6404));

Tfinal = 2;
h      = 0.05;

x0=[0;0];

options=optimset('LargeScale','off','Display','iter','TolX',1e-
5,'TolFun',1e-5,'MaxIter',10,'Hessupdate','dfp');

[x] = fminunc('simsfobj', x0, options, h, Tfinal );

K1 = x(1);
K2 = x(2);

opt =
simset('solver','ode45','SrcWorkspace','Current','FixedStep',h);

[tout,xout,yout] = sim('simsf',[0 Tfinal],opt);

u=-K1*yout(:,1)-K2*yout(:,2);

figure;
subplot(2,1,1); plot(tout,yout(:,1));ylabel('x_1 NLP');grid on;
subplot(2,1,2); plot(tout,yout(:,2)),xlabel('time (s)');ylabel('x_2
NLP'); grid on;
figure;
plot(tout,u);xlabel('time (s)');ylabel('Optimal U');grid on
```

EK A.2

```
function f = simuobj(x,h,Tfinal)

opt =
simset('solver','ode5','SrcWorkspace','Current','FixedStep',h);
U = x;
T = [0:h:Tfinal-h]';
TU = [T U];

[tout,xout,yout] = sim('simu',[0 Tfinal],opt,TU);

im = length(yout(:,1));
w = ones(1,length(yout(:,1)));
w(length(w)) = 100;
D = diag(w);
f = 0.5*(yout(:,1)-0.4)'*D*(yout(:,1)-
0.4)+(yout(:,2))'*(yout(:,2))+ (u-7.6404)'*(u-7.6404));

Tfinal = 2;
h = 0.05;
N = Tfinal/h;
x0=zeros(N,1);

options=optimset('LargeScale','off','Display','iter','TolX',1e-
5,'TolFun',1e-5,'MaxIter',10,'Hessupdate','dfp');

[x] = fminunc('simsfobj', x0, options, h, Tfinal );

opt =
simset('solver','ode45','SrcWorkspace','Current','FixedStep',h);

[tout,xout,yout] = sim('simsf',[0 Tfinal],opt);

subplot(2,1,1);
plot(tout, yout(:,1),tout,r),xlabel('time (s)');ylabel('x_1 NLP');
subplot(2,1,2);
plot(tout, yout(:,2)),xlabel('time (s)');ylabel('x_2 NLP');
figure;
plot(tout(1:N),U),xlabel('time (sn)'),ylabel('input U');
```

EK A.3

```
function f = simuobj(x,h,Tfinal)

opt =
simset('solver','ode5','SrcWorkspace','Current','FixedStep',h);
U = x;
T = [0:h:Tfinal-h]';
TU = [T U];

[tout,xout,yout] = sim('simu',[0 Tfinal],opt,TU);

im = length(yout(:,1));
w = ones(1,length(yout(:,1)));
w(length(w)) = 100;
D = diag(w);
f = 0.5*( (yout(:,1)-0.4)'*D*(yout(:,1)-
0.4)+(yout(:,2))'*(yout(:,2))+ (u-7.6404)'*(u-7.6404));

function [g, geq] = simucon(x,h,Tfinal)
opt =
simset('solver','ode5','SrcWorkspace','Current','FixedStep',h);
U = x;
T = [0:h:Tfinal-h]';
TU = [T U];

[tout,xout,yout] = sim('simu',[0 Tfinal],opt,TU);
Tf = length( yout(:,1) );

% x2 hız kısıtlaması.
g = [yout(:,2)-0.4];
% x1 açılışta 0.4
geq1= yout(Tf,1)-0.4;
geq = [geq1];

Tfinal =2 ;
h = 0.05;
N = Tfinal/h;
x0=zeros(N,1);
% inputta constrainler
xmax = 8*ones(N,1);
xmin = -8*ones(N,1);
options=optimset('LargeScale','off','Display','iter','TolX',0.001,'TolFun',0.001,'MaxIter',10,'HessUpdate','bfgs');
x = fmincon('simuobj', x0, [],[],[],[],[],[], 'simucon',options, h, Tfinal );
% simulasyon
U = x;
T = [0:h:Tfinal-h]';
TU = [T U];
opt =
simset('solver','ode5','SrcWorkspace','Current','FixedStep',h);
[tout,xout,yout] = sim('simu',[0 Tfinal],opt,TU);
% plotlar.
r=0.2;
figure;
```

```
subplot(2,1,1);
plot(tout, yout(:,1),tout,r),xlabel('time (s)');ylabel('x_1 NLP');
subplot(2,1,2);
plot(tout, yout(:,2)),xlabel('time (s)');ylabel('x_2 NLP');
figure;
plot(tout(1:N),U),xlabel('time (sn)'),ylabel('input U');
```


ÖZGEÇMİŞ

Adı Soyadı: Turgay ERAY

Doğum Yeri ve Tarihi: ANTALYA, 1985

Lisans Üniversitesi: Dokuz Eylül Üniversitesi