

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK KULLANICILI ÇOK ANTENLİ AŞAĞI GÖNDERİM SİSTEMLERİ

YÜKSEK LİSANS TEZİ

Çınar YALÇINDURAN

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Ayşe Melda YÜKSEL TURGUT

TEMMUZ 2021

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Çınar YALÇINDURAN

ÖZET

Yüksek Lisans Tezi

ÇOK KULLANICILI ÇOK ANTENLİ AŞAĞI GÖNDERİM SİSTEMLERİ

Çınar YALÇINDURAN

TOBB Ekonomi ve Teknoloji Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ayşe Melda YÜKSEL TURGUT

Tarih: Temmuz 2021

Bu çalışmada çok kullanıcı, çok girdili çok çıktılı aşağı gönderim sistemlerinde (downlink MU-MIMO) kullanıcılar arası girişimi kontrol ederek toplam veri hızını artırmak için kullanılan önkodlama yöntemleri hatasız ve hatalı Rayleigh kanalları için incelenmektedir. Bu sistemler, baz istasyonundaki ve kullanıcılardaki anten sayısına ve kullanıcı sayısına göre az yüklü, kritik yüklü ya da aşırı yüklü olabilmektedir. Bu çalışmada, literatürde yeni önerilmiş olan genelleştirilmiş güç yineleme önkodlamasının (GPIP) ortalama karesel hata önkodlama (MMSE) ve veri hızı bölütleme (RS) ile karşılaştırması yapılmaktadır. GPIP yönteminin hatasız kanallarda tüm yük durumlarında, MMSE yöntemi ile benzer veya ondan daha kötü sonuçlar verdiği, üstelik MMSE yönteminin daha düşük karmaşıklığa sahip olduğu gözlemlenmiştir. GPIP yöntemi hatalı kanallarda ise tüm yük durumlarında MMSE yönteminden daha iyi, RS yönteminden daha kötü sonuç vermektedir ve MMSE yöntemine göre daha kısa sürede çalışmaktadır.

Anahtar Kelimeler: çok kullanıcı, çok girdili çok çıktılı sistemler (MU-MIMO), genelleştirilmiş güç yineleme önkodlaması (GPIP), minimum ortalama karesel hata önkodlaması (MMSE), yarı dik kullanıcı seçimli sifra zorlayıcı önkodlama (SUS-ZF), veri hızı bölütleme (RS)

ABSTRACT

Master of Science

DOWNLINK MULTI-USER MIMO SYSTEMS

Çınar YALÇINDURAN

TOBB University of Economics and Technology

Institute of Science and Technology

Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Ayşe Melda YÜKSEL TURGUT

Date: July 2021

In this paper, precoding methods used in downlink multi-user, multi-input multi-output systems (downlink MU-MIMO) for Rayleigh channels with perfect and imperfect channel knowledge are investigated. Precoding methods provide inter-user interference management and are designed to maximize the achievable sum rate. Downlink MU-MIMO systems can be underloaded, critically loaded or overloaded depending on the number of antennas at the base station and users and the number of users. This study provides a comparison between the generalized power iteration precoding (GPIP) (which is newly proposed in the literature) with the minimum mean square error precoding (MMSE) and rate splitting (RS). It is observed that for channels with perfect knowledge, the GPIP method results in similar or worse sum rate performance than the MMSE method in all load cases. Moreover, the MMSE method has lower complexity. For channels with imperfect knowledge, GPIP has better performance than MMSE method, but worse performance than RS method and performs faster than the MMSE in all different load cases.

Keywords: multi-user multiple input multiple output systems (MU-MIMO), generalized power iteration precoding (GPIP), minimum mean square error precoding (MMSE), semi-orthogonal user selection zero forcing precoding (SUS-ZF), rate splitting(RS)

TEŐEKKÖR

Yüksek lisans sırasında yaptığım çalıřmalarımnda deęerli yardım ve katkılarıyla beni yönlendiren ve destekleyen tez hocam Doç. Dr. Ayře Melda YÜKSEL TURGUT'a, kıymetli tecrübelerinden yararlandığım Doç. Dr. Tolga GİRİCİ'ye ve TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik ve Elektronik Mühendislięi Bölümü öğretim üyelerine çok teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
KISALTMALAR	ix
SEMBOL LİSTESİ	x
1. GİRİŞ	1
1.1 Motivasyon	1
1.2 Tezin Katkısı	2
1.3 Tezin Organizasyonu	2
2. LİTERATÜR ÖZETİ	3
2.1 5. Nesil Haberleşme (5G)	3
2.2 MIMO	5
2.3 Hüzmeleme ve Denkleştirme	7
2.4 Benzer Çalışmalar	10
3. SİSTEM MODELİ	14
3.1 Önkodlayıcı Özellikleri	14
3.2 Kullanıcı Hızlarının Toplamı	14
3.3 Kanal Bilgisi	15
4. GPIP	16
4.1 GPIP Problem Tanımı	16
4.2 GPIP Algoritması	17
5. MMSE	20
5.1 Hatasız Kanal için MMSE Algoritması	20
5.2 Hatalı Kanal için MMSE ve RS Algoritması	22
6. SUS-ZF	27
6.1 SUS-ZF Algoritması	27
7. SAYISAL SONUÇLAR	29
8. SONUÇ VE TARTIŞMA	37
KAYNAKLAR	38
ÖZGEÇMİŞ	40

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : eMBB, mMTC ve URRLC sistemleri için örnekler.....	3
Şekil 2.2 : Örnek bir MIMO sistemi.....	6
Şekil 2.3 : Örnek bir hüzemeleme yapılan MIMO sistemi.....	7
Şekil 2.4 : Örnek bir ZF Denkleştircisi.....	8
Şekil 7.1 : $N = 4$, hatasız kanal ve değişen SNR için az, kritik ve aşırı yüklü sistemler için toplam veri hızı grafiği.....	30
Şekil 7.2 : $N = 4$, hatalı kanal ve değişen SNR için az, kritik ve aşırı yüklü sistemler için toplam veri hızı grafiği.....	31
Şekil 7.3: $N = 4$, SNR = 20dB, hatasız kanal ve değişen kullanıcı sayıları için toplam veri hızı grafiği.....	32
Şekil 7.4: $N = 4$, SNR = 30dB, hatalı kanal ve değişen kullanıcı sayıları için toplam veri hızı grafiği.....	33
Şekil 7.5: $N = 4$ ve hatasız kanal bilgisi için kullanıcı sayısına göre değişen algoritma çalışma süresi.....	34
Şekil 7.6 : $N = 4$ ve hatalı kanal bilgisi için kullanıcı sayısına göre değişen algoritma çalışma süresi.....	35

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : eMBB, mMTC ve URRLC sistemleri için gereken özellikler.....	5
Çizelge 4.1 : GPIP algoritması	19
Çizelge 5.1 : MMSE algoritması.....	22
Çizelge 5.2 : RS algoritması.....	24



KISALTMALAR

5G	: 5. Nesil Haberleşme (5 th Generation Communication)
DoF	: Serbestlik Derecesi (Degree of Freedom)
DPC	: Kirli Kağıt Kodlaması (Dirty Paper Coding)
eMBB	: Geliştirilmiş Mobil Şebeke (Enhanced Mobile Broadband)
GMI	: Genelleştirilmiş Karşılıklı Bilgi (Generalized Mutual Information)
GPIP	: Genelleştirilmiş Güç Yineleme Önkodlaması (Generalized Power Iteration Precoding)
ISI	: Semboller Arası Girişim (Inter Symbol Inteference)
IUI	: Kullanıcılar Arası Girişim (Inter User Inteference)
KKT	: Karush Kuhn Tucker (Karush Kuhn Tucker)
MIMO	: Çok Girdili Çok Çıktılı Sistemler (Multiple Input Multiple Output Systems)
MMSE	: Minimum Ortalama Kareysel Hata Önkodlaması (Minimum Mean Square Error Precoding)
mMTC	: Geniş Ölçekli Nesnelerin İnterneti (Massive Machine Type Communications)
MRT	: Maksimum Oran İletimi (Maximum Raito Transmission)
MU-MIMO	: Çok Kullanıcılı, Çok girdili Çok Çıktılı Sistemler (Multi-User Multiple Input Multiple Output Systems)
RS	: Veri Hızı Bölütleme (Rate Splitting)
RZF	: Düzenlenmiş Sıfıra Zorlayıcı (Regularized Zero Forcing)
SIC	: Ardışık Girişim Giderimi (Successive Interference Cancellation)
SISO	: Tek Girdili Tek Çıktılı Sistemler (Single-Input Single-Output Systems)
SNR	: Sinyal Gürültü Oranı (Signal to Noise Ratio)
SNS	: Ardışık Sıfır Uzay Önkodlayıcısı (Successive Null Space Precoding)
SUS-ZF	: Yarı Dik Kullanıcı Seçimli Sıfıra Zorlayıcı Önkodlama (Semi-Orthogonal User Selection Zero Forcing Precoding)
URRLC	: Çok Düşük Gecikmeli Veri İletimi (Ultra Reliable and Low Latency Communications)
ZF	: Sıfıra Zorlayıcı (Zero Forcing)
ZF-DPC	: Sıfıra Zorlayıcı Kirli Kağıt Kodlaması (Zero Forcing Dirty Paper Coding)

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
A_k	k. kullanıcı için GPIP amaç fonksiyonunun pay kısmı
\bar{A}	A_k 'ların birleşimi
B_k	k. kullanıcı için GPIP amaç fonksiyonunun payda kısmı
\bar{B}	B_k 'ların birleşimi
D	MMSE'de tüm kullanıcılar için alıcı filtresi
d_k	MMSE'de k. kullanıcı için alıcı filtresi
ε	Algoritmalar için eşik değeri
$\varepsilon_{c,k}^{MMSE}$	RS'de k. kullanıcıdaki ortak semboller için MMSE denkleştirici
ε_k^{MMSE}	RS'de k. kullanıcıdaki özel semboller için MMSE denkleştirici
F	Önkodlama matrisi
\hat{F}	GPIP algoritması için tek boyuta çevrilmiş önkodlama vektörü
f_k	k. kullanıcı için önkodlama matrisi
$g_{c,k}^{MMSE}$	RS'de k. kullanıcıdaki ortak semboller için MSE matrisleri
g_k^{MMSE}	RS'de k. kullanıcıdaki özel semboller için MSE matrisleri
H	Kanal matrisi
h_k	k. kullanıcı için kanal matrisi
\hat{H}	Hatalı kanal matrisi
\hat{h}_k	k. kullanıcı için hatalı kanal matrisi

\tilde{H}	Kanal hatası matrisi
\tilde{h}_k	k. kullanıcı için kanal hatası matrisi
$I_{c,k}$	RS'de k. kullanıcı için ortak semboller dışındaki girişim
I_k	RS'de k. kullanıcı için özel semboller dışındaki girişim
K	Toplam kullanıcı sayısı
M	Hatalı kanallar için toplam kanal ölçüm sayısı
N	Baz istasyonundaki toplam anten sayısı
P_t	İleticide kullanılan toplam güç
R_k	k. kullanıcının anlık hızı
\bar{R}_k	k. kullanıcının ortalama hızı
$S_{c,k}$	RS'de k. kullanıcı için ortak sembollerin gücü
S_k	RS'de k. kullanıcı için özel sembollerin gücü
S_0	SUS-ZF'te seçilen kullanıcıların eklendiği küme
$T_{c,k}$	RS'de k. kullanıcı için ortak sembollerin gücü
T_k	RS'de k. kullanıcı için özel sembollerin gücü
$u_{c,k}$	RS'de k. kullanıcıda ortak sembollerin MSE matrisinin ağırlıkları
u_k	RS'de k. kullanıcıda özel sembollerin MSE matrisinin ağırlıkları
x_k	Baz istasyonundan k. kullanıcı için gönderilen sinyal
y_k	k. kullanıcıya gelen sinyal
z_k	Kanaldaki Gaussian gürültüsü
$\gamma_{c,k}$	RS'de k. kullanıcıda ortak semboller için SINR değeri
γ_k	RS'de k. kullanıcıda özel semboller için SINR değeri
Λ	Kanal bilgisi kalite değeri
Θ	Yarı diklik kararı için eşik değeri
$\lambda(F)$	GPIP'de amaç fonksiyonu
$\pi(i)$	SUS-ZF'de i. iterasyonda seçilen kullanıcı
Γ_i	SUS-ZF'de diklik karşılaştırması için seçilen kullanıcılar

1. GİRİŞ

1.1 Motivasyon

5G, 5. nesil yeni haberleşme standardıdır. Bu standardın günlük hayatta daha eski nesillerdeki standartların yerine geçmesi hedeflenmektedir ve bu amaca yönelik çalışmalar yapılmaktadır. Bu standardın daha önceki nesillere göre olan önemli bir avantajı daha bütünleyici olmasıdır. 5G hava ara yüzü ve spektrumu LTE ve WiFi ile birlikte kullanarak daha önceki nesillere göre daha yüksek maksimum hız, daha düşük gecikme süresi ve bağlanılabilir cihaz sayısının daha yüksek olması gibi avantajlara sahiptir. Bu avantajlardan dolayı günümüzde 5G'nin ve kablosuz haberleşmenin kullanım alanları giderek artmaktadır [1].

Günümüzde 5G'nin kullanıldığı alanlar arttığı ve bu alanlarda daha yüksek başarı hedefleri kullanıldığı için, bunları sağlayabilecek daha yüksek karmaşıklığa sahip sistemlerin kullanılması gerekmektedir. Kablosuz haberleşmede kullanılan bu karmaşık sistemlerden birisi, aynı anda birden fazla kullanıcı ile haberleşebilen çok kullanıcı, çok girdili çok çıktılı (multiuser multiple input multiple output, MU-MIMO) sistemlerdir [2].

MU-MIMO sistemleri, alıcıda kanal ile ilgili yeterince bilgi olduğu durumlarda SISO sistemlere göre çok daha yüksek hızlara ulaşma potansiyeline sahiptir [3]. Hücreli sistemlerde MU-MIMO'da aynı anda birden fazla kullanıcıya iletim yaparak kullanıcı hızlarının toplamı artırılabilir [2]. Aynı anda birden fazla kullanıcıya iletim yapılması kullanıcılar arası girişime neden olmaktadır. Kullanıcılar arası girişim, MU-MIMO sistemler için oldukça önemli bir problemdir ve girişimin toplam veri hızını artıracak şekilde azaltılması gerekmektedir [4]. Baz istasyonunda iletimde kullanılan önkodlama tekniklerinin kullanıcılar arası girişimi azaltacak şekilde kullanılması MU-MIMO sistemin başarısı konusunda oldukça önemlidir.

1.2 Tezin Katkısı

GPIP, literatürde bulunan diğer önkodlama yöntemlerine oldukça yeni bir yöntemdir. İlk defa Giuseppe Caire'nin 2020 yılında yazdığı "Joint User Selection, Power Allocation, and Precoding Design With Imperfect CSIT for Multi-Cell MU-MIMO Downlink Systems" isimli makalede kullanılmıştır. Bu makalede kullanıcıların kanal durumları birbiriyle ilintilidir ve sonuçlar kanal bilgisinin hem hatasız hem de hatalı olduğu durumlar için karşılaştırılmıştır. Bu makalede kullanılan MU-MIMO sistemlerde, kullanıcı sayısının anten sayısına eşit ya da anten sayısından daha az olduğu durumlar incelenmiştir. Elde edilen sonuçlar ZF-DPC, Sıfıra Zorlayıcı Kerte Uyarlama (Rank Adaptation with ZF), SUS-ZF, RZF, ZF ve MRT ile karşılaştırılmıştır [4].

Bu çalışmada, kullanıcı sayısının anten sayısından fazla ya da anten sayısına eşit olduğu, birbirinden bağımsız ve eş dağılımlı Rayleigh kanalları kullanılarak kanal bilgisinin hatasız veya hatalı olduğu durumlarda GPIP hakkında bilgi edinmiştir. Literatürde GPIP'in bu özelliklere ait bir MU-MIMO sisteminde kullanıldığına dair bir çalışma bulunmamaktadır ve bu sayede elde edilen sonuçlar kullanılarak GPIP hakkında yeni bilgiler elde edilmektedir.

Bu çalışmada literatürdeki çalışmalardan farklı olarak GPIP, MMSE ile karşılaştırılacaktır. [4]'te MMSE ile ilgili bir karşılaştırma yapılmamıştır. MMSE literatürde önemli bir yere sahip, üzerine çalışmalar yapılmış ve sonuçları bilinen bir yöntemdir. GPIP ile MMSE'in sonuçları karşılaştırarak GPIP ile ilgili yeni bilgiler elde edilmesi hedeflenmektedir.

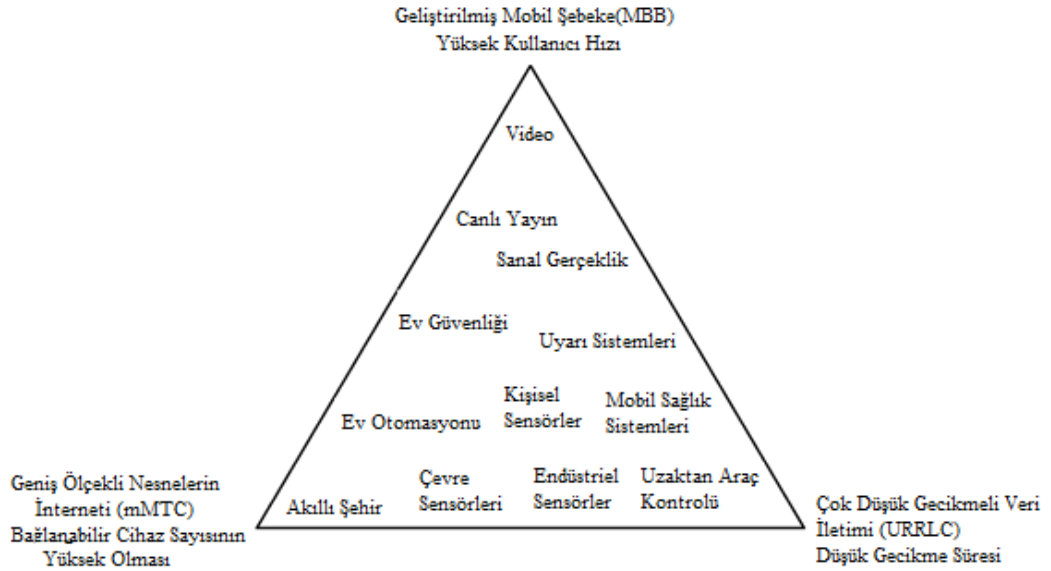
1.3 Tezin Organizasyonu

Bu çalışmada II. bölümde literatür özeti, III. bölümde bu çalışmada kullanılacak sistem modeli anlatılmıştır. IV. bölümde GPIP, V. bölümde MMSE ve RS, VI. bölümde ise SUS-ZF yöntemleri gösterilmiştir. VII. Bölümde sayısal sonuçlar sunulmuş ve VIII. bölümde bu çalışma sonucu elde edilen sonuçlar açıklanmıştır

2. LİTERATÜR ÖZETİ

2.1 5. Nesil Haberleşme (5G)

Son yıllarda, haberleşme uygulamalarındaki kullanıcı sayısının artması ve daha yüksek veri hızlarına duyulan ihtiyaç nedeniyle 4. nesil haberleşme(4G) ve LTE(Long Term Evolution) teknolojilerinin yerini alacak yeni nesil bir standarda ihtiyaç duyulmaktadır. 5. nesil haberleşme(5G) teknolojisi bu sorunları çözmeyi hedeflemektedir. Günümüzde bu teknoloji üzerine hem akademik hem de sektörel olarak yoğun çalışmalar yapılmaktadır. 5G'nin önceki haberleşme standartlarına göre avantajları daha yüksek maksimum veri iletim hızı, daha kısa gecikme süresi ve daha yüksek sayıda cihaz için bağlanılabilirliktir. 5G'de tasarlanan sistemler, başarı kriteri olarak bu avantajlardan bir ya da birden fazlasını içermektedir. Bu sistemler şu şekilde adlandırılmıştır: Geliştirilmiş Mobil Şebeke (Enhanced Mobile Broadband, eMBB), Geniş Ölçekli Nesnelerin İnterneti (Massive Machine Type Communications, mMTC) ve Çok Düşük Gecikmeli Veri İletimi (Ultra Reliable and Low Latency Communications, URRLC) [5].



Şekil 2.1: eMBB, mMTC ve URRLC sistemleri için örnekler

Geliştirilmiş Mobil Şebeke'lerde(eMBB) en önemli başarı kriteri maksimum veri iletim hızıdır. Bu sistemlerin kullanıldığı alanlar şu şekildedir: Çok yüksek iletim hızı gerektiren ses ve video gösterilimi yapılan yerler, kullanıcı sayısının fazla olduğu ofis, mağaza gibi yerler, okul, ofis gibi sadece belli bir alanda iletim yapılan yerler, araçlar ya da hareket eden yayalarda bulunan sistemler, cep telefonu gibi veri hızının değişken olduğu sistemler ve sabit ve hareketli sistemlerin aynı anda kullanılabilirdiği alanlar [5].

Geniş Ölçekli Nesnelerin İnterneti'nde(mMTC) en önemli başarı kriterleri aynı anda çok fazla cihazın bir ağa bağlanabilmesi ve bunu yaparken ucuz ve düşük enerji gerektirmesidir. Bu sistemlerin ana kullanım alanları nesnelerin interneti(internet of things, IoT), saat, bileklik gibi giyilebilir akıllı cihazlar ve sensör ağlarıdır [5].

Çok Düşük Gecikmeli Veri İletimi'nde(URRLC) çok yüksek güvenilirlik ve çok düşük gecikme süresi hedeflenmektedir. Bu sistemler genellikle uzaktan sürülen araçlar ve dronelar, fabrikalarda kapalı döngü devreleri gibi otomasyon sistemlerinde kullanılıyordur [5].

Geliştirilmiş Mobil Şebeke(eMBB), Geniş Ölçekli Nesnelerin İnterneti(mMTC) ve Çok Düşük Gecikmeli Veri İletimi'nde(URRLC) sistemin başarısını ölçmek için çeşitli başarı kriterleri kullanılmaktadır. Bu kriterler kullanılarak sistemin, sistemde belirtilen ihtiyaçları karşılayıp karşılaşamadığı görülmektedir. Bu sistemlerde kullanılan başarı kriterleri örnek olarak şunlardır: En yüksek veri hızı, spektrum verimliliği, bağlantı yoğunluğu, enerji verimliliği, gecikme süresi ve hareket edilebilirliktir. Bu özelliklerin sistem üzerindeki önemi sistemin Geliştirilmiş Mobil Şebeke(eMBB), Geniş Ölçekli Nesnelerin İnterneti(mMTC) ya da Çok Düşük Gecikmeli Veri İletimi (URRLC) olup olmadığına göre değişmektedir. Bu sistemler için gereken özellikler Çizelge 2.1'de gösterilmiştir [6].

	En Yüksek Veri Hızı	Spektrum Verimliliği	Bağlantı Yoğunluğu	Enerji Verimliliği	Gecikme Süresi	Hareket Edilebilirlik
eMBB	Önemli	Önemli	Biraz Önemli	Önemli	Biraz Önemli	Önemli
mMTC	Önemsiz	Önemsiz	Önemli	Biraz Önemli	Önemsiz	Önemsiz
URLLC	Önemsiz	Önemsiz	Önemsiz	Önemsiz	Önemli	Önemli

Çizelge 2.1: eMBB, mMTC ve URLLC sistemleri için gereken özellikler

5G’de Geliştirilmiş Mobil Şebeke(eMBB), Geniş Ölçekli Nesnelerin İnterneti(mMTC) ve Çok Düşük Gecikmeli Veri İletimi(URLLC) sistemlerini tasarlayabilmek için çeşitli teknolojiler kullanılmaktadır. Geliştirilmiş Mobil Şebeke’de(eMBB), MU-MIMO, Geniş Ölçekli Nesnelerin İnterneti’nde(mMTC) ultra yoğunlaştırma ve Düşük Gecikmeli Veri İletimi’nde(URLLC) milimetre dalgaları (mmWave) kullanılmaktadır [1]. Bu çalışmada GPIP ve MMSE yöntemleri karşılaştırılırken başarı kriteri kullanıcı hızlarının toplamı olarak seçilmiştir ve bu yüzden bu sistem bir Geliştirilmiş Mobil Şebeke(eMBB) olarak düşünülebilir ve bu sistemi başarılı bir şekilde çalıştırabilmek için MU-MIMO kullanılması gerekmektedir.

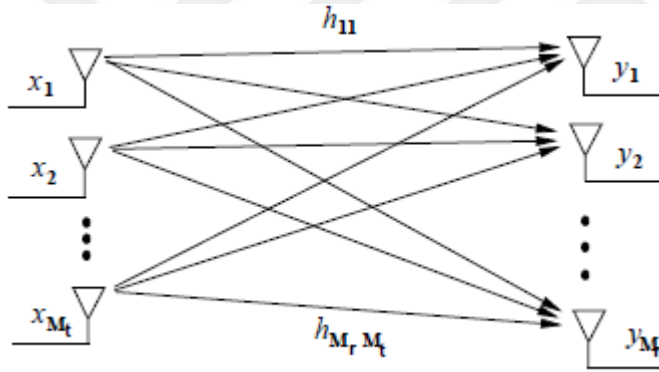
2.2 MIMO

Kullanıcıların toplam hızının önemli olduğu bir sistemde, MU-MIMO sistemler SISO sistemlere göre tercih edilmektedir. Özellikle antenlerin birbirinden bağımsız olduğu durumlarda, MIMO sistemler birbirinden bağımsız birden fazla SISO sistemin bir araya gelmesi gibi düşünülebilir ve sistemdeki bu bağımsızlıklardan yararlanılarak MIMO sistemlerin çok daha yüksek kapasiteye sahip olduğu hesaplanmıştır [3].

Çoklu antenler, sistemin ihtiyacına göre uzamsal çoklama (spatial multiplexing) yaparak toplam hızı artırmaktadır ya da iletim çeşitlemesinden (transmit diversity)

yararlanarak sistemin güvenilirliğini artırmaktadır. Çoklamada kanal kazanç matrisi kullanılarak veriyi gönderebilecek birbirinden bağımsız yolların oluşturulması amaçlanmaktadır. Gönderilmesi gereken verinin tamamı, elde edilen yol sayısına bölünerek, bu yollardan aynı anda gönderim yapılır. Bu sayede toplam veri bir SISO sisteme göre çok daha hızlı iletilir [7].

MIMO sistemlerde güvenilirliğin daha önemli olduğu durumlarda çeşitleme kullanılır. Çeşitlemede kanaldaki sönümlenmenin sistem üzerindeki etkisinin azaltılması amaçlanmaktadır. Çeşitlemede tıpkı çoklamada olduğu gibi sistemde birbirinden bağımsız yollar kullanılmıştır ama çoklamadan farklı olarak tüm yollardan aynı semboller gönderilmiştir. Bu sayede sembollerin gönderildiği yollardan bir kısmı sönümlenmeye uğrasa da sönümlenmeden en az etkilenen yollar kullanılarak güvenli veri iletimi yapılmaktadır [7].

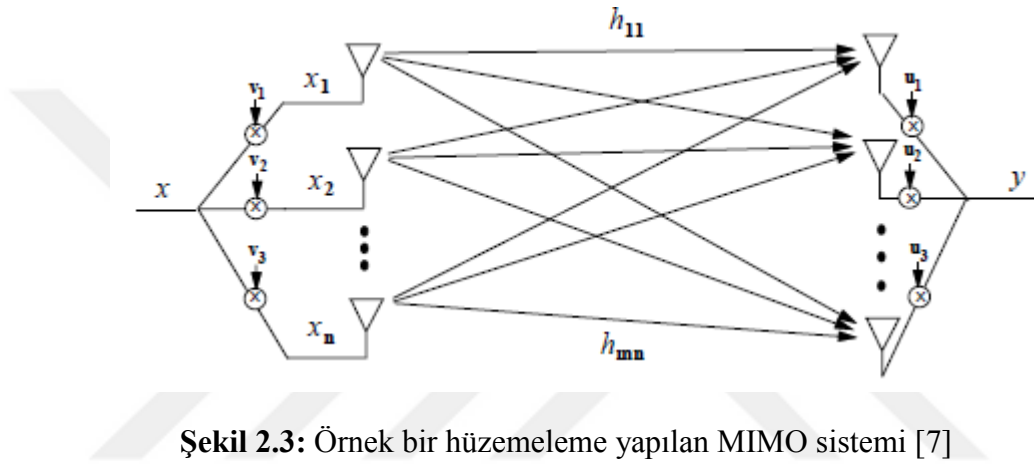


Şekil 2.2: Örnek bir MIMO sistemi [7]

MIMO sistemlerde uzamsal çoklama kapasiteyi artırmaktadır, iletim çeşitlemesi ise bit hata oranını azaltmaktadır. Bu iki yöntem arasındaki fark, sistemdeki farklı yollarda çeşitlemenin aynı, çoklamanın ise birbirinden farklı sembolleri iletmesidir. Bu yöntemler aynı anda tam kullanılamasa da, uzamsal çoklama ve iletim çeşitlemesinin kullanım ağırlıkları arasında bir ayarlama yapılarak, bu iki yöntem aynı anda kullanılıp sistemin kapasite ve hata oranı ihtiyaçları karşılanmaktadır. Sistemde oluşan değişimlerde de yeni duruma göre bu ağırlıklandırma değişebilmektedir [8].

2.3 Hüzmeleme ve Denkleştirme

Hüzmeleme (beamforming) anten sistemlerinde kullanılan ve faz kaydırma ile belirlenen yönde anten kazancı sağlayan bir yöntemdir. Bu yöntemde anten dizisindeki her bir dipol antendeki radyasyon örüntüsü kullanılarak ve bu antenlerin fazları değiştirilerek antenlerin ana loblarının tek aynı doğrultuda gösterilmesi sağlanır. Bu sayede belirlenen yönde bir anten kazancı sağlanırken diğer yönlerde anten kazancı azalır. Anten kazancının yanı sıra semboller arası girişim, kanaldaki düz sönümlenme ve kullanıcılar arası girişim engellenir.

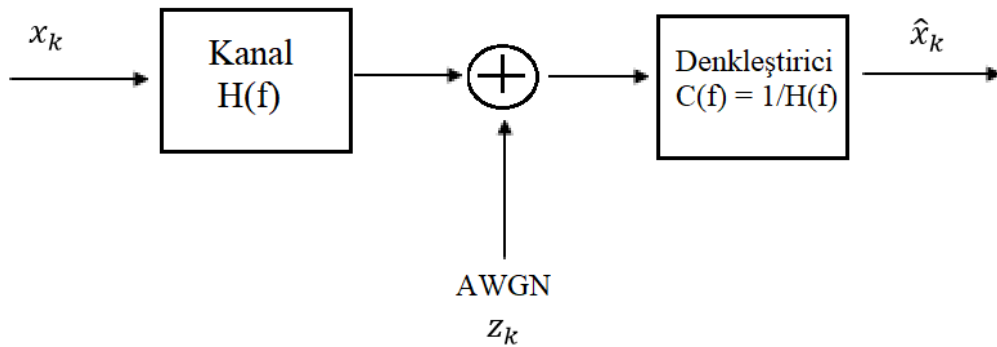


MU-MIMO sistemlerde kanaldaki gecikme dağılımı sebebiyle gönderilen semboller arasında girişim vardır (inter symbol interference, ISI). Semboller arası girişim sistemin performansını oldukça olumsuz etkilemektedir. Bu sorunu çözmek için kanal denkleştirme (channel equalization) kullanılmaktadır. Kanal denkleştirmede verici tarafından gönderilen semboller, kanaldan geçtikten sonra alıcıya geldiğinde, kanal denkleştiriciden geçerek kanalın semboller üzerindeki etkisi giderilmeye ve bu sayede semboller arası girişimin engellenmesi amaçlanır. Kanal denkleştirici kanal bilgisi kullanılarak tasarlanır. İdeal bir durumda kanal bilgisi hatasız hesaplanıp, ona göre kanal denkleştirici tasarlanır ancak gerçek hayatta elde edilen kanal bilgisi hatalı olabilir ve kanal zamanla değiştiği için de elde edilen kanalın düzenli olarak güncellenmesi gerekmektedir. Kanal kestirimi ve elde edilen kanalın belirli aralıklarla güncellenmesi önemli bir problemdir ve önemli bir araştırma konusudur. Gürültü artırılması da dikkat edilmesi gereken önemli bir sorundur. Alıcıdaki kanal denkleştiriciye gönderilen sembollerin yanı sıra kanaldan gelen gürültü de

gelmektedir. Kanal denkleştiriciden geçtikten sonra sembollerin yanı sıra gürültünün de gücü artmaktadır. Gürültünün gücünün artması sistemin performansını olumsuz etkilemektedir [7].

Denkleştiriciler doğrusal ve doğrusal olmayan olarak ikiye ayrılır. Doğrusal denkleştiriciler, doğrusal olmayan denkleştiricilere göre daha basittir ve uygulaması kolaydır ancak doğrusal olmayan denkleştiricilere göre daha kötü performans göstermektedirler. Doğrusal denkleştiricilerdeki en büyük sorun gürültünün gücünün artırılması sorunudur. En çok bilinen doğrusal denkleştiriciler sıfıra zorlayıcı(zero forcing, ZF) ve minimum ortalama karesel hata denkleştiricisidir (minimum mean square error equalizer, MMSE). Doğrusal olmayan denkleştiricide en çok bilinen iki yöntem ise karar geri bildirim denkleştiricisi (decision-feedback equalization, DFE) ve maksimum olabilirlik dizi belirlemesidir(maximum likelihood sequence estimation, MLSE) [7].

Sıfıra zorlayıcı(ZF), doğrusal denkleştiriciler arasında karmaşıklığı en düşük olan yöntemdir. Bu yöntemde gönderilen semboller, kanalın fonksiyonunun tersi alınarak alıcıda denkleştiriciden geçirilir. Kanalın tersinin alınmasının nedeni ise kanalın semboller üzerindeki etkisini yok etmektir. Bu sayede semboller arası girişimin engellenmesi amaçlanmaktadır ama bu yöntem aynı zamanda gürültünün gücünün artmasına da neden olmaktadır.



Şekil 2.4: Örnek bir ZF Denkleştiricisi

Sıfıra zorlayıcı gürültü artırımına neden olmaktadır ve bu yüzden semboller arası girişimin azaltılması ve gürültü artırımını arasında optimizasyon yapabilecek doğrusal bir denkleştiriciye ihtiyaç duyulmaktadır. Minimum ortalama karesel hata denkleştiricisi(MMSE) bu ihtiyacı karşılamak için tasarlanmıştır. MMSE’de amaç denkleştiriciye giren sembol ile çıkan sembol arasında ortalama karesel hatayı en az yapmaktır. Bunu yaparken sembollerde ağırlıklandırma yapılır. Ağırlıkları belirlerken de doğrusal kestirim yöntemlerinden yararlanır.

Maksimum olabilirlik dizi belirlemesi(MLSE), doğrusal denkleştiricilerin aksine gürültü artırımına neden olmadığı için tercih edilmektedir. Burada denkleştirme filtresi kullanılmamaktadır. Onun yerine gönderilen sembollerin dizisi belirlendiği için gürültü artırımını olmamaktadır. MLSE’nin çıktısı sadece örnekleyici çıktısı ve kanal parametrelerine bağlıdır [7].

Karar geri bildirim denkleştiricisinde (DFE) semboller denkleştiriciden geçerken her bir sembol denkleştiriciden geçtikten sonra sembol üzerindeki girişimin etkisi belirlenip, geri bildirim döngüsü kullanılarak elde edilen bilgiye göre denkleştirici güncellenir. Denkleştiriciden çıkan semboldeki girişim giren sembollerden çıkarılarak semboller arası girişimin azaltılması amaçlanır [7].

Denkleştiriciler kanaldaki sönmülemenden dolayı ortaya çıkan semboller arası girişimi engellemek için etkili bir yöntemdir. Denkleştirici kanaldan gönderilen sembollerin alıcıda filtreden geçirilmesiyle mümkün olmaktadır. Günümüzde kablosuz haberleşme ağlarında, özellikler MU-MIMO sistemlerde alıcı olarak kullanılan cep telefonu, akıllı saat, akıllı bileklik gibi aletler vericiye göre çok daha az yer kaplamaktadır. Bu yer sorunundan dolayı alıcının tasarımını daha basit tutmak çok önemlidir. Bu yüzden alıcıda denkleştiricinin yaptığı işi, vericide semboller daha kanaldan gönderilmeden, kanalın neden olacağı semboller arası bozulma önceden tahmin edilerek, semboller bu etkiyi azaltacak şekilde filtreden geçirilir. Vericideki bu filtrelemeye önkodlama(precoding) denir. Önkodlamada önceden gönderilen eğitim sembolleri kullanılarak kanal bilgisi elde edilir ve vericide elde edilen kanal bilgisine gönderilecek sembollere filtreleme yapılır. Önkodlama sayesinde alıcıların tasarımı basitleşir ve her alıcıda bir tane denkleştirici yerine vericide bir önkodlayıcı kullanılarak daha ucuz bir sistem oluşturulur. Bu avantajları sayesinde önkodlama

MU-MIMO sistemler için çok önemli bir araştırma konusudur. Bu çalışmada MU-MIMO sistemdeki çeşitli durumlar için bilinen önkodlama yöntemlerinin karşılaştırması yapılacaktır.

2.4 Benzer Çalışmalar

MU-MIMO sistemlerde birden fazla antenden aynı anda birden fazla kullanıcıya iletim yapılması sayesinde uzamsal çoklama kullanılarak SISO sistemlere göre çok daha yüksek kapasiteye ulaşılabilmektedir [3]. Aynı anda birden fazla iletim yapılması sistemde kullanıcılar arası girişime neden olmaktadır [4]. Günümüzde, günlük hayatın çeşitli alanlarında kullanılan haberleşme sistemlerinde girişim sistemin başarımını azaltan en önemli etkenlerden biridir. Bu yüzden girişimin sistemdeki kullanıcı hızlarının toplamını artıracak şekilde düzenlenmesi gerekmektedir. Baz istasyonundaki iletim tekniklerinin sistemin toplam hızını yükseltecek şekilde tasarlanması çok önemlidir. MU-MIMO sistemleri baz istasyonundaki anten sayısı ve toplam kullanıcı sayısına göre farklı özelliklere sahiptir. Bu sistemler az yüklü, kritik yüklü ve aşırı yüklü olabilmektedir. Eğer kullanıcılardaki toplam anten sayısı, baz istasyonundaki anten sayısından daha az ise sistem az yüklü, baz istasyonundaki anten sayısına eşit ise kritik yüklü ve baz istasyonundaki anten sayısından daha fazla ise aşırı yüklüdür. Özellikle kritik yüklü ve aşırı yüklü sistemler kullanıcılar arası girişimden oldukça olumsuz etkilenmektedir ve bu çalışmada MU-MIMO sistemleri incelenirken kritik yüklü ve aşırı yüklü sistemler kullanılacaktır.

Kullanıcılar arası girişimi azaltmak ve kullanıcıların toplam veri hızını artırmak için baz istasyonunda kullanılan tasarım tekniklerinden birisi önkodlamadır. Önkodlama sayesinde vericideki ve alıcıdaki kanal bilgileri kullanılarak, kanaldan gönderilecek semboller düzenlenir. Bu düzenleme sırasında amaç, veri hızını düşüren kullanıcılar arası girişimi minimum düzeye indirmektir. MU-MIMO sistemlerde kullanıcı hızını en yüksek bulan önkodlama yöntemi kirli kağıt kodlamasıdır (dirty paper coding, DPC) [9]. DPC yöntemi kanalda oluşan girişimin verici tarafından bilindiği durumlarda oluşacak girişimi tamamen yok edecek şekilde önkodlama yapılmasını sağlar. Bu sayede kanalda oluşan girişime rağmen sistem toplam hızda kapasite değerine ulaşabilecektir [10]. Kanalin kapasitesinde ulaşmasına rağmen DPC pratikte pek tercih edilmemektedir çünkü doğrusal olmayan bir önkodlama yöntemidir ve bu

yüzden karmaşıklığı yüksektir ve oldukça masraflıdır. Pratikte kullanmak için DPC yerine, DPC'ye yakın sonuçlar veren daha düşük karmaşıklığa sahip doğrusal önkodlama yöntemlerinin tercih edilmesi gerekmektedir.

Aynı zamanda bir kanal denkleştirme yöntemi olan sıfıra zorlayıcı (zero-forcing, ZF), bilinen en temel doğrusal önkodlama yöntemidir [11]. Bu yöntemde belli kullanıcılara ağırlık vermek yerine alıcıların hiçbirisinde girişim olmamasını amaçlamaktadır. Sadece kanal matrisinin tersi alınarak oluşturulan ZF'in yanı sıra ZF kullanılarak oluşturulan daha karmaşık yöntemler de vardır. Sıfıra zorlayıcı hüznleme (zero-forcing beamforming, ZFBF) ve yarı dik, kullanıcı seçimli, sıfıra zorlayıcı önkodlama (semi-orthogonal user selection zero forcing precoding, SUS-ZF) yöntemleri bunlara örnektir [12]. [12]'deki ZFBF yöntemi kanal matrisinin tersini alarak bir önkodlama matrisi oluşturmayı amaçlamaktadır ama bunu yaparken ZF'ten farklı olarak kullanıcılara SNR değerlerine göre farklı güç paylaşırması yapmaktadır. Bu güç paylaşırması yapılırken su doldurma algoritması kullanılmıştır. [12]'deki başka bir yöntem olan SUS-ZF ise özellikle aşırı yüklü sistemlerde kullanılmaktadır. Bu yöntemde kullanıcılardaki kanal vektörlerinin birbirlerine olan diklikleri hesaplanmaktadır. Belli bir eşiğin üzerindeki bu diklikler kullanılarak kullanıcı seçimi yapılmaktadır. Bu iki yöntem birleştirilerek SUS-ZFBF yöntemi kullanılarak, önce kullanıcı seçimi yapılmış daha sonra su doldurma algoritması kullanılarak sıfıra zorlayıcı yapılmıştır. Elde edilen sonuç çok yüksek sayıdaki kullanıcılarda başarılı sonuç vermiştir. Toplam hız kullanıcı sayısının sonsuza gittiği durumlarda DPC ile aynı asimptotik toplam kapasiteye ulaşmaktadır [12].

Minimum ortalama karesel hata (minimum mean square error, MMSE) önkodlama yöntemi matematikteki ortalama karesel hata hesaplama yönteminden yola çıkılarak düşük karmaşıklıkta önkodlama yapılmasını sağlar. Bu yöntem MU-MIMO sistemde kullanıcı hızlarının toplamını bulmak için oluşturulan ve aynı zamanda bir optimizasyon problemi olan ağırlıklı kullanıcı hızları toplamının (weighted sum rate, WSR) maksimum olduğu durum ile ağırlıklı minimum ortalama karesel hata (weighted minimum mean square error, WMMSE) arasında bir denklik bulup, o denklikten yararlanmaktadır. Bu denklik sayesinde karmaşıklığı daha yüksek WSR probleminin maksimum olduğu durum yerine, karmaşıklığı daha düşük WMMSE probleminin minimum olduğu yer hesaplanır [2]. MMSE yöntemi ZF aksine kullanıcılardaki

girişimin sıfır olması şartını kaldırarak her kullanıcı için daha yüksek sinyal-girişim-gürültü oranı (signal to interference noise ratio, SINR) elde edilmesini sağlamaktadır. [2]'de değişen eniyileme kullanılmaktadır. Bu yöntemde MMSE optimizasyon problemini çözücü ile çözmek yerine iteratif olarak güncelleyerek bir noktada yakınsamasını sağlamaktadır. Her adımda tek bir optimal noktayı aramak yerine önce vericiyi güncelleyip, daha sonra vericideki bilgileri kullanarak alıcıyı güncellemektedir. Bu sayede bu yöntemin karmaşıklığı daha düşük olup, verici ve alıcının ayrı ayrı tasarlanmasına izin vermektedir.

Kanal bilgisinin verici tarafından hatalı olarak bilindiği durumlarda, eğer hata dağılımı biliniyorsa, bu dağılımdan yararlanılarak çoklama kazancı elde edilebilir. Bu sistemlerde önkodlamada hata bilgisinden yararlanılarak serbestlik derecesinde(degree of freedom) daha yüksek bir üst sınır elde edilebilmektedir [13]. Hız bölütleme(Rate Splitting) yöntemi hatalı bilinen kanallardaki serbestlik derecesini kullanarak önkodlama yapmayı ve daha iyi sonuçlar elde etmeyi amaçlamaktadır. Hız Bölütleme Çoklu Erişim(Rate Splitting Multiple Access, RSMA) Uzay Bölmeli Çoklu Erişim(Space Division Multiple Access, SDMA) ve Dik-Olmayan Çoklu Erişim(Non-Orthogonal Multiple Access, NOMA) ile kıyaslandığında daha iyi sonuçlar verdiği gözlemlenmektedir [14]. Hız bölütleme yönteminde kullanıcılara gönderilen mesajlarda semboller iki türe ayrılmaktadır: Ortak sembol ve özel sembol. Tüm kullanıcılarda bulunan sembole ortak sembol denir. Özel sembol ise her kullanıcıya farklı olarak gönderilen semboldür. Burada önkodlama ortak sembol ve her kullanıcıdaki özel semboller için yapılmaktadır. Öncelikle ortak semboller için önkodlama yapılır, burada sadece ortak semboller dikkate alınıp, diğer semboller kanal gürültüsüyümüş gibi davranılır. Daha sonra bu ortak semboller alıcıya gelen toplam sembollerden çıkarılır. Elde edilen sonuçlar için tekrar önkodlama yapılır. Hız Bölütleme yöntemi içindeki bu işleme Ardışık Girişim Giderimi(Successive Interference Cancellation, SIC) denir [15].

RS yönteminde K kullanıcı olduğu durumda, mesajlar $2^K - 1$ gruba kadar bölünebilir. Daha sonra kullanıcılarda bulunan toplam $K2^K - 1$ mesajlar, toplam $2^K - 1$ olacak şekilde birleştirilebilir. Bu işlem sadece ortak ve özel sembol olarak iki grup kullanılan RS işleminden daha iyi ve kanal kapasitesine yakın sonuç vermektedir ancak bu yöntem SIC işleminin birden fazla kullanılmasını gerektirdiği için karmaşıklığı daha

yüksektir [16]. RS yapılırken karmaşıklığın düşük olmasının amaçlandığı durumlarda SIC'in bir kere kullanıldığı yöntemler kullanılmalıdır. Ardışık Sıfır Uzay(Successive Null Space, SNS) yöntemi, hem RS yönteminden yararlanır, hem de kullanıcıları sırayla öncekilerin boş uzaylarına yerleştirir. Bu yöntemde MIMO kanal matrislerinin, sıfır uzay temel vektörlerinin doğrusal birleşimlerini kullanarak kullanıcılar arası girişimi azaltacak şekilde önkodlama yapılır. Bu yöntemde RS kullanılırken SIC işleminin sadece bir kez kullanılması bu yöntemin daha düşük karmaşıklığa sahip olmasını sağlamaktadır. Bu yöntemin az ve kritik yüklü sistemlerde yüksek başarımla elde ettiği gözlemlenmiştir [17].

Sıfıra Zorlayıcı Kerte Uyarlama (Rank Adaptation with ZF), yönteminde kullanıcı hızlarının toplamını en fazla yapacak şekilde kullanıcı seçimi yapmaktadır. İlk adımda kullanıcı hızı kapasitesi en yüksek olan kullanıcıyı seçmektedir. İkinci adımda yine kullanıcı seçimi yapılarak ilk seçilen kullanıcı ile birlikte hesaplandığında kullanıcı hızlarının toplamını en yüksek yapan kullanıcı bulunur. Kullanıcı seçimi, tüm kullanıcılar seçilene ya da hesaplanan toplam kullanıcı hızı, bir önceki adımdan daha düşük çıkana kadar devam eder. Bu yöntem açgözlü bir şekilde kullanıcı seçimi yapmaktadır ve her adımda ZF işlemi için kanal matrisinin tersini almaktadır. Kanal matrisinin tersinin hesaplanması maliyetli bir işlemdir ve bu yöntemin karmaşıklığını artırmaktadır. Bu yöntemin GPIP ve SUS-ZF'e göre karmaşıklığı daha yüksektir. Burada açgözlü kullanıcı seçimi yapıldığı için sadece kullanıcıların bir kısmına iletim yapılmaktadır. İletilen kullanıcı sayısı SNR ile doğru orantılı olarak artmaktadır [4].

3. SİSTEM MODELİ

Bu çalışmada bir MU-MIMO aşağı gönderim sistemi incelenmektedir. Bu sistemde N antene sahip bir baz istasyonu ve K tane tek antene sahip kullanıcı vardır. Bu sistemdeki kanallar, birbirinden bağımsız ve eş dağılıma sahip Rayleigh sönümlü kanallardır. Sistemdeki k . kullanıcı için, kanal vektörü $h_k \in \mathbb{C}^{N \times 1}$, $k \in \{1, \dots, K\}$ ile gösterilmektedir. Tüm kullanıcılar için oluşturulan ortak kanal $H = [h_1 h_2 \dots h_K] \in \mathbb{C}^{N \times K}$ ile gösterilmiştir. Baz istasyonundaki toplam güç ise P_t ile ifade edilmiştir. Sistemde sıfır ortalama ve σ^2 varyansa sahip, dairesel simetrik, karmaşık Gaussian gürültüsü vardır.

3.1 Önkodlayıcı Özellikleri

Baz istasyonunda doğrusal önkodlayıcı kullanılmaktadır. k . kullanıcı için kullanılan önkodlama vektörü $f_k \in \mathbb{C}^{N \times 1}$, $k \in \{1, \dots, K\}$ ile gösterilmiştir. Tüm kullanıcılar için ortak önkodlama matrisi ise $F = [f_1 f_2 \dots f_K] \in \mathbb{C}^{N \times K}$ ile gösterilmiştir. F önkodlayıcısı GPIP, MMSE ve SUS-ZF yöntemlerinde kullanılmıştır. Sadece RS yönteminde $F = [f_c, f_1, \dots, f_K] \in \mathbb{C}^{N \times (K+1)}$ olarak tanımlanmıştır. Önkodlama matrisi birim güce sahiptir ve $\sum_{k=1}^K \|f_k\|_2^2 = 1$ şartını sağlaması gerekmektedir.

3.2 Kullanıcı Hızlarının Toplamı

Bu sistemde baz istasyonundan çıkan $x = \{x_1, x_2 \dots x_K\} \in \mathbb{C}^{N \times K}$ sembolleri birbirinden bağımsız K tane kullanıcıya gönderilmektedir. Bu gönderim sırasında doğrusal önkodlama matrisi $F = [f_1 f_2 \dots f_K]$ ile gösterilmektedir. Tüm bilgi sembolleri sıfır ortalama ve P_t varyansa sahip Gaussian sembolleridir. P_t burada ileticinin toplam gücünü temsil etmektedir. Bu çalışmada hatalı kanal bilgisi de incelendiğinden sistemde kullanıcılar arası girişim vardır. Baz istasyonundan gönderilen semboller hesaplanırken kullanıcıların önkodlayıcıları ve farklı kullanıcılara gönderilecek semboller kullanılmıştır. k . kullanıcıda elde edilen sinyal aşağıdaki şekildedir:

$$y_k = h_k^H f_k x_k + \sum_{i \neq k}^K h_k^H f_i x_i + z_k \quad (1)$$

Bu formülde toplam sembolü içindeki değerler kullanıcılar arası girişim olarak ifade edilmektedir. z_k her kullanıcıda hissedilen Gaussian kanal gürültüsüdür, sıfır ortalamaya ve σ^2 varyansa sahiptir. Kullanıcıların kanal durumu ve önkodlama vektörleri bilindiğinde k. kullanıcı için kullanıcı hızının hesaplanması aşağıda gösterilmiştir.

$$SINR_k = \frac{|h_k^H f_k|^2}{\sum_{i \neq k}^K |h_k^H f_i|^2 + \frac{\sigma^2}{P_t}} \quad (2)$$

$$\bar{R}_k = E[\log_2(1 + SINR_k)] \quad (3)$$

3.3 Kanal Bilgisi

Alıcı ve vericideki kanal bilgisi iki ayrı durum için incelenmiştir. İlk durumda kanal bilgisinin hem alıcı hem de vericide hatasız bilindiği varsayılmıştır. İkinci durumda ise vericideki kanal bilgisinin hatalı olduğu durumlar incelenmiştir. Bu durumlarda kanal hatası, gerçek kanal bilgisi ile aynı dağılıma sahiptir. Kanal bilgisinde kanal hatasının gücü $O(P_t^{-\alpha})$ olarak değişmektedir. Burada α kanal bilgisi kalite değeridir. $\alpha \in [0, 1]$ arasında sabit bir değer olup, bu değer arttıkça elde edilen serbestlik derecesi artmaktadır [18]. k. kullanıcı için tahmin edilen kanal hatası \tilde{h}_k ile gösterilmiştir. Bu hataya göre elde edilen k. kullanıcı için elde edilen hatalı kanal matrisi ise $\hat{h}_k = h_k - \tilde{h}_k$ ile gösterilmiştir. Kullanıcılar için hatalı kanal matrisi $\hat{H} = [\hat{h}_1 \hat{h}_2 \dots \hat{h}_K] \in \mathbb{C}^{N \times K}$ olarak gösterilmiştir. Burada \tilde{h}_k 'nin gücünün h_k 'ya gücünün oranı $\sqrt{P_t^{-\alpha}}$, dır.

$$y_k = \hat{h}_k^H f_k x_k + \sum_{i \neq k}^K \hat{h}_k^H f_i x_i + \tilde{h}_k^H \sum_{i=1}^K f_i x_i + z_k \quad (4)$$

4. GPIIP

GPIIP yönteminin diğer önkodlama yöntemlerine göre en önemli avantajı MU-MIMO sistemlerde aynı anda önkodlama, kullanıcı seçimi ve güç aktarımı yapmasıdır. Bu yöntemde kullanıcı hızlarını en yüksek yapan amaç fonksiyonu Rayleigh bölümlerinin çarpımı olacak şekilde yazılmıştır. Karush Kuhn Tucker (KKT) koşulundan yola çıkılarak doğrusallığı sağlayacak ve yinelemeli bir şekilde yerel en iyi noktası hesaplanmaktadır [4].

4.1. GPIIP Problem Tanımı

Kanal Kodlama Teoremi'ne(Channel Coding Theorem) göre bir kanalda kapasitenin altındaki tüm hızlar erişilebilir durumdadır. Kapasitenin altındaki tüm bu hızlar için hatanın sifira yakınsadığı kodlama yöntemleri vardır. Gönderim hızının kapasitenin üzerine çıktığı durumlarda hata oranı sıfırdan uzaklaşma başlamaktadır [19]. GPIIP'de önkodlama hesabı yapılırken kanal bilgisini anlık olarak hatasız bilmek mümkün değildir. Kanalda oluşabilecek hatalar kanalın spektral verimliliğinin gerçekte olandan fazla hesaplanmasına neden olabilmektedir. Spektral verimliliğin olduğundan fazla hesaplaması, hatalı veri iletimine neden olabilmektedir. Kanal bilgisi, anlık hesaplanamamasına rağmen kanal hatası dağılımı bilindiğinde, hata dağılımına göre ortalaması alınarak ortalama spektral verimlilik hesaplanabilmektedir. Ortalama spektral verimlilikte hatalı kanal dağılımı $f_{h|\hat{h}}(h_k|\hat{h}_k)$ kullanılarak ergodik spektral verimlilik hesaplanmaktadır.

$$R_k(\hat{H}) = E_{H|\hat{H}}[\log_2(1 + SINR_k)|\hat{H}] \quad (5)$$

$$\bar{R}_k = E_{\hat{H}}[R_k(\hat{H})] \quad (6)$$

Hatalı kanal bilgisinden dolayı oluşan girişim ve gürültü Gaussian değildir ve bu yüzden karşılıklı bilgi $I(\{x_k\};\{y_k\})$ 'yi hesaplamak çok zordur. Genelleştirilmiş Karşılıklı Bilgi(Generalized Mutual Information, GMI) kullanılarak anlık spektral verimlilik için aşağıdaki denklem kullanılır. Bu denklem hesaplanırken kanal hatasından dolayı oluşan girişim kullanılmıştır.

$$f_k^H \Phi_k f_k P_t = E[|\tilde{h}_k^H f_k x_k|^2] \quad (7)$$

$$R_k(\hat{H}) \geq \log_2 \left(1 + \frac{|\hat{h}_k^H f_k|^2}{\sum_{i \neq k} |h_k^H f_i|^2 + \sum_{k=1}^K f_i^H \Phi_k f_i + \frac{\sigma^2}{P_t}} \right) \quad (8)$$

$$\sum_{k=1}^K R_k(\hat{H}) \geq \sum_{k=1}^K \log_2 \left(1 + \frac{|\hat{h}_k^H f_k|^2}{\sum_{i \neq k} |h_k^H f_i|^2 + \sum_{k=1}^K f_i^H \Phi_k f_i + \frac{\sigma^2}{P_t}} \right) \quad (9)$$

$$\sum_{k=1}^K R_k(\hat{H}) \geq \log_2 \left(\prod_{k=1}^K \left[\frac{\sum_{k=1}^K f_i^H (\hat{h}_k \hat{h}_k^H + \Phi_k) f_i + \frac{\sigma^2}{P_t}}{\sum_{i \neq k} f_i^H \hat{h}_k \hat{h}_k^H f_i + \sum_{k=1}^K f_i^H \Phi_k f_i + \frac{\sigma^2}{P_t}} \right] \right) \quad (10)$$

Bu formüller kullanılarak bu sistem kullanıcı hızlarının toplamını en fazla yapacak önkodlayıcıları hesaplayacak bir optimizasyon problemi olarak gösterilir.

$$\arg \max_{f_1 \dots f_K} \prod_{k=1}^K \left[\frac{\sum_{k=1}^K f_i^H (\hat{h}_k \hat{h}_k^H + \Phi_k) f_i + \frac{\sigma^2}{P_t}}{\sum_{i \neq k} f_i^H \hat{h}_k \hat{h}_k^H f_i + \sum_{k=1}^K f_i^H \Phi_k f_i + \frac{\sigma^2}{P_t}} \right] \quad (11)$$

$$\text{öyle ki } \sum_{k=1}^K \|f_k\|_2^2 = 1$$

Bu problem dış bükey değildir ve bu problemin birinci-mertebe optimallik kullanılarak çözülmesini sağlayan düşük karmaşıklığa sahip bir algoritma gerekmektedir.

4.2 GPIP Algoritması

Bu algoritma sayesinde yukarıda tanımlanan problem önkodlama, kullanıcı seçimi ve güç aktarımını aynı anda sağlayacak şekilde çözülebilecektir. Bu bölüm için önkodlama matrisi F algoritmada kullanılmak üzere $\hat{F} = [f_1^T f_2^T \dots f_K^T]^T \in$

$\mathbb{C}^{NK \times 1}$ olacak şekilde yazılmıştır. Oluşturulan yeni önkodlama vektörü kullanılarak $k \in \{1 \dots K\}$ kullanıcıları için $A_k \in \mathbb{C}^{NK \times NK}$, $B_k \in \mathbb{C}^{NK \times NK}$ matrisleri hesaplanabilecektir. Bu matrisler kullanılarak amaç fonksiyonu Rayleigh bölümlerinin çarpımı olarak yazılabilecektir.

$$\sum_{k=1}^K f_i^H (\hat{h}_k \hat{h}_k^H + \Phi_k) f_i + \frac{\sigma^2}{P_t} = \hat{F}^H A_k \hat{F} \quad (12)$$

$$\sum_{i \neq k}^K f_i^H \hat{h}_k \hat{h}_k^H f_i + \sum_{k=1}^K f_i^H \Phi_k f_i + \frac{\sigma^2}{P_t} = \hat{F}^H B_k \hat{F} \quad (13)$$

$$A_k = \begin{bmatrix} \hat{h}_k \hat{h}_k^H + \Phi_k & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \hat{h}_k \hat{h}_k^H + \Phi_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \hat{h}_k \hat{h}_k^H + \Phi_k \end{bmatrix} + \frac{\sigma^2}{P_t} I_{NK} \quad (14)$$

$$B_k = A_k - \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \hat{h}_k \hat{h}_k^H + \Phi_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \quad (15)$$

$$\arg \max_{\hat{F} \in \mathbb{C}^{NK \times 1}} \prod_{k=1}^K \left[\frac{\hat{F}^H A_k \hat{F}}{\hat{F}^H B_k \hat{F}} \right] \quad (16)$$

öyle ki $\|\hat{F}\|_2^2 = 1$

Elde edilen amaç fonksiyonu $\lambda(\hat{F})$ iç bükey ya da dış bükey değildir ve bu yüzden bu fonksiyonu kullanarak elde edilecek en iyi onkodlayıcı çok yüksek karmaşıklığa sahip hesaplamalar gerektirmektedir. Karush-Kuhn Tucker (KKT) koşullarından yararlanarak en iyi sonucu vermeyen ancak yine iyi sonuç veren ve daha düşük karmaşıklığa sahip bir yöntem kullanılmıştır. KKT şartlarına göre $\frac{\partial \lambda(\hat{F})}{\partial \hat{F}^H} = 0$ eşitliğini sağlayan \hat{F} için $\bar{A}(\hat{F})\hat{F} = \lambda(\hat{F})\bar{B}(\hat{F})\hat{F}$ ifadesi doğrudur. Burada F matrisi kullanılarak tüm kullanıcılar için ortak \bar{A} ve \bar{B} matrisleri hesaplanmıştır.

$$\bar{A}(\hat{F}) = \sum_{i=1}^K \prod_{k \neq i}^K (\hat{F}^H A_k \hat{F}) A_i \quad (17)$$

$$\bar{B}(\hat{F}) = \sum_{i=1}^K \prod_{k \neq i}^K (\hat{F}^H B_k \hat{F}) B_i \quad (18)$$

KKT koşulundan yararlanmaktaki amaç yinelemeli bir algoritma tasarlamaktır. Sonucu en iyi yapan önkodlayıcıyı tek adımda hesaplamak yerine elde edilen önkodlayıcıyı, algoritma yakınsayana kadar her adımda güncellemek karmaşıklığı daha düşük bir yöntemdir. Bu algorithmada ilk adımda başlangıç için bir önkodlayıcı seçilir. Verilen bu F önkodlayıcısı için daha sonra her adımda \bar{A} ve \bar{B} matrisleri yeniden hesaplanır. Bu algorithmaya göre m . yinelemedeki önkodlayıcı matrisi bir önceki adımdakinin bir fonksiyonu olarak yazılır. Daha sonra $\|\hat{F}\|_2^2 = 1$ şartını sağlayacak şekilde normleştirme yapılır. Bu adımdan sonra yeni bulunan önkodlayıcı, bir önceki adımda bulunan önkodlayıcı ile karşılaştırılmaktadır ve aralarındaki fark belirli eşik değerinden ε daha küçük olana kadar algoritma önkodlayıcıları güncellemeye devam etmektedir. Algorithmada ilk iterasyon için önkodlayıcı \hat{F} , MRT kullanılarak hesaplanmıştır. Algorithmadaki iterasyon sayısı $[n]$ ile gösterilmiştir. Algoritma tamamlandıktan sonra elde edilen $\hat{F} = [f_1^T f_2^T \dots f_K^T]^T \in \mathbb{C}^{N \times K}$ matrisi $F = [f_1 f_2 \dots f_K] \in \mathbb{C}^{N \times K}$ şeklinde yeniden yazılır. Hesaplanan F matrisi kullanılarak kullanıcıların toplam hızı hesaplanmaktadır.

n = 0, $\hat{F}^{[0]}$ MRT ile hesaplanır	
1. Adım	$\hat{F}^{[n]} = [\bar{B}(\hat{F}^{[n-1]})]^{-1} \bar{A}(\hat{F}^{[n-1]}) \hat{F}^{[n-1]}$
2. Adım	$\hat{F}^{[n]} = \frac{\hat{F}^{[n]}}{\ \hat{F}^{[n]}\ _2}$
3. Adım	$\ \hat{F}^{[n-1]} - \hat{F}^{[n]}\ _2 \leq \varepsilon$

Çizelge 4.1: GPIIP algoritması

5. MMSE

Bu çalışmada kullanılan MU-MIMO sistemlerde kullanıcıların ağırlıklı hızlarının toplamını(Weighted Sum-Rate, WSR) en fazla yapmak amaçlanmaktadır. Kullanıcı hızlarının toplamını en fazla yapmak, bir eniyileme problemi olarak ifade edildiğinde bu problemdeki amaç fonksiyonu dış bükey olmadığı için bu problemin çözümünün karmaşıklığı yüksektir. Bu yüzden karmaşıklığı azaltacak yöntemlerin uygulanması gerekmektedir. Ağırlıklı minimum ortalama karesel hata(weighted minimum mean square error, WMMSE) probleminde, farklı kullanıcı ağırlıkları kullanılarak verici filtrelerinin tasarımı yapılmaktadır. Bu tasarım için gerekli olan fonksiyonlar bir eniyileme problemi olarak yazıldığında ve WSR problemindeki fonksiyonlarla karşılaştırıldığında bu fonksiyonlar arasındaki benzerlikten yararlanılarak WSR problemi WMMSE problemi için gereken fonksiyonlarla çözülebilmektedir. Bu yöntemle MMSE önkodlaması denilmektedir. Bu sayede karmaşıklığı yüksek olan WSR problemi, karmaşıklığı daha düşük olan WMMSE problemi kullanılarak çözülebilmektedir [2].

5.1 Hatasız Kanal için MMSE Algoritması

Kanal bilgisinin hatasız bilindiği durumlarda WMMSE problemi eniyileme problem çözücü olmadan analitik bir şekilde çözülebilmektedir. Bu sayede daha düşük karmaşıklığa ulaşılmaktadır. WMMSE problemini çözmek için gereken algorithmada karmaşıklığı azaltmak için yinelemeli bir şekilde verici filtreleri yani önkodlayıcılar $F = [f_1 f_2 \dots f_K] \in \mathbb{C}^{N \times K}$ hesaplanmaktadır. Her adımda önkodlayıcılar karşılaştırarak sabit bir değere yakınsayana kadar hesaplama işlemi devam etmektedir. Burada genel en iyi değer yerine yerel en iyi değer bulunmuştur. WMMSE için bulunan bu sonuç, WSR probleminde de iyi bir sonuç vermektedir. İlk olarak kanal bilgisi kullanılarak verici filtrelerinin başlangıç değeri hesaplanmaktadır.

$$c = \sqrt{\frac{P_t}{\text{Tr}(HH^H)}} \quad (19)$$

$$f_k^{init} = c h_k^H \quad (20)$$

Bu algoritma her iterasyonda üç adımdan oluşmaktadır. İlk adımda önkodlayıcı kullanılarak alıcı filtresi, ikinci adımda yine önkodlayıcı kullanılarak MSE kullanıcı ağırlıkları hesaplanmaktadır. Üçüncü adımda ise hesaplanan alıcı filtresi ve MSE kullanıcı ağırlıkları kullanılarak önkodlayıcı tekrar hesaplanmaktadır. Önkodlayıcı sabit bir değere yakınsayana kadar bu döngü devam etmektedir. Yakınsama durumunda k. kullanıcının hızı R_k hesaplanmaktadır. Bu adımlarda alıcı filtresini hesaplamak için kullanılan r_{v_k} k. kullanıcıda, diğer kullanıcılardan dolayı oluşan gürültüdür. $D = \text{diag}\{d_1, \dots, d_K\}$ ve $W = \text{diag}\{w_1, \dots, w_K\}$ alıcı filtresi ve MSE kullanıcı ağırlıklarının tüm kullanıcılar için ortak bir şekilde yazılmış blok köşegen gösterimidir. $H = [h_1 h_2 \dots h_K]$ matrisinde ise kanal bilgisi tüm kullanıcılar için ortak bir matriste yazılmıştır.

$$r_{v_k} = 1 + \sum_{i \neq k}^K h_k^H f_k f_k^H h_k \quad (21)$$

$$d_k = f_k^H h_k (h_k^H f_k f_k^H h_k + r_{v_k})^{-1} \quad (22)$$

$$w_k = 1 + f_k^H h_k r_{v_k}^{-1} h_k^H f_k \quad (23)$$

$$\bar{F} = \left(HD^H W D H^H + \frac{\text{Tr}(W D D^H)}{P_t} I_N \right)^{-1} H D^H W \quad (24)$$

$$c = \sqrt{\frac{P_t}{\text{Tr}(\bar{F} \bar{F}^H)}} \quad (25)$$

$$F = c \bar{F} \quad (26)$$

$$R_k = \log_2(w_k) \quad (27)$$

n = 0, (19) ve (20) kullanılarak $F^{[0]}$ hesaplanır	
1.Adım	$f_k^{[n]}$ kullanılarak $d_k^{[n]}$ tüm kullanıcılar için hesaplanır
2.Adım	$f_k^{[n]}$ kullanılarak $w_k^{[n]}$ tüm kullanıcılar için hesaplanır
3.Adım	$D^{[n]}$ ve $W^{[n]}$ kullanılarak $C^{[n]}$ hesaplanır
4.Adım	$\ F^{[n-1]} - F^{[n]}\ _2 \leq \varepsilon$

Çizelge 5.1: MMSE algoritması

5.2 Hatalı Kanal için MMSE ve RS Algoritması

RS yönteminde kullanıcılara gönderilen semboller gruplara ayrılarak ve oluşturulan gruplar için yapılan önkodlamalarla toplam kullanıcı hızının artırılması amaçlanmıştır. Burada kullanılan semboller, ortak ve özel olmak üzere iki gruba ayrılmıştır. Ortak semboller tüm kullanıcılarda aynıdır ve özel semboller her kullanıcı için farklılık göstermektedir [15]. Gönderilen semboller $s = [s_c, s_1, \dots, s_K]^T \in \mathbb{C}^{K+1}$ şeklinde, önkodlayıcılar ise $F = [f_c, f_1, \dots, f_K] \in \mathbb{C}^{N \times K+1}$ olarak belirtilmiştir. Baz istasyonundan gönderilen sembollerin toplamı şu şekildedir:

$$x = Fs = f_c s_c + \sum_{i=1}^K f_i s_i \quad (28)$$

Kanal bilgisi ve önkodlayıcı kullanılarak k. kullanıcı için ortak sembollerin gücü $S_{c,k}$, özel sembollerin gücü S_k , ortak semboller dışındaki girişim $I_{c,k}$, özel semboller dışındaki girişim I_k , ortak sembollerden dolayı oluşan toplam güç $T_{c,k}$ ve özel sembollerden dolayı oluşan toplam güç T_k hesaplanır.

$$S_{c,k} = |\hat{h}_k^H f_c|^2 \quad (29)$$

$$S_k = |\hat{h}_k^H f_k|^2 \quad (30)$$

$$I_{c,k} = |\hat{h}_k^H f_k|^2 + \sum_{i \neq k}^K |\hat{h}_k^H f_i|^2 + \sigma^2 \quad (31)$$

$$I_k = \sum_{i \neq k}^K |\hat{h}_k^H f_i|^2 + \sigma^2 \quad (32)$$

$$T_{c,k} = S_{c,k} + I_{c,k} \quad (33)$$

$$T_k = S_k + I_k \quad (34)$$

Elde edilen $S_{c,k}$, S_k , $I_{c,k}$, I_k , $T_{c,k}$, T_k kullanılarak k. kullanıcı için ortak ve özel MSE matrisleri $g_{c,k}^{MMSE}$, g_k^{MMSE} , MMSE denkleştirici $\varepsilon_{c,k}^{MMSE}$, ε_k^{MMSE} ve MSE matrisinin ağırlıkları $u_{c,k}$, u_k hesaplanmıştır.

$$\varepsilon_{c,k}^{MMSE} = \frac{I_{c,k}}{T_{c,k}} \quad (35)$$

$$\varepsilon_k^{MMSE} = \frac{I_k}{T_k} \quad (36)$$

$$g_{c,k}^{MMSE} = \frac{f_c^H h_k}{T_{c,k}} \quad (37)$$

$$g_k^{MMSE} = \frac{f_k^H h_k}{T_k} \quad (38)$$

$$u_{c,k} = \frac{1}{\varepsilon_{c,k}^{MMSE}} \quad (39)$$

$$u_k = \frac{1}{\varepsilon_k^{MMSE}} \quad (40)$$

Kanal bilgisinin hatalı olduğu bilindiği durumlarda, hatanın sadece dağılımı bilindiği için rastgele oluşturulan bir kanala, rastgele eklenen bir gürültü ile hatalı bilinen bir kanal oluşturulmaktadır. Gürültü rastgele oluşturulduğu için aynı kanal için her seferinde birbirinden çok farklı sonuçlar elde edilmektedir ve bu da farklı önkodlama yöntemlerinin karşılaştırılmasını zorlaştırmaktadır. Aynı kanal için birden fazla kanal hatası oluşturulduğunda, birden fazla hatalı kanallar oluşturulup bunların sonuçlarının ortalaması alındığında, aynı kanal için sabit bir sonuca yakınsamaktadır.

Büyük sayılar kanununa göre dağılımı bilinen bir rastgele değişkende birbirinden bağımsız denemeler yapıldığında elde edilen örneklerin dağılımı, deneme sayısı arttıkça rastgele değişkenin gerçek dağılımına yakınsamaktadır [20]. Büyük sayılar

kanunundan yararlanılarak oluşturulan Örnek Ortalama Yaklaşım(Sample Average Approximation, SAA) yöntemi kullanılarak elde edilen sonuçların ortalaması alınarak, teorik ortalamaya yakın bir değer elde edilir [21]. Bu yüzden hatalı kanal bilgisi kullanıldığında M tane kanal oluşturulup, bu kanallar kullanılarak elde edilen parametrelerin ortalaması alınmaktadır. RS ve MMSE algoritmaları ortalaması alınan parametrelere göre çözüldükleri için daha stabil sonuç vermektedir. Bu algortmada ortalama alınarak $\bar{t}_{c,k}, \bar{t}_k, \bar{\Psi}_{c,k}, \bar{\Psi}_k, \bar{\delta}_{c,k}, \bar{\delta}_k, \bar{v}_{c,k}, \bar{v}_k$ hesaplanmaktadır. Hesaplama sırasında parametrelerin üzerindeki (m) işareti, M tane ölçüm arasından yapılan m. ölçümü ifade etmektedir.

$$t_{c,k}^{(m)} = u_{c,k}^{(m)} \left| g_{c,k}^{(m)} \right|^2 \quad (41)$$

$$t_k^{(m)} = u_k^{(m)} \left| g_k^{(m)} \right|^2 \quad (42)$$

$$\Psi_{c,k}^{(m)} = t_{c,k}^{(m)} \hat{h}_k^{(m)} \hat{h}_k^{(m)H} \quad (43)$$

$$\Psi_k^{(m)} = t_k^{(m)} \hat{h}_k^{(m)} \hat{h}_k^{(m)H} \quad (44)$$

$$\delta_{c,k}^{(m)} = u_{c,k}^{(m)} \hat{h}_k^{(m)} g_{c,k}^{(m)} \quad (45)$$

$$\delta_k^{(m)} = u_k^{(m)} \hat{h}_k^{(m)} g_k^{(m)} \quad (46)$$

$$v_{c,k}^{(m)} = \log_2(u_{c,k}^{(m)}) \quad (47)$$

$$v_k^{(m)} = \log_2(u_k^{(m)}) \quad (48)$$

$$\bar{t}_{c,k} = \frac{1}{M} \sum_{m=1}^M t_{c,k}^{(m)} \quad (49)$$

$$\bar{t}_k = \frac{1}{M} \sum_{m=1}^M t_k^{(m)} \quad (50)$$

$$\bar{\Psi}_{c,k} = \frac{1}{M} \sum_{m=1}^M \Psi_{c,k}^{(m)} \quad (51)$$

$$\bar{\Psi}_k = \frac{1}{M} \sum_{m=1}^M \Psi_k^{(m)} \quad (52)$$

$$\bar{\delta}_{c,k} = \frac{1}{M} \sum_{m=1}^M \delta_{c,k}^{(m)} \quad (53)$$

$$\bar{\delta}_k = \frac{1}{M} \sum_{m=1}^M \delta_k^{(m)} \quad (54)$$

$$\bar{v}_{c,k} = \frac{1}{M} \sum_{m=1}^M v_{c,k}^{(m)} \quad (55)$$

$$\bar{v}_k = \frac{1}{M} \sum_{m=1}^M v_k^{(m)} \quad (56)$$

$A_{RS}^{[n]}(F)$ fonksiyonu kullanıcı hızları toplamının ortalamasını vermektedir. Burada ortalama hesabı aynı kanalın farklı ölçümleri için yapılmıştır. Tanımlanan amaç fonksiyonuna göre kullanıcı hızları toplamının ortalamasını en yüksek yapacak önkodlama matrisi hesaplanmaktadır. Burada eniyileme parametreleri önkodlama vektörü F , ortak semboller için artırılmış WMSE'ler ise $\bar{\zeta}_c$ ile gösterilmiştir. Bu problemin çözümü için MATLAB'da kullanılan bir problem çözücü olan cvx problem çözücü kullanılmıştır.

$$A_{RS}^{[n]}(F) = \begin{cases} \min_{\bar{\zeta}_c, F} \sum_{k=1}^K \left(\sum_{i=1}^K f_i^H \bar{\Psi}_k f_i + \sigma^2 \bar{t}_k - 2\Re\{\bar{\delta}_k^H f_k\} + \bar{u}_k - \bar{v}_k \right) \\ \text{öyle ki } f_c^H \bar{\Psi}_{c,k} f_c + \sum_{k=1}^K p_k^H \bar{\Psi}_{c,k} f_i + \sigma^2 \bar{t}_{c,k} - 2\Re\{\bar{\delta}_{c,k}^H f_c\} + \bar{u}_{c,k} \leq \bar{\zeta}_c \\ \|f_c\|^2 + \sum_{k=1}^K \|f_k\|^2 \leq P_t \end{cases} \quad (57)$$

Elde edilen önkodlama matrisine göre ortak ve özel semboller için SINR değerleri $\gamma_{c,k}$ ve γ_k hesaplanmaktadır. SINR değerleri kullanılarak ortak ve özel kullanıcı hızları $R_{c,k}$ ve R_k hesaplanmaktadır.

$$\gamma_{c,k} = \frac{S_{c,k}}{I_{c,k}} \quad (58)$$

$$\gamma_k = \frac{S_k}{I_k} \quad (59)$$

$$R_{c,k} = \log_2 (1 + \gamma_{c,k}) \quad (60)$$

$$R_k = \log_2 (1 + \gamma_k) \quad (61)$$

$$A_{RS} = \min(R_{c,k}) + \sum_{k=1}^K R_k \quad (62)$$

Eniyileme problemi sonucunda elde edilen önkodlama matrisi kullanılarak kullanıcı hızlarının toplamı hesaplanmaktadır. Elde edilen sonuç bir önceki iterasyondaki sonuçla karşılaştırılmaktadır. Kullanıcı hızlarının toplamı sabit bir değere yakınsayana kadar algoritma devam etmektedir.

n = 0 için MRC-SVD kullanılarak $f_c^{[0]}$ ve $f_k^{[0]}$ hesaplanır	
1.Adım	$f_c^{[n-1]}, f_k^{[n-1]}, \hat{h}_k$ kullanarak parametreleri hesapla
2.Adım	$A_{RS}^{[n]}$ kullanarak $F^{[n]}$ hesapla
3.Adım	$\ A_{RS}^{[n]} - A_{RS}^{[n-1]}\ _2 \leq \varepsilon$

Çizelge 5.2: RS algoritması

RS'de kullanıcı hızlarının toplamını hesaplamak için kullanılan bu yöntemde bazı değişiklikler yapılarak hatalı kanal bilgisi için MMSE hesaplanabilir. MMSE'de RS aksine sembol gruplama olmadığı için ortak sembollerle ilgili tüm değişkenler olan $f_c, \bar{t}_{c,k}, \bar{\Psi}_{c,k}, \bar{\delta}_{c,k}, \bar{v}_{c,k}$ değişkenleri sıfıra eşitlenmektedir. Bu sayede amaç fonksiyonu ve kısıt eşitsizlikleri değişmektedir. Elde edilen yeni önkodlayıcıya göre kullanıcıların toplam hızı tekrar hesaplanmaktadır. RS'de olduğu gibi burada da cvx problem çözücü kullanılmıştır.

$$A_{NoRS}^{[n]}(F) = \begin{cases} \min_F \sum_{k=1}^K \left(\sum_{i=1}^K f_i^H \bar{\Psi}_k f_i + \sigma^2 \bar{t}_k - 2\Re\{\bar{\delta}_k^H f_k\} + \bar{u}_k - \bar{v}_k \right) \\ \text{öyle ki } \sum_{k=1}^K \|f_k\|^2 \leq P_t \end{cases} \quad (63)$$

$$A_{NoRS} = \sum_{k=1}^K R_k \quad (64)$$

6. SUS-ZF

Yarı dik, kullanıcı seçimli, sifira zorlayıcı önkodlama (semi-orthogonal user selection zero forcing precoding, SUS-ZF) yöntemi kullanıcı sayısının baz istasyonundaki anten sayısına oranla çok fazla olduğu durumlarda kullanılmaktadır. Bu önkodlama yönteminde öncelikle kullanıcılar arası diklikten yararlanıp kullanıcı seçimi yapılmakta ve daha sonra ZF yöntemi kullanılarak önkodlayıcı tasarlanmıştır. Önkodlayıcı tasarımı sırasında kullanıcıları güç dağıtımını yapılırken su doldurma algoritmasından yararlanılmaktadır [12]. Bu yöntemde kullanıcı seçimi yapıldığı için daha az kullanıcıya iletim yapılmaktadır ve bu yüzden kullanıcılar arası toplam girişim tüm kullanıcıların kullanıldığı durumlara göre daha azdır. Toplam girişimin azlığı toplam kullanıcı hızının daha fazla olmasını sağlamaktadır. Daha az kullanıcı olması, sistemin karmaşıklığının çok daha az olmasını sağlamaktadır.

6.1 SUS-ZF Algoritması

Burada kanal matrisi $H = [h_1 h_2 \dots h_K] \in \mathbb{C}^{N \times K}$ önkodlama matrisi $F = [f_1, \dots, f_K] \in \mathbb{C}^{N \times K}$ olarak gösterilmiştir. Kanal ve önkodlama matrisi kullanılarak kullanıcılar arası diklik hesaplanmıştır. Kullanıcılar arası diklik kullanıcıların kanal vektörlerinin birbirine olan izdüşümü kullanılarak hesaplanmıştır. İzdüşüm değeri, θ eşik değeriyle karşılaştırılarak toplam kullanıcı hızını artıracak şekilde birbirine dik olmaya yakın kullanıcılar seçilmiştir. Algoritma için aşağıdaki başlangıç değerleri verilmiştir.

$$\Gamma_1 = \{1, \dots, K\} \quad (65)$$

$$i = 1 \quad (66)$$

$$S_0 = \emptyset \quad (67)$$

Seçilen kullanıcılarda her kullanıcı için $k \in \Gamma_i$ durumunda $\{g_1, \dots, g_{i-1}\}$ tarafından oluşturulan altuzaya dik olacak şekilde g_k hesaplanır. $i = 1$ durumu için $g_k = h_k$ olarak kullanılmıştır. g_k kullanılarak kullanıcılar arasından en iyisi seçilir ve S_0 kümesine eklenir.

$$g_k = h_k \left(I - \sum_{j=1}^{i-1} \frac{g_j^H g_j}{\|g_j\|^2} \right) \quad (68)$$

$$\pi(i) = \arg \max_{k \in \Gamma_1} \|g_k\| \quad (69)$$

$$S_0 \rightarrow S_0 \cup \{\pi(i)\} \quad (70)$$

$$h_i = h_{\pi(i)} \quad (71)$$

$$g_i = g_{\pi(i)} \quad (72)$$

$$\Gamma_{i+1} = \left\{ k \in \Gamma_1, k \neq \pi(i), \frac{h_k g_i^H}{\|h_k\| \|g_i\|} < \theta \right\}_{i \rightarrow i+1} \quad (73)$$

S_0 kümesine eklenen kullanıcıya eğer en az bir tane yarı dik kullanıcı seçilirse algoritma devam eder. Seçilen kullanıcı sayısı anten sayısına eşit olana kadar ya da herhangi bir adımda yarı dik kullanıcı seçimi yapılmadığında algoritma sona erer.

S kümesi seçilen kullanıcılardan oluşmaktadır. Seçilen kullanıcılara göre kanal matrisi $H(S)$ olarak belirlenmiştir. $H(S)$ kullanılarak önkodlama matrisi $F(S)$ tekrar hesaplanmaktadır. $H(S)$ kullanılarak kanal kazancı τ_i hesaplanmaktadır. Toplam güç P_t su doldurma algoritmasında kanal kazancı bilgisi kullanılarak seçilen kullanıcılara dağıtılır. Dağıtılan güce göre kullanıcı hızlarının toplamı hesaplanır.

$$F(S) = H(S)^\dagger = H(S)^H [H(S)H(S)^H]^{-1} \quad (74)$$

$$\tau_i = \frac{1}{\|f_i\|^2} = \frac{1}{[H(S)H(S)^H]^{-1}} \quad (75)$$

$$\eta_i = (\mu \tau_i - 1)^+ \quad (76)$$

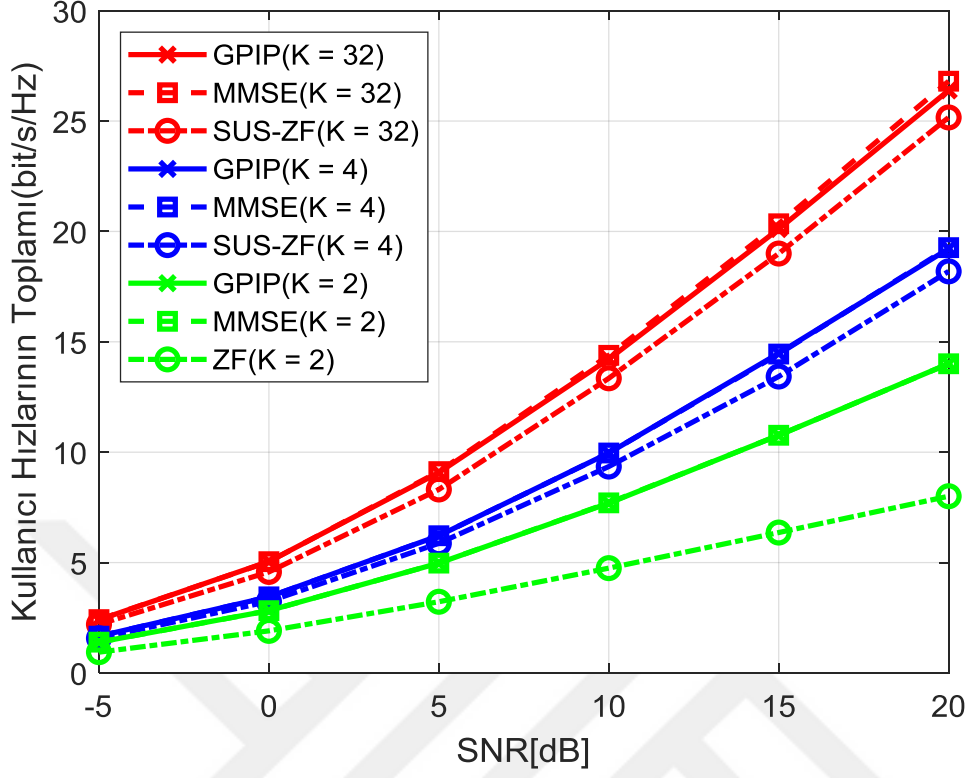
$$\sum_{i \in S} \left(\mu - \frac{1}{\tau_i} \right)^+ = P_t \quad (77)$$

$$R(S) = \max_{\eta_i: \sum_{i \in S} \tau_i^{-1} \eta_i < P_t} \sum_{i \in S} \log_2(1 + \eta_i) \quad (78)$$

7. SAYISAL SONUÇLAR

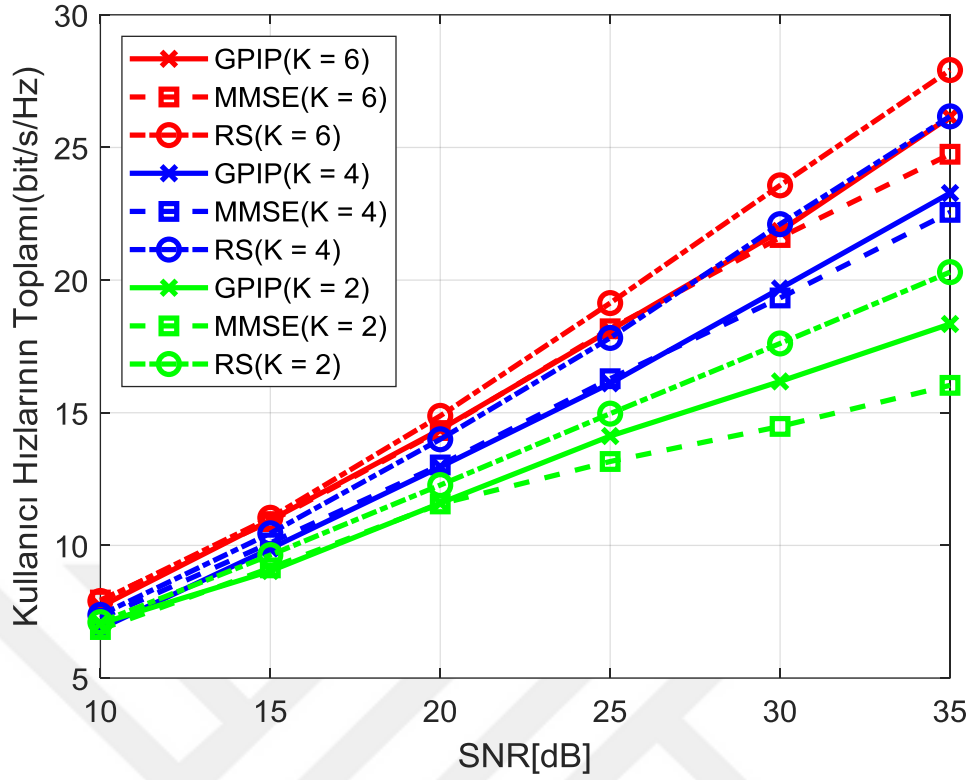
Bu bölümde, bu çalışmada elde edilen sonuçlar grafik üzerinde gösterilmiştir. GPIP, MMSE, SUS-ZF ve RS yöntemleri kanal bilgisinin hatasız ve hatalı bilindiği durumlarda farklı anten sayısı, kullanıcı sayısı ve SNR değerleri için incelenmiştir. Bu yöntemler için kullanıcı hızlarının toplamı ve çalışma süreleri karşılaştırılmıştır.

Şekil 7.1 ve 7.2’de sabit anten sayısı, $N = 4$, için kullanıcı sayılarının farklı olduğu az, kritik ve aşırı yüklü sistemler kanal bilgisinin hatasız ve hatalı olduğu durumlar için incelenmiştir. Az yüklü sistemlerde $K = 2$, kritik yüklü sistemlerde $K = 4$ ve aşırı yüklü sistemlerde hatasız kanal bilgisi için $K = 32$, hatalı kanal bilgisi için $K = 6$ olarak belirlenmiştir. Bu sistemler için hatasız kanal bilgisinde GPIP, MMSE ve SUS-ZF, hatalı kanal bilgisinde ise GPIP, MMSE ve RS yöntemleri kullanılmıştır. Bu yöntemleri kullanırken kullanıcıların toplam hızları karşılaştırılmıştır. Az yüklü sistem için hatasız kanalda $K = 2$ olduğu durumda SUS-ZF uygulanmamış, onun yerine ZF kullanıcı seçimi yapılmamıştır. Bunun nedeni SUS-ZF yönteminin kullanıcı sayılarının en az anten sayısına eşit olduğu durumlar için tanımlanmış olmasıdır. Hatasız kanallar için algoritma yakınsama eşik değeri $\epsilon = 0.001$, hatalı kanallar içinse eşik değeri $\epsilon = 0.001$ olarak belirlenmiştir.



Şekil 7.1: $N = 4$, hatasız kanal ve değişen SNR için az, kritik ve aşırı yüklü sistemler için toplam veri hızı grafiği

Şekil 7.1'in sonuçlarına bakıldığında her üç durumda da tüm SNR durumları için en iyi sonuç GPIP ve MMSE yöntemleri için gelmektedir. SUS-ZF bu ikisine göre biraz daha kötü sonuç vermektedir. GPIP ve MMSE ardışık dışbükey eniyileme temelli oldukları için, aynı amaç fonksiyonunun birinci dereceden türevini alarak hesaplanmaktadır. Bu yöntemler, bu şekilde benzer hesaplar yaptıklarından sonuçların da yakın çıkması anlamlıdır.

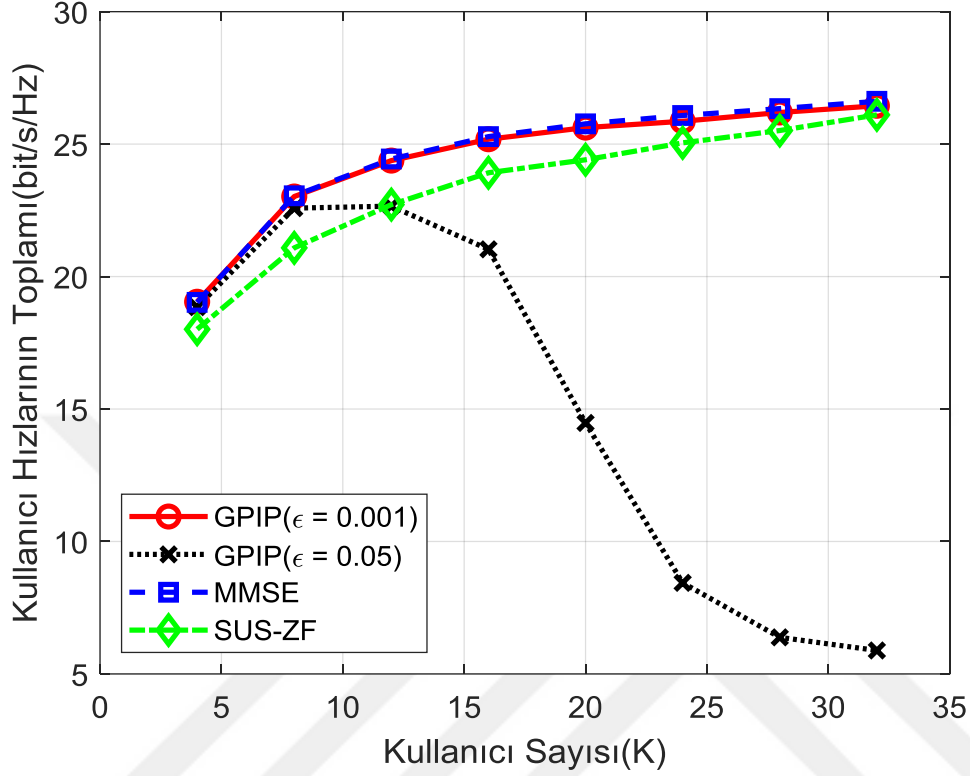


Şekil 7.2: $N = 4$, hatalı kanal ve değişen SNR için az, kritik ve aşırı yüklü sistemler için toplam veri hızı grafiği

Şekil 7.2'nin sonuçlarına bakıldığında her üç durumda da tüm SNR durumları için en iyi sonuç RS için gelmektedir. RS'ten sonra en iyi sonuç GPIP için hesaplanmaktadır. MMSE ise en kötü sonuca sahiptir. RS, MMSE'deki algoritmayı ortak ve özel semboller için kullandığından, RS'nin MMSE'nin daha iyi sonuçlar vermesi normaldir. Şekil 7.1'in aksine bu şekilde GPIP, MMSE'den daha iyi sonuç vermektedir.

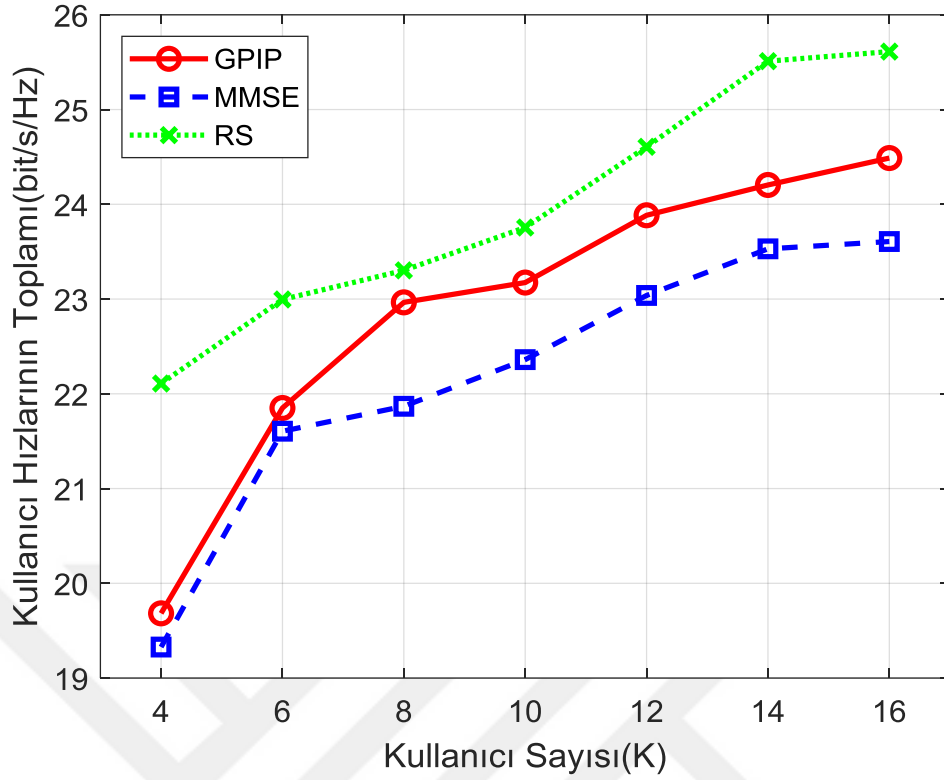
Şekil 7.3 ve 7.4'te sabit anten sayısı, $N = 4$, ve sabit SNR değeri için hatalı ve hatasız kanal durumunda farklı kullanıcı sayıları için sonuçlar gösterilmiştir. Şekil 7.3'te hatasız kanalda $SNR = 20\text{dB}$, için kullanıcı sayısının $K = 4$ 'ten, $K = 32$ 'ye kadar değiştiği durumlar için GPIP, MMSE ve SUS-ZF yöntemlerinin toplam veri hızı karşılaştırılmıştır. MMSE için algoritma yakınsama eşiği $\epsilon = 0.001$, GPIP için $\epsilon = 0.05$ ve $\epsilon = 0.001$ olacak şekilde iki farklı eşik değeri belirlenmiştir. Şekil 7.4'te hatalı kanalda $SNR = 30\text{dB}$, için kullanıcı sayısının $K = 4$ 'ten, $K = 16$ 'ya kadar değiştiği

durumlar için GPIP, MMSE ve RS yöntemlerinin toplam veri hızı karşılaştırılmıştır. Üç yöntem için de eşik değeri $\epsilon = 0.01$ olarak belirlenmiştir.



Şekil 7.3: $N = 4$, $SNR = 20dB$, hatasız kanal ve değişen kullanıcı sayıları için toplam veri hızı grafiği

Şekil 7.3'te GPIP'in eşik değerinin $\epsilon = 0.001$ olduğu durumda tüm kullanıcılarda MMSE'ye çok yakın sonuçlar gelmektedir. Önceki şekildeki gibi SUS-ZF bu iki yönteme göre biraz daha kötü bir sonuç vermektedir. Eşik değerinin $\epsilon = 0.05$ olduğu ve kullanıcı sayısının 8'den fazla olduğu durumlarda GPIP'in sonucu kötüleşmektedir. Bu durum GPIP'in yakınsama eşiği olarak seçilen epsilon değerinin algoritmanın başarısını önemli ölçüde etkilediğini göstermektedir. Kullanıcı sayısının 8'den fazla olduğu durumlarda GPIP için daha düşük epsilon değerleri kullanılmalıdır ancak eşik değerinin çok düşürülmesi GPIP algoritmasındaki önkodlama hesabında tekrarlanan adım sayısını çok artırmaktadır ve bu yüzden bu yöntemin maliyetini artırmaktadır. Bu yüzden daha maliyetsiz şekilde aynı sonucu veren MMSE yöntemi daha avantajlıdır.

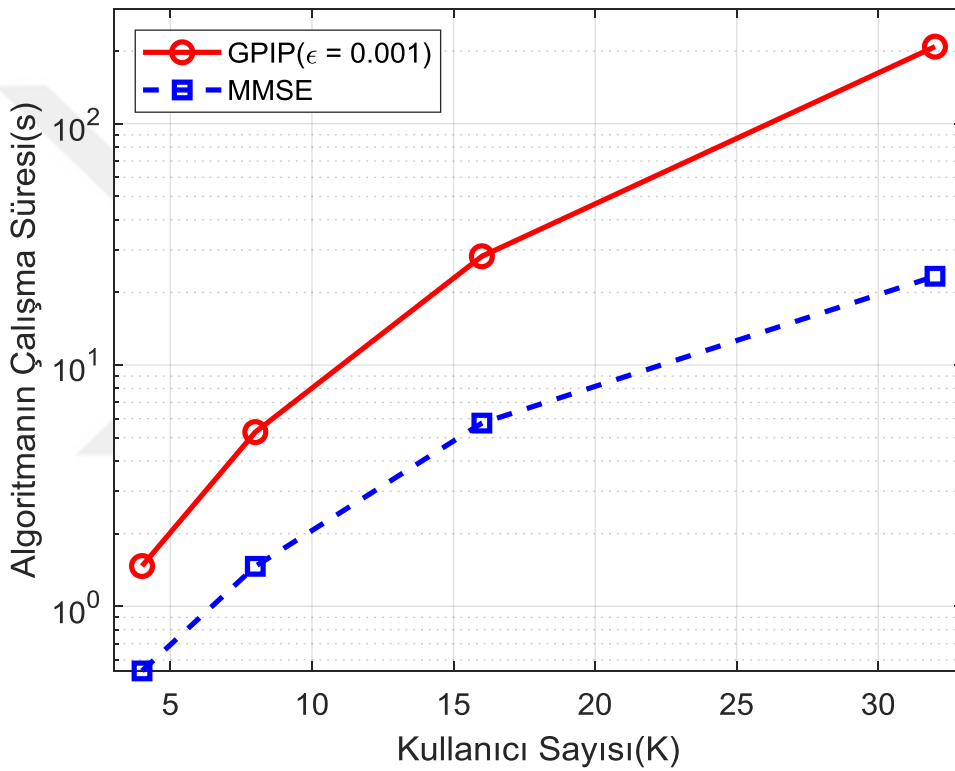


Şekil 7.4: $N = 4$, $SNR = 30dB$, hatalı kanal ve değişen kullanıcı sayıları için toplam veri hızı grafiği

Şekil 7.4'te tüm kullanıcı durumları için en iyi sonuç RS'de gelmektedir. RS'ten sonraki en iyi sonuç GPIP'de gelmektedir. En kötü sonuç MMSE'de gelmektedir. Şekil 7.2 ile karşılaştırıldığında sonuçlar benzerlik göstermektedir. Şekil 7.3'ün aksine GPIP Şekil 7.2'de olduğu gibi MMSE'den daha iyi sonuç vermektedir. RS'in hatalı kanal grafiklerinin ikisinde de en iyi sonucu vermesi bu yöntemin GPIP ve MMSE'ye göre daha iyi bir önkodlama yöntemi olduğunu göstermektedir. Üç yöntem için de kullanıcı sayısı arttıkça, kullanıcı hızlarının toplamı artmaktadır.

Şekil 7.5'te GPIP ve MMSE algoritmalarının, Şekil 7.6'da ise GPIP, MMSE ve RS algoritmalarının, sabit anten sayısı, $N = 4$, ve kullanıcı sayısının değiştiği durumlar için çalışma süreleri verilmiştir. Şekil 7.5'te kullanıcı sayısının $K = 4$ 'ten, $K = 32$ 'ye kadar değiştiği ve yakınsama eşiğinin $\epsilon = 0.001$ olduğu durumlar kullanılmıştır. Bu durumlar için SNR'ın $-5dB$ 'den $5dB$ artarak $20dB$ 'ye kadar, 10 farklı kanal durumu için geçen toplam süre hesaplanmıştır. Şekil 7.6'da kullanıcı sayısının $K = 4$ 'ten, $K =$

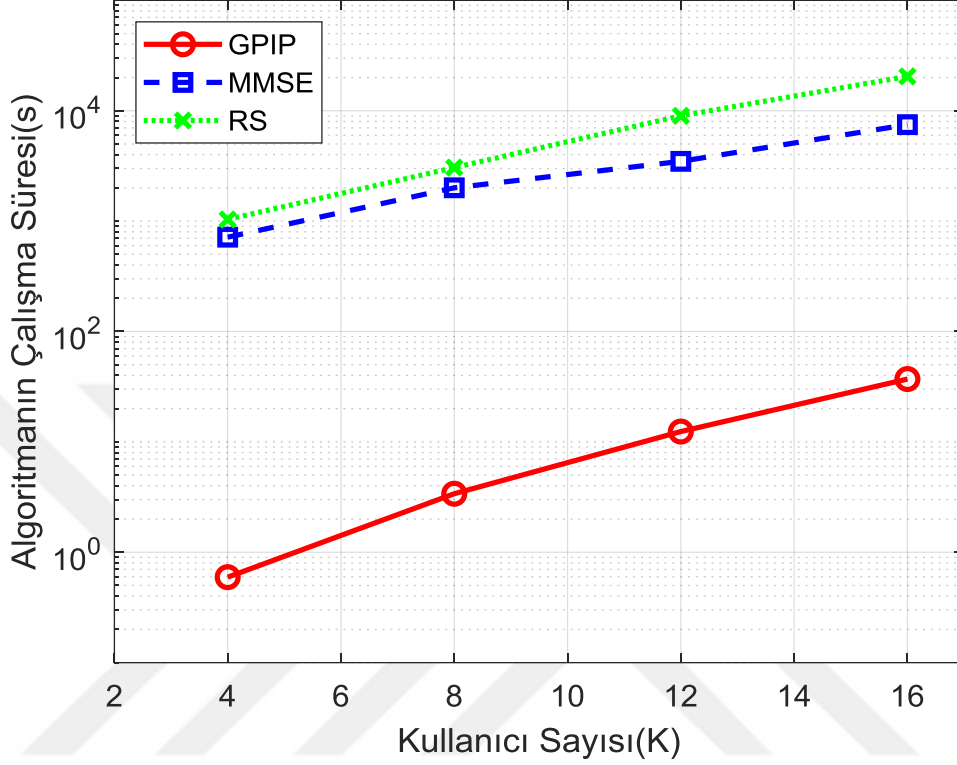
16'ya kadar deđiřtiđi ve yakınsama eřiđinin $\epsilon = 0.01$ olduđu durumlar kullanılmıřtır. Bu durumlar iin SNR'ın 10dB'den 5dB artarak 35dB'ye kadar, 5 farklı kanal durumu iin geen toplam sre hesaplanmıřtır. Bu iki grafik iin GPIIP, MMSE ve RS, Intel Core i7-6500U iřlemcili 8GB RAM'e sahip bir bilgisayarda hesaplanmıřtır. alıřma sresinin saniye cinsinden belirtilmesi farklı algoritmaların yinleme sayıları ve yakınsama sreleri farklı olduđundan ve farklı SNR deđerleri iin yakınsama farklı sayılarda gerekleřtiđinden parametrik incelemelerin sunmadıđı önemli bir ölç sunmaktadır.



řekil 7.5: $N = 4$ ve hatasız kanal bilgisi iin kullanıcı sayısına gre deđerlen algoritma alıřma sresi

řekil 7.5 hatasız kanallarda tm farklı kullanıcı sayıları iin MMSE ynteminin GPIIP yntemine gre daha kısa srede srdđn gstermektedir. Bu yntemlerin kullanıldıđı makalelere bakıldıđında GPIIP'in karmařıklık seviyesi dng bařına [4]'te $O(KN^3)$, MMSE ise [2] dng bařına $O(KN)$ olarak gsterilmiřtir. Grafikte elde edilen sonular bu bilgilerle tutarlı bir sonu vermektedir. MMSE GPIIP'e gre daha

kısa sürede sürmektedir ve en yüksek kullanıcı sayısı olan $K = 32$ durumu için GPIIP MMSE'in 10 katı sürede bitmektedir.



Şekil 7.6: $N = 4$ ve hatalı kanal bilgisi için kullanıcı sayısına göre değişen algoritma çalışma süresi

Şekil 7.6 hatalı kanallarda tüm farklı kullanıcı sayıları için GPIIP yönteminin MMSE yöntemine göre daha kısa sürede sürdüğünü göstermektedir. Burada şekil 7.5'teki sonucun tam tersi elde edilmiştir. Bunun nedeni hatasız kanalda MMSE kullanılırken algoritma analitik bir şekilde yazılmıştır ama hatalı kanalda MMSE, MATLAB'ın eniyileme problemleri için kullandığı bir çözücü olan cvx problem çözücü kullanılarak çözülmüştür. cvx kullanımı algoritmanın çalışma süresini artırmaktadır ve bu yüzden MMSE GPIIP'e göre çok daha uzun sürede çalışmaktadır. $K = 16$ durumu için MMSE GPIIP'in yaklaşık 200 katı sürede bitmektedir. RS ise GPIIP ve MMSE ile karşılaştırıldığında çok daha uzun sürmektedir. Özellikle RS ve GPIIP arasında çok büyük bir fark vardır. $K = 16$ durumu için RS GPIIP'in yaklaşık 550 katı sürede bitmektedir. RS'in bu kadar uzun sürmesinin nedeni MMSE'de olduğu gibi RS'de de

cvx çözücü kullanılmasıdır. RS MMSE'den daha uzun sürede çalışmaktadır, çünkü hatalı kanalda kullanılan MMSE problemi, RS problemi sadeleştirilerek oluşturulmuştur. RS problemi daha karmaşık olduğu için daha uzun sürede çalışmaktadır.

Sonuç olarak, grafiklere bakıldığında hatasız kanalda GPIP, MMSE ile benzer performansı daha yüksek karmaşıklıkta ve daha uzun sürede elde etmektedir. Bu nedenle kanal bilgisinin hatasız bilindiği tüm durumlarda tüm farklı kullanıcı ve anten sayıları için MMSE daha avantajlı bir yöntemdir. Hatalı kanallarda ise en iyi sonucu RS vermektedir, ama diğer yöntemlere göre daha uzun sürede çalışmaktadır. GPIP ise MMSE'ye göre hem daha iyi sonuç vermekte hem de daha kısa sürede çalışmaktadır. Bu yüzden kanal bilgisinin hatalı bilindiği tüm durumlarda tüm farklı kullanıcı ve anten sayıları için RS daha avantajlı bir yöntemdir.

8. SONUÇ VE TARTIŞMA

Bu çalışmada, çok kullanıcılı çok antenli bir aşağı gönderim sisteminde, kanal bilgisinin hatasız ve hatalı bilindiği durumlar için literatürde yeni önerilen GPIP yöntemi, MMSE SUS-ZF ve RS yöntemleriyle karşılaştırılmıştır. Karşılaştırmalarda başarı kriteri olarak toplam veri hızı ve algoritma çalışma süresi kullanılmıştır. Kullanıcı sayısı ve SNR'dan bağımsız bir şekilde, kanal bilgisinin hatasız bilindiği durumlarda GPIP yönteminin MMSE ile benzer sonuçları elde ettiği fakat daha yüksek karmaşıklık gerektirdiği gözlemlenmiştir. Bu yüzden çok kullanıcılı çok antenli aşağı gönderim sistemlerinde önkodlama yöntemi olarak eğer kanal bilgisi hatasız ise MMSE tercih edilmelidir. Kanal bilgisinin hatalı bilindiği durumlarda, tüm farklı kullanıcı sayısı ve SNR değerleri için GPIP, MMSE'ye göre daha yüksek toplam kullanıcı hızına sahiptir ve sonuçlar daha kısa sürede hesaplanmaktadır. GPIP, RS ile karşılaştırıldığında ise daha kötü sonuç vermektedir ama RS'e göre daha kısa sürede çalışmaktadır. Bu yüzden çok kullanıcılı çok antenli aşağı gönderim sistemlerinde önkodlama yöntemi olarak eğer kanal bilgisi hatalı ise RS tercih edilmelidir ama eğer daha kısa sürede çalışan bir yöntemin kullanılması gerekiyorsa, RS'e göre daha kötü sonuç veren ama yine de iyi bir sonuç veren GPIP tercih edilebilir.

GPIP'nin MMSE yöntemi ile karşılaştırılması basit görünse de, MMSE'nin literatürdeki yaygın kullanımı ve çok kullanıcılı çok antenli aşağı gönderim sistemlerindeki farklı optimizasyon problemleri için farklı önkodlama yöntemleriyle karşılaştırılmış olması GPIP'nin literatürdeki yerini anlamamız için önemlidir. Sonuç olarak, [4] makalesindeki sonuçlara ek olarak GPIP ilk defa MMSE ile karşılaştırılmış ve ilk defa aşırı yüklü sistemlerde kullanılmıştır. Bu sayede GPIP hakkında yeni bilgiler edinilmiş ve GPIP'nin literatür içindeki yerini konumlandırmamızı olanaklı kılmıştır.

KAYNAKLAR

- [1] **J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang**, “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.
- [2] **S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi**, “Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, 2008
- [3] **E. Telatar**, “Capacity of Multi-antenna Gaussian Channels,” *Euro. Trans. Telecomm.*, vol. 10, no. 6, pp. 585–596, Nov. 1999
- [4] **J. Choi, N. Lee, S. Hong, and G. Caire**, “Joint user selection, power allocation, and precoding design with imperfect CSIT for multi-cell MUMIMO downlink systems,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 162–176, 2020.
- [5] **C. Johnson**, *5G new radio in bullets*, 1st edition Farnham, England: Chris Johnson, 2019, pp. 8-11.
- [6] **S. Ahmadi**, *5G NR Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards*, Elsevier, 2019, pp. 1-lx
- [7] **A. Goldsmith**, *Wireless Communications*. Cambridge University Press, 2005, pp. 309-312, 333-340.
- [8] **L. Zheng, D. Tse** “Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels”, *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1073-1074, 2003
- [9] **H. Weingarten, Y. Steinberg, and S. S. Shamai**, “The capacity region of the Gaussian multiple-input multiple-output broadcast channel,” *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 3936–3964, 2006.
- [10] **M. Costa**, “Writing on dirty paper (corresp.),” *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, 1983
- [11] **Q. H. Spencer, A. L. Swindlehurst, and M. Haardt**, “Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels,” *IEEE Tran. Sig. Proc.*, vol. 52, no. 2, pp. 461–471, 2004.
- [12] **T. Yoo and A. Goldsmith**, “On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming,” *IEEE Journal on selected areas in communications*, vol. 24, no. 3, pp. 528–541, 2006.
- [13] **S. Yang, M. Kobayashi, D. Gesbert, and X. Yi**, “Degrees of freedom of time correlated MISO broadcast channel with delayed CSIT,” *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 315–328, Jan. 2013.

- [14] **Y. Mao, B. Clerckx, and V. O. Li**, "Rate-splitting multiple access for downlink communication systems: bridging, generalizing, and outperforming SDMA and NOMA," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–54, 2018.
- [15] **H. Joudeh and B. Clerckx**, "Sum-Rate Maximization for Linearly Precoded Downlink Multiuser MISO Systems With Partial CSIT: A Rate-Splitting Approach," in *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4847–4861, Nov. 2016.
- [16] **Z. Li, C. Ye, Y. Cui, S. Yang, and S. Shamai**, "Rate splitting for multiantenna downlink: Precoder design and practical implementation," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1910–1924, Jun. 2020.
- [17] **A. Krishnamoorthy and R. Schober**, "Successive null-space precoder design for downlink MU-MIMO with rate splitting and single-stage SIC," arXiv preprint arXiv:2101.01147, 2021.
- [18] **S. Yang, M. Kobayashi, D. Gesbert, and X. Yi**, "Degrees of freedom of time correlated MISO broadcast channel with delayed CSIT," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 315–328, 2013.
- [19] **T. Cover, J. Thomas**, *Elements of Information Theory*, 2nd edition, New Jersey, United States, Wiley, 2006, pp. 199–205.
- [20] **R. D. Yates, D. J. Goodman**, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, 3rd edition, New Jersey, United States, Wiley, Inc. 2014, pp. 343–345.
- [21] **A. Shaprio, D. Dentcheva, A. Ruszczynski**, *Lectures on Stochastic Programming: Modelling and Theory*, 2nd edition, Philadelphia, United States, SIAM 2014, pp. 175–177.

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Ç. Yalçınduran, M. Yüksel.** “Çok Kullanıcılı Çok Antenli Aşağı Gönderim Sistemleri İçin Bir Karşılaştırma Çalışması”, 29. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Çevrimiçi, Haziran 2021

