

IONOLAB YÖNTEMİ İLE
WEB TABANLI TOPLAM ELEKTRON İÇERİĞİ (TEİ)
HESAPLANMASI VE SUNUMU

WEB BASED COMPUTATION AND PRESENTATION
OF TOTAL ELECTRON CONTENT (TEC)
USING THE IONOLAB METHOD

ORHAN UĞURLU

Hacettepe Üniversitesi
Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK ve ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2007

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :.....
Prof. Dr. Feza ARIKAN

Üye :.....
Yrd. Doç. Dr. Ali Ziya ALKAR

Üye :.....
Yrd. Doç. Dr. Harun ARTUNER

Üye :.....
Yrd. Doç. Dr. Atila YILMAZ

Üye (Danışman) :.....
Dr. Umut SEZEN

ONAY

Bu tez .. / .. / tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

Prof. Dr. Ahmet R. ÖZDURAL
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

IONOLAB YÖNTEMİ İLE WEB TABANLI TOPLAM ELEKTRON İÇERİĞİ (TEİ) HESAPLANMASI VE SUNUMU

Orhan Uğurlu

ÖZ

İyonosferin gözlenmesi radyo dalgalarının yayılımına olan etkileri nedeniyle önem taşımaktadır. Toplam Elektron İçeriği (TEİ) iyonosferik etkilerin gözlenmesinde yaygın olarak kullanılmaktadır. TEİ, iyonosfer ile yer arasındaki bölgenin 1 m^2 'lik kesitinde bulunan toplam serbest elektron sayısı olarak tanımlanmaktadır. IRI modelinin ve IGS analiz merkezlerinin TEİ kestirimleri üretebilmesine rağmen, bu kestirimlerin zamansal çözünürlüğü düşüktür veya kestirimler ampirik veriler üzerine kuruludur. IONOLAB (www.ionolab.org) kestirim yöntemi yüksek zaman çözünürlüklü gürbüz TEİ kestirimleri üretebilmektedir. Bu çalışmada IONOLAB TEİ kestirim yöntemi uygulanmış, kestirimler kolay ulaşılabilir ve kullanıcı dostu bir arayüz üzerinden sunulmuştur. Geliştirilen uygulama seçilen GPS istasyonu ve zaman dilimi için TEİ kestirimi yapabilmektedir. Kestirim için gerekli kullanıcı girdisi GPS istasyon adı ve gün seçiminden oluşmaktadır. Kestirim için gerekli RINEX formatındaki gözlem verileri IGS veri merkezlerinden otomatik olarak indirilmektedir. Geliştirilen uygulama 30 s zaman çözünürlüğünde kestirimleri sunabilen ilk sistemdir. Web tabanlı TEİ kestirim uygulaması kullanıcı tarafından kurulum gerektirmeyen kolay ulaşılabilir bir sistemdir. Uygulama katmanlı bir mimariye sahiptir. Bu modüler tasarım sayesinde, kestirim yöntemiyle ilgili olabilecek değişiklikler kolaylıkla uygulamaya entegre edilebilmektedir. Aynı esneklik veri erişimiyle ilgili olarak da sağlanmaktadır. Sunum katmanı farklı kullanıcı türlerini destekleyebilecek şekilde tasarlanmıştır.

Anahtar Kelimeler: GPS, İyonosfer, Toplam Elektron İçeriği

Danışman: Dr. Umut SEZEN, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı

WEB BASED COMPUTATION AND PRESENTATION OF TOTAL ELECTRON CONTENT (TEC) USING THE IONOLAB METHOD

Orhan Uğurlu

ABSTRACT

Monitoring ionosphere is important because of its effects on radio wave propagation. Total Electron Content (TEC) is widely used in monitoring ionospheric effects. TEC is expressed as the amount of free electrons within 1 m^2 cross-sectional area of the region between ground and ionosphere. Although IRI model and IGS analysis centers can generate and present TEC estimations, either the temporal resolution is low or estimations are based on empirical data. IONOLAB (www.ionolab.org) method provides robust estimates with high temporal resolution. In this study, TEC estimation method by IONOLAB is implemented and estimations are presented by an easily accessible, user-friendly interface. Developed application makes TEC estimation for given GPS station and time period. User input required contains only the selection of GPS station and the date. Observation data needed for estimation is retrieved from IGS data centers in RINEX format. Developed application is the first one that provides TEC estimations with a temporal resolution of 30 s over the internet. Web based TEC estimation application is an easily accessible system that requires no installation on the client side. The application has a layered design. By means of this modular design, possible changes regarding the estimation method can be easily adapted. Same flexibility is also provided for the data access. The presentation of estimation data is architected to support different client types.

Keywords: GPS, Ionosphere, Total Electron Content

Advisor: Dr. Umut SEZEN, Hacettepe University, Department of Electrical and Electronics Engineering, Electrical and Electronics Engineering Section

TEŐEKKÜR

Deęerli hocam Dr. Umut SEZEN'e bu alıŐmanın ortaya ıkmasındaki byk katkılarından dolayı itenlikle teŐekkr ederim.

Ayrıca sayın Prof. Dr. Feza ARIKAN'a gsterdięi ilgiden dolayı teŐekkrlerimi sunarım.

Bu alıŐmayı yaparken bana verdikleri maddi ve manevi destekten dolayı eŐim Zehra ve kızlarım İrem ve Ceren'e de sonsuz teŐekkr ederim.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZ.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER DİZİNİ.....	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	viii
KISALTMALAR	ix
1 GİRİŞ	1
2 TEİ KESTİRİM YÖNTEMİ.....	5
2.1 GPS Verileriyle TEİ Kestirimi	5
2.2 GPS Verileriyle Ölçümler	6
2.2.1 Sözde Menzil Modeli	6
2.2.2 Taşıyıcı Faz Modeli	6
2.3 GPS Verileriyle Toplam Elektron İçeriği Kestirimi	7
2.3.1 Sözde Menzil Ölçümleriyle TEİ Kestirimi	7
2.3.2 Faz Ölçümleriyle TEİ Kestirimi.....	8
2.4 IONOLAB yöntemi ile TEİ kestiriminin Düzgünleştirilmesi	9
2.4.1 Tüm gün için düzgünleştirme yöntemi	9
2.4.2 Belirli bir zaman aralığı için düzgünleştirme yöntemi	12
3 UYGULAMA YAZILIMI.....	13
3.1 İşlevsel Bakış	13
3.2 TEİ Kestirim Yöntemi Gerçekleştirimi	14
3.3 Mimari Tasarım	19
3.4 Detaylı Tasarım	27
3.4.1 Sunum Katmanı.....	28
3.4.2 İş Katmanı	30
3.4.3 Veri Erişim Katmanı.....	34
3.5 Veritabanı Tasarımı	36

3.6	Kullanıcı Arayüzü	38
4	ANALİZ	45
4.1	MATLAB Uygulamasıyla Karşılaştırma	45
4.2	IGS Analiz Merkezleriyle Karşılaştırma	47
4.3	IRI ile Karşılaştırma	49
4.4	Başarım	49
4.5	Alt Örnekleme	53
5	SONUÇ VE ÖNERİLER	57
	KAYNAKLAR	59
	KAVRAMLAR SÖZLÜĞÜ	63
	EKLER	65
A	TASARIM GÖSTERİM SİSTEMİ	66
A.1	Paket Şeması	66
A.2	Sıralama Şeması	66
A.3	Sınıf Şeması	67
B	TASARIM SINIF SÖZLÜĞÜ	68
C	DETAYLI TASARIM	70
C.1	Sunum Katmanı	70
C.2	İş Katmanı	76
C.3	Veri Erişim Katmanı	79
C.4	Diğer Sınıflar	83
D	ÖRNEK GÖZLEM DOSYALARI	88
D.1	Örnek Alıcı Bağımsız Aktarım (RINEX) dosyası	88
D.2	Örnek SP3 dosyası	90
D.3	Örnek İyonosfer Harita Aktarımı (IONEX) dosyası	93
	ÖZGEÇMİŞ	95

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1 İyonosfer ile yer arasında birden fazla sekme ile uzun mesafe yayılım (Hunsucker and Hargreaves, 2003)	1
1.2 IGS izleme ağındaki istasyonlar (IGS, 2006)	3
3.1 Gerçekleştirilen Uygulamanın Akış Şeması	15
3.2 Uygulama Mimarisi	19
3.3 Spring çatısını oluşturan modüller (Walls and Breidenbach, 2005) .	24
3.4 MVC Tasarım Kalıbı (Crawford and Kaplan, 2002)	26
3.5 Uygulama mimarisi detayı	27
3.6 Uygulamayı oluşturan paketler	28
3.7 Spring MVC Yapısı	29
3.8 Web katmanının tek istasyon için sözde menzil verileri ile TEİ kestirimi yapılması senaryosuna ait sıralama şeması	31
3.9 İş katmanı sınıflarının tek istasyon için sözde menzil verileri ile TEİ kestirimi yapılması senaryosu için sıralama şeması	32
3.10 Toplam Elektronik İçeriği (TEİ) kestirim gerçekleştirimi ile ilgili arayüz ve sınıfları içeren sınıf şeması	34
3.11 Düzgünleştirme işlemi gerçekleştirimi ile ilgili sıralama şeması	34
3.12 Gözlem Kaynağı Gerçekleştirimi sınıfının GPS gözlem verilerinin alınması senaryosuna ait sıralama şeması	35
3.13 Gerçekleştirilen uygulamaya ait veritabanı tasarımı	36
3.14 Uygulama kullanıcı arayüzü	38
3.15 Tek istasyon için TEİ Kestirimi veri giriş ekranı	39
3.16 İstasyon seçim ekranı (IGS, 2006)	40
3.17 Tek istasyon için TEİ kestirim grafiği	40
3.18 Çok İstasyon için TEİ Kestirimi veri giriş ekranı	41
3.19 Çok istasyon için TEİ Kestirim grafiği	42
3.20 Tek istasyon-Gün Karşılaştırma için TEİ Kestirimi veri giriş ekranı	42
3.21 Tek istasyon-Gün Karşılaştırma için TEİ Kestirim grafiği	43

3.22	Eksik gözlem verisi içeren durum için TEİ Kestirim grafiği	44
3.23	Tek İstasyon için Excel formatında TEİ Kestirim çıktısı	44
4.1	Geliştirilen uygulama ile MATLAB uygulamasına ait TEİ kestirimlerinin karşılaştırması	46
4.2	IGS analiz merkezlerinin ve geliştirilen uygulamanın TEİ kestirimleri	48
4.3	IRI ile geliştirilen uygulamanın TEİ kestirimleri	50
4.4	Farklı alt örnekleme oranlarıyla elde edilen TEİ kestirimleri	54
A.1	Örnek paket şeması	66
A.2	Örnek sıralama şeması	67
A.3	Örnek sınıf şeması	67
C.1	web paketi için sınıf şeması	73
C.2	calculation paketi için sınıf şeması	77
C.3	dao paketi için sınıf şeması	80
C.4	data paketi için sınıf şeması	84

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
4.1 Geliştirilen uygulama ile MATLAB uygulamasının ürettiği kestirimlerin karşılaştırılması	46
4.2 Geliştirilen uygulama ile MATLAB uygulamasının düzgünleştirme süreleri	51
4.3 Geliştirilen uygulama ile MATLAB uygulamasının matris tersi alma süreleri	51
4.4 MTJ'nin JLAPACK ve NNI ile başarımı	53
4.5 Alt örnekleme oranları için kullanılan kayan ortanca filtrenin pencere genişlikleri	55
4.6 Farklı alt örnekleme oranlarıyla elde edilen kestirimlerin karşılaştırılması	56

KISALTMALAR

AJAX Asenkron Java betiđi ve XML (Asynchronous Javascript and XML)

AOP İlgkiye Yönelik Programlama (Aspect Oriented Programming)

API Uygulama Programlama Arayüzü (Application Programming Interface)

BLAS Temel Lineer Cebir Altprogramları (Basic Linear Algebra Subprograms)

CLAPACK LAPACK'in C sürümü

CODE Avrupa Yörünge Belirleme Merkezi (Center for Orbit Determination in Europe)

COSPAR Uzak Araştırma Komitesi (Committee on Space Research)

CRX Sıkıştırılmış RINEX (Compressed RINEX)

DAO Veri Erişim Nesnesi (Data Access Object)

EJB Kurumsal Java Bileşenleri (Enterprise Java Beans)

ESA Avrupa Uzak Ajansı (European Space Agency)

FTP Dosya Transfer Protokolü (File Transfer Protocol)

GPS Küresel Konumlama Sistemi (Global Positioning System)

HTML Yardımlı Metin İşaretleme Dili (Hypertext Markup Language)

HTTP Yardımlı Metin Transfer Protokolü (Hypertext Transfer Protocol)

IGS Uluslararası Küresel Seyrüsefer Uydu Sistemi Servisi (International GNSS Service)

IoC Denetim Çevrimi (Inversion of Control)

IONEX İyonosfer Harita Aktarımı (Ionosphere Map Exchange)

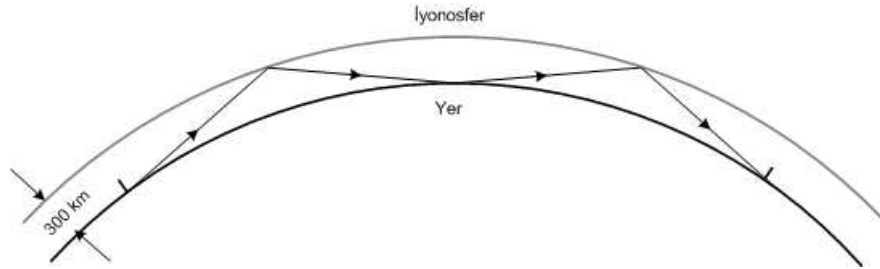
- IONOLAB** Hacettepe Üniversitesi İyonosfer Araştırma Laboratuvarı
- IRI** Uluslararası Referans İyonosferi (International Reference Ionosphere)
- ISO** Uluslararası Standartlaştırma Örgütü (International Organization for Standardization)
- JavaEE** Java Kurumsal Sürümü (Java Enterprise Edition)
- JavaSE** Java Standart Sürümü (Java Standard Edition)
- JDBC** Java Veritabanı Bağlanırlığı (Java Database Connectivity)
- JLAPACK** LAPACK'in Java sürümü
- JNI** Java Sistemine Özgü Arayüzü (Java Native Interface)
- JPL** Jet İtme Laboratuvarı (Jet Propulsion Laboratory)
- JSF** Java Sunucu Yüzleri (Java Server Faces)
- JSP** Java Sunucu Sayfaları (Java Server Pages)
- JSTL** JSP Standart Etiket Kütüphanesi (JSP Standard Tag Library)
- LAPACK** Lineer Cebir Paketi (Linear Algebra Package)
- MTJ** Java İçin Matris Araç Takımı (Matrix Toolkit for Java)
- MVC** Model Görünüm Denetleyici (Model View Controller)
- NNI** Sistemine Özgü Sayısal Arayüz (Native Numeric Interface)
- NRMSE %** Düzgelenmiş Yüzdesele Etkin Hata (Normalized Root Mean Square Error Percent)
- OSI** Açık Sistemler Arabağlaşımı (Open Systems Interconnection)
- PNG** Taşınabilir Ağ Grafiği (Portable Network Graphics)
- POJO** Basit Eski Java Nesneleri (Plain Old Java Object)

- RINEX** Alıcı Bağımsız Aktarım (Receiver Independent Exchange)
- RMI** Uzak Metot Çağırımı (Remote Method Invocation)
- SQL** Standart Sorgu Dili (Standard Query Language)
- STEC** Eğik Toplam Elektron İçeriği (Slant Total Electron Content)
- TECU** Toplam Elektron İçeriği Birimi (Total Electron Content Unit)
- TEİ** Toplam Elektron İçeriği
- UML** Birleşik Modelleme Dili (Unified Modelling Language)
- UPC** Katalunya Politeknik Üniversitesi (La Universitat Politècnica de Catalunya)
- URSI** Uluslararası Radyo Bilimi Birliği (Union Radio-Scientifique Internationale)
- VTEC** Dikey Toplam Elektron İçeriği (Vertical Total Electron Content)
- XML** Genişletilebilir İşaretleme Dili (Extensible Markup Language)

1. GİRİŞ

İyonosfer elektriksel olarak nötr bir ortamda genellikle eşit miktarda serbest elektron ve pozitif iyonlar içeren atmosfer katmanıdır. Yüklü parçacıklar nötr olanlara göre az miktarda olmalarına rağmen ortamın elektriksel özelliklerini önemli ölçüde etkilemektedirler. İyonosferdeki iyonlaşmanın başlıca sebebi Güneş'tir. İyonosferdeki yapılar zamansal olarak Güneş lekeli durumuna, mevsimsel ve günlük dönemlere göre olduğu gibi coğrafi konuma göre de büyük değişiklikler göstermektedir. İyonlaşmanın büyük bir kısmı Güneş'ten gelen X ışınları ve ultraviyole radyasyon sonucu oluşmaktadır. En önemli etki Dünya'nın Güneş'e göre dönmesi sonucu oluşmaktadır. İyonlaşma Güneş tarafından aydınlanan yarımkürede artarken, karanlık bölgede azalmaktadır (Hunsucker and Hargreaves, 2003).

Uzun mesafeli radyo iletişimi yüklü parçacıkların sebep olduğu yansımalar sayesinde mümkün olmaktadır. Şekil 1.1'de görüldüğü gibi, iyonosfer yerin 100 ile 400 km üzerinde yerleştirilmiş yansıyan sinyalleri dünyanın diğer kısımlarına ulaştıran bir ayna gibidir. İyonosferin üst tabakalarından yansıyan



Şekil 1.1: İyonosfer ile yer arasında birden fazla sekme ile uzun mesafe yayılım (Hunsucker and Hargreaves, 2003)

sinyaller şüphesiz daha uzun mesafelere ulaşabilmektedir ve bu tabakalar daha yüksek frekanslı sinyalleri daha iyi yansıtmaktadır. Ancak bu yansıma iyonosferin o andaki yapısına bağlıdır. İyonosferin diğer bir etkisi de alt tabakalarının sinyalleri emme eğilimi göstermesidir. Alçak frekanslı dalgalar daha çok emilmektedir. Bu nedenlerden ötürü radyo haberleşmesinde o anın iyonosfer durumuna

göre doğru frekansı seçmek büyük önem arz etmektedir. Diğer bir deyişle iyonosferin yapısını ve durumunu bilmek, etkili radyo haberleşmesinin temelini oluşturmaktadır (Hunsucker and Hargreaves, 2003). Dolayısıyla iyonosferin yapısının araştırılması ve gözlenmesi önemli bir araştırma konusu olmuştur.

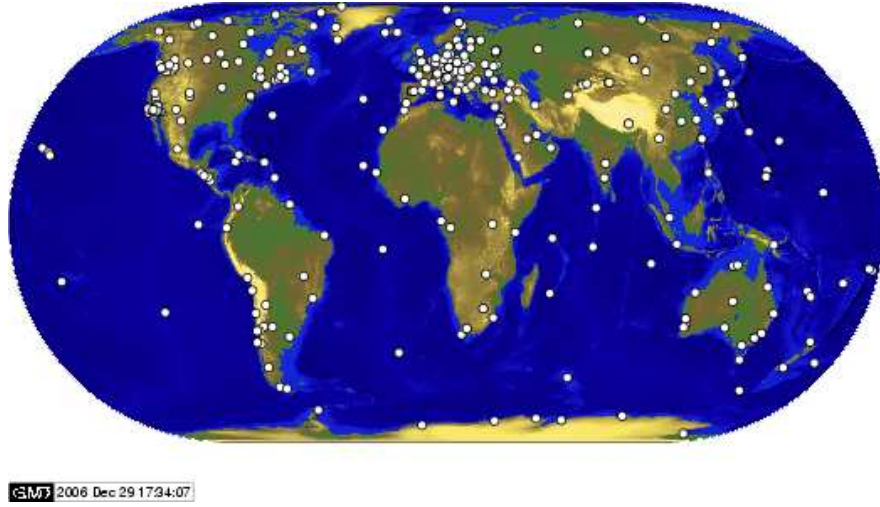
İyonosferle ilgili teorik ve ampirik olmak üzere çeşitli modeller geliştirilmiştir. Bunlardan en öne çıkan ve yaygın olarak kullanılan Uluslararası Referans İyonosferi (IRI) modelidir. IRI modeli Uzay Araştırma Komitesi (COSPAR) ve Uluslararası Radyo Bilimi Birliği (URSI) sponsorluğunda geliştirilmiştir. IRI modeli belirli bir konum ve zaman için elektron yoğunluğu, elektron içeriği, elektron sıcaklığı, iyon sıcaklığı ve iyon bileşimi kestirimleri üretebilmektedir. IRI modeli değişik ölçüm kaynaklarından elde edilen verilerle güncellenen ampirik bir modeldir (Bilitza, 1990).

İyonosfer yapısını ifade eden önemli değişkenlerin başında TEİ gelmektedir. TEİ, yer ile iyonosfer arasında kalan 1 m²'lik kesite sahip sütunun içerdiği serbest elektron sayısı olarak ifade edilmektedir ve birimi Toplam Elektron İçeriği Birimi (TECU)¹'dur. Modellemenin yanı sıra çeşitli sistemler aracılığıyla iyonosfer yapısıyla ilgili ölçümler yapılabilmektedir. Bu ölçümler için yer ve uzay tabanlı sistemler kullanılmaktadır. Bu sistemler Komjathy (1997) tarafından detaylı olarak anlatılmıştır. Küresel Konumlama Sistemi (GPS) uydularının iki L-Bandı frekansından gönderdiği elektromanyetik dalgaların hızı iyonosferden geçtikleri sırada etkilenmektedir. Bu etkiye bağlı olarak sinyalin uydudan alıcıya iletim zamanındaki değişim TEİ ile doğru orantılıdır. İyonosferin varlığı GPS sözde menzil ölçüm değerlerini artırırken (sinyal gecikmesi nedeniyle), taşıyıcı faz ölçüm değerlerini azaltmaktadır (faz ilerlediği için). Ölçümlerin doğrusal kombinasyonları oluşturularak taşıyıcı faz ve sözde menzil ile TEİ hesaplanabilmektedir. GPS'in iyonosferik toplam elektron içeriğinin kestiriminde kullanılabildiği birçok çalışmada gösterilmiştir (Davies and Hartmann, 1997).

GPS yaygınlığı ve maliyet etkinliği sayesinde diğer ölçüm sistemlerinden öne çıkmaktadır. Özellikle Uluslararası Küresel Seyrüsefer Uydu Sistemi Servisi

¹ 1 TECU = 10¹⁶ elektron/m²

(IGS) bünyesinde bulunan GPS alıcı ağı TEİ kestirimi için kapsamlı veri sunmaktadır. IGS uydu izleme istasyonlarından, veri ve analiz merkezlerinden oluşan uluslararası bir organizasyondur. Barındırdığı yer tabanlı GPS alıcıları sürekli olarak görüş alanları içerisindeki GPS uydularını izlemekte, bu alıcılardan elde edilen genellikle 30 s zaman çözünürlüğündeki veriler yaklaşık 24 sa içinde IGS veri merkezleri tarafından yayınlanmaktadır. IGS ağına dahil izleme istasyonları Şekil 1.2’de görülmektedir. Aynı zamanda IGS bünyesindeki analiz merkezleri



Şekil 1.2: IGS izleme ağındaki istasyonlar (IGS, 2006)

olan Avrupa Yörünge Belirleme Merkezi (CODE), Jet İtke Laboratuvarı (JPL), Avrupa Uzay Ajansı (ESA) ve Katalunya Politeknik Üniversitesi (UPC) analiz merkezleri evrensel iyonosfer haritaları üretmektedirler. Bu haritalar iki saatlik zamansal çözünürlüğe ve uzamsal olarak 2.5° enlem ve 5° boylam çözünürlüğüne sahiptir (Beutler et al., 1999).

TEİ kestirimi konusunda IRI modeli detaylı bilgi sağlayabilmektedir (Bilitza, 2001). Ancak IRI ampirik bir modeldir. GPS verilerini kullanan ölçüme dayalı kestirimler şüphesiz daha iyi kestirimler sağlamaktadır. IGS analiz merkezlerinin sağladığı küresel iyonosfer haritaları bu konuda üretilen en önemli üründür. Bu çalışmada kullanılan Arıkan et al. (2003) tarafından geliştirilen Hacettepe Üniversitesi İyonosfer Araştırma Laboratuvarı (IONOLAB) yöntemi

ise istenilen istasyona ait 30 s zamansal çözünürlüğe sahip gürbüz kestirimler sunarak, IGS analiz merkezlerinden daha detaylı veri sağlamaktadır. Bu yöntemle dünya üzerinde herhangi bir GPS alıcı istasyonu ve herhangi bir zaman dilimi için TEİ değerleri alıcının en alt başucu açısı olan 10° üzerindeki tüm uyduların verileri kullanılarak, ve alıcıların sınırı olan 30 s zaman çözünürlüğünde Düzgünleştirilmiş En Küçük Kareler yöntemi ile hesaplanmaktadır. Bu yöntemin IGS merkezlerinin ve IRI'in kestirimlerine göre daha yüksek zaman ve konum çözünürlüğünde son derece başarılı sonuçlar elde ettiği görülmüştür. Bu yöntem tüm hesaplama yöntemlerinden farklı olarak hiç bir tekniğin ulaşamadığı zaman ve konum çözünürlüğüne ulaşmaktadır.

Yapılan çalışma Arıkan et al. (2003) tarafından geliştirilen IONOLAB yönteminin sağladığı yüksek zaman ve konum çözünürlüklü TEİ kestirimlerini web üzerinden çok geniş bir kitleye kullanıcı dostu bir arayüz üzerinden sunacaktır. GPS verileriyle TEİ kestirimi yapılması ve IONOLAB yöntemi Bölüm 2'de anlatılmaktadır. Geliştirilen uygulama ilk defa yüksek çözünürlüklü gürbüz TEİ kestirimlerini web üzerinden kullanıma sunacaktır. Uygulamanın modüler olması hedeflendiği için tasarımı önem kazanmaktadır. Geliştirilen uygulamanın sağladığı işlevler ve tasarımı Bölüm 3'de detaylı olarak açıklanmaktadır. Uygulamanın gerçekleştirilmesinde güvenilirliği sağlamak amacıyla elde edilen veriler IONOLAB proje grubu tarafından gerçekleştirilen MATLAB tabanlı diğer bir uygulamayla karşılaştırılacaktır. Ayrıca kullanılan TEİ kestirim yönteminin güvenilirliğini ve üstünlüklerini irdeleyebilmek için IRI modeli ve IGS merkezlerine ait kestirimlerle karşılaştırmalar yapılacaktır. Bu analizler Bölüm 4'de verilmiştir. Bölüm 5'de ise elde edilen sonuçlar özetlenmekte ve uygulamayla ilgili gelecekte yapılabilecek çalışmalar irdelenmektedir.

2. TEİ KESTİRİM YÖNTEMİ

Bu bölümde geliştirilen uygulamada kullanılan TEİ kestirim yöntemi anlatılmaktadır. Öncelikle Bölüm 2.2’de GPS ölçüm modelleri hakkında bilgi verilmektedir. Bölüm 2.3’de GPS gözlem verileriyle TEİ kestirimi yapılması anlatılmaktadır. Bölüm 2.4’de ise IONOLAB yöntemi anlatılmaktadır.

2.1 GPS Verileriyle TEİ Kestirimi

GPS sinyalleriyle TEİ değerinin hesaplanmasını etkileyen birçok faktör bulunmaktadır. GPS alıcıları belirli bir anda birçok uydudan sinyal alabilmektedir. GPS alıcısına göre bu uydular farklı yükselme ve yanca açılarına sahiptirler. Fakat literatürdeki birçok kestirim yöntemi iyonosferin uzaysal olarak geniş bir yükselme ve yanca açısı boyunca ve zamansal olarak 5 dk ile 1 sa gibi zaman aralıkları boyunca homojen olduğunu varsaymaktadırlar (Hernandez-Pajares et al., 1999). Bu varsayımlara dayanarak yapılan kestirimlerde sadece belirli uyduların belirli zamanlardaki verileri kullanılmaktadır. Bu nedenle iyonosferdeki değişikliklerle ilgili önemli bilgi içeren birçok GPS sinyal verisi kullanılmamaktadır. Bu nedenle bu çalışmada alıcının görüş alanındaki alıcılarından gelen verileri yüksek zaman çözünürlüğünde değerlendiren bir yöntem kullanılması tercih edilmiştir. Bu kriterlere uyan bir teknik Arıkan et al. (2003) tarafından geliştirilmiştir. Bu teknik 10° yükselme açısından daha yukarıdaki tüm GPS uydularından 30 s zaman çözünürlüğünde alınan sinyalleri birleştirip TEİ kestirimi yapabilmekte ve uydu bazında hesaplanan Dikey Toplam Elektron İçeriği (VTEC) değerlerini uydu pozisyonlarına göre ağırlıklandırarak bu verilerdeki çoklu yol etkilerinden kaynaklanan bozulmaları azaltmaktadır. Artık IONOLAB yöntemi olarak bilinen ve tezin sonraki kısımlarda detaylıca incelenecek olan bu yöntem ile bir günü kapsayan ölçümler kullanılarak günlük kestirimler yapılabilirdiği gibi yöntem ile daha kısa zaman dilimleri için de kestirimler yapılabilmektedir (Arıkan et al., 2004).

2.2 GPS Verileriyle Ölçümler

GPS uyduları göndermekte olduğu mesaj ve kodları iki taşıyıcı frekansla modüle etmektedir. Bu iki taşıyıcı frekanstan $f_1 = 1575.42$ MHz ve $f_2 = 1227.60$ MHz değerindedir. İki taşıyıcı frekans kullanılması sayesinde iyonosferden kaynaklanan gecikmenin sebep olduğu hatalar düzeltilebilmektedir (El-Rabbany, 2002). GPS alıcıları bu hatayı düzeltmek için iki farklı model kullanmaktadır: Sözde menzil modeli ve taşıyıcı faz modeli. Bölüm 2.3’de gösterilen yöntemler bu iki modeli kullanarak TEİ hesaplamasını gerçekleştirmektedir.

2.2.1 Sözde Menzil Modeli

Her uydu belirli bir T^S zamanında sinyal gönderir. Daha sonra alıcı bu sinyali T_γ zamanında alır. Bu durumda sözde menzil :

$$P_\gamma^S = c(T_\gamma - T^S) \quad (2.1)$$

eşitliğiyle ifade edilebilir (Blewitt, 1997). Bu eşitlikte P_γ^S sözde menzili, T^S uydudan işaretin gönderilme zamanını, T_γ alıcının işareti aldığı zamanı ve c boşlukta ışık hızını (299792458 m/s) ifade etmektedir. Alıcı ve uydu yanlılıkları ile iyonosfer ve troposferden kaynaklanan gecikmeler de dikkate alındığında sözde menzil modeli her bir sözde menzil için :

$$P = \rho + d_{ion} + d_{trop} + c(\varepsilon_\gamma + \varepsilon_S) \quad (2.2)$$

eşitliğiyle ifade edilebilir (Wells et al., 1999). Bu eşitlikte P sözde menzili, ρ gerçek menzili, d_{ion} iyonosferden kaynaklanan gecikmeyi, d_{trop} troposferden kaynaklanan gecikmeyi, ε_γ alıcı yanlılığını, ε_S ise uydu yanlılığını ifade etmektedir.

2.2.2 Taşıyıcı Faz Modeli

Taşıyıcı faz modeli her bir faz için :

$$\lambda\Phi = \rho + \lambda\Phi_{trop} - \lambda\Phi_{ion} + c(\varepsilon_\gamma + \varepsilon_S) + N \quad (2.3)$$

eşitliğiyle ifade edilebilir (Nayir, 2007). Bu ifadede λ dalga boyunu, Φ alıcının kaydettiği faz sayısını, Φ_{ion} iyonosferden kaynaklanan faz gecikme sayısını, Φ_{trop}

troposferden kaynaklanan faz gecikme sayısını, N ise ilk faz belirsizliğini ifade etmektedir.

2.3 GPS Verileriyle Toplam Elektron İçeriği Kestirimi

GPS uydusu ve alıcısı arasında kalan 1 m^2 kesitli sütun boyunca bulunan serbest elektron miktarına Toplam Elektron İçeriği denilmektedir. TEİ ölçümlerinde birim olarak TECU kullanılmaktadır. GPS alıcıları gelen işaretlerdeki iyonosferden kaynaklanan gecikmeleri sözde menzile çevirip P_1 ve P_2 değerleri olarak kaydetmektedir. Ayrıca işaretlerin fazında olan gecikmeler de L_1 ve L_2 değerleri olarak kaydedilmektedir. Sadece sözde menzil veya sadece faz gecikmesi değerlerinden TEİ hesaplanması mümkün olduğu gibi her ikisinden de faydalanarak TEİ hesaplanabilmektedir. Sözde menzil değerlerinden TEİ hesaplama daha az hesaplama gerektirirken, daha yüksek gürültü içermektedir. Daha düşük gürültü içeren faz gecikmesinden TEİ hesaplama ise başlangıç faz belirsizliğini belirleme gerekliliği nedeniyle daha fazla hesaplama gerektirmektedir.

2.3.1 Sözde Menzil Ölçümleriyle TEİ Kestirimi

Sözde menzil ölçümlerini kullanarak alıcı ile uydu arasındaki hat üzerinde bulunan toplam elektron miktarı olan Eğik Toplam Elektron İçeriği (STEC) elektron/ m^2 cinsinden:

$$STEC_m(n) = \frac{P_{2,m}(n) - P_{1,m}(n)}{A} \frac{f_1^2 f_2^2}{f_1^2 - f_2^2} + b_m + b \quad (2.4)$$

eşitliği ile ifade edilir (Nayir, 2007). Bu eşitlikte A sabiti $40.3 \text{ m}^3/\text{s}^2$, m uydu numarasını, n ölçüm numarasını ifade etmektedir. $P_{1,m}$ ve $P_{2,m}$ sözde menzil ölçümlerini göstermektedir. b_m ve b ise TECU cinsinden uydu ve alıcı yanlılığını göstermektedir. İyonosferin ince bir kabuk varsayımına dayanarak, hesaplanan STEC değerinin yerel zenite projeksiyonu ile alıcıya göre dik doğrultuda toplam serbest elektron miktarını ifade eden VTEC değeri bulunur (Mannucci et al.,

1993):

$$VTEC_m(n) = \frac{STEC_m(n)}{M(\epsilon_m(n))} \quad (2.5)$$

Bu ifadede $M(\epsilon_m(n))$ ifadesi:

$$M(\epsilon_m(n)) = \left[1 - \left(\frac{R \cos \epsilon_m(n)}{R + h} \right)^2 \right]^{-1/2} \quad (2.6)$$

eşitliği ile gösterilmektedir. Bu ifadede $\epsilon_m(n)$ alıcının koordinat sistemine göre m nolu uydunun n nolu ölçüm sırasındaki yükselme açısını göstermektedir. R Dünya'nın ekvatorial yarıçapı olup 6.378,137 m'dir. h ise iyonosfer yüksekliği olup, en yüksek iyonlaşmanın olduğu yükseklik olarak Schaer (1999) tarafından hesaplanan 428.8 km. değeri kullanılmaktadır.

2.3.2 Faz Ölçümleriyle TEİ Kestirimi

Bölüm 2.3.1'de görüldüğü gibi sözde menzil ölçümleriyle TEİ kestirimi kolaylıkla yapılabilmektedir. Ancak faz ölçümlerindeki ilk belirsizlikten dolayı faz ölçümleriyle TEİ kestirimi daha zordur. İlk faz belirsizliğini gidermek için kullanılan en yaygın yöntem faz ölçümlerinin sözde menzil ölçümleri ile seviyelendirilmesidir. Nayir (2007)'in bahsettiği yöntem faz ile sözde menzil ölçüm farklarını kullanarak faz ölçümlerini seviyelendirmektedir. Bu yöntemde faz ile sözde menzil ölçümleri arasındaki farkın, faz kopmalarının olmadığı bir zaman dilimindeki ortalaması dayanak değeri olarak isimlendirilir. Dayanak değeri

$$B = \frac{1}{N_{me}} \sum_{n_{me}=1}^{N_{me}} (P_{4,m}(n_{me}) - L_{4,m}(n_{me})) \cong -\Delta N \quad (2.7)$$

eşitliği ile ifade edilir. N_{me} faz ölçümlerinin sürekli olduğu zaman dilimini göstermektedir. $P_{4,m}$ ifadesi m uydusuna ait sözde menzil ölçümleri $P_{1,m}$ ve $P_{2,m}$ 'nin arasındaki fark olup

$$P_{4,m} = P_{2,m} - P_{1,m} \quad (2.8)$$

ifadesiyle gösterilebilir. $L_{4,m}$ ise faz ölçümleri $L_{1,m}$ ve $L_{2,m}$ 'nin arasındaki fark olup

$$L_{4,m} = L_{2,m} - L_{1,m} \quad (2.9)$$

eşitliğiyle ifade edilmektedir. Faz ölçümleri gün boyunca sürekli olmadığı için birden fazla dayanak değeri hesaplanmaktadır. Faz ölçümleri, hesaplanan bu dayanak değeri ile seviyelendirilerek STEC değeri hesaplanır. Böylece faz belirsizliği giderilerek hesaplanan STEC değeri

$$STEC_m(n) = \frac{1}{A} \left(\frac{f_1^2 f_2^2}{f_1^2 - f_2^2} \right) (B + L_{4,m}(n) + b_m + b) \quad (2.10)$$

eşitliğiyle ifade edilir. B ifadesi Eşitlik 2.7'de hesaplanan dayanak değeridir. b_m ve b ise TECU cinsinden uydu ve alıcı yanlılığını göstermektedir. Hesaplanan STEC değerinden VTEC değerini hesaplamak için yine Eşitlik 2.5 kullanılmaktadır.

2.4 IONOLAB yöntemi ile TEİ kestiriminin Düzgünleştirilmesi

Bu bölümde Bölüm 2.3'de hesaplanan VTEC değerlerinden TEİ kestirimi yapılması anlatılmaktadır. Anlatılan yöntem 10° yükselme açısından daha yukardaki tüm GPS uydularından 30 s zaman çözünürlüğünde alınan sinyallerden hesaplanan VTEC değerlerini birleştirip düzgünleştirerek TEİ kestirimi yapmaktadır. Bu yöntem VTEC değerlerini uydu pozisyonlarına göre ağırlıklandırarak bu verilerdeki çoklu yol etkilerinden kaynaklanan bozulmaları azaltmaktadır.

2.4.1 Tüm gün için düzgünleştirme yöntemi

Arıkan et al. (2003) tarafından geliştirilen algoritma farklı uydular için hesaplanan VTEC değerlerinin birleştirilmesiyle düzgün bir kestirim yapılmasını sağlamaktadır. Belirli bir zaman aralığı ve m nolu uydu için hesaplanan VTEC değerleri

$$\mathbf{x}_m = [x_m(0) \ \dots \ x_m(n) \ \dots \ x_m(N-1)]_N^T \quad (2.11)$$

ile gösterilebilir. Burada $1 < m < M$ olup, M toplam uydu sayısını göstermektedir. $1 < n < N$ olup, burada N toplam ölçüm sayısını ifade etmektedir.

Arıkan et al. (2003) çoklu yol etkilerini azaltmak için bir ağırlıklandırma fonksiyonu tanımlamıştır. Bu fonksiyon

$$w_m(n) = \begin{cases} 1, & : 60^\circ \leq \epsilon_m(n) \leq 90^\circ \\ \exp(-(60 - \epsilon_m(n))^2/2\sigma^2), & : 10^\circ \leq \epsilon_m(n) < 60^\circ \\ 0, & : \epsilon_m(n) < 10^\circ \end{cases} \quad (2.12)$$

şeklinde tanımlanabilir. Burada σ standart sapmadır. N ölçüm için m nolu uyduya ait ölçümlere uygulanacak ağırlıklandırma vektörü

$$\mathbf{w}_m = [x_m(0) \dots 0 \ w_m(1) \dots w_m(N_m) \ 0 \dots 0]_{N_m}^T \quad (2.13)$$

ifadesiyle gösterilir.

24 saatlik zaman aralığı için hesaplanan VTEC ile kestirilecek VTEC değerleri arasındaki hatanın karesini en aza indiren bir TEİ kestirimi yapabilmek için aşağıdaki maliyet fonksiyonu tanımlanmıştır:

$$J_{\mu, k_c}(\mathbf{x}) = \sum_{m=1}^M (\mathbf{x} - \mathbf{x}_m)^T \mathbf{W}_m (\mathbf{x} - \mathbf{x}_m) + \mu \mathbf{x}^T \mathbf{H}(k_c) \mathbf{x} \quad (2.14)$$

Bu eşitlikte \mathbf{w}_m ağırlıklandırma vektörü, $\mathbf{W}_m = \text{diag}(w_m)$ 'dir. $\mathbf{H}(k_c)$ filtresi, k_c frekansından büyük bileşenleri geçirmektedir. μ ise düzgünleştirme katsayısıdır. Verilen μ ve k_c değerleriyle J 'yi minimize eden \mathbf{x} değerlerini bulmak için $\nabla J_{\mu, k_c}(\mathbf{x}) = 0$ olmalıdır. Bu durumda Eşitlik 2.14 aşağıdaki doğrusal sistemin çözümüne dönüşür:

$$\mathbf{A}(\mu, k_c) \mathbf{x} = \mathbf{b} \quad (2.15)$$

Bu eşitlikte \mathbf{A} ifadesi $N \times N$ bir matris olup

$$\mathbf{A}(\mu, k_c) = \sum_{m=1}^M \mathbf{W}_m + \mu \mathbf{H}(k_c) \quad (2.16)$$

\mathbf{b} ise

$$\mathbf{b} = \sum_{m=1}^M \mathbf{W}_m \mathbf{x}_m \quad (2.17)$$

ifadesiyle gösterilebilir. Buradan \mathbf{x} 'in kestirimi olan $\tilde{\mathbf{x}}$ ise

$$\tilde{\mathbf{x}}(\mu, k_c) = \mathbf{A}^{-1}(\mu, k_c) \mathbf{b} \quad (2.18)$$

dir. $\mathbf{H}(k_c)$ alçak geçiren fonksiyonu $h_n(k_c)$ alçak geçiren filtrelerini kullanarak bir Toeplitz matrisi şeklinde tasarlanabilir:

$$\mathbf{H}(k_c) = \begin{bmatrix} h_0(k_c) & h_1(k_c) & \cdots & h_{N-1}(k_c) \\ h_{N-1}(k_c) & h_0(k_c) & \cdots & h_{N-2}(k_c) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(k_c) & h_2(k_c) & \cdots & h_0(k_c) \end{bmatrix} \quad (2.19)$$

Burada

$$h_n(k_c) = \frac{1}{N} \sum_{k=0}^{N-1} H_k(\omega_c) \exp\left(j \frac{2\pi}{N} kn\right) \quad (2.20)$$

dir. Bu ifadede $\omega_c = 2\pi k_c/N$ 'dir. $H_k(\omega_c)$ ise

$$H_k(\omega_c) = \begin{cases} 1, & \text{eğer } \pi - \omega_c \leq \frac{2\pi}{N}k \leq \pi + \omega_c \text{ ise} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (2.21)$$

şeklinde seçilmiştir. Buradan da $h_n(k_c)$ filtresi

$$h_n(k_c) = \begin{cases} 1 - \frac{1}{N}(2k_c + 1), & n = 0 \\ -\sin\left(\frac{\pi n}{N}(2k_c + 1)\right) / \left(N \sin\left(\frac{\pi n}{N}\right)\right), & n \neq 0 \end{cases} \quad (2.22)$$

şeklini alır. μ ve k_c için optimum değerlerini belirlemek için, kestirilen ile hesaplanan VTEC arasındaki hatayı en aza indirecek hata fonksiyonu

$$e(\mu, k_c) = \sum_{m=1}^M \|\mathbf{W}_m (\tilde{\mathbf{x}} - \mathbf{x}_m)\|^2 \quad (2.23)$$

şeklinde ifade edilebilir. Kestirilen TEİ değerlerinde olabilecek düzensizlikleri gidermek için bir kayan ortanca filtre uygulanacaktır. Uygulanacak ortanca filtresinin en optimum uzunluğunu belirlemek için N_f filtre uzunluğu için kestirilen ve ortanca filtrelenen VTEC değeri arasındaki hata fonksiyonu şudur:

$$e_f(N_f) = \|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{N_f}\|^2 \quad (2.24)$$

Bu eşitlikte $\tilde{\mathbf{x}}$ 'in N_f uzunluğundaki ortanca filtre ile filtrelenmiş hali $\tilde{\mathbf{x}}_{N_f}$ ile ifade edilmiştir.

2.4.2 Belirli bir zaman aralığı için düzgünleştirme yöntemi

Bölüm 2.4.1’de anlatılan algoritma 24 saatlik zaman dilimi için gürbüz kestirimler elde edebilmektedir. Fakat, 24 saatten daha kısa zaman dilimleri için bu yöntem uygun değildir, Çünkü, algoritma 24 saatlik zaman diliminin sahip olduğu dönüştürme simetriye dayanarak uç noktaları aynı değerlere getirmeye çalışmaktadır.

Arıkan et al. (2004)’de aynı yöntemin daha kısa zaman dilimleri için uygulanabilmesi için verideki eğilim doğrusal en az kareler yöntemi ile belirlenmekte, TEİ kestiriminde kullanılan maliyet fonksiyonuna uygulanmaktadır:

$$J_{\mu, k_c} = \sum_{m=1}^M (\mathbf{x} - \mathbf{x}_m)^T \mathbf{W}_m (\mathbf{x} - \mathbf{x}_m) + \mu (\mathbf{x} - a\mathbf{t})^T \mathbf{H}(k_c) (\mathbf{x} - a\mathbf{t}) \quad (2.25)$$

Yukarıdaki eşitlikte \mathbf{t} istenilen zaman dilimine ait zaman vektörüdür. Verideki eğilim a eğimine sahip bir çizgi ile belirlenmektedir. \mathbf{x} değerini minimize edecek bir J fonksiyonu arandığı için $\nabla_{\mathbf{x}} J_{\mu, k_c}(\mathbf{x}) = 0$ ve $\partial J / \partial a = 0$ olmalıdır. Dolayısıyla maliyet fonksiyonunun minimizasyonunun en optimum çözümü şu doğrusal sistemin çözümü ile elde edilir:

$$\mathbf{A}(\mu, k_c) \begin{bmatrix} \mathbf{x} \\ a \end{bmatrix} = \mathbf{b} \quad (2.26)$$

Bu eşitlikte

$$\mathbf{A}(\mu, k_c) = \begin{bmatrix} \sum_{m=1}^M \mathbf{W}_m + \mu \mathbf{H}(k_c) & -\mu \mathbf{H}(k_c) \mathbf{t} \\ \mathbf{t}^T \mathbf{H}(k_c) & -\mathbf{t}^T \mathbf{H}(k_c) \mathbf{t} \end{bmatrix} \quad (2.27)$$

ve

$$\mathbf{b} = \begin{bmatrix} \sum_{m=1}^M \mathbf{W}_m \mathbf{x}_m \\ 0 \end{bmatrix} \quad (2.28)$$

dir. Buradan \mathbf{x} kestirimi olan $\tilde{\mathbf{x}}$

$$\begin{bmatrix} \tilde{\mathbf{x}}(\mu, k_c) \\ a \end{bmatrix} = \mathbf{A}^{-1}(\mu, k_c) \mathbf{b} \quad (2.29)$$

şeklinde ifade edilir. Bu kestirimdeki hatayı en aza indiren μ ve k_c değerleri Eşitlik 2.23’deki hata fonksiyonu ile elde edilir. Tüm gün için olduğu gibi Eşitlik 2.24 ile belirlenen uzunlukta bir ortanca filtre ile $\tilde{\mathbf{x}}$ değerlerindeki düzensizlikler giderilir.

3. UYGULAMA YAZILIMI

Bu bölümde gerçekleştirilen sistemin detaylarına yer verilmiştir. Bölüm 3.1 gerçekleştirilen uygulamanın işlevlerini tarif etmekte, Bölüm 3.2’de ise TEİ kestirim yönteminin gerçekleştirimiyle ilgili detaylara yer verilmiştir. Bölüm 3.6’de de uygulamanın kullanıcı arayüzü detaylı olarak anlatılmaktadır. Bölüm 3.3 mimari seviyesinde tasarımı açıklamakta, Bölüm 3.4 ise uygulama bileşenlerinin detaylarını anlatmaktadır.

3.1 İşlevsel Bakış

Bu bölümde Bölüm 2’de anlatılan TEİ kestirim yönteminin gerçekleştirilmesi için hangi adımların gerekli olduğu anlatılacaktır. Bölüm 1’de değinildiği üzere geliştirilmek istenilen uygulamanın web tabanlı olması hedeflenmiş, arayüzünün kullanıcı dostu olması ve işlemlerin otomatik gerçekleşmesi arzulanmıştır.

Geliştirilen uygulamanın istenilen hesaplamaları gerçekleştirebilmesi için Bölüm 2’deki eşitlikler incelendiğinde aşağıdaki bilgilere ihtiyacı olduğu anlaşılmaktadır:

- Alıcının Sözde Menzil ölçümleri
- Alıcının Faz ölçümleri
- Alıcının yanlılığı
- Uyduların yükseklik açıları
- Uyduların yanlılıkları

80’den fazla ülkeden 200’den fazla enstitünün katkı sağladığı bir kuruluş olan IGS¹ internet üzerinden bu bilgileri sağlamaktadır. IGS’in genel izleme ağı 300’den fazla sabit, sürekli çalışan GPS alıcısını bünyesinde barındırmaktadır. Bu alıcılara ait günlük gözlem verileri IGS’in veri merkezlerinde RINEX

¹<http://igsb.jpl.nasa.gov/>

formatında sunulmaktadır² (Gurtner, 1994). RINEX formatında örnek bir dosya EK-D.1'de verilmiştir. Dosyalar günlük bazda yayınlanmakta olup, 30 s zaman çözünürlüğünde görüş alanındaki uydulardan elde edilen ölçümleri barındırmaktadır. Dosyalarda Sözde Menzil ve Faz ölçümlerinin yanı sıra gözlemi sağlayan istasyonun pozisyonu da bulunmaktadır. Uyduların yükseklik açılarını hesaplayabilmek için uyduların konumları da gerekmektedir. Bu bilgiler de yine IGS veri merkezleri tarafından SP3 formatında sunulmaktadır³ (Remondi, 1989). SP3 formatında örnek bir dosya EK-D.2'de verilmiştir. Ayrıca aylık bazda uydu ve alıcı yanlılıkları da bu merkezler tarafından sunulmaktadır.

Geliştirilen uygulamanın gerekli verileri kullanıcı girdilerine uygun olarak internetten otomatik olarak çekmesi, sonrasında bu verileri kullanarak sözde menzil ve/veya faz ölçümlerine göre Bölüm 2'deki algoritmayı kullanarak TEİ kestirimi yapması gerekmektedir. Elde edilen kestirimin kullanıcı dostu bir arayüz üzerinden sunulması gerekmektedir. Ayrıca IGS analiz merkezlerinin ürettiği iyonosfer haritaları IONEX formatında alınmalıdır⁴ (Schaer et al., 1998). IONEX formatında örnek bir dosya EK-D.3'de verilmiştir. Bu haritalardan kestirim yapılan konuma ait IGS analiz merkezlerinin kestirimleri elde edilip gösterilmelidir. Dolayısıyla gerçekleştirilecek uygulamanın akışı Şekil 3.1'da gösterildiği gibi olmalıdır.

3.2 TEİ Kestirim Yöntemi Gerçekleştirimi

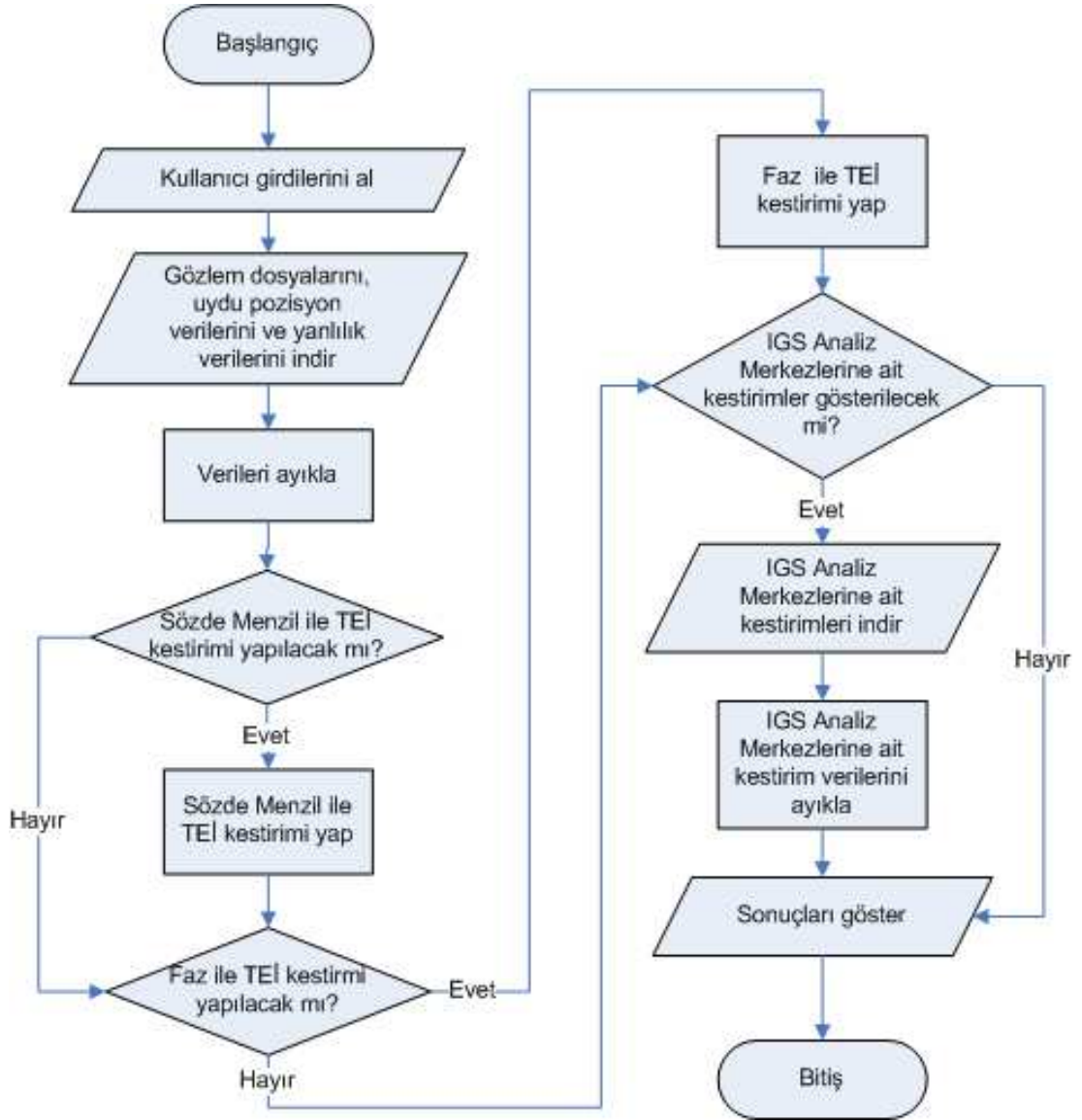
Bu çalışma kapsamında IONOLAB yönteminin gerçekleştiriminde bazı tercihler yapılmıştır. Bu tercihler bu bölümde anlatılmaktadır. Aşağıda TEİ kestiriminin gerçekleştirimiyle ilgili sözde kod verilmiştir:

- İstenilen gün ve istasyona ait CRX formatında gözlem dosyasını çek
- CRX dosyasını crx2rnx programıyla RINEX formatına çevir
- RINEX dosyasını ayıkla

²<http://garner.ucsd.edu/pub/rinex>

³<ftp://ftp.unibe.ch/aiub/CODE>

⁴<ftp://cddis.gsfc.nasa.gov/gps/products/ionex>



Şekil 3.1: Gerçekleştirilen Uygulamamın Akış Şeması

- İstenilen güne ait SP3 formatında uydu konum dosyasını çek
- SP3 dosyasını ayıkla
- Uydu konumlarının zaman çözünürlüğünü doğrusal aradeğerleme ile gözlem çözünürlüğüne çıkar
- İstenilen günün içinde bulunduğu aya ait yanlılık dosyasını çek
- Yanlılık dosyasını ayıkla
- Herbir uydu için
 - Herbir gözlem için
 - * Eşitlik 2.10'e göre STEC değerini hesapla
 - * RINEX dosyasındaki alıcı konumu ve o gözlem anında uydu pozisyonunu kullanarak yükselme açısını hesapla
 - * Eşitlik 2.5'e göre VTEC değerini hesapla
- Tüm VTEC değerlerini -1 ile $+1$ aralığına indirge
- Eşitlik 2.15'deki A ve b değerlerini hesapla
- Eşitlik 2.15'deki doğrusal sistemi çöz
- TEİ kestirimlerini -1 ile $+1$ aralığından normal aralığına getir
- Çözümü kayan ortanca filtre ile işle

Uygulama zaman çözünürlüğü olarak 30 saniyelik gözlem verilerini işleyebilecek şekilde tasarlanmıştır. Ancak bu çözünürlükte veriler kullanıldığında hesaplama işlemi kullanıcı arayüzünün kullanılabilirliğini düşürecek derecede zaman almaktadır. Bu nedenle gözlem verileri beşte bir oranında alt örnekleme tabi tutularak zaman çözünürlüğü 150 saniyeye düşürülmüştür. Bu sayede hesaplama işlemi kabul edilebilir sürede tamamlanmaktadır. Alt örneklemenin yöntemin ürettiği kestirimlere olan etkisi Bölüm 4.5'de incelenmiştir.

Yine TEİ kestiriminin yüksek işlemci gücü harcamasından dolayı, yapılan bir hesaplamanın tekrarlanmaması sağlanmıştır. Bunu sağlamak için elde edilen TEİ kestirim değerleri veritabanında saklanmaktadır. Böylece aynı GPS alıcı konumu ve aynı zaman dilimi için TEİ kestirimi yapılması istenildiğinde hesaplama gerçekleştirilmeden değerler veritabanından çekilmektedir. Bu yapıda TEİ kestirim yöntemi değiştiğinde eski yönteme ait verilerin veritabanından getirilmemesi için uygulama içerisinde kestirim yöntemine bir sürüm numarası verilmiştir. Kestirim yönteminin sürüm numarası değiştiğinde veritabanındaki eski sürümlere ait veriler kullanılmamaktadır.

IGS veri merkezlerinden elde edilen günlük uydu konum bilgileri 15 dakikalık zaman çözünürlüğündedir. Gözlem verileri 30 saniye çözünürlüğünde olduğu için, gözlem anında uydu konumunu bulmak için aradeğerleme yapılması gerekmektedir. Yapılan çalışmada gözlem anında uydu pozisyonunu hesaplamak için doğrusal aradeğerleme kullanılmıştır. Kullanılan doğrusal aradeğerlemeyi

$$p(t) = \frac{t_1 - t}{t_1 - t_0} p_0 + \frac{t - t_0}{t_1 - t_0} p_1 \quad (3.1)$$

ile ifade edebiliriz. Bu eşitlikte p_0 ifadesi t_0 anındaki konumu, p_1 ifadesi t_1 anındaki konumu, p ifadesi ise t_0 ile t_1 arasındaki t anındaki konumu ifade etmektedir.

Eşitlik 2.6'deki ifadede uydunun alıcıya göre yükselme açısı gerekmektedir. Gözlem dosyalarından elde edilen uydu ve alıcılara ait koordinatlar kullanılarak yükselme açısı hesaplanmaktadır. Bu hesaplama için önce aşağıdaki ifade kullanılarak uydu konumu alıcı merkezli koordinat sistemine taşınmaktadır.

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \begin{pmatrix} \cos(\theta) \cos(\phi) & \cos(\theta) \sin(\phi) & -\sin(\theta) \\ -\sin(\phi) & \cos(\phi) & 0 \\ \sin(\theta) \cos(\phi) & \sin(\theta) \sin(\phi) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_s - x_r \\ y_s - y_r \\ z_s - z_r \end{pmatrix} \quad (3.2)$$

Bu ifadede x_s , y_s ve z_s uydunun koordinatlarını x_r , y_r ve z_r ise alıcının koordinatlarını göstermektedir. θ ifadesi

$$\theta = \arccos\left(\frac{z_r}{\sqrt{x_r^2 + y_r^2 + z_r^2}}\right) \quad (3.3)$$

şeklinde gösterilebilir. ϕ ifadesi ise

$$\phi = \arctan\left(\frac{y_r}{x_r}\right) \quad (3.4)$$

olarak gösterilebilir. Buradan yükselme açısı

$$\epsilon = \arctan\left(\frac{z_t}{\sqrt{x_t^2 + y_t^2}}\right) \quad (3.5)$$

olarak hesaplanır.

Bölüm 2.4'de belirtilen düzgünleştirme yönteminin uygulanmasında bazı parametreler gerekmektedir. Bu çalışma kapsamında gerçekleştirilen uygulamanın parametre seçiminde Arıkan et al. (2003)'ün ve Nayir (2007)'in çalışmasından faydalanılmıştır. Bu parametreler için kullanılan değerler şunlardır:

- Eşitlik 2.12'de geçen σ için 0,2817,
- Eşitlik 2.16'de geçen μ için 0,1 ve k_c için 8,0,
- kayan ortanca filtrenin pencere genişliği olarak 17.

Bu parametrelerin yöntemin başarısına olan etkileri Nayir (2007) tarafından detaylı olarak incelenmiştir.

Bölüm 2.4'de düzgünleştirmenin son adımında tek boyutlu kayan ortanca filtre kullanılmaktadır. Tek boyutlu ortanca filtrenin t anındaki çıktısını y ile ifade edersek, bu çıktı t 'ye komşu anların değerlerinin ortancası olarak tanımlanmaktadır:

$$y(t) = \text{ortanca}(x(t - T/2), \dots, x(t), \dots, x(t + T/2)) \quad (3.6)$$

Bu ifadede T ortanca filtrenin pencere genişliğini ifade etmektedir. Bilindiği gibi ortanca, bir değer kümesindeki değerler sıralandığında en ortada bulunan değerdir.

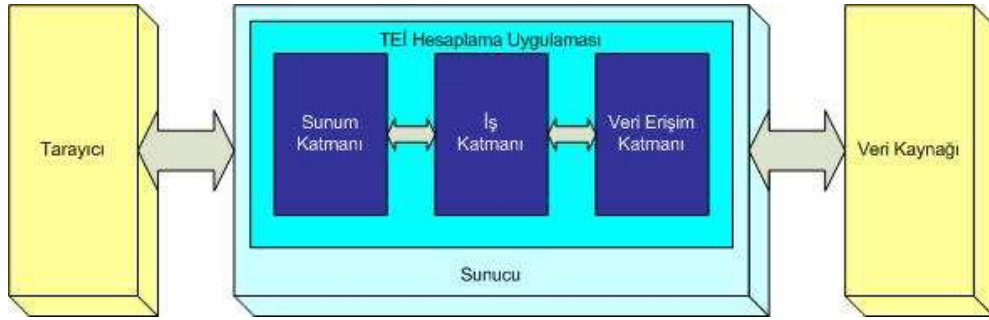
3.3 Mimari Tasarım

Yazılım mimarisi yazılım geliştirme süreçlerinin en önemlilerindedir (McGovern et al., 2003). Mimari tasarımda katmanlara ayırma yazılım tasarımcıları tarafından karmaşık sistemleri parçalara bölmek için yaygın olarak kullanılan bir yöntemdir. Örneğin Uluslararası Standartlaştırma Örgütü (ISO) tarafından geliştirilen Açık Sistemler Arabağlaşımı (OSI) modeli de ağ yapısının karmaşıklığını katmanlara ayırarak bu yaklaşımın faydasını göstermiştir (Tannebaum, 1988).

En yaygın katmanlı mimari üç katmanlı mimaridir. Bu yaklaşım üç katman içerir (Fowler et al., 2002):

- sunum katmanı
- iş katmanı
- veri erişim katmanı.

Bu çalışmada da üç katmanlı mimari kullanılmıştır. Sunum katmanı web tabanlı kullanıcı arayüzünü oluşturmaktadır. İş katmanında ise Toplam Elektronik İçeriği hesaplanmaktadır. Hesaplamaların ihtiyaç duyduğu gözlem verileri ve uydu konumları gibi bilgilere erişim de veri erişim katmanı tarafından sağlanmaktadır. Mimari Şekil 3.2'de gösterilmiştir.



Şekil 3.2: Uygulama Mimarisi

Uygulamanın web tabanlı olması düşünülmüştür. Web tabanlı olması kullanıcı makinesinde kurulum gerektirmemesi nedeniyle, uygulamanın kul-

lanımını ve yeni sürümlerin kurulumunu kolaylaştırmaktadır. Ayrıca uygulamanın daha geniş kitlelere ulaşmasını kolaylaştırmaktadır.

TEİ Hesaplamasının web tabanlı olarak gerçekleştirimi için Java programlama dilinin kullanılması tercih edilmiştir. Java dili aşağıdaki özellikleri nedeniyle tercih edilmiştir (Gosling and McGilton, 1996):

- Java dili nesne tabanlıdır.
- Java dili ile yazılan uygulamalar platformdan bağımsızdır. Hiçbir değişiklik yapmadan aynı program farklı donanım mimarileri üzerinde çalışabilmektedir.
- Bellek yönetimi, Java çalışma ortamı tarafından gerçekleştirildiği için Java ile yazılan programlar daha gürbüzdür.

Web tabanlı uygulamaların katmanlı mimaride Java ile gerçekleştirimi için dağıtık kurumsal uygulamaların geliştirilmesi amacıyla geliştirilen Java Kurumsal Sürümü (JavaEE)⁵ platformu yaygın olarak kullanılmaktadır. Java dilinin ortaya çıkışından günümüze kadar dille ilintili birçok yeni Uygulama Programlama Arayüzü (API) ve standart geliştirilmiştir. Sun şirketi ve endüstrinin önde gelen firmaları özellikle kurumsal uygulamalar ile ilgili olan standartları birleştirerek bu yeni platforma JavaEE adını vermişlerdir (Alur et al., 2003). JavaEE platformu kurumsal uygulamaların gerçekleştirilmesinde şu avantajları sağlamaktadır:

- Veritabanı bağlantılığı, kurumsal iş bileşenleri, mesaj tabanlı arakatman yazılımları, weble ilgili bileşenler, haberleşme protokolleri ve birlikte işlerlik gibi kurumsal uygulamaların ortak ihtiyaçlarıyla ilgili standartlar oluşturmuştur.
- Açık standartların gelişmesine yardımcı olarak teknolojik yatırımlara olan ihtiyacı azaltmaktadır.

⁵<http://java.sun.com/javaee/>

- Uygulama sunucuları arası taşınabilirliği sağlayarak belirli bir uygulama sunucusu satıcısına bağımlılıktan kurtarmaktadır. (JavaEE platformunda uygulamalar genellikle altyapı servislerini sağlayan bir uygulama sunucusu üzerinde çalıştırılmaktadır.)
- Sağladığı altyapı standartları sayesinde uygulama geliştiricilerin altyapı isteklerine değil de kullanıcı isteklerine yoğunlaşmasını sağlamakta ve verimliliklerini arttırmaktadır.
- Katmanlı mimarisi nedeniyle genişletilebilirdir.
- Java teknolojisinin sağladığı taşınabilirlik sayesinde, uygulama farklı işletim sistemleri üzerinde çalıştırılabilmektedir.

Bu avantajlar nedeniyle geliştirilen uygulamanın JavaEE platformunu kullanmasına karar verilmiştir. Aynı zamanda JavaEE platformunun yazılım dünyasında yaygınlığının uygulamanın idamesinde kolaylık sağlayacağı da düşünülmüştür.

JavaEE teknolojisi daha önce belirtildiği gibi bir dizi standardın birleşmesinden oluşmaktadır. Bu standartlardan bir kısmı geliştirilen uygulamanın ihtiyaçlarıyla ilgili değildir. Bu standartlardan geliştirilen uygulamanın katmanlarıyla ilgili olan standartlar şunlardır:

- Sunum katmanı:
 - Servlet⁶ : Servletler web sunucuları üzerinde çalışan Java programcıları şeklinde tarif edilebilir. Servletler bu standardı destekleyen sunucular üzerinde çalışmaktadır. Servletler web kullanıcılarının isteklerini değerlendirip, web tabanlı çıktılar üretebilmektedir (Asbury and Weiner, 1999).
 - Java Sunucu Sayfaları (JSP)⁷ : JavaEE dünyasının sunucu tabanlı betik dilidir. Sunucu tabanlı betikler Yardımlı Metin İşaretleme Dili

⁶<http://java.sun.com/products/servlet/>

⁷<http://java.sun.com/products/jsp/>

(HTML) belgeleri içerisine yerleştirilen kodlardır. Bu kodlar web sunucuları tarafından ayıklanıp dinamik web içeriğine dönüştürülür (Asbury and Weiner, 1999).

- JSP Standart Etiket Kütüphanesi (JSTL)⁸ : JSP standardında betik dilini etiket olarak isimlendirilen yapılarla genişletmek mümkündür. Etiketler karmaşık işlemleri betik dili içerisinde tek bir ifade ile gerçekleştirmeyi sağlamaktadır. JSTL de tüm JSP geliştiricilerinin ihtiyaç duyduğu ortak işlevleri gerçekleştiren etiket kütüphanesidir (Foed, 2004).
- Java Sunucu Yüzleri (JSF)⁹ : Hızlı web tabanlı kullanıcı arayüzlerini geliştirmek için geliştirilmiştir. JSF standardı hazır kullanıcı arayüzü bileşenleri, bir uygulama modeli ve yeni bileşenler geliştirmek için bir bileşen modeli içermektedir (Geary and Horstmann, 2004).

- Uygulama Katmanı:

- Kurumsal Java Bileşenleri (EJB)¹⁰ : Kurumsal uygulamalar için Uzak Metot Çağırımı (RMI) altyapısı üzerine kurulu dağıtık bileşen modelidir. Kurumsal uygulama bileşenlerine birim işlem desteği, sürerlik, güvenlik ve kaynak yönetimi gibi imkanlar sağlamaktadır (Magnusson et al., 1999).

- Veri Katmanı:

- Java Veritabanı Bağlanırlığı (JDBC)¹¹ : Java uygulamaları için Standart Sorgu Dili (SQL) ile veritabanlarına erişim standardıdır (Magnusson et al., 1999). Bu standart JavaEE'de değil, Java Standart Sürümü (JavaSE) içerisinde bulunmaktadır.

⁸<http://java.sun.com/products/jsp/jstl/>

⁹<http://java.sun.com/javaee/javaserverfaces/>

¹⁰<http://java.sun.com/products/ejb/>

¹¹<http://java.sun.com/javase/technologies/database/>

JSF teknolojisi karmaşık web tabanlı kullanıcı arayüzlerinin geliştirilmesine yönelik bir teknolojidir. Geliştirilen uygulamada ekran sayısı düşüktür. Ayrıca bu ekranlar oldukça basit ekranlardır. Bu nedenle uygulamanın geliştirilmesinde JSF standardının kullanılmasına gerek duyulmamıştır. EJB teknolojisinin de yazılım dünyasında edinilen tecrübelerle dayanarak kullanılmamasına karar verilmiştir. Johnson (2003)'ın belirttiği gibi EJB aşağıdaki handikapları beraberinde getirmektedir:

- Uygulamanın test edilmesi zorlaşmaktadır.
- Uygulama kurulumu güçleşmektedir.
- Nesneye Yönelik programlama pratiklerinden sapılmasına neden olmaktadır.
- Tercih edilebilecek uygulama sunucu sayısı azalmaktadır.

JavaEE teknolojisi ara katman yazılımlarının standartlaşması yönünde büyük katkılar sağlamasına rağmen, pratikte fazla karmaşık olması, geliştirme zorlukları ve yetersiz başarımları nedeniyle yeterli başarıyı elde edememiştir. Bu sorunların temeli JavaEE'nin bir programlama modeli olarak algılanmasıdır. Aslında JavaEE platformu kurumsal uygulamalarda kullanılan ara katman yazılımlarının standartlaştırılmasını sağlamaktadır.

En iyi programlama pratiklerini içeren bir uygulama modelinin eksikliği, gerek ticari alanda, gerekse açık kaynak kodlu yazılım alanında arayışların başlamasına neden olmuştur. Bu arayışlar sonucu ortaya çıkan programlama çatılarından Spring¹² en öne çıkan olmuştur. Spring JavaEE teknolojilerinin önemli handikaplarını ortadan kaldırmaktadır. JavaEE API'lerinin kullanımı uygulamayı bu API'lere bağımlı hale getirirken, Spring sunduğu altyapı sayesinde API bağımlılıklarını engellemekte, hatta kendisine bile bağımlı kılmamaktadır. Bir diğer önemli avantajı ise, birim testlerini kolaylaştırması ve özendirmesidir. Zira salt JavaEE kullanılarak geliştirilen uygulamalar için birim

¹²<http://www.springframework.org/>

testleri geliřtirmek neredeyse imkansızdır. Çünkü kullanılan API'ler uygulama sunucusunun varlığını gerektirmektedir. Uygulama sunucularının çalıştırılması ise birim testlerin çevik olmasını engellemektedir (Johnson, 2003). Geliřtirilen uygulamada yukarıda anılan faydalarından dolayı Spring çatısının kullanılmasına karar verilmiřtir.

Spring çatısı Java ile kurumsal uygulama geliřtirmeyi kolaylařtırmak için geliřtirilmiřtir. Spring ile geliřtirilen bileřenler EJB standardında olduđu gibi belirli bir API'ye uymak zorunda olmayıp, Basit Eski Java Nesneleri (POJO)'dirler. Daha basite indirgersek, Spring bir Denetim Çevrimi (IoC) kabıdır. Spring IoC tekniđi sayesinde bileřenlerin gevřek bađlanmasını sađlar. IoC'nin uygulanmasıyla, nesnelerin aktif olarak bađımlı oldukları nesneleri yaratması veya bulması yerine, nesnelere bađımlılıkları pasif olarak verilmektedir. İřte Spring IoC kabı bu bađımlılıkları bileřenlere vermekte, bileřenlerin hayat sürecini yönetmektedir. Spring çatısı bu bađlantıları kendisine verilen Geniřletilebilir İřaretleme Dili (XML) tanım dosyaları ile sađlamaktadır. Geliřtirilen uygulamanın sunum katmanında kullanılan tanım dosyası EK-C'de verilmiřtir. Spring çatısı IoC kabı yanında kurumsal uygulama geliřtirmeyi kolaylařtıran bileřenler de sađlayan modüller içermektedir. İçerdiđi modüller Őekil 3.3'de gösterilmiřtir (Walls and Breidenbach, 2005). Bu çalışmada kullanılan Spring modülleri řunlardır:

- Çekirdek Kabı ve Yardımcıları Modülü: Bu modül Spring'in temelini oluřturmaktadır. Çekirdek kabı IoC tekniđini uygulayarak uygulamanın konfigürasyon ve bađımlılık tanımlarını uygulama kodundan ayırmaktadır. Geliřtirilen uygulamada katmanlar arası bađlantılar Spring IoC kabı tarafından yapılmaktadır.
- Uygulama Bađlamı Modülü: Bu modül IoC kabını bir çatı haline getirmektedir. Ayrıca uygulama iletilerinde çoklu dil desteđi verilebilmesini sađlamaktadır.
- JDBC ve DAO Modülü: JDBC kullanımında bađlantı yaratma, cümle



Şekil 3.3: Spring çatısını oluşturan modüller (Walls and Breidenbach, 2005)

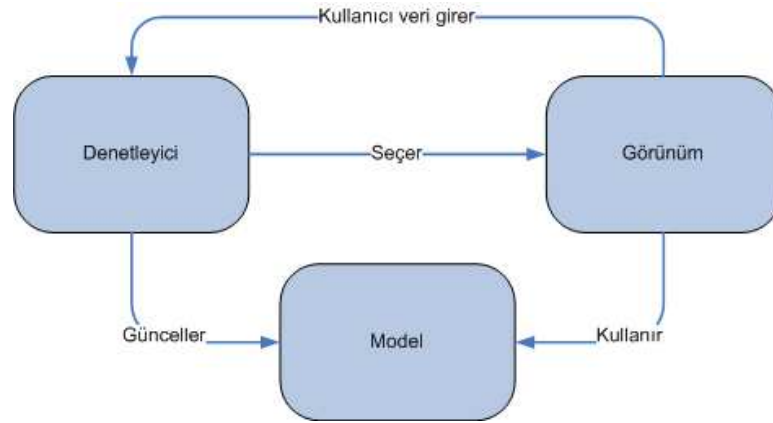
oluşturma, sonuç kümesini işleme ve bağlantıyı kapatma gibi birçok kod yazımı gerekmektedir. JDBC ve DAO modülü bu kodları geliştiriciden soyutlayıp, daha basit, temiz ve sorunsuz veritabanı erişim kodu yazılabilmesini sağlar.

- Web Bağlamı ve Yardımcıları Modülü: Uygulama Bağlamı Modülü üzerine kurulu olan bu modül web uygulamalarına uygun bir bağlam sağlamaktadır.
- Model Görünüm Denetleyici (MVC) Çatısı: Bu modül web uygulamalarına yönelik bir MVC çatısı sağlamaktadır. MVC tasarım kalıbı ilerleyen bölümlerde anlatılmaktadır.

“İlgiye Yönelik Programlama (AOP) modülü” ve “Nesne/İlişkisel Eşleme Modülü” geliştirilen uygulamada kullanılmamaktadır. Uygulamanın web arayüzünün oluşturulmasında Spring’in “Web Bağlamı ve Yardımcıları” ve “MVC Çatısı” modüllerinin yanı sıra JSTL/JSP standartlarından faydalanılmıştır.

Kullanıcı arayüzü ile ilgili en yaygın ve bilinen tasarım kalıbı MVC tasarım kalıbıdır. MVC 1970’lerin sonlarında Smalltalk platformunda geliştirilmiş bir tasarım kalıbıdır. Bu kalıp kullanıcı arayüzü yazılımını üç ana role

ayırır: Model, Görünüm ve Denetleyici. Model görsel olmayan, fakat kullanıcı arayüzünün kullanacağı veri nesnelere. Görünüm modeldeki verileri kullanıcıya gösteren kısımdır. Bu iki kısmı birleştiren parça ise Denetleyici'dir. Denetleyici, kullanıcının girdiği verileri alarak Model'i günceller ve Görünümün de gerekli şekilde güncellenmesini sağlar (Fowler et al., 2002). Tasarım kalıbı Şekil 3.4'de gösterilmiştir. Bu kalıbın kullanılmasının getirdiği en büyük avantaj



Şekil 3.4: MVC Tasarım Kalıbı (Crawford and Kaplan, 2002)

Model ile Görünüm'ün ayrılmasıdır. Bu ayrım sonucunda Model Görünüm'den bağımsız hale gelmektedir. Bu sayede kullanıcı arayüzündeki değişiklikler yani Görünüm'deki değişiklikler Model'in değişmesini gerektirmemekte, uygulamanın esnekliği artmaktadır. Aynı zamanda farklı kullanıcı türlerini destekleyen kullanıcı arayüzlerinin eklenmesi de Model'i etkilememektedir.

Geliştirilen uygulamada bu kalıptan yararlanılarak esnek bir kullanıcı arayüzü sağlanmıştır. Bu kalıbın uygulanmasında Spring'in "MVC Çatısı" ve "Web Bağlamı ve Yardımcıları" modüllerinden faydalanılmıştır. Spring'in "MVC Çatısı" ile gelen bir servlet olan `YönlendiriciServlet`¹³ kullanıcı isteklerini ele almaktadır. Bu servlet web bağlamında tanımlanan eşlemelere uygun olarak ilgili uygulama denetleyici nesnelere kullanılmaktadır. (Johnson et al., 2005). Denetleyici nesnelere iş mantığı nesnelere gerekli bilgileri alıp, "Web Bağlamı ve

¹³Bileşen isimleri uygulamada İngilizce olarak kodlanmış olup, uygulama kodunda kullanılan İngilizce karşılıklar EK-B'de verilmiştir.

Yardımcıları” modülündeki destek nesnelere sayesinde JSP/JSTL kullanarak kullanıcıya web arayüzünü sunmaktadır.

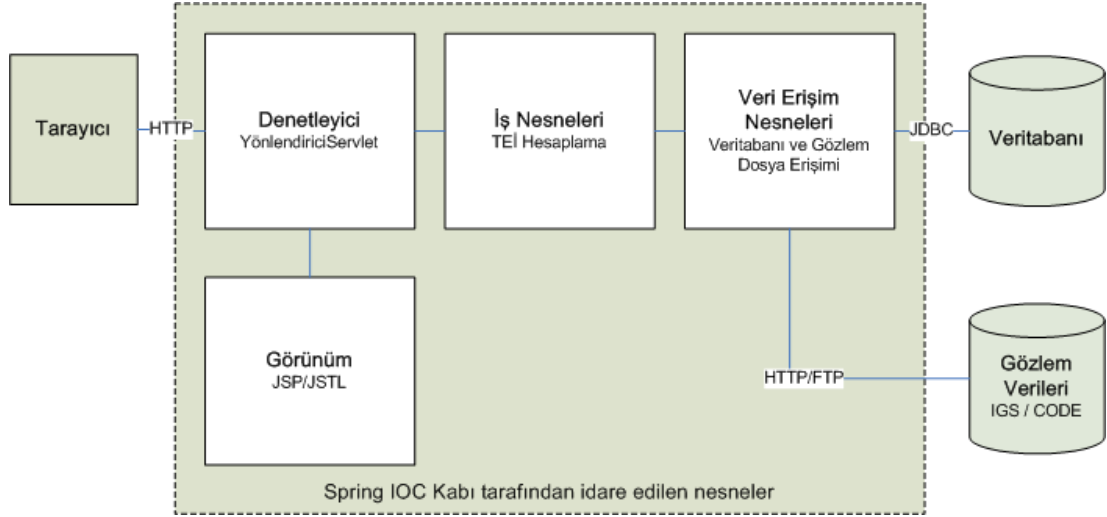
JavaEE tabanlı uygulamalarda veri erişim ve değiştirme işlemlerini bir katmanda birleştirmek için Veri Erişim Nesnesi (DAO) tasarım kalıbı yaygın olarak kullanılmaktadır (Alur et al., 2003). Bu çalışmada da bir veri erişim katmanı tasarladığımız için bu yöntemden faydalanılmıştır. Bu yöntemin kullanımı veri erişim detaylarını diğer katmanlardan saklayıp, bu katmanlara nesneye yönelik bir arayüz sunarak bu katmanı kullanan katmanların işlerinin basitleşmesini sağlamakta, başka veri kaynaklarına geçişi kolaylaştırmaktadır. Böylece veri erişim katmanı modüler hale gelmektedir. Gerçekleştirmede Spring’in “JDBC ve DAO” modülü kullanılarak bu tasarım kalıbının gerçekleştirilmesi sağlanmış, modülün JDBC kullanımını kolaylaştıran sınıfları kullanılarak JDBC API’sinin zorlukları bertaraf edilmiştir (Johnson et al., 2005). İnternet üzerinden erişilen veriler için de aynı kalıp kullanılarak, ağ iletişim kodunun detayları iş mantığı katmanından, sağlanan bir arayüz ile izole edilmiştir. Böylece verilere erişim yöntemindeki değişikliklerden iş mantığı katmanının etkilenmemesi sağlanarak uygulamanın veri erişim katmanı esnek hale getirilmiştir.

Uygulama katmanlarının biraraya getirilmesi de önem arz etmektedir. Bu noktada da Spring çatısından faydalanılmıştır. Spring çatısının temel bileşeni olan IoC kabı ile uygulama bileşenleri birbirine bağlanmıştır. IoC yaklaşımı sayesinde bileşenler arası kontratlar daha net hale gelmekte ve bileşenlerin test edilebilirliği artmaktadır (Fowler, 2004).

3.4 Detaylı Tasarım

Uygulamanın tasarımı Bölüm 3.3’de belirtildiği gibi katmanlı bir yapıdan oluşmaktadır. Mimarinin detayı Şekil 3.5’de gösterilmiştir. Katmanlı mimari sayesinde sunumla ilgili nesnelere iş nesnelere ayrılmış, bu sayede ileride farklı kullanıcı arayüz teknolojilerinin de kullanımına imkan sağlanmıştır. Ayrıca katmanlar arası bağlantılar arayüzler kullanılarak yapıldığı için katmanların içerisindeki değişiklikler diğer katmanları etkilememektedir. Bu yapı sayesinde

veri depolama ve erişimde olabilecek bir değişiklik, iş mantığının değişmesini gerektirmemektedir. Her katmanla ilgili gerçekleştirmeler belirli Java paket-



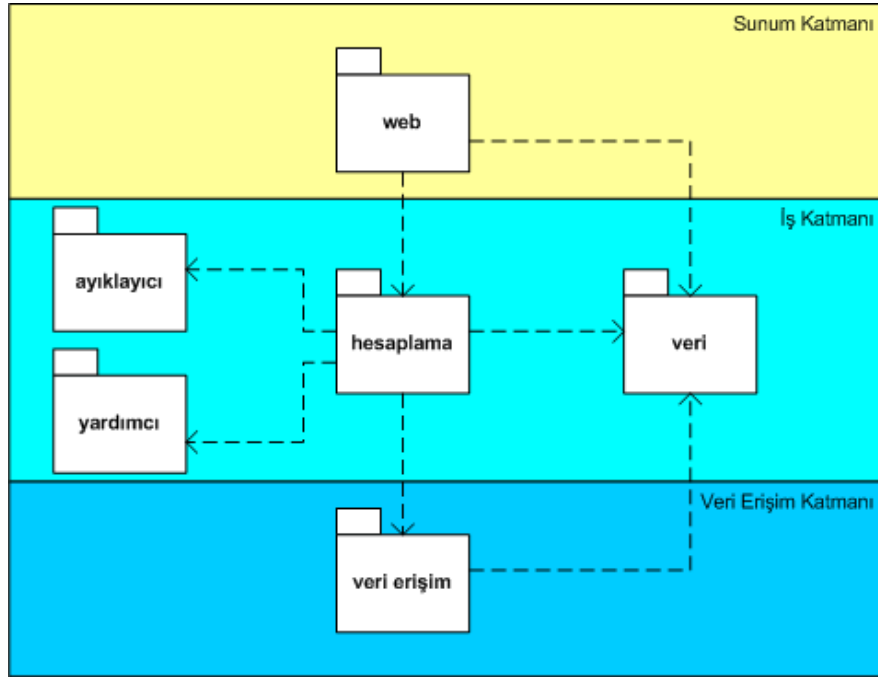
Şekil 3.5: Uygulama mimarisi detayı

lerinde toplanmıştır. Uygulamayı oluşturan paketler Şekil 3.6¹⁴'de gösterilmiştir. İş mantığı katmanını hesaplama ve ayıklayıcı paketleri oluşturmaktadır. yardımcı paketi iş mantığı katmanı tarafından kullanılan yardımcı bileşenleri içermektedir. Veri erişim katmanını veri erişim paketi oluşturmaktadır. Sunum katmanını ise web paketi oluşturmaktadır. veri paketi katmanlar arasında veri transferi için kullanılan bileşenleri içermektedir. Paketlerin detayları alt bölümlerde ve EK-C'de anlatılmaktadır. Uygulamanın gerçekleştirilmesinde sınıf isimlendirmesi İngilizce olarak yapılmıştır. Ancak tasarımın daha anlaşılır olması için bu bölümde sınıf isimlerinin Türkçe karşılıkları kullanılmıştır. Sınıf isimlerinin İngilizce asılları Bölüm B'de verilmiştir.

3.4.1 Sunum Katmanı

Sunum katmanı Bölüm 3.3'de belirtildiği gibi MVC tasarım kalıbı uygulanarak tasarlanmıştır. Bunu sağlamak için Spring'in "MVC Çatısı"

¹⁴Kullanılan tasarım gösterim sistemi EK-A'da anlatılmıştır.

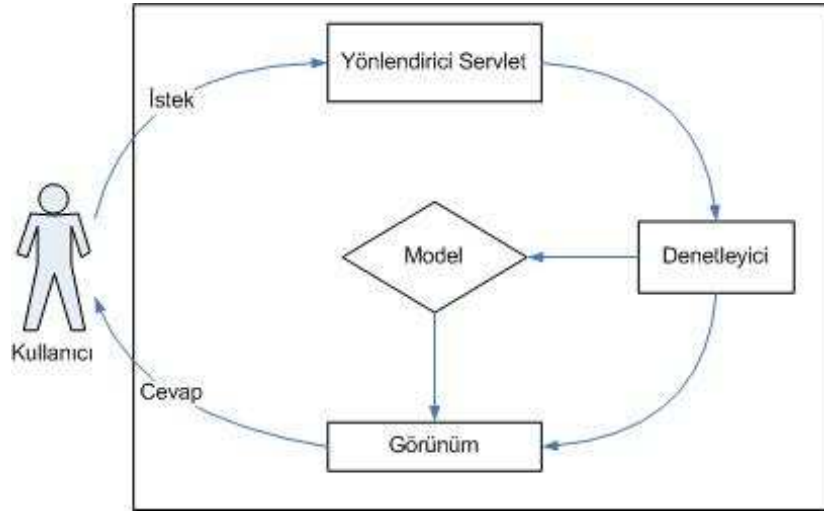


Şekil 3.6: Uygulamayı oluşturan paketler

modülünden faydalanılmıştır. MVC'nin görünüm ayağı JSP/JSTL teknolojileri kullanılarak gerçekleştirilmiştir. Şekil 3.7'de görüldüğü gibi Spring'in "MVC Çatısı" modülünde kullanıcı isteklerini `YönlendiriciServlet` değerlendirip ilgili denetleyiciye yönlendirmektedir. Sunum katmanının denetleyicileri Spring'in "MVC Çatısı" modülünün `Denetleyici` arayüzünü gerçekleştiren sınıflardır. Uygulamamızda 3 adet denetleyici bulunmaktadır. Bunlar `TekIstasyonDenetleyicisi`, `TekIstasyonGünKarşılaştırmaDenetleyicisi` ve `ÇokIstasyonDenetleyicisi`'dir. `TekIstasyonDenetleyicisi` tek istasyona ait TEİ kestirimi için, `TekIstasyonGünKarşılaştırmaDenetleyicisi` tek istasyona ait iki farklı günün TEİ kestirimlerinin karşılaştırılması için, `ÇokIstasyonDenetleyicisi` ise aynı gün için farklı istasyonların TEİ kestirimlerinin karşılaştırılması için geliştirilmiştir.

Sonuçların grafik olarak gösterilebilmesi için açık kaynak grafik üretim kütüphanesi `JFreechart`¹⁵ kullanılmıştır. `JFreechart` ile üretilen grafiklerin web

¹⁵<http://www.jfree.org/jfreechart/>



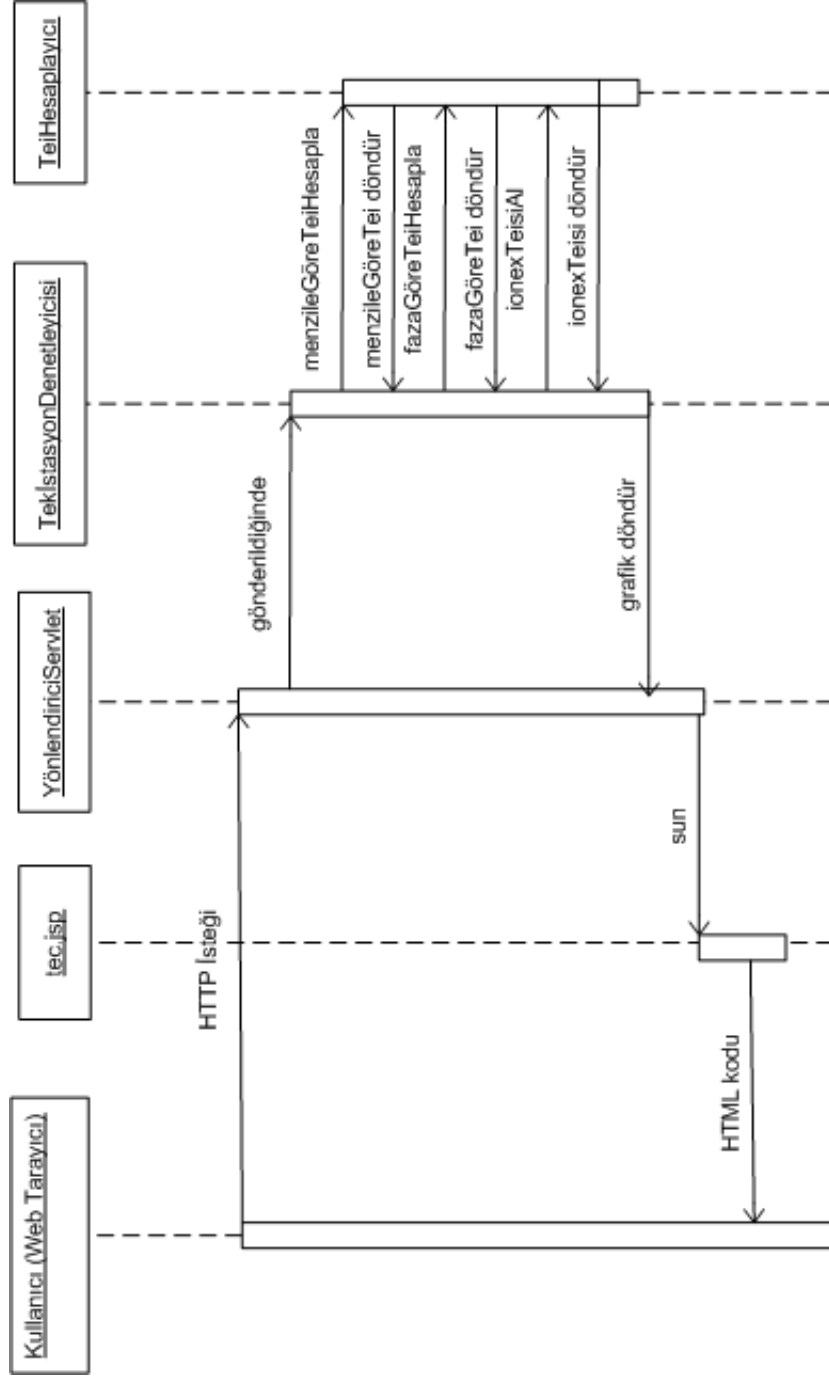
Şekil 3.7: Spring MVC Yapısı

sayfasında gösterimi için ise Cewolf¹⁶ kütüphanesi kullanılmıştır. JFreechart kütüphanesi, girdi olarak sağlanan veriler için farklı türlerde grafikler üretip bu grafikleri farklı formatlarda sunabilmektedir. Cewolf kütüphanesi ise bu grafiklerin bir servlet aracılığıyla web sayfalarında gösterilebilmesini sağlamaktadır, yani JFreechart kütüphanesine web desteği sağlamaktadır.

Sonuçların aynı zamanda tabüer olarak gösterilebilmesi için Excel desteği eklenmiştir. Excel çıktısı Spring'in "Web Bağlamı ve Yardımcıları" modülündeki `SoyutExcelGörünümü` sınıfından türetilerek geliştirilen `TeiExcelGörünümü` sınıfı ile gerçekleştirilmiştir.

Daha önce belirtildiği gibi uygulamada üç denetleyici bulunmaktadır. Bu denetleyicilere ait senaryolar birbirine benzer olduğu için, burada sadece `TekIstasyonDenetleyicisi`'ne ait senaryo tarif edilmiştir. Tek istasyon için sözde menzil verileri ile TEİ kestiriminin yapılması senaryosu için sıralama şeması Şekil 3.8'de gösterilmiştir. Kullanıcı, istasyonu ve tarih aralığını seçip "Hesapla" butonuna basarak tarayıcı aracılığıyla isteğini gönderdiğinde istek Spring'in `YönlendiriciServlet`'ine gelmekte, bu servlet web bağlam tanımlarının bulunduğu `webtec-servlet.xml` dosyasındaki eşlemeye

¹⁶<http://cewolf.sourceforge.net/>



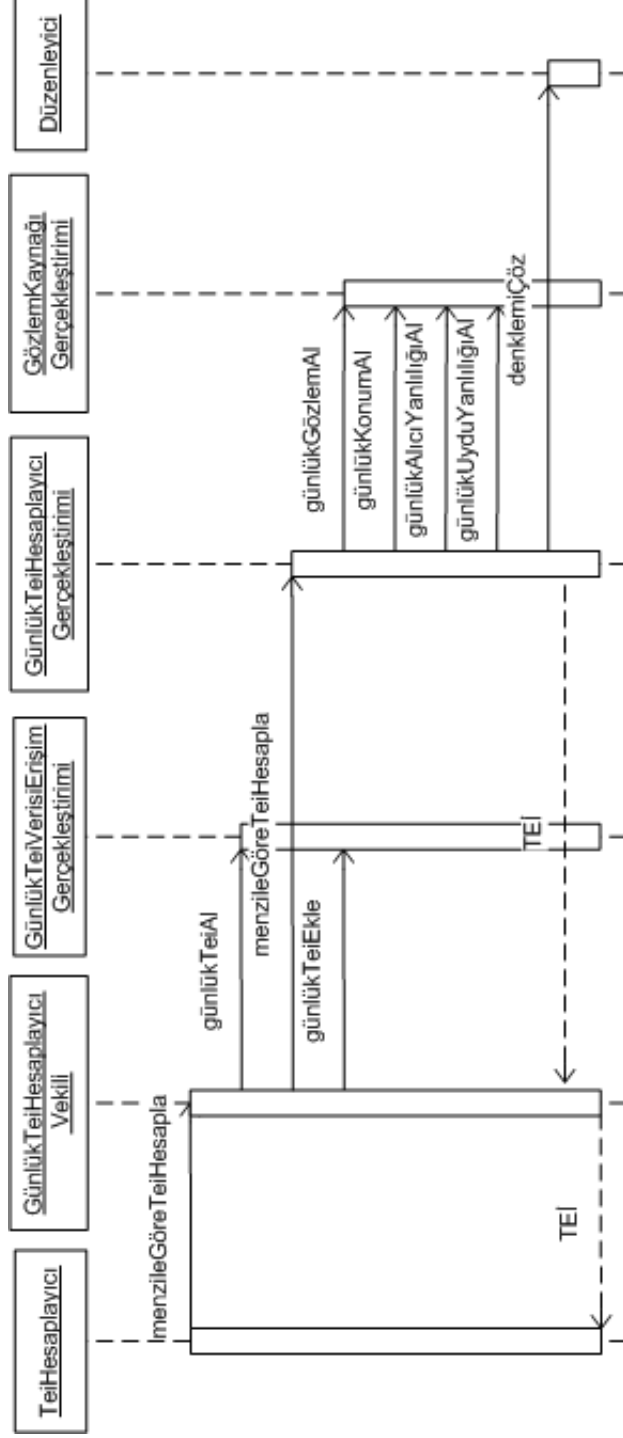
Şekil 3.8: Web katmanının tek istasyon için sözde menzül verileri ile TEİ kestirimi yapılması senaryosuna ait sıralama şeması

göre istekle eşlenmiş denetleyiciye (`TekIstasyonDenetleyicisi`) isteği iletmektedir. Denetleyici modele (`TeiHesaplayıcı`) ulaşarak gerekli bilgileri elde etmekte, daha sonra yine web bağlam tanımlarında belirtilmiş görünüme (`tec.jsp`) yönlendirmektedir. Görünüm ise denetleyici sınıfın kendisine ilettiği modele ait bilgileri göstermektedir.

3.4.2 İş Katmanı

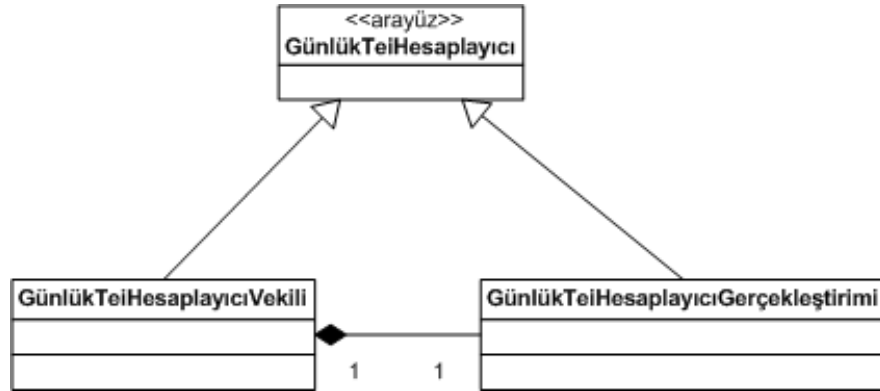
İş katmanındaki nesnelerin tek istasyon için sözde menzil verileri ile TEİ kestirimi yapılması senaryosu için davranışları Şekil 3.9'deki sıralama şemasında gösterilmiştir. Burada `GünlükTeiVerisiErişimGerçekleştirimi` nesnesi daha önce hesaplanan verileri veritabanında saklayan ve veritabanından getiren sınıftır. Bu sınıf sayesinde her seferinde aynı hesaplamalar yapılmamaktadır. Yapılan hesaplamaların sonuçları veritabanına kaydedilmekte, daha sonra tekrar istenildiğinde sonuçlar hesaplanmadan veritabanından getirilmektedir. `GünlükTeiHesaplayıcıVekili` nesnesi istenilen TEİ kestiriminin daha önce hesaplanıp hesaplanmadığını kontrol edip gerekiyorsa `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesine hesaplattırır. `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesi bu hesaplamayı yapabilmek için gerekli verilere `GünlükGözlemKaynağıGerçekleştirimi` nesnesi aracılığıyla ulaşır ve hesaplanan değerlerin düzgülleştirilmesini de `Düzenleyici` nesnesine yaptırır.

TEİ kestirim gerçekleştirimi ile ilgili arayüz ve sınıflar Şekil 3.10'de görülmektedir. Daha önce yapılan hesaplamaların kaydedilmesi işlevi uygulamanın arayüzler kullanılarak tasarlanması sayesinde kolaylıkla eklenmiştir. Hesaplamayı yapan `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesi `GünlükTeiHesaplayıcı` arayüzünü gerçeklemektedir. Uygulamanın ilk aşamalarında `TeiHesaplayıcı` nesnesi hesaplama için `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesini kullanırken, daha sonra veritabanına kayıt işlevi eklendiğinde `TeiHesaplayıcı` nesnesi bu nesnede işlevsel bir değişiklik yapılmadan yine `GünlükTeiHesaplayıcı`



Şekil 3.9: İş katmanı sınıflarının tek istasyon için sözde menzil verileri ile TEI kestirimi yapılması senaryosu için sıralama şeması

arayüzünü gerçekleyen `GünlükTeiHesaplayıcıVekili` nesnesini kullanmaya başlamıştır. `GünlükTeiHesaplayıcıVekili` nesnesi `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesinin vekili şeklinde davranıp hesaplama gerektiğinde bu işi `GünlükTeiHesaplayıcıGerçekleştirimi` nesnesine yaptırmaktadır. Bu tasarım yöntemi uygulamaya yeni TEİ kestirim yöntemlerinin eklenmesinin de kolaylıkla yapılabileceğini göstermektedir.

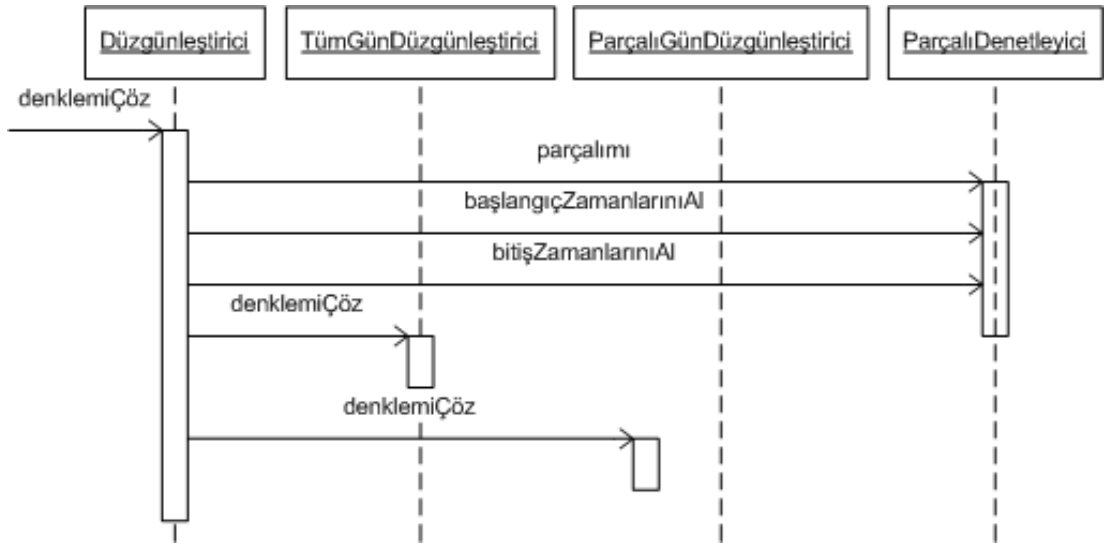


Şekil 3.10: TEİ kestirim gerçekleştirimi ile ilgili arayüz ve sınıfları içeren sınıf şeması

Düzenleştirici nesnesi düzenleştirme işlemi öncesi işlenen güne ait verilerin içerisinde eksik veriler bulunup bulunmadığını `ParçalıDenetleyici` nesnesini kullanarak belirler. Eğer veriler parçalı değilse Bölüm 2.4.1’de bahsedilen yöntemi uygulayan `TümGünDüzenleştirici` nesnesine düzenleştirme işlemi yaptırır. Eğer veriler parçalı ise Bölüm 2.4.2’de bahsedilen yöntemi uygulayan `ParçalıGünDüzenleştirici` nesnesine düzenleştirme işlemi yaptırır. Bu işlemle ilgili sıralama şeması Şekil 3.11’de görülmektedir.

3.4.3 Veri Erişim Katmanı

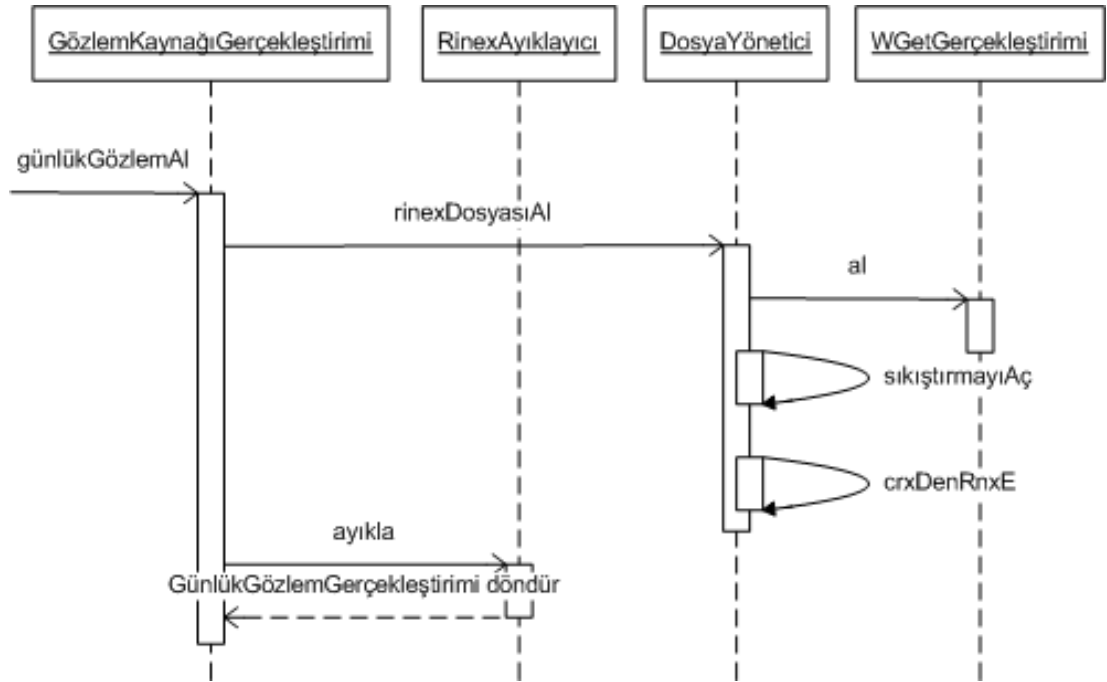
Internet verilerine ulaşmak için kullanılan `GözlemKaynağıGerçekleştirimi` nesnesinin GPS gözlem verilerinin alınması senaryosuna ait sıralama şeması Şekil 3.12’de gösterilmiştir. DAO tasarım kalıbı `GözlemKaynağıGerçekleştirimi` nesnesi tarafından uygulanarak,



Şekil 3.11: Düzgünleştirme işlemi gerçekleştirimi ile ilgili sıralama şeması

gözlem verilerine erişim için basit bir arayüz oluşturulmuştur. DAO tasarım kalıbı veri erişim işlevleriyle ilgili detayları bir katman içerisine koymak şeklinde ifade edilebilir. Bu kalıp sayesinde veri erişim yönteminde şeffaflık sağlanır. Böylece farklı veri erişim yöntemlerinin uygulamaya entegre edilmesi kolaylaşmaktadır. Kullanıcılara daha basit ve nesneye yönelik bir arayüz sağlanmaktadır. GözlemKaynağıGerçekleştirimi nesnesi gözlem verisini elde etmek için FileManager nesnesinden dosyayı istemektedir. DosyaYönetici nesnesi yerel diskte dosya varsa bu dosyayı vermekte, diskte yoksa önce dosyayı internetten WGetGerçekleştirimi nesnesini kullanarak indirip vermektedir. GözlemKaynağıGerçekleştirimi nesnesi aldığı dosyadan ilgili ayıklayıcıyı (RinexAyıklayıcı) kullanarak gözlem verilerini elde etmektedir.

DAO tasarım kalıbı gözlem verilerine erişim için kullanılmasının yanında hesaplanan kestirimlerin veritabanında saklanmasında da kullanılmıştır. GünlükTeiVerisiErişimGerçekleştirimi nesnesi hesaplanan verilerle ilgili veritabanı erişimi için DAO tasarım kalıbını gerçeklemektedir. Bu nesne, hesaplanan TEİ verilerine erişim için gerekli veritabanı erişim detaylarını saklayarak diğer nesnelere basit bir veri erişim arayüzü sağlamıştır. Kalıbın gerçekleşmesinde



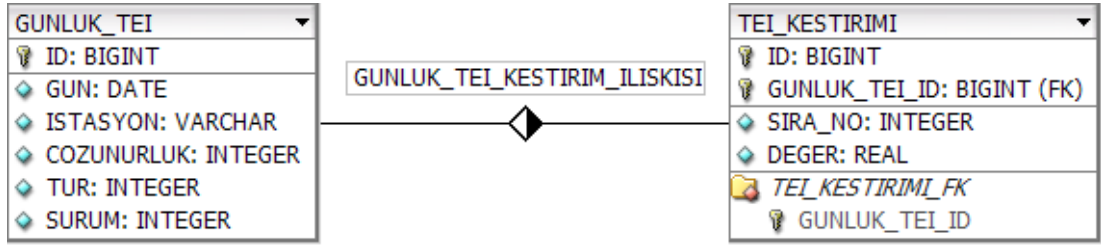
Şekil 3.12: GözlemKaynağıGerçekleştirimi sınıfının GPS gözlem verilerinin alınması senaryosuna ait sıralama şeması

Spring'in "JDBC ve DAO" modülü kullanılmıştır. Bunun yanında modülün sağladığı JDBC ile ilgili yardımcı bileşenler sayesinde JDBC işlemlerinin zorlukları aşılmıştır.

3.5 Veritabanı Tasarımı

Geliştirilen uygulama hesaplama işleminin oluşturduğu işlemci yükünü hafifletmek için hesaplanan TEİ verilerini veritabanında saklamaktadır. Böylece aynı istasyon ve zaman dilimine ait TEİ kestiriminin tekrar hesaplanması istenildiğinde, hesaplama yapılmadan kestirim verileri veritabanından getirilmektedir. TEİ verilerini saklamak için geliştirilen veritabanı modeli Şekil 3.13'de görülmektedir. Model iki tablo içermektedir: GUNLUK.TEI ve TEI.KESTIRIMI.

GUNLUK.TEI tablosunda her benzersiz istasyon ve tarih ikilisi için bir kayıt oluşturulmaktadır. ID sütunu birincil anahtardır. GUN sütunu kestirimin ait olduğu günü belirtmektedir. ISTASYON sütunu ise kestirimin ait olduğu istas-



Şekil 3.13: Gerçekleştirilen uygulamaya ait veritabanı tasarımı

yonu göstermektedir. COZUNURLUK sütununda gözlem verilerine dolayısıyla kestirimlere ait zaman çözünürlüğü tutulmaktadır. Bu sütunun içerdiği değer kullanılan alt örnekleme sonucu genellikle 150 saniyedir. TUR sütununda kestirimin sözde menzil gözlemleriyle mi faz gözlemleriyle mi yapıldığı bilgisi tutulmaktadır. SURUM sütununda da kestirimi yapan algoritmanın sürüm bilgisi tutulmaktadır. Bu sütun sayesinde algortima değişikliğinde eski verilerin kullanılmasının önüne geçilmektedir.

TEI_KESTIRIMI tablosu ise kestirim değerlerini saklamaktadır. GUNLUK_TEI tablosundaki herbir kayıtla ilgili olarak bu tabloda o gün için hesaplanan kestirim kayıtları bulunmaktadır (150 s zaman çözünürlüğü için 576 kayıt). SIRA_NO sütunu kestirimin gün içindeki sırasını belirtmektedir. DEGER sütunu ise TECU cinsinden kestirim değerini saklamaktadır. GUNLUK_TEI_ID sütunu ise TEİ kestirim kaydının ait olduğu GUNLUK_TEI tablosudaki günlük kestirim kaydının birincil anahtar değerini içermektedir. Bu sütun üzerinde GUNLUK_TEI tablosuyla ilgili bir yabancı anahtar kısıtı tanımlanmıştır.

3.6 Kullanıcı Arayüzü

Gerçekleştirilen uygulama bir web uygulama sunucusu üzerinde çalışmaktadır. Kullanıcılar uygulamayı web tarayıcıları ile erişip kullanabilmektedirler. Uygulama kullanıcı arayüzünün giriş ekran görüntüsü Şekil 3.14’de gösterilmiştir. Uygulama çoklu dil desteğine sahip olup bu çalışma kapsamında

Şekil 3.14: Uygulama kullanıcı arayüzü

Türkçe ve İngilizce dil desteği eklenmiştir. Şekil 3.14’de gösterilen giriş ekranın üst tarafında bulunan menü kısmından dil değişikliği yapılabilmektedir. Yeni dil desteği eklenmesi için uygulama kaynak kodlarının değiştirilmesine gerek yoktur. Bu sayede yeni bir dil desteği eklenmesi kolaylaştırılmıştır.

Kullanıcı arayüzü ekranın üst kısmındaki menü ve alt kısımdaki içerik bölümlerinden oluşmaktadır. Menü uygulamanın sağladığı üç temel TEİ analiz işlevine ulaşmayı sağlayan bağlantıları içermektedir: Tek İstasyon, Çok İstasyon ve Tek İstasyon-Gün Karşılaştırma. *Tek İstasyon* bağlantısı ile seçilen bir istasyon ve belirli bir tarih aralığı için TEİ kestirimi elde edilebilmektedir. *Çok İstasyon* bağlantısı birden fazla istasyonun aynı zaman aralığı içerisindeki TEİ kestirimlerinin karşılaştırılabilmesini sağlamaktadır. *Tek İstasyon-Gün Karşılaştırma* bağlantısı ise seçilen bir istasyonun farklı iki gün için TEİ kesti-

riminin karşılaştırılabilmesini mümkün kılmaktadır. Menüden herhangi bir işlem seçildiğinde, o işlev gerçekleştirilebilmesi için gerekli parametrelerin girileceği Şekil 3.15’de gösterilen form ekrana gelmektedir. Hesaplamaların bütününde

Tek İstasyon için TEİ

İstasyon Kodu

Gözlem Başlangıç Tarihi

Gözlem Bitiş Tarihi

Sözde Menzil verilerini kullan

Faz verilerini kullan

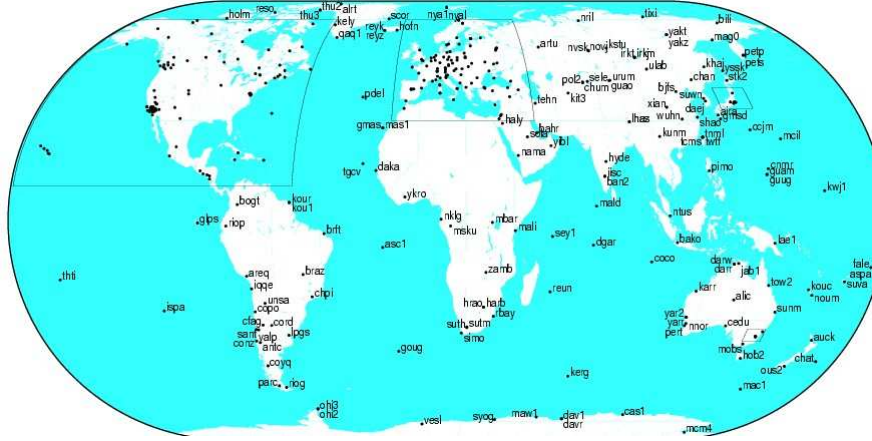
IONEX verilerini göster

Çıktı türü Grafik Excel

Şekil 3.15: Tek istasyon için TEİ Kestirimi veri giriş ekranı

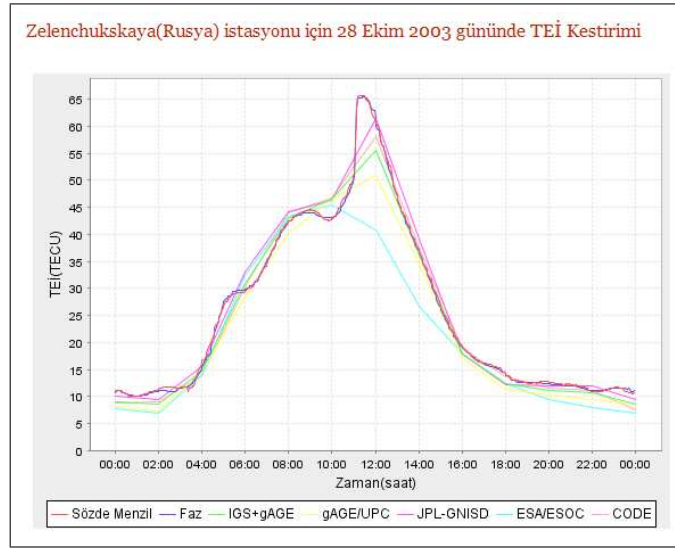
sözde menzil ve/veya faz verileri kullanılarak kestirim yapılabilmektedir. Bunun için Şekil 3.15’de görülen *Sözde Menzil verilerini kullan* ve/veya *Faz verilerini kullan* seçeneklerinin seçilmesi gerekmektedir. Ayrıca IGS merkezleri tarafından yapılan TEİ kestirimleri de gösterilebilmektedir. IGS merkezlerinin kestirimlerini görebilmek için Şekil 3.15’de görülen *IONEX verilerini göster* seçeneğinin seçilmesi gerekmektedir. Bu formda örnek olarak Zelenchukskaya istasyonu için 28 Ekim 2003 tarihi için sözde menzil ile, faz ile TEİ kestirimi ve IONEX verilerinden IGS merkezleri tarafından hesaplanan TEİ kestirimlerinin gösterilmesi için gerekli değerler girilmiştir. Çıktı türü olarak grafik çıktı ve Excel çıktısı desteklenmektedir. Şekil 3.15’de çıktı türü olarak grafik seçilmiştir.

Formun doldurulmasında istasyon seçimi ile ilgili kolaylık sağlayan pencere bu alanın yanında bulunan ikona tıklandığında açılmaktadır. İstasyon seçimi için kullanılan arayüzler Şekil 3.16’de gösterilmiştir. Haritalarda IGS ağı içerisindeki GPS istasyonları gösterilmektedir. Bu harita üzerinden seçilen GPS istasyonuna ait gözlem verileri uygulama tarafından kullanılmaktadır. İlgili form doldurulduktan sonra *Hesapla* butonuna tıklanır. Forma girilen değerler geçerli ise sonuç grafik olarak Şekil 3.17’de gösterildiği gibi ekrana getirilmektedir. Bu



Şekil 3.16: İstasyon seçim ekranı (IGS, 2006)

grafik Şekil 3.15’de gösterilen girdiler sonucu elde edilen grafikdir. Grafikte sözde



Şekil 3.17: Tek istasyon için TEİ kestirim grafiği

menzil verileriyle hesaplanan TEİ kestirimi, faz verileriyle hesaplanan TEİ kestirimi ve IGS analiz merkezlerinin evrensel iyonosfer haritalarından elde edilen kestirimler gösterilmektedir.

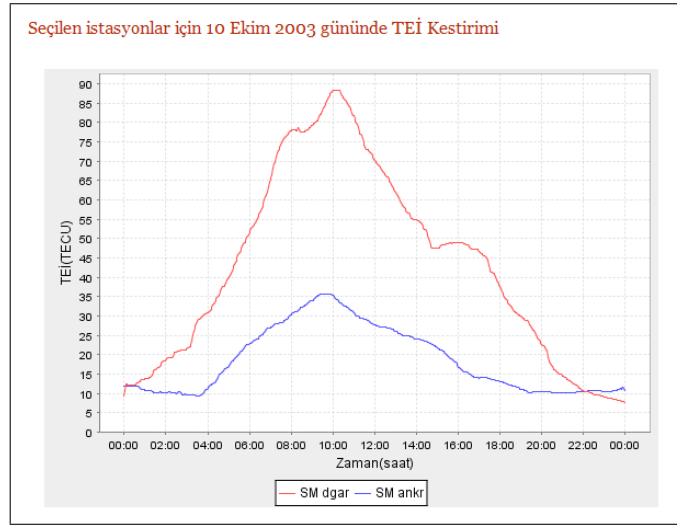
Uygulamanın sağladığı diğer bir işlev de birden fazla istasyon için TEİ kestirimlerinin karşılaştırılabilmesidir. Bu işlevi kullanabilmek için kullanıcı

arayüzünün üst tarafındaki *Çok İstasyon* bağlantısı seçilmelidir. Bu bağlantı seçildiğinde Şekil 3.18'de gösterilen ekran açılmaktadır. Bu ekranda istasyon

Şekil 3.18: Çok İstasyon için TEİ Kestirimi veri giriş ekranı

seçim ikonuna tıkladığında açılan Şekil 3.16'de görülen istasyon seçim ekranı kullanılarak birden fazla istasyon seçilir. Şekil 3.18'de Diego Garcia adası ve Ankara istasyonları seçilmiştir. Diego Garcia adası ekvatorial bölgede, Ankara ise orta enlem bölgesindedir. Başlangıç ve bitiş tarihi olarak da iyonosferin sakin olduğu bir gün olan 10 Ekim 2003 seçilmiştir. TEİ kestiriminin sözde menzil verileriyle hesaplanması seçeneği seçilmiştir. Bu girdilere göre hesaplanan TEİ kestirimine ait grafik Şekil 3.19'de görülmektedir. Komjathy (1997)'nin belirttiği gibi ekvatorial bölge elektron yoğunluğunun en yüksek olduğu bölge iken orta enlem bölgesi iyonosferin en sakin bölgesidir. Bu durum uygulama tarafından üretilen grafikte de görülebilmektedir.

Uygulamanın sağladığı üçüncü işlev ise belirli bir istasyon için farklı iki güne ait TEİ kestirimlerinin karşılaştırılabilmesidir. Bu işlevi kullanabilmek için kullanıcı arayüzünün üst tarafındaki *Tek İstasyon-Gün Karşılaştırma* bağlantısı seçilmelidir. Bu bağlantı seçildiğinde Şekil 3.20'de gösterilen ekran açılmaktadır. Bu ekranda istasyon seçme ikonuna tıkladığında açılan Şekil 3.16'de görülen istasyon seçim ekranı kullanılarak istasyon seçilir. Şekil 3.20'de Ankara istasyonu seçilmiştir. Karşılaştırılacak günler de 10 Ekim 2003 ve 28 Ekim 2003 olarak



Şekil 3.19: Çok istasyon için TEİ Kestirim grafiği

Tek İstasyon için Gün Karşılaştırmalı TEİ

İstasyon Kodu: ankr

İlk Gözlem Tarihi: 10-10-2003

İkinci Gözlem Tarihi: 28-10-2003

Sözde Menzil verilerini kullan:

Faz verilerini kullan:

IONEX verilerini göster:

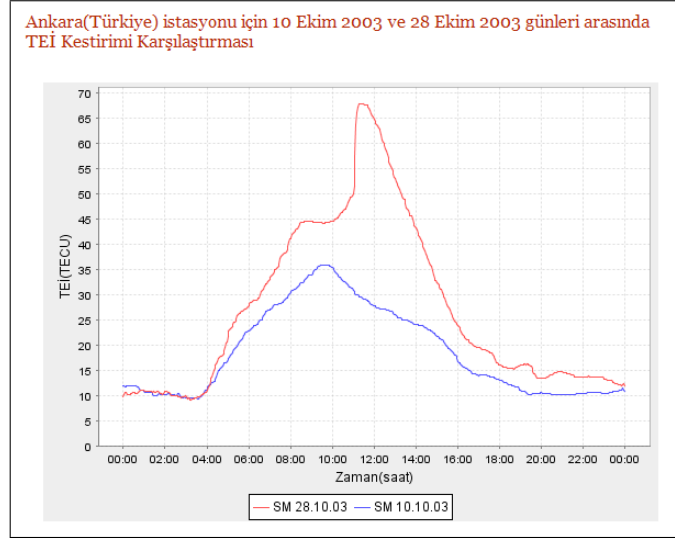
Çıktı türü: Grafik Excel

Hesapla

Şekil 3.20: Tek istasyon-Gün Karşılaştırma için TEİ Kestirimi veri giriş ekranı

seçilmiştir. 10 Ekim 2003 günü iyonosferin sakin olduğu bir gün iken 28 Ekim 2003 günü iyonosferde pozitif bozulma vardır (Nayır, 2007). TEİ kestiriminin sözde menzil verileriyle hesaplanması seçeneği seçilmiştir. Bu girdilere göre iki güne ait TEİ kestirimi Şekil 3.21’de görülmektedir. Beklenildiği gibi pozitif bozulmanın olduğu gündeki TEİ değerleri, iyonosferin sakin olduğu günden daha yüksektir.

Geliştirilen uygulama tüm gün için gözlem verisi bulunmadığı durumlarda, gün içinde gözlem verisi bulunan herbir zaman dilimi için Bölüm 2.4.2’de bahsedilen yöntemi uygulamaktadır. Bu durum için örnek olarak 28 Ekim 2003

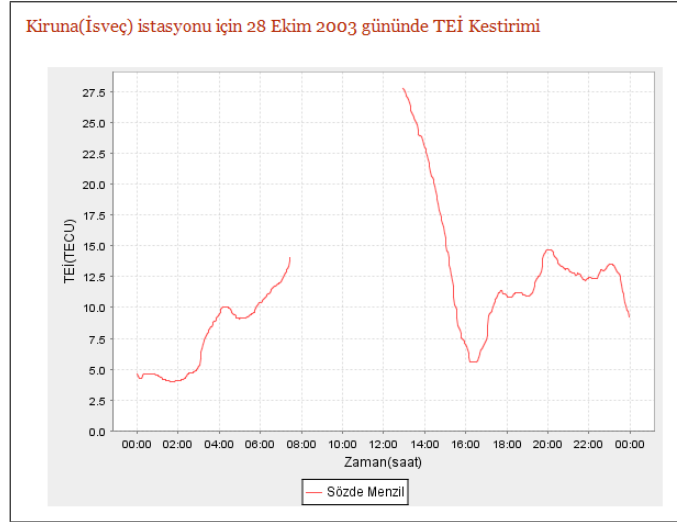


Şekil 3.21: Tek istasyon-Gün Karşılaştırma için TEİ Kestirim grafiği

tarihi ve Kiruna istasyonu seçilmiştir. 28 EKim 2003 tarihli Kiruna istasyonuna ait gözlem dosyalarında 7:29 ile 10:51 arasındaki zaman dilimi için veri bulunmamaktadır. Bu tarih için *Tek İstasyon* seçeneği ile sözde menzil verileri kullanılarak elde edilen TEİ kestirimleri Şekil 3.22'de görülmektedir.

Çıktı türü grafik seçildiğinde üretilen sonuç Taşınabilir Ağ Grafiği (PNG) formatında bir resim dosyasıdır. Bu resim istenirse tarayıcının ilgili menüleri kullanılarak kaydedilebilir. Üretilen kestirim değerlerinin tabüler olarak kaydedilebilmesi için Şekil 3.14'de görüldüğü gibi *Çıktı türü* olarak *Excel* ifadesinin seçilmesi gerekmektedir. Bu seçenek seçildiğinde Şekil 3.23'de görülen Excel formatında çıktı elde edilmektedir. Bu çıktı sözde menzil ve/veya faz verileriyle elde edilen kestirimleri içerebilmektedir. Şekil 3.23'deki çıktı Şekil 3.15'deki girdilerin çıktı türü hariç aynısı girilerek elde edilmiştir. Şekil 3.23'de elde edilen Excel dosyasının sadece başlangıcı gösterilmiştir.

Kullanıcı arayüzü Bölüm 3.3'de anlatılacağı gibi modüler olarak tasarlanmış, bu sayede kullanıcı arayüzünün farklı çıktı türlerini kolaylıkla desteklemesi sağlanmıştır.



Şekil 3.22: Eksik gözlem verisi içeren durum için TEİ Kestirim grafiği

	A	B	C	D	E	F	G
1	Zelenchukskaya(Rusya) istasyonu için 28.10.2003 00:00 gününde TEİ Kestirimi						
2	Tarih	Zaman	Sözde Menzil(TECU)	Faz(TECU)			
3	10/28/2003	0:00:00	10.74414569	11.02184337			
4	10/28/2003	0:02:30	10.80385805	11.02156365			
5	10/28/2003	0:05:00	10.8550674	10.97325364			
6	10/28/2003	0:07:30	10.8550674	10.87965957			
7	10/28/2003	0:10:00	10.79535504	10.78502726			
8	10/28/2003	0:12:30	10.79535504	10.68714994			
9	10/28/2003	0:15:00	10.79535504	10.59337429			
10	10/28/2003	0:17:30	10.64689472	10.51625676			
11	10/28/2003	0:20:00	10.5425615	10.45452362			
12	10/28/2003	0:22:30	10.49626633	10.39749218			

Şekil 3.23: Tek İstasyon için Excel formatında TEİ Kestirim çıktısı

4. ANALİZ

Bu bölümde uygulamanın güvenilirliği ve başarımı ile ilgili analizler verilmiştir. Yapılan analizlerle ilgili kısa değerlendirmeler verilmiştir. Bu değerlendirmelere Bölüm 5’de ayrıca değinilmiştir.

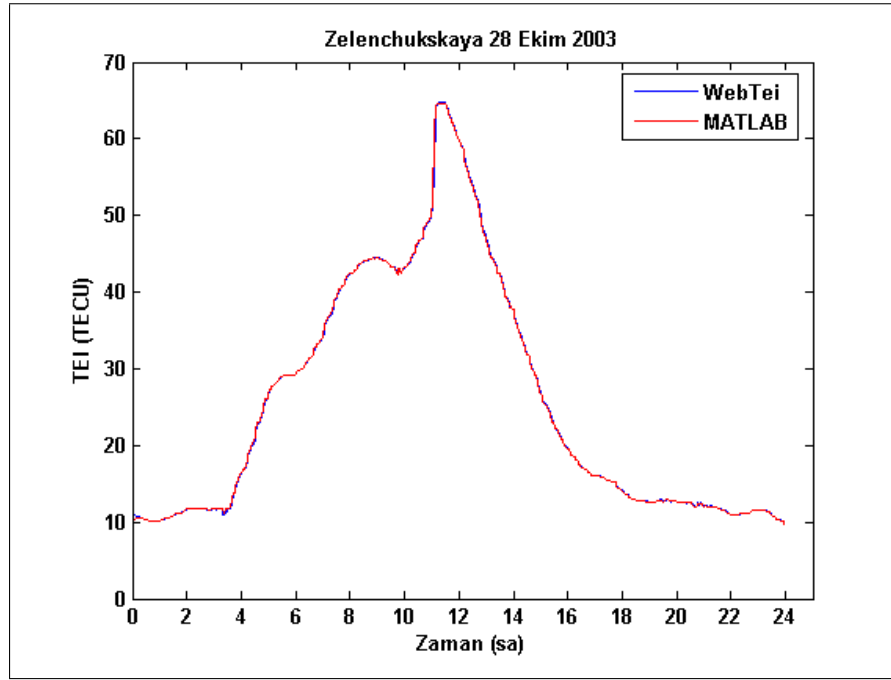
4.1 MATLAB Uygulamasıyla Karşılaştırma

Bu çalışmada Arıkan et al. (2003) ve Arıkan et al. (2004)’de açıklanan TEİ kestirim yöntemlerinin gerçekleştirimi yapılmıştır. Bu gerçekleştirimle eş zamanlı ve bağımsız olarak IONOLAB proje grubu tarafından aynı yöntemlerin MATLAB platformunda gerçekleştirimi yapılmıştır. MATLAB tabanlı uygulama ile bu çalışma karşılaştırılarak yöntemin gerçekleştiriminin güvenilirliği sınanmıştır.

Şekil 4.1’de Zelenchukskaya istasyonuna ait 28 Ekim 2003 günü için geliştirilen uygulama ile MATLAB uygulamasının ürettiği sözde menzil verilerini kullanan TEİ kestirimleri görülmektedir. Geliştirilen uygulamaya ait kestirimler “WebTei” isimli seri ile gösterilmiştir. Görüldüğü gibi kestirimler birbiriyle uyumludur. MATLAB uygulaması ve geliştirilen uygulamanın sözde menzil gözlemlerini kullanarak yaptığı kestirimler Eşitlik 4.1’de verilen Düzgelenmiş Yüzdesel Etkin Hata (NRMSE %) ifadesi kullanılarak karşılaştırılmıştır. Eşitlik 4.1 geliştirilen uygulamanın kestirimlerine göre düzgelenmiş hatayı ifade etmektedir.

$$NRMSE \% = 100 \times \sqrt{\frac{\sum_{n=1}^N [\tilde{x}_M(n) - \tilde{x}_W(n)]^2}{\sum_{n=1}^N \tilde{x}_W^2(n)}} \quad (4.1)$$

Bu ifadede N gün içinde hesaplanan kestirim örnek sayısını göstermekte olup geliştirilen uygulama ve MATLAB uygulaması 30 s zaman çözünürlüğüne sahip gözlem verilerini 1/5 oranında örnekleedikleri için örnek sayısı 576’dır. \tilde{x}_M ifadesi MATLAB uygulamasına ait kestirim değerini, \tilde{x}_W ise geliştirilen web tabanlı uygulamaya ait kestirim değerini ifade etmektedir. Ölçümler toplam 10 günlük (1 Ekim-10 Ekim 2003) veri kullanılarak yapılmıştır. Yapılan karşılaştırma sonuçları



Şekil 4.1: Geliştirilen uygulama ile MATLAB uygulamasına ait TEİ kestirimlerinin karşılaştırması

İstasyon	Tarih	NRMSE %
Zelenchukskaya	1-10 Ekim 2003	0,52
Arti	1-10 Ekim 2003	0,52
Kiruna	1-10 Ekim 2003	0,81

Çizelge 4.1: Geliştirilen uygulama ile MATLAB uygulamasının ürettiği kestirimlerin karşılaştırılması

Çizelge 4.1’de gösterilmiştir. Çizelge 4.1 incelendiğinde farklılıkların düşük olduğu görülmektedir. Dolayısıyla TEİ kestirim yönteminin gerçekleştiriminin güvenilir olduğu sonucuna varılmıştır.

Çizelge 4.1’deki farklılıkların iki temel sebebi olabileceği düşünülmektedir. Birinci sebep kayan noktalı sayıların işlenmesinde Java ve MATLAB platformlarının farklı davranışlarıdır. İkinci sebep ise uydu konum verilerine uygulanan aradeğerleme tekniklerinin farklı olmasıdır. Geliştirilen uygulama doğrusal aradeğerleme kullanırken, MATLAB uygulaması alçak geçiren

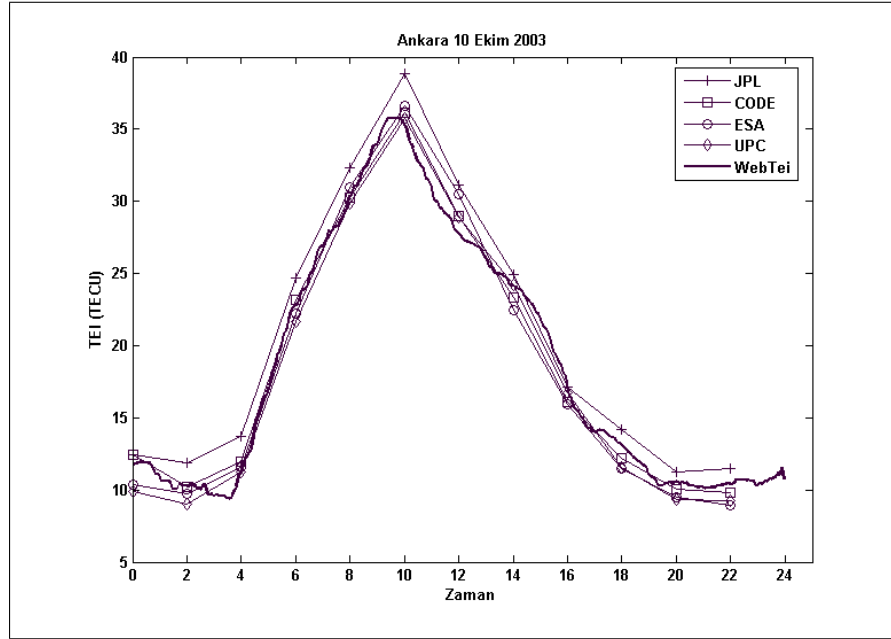
aradeğerleme algoritmasını uygulayan `interp` fonksiyonunu kullanmaktadır.

4.2 IGS Analiz Merkezleriyle Karşılaştırma

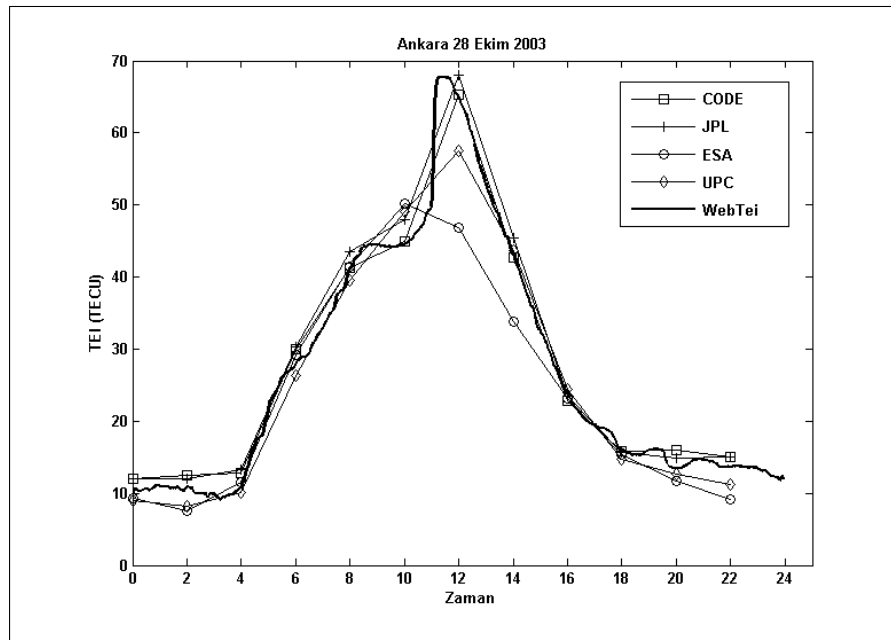
Uygulanan yöntemin güvenilirliği ve üstünlüğü Nayir (2007) tarafından gösterilmiştir. Burada da kısaca yöntemin doğruluğunu göstermek için iyonosferin sakin olduğu 10 Ekim 2003 gününde Ankara istasyonuna ait TEİ kestirimleri karşılaştırılmıştır. IGS analiz merkezlerine ait kestirimler ile geliştirilen uygulamanın sözde menzil gözlemleri kullanarak hesapladığı kestirimler Şekil 4.2(a)'de gösterilmiştir. Şekil 4.2(a)'de WebTei ile gösterilen seri geliştirilen uygulamaya aittir. Görüldüğü gibi elde edilen kestirimler birbiriyle uyumludur.

Bu uyumun sakin günler dışında da geçerli olduğunu gösterebilmek için iyonosferde pozitif bozulma olduğu bir gün olan 28 Ekim 2003 için Ankara istasyonuna ait TEİ kestirimler karşılaştırılmıştır. IGS analiz merkezlerine ait kestirimler ile geliştirilen uygulamanın sözde menzil gözlemleri kullanarak 28 Ekim 2003 günü için Ankara istasyonuna ait hesapladığı kestirimler Şekil 4.2(b)'de gösterilmiştir. Şekil 4.2(b)'de WebTei ile gösterilen seri geliştirilen uygulamaya aittir. Sakin günde olduğu gibi pozitif bozulmalı günde de kestirimlerin uyumlu olduğu görülmektedir. Dolayısıyla kullanılan yöntem diğer yöntemlerle uyumlu sonuçlar üretirken, sağladığı yüksek zaman çözünürlüğü sayesinde öne çıkmaktadır.

17 Ekim 2000 tarihinde jeomagnetik bir fırtına gerçekleşmiş, bu fırtına sırasında saat 22:00 sularında Güney Amerika'da TEİ değerleri çok yüksek değerlere ulaşmıştır. Fedrizzi et al. (2001)'nin çalışmasında New Brunswick Üniversitesi tarafından geliştirilen iyonosfer modelleme tekniği kullanılarak üretilen bu fırtına zaman dilimi için Güney Amerika bölgesine ait TEİ kestirim haritaları incelenmiş, bu haritalarda gösterilen kestirimlerin geliştirilen uygulamanın kestirim değerleriyle uyumlu olduğu, ancak IGS analiz merkezlerinin iyonosfer haritalarında bu ani yükselmenin büyüklük olarak yakalanmadığı görülmüştür. Bu da kullanılan yöntemin yüksek zaman çözünürlüğü ve daha fazla uydudan gelen gözlem verisini değerlendirmesi sayesinde IGS analiz



(a) İyonosferin sakin olduğu gün



(b) İyonosferde pozitif bozulma olduğu gün

Şekil 4.2: IGS analiz merkezlerinin ve geliştirilen uygulamanın TEİ kestirimleri

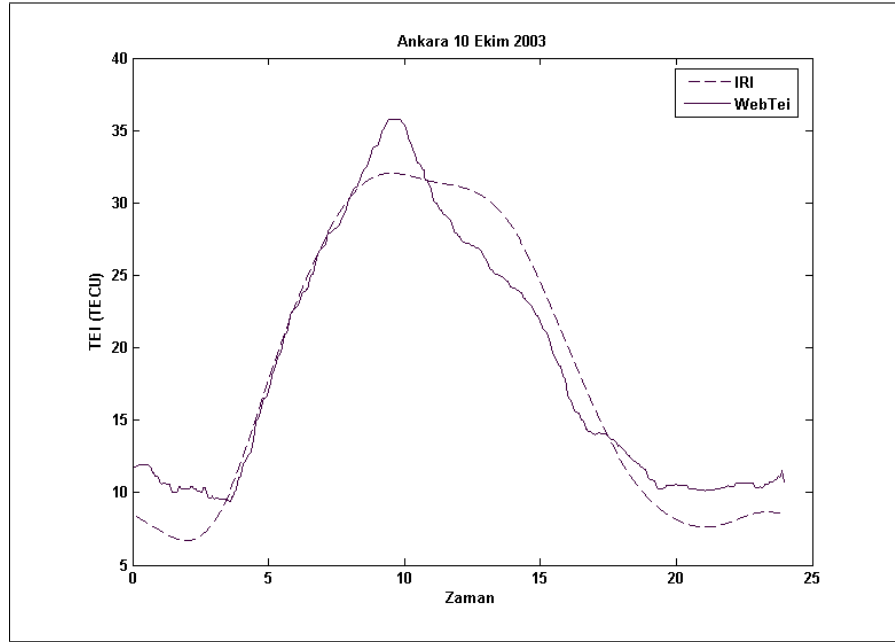
merkezlerinin kestirimlerinden daha hassas ve gürbüz kestirimler yapabildiğini göstermektedir.

4.3 IRI ile Karşılaştırma

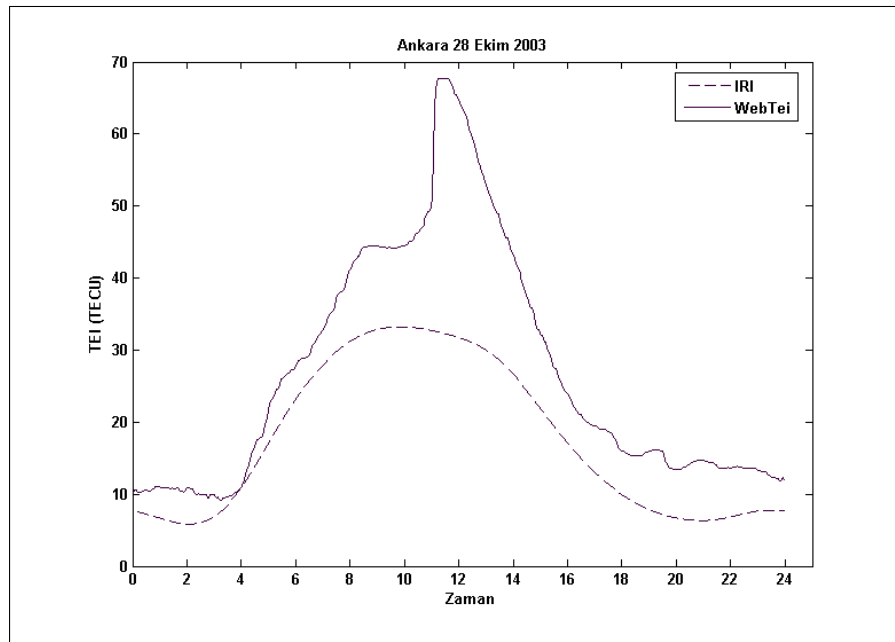
IRI modeline ait kestirimler <http://umlcar.uml.edu/IRI-2001/> adresinden indirilen programla elde edilmiştir. Bu programla Ankara istasyonu için iyonosferin sakin olduğu 10 Ekim 2003 tarihine ait kestirimler ve iyonosferde pozitif bozulma olduğu 28 Ekim 2003 gününe ait kestirimler elde edilmiştir. Geliştirilen uygulama ile de aynı istasyon ve günler için kestirimler elde edilmiştir. 10 Ekim 2003 tarihine ait kestirimler Şekil 4.3(a)'de, 28 Ekim 2003 tarihine ait kestirimler ise Şekil 4.3(b)'de gösterilmiştir. IRI programında tüm parametreler varsayılan değerleriyle kullanılmıştır. Görüldüğü gibi iyonosferin sakin olduğu bir gün olan 10 Ekim 2003 günü için IRI geliştirilen uygulamaya yakın kestirimler verse de, iyonosferin pozitif bozulmalı olduğu 28 Ekim 2003 günü için doğru kestirimler yapamamaktadır. Bu gözlemlere dayanarak geliştirilen uygulamanın güncel gözlem verilerine dayanması nedeniyle IRI modeline göre daha doğru ve yüksek hassasiyetli kestirimler ürettiği söylenebilir.

4.4 Başarım

Geliştirilen uygulamanın başarımı IONOLAB proje grubu tarafınan geliştirilen MATLAB tabanlı uygulama ile karşılaştırılmıştır. Yapılan karşılaştırmada dosya indirilmesi, gözlem dosyalarının ayıklanması, uydu bazında TEİ değerlerinin hesaplanması ve grafik üretimi gibi işlemler yerine düzgülendirme işleminin harcadığı süre ölçülmüştür. Bunun sebebi, algoritmanın en işlem yoğun kısmının bu kısmı olmasıdır. Çizelge 4.2'de 10 Ekim 2003 tarihinde Ankara, Zelenchukskaya, Kiruna ve Arti istasyonları için web uygulaması ile MATLAB uygulamasının TEİ kestirimleri sırasında alınan ölçümler görülmektedir. Bu ölçümler milisaniye cinsinden olup, belirtildiği gibi sadece düzgülendirme işlemi sırasında geçen süre ölçülmüştür. Ölçümler her istasyon için 100 defa tekrarlanmış olup, ölçülen sürelerin ortalaması alınmıştır. Ölçümler



(a) İyonosferin sakin olduğu gün



(b) İyonosferde pozitif bozulma olduğu gün

Şekil 4.3: IRI ile geliştirilen uygulamanın TEİ kestirimleri

için Java platformunda `System.currentTimeMillis()` fonksiyonu kullanılırken, MATLAB platformunda `cputime` fonksiyonu kullanılmıştır. Ölçümler 1,4 GHz Pentium M işlemciye ve 1 GB RAM hafızaya sahip bir bilgisayar üzerinde yapılmıştır. Çizelge 4.2 incelendiğinde MATLAB uygulamasının daha iyi bir

Uygulama	Ortalama Süre(ms.)
Web Uygulaması	1493
MATLAB Uygulaması	692

Çizelge 4.2: Geliştirilen uygulama ile MATLAB uygulamasının düzgünleştirme süreleri

başarım sağladığı görülmektedir. Geliştirilen uygulamada sisteme özgü matris kütüphanesi kullanılmasına rağmen MATLAB uygulamasının daha iyi başarım sağladığı görülmesi üzerine düzgünleştirme işlemi sırasında gerçekleştirilen matris tersi alma işlemi için de ölçümler yapılmıştır. Bu ölçümler Çizelge 4.3'de görülmektedir. Ölçümler Ankara istasyonu için 10 Ekim 2003 tarihine ait kestirim işlemi 100 defa tekrar edilerek alınmış olup, Çizelge 4.3'de alınan ölçümlerin ortalaması verilmiştir. Burada iki platformunun genel başarımı ile ilgili bir çıkarımda

Uygulama	Ortalama Süre(ms.)
Web Uygulaması	118
MATLAB Uygulaması	337

Çizelge 4.3: Geliştirilen uygulama ile MATLAB uygulamasının matris tersi alma süreleri

bulunmak mümkün değildir. Ancak TEİ kestirim uygulaması için web tabanlı versiyonun matris başarımı olarak MATLAB tabanlı versiyondan üstün olmasına rağmen düzgünleştirme işlemi açısından geride kaldığı görülmektedir. Bu nedenle geliştirilen uygulamanın düzgünleştirme işlemi sırasında yaptığı diğer işlemlerin iyileştirmeye açık olduğu anlaşılmaktadır. Bunun yanında MATLAB tabanlı uygulamanın lisans gerektirmesi ve kurulum gerekliliği gibi dezavantajları bulunmaktadır.

Düzgünleştirme algoritmasında yoğun matris işlemleri sözkonusudur. Java platformunun içinde bu ihtiyacı karşılayan standart bir uygulama kütüphanesi olmaması nedeniyle açık kaynaklı kütüphaneler araştırılmıştır. Bu araştırmalar sonucu Java İçin Matris Araç Takımı (MTJ) seçilmiştir. MTJ'nin seçilmesindeki en önemli faktör lineer matris denklemlerini çözmede yaygın olarak kullanılan Temel Lineer Cebir Altprogramları (BLAS)¹ ve Lineer Cebir Paketi (LAPACK)² paketlerini Java ile gerçekleştirmiş olması, bunun yanında bu paketlerin sisteme özgü sürümlerine de Java Sisteme Özgü Arayüzü (JNI)³ ile ulaşabilmesidir. Sisteme özgü gerçekleştirime ulaşabilmenin avantaj sağlayıp sağlamadığını anlayabilmek için iki farklı konfigürasyonda ölçümler yapılmıştır. Birinci konfigürasyonda MTJ kütüphanesi BLAS ve LAPACK arayüzlerinin Java tabanlı gerçekleştirimi olan LAPACK'in Java sürümü (JLAPACK)⁴ ile çalıştırılmıştır. İkinci konfigürasyonda ise BLAS ve LAPACK arayüzlerinin sisteme özgü gerçekleştirimi olan LAPACK'in C sürümü (CLAPACK)⁵ kullanılmıştır. MTJ sisteme özgü kütüphaneleri JNI üzerinden kullanma özelliğini Sisteme Özgü Sayısal Arayüz (NNI) olarak isimlendirmektedir. Ölçümler Ankara, Zelenchukskaya, Kiruna ve Arti istasyonlarının herbiri için 10 Ekim 2003 tarihine ait kestirimlerin düzgünleştirilmesi işleminin 100 defa çalıştırılması ile elde edilmiş, Çizelge 4.4'de elde edilen sürelerin ortalaması alınmıştır. Ölçümler iki farklı çözünürlük kullanılarak gerçekleştirilmiştir. İlk ölçümler 30 saniye zaman çözünürlüğüne sahip gözlem verilerinin olduğu gibi kullanılmasıyla gerçekleştirilmiştir. İkinci ölçümler ise gözlem verilerinin 1/5 oranında alt örneklenmesi sonucu elde edilen veriler ile gerçekleştirilmiştir. Böylece uydu başına ilk ölçümlerde 30 saniye çözünürlük (2880 örnek) , ikinci ölçümlerde ise 150 saniye çözünürlük (576 örnek) kullanılmıştır. Kullanılan çözünürlük Eşitlik 2.18'de tersi alınan matrisin boyutunu belirlemektedir. Bu nedenle

¹<http://www.netlib.org/blas/>

²<http://www.netlib.org/lapack/>

³<http://java.sun.com/javase/6/docs/technotes/guides/jni/spec/jniTOC.html>

⁴<http://icl.cs.utk.edu/f2j/>

⁵<http://www.netlib.org/clapack/> adresindeki gerçekleştirimin ATLAS (<http://math-atlas.sourceforge.net/>) kullanılarak oluşturulmuş hali kullanılmıştır.

30 saniye çözünürlükte 2880×2880 boyutunda bir matrisin tersinin alınması gerekirken, 150 saniye çözünürlükte 576×576 boyutunda bir matrisin tersinin alınması gerekmektedir. Çizelge 4.4 incelendiğinde NNI kullanımını matris boyut-

Matris Paketi	Matris Boyutu	Ortalama Süre(ms.)
NNI	576×576	1493
JLAPACK	576×576	2229
NNI	2880×2880	47934
JLAPACK	2880×2880	149229

Çizelge 4.4: MTJ'nin JLAPACK ve NNI ile başarımı

ları küçük olduğu durumlarda işlem süresinde %33'lük bir iyileşme sağlarken matris boyutları arttığı durumda ise %68'lik bir iyileşme sağladığı görülmektedir. Bu sonuçlara göre NNI kullanımını özellikle matris boyutlarının büyük olduğu durumlarda ciddi bir başarımla artışı sağlamaktadır. NNI kullanımını sayesinde daha işlem yoğun algortimalarda da yüksek başarımla elde edilmesi mümkün olacaktır. Bu da uygulamanın ölçeklenebilirliğini arttırmaktadır. Ayrıca geliştirilen uygulamada MTJ'nin kullanımını sisteme özgü uygulamalara yakın başarımla elde edilmesine olanak sağlamıştır.

4.5 Alt Örnekleme

Bu bölümde alt örneklemenin yöntemin kestirimlerine olan etkisi incelenmiştir. Bu amaçla Ankara istasyonu için iyonosferin sakin olduğu 10 Ekim 2003 ve iyonosferin pozitif bozulmalı olduğu 28 Ekim 2003 günlerine ait kestirimler farklı alt örnekleme oranlarıyla hesaplanmıştır. Bu kestirimler sırasında Eşitlik 2.12'de geçen σ için 0,2817, Eşitlik 2.16'de geçen μ için 0,1 ve k_c için 8,0 değerleri kullanılmıştır. Ölçümler sırasında en büyük alt örnekleme oranı olarak 1/60 oranı denenmiş, ancak bu oran kullanıldığında, örnekleme yapılmadan elde edilen kestirimlerden büyük ölçüde sapmalar olduğu gözlenmiştir. Farklı örnekleme oranları için kullanılan kayan ortanca filtrenin pencere genişlikleri Çizelge 4.5'de verilmiştir. Şekil 4.4(a)'de iyonosferin sakin olduğu günde farklı

Alt örnekleme oranı	Çözünürlük	Pencere genişliği
1	30 s	81
1/2	1 dk	41
1/5	2,5 dk	17
1/10	5 dk	9
1/30	15 dk	5

Çizelge 4.5: Alt örnekleme oranları için kullanılan kayan ortanca filtrenin pencere genişlikleri

alt örnekleme oranlarıyla elde edilen kestirimler görülmektedir. Şekil 4.4(b)'de ise iyonosferin pozitif bozulmalı olduğu gün için farklı alt örnekleme oranlarıyla elde edilen kestirimler görülmektedir. Şekil 4.4'de görülen ölçümler Eşitlik 4.2'de verilen NRMSE % ifadesi kullanılarak karşılaştırılmıştır. Eşitlik 4.2 alt örnekleme yapılmadan elde edilen kestirime göre düzelenmiş hatayı ifade etmektedir.

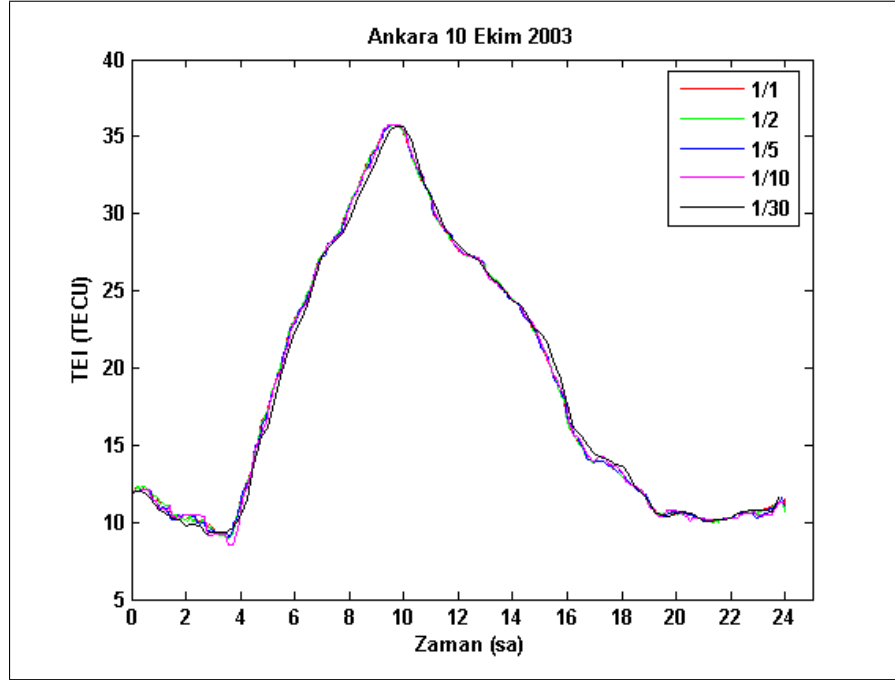
$$NRMSE_{1/k} \% = 100 \times \sqrt{\frac{\sum_{n=1}^{N/k} [\tilde{x}_{1/k}(n) - \hat{x}_{1/k}(n)]^2}{\sum_{n=1}^{N/k} \hat{x}_{1/k}^2(n)}} \quad (4.2)$$

Bu ifadede N alt örnekleme yapılmadan elde edilen örnek sayısı olup 30 s çözünürlüğündeki kestirimler için 2880'dir. $1/k$ örnekleme oranı olup $\tilde{x}_{1/k}$ ifadesi bu örnekleme oranı ile elde edilen kestirimi ifade etmektedir. $\hat{x}_{1/k}$ ise gözlem verilerinde alt örnekleme yapılmadan elde edilen kestirimlerin $1/k$ oranında alt örneklemeyle elde edilen değerler olup

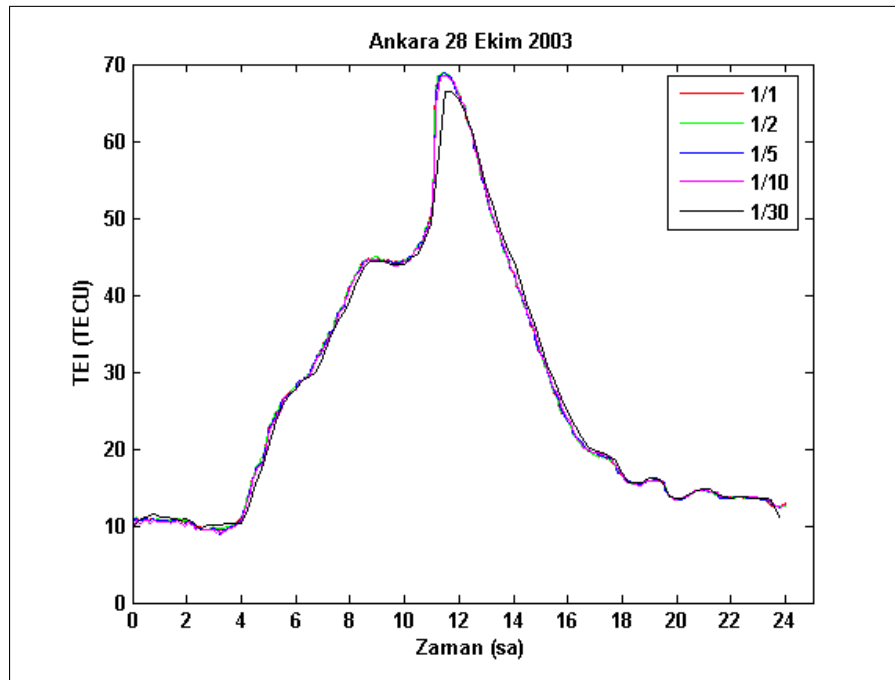
$$\hat{x}_{1/k}(n) = \tilde{x}_{1/1}(kn) \quad (4.3)$$

ifadesiyle gösterilebilir. Bu eşitlikte $\tilde{x}_{1/1}$ alt örnekleme yapılmadan elde edilen kestirim değerini ifade etmektedir. Çizelge 4.6'de 10 Ekim 2003 ve 28 Ekim 2003 günleri için Ankara istasyonuna ait farklı örnekleme oranları ile elde edilmiş kestirimlere ait NRMSE % değerleri verilmiştir.

Şekil 4.4'de ve Çizelge 4.6'de görüldüğü gibi farklı alt örnekleme oranları kullanılması yöntemin kestirimlerinde önemli ölçüde farklılıklar



(a) İyonosferin sakin olduğu gün



(b) İyonosferde pozitif bozulma olduğu gün

Şekil 4.4: Farklı alt örnekleme oranlarıyla elde edilen TEİ kestirimleri

Alt örnekleme oranı	Çözünürlük	NRMSE % 10/10/2003	NRMSE % 28/10/2003
1/2	1 dk	0,28	0,31
1/5	2,5 dk	0,72	0,85
1/10	5 dk	1,24	1,58
1/30	15 dk	2,43	4,24

Çizelge 4.6: Farklı alt örnekleme oranlarıyla elde edilen kestirimlerin karşılaştırılması

oluşturmamaktadır. Geliştirilen uygulamada alt örnekleme kullanılmasının temel nedeni kestirim süresinin uzun olması, dolayısıyla uygulamanın kullanılabilirliğinin azalmasıdır. Bu nedenle alt örnekleme yapılmasına karar verilmiş, ancak alt örnekleme işleminin kestirimlerin güvenilirliğini etkilememesi için %1'den az hata içeren alt örnekleme oranları değerlendirilmiştir. Bu değerlendirmeler sonucu geliştirilen uygulamada 1/5 oranında alt örnekleme kullanılmasına karar verilmiştir.

5. SONUÇ VE ÖNERİLER

İyonosferin gözlenmesiyle ilgili GPS gözlemlerine dayalı yüksek zamansal çözünürlüklü kestirimlere ulaşmak mümkün değildir. Bu çalışmada Arıkan et al. (2003) tarafından geliştirilen gürbüz TEİ kestirim yönteminin uygulanması yapılmış ve bu yöntemle elde edilen kestirimlerden azami fayda elde edilmesi sağlanmıştır. Yöntemin uygulanmasında Nayir (2007)'in çalışmasından da faydalanılmıştır.

Yazılımın geliştirilmesinde endüstride güvenilirliği ispatlanmış ve en iyi pratikleri içeren teknolojiler kullanılmıştır (Nesneye yönelik programlama, Java, JavaEE, Spring çatısı).

Geliştirilen sistem modüler yapısı sayesinde gelecekte olabilecek değişikliklere izin veren bir yapıda tasarlanmıştır. Bu sayede sistemin uzun ömürlü olması mümkün olacaktır. TEİ kestirim yönteminde olabilecek değişiklikler bu modüler yapı sayesinde uygulamaya kolaylıkla yansıtılabilecektir. Kullanıcı arayüzünün de web tabanlı tasarlanması sayesinde uzaktan ulaşılabilir ve kolay kullanılabilir olması sağlanmıştır. Ayrıca uygulamanın kullanıcı arayüzü katmanında MVC tasarım kalıbından faydalanılması sayesinde verilerin gösterimi verilerin üretiminden bağımsız hale getirilmiş, bu sayede verilerin farklı arayüzlerle sunulması için gerekli esnek yapı sağlanmıştır. Uygulamanın modülerliği gözlem verilerine ulaşım kısmında da sağlanmış, gözlem verilerinin ayıklanmasıyla ilgili detaylar, TEİ kestiriminden izole edilmiştir. Bu sayede gelecekte ortaya çıkabilecek farklı gözlem kaynakları da sisteme kolaylıkla eklenebilecektir.

Sistem birçok GPS uydusundan alınan gözlem verilerini işlediği için yoğun işlemci gücü kullanmaktadır. Kullanım kolaylığını kısıtlayan bu durum, yapılan hesaplamaların veritabanında saklanması sayesinde aşılabilmektedir.

Geliştirilen uygulamanın TEİ kestirim yöntemini doğru gerçekleştirdiğini teyit etmek için elde edilen TEİ kestirimleri aynı yöntemi gerçekleştiren MATLAB tabanlı diğer bir uygulamanın kestirimleriyle karşılaştırılmış, elde edilen sonuçların uyumlu olduğu görülmüştür. Bu şekilde sistemin güvenilir olduğu

gösterilmiştir. Ayrıca IGS merkezlerinin TEİ kestirimleri de gösterilerek kullanılan TEİ kestirim yönteminin de güvenilir olduğu gösterilmiştir. Uygulamanın geliştirilmesinde kullanılan matris kütüphanesinin başarımıyla ilgili ölçümler yapılmış olup, sisteme özgü kütüphane kullanımının uygulamanın başarımını arttırdığı gösterilmiştir.

IONOLAB yönteminin IRI modeline ve IGS analiz merkezlerine göre daha gürbüz TEİ kestirimleri üretebildiği görülmüştür. Yöntemin görüş alanındaki tüm GPS uydularına ait yüksek çözünürlüklü gözlem verilerini değerlendirmesi sayesinde diğer yöntemlerin tespit edemediği iyonosfer olaylarını tespit edebildiği gösterilmiştir.

Uygulamanın arayüzü sonuçları grafik olarak veya Excel formatında sunmaktadır. Daha fazla fayda sağlamak ve diğer sistemlere arayüz sağlamak aracılığıyla kestirim değerlerinin farklı kanallarla dışa verilmesi sağlanabilir. Örneğin uygulamanın hesapladığı kestirimler web servisi olarak sunulabilir, böylece farklı platformlardaki uygulamalar hesaplanan kestirimlerden faydalanabilir. Hesaplanan istasyon bazındaki kestirim değerleri birleştirilerek iyonosfer haritaları oluşturulabilir.

Uygulamanın düşük çözünürlüklü uydu pozisyon bilgileri için kullanıldığı doğrusal aradeğerleme yerine daha gelişmiş aradeğerleme yöntemleri kullanılabilir. Ayrıca uygulamanın kullanıcı arayüzü Asenkron Java betiği ve XML (AJAX) teknolojisi ile daha kullanışlı hale getirilebilir.

KAYNAKLAR

- Alur, D., Crupi, J., and Malks, D. (2003). *Core J2EE Patterns: Best Practices and Design Strategies*. Prentice Hall PTR, Palo Alto, California, second edition.
- Arikan, F., Erol, C., and Arikan, O. (2003). Regularized estimation of vertical total electron content from global positioning system data. *Journal of Geophysical Research*, 108(A12):1469–1480.
- Arikan, F., Erol, C., and Arikan, O. (2004). Regularized estimation of vertical total electron content from gps data for a desired time period. *Radio Science*, 39:RS6012.
- Asbury, S. and Weiner, S. R. (1999). *Developing Java Enterprise Applications*. John Wiley & Sons.
- Beutler, G., Rothacher, M., Schaer, S., Springer, T., Kouba, J., and Neilan, R. (1999). The international gps service (igs): an interdisciplinary service in support of earth sciences. *Advances in Space Research*, 23(4):631–653.
- Bilitza, D. (1990). International reference ionosphere 1990. Technical Report NSSDC/WDC-A-R&S 90-22, National Space Science Data Center/World Data Center A for Rockets and Satellites, Lanham, MD.
- Bilitza, D. (2001). International reference ionosphere 2000. *Radio Science*, 36:261–275.
- Blewitt, G. (1997). *Geodetic Applications of GPS*, chapter Basics of the GPS Technique: Observation Equations. Nordic Geodetic Commission, Sweden.
- Committee”, D. S. P. (1979). *Programs for Digital Signal Processing*. IEEE Press.
- Crawford, W. and Kaplan, J. (2002). *J2EE Design Patterns*. O’Reilly.
- Davies, K. and Hartmann, G. (1997). Studying the ionosphere with the global positioning system. *Radio Science*, 32(4):1695–1704.

- El-Rabbany, A. (2002). *Introduction to GPS : the Global Positioning System*. Artech House, Inc., Norwood, MA.
- Fedrizzi, M., Kantor, I. J., de Paula, E. R., Langley, R. B., Santos, M. C., and Komjathy, A. (2001). The low-latitude ionosphere: Monitoring its behaviour with gps. In *14th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Salt Lake City, Utah,. ION GPS 2001.
- Foed, N. (2004). *Art of Java Web Development*. Manning Publications Co.
- Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison Wesley, 3rd edition.
- Fowler, M. (2004). Inversion of control containers and the dependency injection pattern. <http://www.martinfowler.com/articles/injection.html>.
- Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., and Stafford, R. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Geary, D. and Horstmann, C. (2004). *Core JavaServer Faces*. Core Series. Prentice Hall PTR.
- Gosling, J. and McGilton, H. (1996). The java language environment a white paper. <http://java.sun.com/docs/white/langenv>.
- Gurtner, W. (1994). Rinex: The receiver-independent exchange format. *GPS World*, 5(7):48–52.
- Hernandez-Pajares, M., Juan, J., and Sanz, J. (1999). New approaches in global ionospheric determination using ground gps data. *Journal of Atmospheric and Solar-Terrestrial Physics*, (61):1237–1247.
- Hunsucker, R. and Hargreaves, J. (2003). *The High-latitude ionosphere and its effects on radio propagation*. Cambridge University Press.
- IGS (2006). Igs web site. <http://igsb.jpl.nasa.gov>.

- Johnson, R. (2003). *Expert One-on-One J2EE Design and Development*. Programmer to Programmer. Wiley Publishing, Inc., Indianapolis, Indiana.
- Johnson, R., Hoeller, J., Arendsen, A., Risberg, T., and Sampaleanu, C. (2005). *Professional Java Development with the Spring Framework*. Wiley Publishing, Inc., Indianapolis, Indiana.
- Komjathy, A. (1997). *Global ionospheric total electron content mapping using the global positioning system*. Doktora tezi, The University of New Brunswick (Kanada).
- Magnusson, K., Flanagan, D., Farley, J., and Crawford, W. (1999). *Java Enterprise in a Nutshell*. O'Reilly.
- Mannucci, A., Wilson, B., and Edwards, C. (1993). A new method for monitoring the earth's ionospheric total electron content using the gps global network. In *Proceedings of ION GPS-93, the 6th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pages 1323–1332, Salt Lake City, UT. Division of The Institute of Navigation.
- McGovern, J., Ambler, S., Stevens, M., Linn, J., Sharan, V., and Jo, E. (2003). *The Practical Guide to Enterprise Architecture*. Prentice Hall PTR.
- Meeus, J. (1998). *Astronomical Algorithms*. Willmann-Bell, Inc., 2nd edition.
- Nayir, H. (2007). Küresel konumlama sistemi kullanılarak toplam elektron içeriği kestirimi. Yüksek lisans tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü.
- Remondi, B. (1989). Extending the national geodetic survey standard for gps orbit formats. Technical Report NOS 133 NGS 46, National Geodetic Information Branch, NOAA, Rockville, MD, 20852.
- Schaer, S. (1999). *Mapping and Predicting the Earth's Ionosphere Using the Global Positioning System*. Doktora tezi, Astronomical Institute, University of Berne, Berne, İsviçre.

- Schaer, S., Gurtner, W., and Feltens, J. (1998). Ionex: The ionosphere map exchange format version 1. In *Proceedings of the 1998 IGS Analysis Centres Workshop*, number 233-247, Darmstadt, Almany. ESOC.
- Tannebaum, A. (1988). *Computer Networks*. Prentice Hall, 2nd edition.
- Walls, C. and Breidenbach, R. (2005). *Spring in Action*. Manning Publications Co.
- Wells, D., Beck, N., Delikaraoglou, D., Kleusberg, A., Krakiwsky, E., Lachapelle, G., Langley, R., Nakiboglu, M., Schwarz, K., Tranquilla, J., and Vanicek, J. (1999). *Guide to GPS positioning*. Canadian GPS Associates.
- Yizengaw, E. and Essex, E. (2002). Use of gps signals to study total electron content of the ionosphere during the geomagnetic storm on 22 september 1999. Blue Mountains, Australia. Workshop on Applications of Radio Science (WARS).

KAVRAMLAR SÖZLÜĞÜ

alt örnekleme	:	subsampling
aradeğerleme	:	interpolation
ara katman yazılımı	:	middleware
arayüz	:	interface
ayıklayıcı	:	parser
bağlam	:	context
bellek yönetimi	:	memory management
betik dili	:	scripting language
birincil anahtar	:	primary key
birlikte işlerlik	:	concurrency
birim işlem	:	transaction
birim testi	:	unit test
çalıştırma ortamı	:	runtime
çatı	:	framework
çekirdek	:	core
çoklu dil desteği	:	internationalization
dağıtık kurumsal uygulama	:	distributed enterprise application
denetleyici	:	controller
donanım mimarisi	:	hardware architecture
etiket	:	tag
gerçekleştirim	:	implementation
gevşek bağlanma	:	loose coupling
hayat süreci	:	lifecycle
ister	:	requirement
iş bileşeni	:	business component
kalıp	:	pattern
kap	:	container
katman	:	layer

nesne/işikisel eşleme	:	object to relational mapping
sonuç kümesi	:	result set
sunucu tabanlı betik	:	server-side script
nesne	:	object
nesne tabanlı	:	object oriented
ortanca	:	median
sınıf şeması	:	class diagram
sıralama şeması	:	sequence diagram
sisteme özgü	:	native
sürerlik	:	persistence
uyarlayıcı	:	adapter
uygulama sunucusu	:	application server
veritabanı bağlanırlığı	:	database connectivity
yabancı anahtar kısıtı	:	foreign key constraint

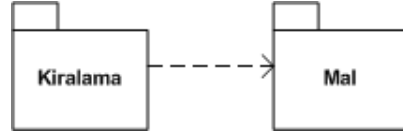
EKLER

A. TASARIM GÖSTERİM SİSTEMİ

Uygulamanın tasarımının anlatımında Birleşik Modelleme Dili (UML) kullanılmıştır. Bu bölümde UML gösterim sisteminin ilgili kısımları kısaca açıklanmaktadır. UML gösterim sisteminin detayları için Fowler (2003)'e başvurulabilir.

A.1 Paket Şeması

Şekil A.1'de örnek bir paket şeması verilmiştir. Burada paketler uygu-

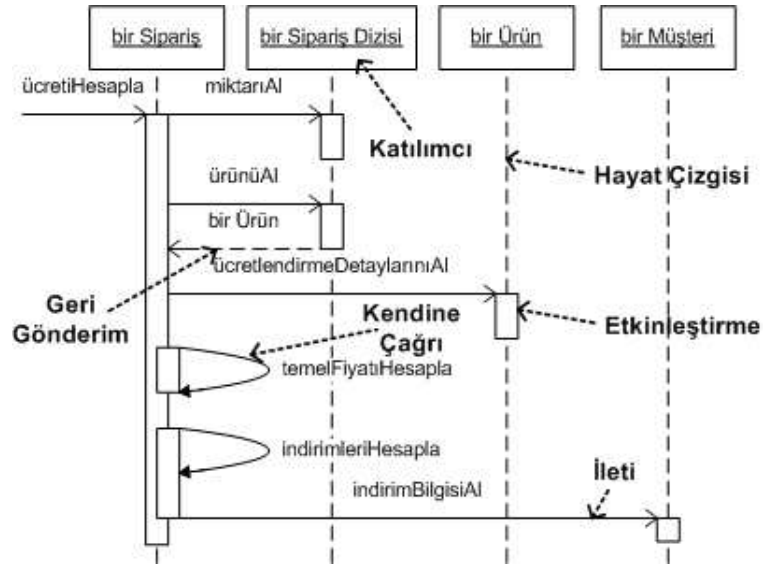


Şekil A.1: Örnek paket şeması

lamamanın temel parçalarını (Kiralama ve Mal) ifade etmektedir. Kesikli ok işareti de bir paketin diğer pakete bağımlı olduğunu göstermektedir. Şekil A.1'e göre Kiralama paketi Mal paketine bağımlıdır.

A.2 Sıralama Şeması

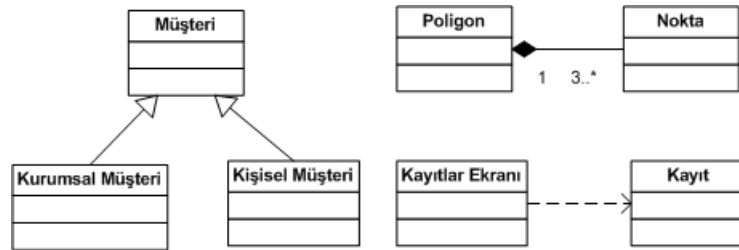
Şekil A.2'de örnek bir sıralama şeması verilmiştir. Şemanın üst tarafındaki kutular katılımcı nesnelere göstermektedir (bir Sipariş Dizisi). Uygulamamız için verilen sıralama şemalarında katılımcı nesne isimleri yerine sınıf isimleri verilmiştir. Kutuların hizasında uzanan kesikli çizgi nesnenin hayat çizgisini ifade etmektedir ve aşağıya doğru zaman ilerlemektedir. Bu kesikli çizgi üzerindeki dikdörtgenler ise o nesnenin etkin hale geldiği zaman dilimlerini ifade etmektedir. Katılımcılar arasındaki iletileri göstermek için ise düz çizgili oklar kullanılmaktadır (indirimBilgisiniAl). Nesnenin kendi hayat çizgisinden başlayıp yine kendine dönen oklar ise nesnenin kendisine olan bir çağrısını ifade etmektedir (ücretlendirmeDetaylarınıAl). Kesikli çizgi ile gösterilen oklar ise bir nesnenin diğerine geri gönderdiği cevabı belirtmektedir (bir Ürün).



Şekil A.2: Örnek sıralama şeması

A.3 Sınıf Şeması

Şekil A.3'de örnek bir sınıf şeması verilmiştir. Müşteri sınıfı Kurumsal



Şekil A.3: Örnek sınıf şeması

Müşteri ve Kişisel Müşteri sınıfları ile bir genelleştirme ilişkisi içindedir. Yani Müşteri sınıfı Kurumsal Müşteri ve Kişisel Müşteri sınıflarının genel halidir. Poligon sınıfının Nokta sınıfı ile olan ilişkisi bir içerme ilişkisidir, yani Poligon sınıfı Nokta sınıfını içermektedir. Poligon ve Nokta sınıflarının arasındaki çizgi üzerinde gösterilen rakamlarda çokluğu ifade etmektedir. Poligon ve Nokta ilişkisinde bir Poligon 3 veya daha fazla Nokta içermektedir. Kayıtlar Ekranı ile Kayıt sınıfı arasındaki ilişki ise bağımlılık ilişkisidir.

B. TASARIM SINIF SÖZLÜĞÜ

Bölüm 3’de kullanılan Türkçe sınıf isimlerinin İngilizce karşılıkları bu bölümde verilmiştir.

al	: get
ayıkla	: parse
ayıklayıcı	: parser
başlangıçZamanlarınıAl	: getStartTimes
bitişZamanlarınıAl	: getEndTimes
crxDenRnxE	: crx2rnx
ÇokİstasyonDenetleyicisi	: MultipleStationController
Denetleyici	: Controller
denklemiÇöz	: calcX
DosyaYönetici	: FileManager
döndür	: return
Düzenleştirici	: Regulator
fazaGöreTei	: vtecByPhase
fazaGöreTeiHesapla	: calculateVtecByPhase
gönderildiğinde	: onSubmit
GözlemKaynağıGerçekleştirimi	: ObservationSourceImpl
grafik	: chart
günlükAlıcıYanlılığıAl	: getReceiverBias
günlükGözlemAl	: getDailyObservation
GünlükGözlemGerçekleştirimi	: DailyObservationImpl
GünlükGözlemKaynağıGerçekleştirimi	: ObservationSourceImpl
günlükKonumAl	: getDailyPosition
günlükTeiAl	: getDailyVtec
günlükTeiEkle	: addDailyVtec
GünlükTeiHesaplayıcıGerçekleştirimi	: DailyVtecCalculatorImpl
GünlükTeiHesaplayıcıVekili	: DailyVtecCalculatorProxy

GünlükTeiVerisiErişimGerçekleştirimi	: DailyVtecDaoImpl
günlükUyduYanlılığıAl	: getSatelliteBias
hesaplama	: calculation
ionexTeisi	: ionexByVtec
ionexTeisiAl	: getIonexByVtec
menzileGöreTei	: vtecByRange
menzileGöreTeiHesapla	: calculateVtecByRange
ParçalıDenetleyici	: PartedChecker
ParçalıGünDüzgünleştirici	: PartDayRegulator
parçalımı	: isParted
RinexAyıklayıcı	: RinexParser
rinexDosyasıAl	: getRinexFile
sıkıştırmayıAç	: decompress
SoyutExcelGörünümü	: AbstractExcelView
sun	: render
TEİ	: Vtec
TeiExcelGörünümü	: TecExcelView
TeiHesaplayıcı	: VtecCalculator
TekİstasyonDenetleyicisi	: SingleStationController
TekİstasyonGünKarşılaştırmaDenetleyicisi	: SingleStationDayComparisonController
TümGünDüzgünleştirici	: FullDayRegulator
veri erişim	: dao
veri	: data
WGetGerçekleştirimi	: WGetImpl
yardımcı	: utility
YönlendiriciServlet	: DispatcherServlet

C. DETAYLI TASARIM

Bu bölüm geliştirilen uygulamanın önemli bileşenlerini anlatmaktadır.

C.1 Sunum Katmanı

Sunum katmanını oluşturan web paketine ait sınıf şeması Şekil C.1'de gösterilmiştir. Sunum katmanındaki bileşenlerin birbirine bağlanması için geliştirilen Spring bağlam konfigürasyon dosyası webtec-servlet.xml'in içeriği aşağıda verilmiştir:

```
<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE beans PUBLIC
"-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<!--
- "webtei" YönlendiriciServlet'i için b?lam dosyası.
-->
<beans>

    <!-- Çoklu dil desteği -->

    <bean id="localeResolver"
        class="org.springframework.web.servlet.i18n.CookieLocaleResolver"/>
    <bean id="localeChangeInterceptor"
        class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"/>

    <bean id="messageSource"
        class="org.springframework.context.support.ResourceBundleMessageSource">
        <property name="basename" value="messages"/>
    </bean>

    <!-- Denetleyiciler -->

    <bean name="singleStationController" class="vtec.web.SingleStationController">
        <property name="validator" ref="singleStationQueryValidator"/>
        <property name="calculator" ref="vtecCalculator"/>
    </bean>

    <bean name="singleStationDayComparisonController"
        class="vtec.web.SingleStationDayComparisonController">
        <property name="validator" ref="singleStationQueryValidator"/>
        <property name="calculator" ref="vtecCalculator"/>
    </bean>
```

```

<bean name="multipleStationController"
      class="vtec.web.MultipleStationController">
  <property name="validator" ref="multipleStationQueryValidator"/>
  <property name="calculator" ref="vtecCalculator"/>
</bean>

<bean id="urlFilenameViewController"
      class="org.springframework.web.servlet.mvc.UrlFilenameViewController"/>

<!-- Geçerlileyiciler -->

<bean id="singleStationQueryValidator"
      class="vtec.web.SingleStationQueryValidator"/>

<bean id="multipleStationQueryValidator"
      class="vtec.web.MultipleStationQueryValidator"/>

<!-- Görünümler -->

<bean id="excelView" class="vtec.web.TecExcelView"/>

<!-- Adres-denetleyici eşleme -->

<bean id="urlMapping"
      class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
  <property name="interceptors">
    <list>
      <ref bean="localeChangeInterceptor"/>
    </list>
  </property>
  <property name="mappings">
    <props>
      <prop key="/single.html">singleStationController</prop>
      <prop key="/singlecompare.html">singleStationDayComparisonController</prop>
      <prop key="/multiple.html">multipleStationController</prop>
      <prop key="/static/*.html">urlFilenameViewController</prop>
    </props>
  </property>
</bean>

<!-- Görünüm çözümleme -->

<bean id="resourceViewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="viewClass">

```

```

        <value>org.springframework.web.servlet.view.JstlView</value>
    </property>
    <property name="prefix"><value>/WEB-INF/jsp/</value></property>
    <property name="suffix"><value>.jsp</value></property>
    <property name="contentType"><value>text/html;charset=UTF-8</value></property>
</bean>

<bean id="beanNameViewResolver"
    class="org.springframework.web.servlet.view.BeanNameViewResolver">
    <property name="order"><value>2</value></property>
</bean>

</beans>

```

Sunum katmanı sınıflarının açıklamaları alt bölümlerde verilmiştir:

AbstractSingleStationController

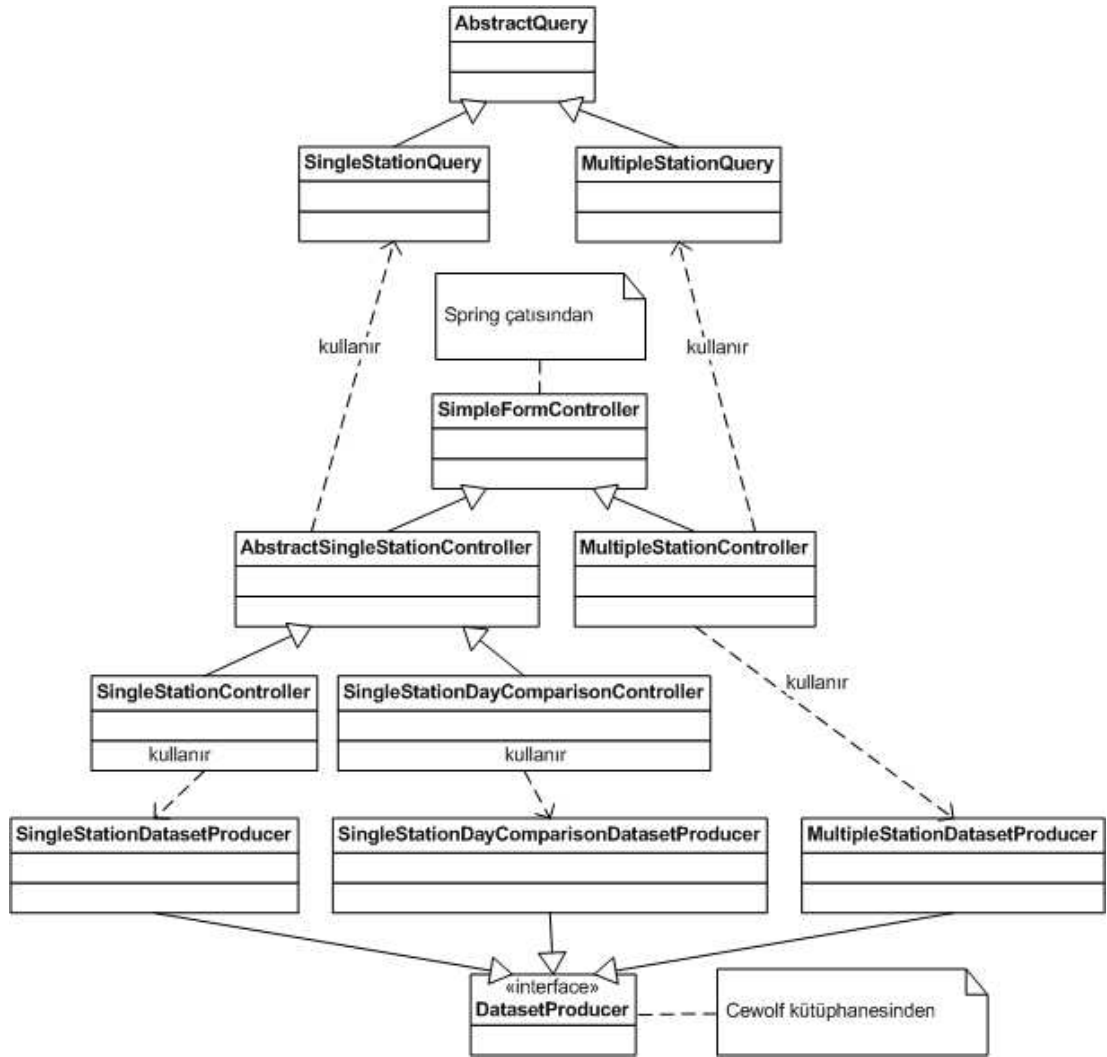
Spring çatısının SimpleFormController sınıfından türeyen bu denetleyici soyut bir sınıf olup, SingleStationController ve SingleStationDayComparisonController sınıflarının ihtiyaç duyduğu ortak işlevleri barındırır. En önemli metodu onSubmit metodu olup soyut getDatasetProducer metodunu sorgu parametreleriyle çağırıp grafik üretim için gerekli DatasetProducer arayüzünü gerçekleyen bir nesne elde eder. Bu nesneyle beraber gerekli verileri ilgili görünüm olan tec.jsp sayfasına yönlendirir.

SingleStationController

Bu denetleyici sadece soyut getDatasetProducer metodunu gerçekler. Bu metod VtecCalculator sınıfına hesaplamayı yaptırıp elde edilen Vtec nesnesinden SingleStationDatasetProducer sınıfıyla grafik için gerekli veriyi oluşturur.

SingleStationDayComparisonController

Bu denetleyici de sadece soyut getDatasetProducer metodunu gerçekler. SingleStationController'dan farkı bu işi iki farklı gün için yapmasıdır.



Şekil C.1: web paketi için sınıf şeması

MultipleStationController

Spring çatısının SimpleFormController sınıfından türemektedir.

- `onSubmit` `getDatasetProducer` metodunu sorgu parametreleriyle çağırıp grafik üretim için gerekli `DatasetProducer` arayüzünü gerçekleyen bir nesne elde eder. Bu nesneyle beraber gerekli verileri ilgili görünüm olan `tec.jsp` sayfasına yönlendirir.
- `getDatasetProducer` Bu metod `VtecCalculator` sınıfına herbir istasyon için hesaplama yaptırıp elde edilen `Vtec` nesnelere `MultipleStationDatasetProducer` sınıfıyla grafik için gerekli veriyi oluşturur.

AbstractQuery

Spring çatısının SimpleFormController sınıfının sağladığı form nesnelere ait alanların HTML formdaki bilgilere göre otomatik doldurulması özelliğinden faydalanmak için oluşturulmuş soyut form sınıfıdır. Web sayfasından girilen değerler bu sınıftan türeyen nesnelere aracılığıyla denetleyici sınıfa iletilmektedir. Önemli alanları şunlardır:

- `startDate` Hesaplamanın yapılacağı zaman diliminin başlangıç tarihidir. Gün karşılaştırması için kullanıldığında karşılaştırılacak ilk günü ifade eder.
- `endDate` Hesaplamanın yapılacağı zaman diliminin bitiş tarihidir. Gün karşılaştırması için kullanıldığında karşılaştırılacak ikinci günü ifade eder.
- `vtecByRange` Hesaplamanın sözde menzil verileriyle yapılıp yapılmayacağını belirtir.
- `vtecByPhase` Hesaplamanın faz verileriyle yapılıp yapılmayacağını belirtir.
- `includeIonex` IGS analiz merkezlerine ait kestirimlerin gösterilip gösterilmeyeceğini belirtir.
- `outputType` Çıktı türünü belirtir: Grafik veya Excel.

SingleStationQuery

`AbstractQuery` sınıfından türeyen bu sınıf, tek istasyonla ilgili hesaplamalar için kullanılmakta olup, `AbstractQuery` sınıfındaki alanlara ek olarak şu alanları barındırır:

- `receiverCode` İstasyon kodu bilgisini içerir.
- `actionId` Tek gün için mi veya gün karşılaştırması için mi hesaplamanın yapıldığı bilgisini içerir.

MultipleStationQuery

`AbstractQuery` sınıfından türeyen bu sınıf, birden fazla istasyonla ilgili hesaplamalar için kullanılmakta olup, `AbstractQuery` sınıfındaki alanlara ek olarak şu alanları barındırır:

- `receiverCodes` Hesaplamanın yapılacağı istasyonların kodlarını içerir.

SingleStationDatasetProducer

Cewolf kütüphanesinin `DatatasetProducer` arayüzünü gerçekleştirir. Tek istasyona ait `Vtec` nesnesinden Cewolf kütüphanesinin `XYSeriesCollection` nesnesini oluşturur. Bu nesne sayesinde Cewolf kütüphanesinin servleti sözde menzil ve faz ile hesaplanan TEİ değerleri ile IGS analiz merkezlerinin kestirimlerini birer seri olarak X-Y eksenli bir grafik olarak üretir.

SingleStationDayComparisonDatasetProducer

`SingleStationDatasetProducer` için anlatılan işlevin aynısını karşılaştırılacak iki gün için yapar.

MultipleStationDatasetProducer

`SingleStationDatasetProducer` için anlatılan işlevin aynısını birden fazla istasyon için yapar.

C.2 İş Katmanı

İş katmanını oluşturan paketlerden olan `calculation` paketine ait sınıf şeması Şekil C.2’de gösterilmiştir. Bu sınıfların açıklamaları alt bölümlerde verilmiştir:

VtecCalculator

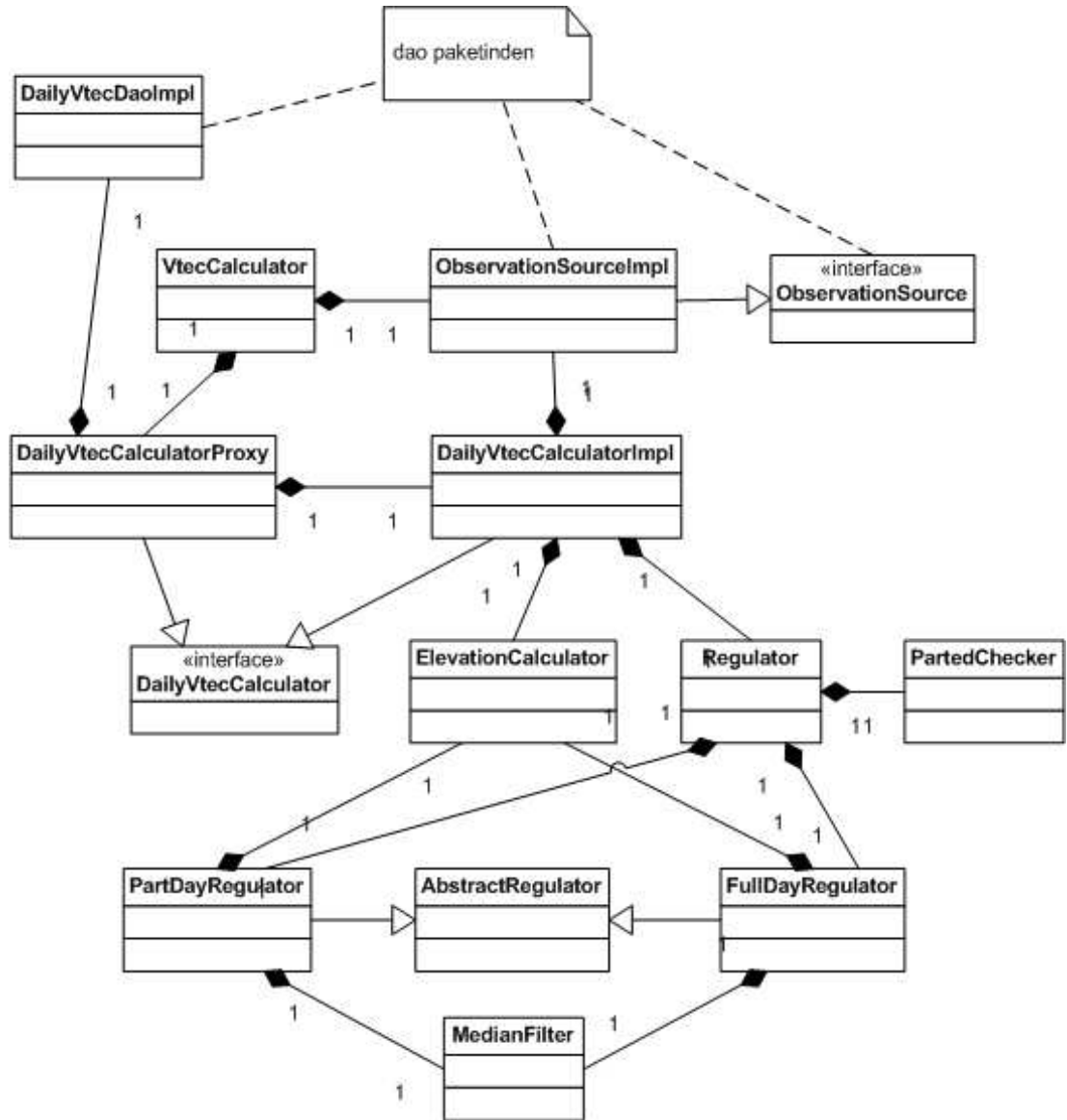
Bu sınıf TEİ hesaplama işlemlerinin ön yüzüdür. Bu sınıf aracılığıyla hesaplamanın detayları saklanarak Façade tasarım kalıbı uygulanmıştır. Verilen gün aralığı için günlük hesaplamaları `DailyVtecCalculator` arayüzünü gerçekleyen `DailyVtecCalculatorProxy` sınıfına yaptırıp toplu sonucu elde eder. `VtecCalculator` sınıfının önemli metodları şunlardır:

- `calculateVtecByRange` Başlangıç, bitiş tarihlerini ve istasyon kodunu alarak sözde menzil ölçümleriyle TEİ hesaplamasını gerçekleştirir.
- `calculateVtecByPhase` Başlangıç, bitiş tarihlerini ve istasyon kodunu alarak faz ölçümleriyle TEİ hesaplamasını gerçekleştirir.
- `getIonexVtec` IGS Analiz merkezlerinin kestirimlerini getirir.
- `receiverBiasExists` Verilen zaman aralığı için istasyona ait yanlışlık bilgisi olup olmadığını kontrol eder.

DailyVtecCalculator

Günlük TEİ hesaplaması yapan sınıfların gerçekleştireceği arayüzdür.

- `calculateVtecByRange` Verilen gün ve istasyon için sözde menzil ölçümleriyle TEİ hesaplamasını gerçekleştirir.
- `calculateVtecByPhase` Verilen gün ve istasyon için faz ölçümleriyle TEİ hesaplamasını gerçekleştirir.



Şekil C.2: calculation paketi için sınıf şeması

DailyVtecCalculatorImpl

DailyVtecCalculator arayüzünü Arıkan et al. (2003)'in yöntemine göre gerçekleştirir. DailyVtecCalculator arayüzünde belirtilen metotları gerçekler. Bu işi gerçekleştirmek için ObservationSourceImpl sınıfıyla gerekli verilere ulaşır, Regulator sınıfı ile hesapladığı TEİ değerlerini düzgünleştirir. Matris işlemleri için MTJ¹ kütüphanesi kullanılmıştır.

DailyVtecCalculatorProxy

DailyVtecCalculator arayüzünü gerçekler. Hesaplamaların daha önce yapıp yapılmadığını DailyVtecDaoImpl sınıfı aracılığıyla kontrol eder, eğer yapılmışsa yapılan hesaplamayı kullanır, yapılmamışsa DailyVtecCalculatorImpl sınıfına bu hesaplamayı yaptırır. Yapılan hesaplamaları DailyVtecDaoImpl sınıfını kullanarak kaydedip, sonucu döndürür.

ElevationCalculator

Uydunun istasyona göre yükselme açısını hesaplar. Hesaplama için Position sınıfı nesnesi olarak uydu ve istasyon konumlarını alır.

Regulator

Uydu bazında hesaplanan VTEC'leri birleştirip düzgünleştirmektedir. PartedChecker sınıfı ile gözlem verilerinde kopukluklar olup olmadığını kontrol eder. Eğer kopukluk yoksa FullDayRegulator sınıfı aracılığıyla işlemi gerçekleştirir. Eğer kopukluk varsa PartDayRegulator sınıfını kullanarak işlemi gerçekleştirir.

FullDayRegulator

Gözlem verilerinde kopukluk olmadığı durumlarda uydu bazında hesaplanmış VTEC verilerini DailyPerSatelliteVtec nesnesi olarak alıp, bu verileri

¹<http://rs.cipr.uib.no/mtj/>

IONOLAB yöntemiyle düzgünleştirir. Ortanca filtre işlemini `MedianFilter` sınıfıyla yapar.

PartDayRegulator

Gözlem verilerinde kopukluk oluğu durumlarda uydu bazında hesaplanmış VTEC verilerini `DailyPerSatelliteVtec` nesnesi olarak alıp, bu verileri Arıkan et al. (2004)'in yöntemiyle düzgünleştirir. `MedianFilter` sınıfını da kullanır.

AbsractRegulator

Düzgünleştirme işlemini yapan `FullDayRegulator` ve `PartDayRegulator` sınıflarının ortak ihtiyacı olan işlevleri içerir. Bu metotlar şunlardır:

- `getWeight` Verilen yükselme açısı için ağırlık değerini döndürür.
- `narrowRange` Düzgünleştirme öncesi verileri 0 ile 1 aralığına getirir.
- `restoreRange` Düzgünleştirme sonrası verileri 0 ile 1 aralığından orijinal değerlerine getirir.
- `getToeplitz` İstenilen boyutta Toeplitz matrisi oluşturur.

PartedChecker

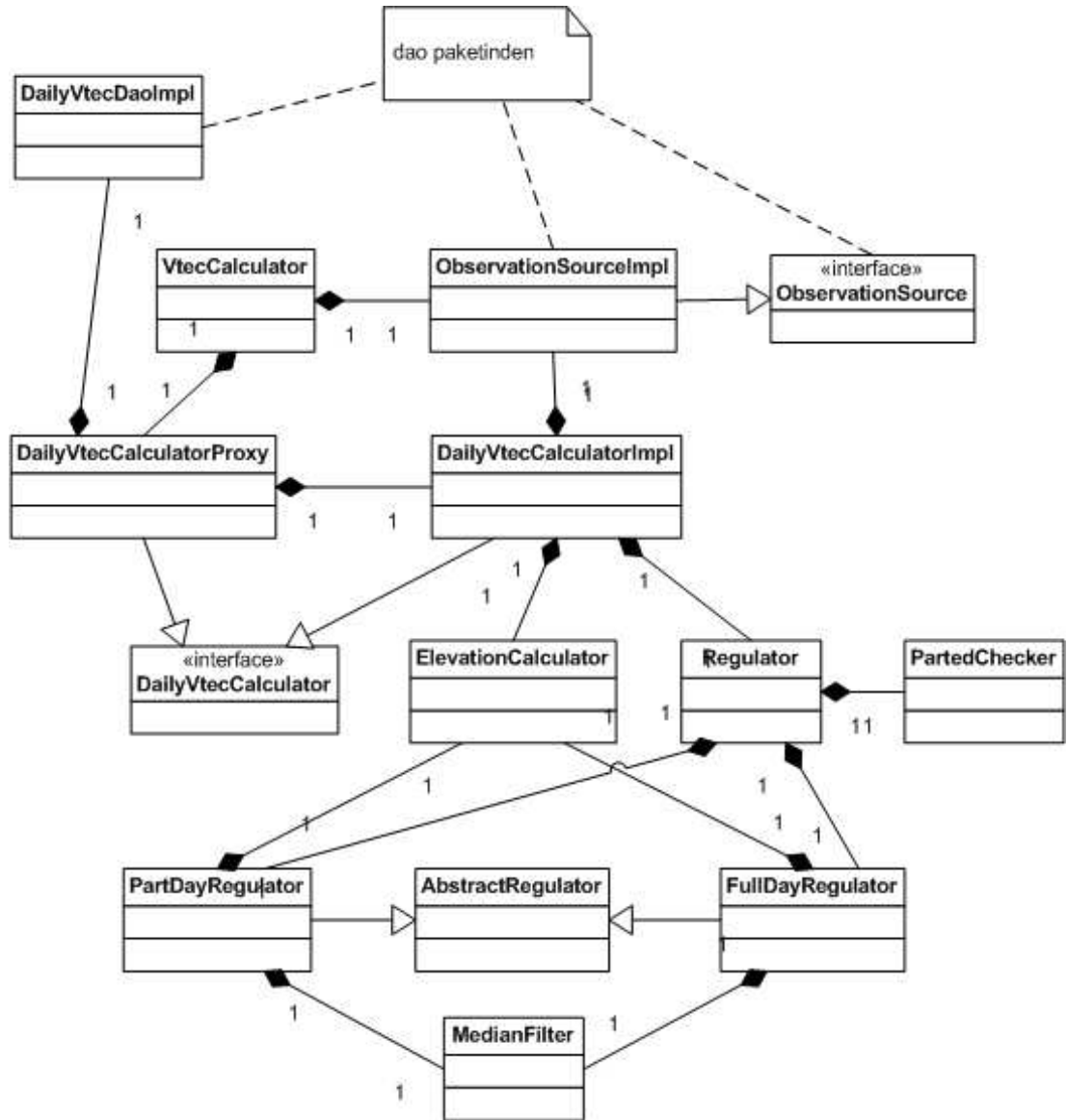
Gözlem verilerini inceleyip, verilerde kopukluk olup olmadığını belirler. Bu işlem uygulanacak düzgünleştirme yönteminin belirlenmesinde kullanılmaktadır.

MedianFilter

Verilen pencere büyüklüğüne göre aldığı değerlere ortanca filtre uygular.

C.3 Veri Erişim Katmanı

Veri erişim katmanını oluşturan `dao` paketine ait sınıf şeması Şekil C.3'de gösterilmiştir. Bu sınıfların açıklamaları alt bölümlerde verilmiştir:



Şekil C.3: dao paketi için sınıf şeması

DailyVtecDaoImpl

`DailyVtecDao` arayüzünü gerçekleştirir. Hesaplanan TEİ kestirimlerinin veritabanında saklanması sağlar. Kullanılan arayüz sayesinde farklı saklama yöntemlerinin gerektiğinde entegre edilmesi kolaylıkla mümkün olacaktır. Spring çatısının `textttJdbcDaoSupport` sınıfından türeyerek JDBC tabanlı veritabanı erişimini sağlamaktadır. Verileri saklarken TEİ hesaplama algoritmasının versiyon bilgisiyle ilişkili olarak saklanmaktadır. Bu sayede algortmada değişiklik olduğunda eski veriler kullanılmamaktadır. `DailyVtecDao` arayüzünde bulunup gerçekleştirildiği metodlar şunlardır:

- `addDailyVtec` Verilen TEİ değerlerini veritabanında saklar. TEİ değerlerini, tarihi, istasyon kodunu, hesaplanan TEİ değerlerinin zaman çözünürlüğünü, hesaplanan verilerin sözde menzille mi fazla mı hesaplandığını saklar.
- `getDailyVtec` İstenilen tarih, istasyon ve hesaplama türüne ait TEİ değerlerini veritabanından getirir.

ObservationSourceImpl

Gözlem verilerine erişim için tasarlanan `ObservationSource` arayüzünü gerçekleştirir. Verilerin internet üzerinden getirilmesi için `FileManager` sınıfını kullanır. Alınan verilerin ayıklanması için ayıklayıcı sınıfları kullanır. Bu sınıf gözlem verilerine erişimin detaylarını saklayarak bu verilere ihtiyacı olan sistemlerin işini kolaylaştırmaktadır. Gözlem verilerine farklı ortamlardan ulaşmak gerektiğinde `ObsevationSource` arayüzünü gerçekleyen başka bir sınıf geliştirilerek yeni bir gözlem erişim yöntemine kolaylıkla geçilebilir. Önemli metodları şunlardır:

- `getDailyObservation` İstenilen gün ve istasyon için gözlem verilerini `DailyObservation` nesnesi olarak elde eder.
- `getDailyPosition` İstenilen gün için uydu konum verilerini `DailyPosition` nesnesi olarak elde eder.

- `getReceiverBias` İstenilen gün ve istasyon için istasyonun yanlışlığını elde eder.
- `getSatelliteBias` İstenilen gün ve uydu için uydunun yanlışlığını elde eder.
- `getIonexData` İstenilen konum, gün ve IGS analiz merkezi için TEİ kestirimlerini elde eder.

FileManager

Gözlem dosyalarına ulaşımı sağlar. Eğer dosya daha önce indirilmemişse, internetten `WGetImpl` sınıfı ile indirir. Şıkıştırılmış dosyaları `UncompressInputStream` sınıfını kullanarak açar. İndirilen dosyanın içeriğini diğer sınıflara sunar. Önemli metodları şunlardır:

- `getIonexFile` İstenilen tarihe ve IGS analiz merkezine ait IONEX formatındaki kestirim haritasının içeriğini elde eder.
- `getRinexFile` İstenilen tarihe ve istasyona ait Sıkıştırılmış RINEX (CRX) formatındaki dosyaları indirip RINEX formatına çevirip dosyanın içeriğini elde eder.
- `getSatPosFile` İstenilen güne ait uydu konumlarını içeren dosyayı indirip içeriğini elde eder.
- `getBiasFile` İstenilen güne ait alıcı ve uydu yanlışlıklarını içeren dosyayı indirip içeriğini elde eder.

RinexParser

RINEX formatındaki dosya içeriğini ayıklayıp uydu bazındaki günlük gözlem verilerini ayıklar. Ayıklanan veriyi `DailyObservationImpl` nesnesi olarak döndürür.

Sp3Parser

SP3 formatındaki dosya içeriğini ayıklayıp GPS uydularının günlük konum bilgilerini ayıklar. Ayıklanan veriyi `DailyPosition` nesnesi olarak döndürür.

IonexParser

IONEX formatındaki dosyadan IGS analiz merkezlerine ait günlük TEİ haritalarını ayıklayıp istenilen konuma ait kestirimleri bulur. Bu işlem sırasında 2.5 derecelik enlem 5 derecelik boylam çözünürlüğüne sahip haritadan istenilen konuma ait kestirim aradeğerleme ile elde edilmektedir. Ayıklanan veriyi `DailyPosition` nesnesi olarak döndürür.

BiasParser

İstenilen GPS uydusu veya GPS alıcısı için yanlışlık bilgisini ayıklayıp döndürür.

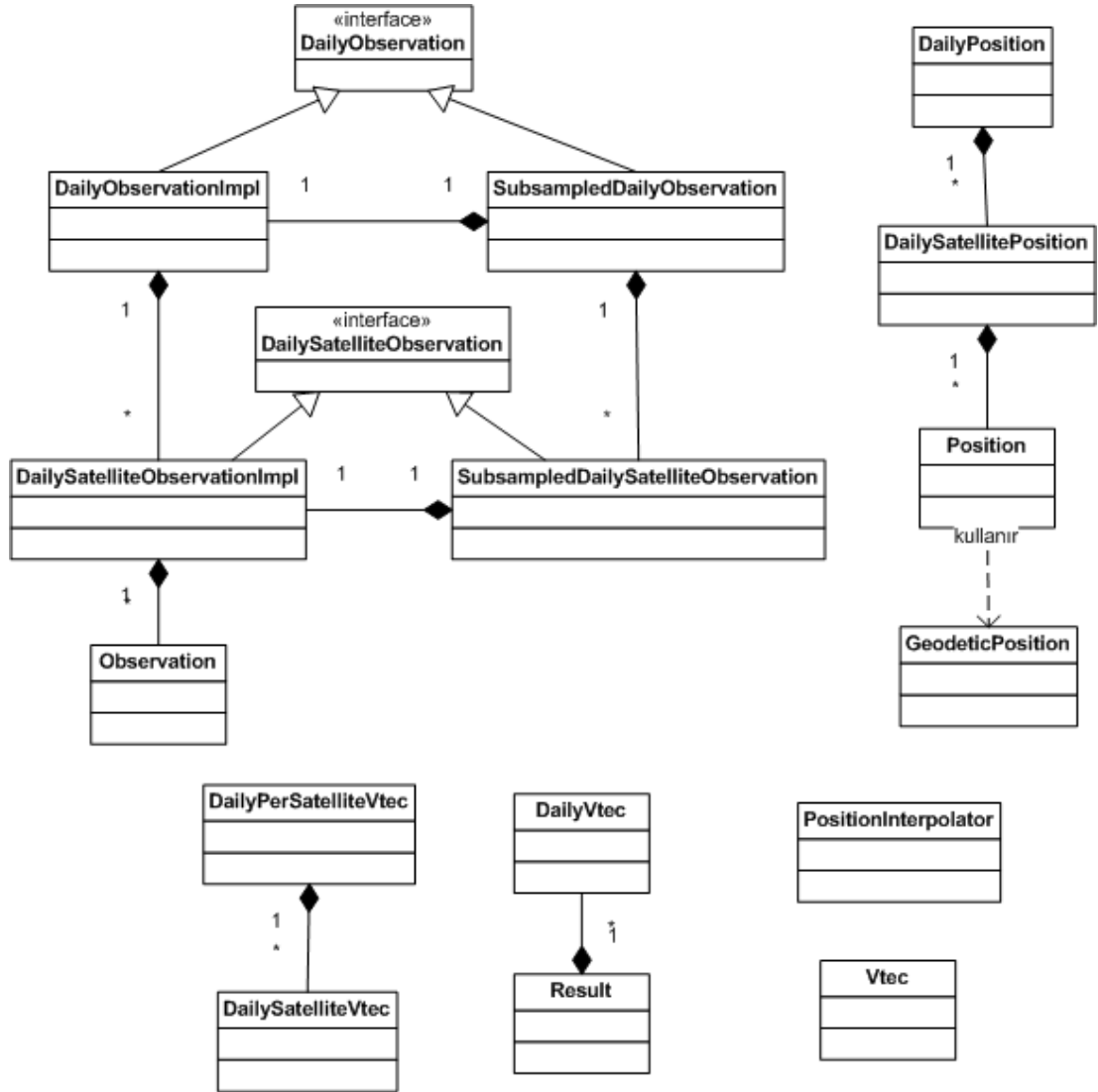
WGetImpl

Yardımlı Metin Transfer Protokolü (HTTP) ve Dosya Transfer Protokolü (FTP) protokollerini kullanarak dosyaların internetten indirilmesini sağlar.

C.4 Diğer Sınıflar

Uygulamanın gerçekleştiriminde ihtiyaç duyulan ve diğer katmanların ana işlevleriyle ilgili olmayan bazı yardımcı bileşenler de geliştirilmiştir. Bunlar iki Java paketinde toplanmıştır. `data` paketindeki sınıflar katmanlar arası veri alışverişinde kullanılan ve veri taşıyan sınıfları içermektedir. `utility` paketi ise sıkıştırılmış dosyaları açma, Jülyen takvimine göre tarih çevrimleri yapma ve dış uygulamaları çalıştırma gibi fonksiyonları içeren sınıflardan oluşmaktadır.

`data` paketine ait sınıf şeması Şekil C.4'de gösterilmiştir. `data` ve `utility` paketlerinde ait sınıfların açıklamaları alt bölümlerde verilmiştir:



Şekil C.4: data paketi için sınıf şeması

DailyObservation

Tüm uydulara ait günlük gözlemleri içeren sınıfların gerçekleştirilmesi gereken arayüzdür.

DailyObservationImpl

Her uyduya ait günlük gözlemleri `DailySatelliteObservationImpl` nesneleri olarak tutar.

SubsampledDailyObservation

`DailyObservationImpl` sınıfındaki tüm uydulara ait nesneleri zaman çözünürlüğü olarak daha az bir çözünürlükle sunar. Yani Uyarlayıcı tasarım kalıbını gerçeklemektedir.

DailySatelliteObservation

Bir uyduya ait günlük gözlemleri içeren sınıfların gerçekleştirilmesi gereken arayüzdür.

DailySatelliteObservationImpl

Bir uyduya ait günlük gözlemleri `Observation` nesneleri olarak tutar.

SubsampledDailySatelliteObservation

`DailySatelliteObservationImpl` sınıfındaki bir uyduya ait gözlemleri zaman çözünürlüğü olarak daha az bir çözünürlükle sunar.

Observation

Belirli bir ana ait sözde menzil ve faz gözlemlerini içerir.

DailyPosition

Her uyduya ait günlük konum bilgilerini `DailySatellitePosition` nesneleri olarak tutar.

DailySatellitePosition

Bir uyduya ait günlük konum bilgilerini `Position` nesneleri olarak tutar.

Position

Belirli bir ana ait konum bilgisini içerir.

GeodeticPosition

Belirli bir ana ait konum bilgisini jeodezik koordinatlarda içerir.

DailyPerSatelliteVtec

Her uyduya ait günlük TEİ kestirimlerini `DailySatelliteVtec` nesneleri olarak tutar.

DailySatelliteVtec

Bir uyduya ait günlük TEİ kestirimlerini içerir.

DailyVtec

Belirli bir zaman dilimi için TEİ kestirimlerini içerir.

Result

Belirli bir hesaplama isteği için elde edilen tüm verileri içerir. Bu veriler: sözde menzil ile hesaplanan TEİ kestirimleri, faz ile hesaplanan TEİ kestirimleri ve IGS analiz merkezlerinin TEİ kestirimleridir.

PositionInterpolator

Bir `DailyPosition` nesnesini daha hassas bir zaman çözünürlüğünde konum bilgilerine çevirmek için kullanılır.

Vtec

Bir güne ait TEİ kestirimlerini içerir.

CommandExecutor

İşletim sistemine özgü geliştirilmiş uygulamaların çalıştırılmasını sağlar. Bu sınıf temelde CRX formatındaki dosyaları RINEX formatına çeviren crx2rnx.exe uygulamasını çalıştırmakta kullanılmaktadır.

JulianDay

Meeus (1998)'da belirtilen algoritmalar kullanılarak Jülyen gününün hesaplanmasını sağlar.

UncompressInputStream

Unix tabanlı işletim sistemlerinde kullanılan “compress” sıkıştırma programıyla sıkıştırılan dosyaların açılmasını sağlar.

D. ÖRNEK GÖZLEM DOSYALARI

Bu bölümde geliştirilen uygulamanın ayıklayabildiği RINEX, SP3 ve IONEX formatlarında örnek dosyalar verilmiştir.

D.1 Örnek RINEX dosyası

Aşağıda 29 Ekim 2003 tarihine ait Zelenchukskaya istasyonu için IGS veri merkezlerinden alınmış RINEX formatındaki dosyanın bir parçası verilmiştir.

```

2          OBSERVATION DATA  GPS          RINEX VERSION / TYPE
GEOTRACER GPS Decoder Ver. 2.2      30-Oct-1903, 00:35 PGM /
RUN BY / DATE ZECK
MARKER NAME 12351M001
MARKER NUMBER AUTOMATIC          BKG-WETTZELL
OBSERVER / AGENCY T-322U          AOA SNR-8000 ACT      3.3.32.4
REC # / TYPE / VERS 290          AOAD/M_T      NONE
ANT # / TYPE
3451175.1519 3060335.5553 4391955.9693      APPROX POSITION XYZ
          0.0450      -0.0000      -0.0000      ANTENNA: DELTA H/E/N
1      1      0      WAVELENGTH FACT L1/2
5      C1      P1      P2      L1      L2      # / TYPES OF OBSERV
30      INTERVAL
2003      10      29      0      0      0.000000      TIME OF FIRST OBS
2003      10      29      23      59      30.000000      TIME OF LAST OBS
28      # OF SATELLITES
G01      809      809      809      809      809      PRN / # OF OBS
G02      1024      1024      1024      1024      1024      PRN / # OF OBS
.
.
.
G30      684      684      684      684      684      PRN / # OF OBS
G31      914      914      914      914      914      PRN / # OF OBS
END OF HEADER
3 10 29 0 0 0.0000000 0 8G03G11G14G15G20G23G25G31
23379061.96509 23379061.23109 23379065.71209 -8422855.97509 -6563267.23909
22087334.18309 22087333.90809 22087335.06109 -16096256.92109 -12542535.20009
20524187.70809 20524187.10709 20524188.72209 -18031078.95809 -14050182.11809
23027433.49208 23027432.80908 23027437.40108 -10282221.20908 -8012109.11208
24684535.55107 24684534.43407 24684537.34807 -360968.93007 -281273.99707
21900378.32909 21900377.55909 21900380.58109 -15356880.49209 -11966386.85109
24418196.02506 24418195.58906 24418198.93206 -1036156.18206 -807394.62506
21887903.56909 21887902.83209 21887905.96409 -12729699.90409 -9919245.93509
3 10 29 0 0 30.0000000 0 8G03G11G14G15G20G23G25G31
23397983.83709 23397983.30209 23397987.26909 -8323420.81109 -6485785.32909

```

22072578.11709 22072578.10909 22072579.17109 -16173799.45909 -12602957.95509
 20527136.60009 20527136.26309 20527137.92409 -18015580.76509 -14038105.59709
 23045975.13409 23045974.44809 23045978.12709 -10184788.43209 -7936187.49809
 24667395.59806 24667395.77006 24667399.41006 -451032.09806 -351452.98406
 21910811.51709 21910811.07409 21910814.04109 -15302052.79509 -11923663.98309
 24400281.18107 24400280.57407 24400285.01707 -1130305.12907 -880757.46107
 21899902.06609 21899901.46209 21899904.72209 -12666646.42109 -9870113.37309
 3 10 29 0 1 0.0000000 0 8G03G11G14G15G20G23G25G31
 23416932.66009 23416932.10409 23416936.23409 -8223844.09509 -6408193.11309
 22057872.42409 22057872.15909 22057873.00909 -16251080.34809 -12663176.82009
 20530191.61409 20530191.30309 20530192.94909 -17999526.53109 -14025595.80009
 23064557.76208 23064557.16308 23064561.92408 -10087131.27008 -7860091.05708
 24650271.63207 24650270.86607 24650273.39907 -541024.99107 -421577.26407
 21921271.54509 21921270.58109 21921273.72309 -15247086.89809 -11880833.42309
 24382367.40907 24382367.26407 24382370.49907 -1224438.59707 -954108.13807
 21911986.72909 21911986.11609 21911989.21109 -12603141.64409 -9820629.14909
 .
 .
 .
 3 10 29 23 59 0.0000000 0 8G03G11G14G15G20G23G25G31
 23497309.49108 23497309.22708 23497312.00408 -7875954.97808 -6137109.04108
 21997705.69209 21997705.73809 21997707.21109 -16554273.99709 -12899432.55109
 20543200.56109 20543200.10209 20543201.41909 -17594615.11909 -13710083.13109
 23138246.69308 23138245.85308 23138249.08308 -8660085.58108 -6748087.43508
 24583913.62307 24583912.95007 24583915.08907 -719618.21707 -560740.41407
 21960060.04409 21960059.36609 21960062.06409 -13416455.22009 -10454355.39409
 24310105.56707 24310104.96607 24310108.03907 -1601046.99507 -1247567.07407
 21962475.67109 21962474.93009 21962477.21909 -12445069.68809 -9697454.10009
 3 10 29 23 59 30.0000000 0 8G03G11G14G15G20G23G25G31
 23516397.77808 23516397.59708 23516400.68808 -7775643.60408 -6058944.37408
 21983269.76809 21983269.46809 21983270.89609 -16630137.15509 -12958546.72209
 20546790.38109 20546790.03209 20546791.34109 -17575749.73409 -13695382.84509
 23157048.60909 23157047.67509 23157050.80609 -8561284.05009 -6671099.21609
 24566863.31606 24566862.78006 24566864.67406 -809217.53606 -630558.12806
 21970643.06109 21970642.25109 21970645.08609 -13360841.22109 -10411019.81109
 24292219.30407 24292218.58707 24292221.14507 -1695041.66207 -1320809.54107
 21974992.49609 21974991.86409 21974994.37809 -12379292.47909 -9646199.13409

D.2 Örnek SP3 dosyası

Aşağıda 1 Ocak 2006 tarihine ait IGS veri merkezlerinden alınmış SP3 formatındaki dosyanın bir parçası verilmiştir.

```
#cP2006 1 1 0 0 0.00000000 96 d+D IGb00 FIT AIUB
## 1356 0.00000000 900.00000000 53736 0.00000000000000
+ 42 G01G02G03G04G05G06G07G08G09G10G11G13G14G15G16G17G18
+ G19G20G21G22G23G24G25G26G27G28G29G30R01R02R03R04R05
+ R06R07R08R17R18R21R22R23 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 4 4 4 4 4 7 5 6 6 5 4 4 4 4 5 5 4
++ 5 4 4 4 4 4 10 4 6 4 6 4 6 7 6 7 10
++ 7 10 7 6 6 8 7 7 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%c M cc GPS ccc cccc cccc cccc cccc cccc cccc cccc
%c cc cc ccc ccc cccc cccc cccc cccc cccc cccc cccc
%f 1.2500000 1.025000000 0.00000000000 0.0000000000000000
%f 0.0000000 0.000000000 0.00000000000 0.0000000000000000
%i 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0
/* CENTER FOR ORBIT DETERMINATION IN EUROPE (CODE)
/* FINAL 3-DAY ORBIT FOR SOLUTION F3_06001
/* INCLUDING PRECISE CLOCK INFORMATION
/* CLK ANT Z-OFFSET(M): II&IIA 1.023; IIR 0.000
* 2006 1 1 0 0 0.00000000
PG01 -15156.118429 18164.021936 -11829.887637 28.086093
PG02 20133.793340 -12517.384392 11531.263748 -23.263434
PG03 -11366.447141 22473.599118 7788.691386 63.024585
PG04 26109.950831 -5714.160596 1216.379816 103.695405
PG05 -11013.232276 -21731.530247 -10967.998531 447.813170
PG06 -10108.117513 -15606.106231 19125.583195 170.870847
PG07 16494.168138 -8125.986103 -18697.734760 459.606374
PG08 26002.033561 50.071985 6220.016704 -51.940054
PG09 -872.962998 -15641.294924 -21956.384909 1.771614
PG10 5783.888420 -15080.250575 20946.780862 76.348415
PG11 2597.964755 16510.768177 -20675.855554 295.573948
PG13 9325.963062 13935.499537 20539.337550 32.334897
PG14 -16100.083587 5178.413696 -20396.331913 -21.652793
PG15 -25715.531487 -1681.707743 7009.564841 542.909958
PG16 -11914.278557 9505.216918 21831.484504 19.971335
PG17 16309.020437 -3004.241310 -20755.534316 15.369519
PG18 -19253.824115 -17671.386816 -5446.551548 -212.056578
```

PG19	-5992.412318	25262.406659	-5427.094259	-24.479362
PG20	11008.920386	22233.527507	-9597.463306	-36.514558
PG21	-20832.319899	-7587.022445	15024.664073	163.664986
PG22	-19865.195038	-6643.772310	-16167.063120	59.216482
PG23	1885.873409	22933.543337	13040.396319	156.548176
PG24	23524.943175	4517.776698	-11480.049273	73.798879
PG25	-23900.353070	10820.504183	4855.962418	-78.237891
PG26	7913.693318	-24985.471974	-1494.300907	-6.919907
PG27	20796.672434	6205.745955	16146.672089	29.597304
PG28	20729.434409	9470.259184	-13664.742985	35.982499
PG29	10439.132959	-24000.095043	3948.180664	481.357268
PG30	-17452.752911	-20057.211311	-1631.816774	9.067912
PR01	17097.438168	5134.823352	18187.141144	999999.999999
PR02	5913.684541	-9396.237671	22841.883158	999999.999999
PR03	-9804.061506	-19010.251087	14004.193206	999999.999999
PR04	-18263.462336	-17646.951932	-2390.309622	999999.999999
PR05	-17091.373254	-4468.299392	-18415.691034	999999.999999
PR06	-4064.914648	11232.827274	-22662.577659	999999.999999
PR07	9630.167741	19147.544532	-13826.794754	999999.999999
PR08	18780.472838	16944.911549	3259.595616	999999.999999
PR17	2502.336244	-23888.856489	8599.302572	999999.999999
PR18	-10046.297747	-4824.881632	23006.323678	999999.999999
PR21	-2377.904818	23769.945031	-8921.260528	999999.999999
PR22	6887.814606	12430.434366	-21194.389838	999999.999999
PR23	12399.690283	-7054.978466	-21155.294965	999999.999999
* 2006	1	1	0	15
	0.00000000			
PG01	-14819.264692	16678.617754	-14176.338563	28.085947
PG02	21380.590954	-12546.091597	9030.298628	-23.262406
PG03	-11490.629763	21356.656900	10318.857393	63.026078
PG04	26145.690996	-5462.952064	-1600.756165	103.706344
PG05	-9834.870207	-21008.801631	-13277.411096	447.835485
PG06	-9412.500341	-17577.266033	17695.937413	170.911108
PG07	16109.536519	-5854.007814	-19833.883117	459.615494
PG08	25180.963464	400.374701	8898.696578	-51.939508
PG09	1430.012513	-16230.352941	-21456.312181	1.772902
PG10	7481.144671	-13308.483901	21623.843368	76.349972
PG11	372.933616	17286.683450	-20172.534060	295.576910
PG13	8558.766243	16087.928177	19270.359905	32.337020
PG14	-15456.969147	2868.509824	-21329.234399	-21.652314
PG15	-24798.626373	-2049.666324	9624.603889	542.914917
PG16	-13583.899845	7690.369534	21576.127083	19.972743
PG17	15686.871305	-674.528679	-21437.899367	15.378995
PG18	-19385.945397	-18179.834176	-2659.834961	-212.058441
PG19	-6399.854389	25617.118484	-2624.726732	-24.478867
PG20	9996.493880	21503.194685	-12069.152233	-36.513634
PG21	-19093.268893	-8104.705670	16904.463857	163.668423

PG22	-20860.122157	-8109.723333	-14137.978872	59.218548
PG23	1294.307182	24188.228312	10626.565191	156.546602
PG24	22112.107049	4978.979359	-13791.953753	73.801780
PG25	-24318.288044	10681.285001	2058.833137	-78.354613
PG26	8112.869265	-24882.155462	1451.866571	-6.926326
PG27	19076.625094	6867.097134	17935.196395	29.599190
PG28	21569.893511	10676.618969	-11351.313974	35.981890
PG29	10465.046295	-23382.120100	6778.841619	481.370912
PG30	-17024.239732	-20041.922928	-4419.258686	9.073142
PR01	18696.165496	6525.287584	16040.263549	999999.999999
PR02	7942.004686	-7382.365791	22963.243378	999999.999999
PR03	-8505.076699	-17664.897415	16406.053505	999999.999999
PR04	-18431.269476	-17594.360178	825.532670	999999.999999
PR05	-18729.108406	-5850.418333	-16317.196576	999999.999999
PR06	-6036.149229	9253.616650	-23119.366454	999999.999999
PR07	8365.048463	17785.405154	-16258.328166	999999.999999
PR08	19031.934250	16978.300026	53.961195	999999.999999
PR17	2582.098624	-24770.614040	5547.786293	999999.999999
PR18	-9392.303982	-7541.338144	22532.596180	999999.999999
PR21	-2460.624462	24691.315880	-5877.101972	999999.999999
PR22	6346.015386	14894.984163	-19723.446237	999999.999999
PR23	11707.030213	-4472.610479	-22231.212630	999999.999999
.				
.				
.				

D.3 Örnek IONEX dosyası

Aşağıda 2 Ocak 2005 tarihine ait IGS bünyesindeki JPL analiz merkezinin ürettiği IONEX formatındaki dosyanın bir parçası verilmiştir.

```

1.0          IONOSPHERE MAPS      GPS          IONEX VERSION / TYPE
GIM V3.0     JPL - GNISD          06-jan-2005 02:40  PGM /
RUN BY / DATE JPL'S GLOBAL IONOSPHERE MAPS YEAR 2005 DAY 002
COMMENT Global Ionospheric Maps (GIM) are generated on an hourly
DESCRIPTION and daily basis at JPL using data from up to 100 GPS
sites DESCRIPTION of the IGS and others institutions.
DESCRIPTION The vertical TEC is modeled in a solar-geomagnetic
DESCRIPTION reference frame using bi-cubic splines on a spherical
grid. DESCRIPTION A Kalman filter is used to solve simultaneously
for          DESCRIPTION instrumental biases and VTEC on the grid (as
stochastic   DESCRIPTION parameters).
DESCRIPTION Contact Address:  gpsiono@cobra.jpl.nasa.gov .
DESCRIPTION
2005      1      2      0      0      0          EPOCH OF FIRST MAP
2005      1      3      0      0      0          EPOCH OF LAST MAP
7200                                INTERVAL
13                                # OF MAPS IN FILE
NONE                                MAPPING FUNCTION
10.0                                ELEVATION CUTOFF

One-way carrier phase leveled to code
OBSERVABLES USED
180                                # OF STATIONS
28                                # OF SATELLITES
6371.0                             BASE RADIUS
2                                MAP DIMENSION
450.0 450.0  0.0                   HGT1 / HGT2 / DHGT
87.5 -87.5  -2.5                   LAT1 / LAT2 / DLAT
-180.0 180.0  5.0                  LON1 / LON2 / DLON
-1                                EXPONENT
TEC/RMS values in 0.1 TECU; 9999, if no value available COMMENT
DIFFERENTIAL CODE BIASES          START OF
AUX DATA
01      -2.675      0.007          PRN / BIAS / RMS
03      -2.149      0.007          PRN / BIAS / RMS
.
.
.
29      -0.570      0.007          PRN / BIAS / RMS
30      0.518       0.007          PRN / BIAS / RMS
AIS1                                -4.237      0.014          STATION / BIAS / RMS

```

```

ALBH                10.465    0.014                STATION / BIAS / RMS
.
.
.
ZAMB                -1.079    0.021                STATION / BIAS / RMS
ZIMJ                -10.904   0.014                STATION / BIAS / RMS
DIFFERENTIAL CODE BIASES                END OF
AUX DATA

                                END OF HEADER
1                                START OF TEC MAP
2005    1    2    0    0    0                EPOCH OF CURRENT MAP
87.5-180.0 180.0    5.0 450.0                LAT/LON1/LON2/DLON/H
74  74  74  73  73  73  73  73  73  72  72  72  71  71  71
70  70  69  69  68  68  67  67  66  66  65  65  64  64  63
63  63  63  63  63  63  63  63  63  64  64  64  65  65  66
67  67  68  69  69  70  70  71  71  71  72  72  72  73  73
73  73  73  74  74  74  74  74  74
85.0-180.0 180.0    5.0 450.0                LAT/LON1/LON2/DLON/H
76  76  76  76  76  76  76  76  76  75  75  75  75  74  74  73
73  72  71  70  69  68  67  66  65  63  62  61  60  59  58  57
56  56  55  55  55  55  55  56  57  57  58  59  61  62  63  64
66  67  68  69  70  71  72  73  73  74  75  75  75  76  76  76
76  76  76  76  76  76  76  76  76
.
.
.
-87.5-180.0 180.0    5.0 450.0                LAT/LON1/LON2/DLON/H
130 131 132 132 133 134 135 135 136 137 137 138 139 139 139 140
140 140 140 140 140 140 140 139 139 138 138 137 137 136 135 135
134 133 132 132 131 130 129 129 128 128 127 127 126 126 126 125
125 125 125 125 125 125 125 125 125 125 125 126 126 126 126 126
127 127 127 128 128 129 129 130 130
1                                END OF TEC MAP
2                                START OF TEC MAP
2005    1    2    2    0    0                EPOCH OF CURRENT MAP
87.5-180.0 180.0    5.0 450.0                LAT/LON1/LON2/DLON/H
63  63  63  63  62  62  62  62  61  61  61  61  60  60  60  60
60  60  59  59  59  59  59  59  59  59  59  59  59  59  59  59
59  59  59  59  59  59  59  59  60  60  60  60  60  60  60  60
60  60  60  61  61  61  61  61  62  62  62  62  62  63  63  63
63  63  63  63  63  63  63  63  63
.
.
.
                                END OF FILE

```

ÖZGEÇMİŞ

Adı Soyadı : Orhan Uğurlu

Doğum Yeri : Gerede

Doğum Yılı : 1975

Medeni Hali : Evli

Eğitim ve Akademik Durumu :

Lise 1989-1992 Özel Samanyolu Lisesi

Lisans 1992-1996 Boğaziçi Üniversitesi

Elektrik ve Elektronik Mühendisliği

Yabancı Dil :

İngilizce

İş Tecrübesi:

1998 - 2000 AYESAŞ, Sistem ve Yazılım Mühendisi

2000 - 2002 ISF, Yazılım Mühendisi

2002 - 2003 İnnova Bilişim Çözümleri A.Ş., Çözüm Uzmanı

2003 - 2004 Havelsan A.Ş., Takım Lideri

2004 - 2006 AYESAŞ, Takım Lideri